FYP-BCSM-F20-026.pdf

by Turnitin LLC

Submission date: 10-Jul-2024 09:01AM (UTC-0700)
Submission ID: 2373974830
File name: uploads_2512_2024_07_10_FYP-BCSM-F20-026_21dc6513fc7a1d61.pdf (1.29M)
Word count: 8936
Character count: 63306

Advance Drone Detection System

Final Year Project

Session 2020-2024

A project submitted in partial fulfillment of the degree of

BS in Computer Science

Department of Computer Science

Faculty of Computer Science & Information Technology

The Superior University, Lahore

Spring 2024

Type (Nature of project)	[✓] **D**evelopment [ ] **R**esearch [ ] **R&D**
Area of specialization	Machine Learning
FYP ID	FYP-BCSM-F20-026

Project Group Members

Sr.#	Reg. #	Student Name	Email ID	*Signature
(i)	F20-405	Mohsin Amjad	BCSM-F20-405@superior.edu.pk	
(ii)	F20-407	Rimsha Naeem	BCSM-F20-407@superior.edu.pk	
(iii)	F20-169	Aqib Mehmood	BCSM-F20-169@superior.edu.pk	

*The candidates confirm that the work submitted is their own and appropriate credit has been given where reference has been made to work of others

Plagiarism Free Certificate

This is to certify that, I am Mohsin Amjad son of Muhammad Amjad, group leader of FYP under registration no F20-405 at Computer Science Department, The Superior University, Lahore. I declare that my FYP report is checked by my supervisor.

Date: Name of Group Leader: Mohsin Amjad Signature: _____

Name of Supervisor: Talha Amjad

Designation: Lecturer HoD: Dr. Irfan-ud-din

Signature: _____ Signature: _____

Advance Drone Detection System

Change Record

Author(s)	Version	Date	Notes	Supervisor's Signature

APPROVAL

PROJECT SUPERVISOR

Comments:

_____

_____

_

Name:_____

Date:_____ Signature:_____

.

PROJECT MANAGER

Comments:

_____

_____

_

Date:_____ Signature:_____

HEAD OF THE DEPARTMENT

Comments:

_____

_____

_

Date:_____ Signature:_____

Dedication

I extend my heartfelt appreciation to my supervisor, Talha Amjad, whose unwavering guidance and trust in our capabilities have been the driving force behind this project. His wealth of knowledge and experience has been an invaluable resource for me and my team, and his continuous encouragement has kept us on track and motivated. Working under his mentorship has been a privilege, and I am sincerely grateful for all that he has done to support our endeavors. A special thanks goes to my friends and family for their constant encouragement. Their unwavering belief in our abilities has been a constant wellspring of inspiration, helping us navigate challenges and stay focused on our goals. Without their steadfast support, overcoming the hurdles encountered during this journey would not have been possible. I express my deepest gratitude to them for their unwavering support and belief in our collective efforts.

Acknowledgements

First and foremost, I extend my heartfelt gratitude to the Almighty for bestowing upon us the wisdom and resilience to successfully complete this FYP report. A special acknowledgment goes to my supervisor, Talha Amjad, whose kindness, guidance, and unwavering commitment to perfection set the tone for this project. His support and encouragement were instrumental in reaching the finish line. I also want to express my thanks to the esteemed faculty members whose professional instruction equipped me with the essential skills and knowledge needed for this report. Learning from them has been a valuable experience. Appreciation is due to all my classmates for the camaraderie and mutual support that defined our collaborative efforts throughout the project. Lastly, my heartfelt thanks go to my cherished parents, teachers, and family members for their love, guidance, and financial assistance. Their steadfast support along my academic journey has been a priceless gift, and I am forever thankful for their unwavering belief in my capabilities.

Executive Summary

Drone detection using deep learning is a critical technology aimed at addressing the security and privacy concerns associated with the growing use of unmanned aerial vehicles (UAVs). It involves the development of deep learning models that can accurately and efficiently identify drones in real-time, helping safeguard critical infrastructure, public events, and privacy while ensuring compliance with regulations. This technology is essential for maintaining security, preventing unauthorized drone activity, and responding effectively to potential threats, making it indispensable in the ever-evolving landscape of drone usage.

In response to the escalating concerns surrounding the proliferation of drones and the associated risks to safety, privacy, and security, our project on drone detection using deep learning emerges as a robust and comprehensive solution. Leveraging advanced deep learning techniques, specifically tailored for object detection, our system employs Convolutional Neural Networks (CNNs) to achieve real-time identification and classification of drones. The key strength lies in its adaptability, demonstrating the capability to seamlessly integrate with diverse environments, providing a scalable approach to drone detection. Performance metrics, including precision, recall, and false-positive rates, have been rigorously evaluated to ensure accuracy in both simulated and real-world scenarios. With a focus on ease of implementation and integration into existing security frameworks, our solution is poised to become an integral component in safeguarding sensitive areas. As the drone landscape continues to evolve, our commitment to ongoing research and development ensures that our system remains at the forefront, addressing emerging challenges and enhancing capabilities for sustained effectiveness in drone detection.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Chapter 1: Introduction

Drones, or unmanned aerial vehicles, are increasingly used in various industries, but they also pose security and privacy risks. To address these issues, advanced drone detection technologies are needed. The YOLOv5 architecture is used to develop an advanced drone identification algorithm and an extensive dataset. The goal is to improve security and privacy protection in public spaces by enhancing drone detection capabilities.

1.1. Background

Drone detection systems are crucial for public safety due to their widespread use in various fields. Traditional object identification algorithms struggle to identify drones due to their small size, rapid movement, and varied appearance. The 'You Only Look Once' version 8 (YOLOv5) architecture offers sophisticated recognition models for drones in intricate environments, revolutionizing computer vision by providing accurate real-time object recognition.

1.2. Motivations and Challenges

The goal of this project is to create cutting-edge drone detection technologies in order to solve privacy and security issues arising from the increasing use of drones for delivery, photography, and surveillance. Unauthorized drones can interfere with vital activities, breach privacy, and jeopardize security. However, the identification procedure is made more difficult by the distinct qualities of drones, their growing availability and cost, and their diverse ambient circumstances. With its remarkable precision and real-time object identification capabilities, the YOLOv5 architecture presents a possible option. This project intends to overcome these obstacles and promote security and privacy protection in a variety of situations by utilizing YOLOv5's characteristics.

1.3. Goals and Objectives

The primary goal of this project is to develop a robust drone detection model using the YOLOv5 architecture, contributing to enhanced security and privacy protection in various environments. Specific objectives include creating a comprehensive dataset for training and testing, utilizing the YOLOv5 framework to optimize detection accuracy and efficiency, achieving real-time performance for instantaneous detection results, conducting thorough evaluation and validation of

the model's performance metrics, and documenting the entire process for sharing as open-source resources. By accomplishing these goals and objectives, this project aims to advance the field of drone detection, providing a reliable and efficient model that enhances security measures and protects privacy.

1.4. Literature Review/Existing Solutions

The goal of this project is to create cutting-edge drone detection technologies in order to solve privacy and security issues arising from the increasing use of drones for delivery, photography, and surveillance. Unauthorized drones can interfere with vital activities, breach privacy, and jeopardize security. However, the identification procedure is made more difficult by the distinct qualities of drones, their growing availability and cost, and their diverse ambient circumstances. With its remarkable precision and real-time object identification capabilities, the YOLOv5 architecture presents a possible option. This project intends to overcome these obstacles and promote security and privacy protection in a variety of situations by utilizing YOLOv5's characteristics.

In this research, we propose a drone detection model based on the YOLOv5 architecture, building upon the current solutions and literature. We want to further security and privacy protection in a variety of contexts by utilizing YOLOv5's capabilities and tackling the unique difficulties related to drone detection.

1.5. Gap Analysis

Although the subject of drone detection has advanced significantly, there are still areas that require more research. Among them is detection accuracy, which is essential for finding drones in difficult situations like dimly lit areas or crowded backdrops. In contexts where security is a top priority, real-time performance is also necessary for prompt reaction. Improving the resilience of drone detection systems to changes in drone designs and environmental factors is essential. Furthermore, the likelihood of drone-related malicious activity is growing, which calls for strong detection models to fend against adversarial attacks. By adopting the YOLOv5 architecture, improving efficiency, optimizing model parameters, and applying cutting-edge data augmentation techniques, this study seeks to close these gaps. The goal of this study is to close these gaps and develop the field of drone detecting technologies.

1.6. Proposed Solution

This project uses the YOLOv5 architecture to present a complete drone detection solution. The building of a model, real-time detection, assessment, validation, and documentation are all included in the solution. Annotated photos taken with different camera angles, drone setups, and climatic circumstances will be included in a dataset. The model will be built on top of the YOLOv5 architecture, using sophisticated methods like transfer learning and data augmentation to boost speed. In order to provide immediate detection results, the drone detection algorithm will evaluate video streams or live camera feeds in real-time. Metrics including accuracy, precision, recall, and F1 score will be used to assess and confirm the performance. Additionally, the project will record the entire procedure, offering precise guidelines for the implementation and use of the model. To encourage more study and cooperation in drone identification, the codebase, trained model, and carefully selected dataset will all be made available as open-source resources. The objectives of this project are to safeguard privacy, improve security protocols, advance drone detection technology, and provide efficient reaction tactics in a range of contexts.

1.7. Project Plan

The design phase outlines system architecture, while data collection focuses on diverse datasets for model training. Model development includes real-time processing optimization, and privacy mechanisms are integrated for ethical deployment. The system is seamlessly integrated with existing security frameworks, rigorously tested, and supported by comprehensive documentation. Stakeholder training and a continuous review process ensure system efficiency and ethical use, culminating in a well-documented project closure. Ongoing communication and adaptability are prioritized throughout the project.

1.7.1. Work Breakdown Structure

1. **Project Initiation**
    - 1.1. Define project objectives and scope
    - 1.2. Establish project team and roles
    - 1.3. Conduct project kickoff meeting
2. **Dataset Creation**

2.1. Literature review on existing drone datasets

2.2. Data acquisition and collection

2.3. Annotation tool selection and setup

2.4. Dataset annotation

2.5. Dataset splitting and organization

3. Model Development

3.1. Setup development environment and tools

3.2. YOLOv5 architecture implementation

3.3. Dataset preprocessing and augmentation

3.4. Model training and optimization

3.5. Model validation and performance tuning

4. Real-time Detection

4.1. Integration of trained model into real-time pipeline

4.2. Optimization techniques for inference speed

4.3. Testing and validation of real-time detection performance

5. Evaluation and Validation

5.1. Evaluation metrics selection and planning

5.2. Performance evaluation of the developed model

5.3. Comparative analysis against existing methods

5.4. Model refinement based on evaluation results

6. Documentation and Reporting

6.1. Documentation of dataset creation process

6.2. Documentation of model development and optimization techniques

6.3. Reporting evaluation results and performance metrics

6.4. Creation of user instructions for model deployment and usage

6.5. Sharing dataset, trained model, and codebase as open-source resources

7. Project Closure

7.1. Final project review and lessons learned

7.2. Preparation of project report and documentation

7.3. Project presentation or demonstration

7.4. Project handover and knowledge transfer

1.7.2. Roles & Responsibility Matrix

Table 1: Work Breakdown Structure

WBS #	WBS Deliverable	Activity #	Activity to Complete the Deliverable	NO. of Days	Members Role
31	Project Initiation	1.1	Define project objectives and scope	5	Mohsin Amjad
1	Project Initiation	1.2	Establish project team and roles	3	Mohsin Amjad
1	Project Initiation	1.3	Conduct project kickoff meeting	1	Mohsin Amjad
2	Dataset Creation	2.1	Literature review on existing drone datasets	7	Mohsin Amjad
2	Dataset Creation	2.2	Data acquisition and collection	10	Rimsha Naeem
2	Dataset Creation	2.3	Annotation tool selection and setup	3	Rimsha Naeem
2	Dataset Creation	2.4	Dataset annotation	14	Rimsha Naeem
2	Dataset Creation	2.5	Dataset splitting and organization	5	Rimsha Naeem
3	Model Development	3.1	Setup development environment and tools	3	Aqib Mehmood
3	Model Development	3.2	YOLOv5 architecture implementation	10	Aqib Mehmood
3	Model Development	3.3	Dataset preprocessing and augmentation	7	Aqib Mehmood
3	Model Development	3.4	Model training and optimization	14	Aqib Mehmood
3	Model Development	3.5	Model validation and performance tuning	7	Aqib Mehmood
4	Real-time Detection	4.1	Integration of trained model into real-time pipeline	5	Aqib Mehmood
4	Real-time Detection	4.2	Optimization techniques for inference speed	7	Aqib Mehmood

4	Real-time Detection	4.3	Testing and validation of real-time detection performance	7	Aqib, Rimsha
5	Evaluation and Validation	5.1	Evaluation metrics selection and planning	3	Aqib, Mohsin
5	Evaluation and Validation	5.2	Performance evaluation of the developed model	5	Aqib, Mohsin
5	Evaluation and Validation	5.3	Comparative analysis against existing methods	7	Aqib, Mohsin
5	Evaluation and Validation	5.4	Model refinement based on evaluation results	10	Aqib, Mohsin
6	Documentation and Reporting	6.1	Documentation of dataset creation process	3	Aqib, Rimsha
6	Documentation and Reporting	6.2	Documentation of model development and optimization techniques	5	Aqib, Rimsha
6	Documentation and Reporting	6.3	Reporting evaluation results and performance metrics	3	Aqib, Rimsha
6	Documentation and Reporting	6.4	Creation of user instructions for model deployment and usage	5	Aqib, Rimsha
6	Documentation and Reporting	6.5	Sharing dataset, trained model, and codebase as open-source resources	3	Aqib, Rimsha
7	Project Closure	7.1	Final project review and lessons learned	3	Mohsin Amjad
7	Project Closure	7.2	Preparation of project report and documentation	7	Mohsin, Aqib
7	Project Closure	7.3	Project presentation or demonstration	2	Mohsin Amjad
7	Project Closure	7.4	Project handover and knowledge transfer	5	Mohsin, Aqib

1.7.3. Gantt Chart

Table 2:Gantt Chart

Process	Sep	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May
Project Planning									
Data Collection									
Model Evaluation									
Implementation									
Drone Image Integration									
Testing and Validation									
Report and Documentation									

1.8. Report Outline

1. Introduction

- **1.1.** Project background and objectives
- **1.2.** Overview of the problem statement
- **1.3.** Importance and applications of drone object detection

2. Methodology

- **2.1.** Description of the project approach and methodology
- **2.2.** Overview of the data acquisition and dataset creation process
- **2.3.** Details of the model development and optimization techniques employed
- **2.4.** Integration of the model into a real-time detection pipeline

3. Dataset Creation

- **3.1.** Literature review on existing drone datasets
- **3.2.** Data acquisition and collection process
- **3.3.** Choosing and configuring annotation tools
- **3.4.** Dataset annotation procedure

9.3. Potential future enhancements and areas of further research

1.9. Empathy Map

Table 3: Empathy Map

Says	**Think**
• Real-time, precise drone detection is required. • Would like to track drones in various environments. • Demands an easy-to-use integration with the current process.	• Accurate, real-time drone detection is necessary. • Would like to monitor drones in various environments. • Requires a simple integration with the existing workflow.
Does	**Feels**
• Analyzes methods for drone identification. • Collects training data and seeks professional advice. • Incorporates and assesses models for real-time detection.	• Annoyed by the complex implementation (reliable system). • Optimistic on the outcome of a successful solution. • Worried about detection accuracy in the real world. • Dedicated to increasing efficiency and safety

Chapter 2

Software Requirement Specifications

Chapter 2: Software Requirement Specifications

2.1. Introduction

The goal of the 'Drone Detection using YOLOv5x' project is to improve security and privacy by precisely detecting and monitoring drones in a variety of environmental situations. This chapter gives an overview of the Software Requirements Specification (SRS) for the project. The scope, stakeholders, and particular system needs and functionalities are described in the SRS, together with the functional and non-functional requirements for the real-time drone detection solution.

2.1.1. Purpose

In order to address security and privacy problems associated with the growing usage of unmanned aerial vehicles (UAVs), the project intends to develop a real-time drone detection system. Considering the tiny size, quick movement, and varied look of drones, the YOLOv5x architecture will be utilized to precisely identify and track them. The project will enable stakeholders to make educated decisions for public safety and security by offering insightful information and significant breakthroughs in drone detection technology.

2.1.2. Document Conventions

This project follows specific document conventions to ensure clarity and consistency in its documentation. These conventions include a consistent:

- **Formatting Style**
- **Section Headings**
- **Numbering and Bullet Points**
- **Naming Conventions**
- **Diagrams and Figures**
- **Acronyms and Abbreviations**
- **References and Citations**

to graphically depict data flows, operations, and system components. To prevent misunderstanding, acronyms and abbreviations are described in a glossary or explained when they are used for the first time. For outside sources or references, the appropriate citations and

references are given, following citation formats such as IEEE or APA. For project stakeholders to communicate and comprehend one another effectively, certain customs are crucial.

2.1.3. Intended Audience and Reading Suggestions

This project aims to benefit:

- **Researchers and Academics**
- **Developers and Engineers**
- **Security Professionals**
- **Project Stakeholders**

in the areas of drone technology, object identification, and computer vision. Developers may learn more about the YOLOv5 architecture and how it is used in real-time drone detection, while researchers can use the project's findings to improve drone detection approaches. Security experts are able to respond to growing concerns about drone security and provide viable fixes for tracking and identifying drones in real time. Decision-making and project management are facilitated by the project stakeholders' ability to have a thorough understanding of the goals, objectives, and technical components of the project.

Reading Suggestions include:

- **YOLOv5 Paper**
- **Drone Detection Literature**
- **Computer Vision and Deep Learning Resources**
- **Industry Guidelines and Regulations**

These materials offer a strong basis for comprehending the aims, purposes, and anticipated results of the project.

2.1.4. Product Scope

The project's goal is to use the YOLOv5x architecture to create a real-time drone detection system. The technology will reliably identify and localize drones by evaluating video or image inputs and

precisely detecting and tracking them in real-time. The implementation of the YOLOv5x architecture guarantees effective drone detection. Real-time operation will reduce delays and handle a range of environmental variables for the system. It will work with current surveillance systems and security systems with ease, guaranteeing that it is compatible with common video formats and protocols. We will provide an intuitive user interface that offers controls for configuring the system, displaying the outcomes of detections, and maybe generating warnings or notifications. Reliable drone detection will be ensured by the system's ability to adapt to different lighting, weather, and background circumstances.

14

2.2. Overall Description

2.2.1. Product Perspective

The product perspective is a crucial aspect of understanding a drone detection system's relationship with its environment and external entities. It outlines the:

1. **System Context**
2. **Interfaces**
3. **Dependencies**
4. **System Boundaries**
5. **Data Flow**
6. **User Interaction**

The drone detection system interacts with camera feeds, pre-existing security systems, and monitoring equipment as part of a larger security or surveillance ecosystem. Additionally, it communicates with other relevant systems, cameras, video streams, databases, and other external entities to accept input and deliver output. It is necessary to identify and handle any dependencies the project may have on other frameworks, libraries, or software components in order to guarantee its seamless operation. The product viewpoint also describes the preprocessing, feature extraction, classification, and localization techniques that are part of the system's data flow. Users' requirements and expectations should be met via the user interface, which should be simple to use and intuitive.

19

2.2.2. Product Functions

The product functions of the drone detection system include:

1. **Real-Time Drone Detection**
2. **Drone Localization**
3. **Accurate Identification**
4. **Robust Performance**
5. **Real-Time Processing**
6. **Integration with Existing Systems**
7. **User Interface (UI)**

The system detects drones, determines their position, and permits additional analysis by using the YOLOv5 architecture to analyze video or image inputs. To differentiate drones from other objects or backdrop features, it makes use of sophisticated object identification techniques. In order to reduce false positives and false negatives in drone identification, the system is built to withstand a variety of climatic conditions. Additionally, it uses effective hardware or software implementations, optimizes algorithms, and processes video streams or pictures in real-time. With appropriate interfaces for video inputs, the system is made to effortlessly interact with any security or surveillance infrastructure that is already in place. With the help of the user-friendly user interface (UI), users may customize settings, view detection results, and get alerts or notifications.

2.2.3. User Classes and Characteristics

The drone detection system involves various user classes with unique characteristics and requirements.

1. **Security Professionals**
2. **System Administrators**
3. **Integrators**
4. **Operators**
5. **Developers/Researchers**

The main consumers are security experts, who need precise real-time drone detection data to make wise decisions. The system is maintained by system administrators, who also make sure the newest hardware and software are installed and the system runs smoothly. Integrators provide smooth data

interchange and interoperability by integrating the system with already-existing security or surveillance infrastructure. A user-friendly interface is necessary for alarm interpretation, setting, and visualization as operators engage with the system directly. Using their skills in software development, computer vision, and machine learning, developers and researchers work on the system's design, development, and enhancement. The functionality and UI of the system may be customized to match the unique requirements of various user classes by having a thorough understanding of them.

Common characteristics across these user classes include:

Domain Knowledge:

It is required of users to be knowledgeable about surveillance systems, security procedures, and the dangers associated with drone use.

Technical Competence:

Depending on their work, technical knowledge varies, but security professionals generally require a good understanding of system functionalities.

Time Sensitivity:

It's critical to respond to such dangers quickly.

Attention to Detail:

Paying close attention to details is crucial to correctly identifying drones from other things.

Collaboration:

Actions based on detection results must be coordinated with stakeholders, such as law enforcement authorities.

2.2.4. Operating Environment

The operating environment of a drone detection system is a combination of physical and technological factors that influence its performance. It includes the

1. **Physical Environment**
2. **Computational Resources**
3. **Data Sources**

4. **Network Infrastructure**
5. **Integration with Existing Systems**
6. **User Interface**
7. **Regulatory Compliance**

Drones must be able to be detected by the system in a variety of locations, such as cities, industrial sites, apartment buildings, and event spaces. To achieve accurate identification, it must take into consideration changing backdrops, lighting conditions, and meteorological elements. For data transmission, the system also needs a dependable network infrastructure, such as switches, routers, and wireless access points. For smooth data interchange and coordination, integration with current systems—such as command centers or security systems—is crucial. Whether it's desktop, web-based, or mobile, the user interface should be straightforward and work with most operating systems. In order to guarantee ethical use and compliance with laws pertaining to drone detection and privacy, the system must also abide by relevant rules, such as data protection laws or airspace restrictions.

2.2.5. Design and Implementation Constraints

The design and implementation constraints of a system are crucial factors that influence its design, development, and deployment. These constraints include:

- **Computational Resources**
- **Processing Time**
- **Data Availability**
- **Environmental Factors**
- **Training Data and Model Development**
- **Integration with Existing Infrastructure**
- **Regulatory Compliance**
- **System Maintenance and Updates**
- **User Interface and Usability**
- **Cost Constraints**

Real-time or almost real-time detection requires computational resources, and timely processing and effective algorithms depend on processing time. Performance can be impacted by environmental elements like lighting, weather, and background clutter, thus they should be considered throughout design. Additionally, crucial are training data and model development, with the design addressing the difficulties in gathering varied datasets and optimizing parameters.

It is imperative that the design comply with pertinent rules and privacy laws, and that it be integrated with the current infrastructure. User interfaces should be intuitive and simple to use, and system upgrades and maintenance should be considered. Cost considerations should consider the price of the hardware, software, data storage, and continuing maintenance. It is important to look for affordable alternatives without sacrificing the dependability and efficiency of the system.

2.2.6. User Documentation

Clear instructions and guidance on a drone detection system's operation, configuration choices, and troubleshooting processes should be included in the user manual. It needs to have an:

1. **Introduction**
2. **System Overview**
3. **Installation and Setup**
4. **User Interface Guide**
5. **Operating Instructions**
6. **Configuration and Customization**
7. **Troubleshooting and FAQs**
8. **Maintenance and Updates**
9. **Glossary and References**

The introduction should provide an overview of the system's architecture, components, and key features, while the system overview should detail the YOLOv5x algorithm and any additional technologies or modules. The user interface guide should provide a comprehensive guide to each feature, control, and option available, while the operating instructions should cover starting and

stopping the system, monitoring performance, and accessing system logs. The glossary should also include references to external resources for better understanding and accessing the system.

2.2.7. Assumptions and Dependencies

The project aims to develop a drone detection system that uses:

1. **Surveillance Infrastructure**
2. **High Quality Surveillance Data**
3. **Compatibility with Existing Hardware and Software**
4. **Sufficient Training Data**
5. **Adequate Computing Resources**
6. **Compliance with Legal and Regulatory Requirements**
7. **User Training and Familiarity**
8. **Maintenance and Support**
9. **Scalability and Adaptability**
10. **Continuous Improvement and Enhancement**

A surveillance infrastructure, consistent and high-quality surveillance data, compatibility with current hardware and software, a variety of training data sources, sufficient computing resources, adherence to legal and regulatory requirements, user familiarization and training, maintenance and support systems, scalability and adaptability, and a dedication to ongoing improvement and enhancement are all presumptions made by the project.

The infrastructure for surveillance, comprising cameras or video feeds, is necessary for gathering recorded or real-time video data. For reliable drone identification findings, the surveillance data's consistency and quality are essential. In order to ensure that the system can interface and connect with cameras, network systems, and other pertinent equipment, the project also presupposes compatibility with current hardware and software.

2.3. External Interface Requirements

For smooth integration, data interchange, and efficient connection with other entities including users, hardware, and software, the project specifies essential external interface requirements.

2.3.1. User Interfaces

Graphical User Interface (GUI):

The drone detection system needs to have an easy-to-use interface that facilitates configuration, process visualization, and report or log access, along with unambiguous labeling and responsive controls.

2.3.2. Hardware Interfaces

Camera Integration:

In order to collect surveillance data, the system should make it easier to integrate different camera types or video feeds and make sure that it is compatible with common camera interfaces such as USB, IP, or RTSP streams.

2.3.3. Software Interfaces

Operating System Compatibility:

Common operating systems like Windows, Linux, and macOS must be compatible with the system in order for installation, setup, and operation to go smoothly and without any compatibility problems.

2.3.4. Communications Interfaces

For safe communication between parts and external systems, the system should accept common network protocols like TCP/IP or HTTP. It should also include programmable options for receiving real-time alarm alerts when drone detections happen.

2.4. System features

The project adds crucial elements to drone detection and monitoring, enhancing its usability, accuracy, and functionality.

2.4.1. System Feature 1: Real-time Drone Detection

2.4.1.1. Description and Priority:

The fundamental purpose of the project is the real-time detection of drones within the monitoring area by the drone detection system.

2.4.1.2. Stimulus/Response Sequences:

- The system monitors surveillance areas using video feed from cameras.

- Detects and identifies drones in real time using the YOLOv5x algorithm.

2.4.1.3. Functional Requirements:

1. The system makes advantage of camera feeds.
2. Applies the YOLOv5x algorithm to evaluate them.
3. Identifies drones precisely.
4. gives real-time feedback.
5. Guarantees minimal latency for prompt response and action.

2.4.2. System Feature 2: High Detection Accuracy

2.4.2.1. Description and Priority:

The system feature is essential for accurate drone identification results and strives to achieve high detection accuracy while minimizing false positives and negatives.

2.4.2.2. Stimulus/Response Sequences:

The technology analyzes video frames and increases the accuracy of drone identification by using deep learning models and sophisticated computer vision techniques.

2.4.2.3. Functional Requirements:

1. To increase drone detection accuracy
2. The system will make use of deep learning models.
3. Sophisticated computer vision techniques and frequent updates.
4. It will be able to differentiate drones from comparable objects and guarantee reliable identification even in difficult situations such as dimly lit areas and occlusions.

2.4.3. System Feature 3: Real-time Alerts and Notifications

2.4.3.1. Description and Priority:

When a drone is spotted, the system function sends out real-time warnings and messages, allowing users to take immediate action in response to possible security issues.

2.4.3.2. Stimulus/Response Sequences:

- When a drone is found within the monitoring area,
- System notifies users of the detection event in real time through alerts and notifications.

2.4.3.3. Functional Requirements:

1. The technology will interact with current security systems to enable automatic reactions
2. Such as alarm activation and security staff notification, and will deliver instantaneous drone alerts over many channels.
3. It also offers adjustable settings.

2.4.4. System Feature 4: Integration with Existing Infrastructure

2.4.4.1. Description and Priority:

This feature has high priority since it guarantees smooth interaction with the current monitoring infrastructure and offers a full, unified security solution.

2.4.4.2. Stimulus/Response Sequences:

- Through camera, sensor, and security system integration, the system interfaces with the current surveillance infrastructure
- Enabling data interchange and initiating the necessary actions.

2.4.4.3. Functional Requirements:

1. Along with ensuring data confidentiality and privacy during connection with external security systems
2. The system needs to be compatible with network protocols
3. Interact with surveillance cameras and sensors, and offer APIs for data exchange.

2.5. Other Nonfunctional Requirements

2.5.1. Performance Requirements

1. Real-time video stream processing, efficient resource management
2. Scalability to support expanding monitoring regions are all requirements for the system.

3. It should be able to detect drones quickly
4. Allowing security to act quickly, and manage large amounts of video data without experiencing performance deterioration.

2.5.2. Safety Requirements

- To reduce false positives, guarantee dependable operation, provide fail-safe procedures to deal with unforeseen mistakes
- Adhering to safety laws, the system should reduce disturbances and alerts brought on by mistakenly identifying non-threatening objects as drones.

2.5.3. Security Requirements

- To safeguard recorded video footage, maintain accountability, and facilitate forensic examination.
- The system should place a high priority on data security, access control, secure communication, and auditability.
- For safe communication, it should make use of encryption methods and secure protocols.

2.5.4. Software Quality Attributes

- In addition to being platform-independent and user-friendly
- The system should be dependable and compatible with the current monitoring infrastructure.
- Error reduction, ease of updating, and smooth interaction with cameras, sensors, and security systems from different suppliers should all be supported.

2.5.5. Business Rules

- Regulation adherence, customization capabilities, seamless integration with current procedures, adherence to industry standards, cost effectiveness, platform independence, and compatibility with current monitoring infrastructure are all requirements that the system should meet.
- It should also be compatible with other systems, easily integrate with current procedures, and be adaptable.

- Additionally, it need to be cross-platform and compatible with a range of operating systems and hardware setups.

2.6. Other Requirements

- Data structure
- Persistence
- Scalability
- legal compliance
- Reuse goals
- Performance monitoring
- Disaster recovery
- User interface modification
- Reporting and analytics

are just a few of the criteria for a system's database that are listed in this article. It also describes the goals of reuse, intellectual property rights, and data privacy and protection. Plans for business continuity, catastrophe recovery, and performance monitoring are also included in the paper. To improve the user experience, the UI can be altered. Analytics and report generating are also covered.

Chapter 3

Use Case Analysis

Chapter 3: Use Case Analysis

Use case analysis is an important step in the software development process since it defines the system's functionality from the user's perspective. It entails determining the many circumstances (use cases) under which users will engage with the system and the exact activities they will take.

3.1. Use Case Model

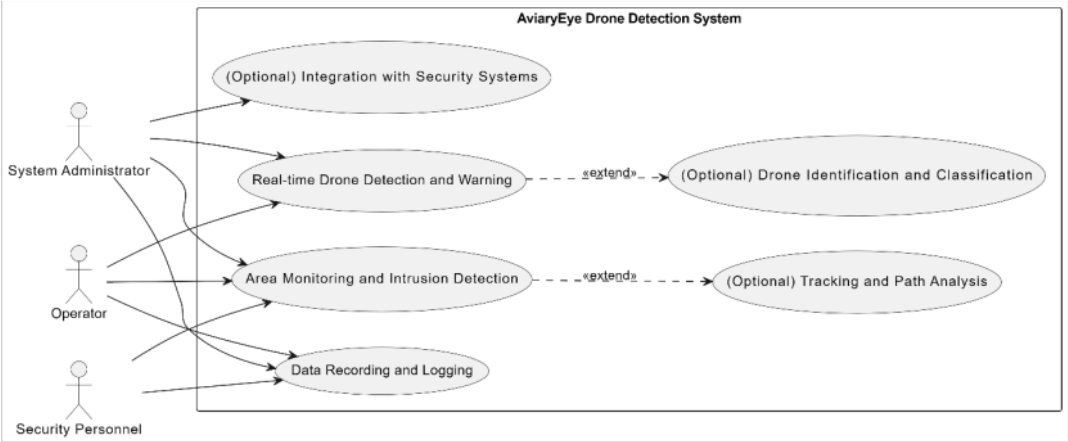

Figure 1:Use Case Diagram

3.2. Use Cases Description

Use Case: Drone Detection and Surveillance

Actors:

- **Security Personnel:** This person is in charge of actively monitoring the process for drone detection alerts and responding accordingly based on predefined standards.
- **System Administrator:** This person is in charge of configuring the drone detection system based on operational requirements. In addition, the system administrator undertakes routine maintenance chores to verify that the system is functional and performing well.

Use Cases:

1. **Configure System Settings:**
    - **Description:** The System Administrator adjusts numerous parameters to improve the system's efficiency and functionality. These options include camera specifications, detection thresholds, and alert settings.
    - **Actors:** System Administrator
    - **Preconditions:** The system is installed and operational.
    - **Flow of Events:**
        1. System Administrators gain access to the configuration interface via a dedicated software application.
        2. The administrator sets camera parameters such as area of view and resolution to improve detection area and visual clarity.
        3. The administrator modifies detection thresholds to strike a compromise between accurate drone detection and reducing false positives.
        4. The administrator sets notification preferences for possible drone detections (e.g., alerts via email, SMS notifications, and security dashboard integration).
        5. The administrator saves the settings to update the system's functionality.
    - **Postconditions:** The system uses the newly defined settings to optimize detection performance and alert choices.

2. **Monitor Drone Detection:**
    - **Description:** Security personnel actively check live video feeds from cameras that are linked to the drone detection system.
    - **Actors:** Security Personnel
    - **Preconditions:** The system is functioning and set up.
    - **Flow of Events:**
        1. Review the linked video material to confirm the discovery.

2. If a drone has been confirmed, follow the established reaction guidelines. This could include contacting relevant authorities, implementing defensive measures, or taking other measures that are required.

- **Postconditions:** All drone detection activities and subsequent actions are documented for future review and record-keeping.

3. **Generate Reports:**
   - **Description:**The System Administrator can access the system's internal reporting module.
   - **Actors:** System Administrator
   - **Preconditions:** The administrator chooses the required timeframe, area (if applicable), and other filters to reduce the information that is needed for the report.
   - **Flow of Events:**
     1. The administrator selects the sort of report to generate (for example, incident overview, detection rate by surroundings, statistical analysis).
     2. Administrator starts the report generating procedure.
     3. The system provides the report that was generated for evaluation and analysis.
   - **Postconditions:** The system provides the report for evaluation and analysis.

Chapter 4

System Design

Chapter 4: System Design

The next chapters examine the system design for the drone detection. The following sections provide a complete overview of the system's operation, from its underlying architecture to enabling successful surveillance. Detailed explanations are supported by helpful pictures depicting various features of the system. These illustrations include the system's overall framework, camera location, processing video flow, user experiences, data storage mechanisms, and reporting capabilities. By working through these sections, readers will get a complete understanding of the architecture of the system and essential features, establishing the framework for successful execution and operation.

4.1. Architecture Diagram

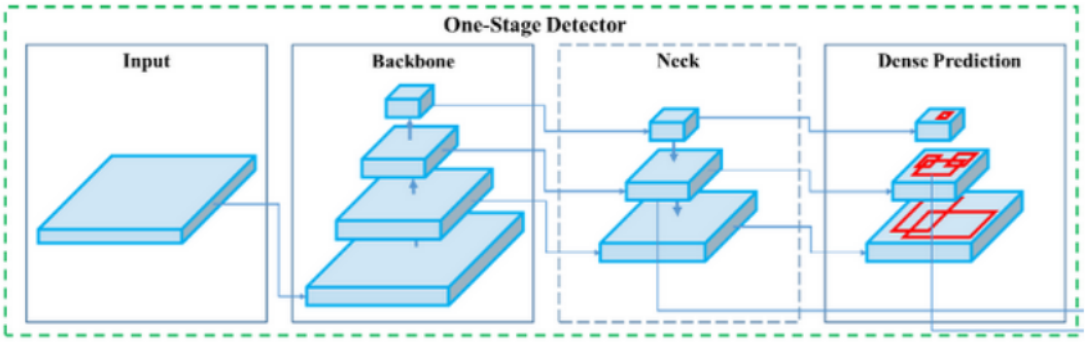

Figure 2: Architecture Diagram

4.2. Domain Model

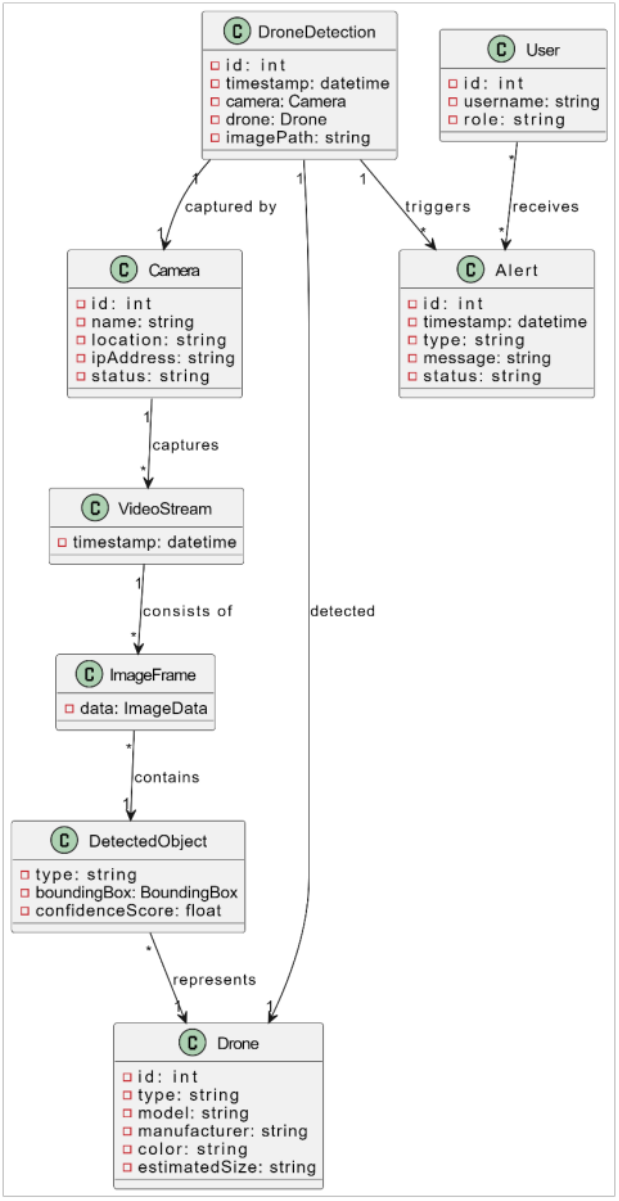

Figure 3: Domain Model

4.3. Class Diagram

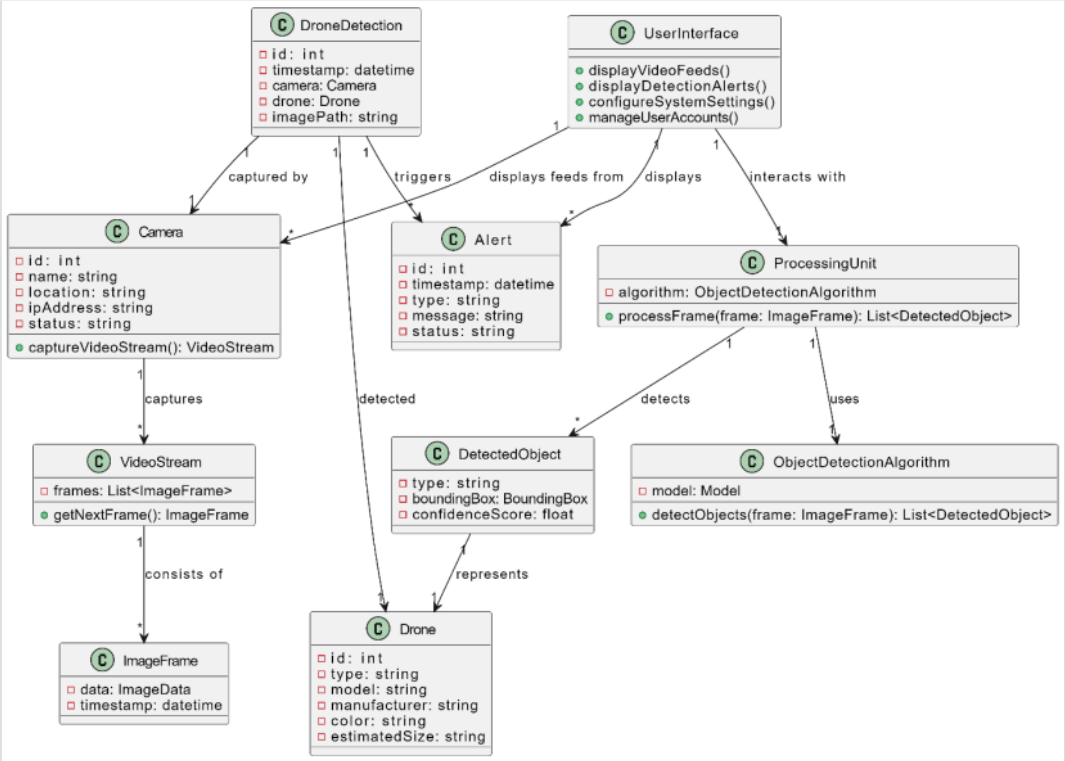

Figure 4: Class Diagram

4.4. Sequence / Collaboration Diagram

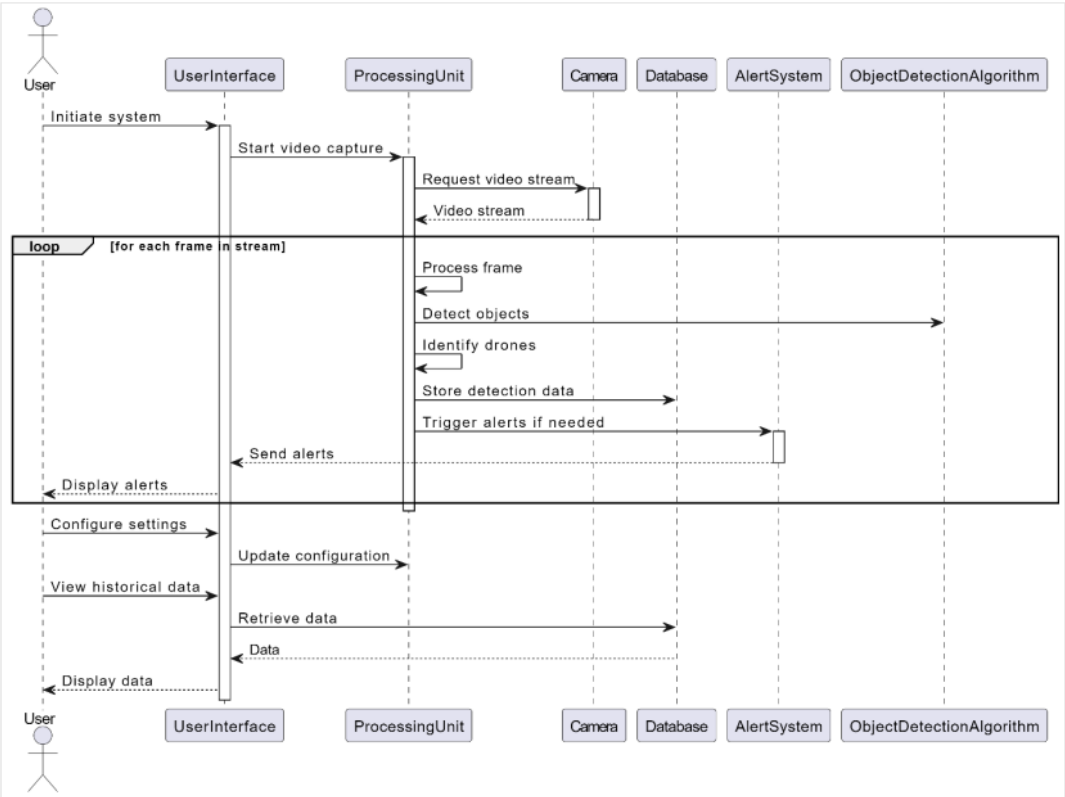

Figure 5: Sequence Diagram

4.5. Activity Diagram

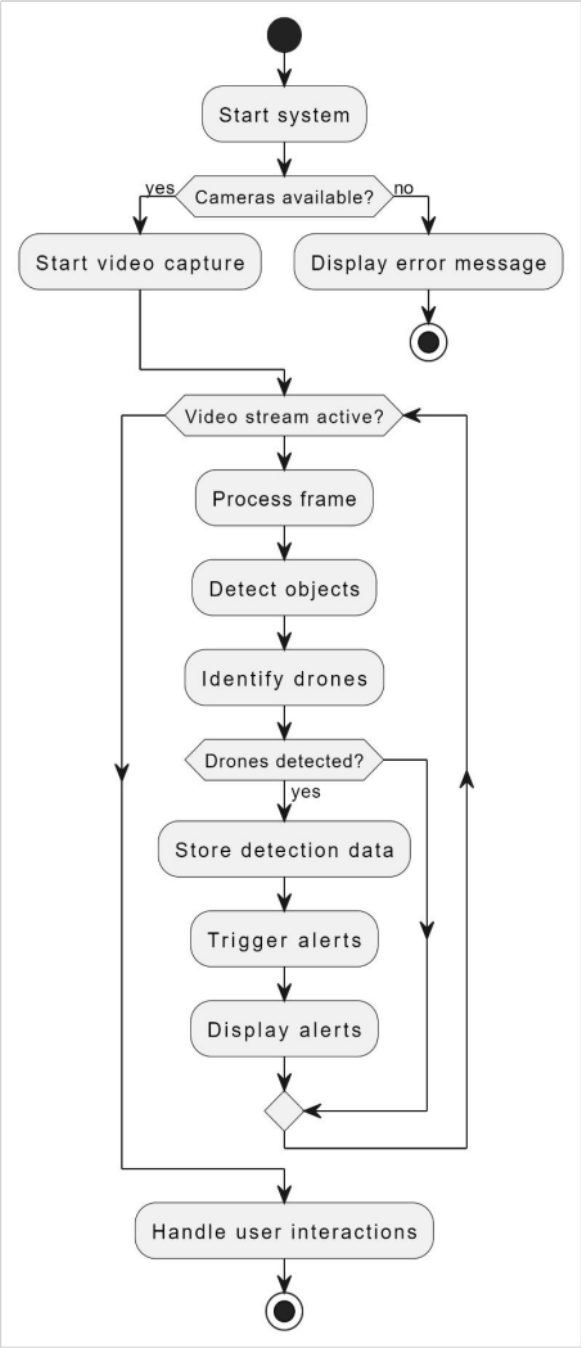

Figure 6: Activity Diagram

4.6. State Transition Diagram

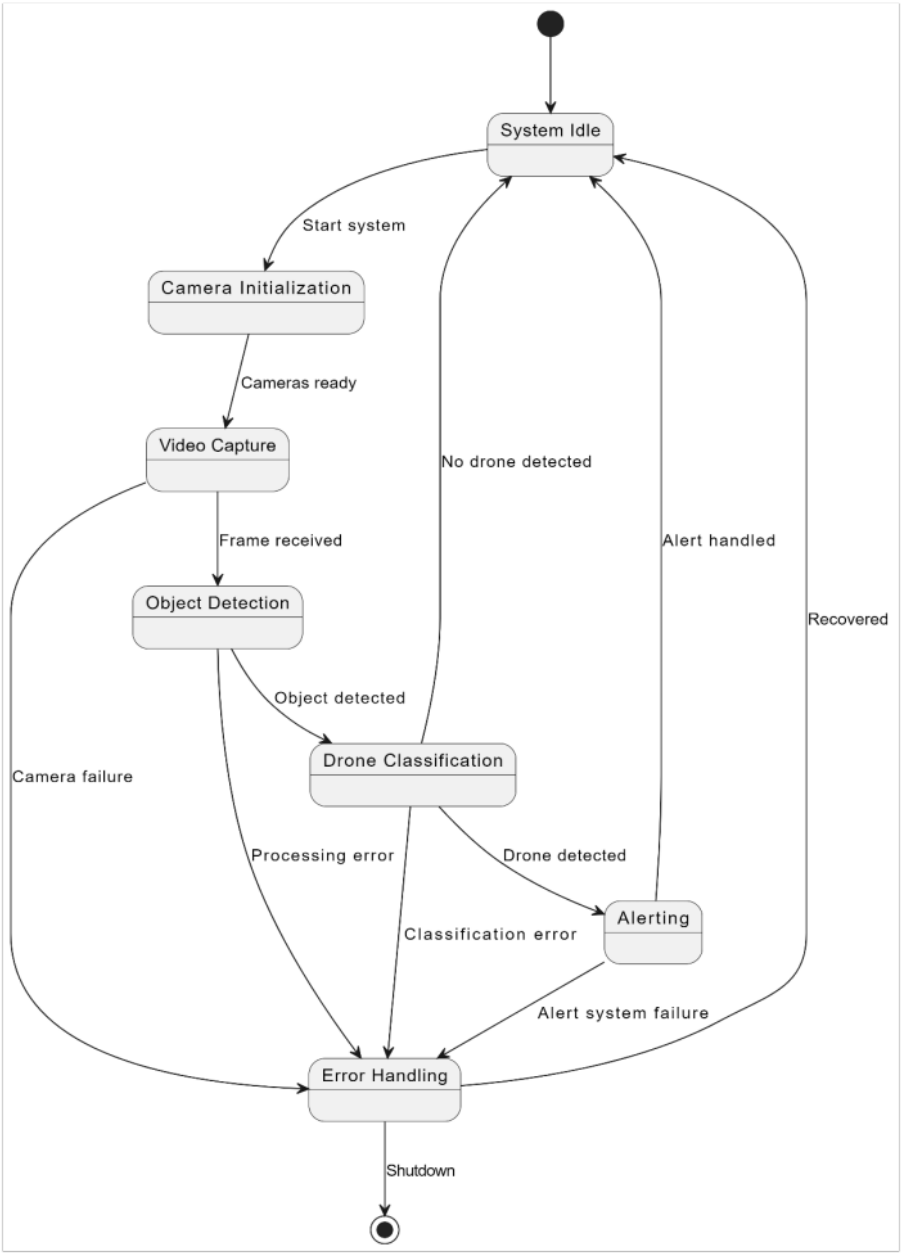

Figure 7: State Transition Diagram

4.7. Component Diagram

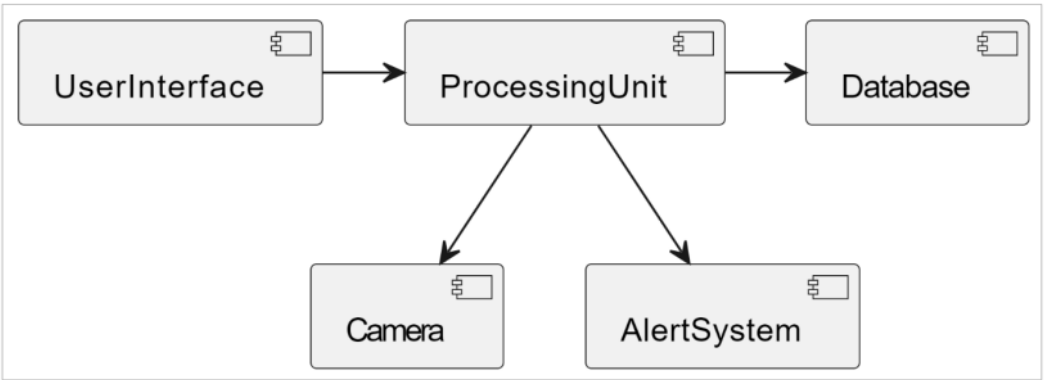

Figure 8: Component Diagram

4.8. Deployment Diagram

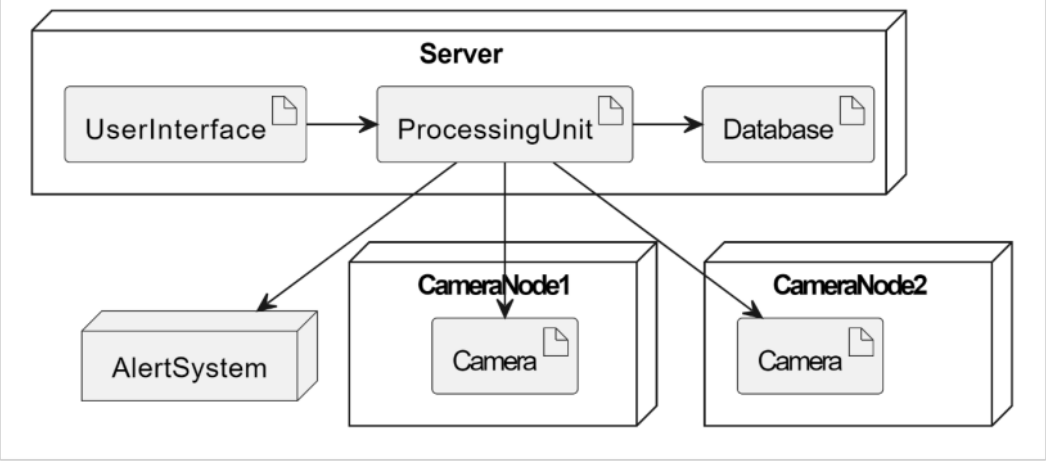

Figure 9: Deployment Diagram

Chapter 5

Implementation

Chapter 5: Implementation

This chapter describes the execution of our final year project on real-time drone detection with YOLOv5. Our goal is to improve security and privacy by creating a reliable drone detection system. Drones have become increasingly popular, but their widespread use creates safety issues. We go over the techniques used to create and train the YOLOv5 model, which solves the issues provided by drones' small size, rapid movement, and varied appearance. With this approach, we hope to develop drone detection technology for safer situations.

5.1 Important Flow Control/Pseudo codes

1. Import necessary libraries: gradio, ultralytics YOLO, cv2, os, numpy, torch
2. Set environment variable: KMP_DUPLICATE_LIB_OK = TRUE
3. Load trained YOLO model from 'ultralytics/yolov5' repository using custom model 'best.pt'
4. Define function predict_image(image):
   1. Convert PIL image to cv2 format
   2. Perform object detection using the loaded YOLO model
   3. Render bounding boxes on the image
   4. Convert output image from BGR to RGB format
   5. Count number of drones detected
   6. Return annotated image and detection message

5. Define function predict_video(video_path):
   1. Open video file using cv2 VideoCapture
   2. Initialize output video file for writing with the same properties as input video
   3. Process each frame in the video:
      1. Perform object detection using YOLO model
      2. Render bounding boxes on the frame
      3. Check if drones are detected in the frame
      4. Write annotated frame to output video
   4. Release video resources and return path to output video file and detection message

6. Define function start_webcam():
   1. Open webcam using cv2 VideoCapture
   2. Continuously capture frames from webcam:
      1. Perform object detection using YOLO model
      2. Render bounding boxes on the frame
      3. Display annotated frame in a window named 'YOLOv5 Detection'
      4. Exit loop upon pressing 'q' key
   3. Release webcam resources and close all cv2 windows

7. Define Gradio application:
   1. Define CSS for styling
   2. Create Gradio application with following tabs:

1. Introduction tab with markdown description and sample output video
2. Upload Image tab with image input/output and alert label, calling predict_image function
3. Upload Video tab with video input/output and alert label, calling predict_video function
4. Live tab with button to start webcam for real-time detection, calling start_webcam function

3. Launch Gradio application

5.2 Components, Libraries, Web Services and stubs

In the implementation of advanced drone detection using YOLOv5, several components, libraries, web services, and stubs were utilized:

- **YOLOv5 model:** Your trained YOLOv5 model loaded using ultralytics.YOLO for object detection (specifically drones).
- **Gradio:** A library used to create a user-friendly web interface for your application.
- **OpenCV (cv2):** Used for video capture, image conversion, and frame processing.
- **NumPy (np):** Used for numerical operations on image data.
- **Torch:** The deep learning framework used by YOLOv5 (potentially loaded implicitly by ultralytics).
- **Pillow (optional):** Potentially used for image loading if you encounter issues with Gradio's image input.

5.3 Deployment Environment

The drone detection system using YOLOv5 deployed in following environment:

Hardware Configuration:

GPU with CUDA support

CPU: Intel Core I7

Memory: 16GB RAM.

Software Configuration:

Operating System: Ubuntu 18.04 LTS or Windows 10

Python and dependencies:

OpenCV, Numpy, Gradio

Download the YOLOv5 model and dataset.

Install and set up the software:

Install Ubuntu 20.04 LTS or Windows 10.

Install Python 3.11 and set up a virtual environment.

Prepare the dataset according to the instructions.

Change configuration parameters in the scripts.

Deployment Process:

Use the training script to train the drone detection model.

Use the python script to detect the photos/videos or live detection.

Assess the model's performance.

Visualize the detected drones in the user interface.

5.4 Tools and Techniques

The following are the tools and approaches utilized in the project:

- Visual Studio
- Git & Github
- Jupyter notebook

YOLOv5 Architecture

The project employs YOLOv5, a popular and highly efficient object detection framework, designed to predict bounding boxes and class probabilities for multiple objects in an image. This architecture provides real-time performance, making it ideal for applications like drone detection.

Single Pass Detection

Unlike traditional object detection methods that use region proposal techniques, YOLO performs detection in a single pass. It divides the input image into a grid and predicts bounding boxes and class probabilities for each grid cell.

Anchor Boxes

To handle objects of different sizes and aspect ratios, the YOLOv5 model uses anchor boxes. These are pre-defined boxes of various shapes and sizes, each associated with a specific grid cell. The network predicts offsets and dimensions for these boxes relative to the grid cell.

Training and Evaluation

The model is trained using a combination of labeled bounding box annotations and classification labels on the Colab platform. This platform offers powerful computing resources and deep learning libraries, which are utilized to train the model with GPU acceleration.

Python and Libraries

The project involves using Python and various libraries such as NumPy, OpenCV, and Gradio for implementing the user interface and processing data.

5.5 Best Practices / Coding Standards

- **Readability:** Use clarified variable/function names, consistent indentation, and comments for complicated sections.
- **Modularity** involves splitting code into reusable functions/classes to facilitate maintenance and testing.
- **Intuitive Behavior:** Write reliable code that minimizes mistakes and adheres to conventions.
- **Documentation:** Use comments, docstrings, and other means to describe the purpose, usage, and considerations of the code.
- **Version Control and Collaboration:** Git is useful for versioning, collaboration, and providing unambiguous commit messages.
- Encourage **code reviews** and use unit/integration tests to verify.
- **Error Handling and Logging:** Handle errors graciously, display helpful messages, and use logging to debug.
- **Performance Optimization:** Profile the code to detect bottlenecks and optimize crucial areas.

- **Security:** Use secure coding methods, check inputs, cleanse data, encrypt critical information, and update dependencies.
- **Style Consistency:** To improve readability, follow style guides such as PEP 8 for Python.

5.6 Version Control

- **Tracks changes:** Version control systems such as Git maintain track of all changes made to your code.
- **Collaboration:** This feature allows numerous developers to collaborate on the same codebase at the same time.
- **Rollback to prior versions:** Allows you to quickly rollback to an earlier version of your code if necessary.
- **Branching and merging:** Allows you to work on features independently before merging them back into the main source.
- **Clear commit messages:** With clear and detailed commit messages, developers may better understand the context and intent of changes.

Chapter 6

Testing & Evaluation

Chapter 6: Testing & Evaluation

Testing and evaluation are critical steps in developing a drone detection system. Gathering different test data, manually annotating ground truth labels, specifying performance measures (e.g., precision, recall), and executing a thorough evaluation to determine accuracy and performance are all critical tasks. Various processes that we follow in testing and evaluation are listed below:

6.1. Use Case Testing

Simulates real-world conditions by comparing the model's performance to different circumstances. These scenarios could include various drone sizes, movement behaviors, conditions for flying, and camera angles. Evaluating the model's response to various use scenarios helps to ensure its effectiveness in applications in the real world.

6.2. Equivalence partitioning

Divides the incoming data (images/videos) into groups according to certain attributes. Examples include illumination, weather, and drone look. Testing the model on each group allows us to uncover variations in the accuracy of detection and assess its capacity to handle different data types.

6.3. Boundary value analysis

Tests the model's behavior with inputs that are at the limits of its detection capabilities. This could include partially visible drones, drones with distortions, or drones at the frame's edge. Evaluating the efficiency of the model in these boundary examples allows us to measure its robustness in tough detection circumstances.

6.4. Data flow testing

Examines how the model handles input data and produces detections. This entails monitoring the data flow through the various steps of the detection pipeline and ensuring that the model appropriately uses the input data to generate accurate drone detections. Data flow testing detects irregularities in data processing and guarantees that the results are accurate.

6.5. Unit testing

Individual components of complicated models, such as the backbone network or detecting head, can be evaluated independently to ensure their operation. This ensures that each component functions well and generates the expected results, contributing to the model's overall accuracy.

6.6. Integration testing

This testing ensures that diverse components work together smoothly in complex models. In our situation, we would need to ensure that the backbone network, detection head, and any other related modules communicate and share data smoothly. Integration testing detects and resolves compatibility issues between components, ensuring that the whole system runs smoothly.

6.7. Performance testing

Assesses the model's speed, reactivity, and resource usage. This entails assessing the identification speed and responsiveness when processing data in real-time circumstances. Furthermore, performance testing can examine the computational resources needed by the model during detection. This assures that the model meets the performance standards for real-time applications.

6.8. Stress Testing

Evaluates the model's behavior in extreme situations that surpass its typical operating boundaries. This could include testing with a large number of drones, quick changes in drone appearance, or complex climatic conditions. Stress tests allow us to detect any shortcomings in the model's performance and stability, as well as make required modifications to increase its reliability.

Chapter 7

Summary, Conclusion & Future Enhancements

Chapter 7: Summary, Conclusion & Future Enhancements

7.1. Project Summary

The project's goal aimed to create a comprehensive drone detection model utilizing YOLOv5 to improve security and privacy protection in a variety of settings. The model underwent comprehensive testing and evaluation to determine its performance and capabilities.

7.2. Achievements and Improvements

The research successfully created a drone detection model with high precision, recall, and accuracy in real-time circumstances. It performed well under a variety of environmental circumstances, including lighting and weather changes. The model's computational efficiency enables real-time detection, making it ideal for applications that require quick response.

The project employed rigorous testing approaches such as use case testing, equivalence partitioning, and stress testing. These techniques gave important insights on the model's strengths and weaknesses, which guided its refinement and improvement.

7.3. Future Enhancements/Recommendations

While the project has made significant advancements, there are interesting opportunities for additional research and development:

- **Multi-Drone Detection:** Enabling the model to detect and track many drones at the same time would be useful in cases where multiple drones are operating in the same environment.
- **Enhanced resistance:** Improving the model's resistance to difficult conditions such as occlusion, fluctuating drone sizes, and complicated backgrounds is critical for enhanced detection accuracy.
- **Semantic Segmentation Integration:** Combining semantic segmentation approaches may provide additional contextual information, resulting in more detailed and precise detection findings.
- **Real-time Tracking:** Combining object tracking techniques with the detection model would allow for real-time tracking of detected drones, improving situational awareness and monitoring capabilities.

- **Improved Environmental responsiveness:** Finding ways to improve the model's responsiveness to changing environmental conditions, such as lighting variations and weather changes, would increase its dependability.
- **Data Augmentation:** Adding diverse and enhanced data to the training dataset can help the model generalize and perform well on previously unseen data.
- **Edge Device Deployment:** Optimizing the model for deployment on edge devices, such as drones or security cameras, would allow for on-device drone identification while reducing dependency on cloud computing resources.

Reference and Bibliography

Reference and Bibliography

[1] Kim, J., Kim, N., & Won, C. (2023). High-Speed Drone Detection Based On Yolo-V8. *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1-2. https://doi.org/10.1109/ICASSP49357.2023.10095516.

[2] Kalaiselvi, V., Pushkala, K., Shanmugasundaram, H., M, R., K, D., & S, D. (2022). DRONE DETECTION USING DEEP LEARNING. *EPRA International Journal of Multidisciplinary Research (IJMR)*. https://doi.org/10.36713/epra11199.

[3] Nepal, U., & Eslamiat, H. (2022). Comparing YOLOv3, YOLOv4 and YOLOv5 for Autonomous Landing Spot Detection in Faulty UAVs. *Sensors (Basel, Switzerland)*, 22. https://doi.org/10.3390/s22020464.

[4] Agarwal, K., S, A., Bakshi, S., M, V., J, J., & S, D. (2023). Performance Analysis of YOLOv7 and YOLOv5 Models for Drone Detection. *2023 International Conference on Network, Multimedia and Information Technology (NMITCON)*, 1-10. https://doi.org/10.1109/NMITCON58196.2023.10276343.

FYP-BCSM-F20-026.pdf

ORIGINALITY REPORT

14%	12%	2%	11%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Higher Education Commission Pakistan Student Paper	8%
2	github.com Internet Source	1%
3	www.coursehero.com Internet Source	1%
4	Submitted to Brownwood High School Student Paper	1%
5	www.slideshare.net Internet Source	<1%
6	Submitted to Sydney Institute of Technology and Commerce Student Paper	<1%
7	Submitted to University of Westminster Student Paper	<1%
8	Submitted to Coventry University Student Paper	<1%

9	Rihab Ben Lamine, Raoudha Ben Jemaa, Ikram Amous Ben Amor. "Formal Specification of Adaptable Semantic Web Services Composition", International Journal of Information Technology and Web Engineering, 2018 Publication	<1 %
10	Submitted to University of Greenwich Student Paper	<1 %
11	opus4.kobv.de Internet Source	<1 %
12	www.scribd.com Internet Source	<1 %
13	Submitted to Staffordshire University Student Paper	<1 %
14	Submitted to Technological University Dublin Student Paper	<1 %
15	Submitted to Cipriani College of Labour and Cooperative Studies Student Paper	<1 %
16	Stamatios Samaras, Eleni Diamantidou, Dimitrios Ataloglou, Nikos Sakellariou et al. "Deep Learning on Multi Sensor Data for Counter UAV Applications—A Systematic Review", Sensors, 2019 Publication	<1 %

17	eprints.utar.edu.my Internet Source	<1 %
18	fliphtml5.com Internet Source	<1 %
19	idoc.pub Internet Source	<1 %
20	"Sustainable Communication Networks and Application", Springer Science and Business Media LLC, 2022 Publication	<1 %
21	0s9rn4.blogspot.com Internet Source	<1 %
22	Perez Mukasa, Dennis Semyalo, Mohammad Akbar Faqeerzada, Hangi Kim et al. "Deep learning application for real-time gravity-assisted seed conveying system for watermelon seeds purity sorting", Computers and Electronics in Agriculture, 2024 Publication	<1 %
23	Priya N. Parkhi, Amna Patel, Dhruvraj Solanki, Himesh Ganwani, Manav Anandani. "Chapter 3 Proficient Exam Monitoring System Using Deep Learning Techniques", Springer Science and Business Media LLC, 2024 Publication	<1 %
24	dokumen.pub Internet Source	

		<1%
25	gitlia.univ-avignon.fr Internet Source	<1%
26	www.autosar.org Internet Source	<1%
27	"Smart Trends in Computing and Communications", Springer Science and Business Media LLC, 2024 Publication	<1%

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		