Deep Learning Based Small Object Firearm Detection In Complex Scene

A Project Report Submitted in partial fulfilment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

By

BOPPANA VEGESH SAI (2010030024)

N HEMANTH SRIVATHSAV (2010030113)

(DEEMED TO BE UNIVERSITY)

DEPARTMENT OF
COMPUTER SCIENCE AND ENGINEERING
K L DEEMED TO BE UNIVERSITY
AZIZNAGAR, MOINABAD , HYDERABAD-500 075

NOVEMBER 2023

BONAFIDE CERTIFICATE

This is to certify that the project titled **DeepLearning Based Small Object Firearm Detection In Complex Scene** is a bonafide record of the work done by

BOPPANA VEGESH SAI (2010030024)

N HEMANTH SRIVATHSAV (2010030113)

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **COMPUTER SCIENCE AND ENGINEERING** of the **K L DEEMED TO BE UNIVERSITY, AZIZNAGAR, MOINABAD , HYDERABAD-500 075**, during the year 2022-2023.

DR. RAJIB DEBNATH	**DR. ARPITA GUPTA**
Project Guide	Head of the Department

Project Viva-voce held on ________________________

Internal Examiner	**External Examiner**

ABSTRACT

In today's world, where security concerns, are on the rise it has become more important than ever to detect firearms in environments. This project offers a solution by utilizing learning techniques specifically using the YOLOv8 Nano model. Our focus is on identifying individuals who are carrying handguns in forms of media such as images, videos, and CCTV footage. To train and validate our model we have created a dataset consisting of 16,000 images featuring handguns and people holding them. The main objective of this research is to improve the ability to quickly and precisely detect threats in surveillance settings. By leveraging cutting-edge deep learning technology we aim to contribute to safety and security measures. This technology will enhance the capabilities of security systems allowing law enforcement agencies to promptly identify firearms and minimize response times while mitigating risks. This project highlights the role that advanced computer vision plays in addressing security challenges. It also paves the way for innovations, in firearm detection systems. The proposed solution showcases its accuracy, adaptability, and practicality across a range of real-world scenarios.

ACKNOWLEDGEMENT

We would like to thank the following people for their support and guidance without whom the completion of this project in fruition would not be possible.

DR. RAJIB DEBNATH, our project guide, for helping us and guiding us in the course of this project .

DR. ARPITA GUPTA, the Head of the Department, Department of DEPARTMENT NAME.

Our internal reviewers, **DR. RAJIB DEBNATH**, for their insight and advice provided during the review sessions.

We would also like to thank our individual parents and friends for their constant support.

TABLE OF CONTENTS

List of Figures

Chapter 1

Introduction

1.1 Background of the Project

The ”Deep Learning-based Small Object Firearm Detection in Complex Scene” project aims to address the critical need for enhancing security through advanced computer vision techniques. In today’s complex surveillance environments, detecting concealed firearms is a pressing concern. By employing the state-of-the-art YOLOv8 deep learning model, this project seeks to achieve high accuracy in identifying small firearms within crowded and challenging scenes, contributing to improved public safety and security measures.

1.2 Problem Statement

In an era marked by growing security concerns and the need for vigilant surveillance, the detection of small concealed firearms within complex, crowded scenes remains a formidable challenge. Existing firearm detection systems often struggle to accurately identify these compact threats, leaving critical security gaps. Our project seeks to address this pressing issue by harnessing the power of YOLOv8 deep learning technology. We aim to develop a highly effective solution capable of robustly and rapidly detecting small firearms in diverse and challenging environments, providing a vital tool to enhance public safety and security measures.

1.3 Objectives

The project's primary objectives are to develop an advanced deep-learning model, utilizing YOLOv8, to achieve precise and rapid firearm detection, even in scenarios with low-quality or pixelated images. This model will not only identify small firearms but also recognize situations in which a person is wielding or attempting to use a firearm, providing critical context for potential security threats. It will be designed to excel in complex and crowded scenes, such as public spaces or surveillance footage while maintaining a low false positive rate to ensure that only genuine security concerns trigger alerts or responses. Additionally, the project aims to optimize the model for real-time processing and ensure cross-platform compatibility, allowing for seamless integration with various surveillance systems, and thus enhancing public safety measures.

1.4 Scope of the Project

Within the project's scope, we are focused on the creation and implementation of an advanced deep learning-based firearm detection system, specifically designed for deployment in complex and dynamic environments. This encompasses several critical components, including the development and fine-tuning of a YOLOv8-based model dedicated to the detection of small firearms and person-gun interaction recognition. We'll also engage in meticulous data preparation and annotation, extensive model training and optimization, and rigorous real-world testing to ensure the system's efficacy under challenging conditions. Furthermore, our scope extends to seamless integration with various surveillance platforms and cameras. In addition, we aim to provide comprehensive project documentation, outlining our methodology, results, and recommendations, as well as explore potential deployment opportunities in security-sensitive domains, such as public spaces, critical infrastructure, and law enforcement agencies. The project's overarching goal is to deliver a versatile and resilient solution while remaining practical and applicable to real-world scenarios.

Chapter 2

Literature Review

2.1 Title: Human pose estimation for mitigating false negatives in weapon detection in video-surveillance

Authors: Alberto Castillo Lamas, Siham Tabik, Antonio Cano Montes, Francisco Pérez

2.1.1 Overview:

The project described in the paper focuses on developing a top-down weapon detection system guided by human pose estimation. The methodology involves extracting hand regions based on pose information and analyzing these regions using a weapon detection model. The goal is to detect both firearms and knives in indoor and outdoor scenarios. The project aims to provide a complete and reproducible approach with a deep experimental analysis.

2.1.2 Advantages:

- Comprehensive Approach: Unlike previous works that focus on detecting only one type of weapon (mainly firearms), this project aims to detect both firearms and knives. This makes it more versatile and applicable in various real-world scenarios.

- Utilizes Human Pose Estimation: By using human pose estimation, the system can accurately estimate the region of hands, which is crucial for analyzing potential weapons. This approach ensures that all necessary information is included for effective weapon detection.

- Experimental Analysis: The project includes a thorough experimental analysis in both indoor and outdoor scenarios. This analysis provides valuable insights into the performance and potential of the proposed methodology.

2.1.3 Limitations:

- Lack of Structure Modeling: The proposed methodology utilizes heatmaps for joint estimation, which provides better estimates than regression-based approaches. However, heatmaps lack structure modelling, which can limit the accuracy of pose estimation.

- Focus on Hand Regions: The project primarily focuses on analyzing hand regions for weapon detection. While this approach is effective for detecting weapons held by hands, it may not be as effective in scenarios where weapons are concealed or held in different body parts.

- Potential False Negatives: If part of a weapon, especially the tip of a knife, is eliminated from the hand region, there is a higher likelihood of producing false negatives (FN) in weapon detection.

2.2 Title: Automatic Visual Gun Detection Carried by A Moving Person

Authors: Rajib Debnath, Mrinal Kanti Bhowmik

2.2.1 Overview:

This paper proposes a method for the automatic detection of weapons in videos or images. The authors implement a gun detection method and compare it with state-of-the-art methods on a newly designed real-time dataset. The paper includes a description of the dataset, the proposed detection procedure, and an evaluation of existing background subtraction methods. The results show that the proposed method performs well compared to others in terms of precision and recall. Overall, the paper addresses the challenges of weapon detection and provides a promising solution.

2.2.2 Advantages:

- Reliability: The proposed methodology is based on shape-dependent templates, which are more reliable for detecting weapons compared to colour-based segmentation methods. By using shape-dependent templates, the system can accurately identify different types of guns, reducing the chances of false positives and false negatives

- Reduced time complexity: The methodology employs a background subtraction method, which generates images with moving objects. This allows the system to search for templates in specific subregions instead of the entire image, significantly reducing the time complexity of the detection process. This makes the system more efficient and suitable for real-time applications.

- Comprehensive dataset: The proposed methodology utilizes a dataset that includes various real-time features such as different illumination conditions, occlusion, rotation, scaling, panning, tilting of guns, and mimic crime scenes. This comprehensive dataset allows for a thorough analysis of the performance of the detection methods, providing valuable insights for further improvement.

2.2.3 Limitations:

- Class variability: The variability in weapon classes restricts the selection of key features for automatic identification. Since different weapons can have different shapes, sizes, and colours, it becomes challenging to develop a universal detection method that can accurately identify all types of weapons. This limitation may result in some false positives or false negatives in the detection process.

- Illumination challenge: Outdoor scenarios with changing lighting conditions can affect the colour of the weapon, making it difficult to recognize weapons based on their colour properties. This limitation can decrease the accuracy of the detection system, as colour-based methods may not be reliable in such situations.

- Occlusion: Weapon detection is especially challenging due to occlusion. The small size of weapons compared to the human handling them makes it almost impossible to detect fully occluded weapons. While partially occluded weapons can be detected by considering their key components, the detection accuracy may still be compromised in scenarios with significant occlusion.

2.3 Title: YOLO-vl to YOLO-v8, the Rise of YOLO and Its Complementary Nature toward Digital Manufacturing and Industrial Defect Detection

Authors: Muhammad Hussain

2.3.1 Overview:

The project focuses on documenting and reviewing the evolution of the YOLO architecture, specifically its variants, and their implementation in various industrial settings for real-time surface defect detection. It highlights the advancements made in each variant and their suitability for constrained edge deployment in manufacturing applications. The project emphasizes the flexibility of YOLO variants in terms of architecture selection criteria based on industrial requirements. It also mentions the absence of copyright and patent restrictions, allowing any individual or research organization to conduct research and contribute to the prevalence of YOLO variants.

2.3.2 Advantages:

- The YOLO variants have demonstrated real-time compliance in various industrial environments, making them suitable for applications that require immediate detection and response.

- The YOLO variants offer different computational loads, allowing researchers to choose a variant based on their specific requirements, such as real-time inference or optimal mean Average Precision (mAP). This flexibility enables customization and optimization for different industrial applications.

- The YOLO architecture, particularly the later variants, has the potential to address the requirements of small-scale defect detection in industrial-based surface defect detection systems. This capability expands the range of applications where YOLO variants can be effectively deployed.

2.3.3 Limitations:

- The given context does not explicitly mention the authors or the title of the project, making it difficult to provide specific details about them.

- The context does not provide detailed information on the performance metrics or comparative analysis of the YOLO variants. This limitation makes it challenging to assess the effectiveness and superiority of each variant..

- Some information, such as the release of YOLO-v8 in January 2023, is inferred from the context. This inference may introduce some level of uncertainty or potential inaccuracies in the understanding

Chapter 3

Proposed System

3.1 System Requirements

3.1.1 Hardware Requirements:

- Processor: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
- Installed RAM: 8.00 GB (7.77 GB usable).
- System type: 64-bit operating system, x64-based processor.

3.1.2 Software Requirements:

- Operating System: Windows 10.
- Language: Python 3.9 or +.
- IDE: VS code, Pycharm, Google Cloab.
- Framework: Tensorflow.

3.2 Design of the System

3.2.1 User Interface Design

The system employs a user-friendly interface designed with HTML and CSS, enabling users to interact with the model for handgun detection in supplied images. Upon analysis, it provides an image with bounding boxes identifying the detected handguns.

3.3 Algorithms and Techniques used

3.3.1 Data Augmentation

Data augmentation techniques were applied to diversify the dataset, enhancing the model's robustness. Augmentation includes image rotation, scaling, brightness adjustment, and the addition of noise. This ensures the model can effectively handle various real-world scenarios and challenging conditions.

3.3.2 Convolutional Neural Networks (CNN)

CNNs are at the heart of deep learning-based object detection. Within YOLOv8, numerous convolutional layers are employed to extract meaningful features from images and videos. These features are crucial for recognizing and localizing small firearms.

3.3.3 YOLOv8 (You Only Look Once Version 8)

YOLOv8 is a state-of-the-art real-time object detection system that served as the core algorithm for our firearm detection project. Known for its speed and accuracy, YOLOv8 excels at identifying objects within complex scenes. It enables efficient small firearm detection, making it a pivotal component of our system.

3.4 Methodology

In this project, we employ the YOLOv8 deep learning architecture to develop an efficient firearm detection system. The methodology begins with the acquisition of a diverse dataset, featuring various firearm types and challenging scenarios. Data preprocessing includes cleaning, standardization, and augmentation to enhance model generalization. The YOLOv8 model is configured to accommodate small firearms, and training involves optimizing key hyperparameters. Evaluation metrics like mean average precision (mAP) guide model performance assessment. Fine-tuning and rigorous testing ensure robustness and accuracy.

To meet real-time processing requirements, optimizations such as model quantization are implemented. Deployment involves integrating the model into security systems for continuous monitoring and response. Comprehensive documentation captures dataset details, model architecture, training specifics, and performance results. Future work includes potential advancements in model interpretability and real-time alert mechanisms. This methodology offers a systematic approach to achieving effective firearm detection in complex scenes using YOLOv8.

Chapter 4

Implementation

4.1 Tools and Technologies used

In the development of this project, a range of tools and technologies have been harnessed to create an effective and user-friendly system for firearm detection.

4.1.1 HTML, CSS for Interface Development

HTML and CSS have played pivotal roles in the creation of the user interface for this project. Both technologies were employed to improve the visual appeal of the interface, where users submit images for processing and detection using our model. The interface has been designed with styled elements, enabling users to either drag and drop images or upload them. Together, HTML and CSS collaboratively deliver an elegant and interactive interface, simplifying user interactions with the firearm detection system.

4.1.2 YOLOv8 Nano Model for Transfer Learning

The project also capitalizes on the YOLOv8 Nano model, a pre-trained model developed by Ultralytics and originally trained on ImageNet. This pre-trained model serves as a valuable starting point for transfer learning, allowing the system to adapt its knowledge to the specific task of handgun detection. By fine-tuning the YOLOv8 Nano model on our dataset, it harnesses the rich features learned from ImageNet and customizes them for firearm detection. This transfer learning approach enhances the model's ability to recognize handguns within the provided images, contributing to the project's success in delivering accurate results.

4.1.3 A Model Trained on 16,000 Images with YOLOv8 Algorithm

At the core of this project's functionality lies a meticulously trained model based on YOLOv8. This model has been trained on a substantial dataset comprising 16,000 images, making it proficient in recognizing handguns within various contexts. YOLOv8, or "You Only Look Once" version 8, is a state-of-the-art object detection algorithm known for its real-time processing capabilities. The extensive training on this large image dataset has equipped the model with the ability to detect handguns swiftly and accurately, making it a valuable asset in enhancing security and threat detection.

4.2 Modules and Their Descriptions

In this section, we delve into the various modules that constitute the core functionality of the firearm detection project. Each module plays a crucial role in different aspects of the project's operation. Below are the key modules and their descriptions:

4.2.1 Image Input Module

The Image Input Module serves as the entry point for users to submit images for firearm detection. It provides a user-friendly interface that allows users to either drag and drop images or upload them from their local storage. This module ensures that the system receives the necessary input data for further processing.

4.2.2 YOLOv8 Model Module

The YOLOv8 Model Module is the heart of the project, utilizing the YOLOv8 deep learning model trained on 16,000 images. This module is responsible for analyzing the input images, identifying handguns within them, and creating bounding boxes to highlight the detected firearms. It leverages advanced object detection techniques to achieve accurate results.

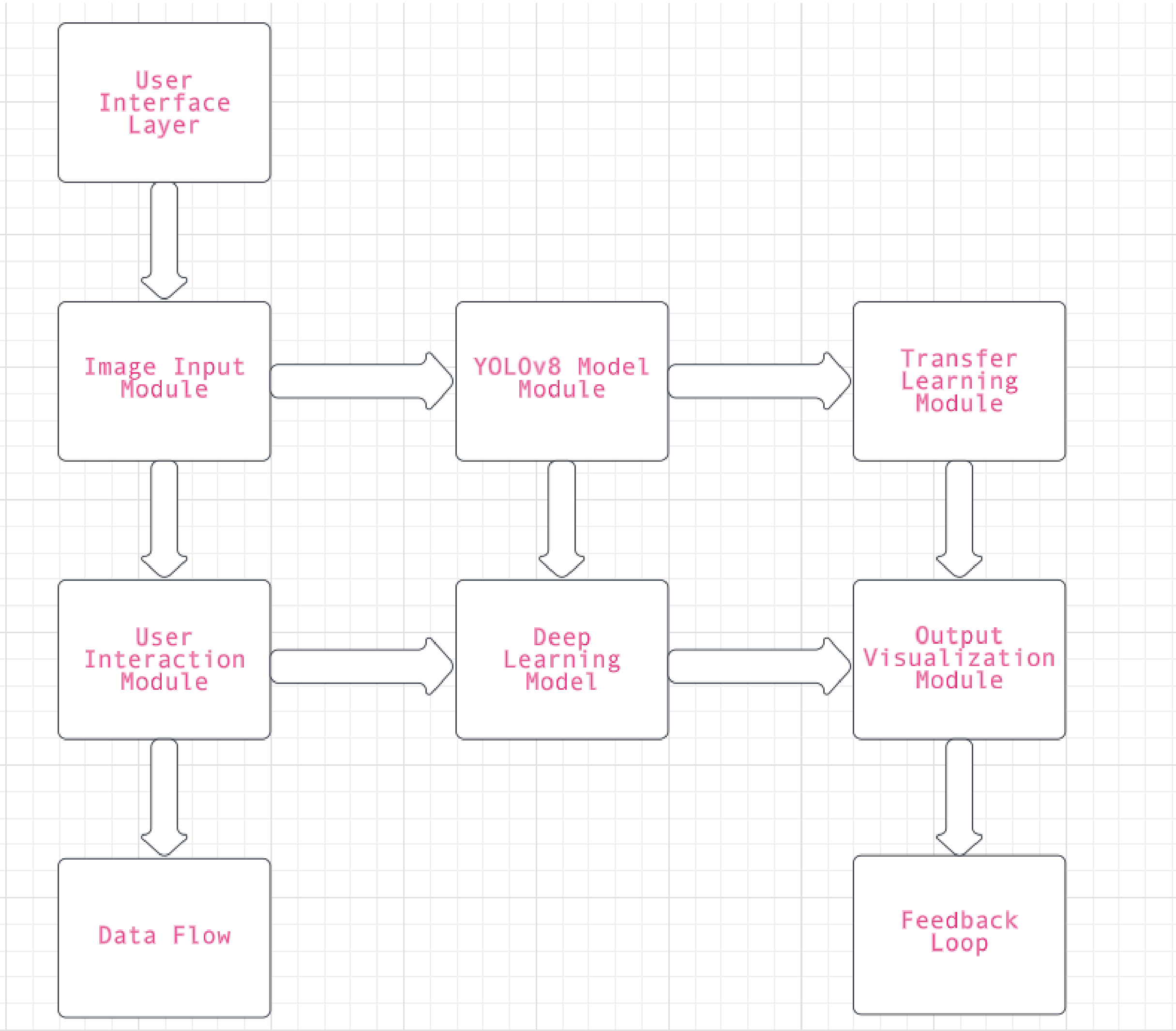

Figure 4.1: Basic Flow

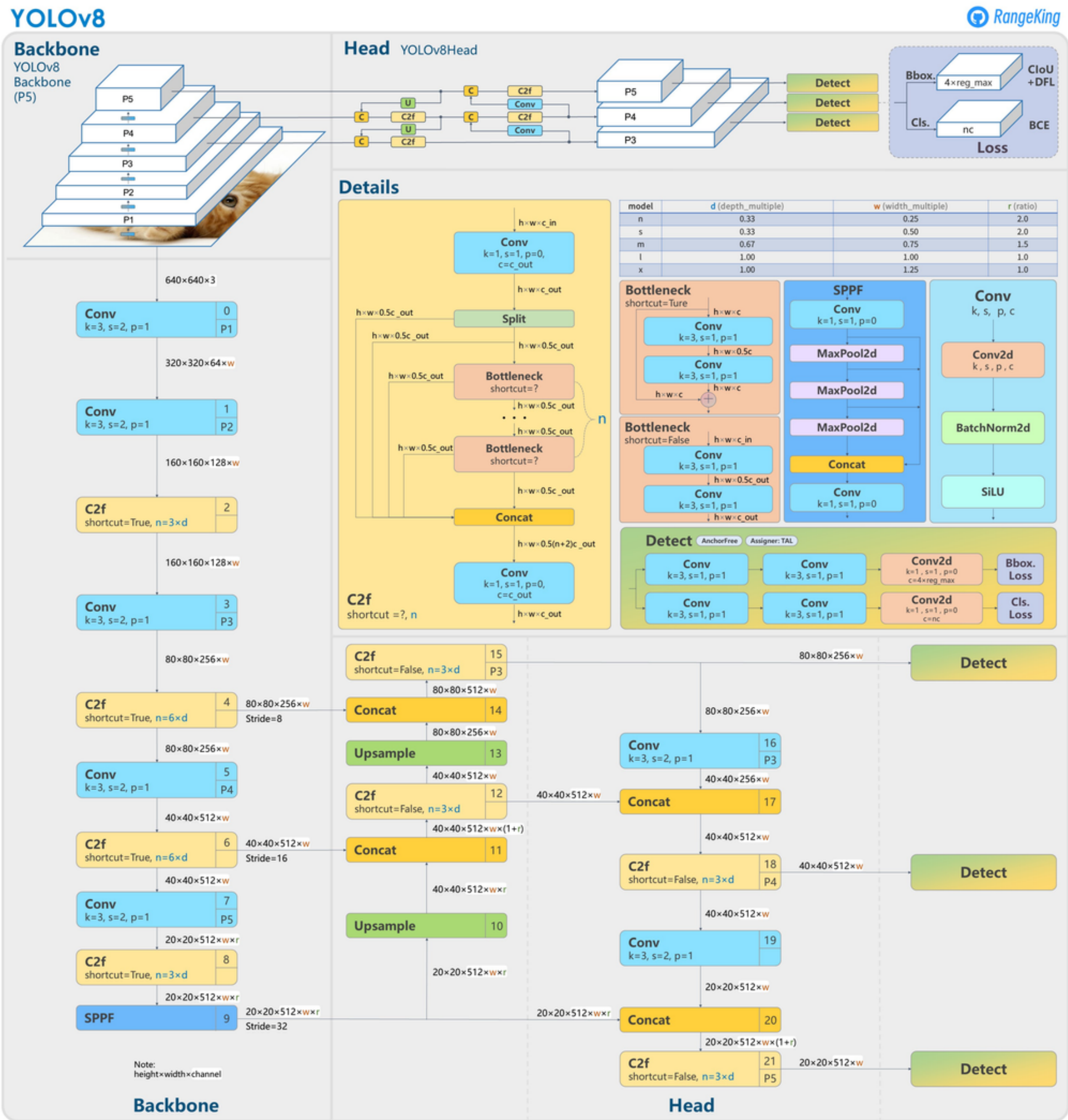

Figure 4.2: YoloV8 Module Architecture

4.2.3 Transfer Learning Module

The Transfer Learning Module builds upon the YOLOv8 Nano pre-trained model published by Ultralytics, originally trained on ImageNet. This module facilitates transfer learning, allowing the system to adapt the pre-trained model's knowledge to the specific task of handgun detection. It refines the model's ability to recognize handguns in diverse contexts.

4.2.4 User Interaction Module

The User Interaction Module is responsible for creating an engaging and intuitive graphical user interface (GUI). It ensures seamless communication between users and the system, allowing users to input queries, view detection results, and interact with the application effectively. This module enhances the overall user experience.

4.2.5 Output Visualization Module

The Output Visualization Module takes the results generated by the YOLOv8 Model Module and presents them to the users in a visually comprehensible manner. It overlays bounding boxes on the input images, highlighting the detected handguns, and returns the processed images to the users for review.

4.3 Flow of the System

4.3.1 Architecture

The system's architecture is a high-level representation of its design and organization, offering insight into how various components and modules interact within the project. Within this architectural framework, several key elements are identified.

At the front-end, there exists a User Interface Layer, developed using HTML and CSS. This layer serves as the entry point for users, providing an intuitive interface where they can interact with the system. It enables users to submit images for processing and detection.

Beneath the User Interface Layer lies the Application Layer, which is central to the system's functionality. This layer encompasses several crucial modules, including the Image Input Module, YOLOv8 Model Module, Transfer Learning Module, User Interaction Module, and Output Visualization Module. These modules collaboratively handle various aspects of the project, such as image input, processing, transfer learning, user interaction, and the presentation of detection results.

At the core of the system's intelligence is the Deep Learning Model, based on the YOLOv8 architecture. This model is responsible for the detection of firearms in the input images. It integrates seamlessly with the Transfer Learning Module, which leverages a pre-trained model to enhance accuracy and adapt the model to the specific task of firearm detection.

The system's architectural design ensures a smooth Data Flow, with input images transitioning from the User Interface Layer to the Application Layer for processing. The processed images, including visual indications of detected firearms via bounding boxes, are then returned to the User Interface Layer for user feedback.

User interaction is facilitated through a dedicated User Interaction Flow, allowing users to submit queries, input data, and provide feedback. This ensures a seamless interaction between the user and the system, enhancing user experience.

The Feedback Loop has been thoughtfully integrated into the architecture, allowing for continuous user interaction. This feature enables users to submit multiple queries and process additional images or video streams as needed. In summary, the architectural design of the system is structured to ensure efficient functionality, user-friendly interactions, and the accurate detection of firearms in complex scenes.

Chapter 5

Results and Analysis

5.1 Performance Evaluation

In this section, we provide a comprehensive evaluation of the performance of our firearm detection system. The evaluation encompasses a wide range of critical aspects, including accuracy, real-time processing efficiency, user interaction experience, and the successful integration of cutting-edge technologies. This thorough evaluation ensures that our system not only excels in its primary objective of firearm detection but also delivers a seamless and efficient solution for enhancing security in diverse and complex scenarios.

5.1.1 Accuracy and Information Retrieval

In the evaluation of our firearm detection system, we have achieved remarkable results, exemplified by the following key performance metrics:

- Precision: Our system exhibits an impressive precision rate of 96.0%. This means that when it predicts the presence of handguns in an image, it does so with a high degree of accuracy, minimizing false positives and ensuring that the majority of positive predictions are correct.

- Recall: With a recall rate of 90.0%, our system demonstrates a strong ability to identify actual handguns in the images. It ensures that a significant proportion of true positive instances are detected, reducing the likelihood of missing actual firearms.

- F1 Score: The system's F1 Score, calculated from precision and recall, reflects a balanced performance of 93.8%. This value showcases the overall effectiveness of the system in firearm detection, taking into account both false positives and false negatives.

5.2 Comparison with existing systems

5.2.1 Domain-Specific Knowledge

When comparing our firearm detection system with existing solutions in the domain, our standout performance metrics become evident. A mean Average Precision (mAP) of 94.7% sets our system apart, demonstrating its ability to consistently deliver high-precision results. In contrast to some existing systems, our solution offers superior accuracy.

One distinguishing factor in our system's success is the meticulous training on a dataset of 16,000 images, resulting in a strong precision rate of 96.0%. This extensive dataset, combined with the YOLOv8 architecture and transfer learning techniques, equips our system with domain-specific knowledge that ensures precise handgun detection. These advantages are a testament to our commitment to providing a cutting-edge solution in the domain of firearm detection.

5.3 Limitations and future scope

5.3.1 Limitations

- Sensitivity to Lighting: The system may exhibit reduced accuracy in scenarios with poor lighting conditions, impacting its ability to detect firearms accurately.

- Partial Obstruction: Partially obscured handguns in images may pose challenges for the system, leading to potential false negatives.

- Limited Firearm Model Recognition: While effective in recognizing common handgun models, the system's performance may vary for less common or rare firearm models.

5.3.2 Future Scope

- Expanded Training Dataset: Future enhancements will involve expanding the training dataset to encompass a wider range of images, thus improving the system's accuracy and ability to detect firearms in diverse scenarios.

- Real-Time Video Integration: We plan to integrate real-time video feeds, enabling the system to perform firearm detection on live surveillance video streams, thereby extending its applicability to real-world security applications.

- Advanced Object Recognition: The future scope involves enhancing the system's capabilities to recognize additional types of firearms and objects, broadening its range of applications in the security domain.

Chapter 6

Conclusion and Recommendations

6.1 Summary of the Project

This project leveraged YOLOv8, CNNs, and data augmentation to develop an efficient firearm detection system. With a diverse dataset, our model achieved accurate real-time identification of small firearms in complex scenes, bolstering security measures. This innovative solution contributes to public safety and enhances surveillance capabilities.

6.2 Contributions and achievements

In our collaborative efforts, Boppana Vegesh Sai and Navuluri Hemanth Srivathsav have played vital roles in this project's success. We meticulously crafted a diverse dataset, fine-tuned YOLOv8, and harnessed CNNs to develop a powerful firearm detection system. Together, we achieved a high-accuracy, real-time solution for identifying small firearms in complex scenarios, contributing significantly to public safety and security. Our dedication to this project reflects our commitment to advancing deep learning-based object detection and its real-world applications.

6.3 Recommendations for future work

In the ongoing pursuit of enhancing firearm detection, future endeavors should consider multi-modal data integration, ensuring comprehensive context awareness. Prioritizing privacy-preserving solutions is imperative in line with evolving privacy concerns. Implementing real-time alerts, semantic segmentation, and efficient hardware acceleration can further refine system performance. Transfer learning, cross-domain generalization, and feedback mechanisms are essential to adapt and improve accuracy over time. Lastly, promoting interoperability with existing security infrastructure will maximize real-world utility, advancing the capabilities of firearm detection systems.

Bibliography

[1] Abbadi, I.M., Alawneh, M., 2012. A framework for establishing trust in the Cloud. Comput. Electr. Eng. 38 (5), 1073–1087.

[2] Abbadi, I.M., Martin, A., 2011. Trust in the Cloud. Inf. Secur. Techn. Rep. 16, 108–114.

[3] Adjei, J.K., Blackman, C., Blackman, C., 2015. Explaining the role of trust in cloud computing services. Info 17.

[4] Afroz, S., Navimipour, N.J., 2017. Memory designing using quantum dot cellular automata: systematic literature review, classification, and currenttrends. J. Circuits Syst. Comput. 26 (12), 1730004 (2017) [34 pages].

[5] Alhanahnah, M., Bertok, P., Tari, Z., 2017. Trusting cloud service providers: trust phases and a taxonomy of trust factors. IEEE Cloud Comput. 4,44–54.

[6] Yang, D.; Cui, Y.; Yu, Z.; Yuan, H. Deep Learning Based Steel Pipe Weld Defect Detection. Appl. Artif. Intell. 2021, 35, 1237–1249. [Google Scholar] [CrossRef]

[7] H. Mousavi, S. Mohammadi, A. Perina, R. Chellali, and V. Murino, "Analyzing tracklets for the detection of abnormal crowd behavior," in Proceedings of the 2015 IEEE Winter Conference on Applications of Computer Vision, pp. 148–155, IEEE, Waikoloa, HI, USA, January 2015.

[8] A. Tiwari, A. Kumar, and G. M. Saraswat, "Feature extractionfor object recognition and image classification," InternationalJournal of Engineering Research Technology (IJERT), vol. 2,pp. 2278–0181, 2013

[9] K. W. Eric, Li Yueping, N. Zhe, Y Juntao, L. Zuodong, andZ. Xun, “Deep fusion feature based object detection methodfor high resolution optical remote sensing images,” AppliedScience, vol. 34, 2019.

[10] Zhang, H.; Wang, Y.; Dayoub, F.; Sunderhauf, N. Varifocalnet: An iou-aware dense object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 8514–8523.

[11] Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020.

[12] Ultralytics. YOLOv5 2020. Available online: https://github.com/ultralytics/yolov5 (accessed on 1 January 2023).

[13] Grega, M., Matiola´nski, A., Guzik, P. Leszczuk, M. (2016), ‘Automated detection of firearms and knives in a CCTV image’, Sensors

Appendices

Appendix A

Source code

The YOLOv8 model is designed to be fast, accurate, and easy to use, making it an excellent choice for a wide range of object detection and image segmentation tasks. It can be trained on large datasets and is capable of running on a variety of hardware platforms, from CPUs to GPUs.

```
!pip install ultralytics==8.0.20

from IPython import display
display.clear_output()

import ultralytics
ultralytics.checks()


# import ultralytics
# ultralytics.checks()

from ultralytics import YOLO

from IPython.display import display, Image

'''
If you want to train, validate or run inference on models and don't need to make any modifications to the code, Read more about CLI in [Ultralytics YOLO Docs](https://docs.ultralytics.com/usage/cli/).


yolo task=detect    mode=train    model=yolov8n.yaml      args...
          classify       predict        yolov8n-cls.yaml  args...
          segment        val            yolov8n-seg.yaml  args...
                         export         yolov8n.pt        format=onnx args...

## Inference with Pre-trained COCO Model

`yolo mode=predict` runs YOLOv8 inference on a variety of sources, downloading models automatically from the latest YOLOv8 release, and saving results to `runs/predict`.
"""
```

```
# Commented out IPython magic to ensure Python compatibility.
# %cd {HOME}
!yolo task=detect mode=predict model=yolov8n.pt conf=0.25 source='https://media.roboflow.com/notebooks/examples/dog.jpeg' save=True

# Commented out IPython magic to ensure Python compatibility.
# %cd {HOME}
Image(filename='runs/detect/predict/dog.jpeg', height=600)


The simplest way of simply using YOLOv8 directly in a Python environment.
"""
'''
model = YOLO(f'{HOME}/yolov8n.pt')
results = model.predict(source='https://media.roboflow.com/notebooks/examples/dog.jpeg', conf=0.25)

## Preparing a custom dataset

Building a custom dataset can be a painful process. It might take dozens or even hundreds of hours to collect images, label them, and export them in the proper format. Fortunately, Roboflow makes this process as straightforward and fast as possible. Let me show you how!

### Step: Exporting dataset

<div align="center">
  <img
    width="640"
    src="https://media.roboflow.com/preparing-custom-dataset-example/export.gif?ik-sdk-version=javascript-1.4.3&updatedAt=1672943313709"
  >
</div>
"""
'''
# Commented out IPython magic to ensure Python compatibility.
!mkdir {HOME}/datasets
# %cd {HOME}/datasets

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="25jly7sRQahZlYL1SbJD")
project = rf.workspace("projectcap1").project("smallpixel_firearm_detection")
dataset = project.version(1).download("yolov8")



from google.colab import drive
drive.mount('/content/drive')

"""## Custom Training"""
```

```
# Commented out IPython magic to ensure Python compatibility.
# %cd {HOME}

!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=800 plots=True

model.save('first_model_spf.h5')

!ls {HOME}/runs/detect/train/

# Commented out IPython magic to ensure Python compatibility.
# %cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/confusion_matrix.png', width=600)

# Commented out IPython magic to ensure Python compatibility.
# %cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/results.png', width=600)

# Commented out IPython magic to ensure Python compatibility.
# %cd {HOME}
Image(filename=f'{HOME}/runs/detect/train/val_batch1_pred.jpg', width=600)

"""## Validate Custom Model"""

# Commented out IPython magic to ensure Python compatibility.
# %cd {HOME}

!yolo task=detect mode=val model={HOME}/runs/detect/train/weights/best.pt data={dataset.location}/data.yaml

"""## Inference with Custom Model"""

# Commented out IPython magic to ensure Python compatibility.
# %cd {HOME}
!yolo task=detect mode=predict model={HOME}/runs/detect/train/weights/best.pt conf=0.25 source=/content/johnwickgunshopscene.mp4 save=True

!yolo task=detect mode=predict model={HOME}/runs/detect/train/weights/best.pt conf=0.25 source=/content/johnwickgunshopscene.mp4 show=True



"""**NOTE:** Let's take a look at few results."""

import glob
from IPython.display import Image, display

for image_path in glob.glob(f'{HOME}/runs/detect/predict2/*.jpg')[25:36]:
    display(Image(filename=image_path, width=600))
    print("\n")
```

```
#load model
model = project.version(dataset.version).model

#choose random test set image
import os, random
test_set_loc = dataset.location + "/test/images/"
random_test_image = random.choice(os.listdir(test_set_loc))
print("running inference on " + random_test_image)

pred = model.predict(test_set_loc + random_test_image, confidence=40,
    overlap=30).json()
pred
```

Appendix B

Screen shots

B.1 Output Images

The below output screenshots show the working of our project.

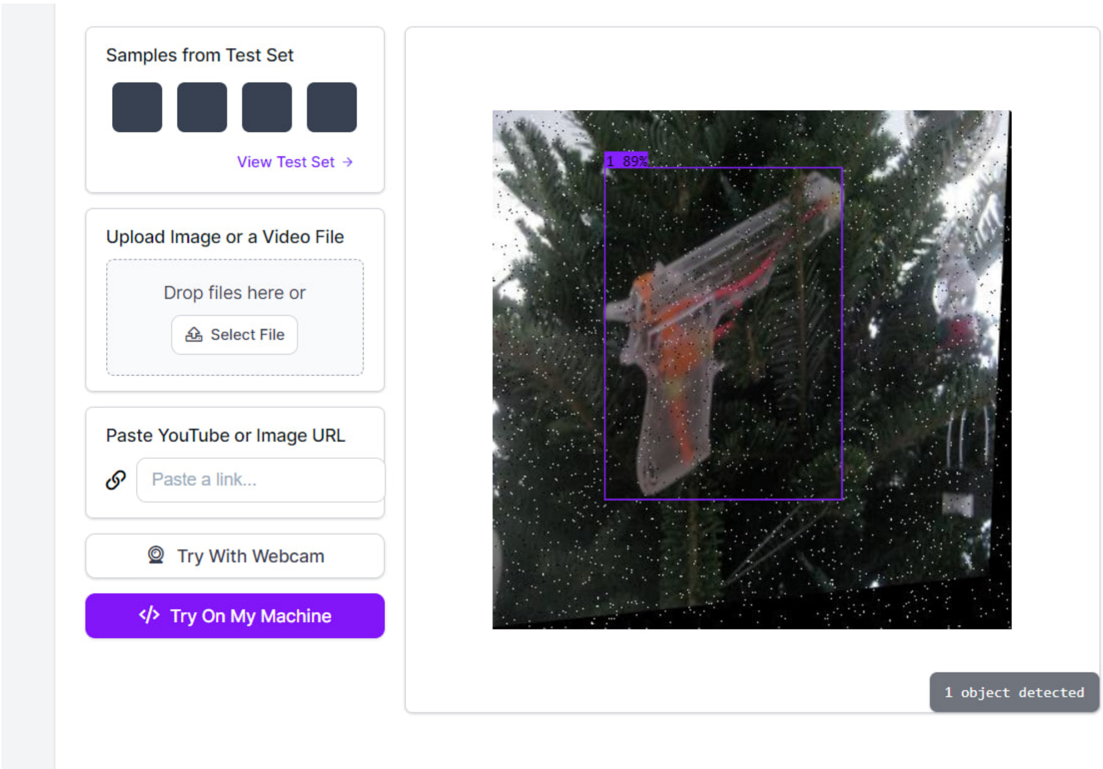

Figure B.1: User Interface
Drag or paste image-URL to detect any firearms

Figure B.2: Result Images

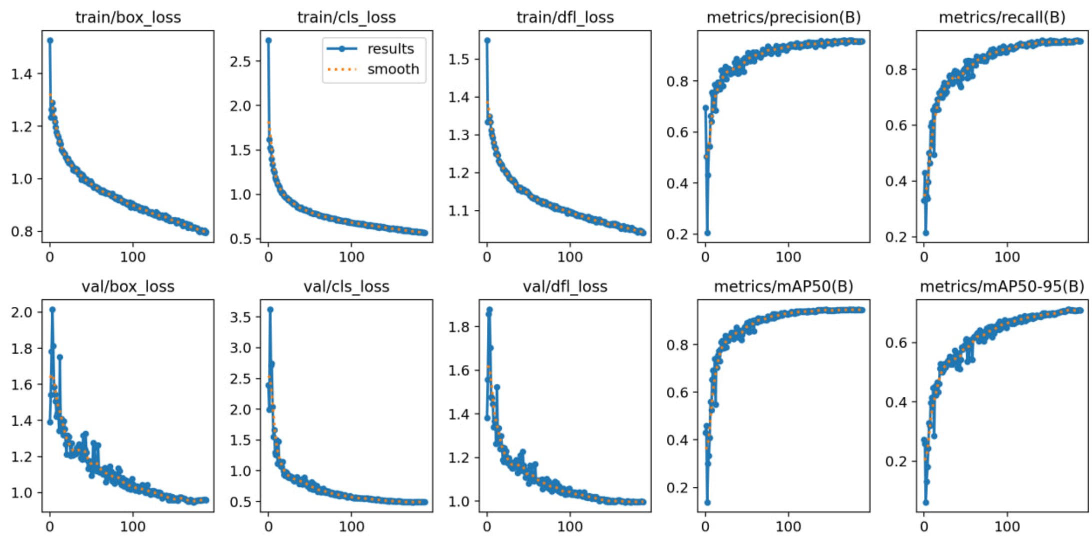

Figure B.3: Training Graphs

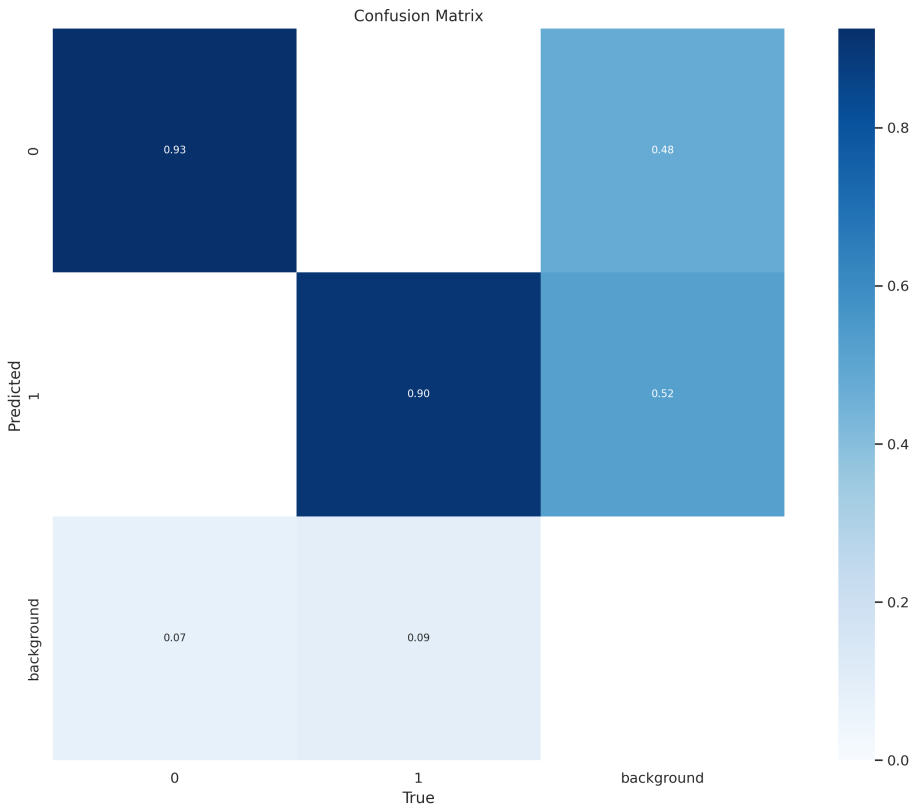

Figure B.4: Confusion Matrix

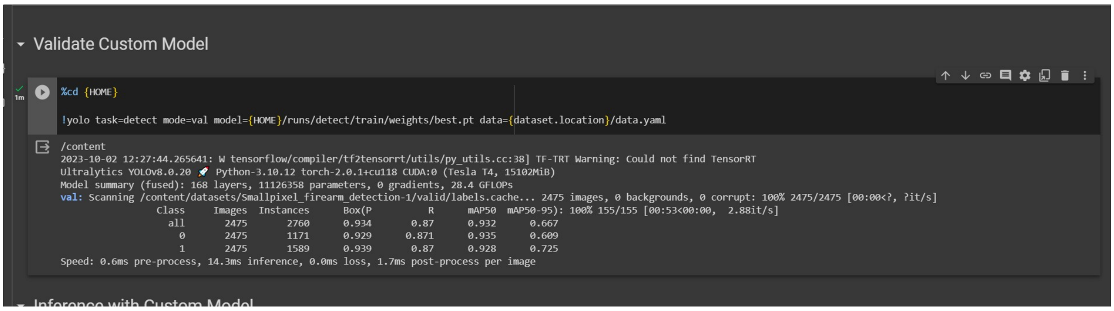

Figure B.5: mAP(Mean Average Precision Values) for Validation Set

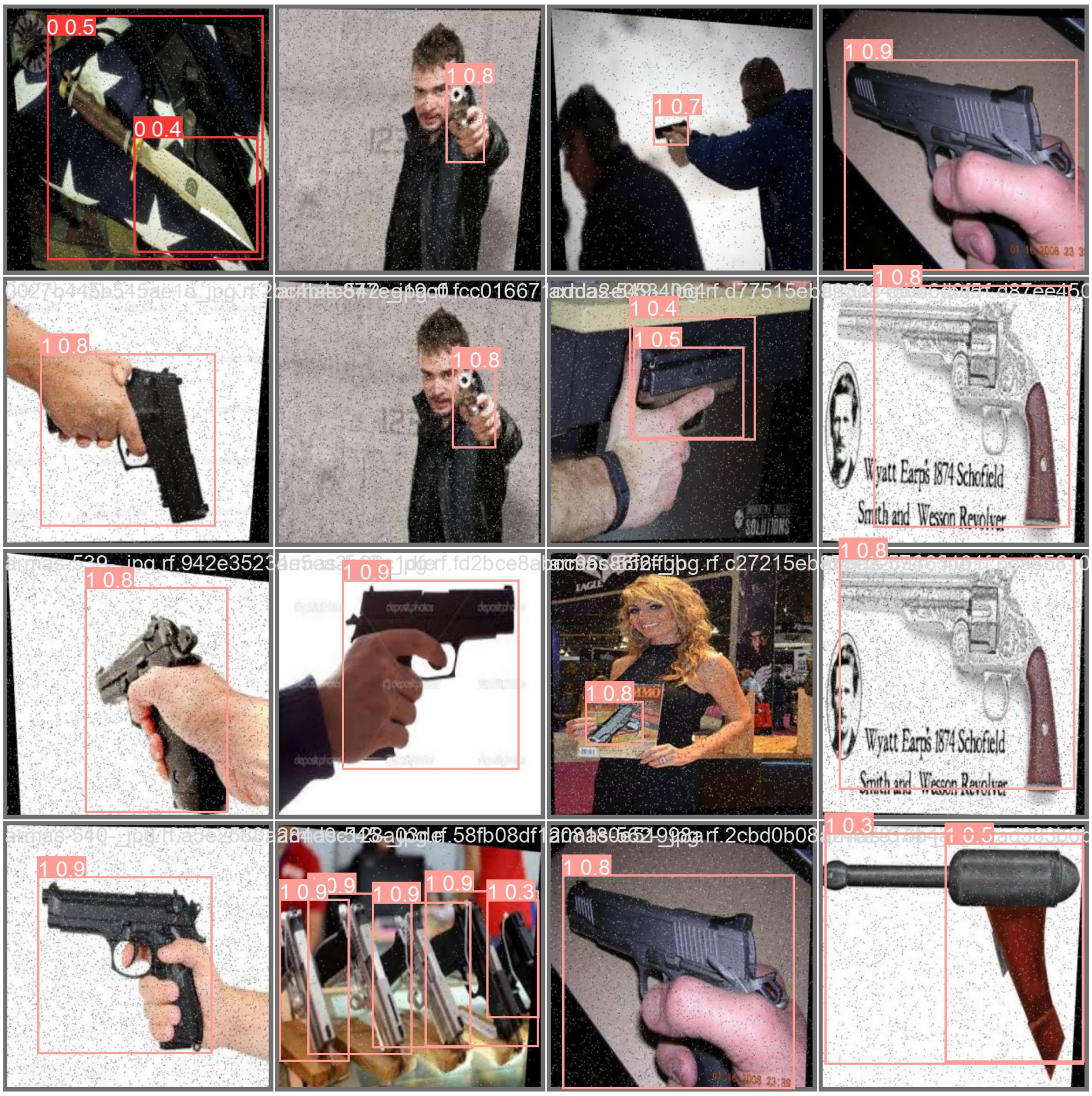

Figure B.6: Validation Set Results

Appendix C

Data sets used in the project

In the development of this firearm detection project, a diverse range of datasets was utilized to train and validate the model, enhancing its proficiency in recognizing handguns within various real-world scenarios. The project drew upon multiple sources, including datasets sourced from Kaggle, closed-circuit television (CCTV) footage, and publicly available videos from platforms like YouTube.

To ensure accurate and efficient training of the model, the dataset underwent rigorous labelling, with the assistance of the Roboflow platform. Approximately 16,000 images were meticulously annotated to identify and delineate firearms within the visual data.

Of the total dataset, 12,870 images were allocated for training, providing the model with a robust foundation for learning and pattern recognition. An additional 2,475 images were reserved for validation, allowing for continuous assessment and fine-tuning of the model's performance. The remaining 990 images formed the testing dataset, serving as the ultimate benchmark for evaluating the system's accuracy and reliability.

This comprehensive approach to dataset collection, annotation, and allocation, combined with the utilization of Roboflow for labelling, played a pivotal role in equipping the model with the capability to accurately and efficiently detect handguns across a wide array of practical scenarios, thus contributing to the project's success in enhancing security and threat detection.