Democratic and Popular Algerian Republic
Ministry of Higher Education and Scientific Research



Yahia Fares University of Medea
Faculty of Sciences
Department of Mathematics and Computer Science

Final Master Project

Submitted for Obtaining The Master's Degree in Computer Science

Speciality: Applied Artificial Intelligence

An Integrated AI-Powered Air Defense System for Threat Detection and Neutralization

---

Prepared & Presented by

Aid Younes
Boucherit Faycal

Supervisor

Dr. Bensaissa Azzedine

2025-2026

Younes Acknowledgments,

I am deeply proud to have reached this level of knowledge after many years of hard work, perseverance, and dedication. This achievement represents not only my efforts, but also the invaluable support of those who stood beside me throughout this journey.

First and foremost, I would like to express my deepest gratitude to ***my mother****, whose unconditional love, patience, and encouragement never wavered for a single moment. Her constant presence and belief in me have been my greatest source of strength and motivation.*

I would also like to sincerely thank ***my younger sister****, who has always been by my side, supporting me and caring for me. Her presence in my life is truly a blessing for which I am deeply grateful.*

I would like to express my sincere gratitude to ***Dr.Bensaissa Azzedine*** *for his guidance, valuable insights, and continuous support throughout this work.*

I am equally thankful to everyone who has contributed to my journey in one way or another. Every kind word, every gesture of support, and every piece of advice has played a meaningful role in helping me reach this point.

True success is never achieved alone; it is built upon the love, support, and sacrifices of those who believe in us.

Faycal Acknowledgments

This work would not have been possible without the support and encouragement of many wonderful people, to whom I owe my deepest gratitude.

First and foremost, I wish to express my heartfelt thanks to my thesis supervisor ***Dr. Bensaissa Azzedine****, whose guidance, patience, and unwavering belief in my abilities shaped this work in more ways than I can express. Your insightful feedback, thoughtful advice, and constant encouragement pushed me to think deeper and aim higher. It has been a true privilege to work under your mentorship.*

Above all, my deepest and most heartfelt appreciation goes to my beloved family. To my ***mother****, whose tender love, warm encouragement, and endless sacrifices have been my greatest comfort — thank you for always believing in me and for being my safe haven through every challenge. To my* ***father****, whose strength, wisdom, and quiet support have guided me more than words can say — thank you for teaching me the value of hard work and perseverance. To my* ***brothers****, thank you for the laughter, the loyalty, and for standing by my side at every step — your presence made this journey lighter and more joyful.*

This work is as much yours as it is mine.

Abstract

The growing complexity of low-altitude airspace and the rapid proliferation of unmanned aerial vehicles have created an urgent demand for automated systems capable of identifying aerial threats with speed and precision. This project presents the design and development of an integrated AI-powered aerial object detection and classification system, trained on two datasets spanning seven distinct classes: drones, helicopters, airplanes, birds, military airplanes, missiles, and UAVs. Three deep learning architectures were evaluated for image classification — YOLOv26x, EfficientNetV2-L, and Swin Transformer — achieving accuracies of 99.40%, 98.72%, and 99.19% respectively on the custom DAOD dataset. For spatial object detection, YOLOv26x combined with Slicing Aided Hyper Inference (SAHI) achieved an mAP@0.5 of 0.981 on the AOD4 dataset and 0.931 on the more complex DAOD dataset, demonstrating strong generalization across varying target sizes and backgrounds. The system addresses key challenges including class imbalance, inter-class visual similarity between drones and birds, and small object detection at long range. Results confirm that a well-regularized, medium-scale YOLO architecture provides the optimal balance between detection accuracy and computational efficiency for real-time air defense applications.

Keywords: Aerial Object Detection, Aerial Object Classification, Convolutional Neural Network, YOLO, Drone Detection, Deep Learning, SAHI, Air Defense, UAV

Résumé

La prolifération rapide des véhicules aériens sans pilote et la complexité croissante de l'espace aérien à basse altitude ont créé un besoin urgent de systèmes automatisés capables d'identifier les menaces aériennes avec rapidité et précision. Ce projet présente la conception et le développement d'un système intégré de détection et de classification d'objets aériens basé sur l'intelligence artificielle, entraîné sur deux ensembles de données couvrant sept classes distinctes : drones, hélicoptères, avions, oiseaux, avions militaires, missiles et véhicules aériens sans pilote. Trois architectures d'apprentissage profond ont été évaluées pour la classification d'images — YOLOv26x, EfficientNetV2-L et Swin Transformer — atteignant des précisions de 99,40%, 98,72% et 99,19% respectivement sur l'ensemble de données personnalisé DAOD. Pour la détection spatiale d'objets, YOLOv26x combiné à l'inférence hyperspectrale par découpage SAHI a atteint un mAP@0,5 de 0,981 sur l'ensemble AOD4 et de 0,931 sur l'ensemble DAOD plus complexe, démontrant une forte capacité de généralisation à travers des tailles de cibles et des arrière-plans variés. Le système répond à des défis majeurs notamment le déséquilibre des classes, la similarité visuelle entre les drones et les oiseaux, ainsi que la détection de petits objets à longue distance. Les résultats confirment qu'une architecture YOLO de taille moyenne bien régularisée offre l'équilibre optimal entre précision de détection et efficacité computationnelle pour les applications de défense aérienne en temps réel.

Mots-clés :Détection d'objets aériens, Classification d'objets aériens, Réseau de neurones convolutif, YOLO, Détection de drones, Apprentissage profond, SAHI, Défense aérienne, Véhicule aérien sans pilote

الملخص

أدى الانتشار السريع للمركبات الجوية غير المأهولة وتزايدأفرز التطور المتسارع في مجال المركبات الجوية غير المأهولة والتعقيد المتزايد للمجال الجوي المنخفض الارتفاع حاجة ملحّة إلى أنظمة آلية قادرة على التعرف على التهديدات الجوية بسرعة ودقة عالية. يقدّم هذا المشروع تصميم وتطوير نظام متكامل مدعوم بالذكاء الاصطناعي للكشف عن الأجسام الجوية وتصنيفها، وقد تم تدريبه على مجموعتَي بيانات تغطيان سبع فئات متمايزة: الطائرات المسيّرة، والمروحيات، والطائرات، والطيور، والطائرات العسكرية، والصواريخ، والمركبات الجوية غير المأهولة. جرى تقييم ثلاثة معماريات تعلم عميق لمهمة التصنيف — YOLOv26x وEfficientNetV2-L وTransformer Swin — حيث حققت دقة تصنيف بلغت 40.99 و72.98 و19.99 على التوالي على مجموعة البيانات المخصصة .DAOD أما على صعيد الكشف المكاني عن الأجسام، فقد حقق نموذج YOLOv26x المدمج مع تقنية الاستدلال الفائق بالتقطيع SAHI قيمة mAP@0.5 بلغت 981.0 على مجموعة البيانات ،AOD4 و931.0 على مجموعة DAOD الأكثر تعقيداً، مما يُثبت قدرة النظام على التعميم عبر أحجام أهداف وخلفيات متنوعة. يعالج النظام تحديات جوهرية تشمل اختلال توازن الفئات، والتشابه البصري بين الطائرات المسيّرة والطيور، والكشف عن الأجسام الصغيرة على مسافات بعيدة. تؤكد النتائج أن معمارية YOLO متوسطة الحجم ذات تنظيم جيد توفر التوازن الأمثل بين دقة الكشف والكفاءة الحسابية في تطبيقات الدفاع الجوي الآني.

الكلمات المفتاحية: الكشف عن الأجسام الجوية، تصنيف الأجسام الجوية، الشبكات العصبية التلافيفية، ،YOLO الكشف عن الطائرات المسيّرة، التعلم العميق، ،SAHI الدفاع الجوي، المركبات الجوية غير المأهولة

Contents

List of Tables **iv**

List of Figures **v**

Acronyms **vii**

General Introduction **1**
Problem Statement 2
Research Objectives 2

1 Background and Literature Review **6**
1.1 Introduction 6
1.2 Background and Context 7
1.3 Key Concepts 8
1.3.1 AI-Powered Air Defense System 8
1.3.2 Aerial Threat Detection 10
1.3.3 Aerial Threat Classification 11
1.4 Types of Aircraft Threats 12
1.5 Evolution and Recent Developments 13
1.5.1 Evolution of Aerial Threat Detection and Classification 13
1.6 Related Work 14
1.7 Generalities On Artificial Intelligence(AI) 16
1.8 Subfields of Artificial Intelligence 17
1.8.1 Machine Learning 17
1.8.2 Deep Learning 20
1.9 YOLO (You Only Look Once) 21
1.10 EfficientNet 22
1.10.1 Key Concepts: 22
1.11 Swin Transformer 23
1.11.1 Key Concepts: 24

1.12 Evaluation Metrics of Models 25
1.13 Conclusion 27

2 Proposed Method 30
2.1 Introduction 30
2.2 Overview of the Study 30
2.3 Datasets 31
2.3.1 Aerial object detection dataset 4 (AOD4) 31
2.3.2 Why Aerial object detection dataset 4 32
2.3.3 Diverse Airborne Objects Dataset (DAOD) 33
2.3.4 Why Diverse Airborne Objects Dataset 34
2.4 Data Preprocessing 35
2.5 Cleaning Data 36
2.6 Training and Validation Split 36
2.7 Labeling Dataset 38
2.8 Training and Evaluation 39
2.8.1 Image Classification 40
2.8.2 Object Detection 41
2.8.3 Sahi with YOLO26x 42
2.9 Tracking 43
2.10 Training Hyperparameter 44
2.10.1 Justification of Hyperparameter Selection 45
2.11 Conclusion 45

3 Results & Discussion 48
3.1 Introduction 48
3.2 Development Environment 48
3.2.1 Hardware specifications 48
3.2.2 Software Environment 49
3.2.3 Libraries 49
3.3 to Threat Classification Performance Results 50
3.4 Per-Class Threat classification Performance Results 52
3.4.1 DAOD Dataset Performance 52
3.4.2 AOD4 Dataset Performance 57
3.5 Object Detection Performance Results 61
3.6 Comparison Performance Across All Tested Models 65
3.7 Discussion 66
3.7.1 Comparison Results with Models Proposed in Literature 66

3.8 Detecting and Tracking Graphical Interface 67
3.9 Challenges and Limitations 68
3.10 Future Work 69
3.10.1 Multi Modal Sensor Integration 69
3.10.2 Edge Deployment and Model Optimization 69
3.10.3 Continuous Learning and Adaptive Retraining . . . 70
3.11 Conclusion 70

General Conclusion **72**

Bibliography **74**

List of Tables

1.1 Summary of Related Work on AI-Based Aircraft Detection and Defense 16
1.2 Comparison of Machine Learning Methods 19
1.3 Deep Learning methods 20

2.1 Information about the datasets 35
2.2 Training Configuration per Model 45

3.1 YOLO Class-Level Performance on DAOD 53
3.2 Swin Transformer Class-Level Performance on DAOD . . . 54
3.3 EfficientNet Class-Level Performance on DAOD 56
3.4 Swin Transformer Class-Level Performance on AOD4 . . . 57
3.5 EfficientNet Class-Level Performance on AOD4 59
3.6 YOLO Class-Level Performance on AOD4 60
3.7 Object Detection Performance Metrics on the AOD4 Dataset 62
3.8 Object Detection Performance Metrics on the DAOD Dataset 63
3.9 Overall Classification Performance Across Datasets 65
3.10 Overall Object Detection Performance Across Datasets . . 66
3.11 Image Classification Accuracy Comparison on the AOD4 Dataset 67

List of Figures

1.1 Supervised Learning 18
1.2 Reinforcement Learning 18
1.3 Unsupervised Learning 19
1.4 Architecture of YOLOV11 Model. 22
1.5 EfficientNet Architecture V2. 23
1.6 The architecture of a Swin Transformer (Swin-T). 25
1.7 Confusion Matrix and Classification Metrics 26

2.1 Air Defense System with Detection and Classification Pipelines 31
2.2 Sample Images Class from the AOD4 Dataset 32
2.3 Sample Images from the DAOD Dataset 34
2.4 Comparison of Helicopter Samples: AOD4 vs. DAOD . . . 35
2.5 The Distribution of AOD4 Dataset 37
2.6 The Distribution of DAOD Dataset 38
2.7 Flowchart of the Dataset Annotation Process 39
2.8 Used Models for Training 43
2.9 Operational Workflow of the Aerial Object Tracking Module 44

3.1 Comparative Classification Accuracy and Macro F1 Scores 51
3.2 Object Detection Performance Comparison between the AOD4 and DAOD 52
3.3 Confusion Matrix for YOLO Model on DAOD dataset . . 54
3.4 Confusion Matrix for Swin Transformer Model on DAOD dataset 55
3.5 Confusion Matrix for EfficientNet on DAOD dataset . . . 57
3.6 Confusion Matrix for Swin Transformer Model on AOD4 dataset 58
3.7 Confusion Matrix for EfficientNet Model on AOD4 dataset 60
3.8 Confusion Matrix for Clasification YOLOv26x Model on AOD4 dataset 61

3.9 Confusion Matrix for Object Detection YOLOv26x on AOD4 dataset 63
3.10 Confusion Matrix for Object Detection YOLOv26x on DAOD dataset 65
3.11 Examples of misclassified objects in AOD4 model results. . 68
3.12 Examples of accurate classification in DAOD model results. 68

Acronyms

AI	Artificial Intelligence
AOD4	Aerial Object Detection Dataset 4
AP	Average Precision
AUC	Area Under Curve
CNN	Convolutional Neural Network
CSPNet	Cross Stage Partial Network
DAOD	Diverse Airborne Objects Dataset
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
FN	False Negative
FP	False Positive
FPN	Feature Pyramid Network
FPS	Frames Per Second
GUI	Graphical User Interface
IoU	Intersection over Union
mAP	Mean Average Precision
MBConv	Mobile Inverted Bottleneck Convolution
ML	Machine Learning
NMS	Non-Maximum Suppression
OOM	Out Of Memory
PAN	Path Aggregation Network
R-CNN	Region-based Convolutional Neural Network
RNN	Recurrent Neural Network
ROC	Receiver Operating Characteristic
SA	Self-Attention
SAHI	Slicing Aided Hyper Inference
SE	Squeeze-and-Excitation
SOD	Small Object Detection
TN	True Negative
TP	True Positive
UAV	Unmanned Aerial Vehicle
ViT	Vision Transformer
YOLO	You Only Look Once

General Introduction

In the modern era, characterized by rapid technological advancement and increasing security challenges, the ability to accurately monitor and analyze airspace has become critically important. Traditional air surveillance systems, which rely heavily on manual observation and conventional radar technologies, often struggle to cope with the growing complexity and volume of airborne objects. These limitations highlight the need for intelligent systems capable of processing large amounts of visual data and making fast, reliable decisions in real time.

Computer vision has emerged as a transformative approach that enables machines to interpret and understand visual information from the environment. With the rise of artificial intelligence, particularly deep learning, it is now possible to automatically extract meaningful features from images and perform complex recognition tasks with high accuracy. This advancement opens new opportunities for enhancing air defense systems through automated aerial object recognition.

In this work, visual data is utilized as the primary source of information, where images collected from diverse aerial scenarios are processed and analyzed. A convolutional neural network (CNN), known for its effectiveness in image classification tasks, is employed to learn distinguishing features from the dataset. Through this approach, the system is able to identify and categorize airborne objects based on their visual characteristics, enabling efficient differentiation between various types of entities present in the airspace.

Image classification, once addressed using traditional machine learning techniques with handcrafted features, has significantly evolved with the introduction of deep learning architectures. Modern CNN-based models automatically learn hierarchical representations from raw image data, improving both accuracy and generalization. To enhance model performance, this project incorporates data preprocessing, augmentation techniques, and optimized training strategies, ensuring robustness under varying environmental conditions such as lighting, scale, and background com-

plexity.

Furthermore, the system focuses on minimizing misclassification and false detections, which are critical challenges in real-world applications. By improving the reliability of predictions, the proposed framework contributes to building a dependable first layer of automated airspace monitoring.

By integrating advanced computer vision techniques with artificial intelligence, this work presents a practical framework for aerial object classification that can support intelligent air defense systems. The objective of this research is to provide an efficient and scalable solution for automated visual recognition, paving the way for future developments in object detection, tracking, and autonomous response systems. Ultimately, this contributes to more adaptive, accurate, and intelligent security infrastructures capable of addressing evolving aerial threats.

Problem Statement

Protecting airspace is one of the most prominent challenges facing modern armies, especially in light of the increasing and diverse air threats such as drones and guided missiles. Despite technological advancements, some defense systems still rely in part on conventional methods that limit their ability to detect flying objects early and accurately, especially those with a weak radar footprint or flying at low altitudes.

Research Objectives

The primary objectives of this research are as follows:

- Develop an end-to-end computer vision system to detect and classify aerial objects in real time using deep learning. The system will be able to classify the objects and detect their spatial location on the screen.
- The system must also function in different environmental conditions and at different distances to the target. For this purpose, the system must undergo a rigorous data preprocessing and utilize techniques for image augmentation. Furthermore, the system must utilize specialized inference techniques for the detection of small objects, such as the SAHI.

- Implement temporal tracking of detected aerial threats in real time by associating corresponding objects in subsequent frames of a video to enable accurate tracking and situational awareness.

Thus, the main purpose of this research is to contribute to the development of computer vision technology in air defense systems.

This thesis is organized into three chapters:

- **Chapter 1: Background and Literature Review:**

  In this chapter, the key concepts of aerial object recognition, image classification, Object Detection, and computer vision are discussed, focusing on their significance in modern air defense systems. Furthermore, an overview of artificial intelligence, including machine learning, deep learning, and convolutional neural networks, is provided, which is the basis of image classification systems. This chapter is concluded by a discussion of related work and existing approaches in the field of aerial object detection and classification.

- **Chapter 2: Research Methodology:**

  This chapter focuses on the methodology that has been used in this study, including the dataset that has been used for training and evaluation, data collection methods, and preprocessing methods such as normalization and data augmentation. It also includes the design and implementation of the convolutional neural network model and the training process that has been used to improve the performance of the model.

- **Chapter 3: Results and Discussion:**

  The experimental setup, including the environment and tools used, is discussed in this chapter. Subsequent to that, the evaluation of the proposed model based on relevant performance measures, i.e., accuracy, precision, recall, and F1-score, is discussed. In addition, a comparative evaluation is provided to assess the efficacy of the proposed model, followed by a discussion of the obtained results and their implications for aerial object classification and air defense systems.

Chapter 1: Background and Literature Review

Chapter 1

Background and Literature Review

1.1 Introduction

The nature of aerial threats has changed faster than the systems designed to counter them. For decades, air defense relied on a straightforward premise: radar detects, humans decide, systems respond. That premise held when threats were large, fast, and few. It no longer does. Today's airspace is contested by objects that were never anticipated when legacy surveillance infrastructure was designed small commercial drones repurposed for reconnaissance or attack, low-altitude unmanned vehicles that hug terrain to stay below radar coverage, and swarms of cheap, expendable platforms that overwhelm any system dependent on human attention. These threats do not announce themselves. They are slow enough to evade velocity based detection, small enough to disappear against cluttered backgrounds, and numerous enough to exhaust the cognitive capacity of any operator team. The response to this shift cannot simply be more radar or more personnel. It requires a fundamentally different kind of perception one that operates at machine speed, learns from visual data, and scales without degrading. Deep learning has made this possible. Convolutional neural networks, trained on large and diverse image datasets, can now distinguish a drone from a bird, a military aircraft from a civilian one, and a missile from background noise even at distance, under poor lighting, and against complex backgrounds. These are discriminations that would challenge a trained human analyst with unlimited time; a well-trained model makes them in milliseconds. This research is grounded in that capability. Rather than treating computer vision as a supplementary tool bolted onto existing defense infrastructure, this work positions it as the perceptual foundation of an integrated air defense pipeline the first and most critical layer of awareness on which detection, classification, tracking, and

ultimately response decisions depend.

1.2 Background and Context

In recent times, there have been major developments in artificial intelligence, along with an exponential increase in digital data. This has significantly impacted several fields, including surveillance and defense technologies. In addition, the availability of quality image data along with adequate computational power has enabled the development of intelligent technologies that can analyze and interpret complex image-based environments. This has opened avenues to apply computer vision technology in critical surveillance tasks such as airspace surveillance and recognition of objects in the air.

Traditional surveillance technologies use radar-based technologies along with human intervention to identify airborne objects. Even though such technologies have been widely implemented, there are several limitations associated with such technologies when implemented in complex environments. This includes a lack of ability to accurately identify airborne objects, which may result in false alarms. This significantly affects the efficiency of such surveillance technologies. Recently, deep learning-based CNN architectures have been widely implemented to classify images. This technology has significantly improved image classification tasks, as such architectures can learn features from images. This enables such architectures to accurately classify images. In addition, such architectures have been widely implemented in several computer vision tasks, such as object recognition, medical imaging, etc.

In the context of aerial monitoring systems, classification systems based on images can be a supplementary solution to the traditional approach by using visual data for identifying and classifying aerial objects in the airspace. This approach can be particularly useful in situations in which radar systems may be ineffective or when confirmation is needed. Using visual patterns such as shape, motion, and texture, deep learning algorithms can effectively classify various types of aerial objects.

However, there are still challenges in ensuring high accuracy and robustness in these systems for real-world applications. There are various factors that can influence the performance of these systems. Therefore, the development of an efficient and reliable image classification system is still a significant research issue in various fields, particularly in safety-critical applications such as air defense systems.

This research is motivated by the need to create an intelligent system that can efficiently classify aerial objects in images for the enhancement of automated airspace monitoring systems and for contributing to the development of intelligent defense systems in the future[1].

1.3 Key Concepts

To establish a clear basis for this research, it is necessary to define the major concepts used during this work.

1.3.1 AI-Powered Air Defense System

Building on this foundation, we now turn to the concept of AI-Powered Air Defense System.

a) Definition of AI-Powered Air Defense System

The study focuses on an air defense system designed to identify, analyze, and respond to aerial attacks by employing artificial intelligence methods and data. Such defense systems integrate various sources of information, including radar, sensor information, and visual information. These defense systems identify various objects such as drones, aircraft, and missiles in the air. Conventional air defense systems utilize rules to monitor information. However, these rules impose limitations on these defense systems in dealing effectively with complex situations, Modern air defense systems utilize machine learning and deep learning methods to improve detection accuracy, speed, and flexibility.

b) Importance of AI-Powered Air Defense System

The application of AI-based air defense systems plays a crucial role in the military and security operations of the present day. This helps in faster and more accurate decision-making against aerial attacks. This intelligent system uses data collected from various sources, including radars, sensors, and visual surveillance systems, The accurate detection of aerial threats helps defense systems respond accordingly by adopting the right strategies to neutralize the threat. This minimizes the extent of damage and provides maximum security. In addition, these intelligent systems help reduce the involvement of human error, thereby reducing the time taken to respond to aerial attacks. In unpredictable and ever-changing scenarios, intelligent air defense systems provide defense forces with the capability to respond

to emerging aerial threats. This includes drones and advanced missile systems. This helps them stay prepared to face any challenge.

c) Types of AI-Powered Air Defense System

The different types of AI-powered air defense system models based on their functionality and type of data processed are as follows:

- **AI-enhanced radar and multisensor detection:** Deep learning is integrated to ground-based radar to enable improved discrimination between real targets and background clutter in real-time and on current operational radar systems[2]. Multi-sensor networks (LP + mini-radars and optics) are managed by AI for the detection, tracking and recognition of "killer drones" by means of their micro-Doppler signatures[3]
- **Object recognition from imagery:** M couple of prototypes for autonomous air defense from various countries employ CNNs and also a YOLO-model to detect and then track various kind of air objects such as drones and various kinds of missiles and aerial warheads by using cameras[4][5][6].
- **Missile/munition type identification:** An AI missile target identification model on a memristor array in seconds will accurately identify the types of missiles and other munitions (e.g. non threat, mortars, rockets, etc.)[6].

d) Role of AI-Powered Air Defense System

The role of AI-based air defense systems in the military and defense sector is critical. In the context of surveillance and monitoring, AI-based systems are employed to monitor the airspace and identify aerial threats like drones, aircraft, and missiles. Defense forces use AI models to analyze the level of threats and the pattern of movement of these aerial threats. In critical defense operations, AI-based systems are employed to make quick decisions in critical defense operations. AI models are used to process vast amounts of data with high accuracy and make quick decisions. This helps in the efficient identification of the level of threats and the priority of addressing them. AI-based systems are also employed by the government and defense organizations in the context of strategic planning and the protection of the airspace and critical infrastructure from emerging aerial threats.

e) Advantages of AI-Powered Air Defense System

There are a number of advantages associated with the use of AI-based air defense systems. Firstly, the decision-making process is improved, allowing the operators of the air defense system to make the right decisions based on the information obtained. Secondly the risk of a successful attack is minimized, as the AI-based system can identify the threats at an early stage, thereby reducing the risks involved in the process. The air defense system also offers a strong operational advantage, as the responses of the system can be made faster and more accurately, as opposed to the conventional method of operation. Thirdly, the use of the AI-based system improves the level of automation and efficiency, as the use of the AI model improves the level of accuracy while reducing the human intervention required in the process. Finally, the use of the real-time situation awareness is improved by the use of the data obtained from the sensors, thereby allowing the air defense system to adapt to the dynamic and unpredictable environment.

1.3.2 Aerial Threat Detection

This section provides an overview of Aerial Threat Detection.

a) Definition of Aerial Threat Detection

Aerial threat detection is a computer vision problem in which objects are localized in aerial images to determine if they are potential threats. It combines object detection using bounding boxes with object classification to determine the type of objects. This enables the detection of objects in aerial images and differentiates them from background objects.

b) Types of Aerial Threat Detection Approaches

Aerial threat detection methods can be categorized based on their architecture and processing strategy

- **Two-stage detection approaches:**These methods, like Faster R-CNN, first propose regions and then classify each region. They tend to be more accurate but require more computation time and are not preferred for strict real-time requirements.[7]
- **One-stage detection approaches:**Object localization and classification are done in a single forward pass in models like YOLO (You

Only Look Once). Such models are commonly used in real-time systems due to their efficiency in terms of speed and accuracy[8].

c) Role of Aerial Threat Detection in This Work

in this thesis, the concept of aerial threat detection is considered as the primary functionality of the proposed system. In this context, the proposed model analyzes the aerial images, detects the objects using bounding boxes, and assigns the respective class labels to the objects.

d) Advantages of Aerial Threat Detection

There are a number of significant advantages that aerial threat detection provides in surveillance and air defense applications:

- **Better situational awareness:** It allows for the identification and localization of aerial objects, giving a better understanding of the surveillance area.
- **Real-time processing:** Deep learning-based systems have the ability to process images in real-time, making it suitable for surveillance applications.
- **Improved accuracy:** Deep learning-based systems increase the accuracy of detection by minimizing false alarms in a complex environment with multiple targets.
- **Automation of surveillance:** It minimizes the need for constant surveillance by humans by automating the detection and identification process.
- **Decision support:** It provides a better understanding of the surveillance area by giving accurate information in real-time, helping make quick and informed decisions.

1.3.3 Aerial Threat Classification

a) Definition of Aerial Threat Classification

Aerial Threat Classification is a computer vision problem that aims to assign a particular class label to a detected aerial object based on its visual features. Following object detection and localization in a given image, classification is a process that identifies what kind of aerial object is being detected. This is a crucial process in identifying different objects in the air,

including those that may be of relevance, In terms of deep learning, Aerial Threat Classification is usually done by employing convolutional neural networks that learn discriminative features from a given dataset. These neural networks examine features such as shape, size, and motion-related features in order to classify a given detected object.

b) Importance of a Aerial Threat Classification

Aerial threat classification is an important feature of surveillance and air defense systems. This is because it helps the system not only recognize that there are objects in the air but also understand the nature of such objects. By classifying the detected objects, the system can recognize the various kinds of objects that are in the air and identify which ones could potentially pose a threat.

c) Role of Aerial Threat Classification

One of the main uses of aerial threat classification is in the interpretation and understanding of aerial objects that are detected within a surveillance system. Following the detection and localization of an object. classification is used to assign a particular class to the detected object, thus enabling it to be distinguished from other objects, In this particular study, aerial threat classification is used in the conversion of raw detection data into useful information that may be used in decision-making. This ensures that it is possible to distinguish those objects that are of concern in surveillance from those that are of a non-threatening class.

1.4 Types of Aircraft Threats

There exist numerous types of aerial threats, including various kinds of airborne objects that might be used as a source of security risks or pose danger for some infrastructures and military activities. The development of technologies leads to an increase in diversity among various aerial threats, which include traditional piloted aircrafts as well as new devices, such as drones and hypersonic weapon delivery platforms. Hence, it is vital to be familiar with the types of aerial threats in order to conduct proper analysis due to the distinctive traits of each type.

- **Manned Aircraft:** The manned aircraft is an airborne vehicle that has been flown manually by a person. This includes military aircraft like fighter planes, bombers, and helicopters. These may have

weapons like missiles and bombs. Missions they undertake include attacks, interceptions.[9].

- **Drones (UAVs)** UAVS stand for Unmanned Aircraft Vehicles and refer to unmanned aerial vehicles, better known as drones. They are unmanned aircraft that can be operated manually or automatically. They are mostly used for surveillance, reconnaissance purposes, and nowadays for combative missions. They are inexpensive and hard to notice[10].
- **Missiles** refer to guided projectiles that have been created to carry a payload which is mainly explosive to hit a target accurately. They use power sources such as rocket or jet engines to propel them through space and rely on guidance technology [11].

1.5 Evolution and Recent Developments

The evolution of aerial threat detection, identification, classification, and coordination of responses provides insights into the increased significance of AI in the field of aerial defense.

1.5.1 Evolution of Aerial Threat Detection and Classification

Earlier models for detecting aerial threats mainly used conventional radar technology and identification rules, which faced problems while processing complex environments and were prone to a lot of false positives. The incorporation of machine learning models during the early 2000s allowed for the use of classification algorithms like support vector machines and decision trees, but still needed human interference for selecting features.

The development of deep learning models that included CNN and RNN models during the 2010s was quite effective in processing complex radar information. These models are useful in identifying complex aerial images. Recently, models that incorporate AI and sensors have been employed to integrate radar signals along with infrared and communication data. Multi-device edge AI systems use radar sensing plus wireless links to send quantized features to an edge server; AI manages end-to-end sensing–computation–communication to maximize classification accuracy[12].

1.6 Related Work

The authors of [13] have created an autonomous threat detection system using DCNN that is designed on the basis of deep convolutional neural network along with Django-based implementation. For detecting different objects in the air and classifying whether it is a military aircraft or civilian one, AlexNet and ResNet architecture has been used. Meanwhile, the system uses continual learning to adjust to different types of threats. According to the outcome of the research, it was found that DCNN systems improve the accuracy of detection up to 95% more compared to previous methods, decrease system latency rate by 40%, and improve adaptability by 30%.

In [14], a deep learning model using a dual-task framework with EfficientNetB4 as the backbone for airborne objects classification and threat level prediction is presented. The lack of clean and evenly distributed training data is compensated for by introducing the AODTA dataset, which consists of combining and filtering several publicly available data sources with regards to four categories of airborne objects: airplane, drone, helicopter, and bird. EfficientNetB4 proved superior to the ResNet-50 used as a baseline model which was consistently outperforming it. EfficientNetB4 reached 96% accuracy for object classification and 90% for threat level prediction on the AODTA data, while maintaining precisions and recalls above 0.9 for all types of airborne objects. It is proven feasible and efficient to incorporate object classification and threat prediction within one deep learning model.

The research in [8] aims to leverage YOLOv11 for object detection and BoT-SORT for multi-object tracking in order to create a real-time aircraft detection and tracking framework for the purpose of national air defense. The YOLOv11 framework was fine-tuned over 200 epochs on a dataset comprising 1,800 hand-labeled civil and military aircraft images obtained from sources such as Roboflow, YouTube, and Instagram via data augmentation methods like flipping, rotating, and adjusting exposure. YOLOv11 is compared against competing approaches, such as YOLOv9, YOLOv10, and RT-DETR, as well as the ByteTrack tracking framework. The results of experiments showed that YOLOv11 delivered the best results in terms of aircraft detection accuracy, achieving mAP50 of 0.934 and recall of 0.918, whereas BoT-SORT delivered high tracking stability in terms of mIDF1 = 0.786 and a frame rate of 71.43 FPS, being superior to ByteTrack by both metrics. It can be concluded that the presented solution based on

YOLOv11 and BoT-SORT represents a promising basis for implementing visual AI-based analytics for national air defenses, thus overcoming the shortcomings of radar systems in detection of stealth and low-altitude aircraft.

The authors in reference [15] propose a YOLOv8-based method for detecting fighter aircraft patterns implemented in a web-based application interface for real-time analysis. The dataset used is made up of 6,178 images containing 40 types of military aircrafts, such as A-10, F-4, Rafale, Tornado, and Vulcan, with each image annotated with corresponding bounding boxes and class labels for training. The proposed model is tested against a CNN model and a ResNet-50 based model with respect to precision-recall curves and Mean Average Precision. As a result, YOLOv8 model scored a mean AP of 0.860 at an IoU threshold of 0.5 and showed superior performance compared to CNN and ResNet-50 models for aircrafts detection from various angles and under different lighting. Therefore, it can be concluded that YOLOv8-based models greatly lower misclassification rate and are a robust approach to aircraft pattern recognition.

Authors	Approach	Model(s) Used	Application Area	Key Insight
Chatterjee & Zaman	Dual-task deep learning for simultaneous object classification and threat-level prediction	EfficientNetB4, ResNet-50 (baseline)	Airborne object surveillance & defense	EfficientNetB4 outperforms ResNet-50, achieving 96% class accuracy and 90% threat-level accuracy on the custom AODTA dataset; multi-task framing is effective.
Kannan A, Barath S.S, Dr. Nivetha	DCNN-based autonomous threat detection deployed via Django framework	AlexNet, ResNet (DCNN variants)	Aircraft defense & autonomous systems	DCNN-based systems improved object detection accuracy by over 95%, reduced response time by 40%, and increased threat adaptability by 30%.
Dinisfusya'ban, Bambang Suharjo et al.	Real-time detection and multi-object tracking using a combined YOLO + tracker pipeline	YOLOv11, BoT-SORT (vs. ByteTrack, YOLOv9/10, RTDETR)	National airspace defense (Indonesia)	YOLOv11 + BoT-SORT achieves mAP50 = 0.934 and 85.47 FPS; outperforms radar-only systems for stealth/low-altitude aircraft; BoT-SORT superior to ByteTrack in identity consistency.
Jayadharshini P, Vasuki C et al.	Pattern detection for fighter aircraft using deep learning and a web-based deployment	YOLOv8, CNN, ResNet-50	Aviation security & airspace surveillance	YOLOv8 achieves mAP@0.5 = 0.860 across 40 military aircraft types; outperforms CNN and ResNet-50 in real-time detection under varied orientations and lighting conditions.

Table 1.1: Summary of Related Work on AI-Based Aircraft Detection and Defense

1.7 Generalities On Artificial Intelligence(AI)

Artificial Intelligence (AI) has become a cornerstone of modern technology, influencing a wide range of domains. it has developed rapidly in the last few decades, revolutionizing various industries and reshaping the way humans interact with technology. From self-driving cars to advanced medical diagnostics, AI-driven systems are becoming increasingly integral to daily life.

At its core, AI is a multidisciplinary field of computer science that

aims to create systems capable of performing tasks that typically require human intelligence. These tasks include problem-solving, decision-making, learning, perception, and natural language understanding.

AI is designed to perform specific functions, such as image recognition, speech processing and recommendation systems. It can also undertake any intellectual work that a human can do. While general AI remains a theoretical concept, narrow AI applications have seen widespread adoption across numerous industries.

The evolution of AI has been fueled by advancements in computational power, improved algorithms, and the availability of large-scale data. It draws from multiple disciplines: mathematics (logic and probability) for formal reasoning, neuroscience for inspiration in neural network architectures, linguistics for natural language processing, cybernetics for control systems, experimental psychology for behavior modeling, and economics for decision and game theory. Together, these fields form the foundation upon which modern AI continues to grow and innovate.

1.8 Subfields of Artificial Intelligence

Artificial intelligence is composed of two primary components: machine learning and deep learning.

1.8.1 Machine Learning

Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to perform a specific task without explicit programming. Learning algorithms are used in many applications that we make use of daily. Every time a web-search engine like Google is used to search the internet, one of the reasons it works so well is because of a learning algorithm that has learned how to rank web pages. These algorithms are used for various purposes like data mining, image processing, predictive analytics, etc. to name a few. The main advantage of using machine learning is that, once an algorithm learns what to do with data, it can do its work automatically[16]. Machine Learning algorithms are mainly divided into four categories: Supervised learning, Unsupervised learning, Semi-supervised learning, and Reinforcement learning, these will be explained below.

Supervised Learning

Supervised learning uses labeled data to train models for classification and regression tasks. The model learns by comparing predictions with known outputs, as shown in Figure 1.1[16]. It involves training, evaluating, tuning, and deployment[17].

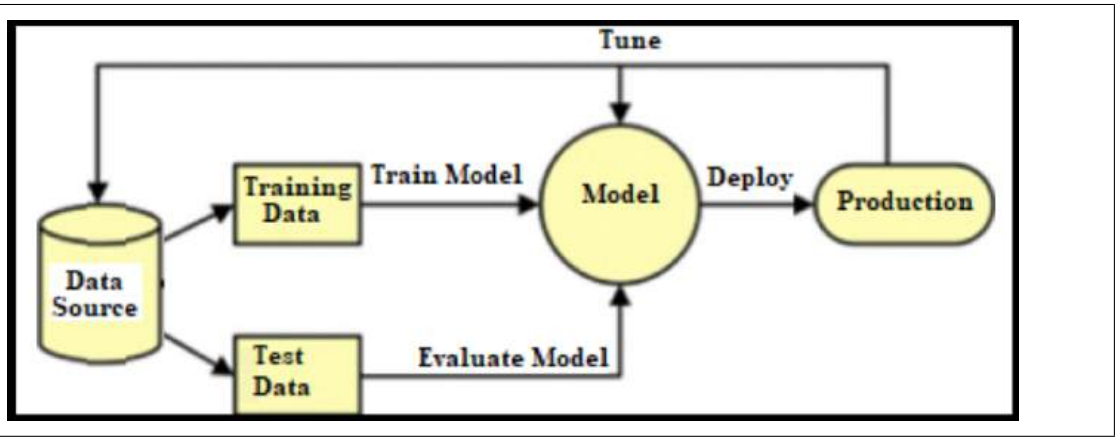


Figure 1.1: Supervised Learning

Semi-Supervised Learning

This approach combines a small amount of labeled data with a large set of unlabeled data, improving learning efficiency when labels are scarce.

Reinforcement Learning

In reinforcement learning, an agent interacts with an environment to maximize rewards, learning from feedback over time, as shown in Figure 1.2[16].

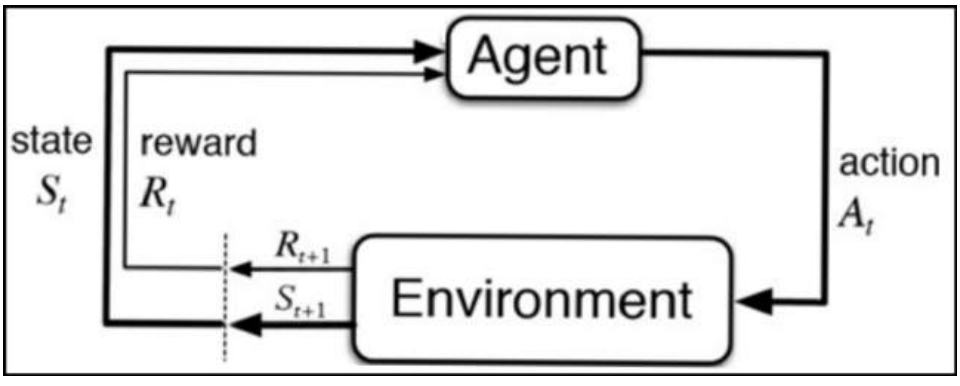


Figure 1.2: Reinforcement Learning

Unsupervised Learning

Unsupervised learning finds patterns in unlabeled data without human input. As in Figure 1.3[16], it is mainly used for clustering, association, and dimensionality reduction[17].

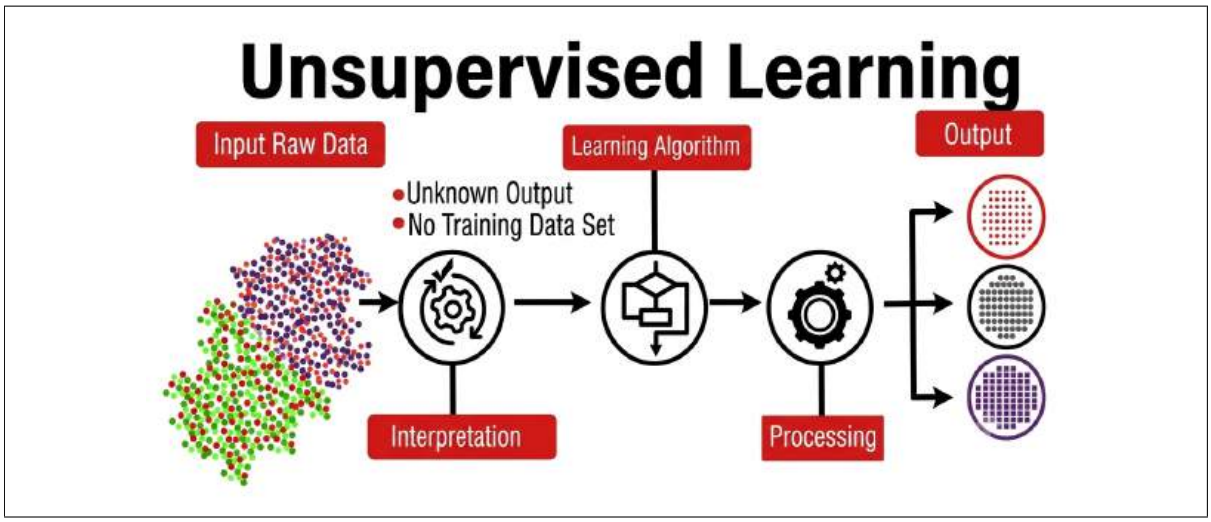


Figure 1.3: Unsupervised Learning

Methods	Description	Data Type	Common Algorithms	Applications
Supervised Learning	Learns from labeled data by mapping inputs to outputs.	Labeled data	Decision Trees, SVM, Neural Networks	Spam detection, Medical diagnosis, Sentiment analysis
Semi-Supervised Learning	Uses a small amount of labeled data and a large amount of unlabeled data.	Mix of labeled & unlabeled data	Self-training, Co-training, Graph-based methods	Speech recognition, Text classification, Medical imaging
Reinforcement Learning	Trains an agent to make sequential decisions based on rewards.	Interaction with environment	Q-Learning, Deep Q-Networks (DQN), Policy Gradient Methods	Game playing (e.g., AlphaGo), Robotics, Autonomous driving
Unsupervised Learning	Identifies patterns or structures in unlabeled data.	Unlabeled data	K-Means Clustering, PCA, Autoencoders	Customer segmentation, Anomaly detection, Topic modeling

Table 1.2: Comparison of Machine Learning Methods

1.8.2 Deep Learning

Deep Learning (DL) is a specialized branch of Machine Learning that focuses on training artificial neural networks to recognize patterns and make decisions. Inspired by the structure and function of the human brain, deep learning models consist of multiple layers of artificial neurons, allowing them to learn complex representations of data[18].

Deep Learning Methods

Some of the most effective techniques in deep learning are designed to accelerate training and improve model performance. These methods address common challenges such as overfitting, slow convergence, and instability during training. The following section introduces several widely adopted strategie, ranging from optimization algorithms to architectural enhancements, each with its own strengths and potential drawbacks. Table 1.3[19] outlines these techniques, providing a concise summary of their purpose, benefits, and limitations, and illustrating their relevance in different deep learning contexts.

Method	**Description**	**Merits**	**Demerits**
Backpropagation	Used in optimization problems	Calculates gradient efficiently	Sensitive to noisy data
Stochastic Gradient Descent	Finds optimal minimum in optimization problems	Avoids trapping in local minimum	Longer convergence time
Learning Rate Decay	Reduces learning rate gradually	Increases performance, reduces training time	Computationally expensive
Dropout	Drops units/connections during training	Prevents overfitting	Increases training time
Max Pooling	Applies a max filter	Reduces dimensions	Might lose important features
Batch Normalization	Normalizes inputs per batch	Stabilizes training, speeds up learning	Extra computation overhead
Transfer Learning	Transfers knowledge from one model to another	Faster training, improves performance	Works only for similar tasks

Table 1.3: Deep Learning methods

1.9 YOLO (You Only Look Once)

YOLO is a family of deep-learning models for real-time object detection. It detects and classifies objects in a single pass through a neural network, making it much faster than many older methods, while keeping competitive accuracy [20][21]. The strength of YOLO lies in its ability to quickly analyze entire images or video frames, dividing them into a grid and predicting bounding boxes and class probabilities simultaneously. This efficiency positions YOLO as a prominent solution in applicationsrequiring rapid and accurate object detection.

The Architectural Trinity: The YOLO architecture is divided into three functional segments:

1. **Backbone (Feature Extractor):** During this phase, the backbone network is a CSPDarknet or an EfficientViT. The 'backbone' is used to transform the raw pixel data to high-level features by downsampling, which makes it simpler to extract patterns from the input image data. This transformation makes it possible for the AI algorithm to differentiate between the military target and the natural environment.

2. **Neck (Feature Aggregator):**It acts as an intermediate layer, which merges and refines the features extracted at various levels of the backbone network. This layer typically employs feature pyramids, including FPN (Feature Pyramid Network) or PAN (Path Aggregation Network). The use of this layer helps the model to detect objects of different sizes through their detailed and semantically rich features at multiple scales.

3. **Head (Prediction Module):**Finally, there is the output layer, which is tasked with the creation of prediction outputs. This involves providing coordinates of the boxes, objectness score, and probability of classes. The outputs can be created in multiple scales to ensure more effective object detection, especially for small and mid-size objects.

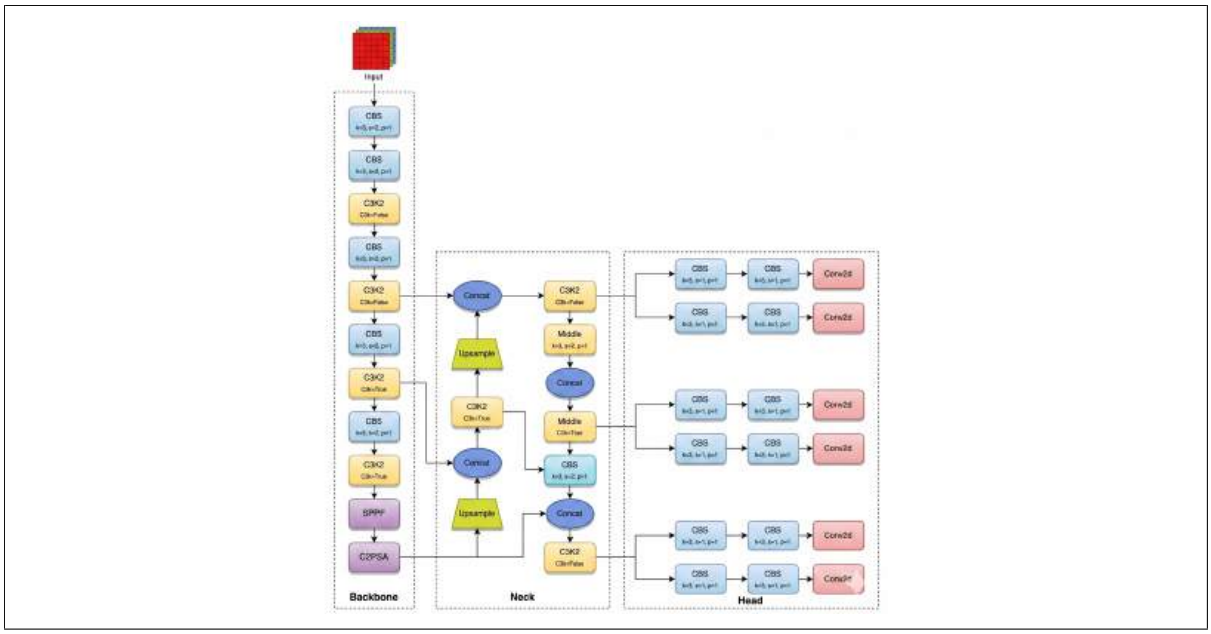


Figure 1.4: Architecture of YOLOV11 Model [22]

1.10 EfficientNet

EfficientNet is an advanced CNN model used to classify images efficiently and effectively in the fields of deep learning and computer vision. This architecture uses a compound scaling approach, where all the dimensions of the model, including network depth, network width, and input image size, are scaled in a systematic manner using fixed scaling factors, resulting in higher accuracy with low computational costs. In contrast to conventional CNN architectures that have independently scalable dimensions, EfficientNet scales these dimensions efficiently and ensures higher performance. The model is created using MBConv blocks together with squeeze-and-excitation optimization, providing better features while reducing the number of parameters and operations with floating numbers. EfficientNet has emerged as one of the most widely-used CNN models due to its capability to offer high accuracy with low computational costs.

1.10.1 Key Concepts:

1. **Compound Scaling:**A scaling technique where depth, width, and image size are scaled up in a synchronized way to boost both effectiveness and efficiency[23]

2. **MBConv Blocks:** Enhancing the layers in the neural network to enable deeper learning.

3. **Depthwise Separable Convolution:** A lightweight convolution technique that decreases the number of computations and parameters compared to standard convolution operations
4. **Squeeze-and-Excitation (SE):** A mechanism for recalibrating features in order to assist the network in focusing on the key image channels
5. **Swish Activation Function:** An activation function employed by EfficientNet that promotes gradient flow and improves learning efficiency.
6. **Residual Connections** Skip connections that help avoid vanishing gradient problems and improve deep network training stability

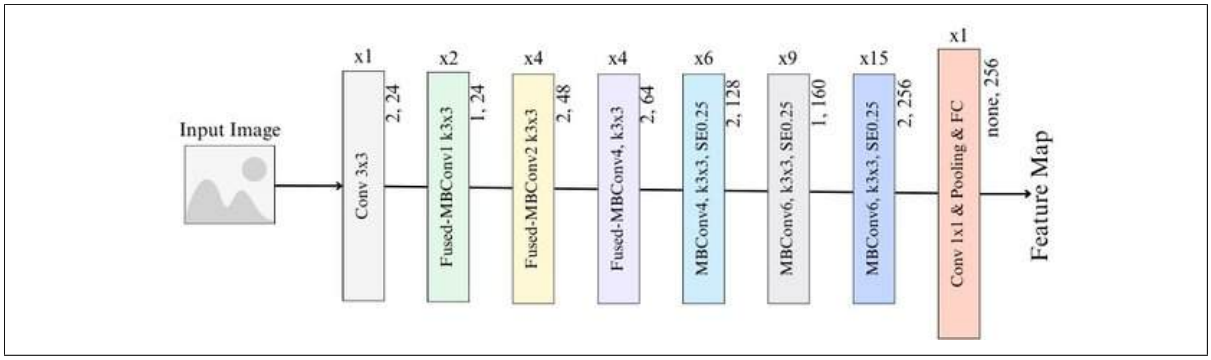


Figure 1.5: EfficientNet Architecture V2 [24].

1.11 Swin Transformer

The Swin Transformer is an advanced Vision Transformer (ViT) model used to classify and detect objects efficiently and effectively in the fields of deep learning and computer vision. Unlike conventional transformers that compute self-attention globally across the entire image, the Swin Transformer adopts a hierarchical architecture based on shifted windows, where attention is computed locally within non-overlapping windows and progressively merged across layers. This approach results in linear computational complexity with respect to image size, enabling the model to handle high-resolution images with significantly lower computational costs. The model produces hierarchical feature maps at multiple scales[25], making it suitable for both image classification and dense prediction tasks such as object detection and segmentation. The Swin Transformer has emerged as one of the most widely-used Vision Transformer architectures due to its

capability to combine the local feature extraction strength of CNNs with the long-range dependency modeling power of transformers.

1.11.1 Key Concepts:

1. **Shifted Window Self-Attention:** A mechanism that computes self-attention within local non-overlapping windows and shifts them across layers to enable cross-window connections, capturing both local and global spatial relationships[26].

2. **Hierarchical Feature Maps:** Unlike standard ViTs that maintain a single resolution throughout, the Swin Transformer progressively merges image patches across stages to produce multi-scale feature representations, similar to the feature pyramids used in CNNs.

3. **Patch Partitioning and Embedding:** The input image is divided into non-overlapping patches, each treated as a token and projected into a high-dimensional embedding space, serving as the fundamental input unit for the transformer layers.

4. **Patch Merging:** A downsampling operation applied between stages that concatenates neighboring patches and reduces the spatial resolution while increasing the channel dimensions, enabling the model to capture features at progressively larger scales.

5. **Relative Position Bias:**A learnable bias term added to the attention computation that encodes the relative spatial positions of tokens within each window, improving the model's sensitivity to the geometric structure of the input image.

6. **Layer Normalization:** A normalization technique applied before each attention and feed-forward block in the transformer, stabilizing the training process and improving convergence compared to batch normalization used in CNN architectures.

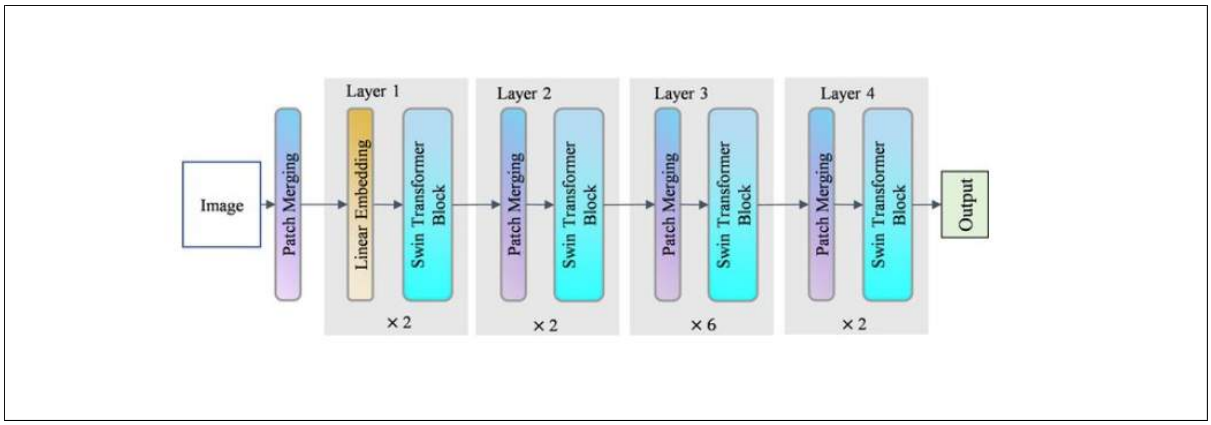


Figure 1.6: The architecture of a Swin Transformer (Swin-T)[27].

1.12 Evaluation Metrics of Models

Evaluation metrics are quantitative measures used to assess the performance and effectiveness of a statistical or machine learning model. These metrics provide insights into how well the model is performing and help in comparing different models or algorithms. When new machine learning models are created, it is necessary to compare their performance to the already existing ones. Evaluation serves two purposes: methods that do not perform well can be discarded, and the ones that seem promising can be further optimized[28].

For Classification:

- **Accuracy:** The accuracy of a classification model is defined as the proportion of instances that were correctly classified compared to all instances in the dataset. Accuracy can be expressed as a ratio or percentage, and is calculated as follows:

$$\text{Accuracy} = \frac{\text{Correct Predictions}}{\text{Total Predictions}}$$

- **Precision (Positive Predictive Value):** Precision measures the proportion of true positive predictions among all positive predictions.

$$\text{Precision} = \frac{TP}{TP + FP}$$

- **Recall (Sensitivity or True Positive Rate):** Recall measures the proportion of true positives among all actual positive instances.

Calculated using the following formula:

$$\text{Recall} = \frac{TP}{TP + FN}$$

- **F1-Score:** F1-Score is the harmonic mean of precision and recall, balancing both.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Confusion Matrix:** A confusion matrix is a table showing the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).
- **ROC Curve (Receiver Operating Characteristic Curve):** The ROC curve is a graphical representation of the trade-off between true positive rate and false positive rate.
- **AUC (Area Under Curve):** AUC measures the area under the ROC curve, indicating the model's ability to distinguish between classes.

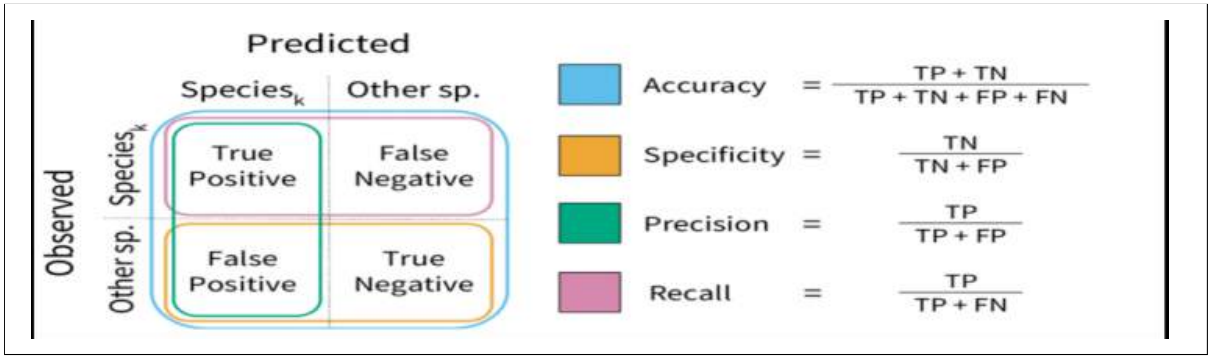


Figure 1.7: Confusion Matrix and Classification Metrics

This figure 1.7 shows the confusion matrix along with key classification metrics explained previously: accuracy, specificity, precision, and recall.

For Object Detection:

- **Intersection over Union (IoU):** IoU measures the overlap between the predicted bounding box (B_p) and the ground truth bounding box (B_g). It determines whether a prediction is a "hit" or a "miss" based on a set threshold (usually 0.5).

$$\text{IoU} = \frac{\text{Area of } (B_p \cap B_g)}{\text{Area of } (B_p \cup B_g)}$$

- **Average Precision (AP):** AP calculates the area under the Precision-Recall curve. Because a model outputs a confidence score for each box, varying the score threshold changes the precision and recall; AP summarizes this entire trade-off into a single number for a specific class.

$$\text{AP} = \int_0^1 p(r)\, dr$$

(Where $p(r)$ is precision as a function of recall)

- **Mean Average Precision (mAP):** mAP is a primary benchmark metric for object detection. It is the average of the Average Precision (AP) calculated across all K distinct object classes in the dataset.

$$\text{mAP} = \frac{1}{K} \sum_{i=1}^{K} \text{AP}_i$$

- **Mean Average Precision ($\textbf{mAP}_{50:95}$):** This is the standard benchmark metric used in modern datasets like Microsoft COCO. It represents the overall average mAP calculated across ten different IoU thresholds, spanning from 0.50 to 0.95 with step increments of 0.05 $(0.50, 0.55, 0.60, \ldots, 0.95)$. This metric strictly rewards models that achieve highly precise bounding box localizations.

$$\text{mAP}_{50:95} = \frac{1}{10} \sum_{\text{IoU}=0.50}^{0.95} \text{mAP}_{\text{IoU}}$$

1.13 Conclusion

This chapter outlines the necessary theoretical groundwork which will allow for a more in-depth exploration of the intersection between AI and Airspace Security in the subsequent chapter. A historical analysis between Radar mechanics and Deep learning have highlighted the key areas in which a range of different architectures can provide a means to automate the task of observation in Airspace. In particular the chapter outlines the current state of the art for Real-time Detection architectures, such as the YOLO family, as well as the current state of the art in compound scaled architectures for classification such as EfficientNet. Additionally the means by which to test models for low-altitude Airspace such as by utilizing visual similarity were highlighted. Furthermore a range of key

metrics and means by which to test a number of key properties of models for Airspace Security were outlined. A number of key concepts have been established that will allow for the following chapter of structural data engineering and a multi-architecture methodology for Airspace Security.

Chapter 2: Proposed Method

Chapter 2

Proposed Method

2.1 Introduction

The following chapter describes the methods employed, and the frameworks established to carry out this research. Furthermore, the chapter gives an overview of all the sequential engineering steps involved in this project, from acquiring and cleaning visual data, through to the process of annotating the data accurately. In addition, this chapter details the selection procedure for the optimum deep learning architecture, as well as an extensive analysis of how to configure the hyperparameters of these architectures appropriately. Every methodological step is a critical pillar and establishes a foundation for developing a robust and intelligent aerial surveillance system that can detect, classify and track targets in the air.

2.2 Overview of the Study

This chapter addresses the methodology used to create the aerial surveillance and defence system. The reference dataset from previous research contained only four categories of visual data needed to create a precise prediction model, containing birds, aircraft, helicopters and drones, hence data was not ready for training. Subsequently, we developed an entirely new dataset from the beginning, called Diverse Airborne Object Detection Dataset(DAOD). This dataset now includes seven general classes including birds, drones, airplane, helicopters, missiles, military planes and UAVs, all of which have been grabbed from the Roboflow site using a structured method. All of the data was combined to provide a complete dataset that was processed for cleaning and standardisation to produce a quality dataset to be used for the purpose of developing a predictive model.

Once the data preparation phase was completed, the architecture of the system was designed in two main functional areas. The first func-

tional area focused on classifying images by analyzing and comparing three of the best and most modern architecture-based models that are currently available, these being YOLOv26x-cls, EfficientNetV2-L and a Transformer-based model called Swin-Tiny. The second functional area of the architecture dealt with the detection of spatial objects using the YOLO26x model; then, using the same trained weights from the first functional area, a moving track was created for video processing. Finally, all models were subjected to extensive performance evaluations based on an analysis of accuracy, validity and overall usability of the architectures presented in the following architectural flowchart of the models used.

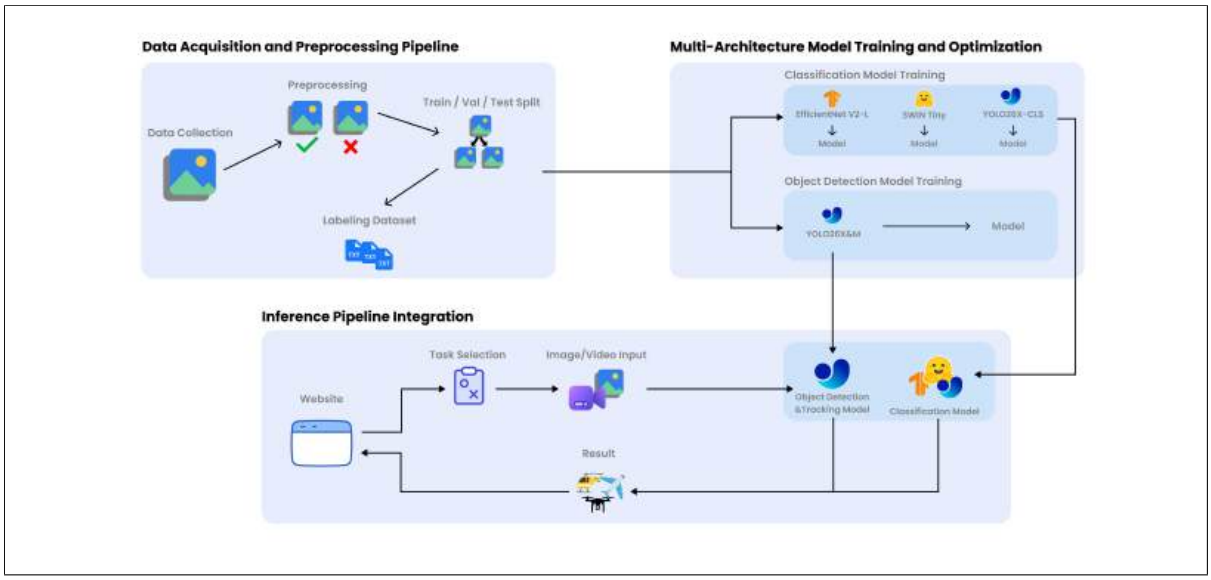


Figure 2.1: Air Defense System with Detection and Classification Pipelines

2.3 Datasets

Multiple datasets were employed in this study to train and assess the proposed system. These datasets are:

2.3.1 Aerial object detection dataset 4 (AOD4)

At the start of this project is based on the use of an open source dataset that is available to the general public from a number of different repositories. The dataset [29] includes a total of 26149 images evenly distributed among four primary groups of objects: birds and UAV, fixed-wing aircraft and rotary-wing (helicopter) aircraft. The uniqueness of this dataset is its vast range of different backgrounds; this dataset contains many different types of environments from open skies, to land, to snow-covered land; and

then again at nighttime with some light source. This difference in environment was fundamental to training a high-quality model that could operate in significantly different levels of weather, atmosphere, and/or operating conditions.

Despite the aforementioned qualities of the dataset used to train the model, evaluations conducted on the model revealed serious issues with regards to the model's ability to generalize; as a result, the dataset has a major bias on objects found at relatively short distances (i.e., objects which are less than about 100 metres away from the observer). As a result, while the model performed exceptionally well with close objects (probability of success $> 99\%$), the model performed poorly with far objects (specifically, small objects that only occupy approximately 1%-10% of the overall image). Therefore, this limitation (formally referred to as the Small Object Detection [SOD] Problem) required the investigation of additional strategies to further develop the performance of the model in long distance intercepting operations.



Figure 2.2: Sample Images Class from the AOD4 Dataset

2.3.2 Why Aerial object detection dataset 4

There are a number of reasons for choosing the AOD4 dataset; they include many technical/legal issues as well as practical issues. The first reason is that AOD4 contains a very large, well-organized, and accurately labeled set of images. Hence, any type of work done on AOD4 should greatly speed up the amount of time required for both data collection and processing.

A second reason for using AOD4 data is that all images in AOD4 have the same image dimensions (640 pixels by 640 pixels); thus, this uniform

image size allows for model training stability.

Finally, since AOD4 is an open source data set, individuals using this data set will not have any proprietary restrictions on their use or access and therefore there will be total freedom of use and availability for the entire length of this research project.

2.3.3 Diverse Airborne Objects Dataset (DAOD)

After conducting some initial testing of the model using the original dataset, it was determined that long-range target classification was not accurately recognized by the model, especially in regard to very small targets that only account for one to ten percent of the overall area of the image. An effort to create a new system capable of highly accurate target classification resulted in the development of a custom dataset referred to as the Diverse Airborne Objects Dataset (DAOD)[1].

DAOD contains a total of over 34,000 images and has been designed with a balanced distribution of classes in order to eliminate class imbalance in the training process. The DAOD has been divided into seven strategic aerial classification categories:

- Birds [2]
- Drones [3]
- Airplane [4][5]
- Helicopters [6]
- Militery Airplanes [7]
- Missiles [8]
- Unmanned Aerial Vehicles (UAVs) [9]

[1] https://www.kaggle.com/datasets/younesaid/diverse-airborne-objects-dataset-6/
[2] https://universe.roboflow.com/uav-detection-fmpp2/birds-6cufw
[3] https://app.roboflow.com/younes-efff5/drones-jvii8-dkddm/
[4] https://universe.roboflow.com/custom-szym2/airplane-yktzg
[5] https://universe.roboflow.com/custom-szym2/airplane-yktzg
[6] https://universe.roboflow.com/helicoptersofdc/helicopters-of-dc-ghwuq
[7] https://universe.roboflow.com/fighter-jets-qvvlh/fighter-jets-n4y3k
[8] https://universe.roboflow.com/qedwdqw/final-missiles
[9] https://universe.roboflow.com/fighter-jets-qvvlh/fighter-jets-n4y3khttps://universe.roboflow.com/iha-byytp/iha-lvj6k



Figure 2.3: Sample Images from the DAOD Dataset

2.3.4 Why Diverse Airborne Objects Dataset

The key reason for the design of the Diverse Airborne Objects Dataset (DAOD) was related to the considerable deficiencies of the previous dataset, namely, its poor efficiency in terms of detecting long-distance objects. Besides the problems in identifying small-scale objects, the main drawback of the previous dataset was associated with its limited number of classes, which negatively affected the overall performance of the proposed air defense system. As the result of an in-depth analysis of all possible open source datasets, it became clear that none of them meets the demands of this study.

Therefore, it was decided to create the DAOD from scratch. The custom dataset included seven categories of objects; at the same time, a variety of background images was considered in order to improve the performance of the algorithm in practice. All the images in the database have been resized into 640×640 pixels to enhance the training stability and avoid any data biases among the seven classes of objects

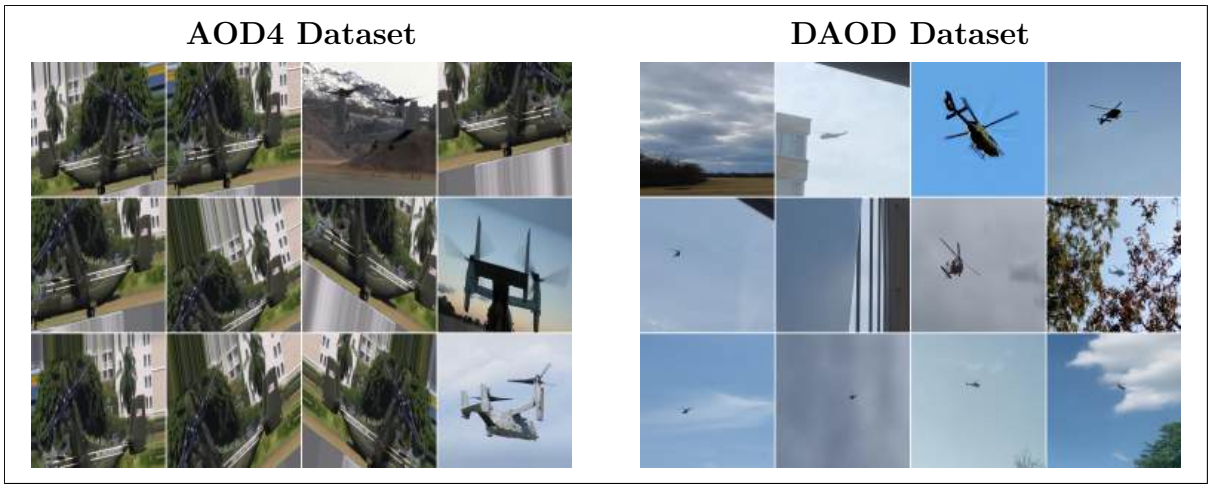


Figure 2.4: Comparison of Helicopter Samples: AOD4 vs. DAOD

Dataset	Number of images	Image Size	Number of Classes	Labels
AOD4	26149	Mixed	4	True
DAOD	34613	640x640	7	True

Table 2.1: Information about the datasets

2.4 Data Preprocessing

The process of preprocessing the data is very important because it provides assurance that the models we will be developing have been created using quality standardised and reliable data. The AOD4 was the initial dataset provided, as well as the DAOD, and a systematic clean, structured and normalised pipeline was developed for each of these datasets to facilitate deep learning usability. Each of the two datasets went through the following two phases of development:

- **Phase 1: Structuring of the standard dataset (AOD4)**

  The original review of the AOD4 dataset determined that it contained four principal classes of objects with each having approximately 6,000 images of the respective object class. However, the AOD4 dataset was in unprogrammed format without partitioning for evaluation. Therefore, all images in the AOD4 dataset were restructured and then separated into three separate datasets (one for each partition) (training, validation, and testing) via programmatic processes.

- **Phase 2: Balancing and processing of the custom dataset (DAOD)**

  The DAOD also was compiled from numerous different sources of data. Similar to the AOD4 dataset, an initial review of the DAOD dataset shows a distinct imbalance in images in each of the seven defined object classes. Therefore, a strict systemic process of data balancing was used to ensure balanced images. Each of the seven object classes was again standardised and set to 4,500 images per class and to a total of 31,500 images for the entire DAOD dataset. The DAOD dataset images were then programmatically partitioned into three new files with images separated into training, validation and testing directories.

2.5 Cleaning Data

The aggregation of dataset resulted in fulfilling the first stage of the data pipeline, which is to begin with data cleaning and data standardization for optimal quality of input. Review of the annotation file's data would show that there are inconsistencies (typos and random assignment) with the class identifiers (Class IDs). These types of inconsistencies typically happen when you have combined different sources of data and integrated them. In order for there to be strict programmatic compatibility with YOLO Model architectures, a Re-indexing was completed against all text files. The classes were remapped from 0-6 sequentially in the following manner: (0: Airplane; 1: Birds; 2: Drones; 3: Helicopter; 4: Military Airplane; 5: Missiles; 6: UAV.)

For the geometrical mis-matches of the images, all of the images were cleaned to be of the same size, using Roboflow, prior to any data refinement activities. The spatial resolution of the overall dataset was also standardized to a treatment size of 640x640 pixels to maintain consistent or homogeneous input into the neural network layers during the model training process.

2.6 Training and Validation Split

The dataset was divided into three sets after it was cleaned and organized in order to facilitate the training and evaluation of models. All of the

data was split into three data sets: 70% of the data was used for training (to learn features and patterns), 15% of the data was used for validation (to tune hyperparameters and monitor training performance), and 15% of the data was left for testing a model's ability to generalize. The dataset was split efficiently and accurately via code Python, which allowed for a reliable automated tool for quickly and consistently splitting data into balanced randomized distributions.

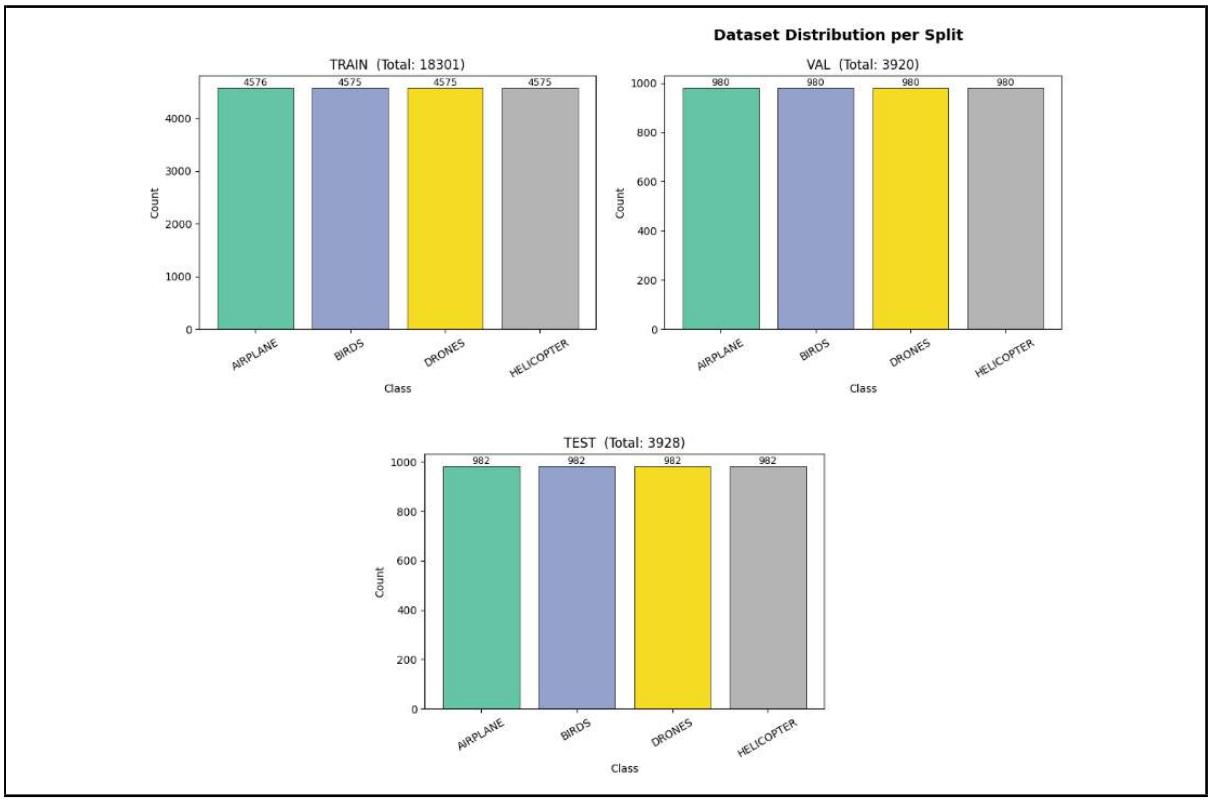


Figure 2.5: The Distribution of AOD4 Dataset

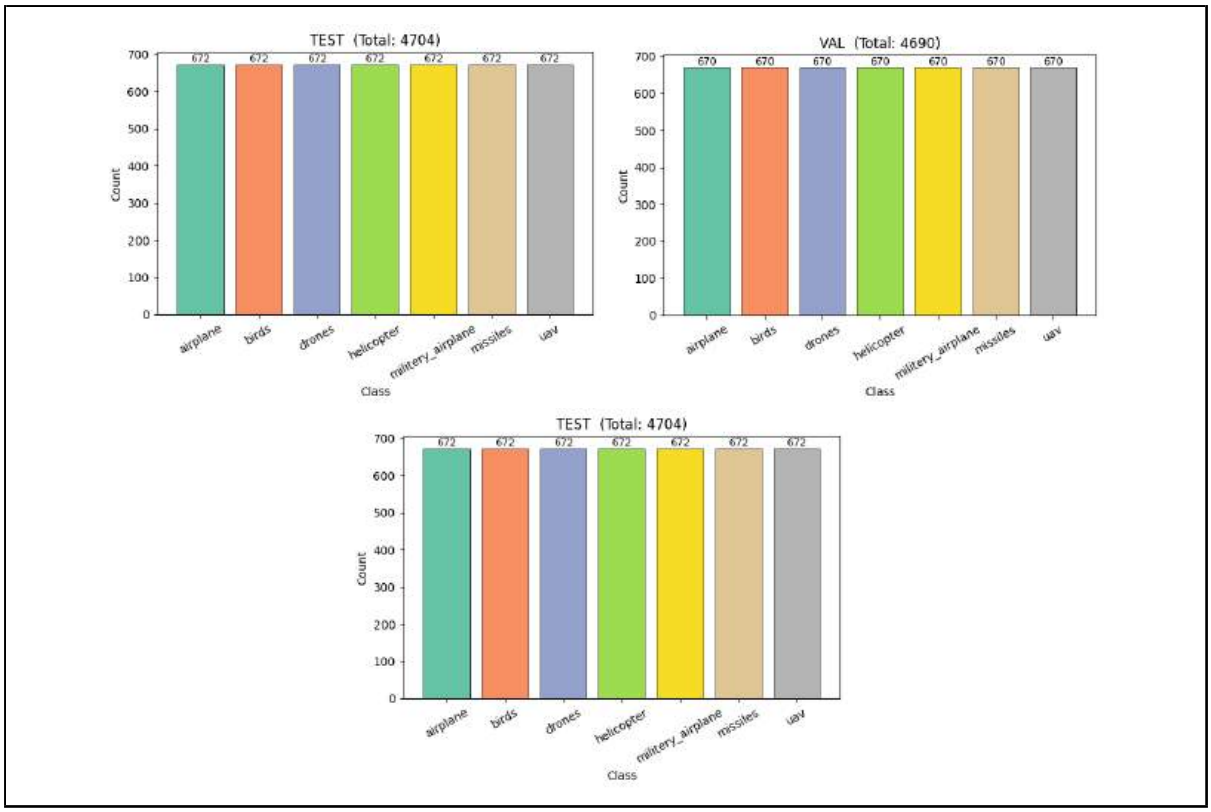


Figure 2.6: The Distribution of DAOD Dataset

2.7 Labeling Dataset

In preparation for baseline image classification tasks, class balance and dataset preparation work has now been completed, and the images are now ready for use. Transitioning from general image classification to object detection means that for each predefined target within an image we will need to know where that object is located on the image. This information is provided in the form of a bounding box around the area where the object exists in that image. For the DAOD dataset, we took care of this task during dataset preparation and already had the relevant labels of the ground-truth in our prepared dataset. For the AOD4 dataset, this required a full annotation for each image where the relevant bounding box would be provided for the target object.

The automated annotation process will reduce errors made during the annotation process; these errors could have a catastrophic effect on how well the model performs with respect to object detection; thus, we have developed an automated pipeline for generating bounding box annotations for the aerial imagery from the AOD4 database. For the purpose of this annotation, we used the advanced YOLOv8x-World Model in conjunction

with the Slicing Aided Hyper Inference (SAHI) algorithm, which together dynamically extract the spatial coordinates and create the standard YOLO (.txt) image format annotation for each aerial target in the AOD4 dataset. By combining the capabilities of both models we will successfully resolve many of the complexity issues associated with detecting micro-objects and provide a fully annotated dataset for model training.

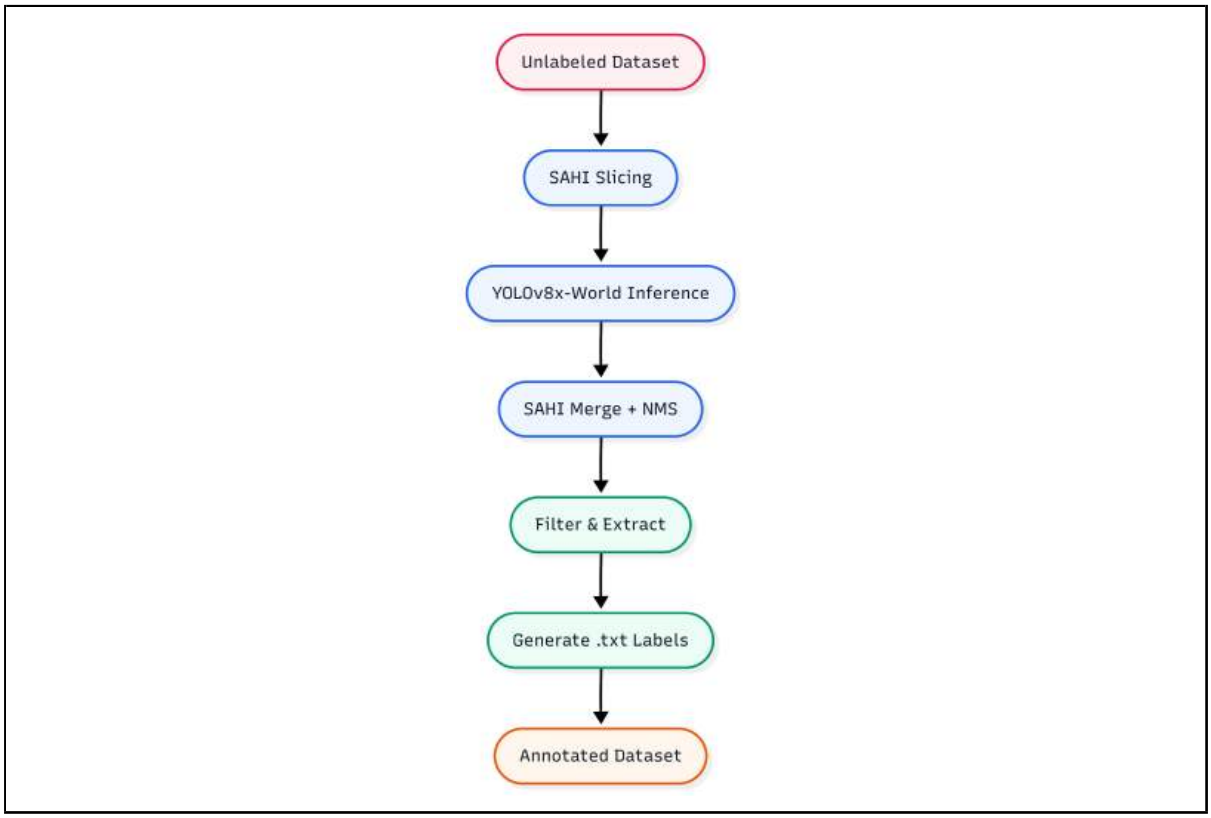


Figure 2.7: Flowchart of the Dataset Annotation Process

2.8 Training and Evaluation

With the data pre-processing stage and all data structure established, training of the models followed as the next major milestone. Selecting appropriate structures is essential because using a structure that does not fit the needs of the data will result in a poor use of computing resources and consequently lower levels of accuracy. An extensive survey of peer-reviewed literature allowed us to separate the procedure for data collection and model development into two major phases:

2.8.1 Image Classification

Image Classification is the foundation of all computer vision applications as well as the first step of understanding a scene. For the comparison of three advanced models representing different architectural paradigms, the goal was to get an accurate and reliable way to identify aerial threats.

- **EfficientNetV2-L model :**

  EfficientNetV2-L is one of the first architectures for Convolutional Neural Networks (CNNs) used in computer vision that utilizes a type of Fused-MBConv layer, resulting in an architecture that achieves high accuracy (or usable quality) and efficiency (or speed), both of which are extremely important when resource constraints exist.

- **Why EfficientNetV2-L model :**

  Based on documented superiority over conventional architectures within recent studies, this particular Model was selected for use in our system. According to the Key Paper[10] at the CVPR 2021 conference, the EfficientNetV2-L model has 5x-11x faster training time and 6.8x fewer parameters compared to previous baseline models (e.g., ResNet, standard ViT), while achieving a state-of-the-art top-1 accuracy of 85.7% on ImageNet. The unique combination of high performance and low computation makes EfficientNetV2-L the ideal engineering solution for our system.

- **YOLO26x-cls model :**

  The YOLO26x-cls model is one of the newest and most powerful models developed for computer vision today. Unlike previous classical versions of YOLO which were designed specifically for object detection and bounding box regression, this new model has been optimised for image classification tasks. The key to the success of this model is that it provides extremely fast inference speeds with very accurate results.

- **Why YOLO26x-cls model :**

  This was the model chosen to assess cutting-edge technology in a real-world application. According to the Ultralytics official Benchmark, YOLO26x-cls can achieve a Top-1 Accuracy of 79.9% [11] as

[10] https://arxiv.org/abs/2104.00298
[11] https://docs.ultralytics.com/tasks/classifymodels

well as nearly instantaneous processing times, on this test, of 1.5 ms per image at the same time it achieves this accuracy.This is the first combination of both of these aspects of accuracy and speed concerning real-time processing of airborne threat classification will enable realtime classification of airborne threats compared to previous conventional architectures.

- **Swin Transformer Model (Tiny Variant) :**

  After the shift in paradigm that transformers brought to Natural Language Processing (NLP), this framework was adapted to work well on computer vision problems; therefore, Vision Transformers (ViTs) were born. The Swin-Transformer is an example of one of the leading architectures in the Vision Transformers domain and it represents a good substitute for traditional Convolutional Neural Networks (CNN). Its use of 'Shifted Window Self-Attention' enables the model to discover complicated spatial hierarchy and relationships in the images being processed.

- **Why Swin Transformer Model (Tiny Variant) :**

  The 'Tiny' (Swin-Tiny) variation of the transformer architecture has been developed to evaluate the potential of transformer technology in resource-limited environments. While the base transformer (Quad Base) cannot be run because of the constraints (GPU memory limitation) of the hardware, the tiny model presents a balance between calculation cost and high efficiency. The tiny model also has a high classification accuracy rating of 81.3% on image classification of the ImageNet dataset[12], thus providing an effective option for classifying aerial targets "efficiently" without requiring a large computational infrastructure.

2.8.2 Object Detection

In today's world, object detection represents one of the most sophisticated and crucial branches of computer vision science. As opposed to regular image classification, which simply determines what elements are found in an image as a whole, object detection is designed to accomplish not one, but two tasks – localization (via bounding boxes), and multi-class classification simultaneously. Naturally, for advanced applications demanding

[12]https://arxiv.org/abs/2103.14030

instant reaction time such as automated tracking and monitoring, object detection is indispensable. For these purposes, we used:

- **YOLO26x:** Ultralytics is a system that consists of a full-functioning software framework that enables the production and support of the most advanced deep learning models available for multiple computer vision applications. Examples include high accuracy image classification and spatial object detection in real time. For these reasons, YOLO26x architecture has been intentionally selected as the core processing engine in our system; this was determined through review of the following scientific basis and rationale
- **Why YOLO26x :**

  In this context, the YOLO26x architecture was selected to represent an object detection model, which is the latest released variant in 2026. The reasons that inspired this option value the selection not only based on the model's technological novelty but mainly in its unparalleled benchmark performance. Official evaluations indicate that, among the model's 'X' variants, the mean average precision achieved on the standard COCO dataset was 57.5-very satisfying accuracy to ensure a needed reliability in engineering towards proper detection and localization of objects in our system.

2.8.3 Sahi with YOLO26x

YOLO demonstrates significant accuracy in object detection generally; however, it has some issues with detecting small objects that are far away on an aerial plane. These problems occur mainly because of image resizing to perform a detection using the convolutional portion of the architecture. Therefore, fine detail about smaller objects is lost during the image resizing pipeline from the convolutional layers.

Therefore, to address and overcome this critical technical limitation, Slicing Aided Hyper Inference (SAHI) was adopted and integrated with the YOLO model architecturally and performatively. The SAHI approach uses a grid of overlapping patches created from the high-resolution input image that can be run through independently from one another rather than the total, initial decompressed image. As a result, the visual characteristics and native pixel resolution of a small object remain intact during the inference process from YOLO, and then the SAHI algorithm merges the independent local patch predictions together to with the original image

by utilizing Non-Maximum Suppression (NMS) to eliminate overlapping bounding boxes. Thus, this approach dramatically improved the YOLO-based detection's ability to accurately detect aerial threats that are far away, while complementary supporting the primary computational architecture of YOLO with an application suited for the defense system.

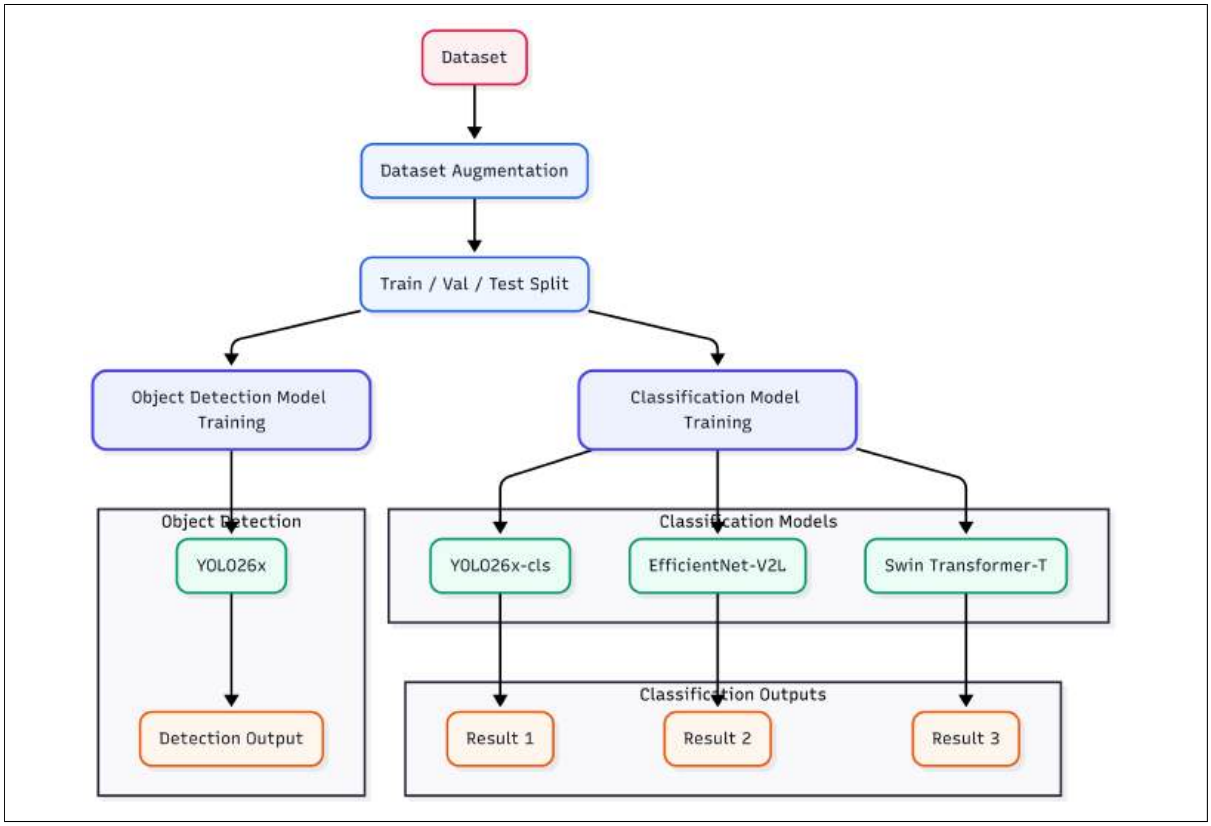


Figure 2.8: Used Models for Training

2.9 Tracking

The successful training and evaluation of the YOLO model have resulted in a set of weights to help develop a real-time aerial tracking system for video imagery. Since the YOLO architecture is very robust, it is unnecessary to retrain the YOLO model from scratch for a stream of visual data. Each frame of the real-time video input will be assessed individually and processed through spatial detection and classification, while also using a variety of advanced tracking algorithms to maintain temporal consistency and continuity anywhere a target is detected across the series of frames. Lastly, the processed frames will be assembled in the correct order and output as a high-definition annotated video (in.avi file format) with targets and target flight paths; clearly identified.

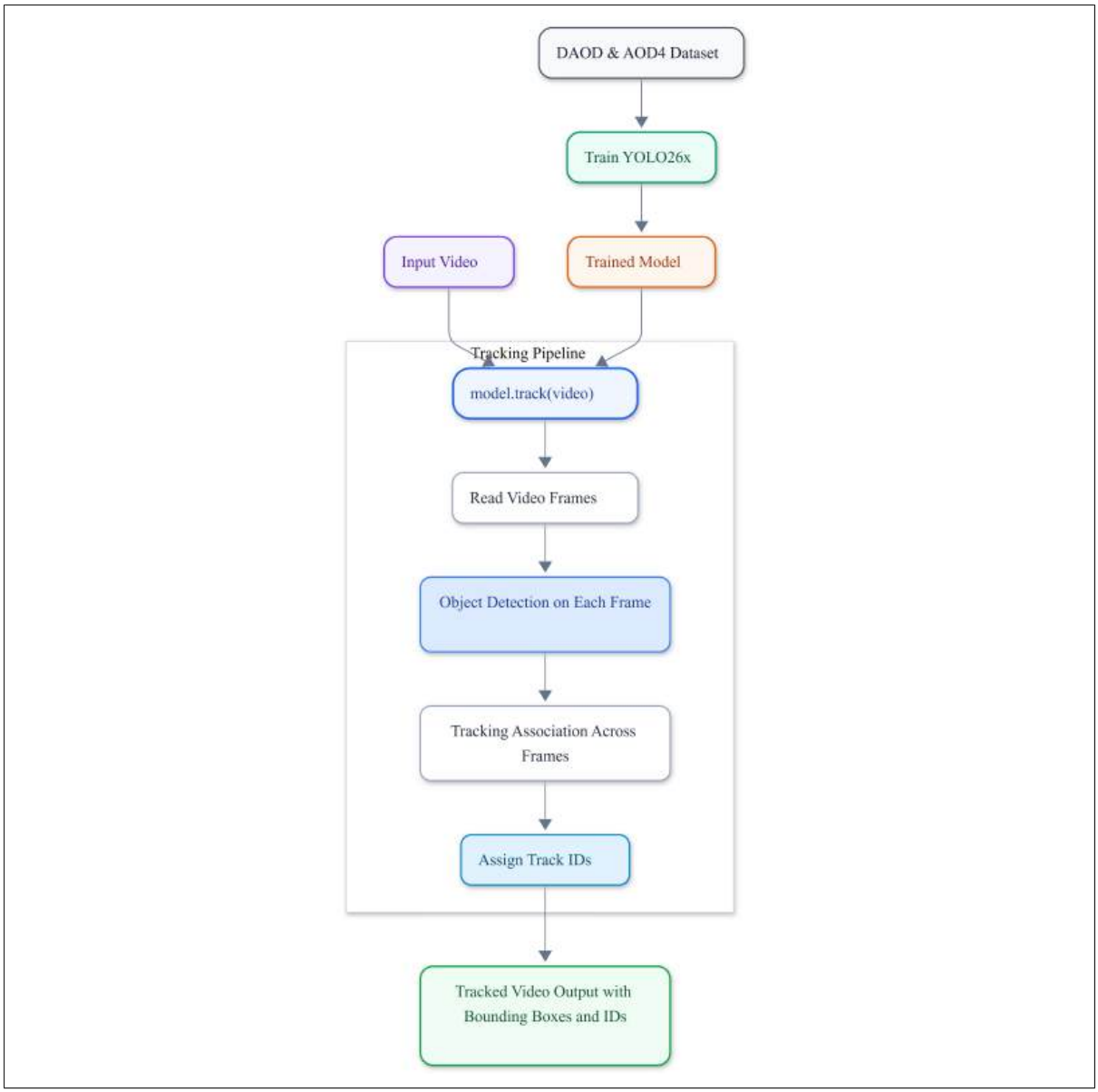


Figure 2.9: Operational Workflow of the Aerial Object Tracking Module

2.10 Training Hyperparameter

The training hyperparameters were carefully tuned at the outset to form an appropriate predictive model that is both accurate and generalizable. Proper tuning of the training variables was essential for controlling overfitting and ensuring stable training. For consistency across models with regards to an evaluation environment that is both standardized and equal, please refer to the following table that represents the primary hyperparameters used in relation to efficiently training the models :

Parameter	EfficientNet V2L	Swin-T	YOLO26x-Cls	YOLOv26m	YOLOv26x
Task	Classification	Classification	Classification	Object Detection	Object Detection
Nb Of Params	118.5M	48.8M	29.6M	55.7M	10M
Epochs	130	45	120	257	149
Batch	8	64	64	16	64
Resolution	384	224	640	640	640
Learning Rate	1e-6	5e-5	5e-5	2e-6	1e-4
Total Training Time	82	12	88	72	90
Hardware (GPU)	T4x2	T4x2	T4x2	RTX 3050	T4x2
Software	PyTorch	PyTorch	Ultralytics	Ultralytics	Ultralytics

Table 2.2: Training Configuration per Model

2.10.1 Justification of Hyperparameter Selection

The hyperparameters that were selected did not happen on a whim or by chance while trialing out the model; They were carefully and systematically adjusted according to the actual results of extensive trials in actual empirical experimentation. The types of computational resources and limitations imposed by the partitioning of the available hardware heavily influenced how and what specifications would be finalized. For example, YOLO was limited to an input image resolution of 640×640 pixels, while Swin-Tiny was limited to a resolution of 224×224 due to the potential for out-of-memory (OOM) errors with the GPU and to fully comply with the corresponding requirements of each architecture. Similarly, the important variable settings, like the learning rate and batch size, were established through numerous sets of heuristic tests until reaching their optimum settings. These various methods of accurately calibrating the hyperparameters provided a balanced consideration of the trade-offs between convergence speed, stability during training, and prediction accuracy.

2.11 Conclusion

An overview was given to demonstrate the standard engineering methods that were employed to develop a complete, integrated system to classify and detect airborne targets. The workflow began with a combined dataset that used the standard AOD4 dataset and a custom made dataset called DAOD, using seven unique classes of targets. The first step was to create a high quality input to the system; therefore, the two datasets were preprocessed. This included the application of strict class balancing to prevent bias in the system and extensive data purging to remove contamination between classes (e.g., removing images of misclassified helicopters from the Airplane class).

After preparing the dataset for the actual object detection work, an advanced auto-labeling pipeline was built using the YOLOv8x-World model and combined with Slicing Aided Hyper Inference (SAHI). This integration assured maximum accuracy of the spatial location of micro-targets, thus providing a fully annotated and valid dataset.

From an algorithmic perspective, the curriculum development occurred through two primary distinct algorithms. The first algorithm that focused on the target classification used a comparative analysis of three cutting-edge architectures: EfficientNetV2-L, which represented the apex of convolutional neural networks (CNNs); YOLO-cls, which was the apex of state-of-the-art real-time classification; and the Swin-Tiny transformer to investigate the performance of Vision Transformers (ViTs) as an emergent class of architectures that have transformed the application of natural language processing(NLP) to complicated visual tasks. The second algorithm focused on object detection and localization and utilized the most current iteration of YOLO as the framework for measuring the performance of the above-mentioned three architectures. The results from this phase were used as a basis for experimentation and performance analysis discussed in the following chapter.

Chapter 3: Results & Discussion

Chapter 3

Results & Discussion

3.1 Introduction

This chapter presents a comprehensive analysis of the experimental results obtained from the development and testing of an integrated AI-powered air defense system. The research focuses on the dual objectives of real-time threat detection and automated neutralization strategies. It begins by outlining the development environment, including hardware and software specifications

3.2 Development Environment

To support the implementation of the proposed models, we begin by outlining the development environment in which the experiments were conducted.

3.2.1 Hardware specifications

In order to implement our solution, we used:

- A laptop computer equipped with an Intel i5-8665U processor, operating at 1.90 GHz (up to 2.11 GHz), 16 GB of RAM, 512 GB Samsung SSD and running on Windows 11 Pro as an operating system.
- A laptop computer equipped with an Intel i7-85600U processor, operating at 2.80 GHz (up to 2.90 GHz), 16 GB of RAM, a 512 GB TeamGroup SSD and on Windows 11 Pro as an operating system.
- A laptop computer equipped with an Intel i5-12500H processor, operating at 2.80 GHz (up to 2.90 GHz), 16 GB of RAM, with a GPU RTX3060 and 8GB VRam .and on Windows 11 Pro as an operating system.

3.2.2 Software Environment

In order to implement our solution, we used:

a) Kaggle Kernels (Kaggle Notebooks)

Kaggle Notebooks (formerly "kernels") are a central medium for doing, sharing, and studying data science and machine learning work. Research uses large corpora of Kaggle notebooks both to understand real-world practice and to build tools that improve quality, collaboration, and reproducibility.

Role and Characteristics of Kaggle Notebooks

- Kaggle provides a cloud-based notebook environment where notebooks are called "kernels" [30]
- Millions of public notebooks exist; the Meta Kaggle Code dataset includes 5.9M notebooks from 2015 onward [31]
- High-voted Kaggle notebooks tend to have many cells, with a roughly equal number of code and markdown cells and far richer documentation than typical GitHub notebooks [32]
- it provide a powerfull hardware including NVIDIA Tesla P100 or T4 GPUs and TPU v3-8 that we used to train our models

b) Visual Studio Code

Visual Studio Code (VSCode) is a source code editor and integrated development environment (IDE) developed by Microsoft. It is open-source and cross-platform, meaning it runs on Windows, Linux, and Mac. It was designed for web developers but supports many other programming languages such as C++, C, Python, Java, etc. Its flexibility and ease of customization make it a popular choice among developers for writing, editing, and testing code across various platforms.

3.2.3 Libraries

- **PyTorch:** PyTorch is open-source Python library which supports deep learning applications. It has an expansive scope and is applied for various applications. As Python is for programming, PyTorch is both a magnificent prologue to profound learning just as an instrument usable in proficient real-world applications[33].

- **Ultralytics:** Ultralytics-maintained YOLO releases illustrate a steady trajectory toward modularity, task unification, and deployment efficiency, culminating in YOLO26 as the first fully integrated framework for detection, segmentation, pose estimation, oriented bounding boxes, and classification[34].
- **Matplotlib:** Python graphing and plotting library that generates many types of charts in many output formats [35]widely used in data science, engineering, and research. It underpins many higher-level plotting tools and is considered a standard for publication quality figures.
- **Hugging Face:** is a Git-based online code hosting platform aimed at providing a hosting site for all kinds of ML artifacts showcase models[36].
- **OpenCV:** is a powerful, open-source toolkit designed for handling real-time computer vision challenges and advanced image processing tasks[37]

3.3 to Threat Classification Performance Results

The results of Accuracy and Macro F1 for the three models on both datasets are shown in Figure 3.1. All models achieve near-perfect scores on AOD4, ranging from 0.998 to 0.999. Despite DAOD's greater complexity (7 classes vs. 4), performance remains consistently high, with scores dropping by only 0.003–0.004, indicating that all three architectures generalize well across varying class complexities.

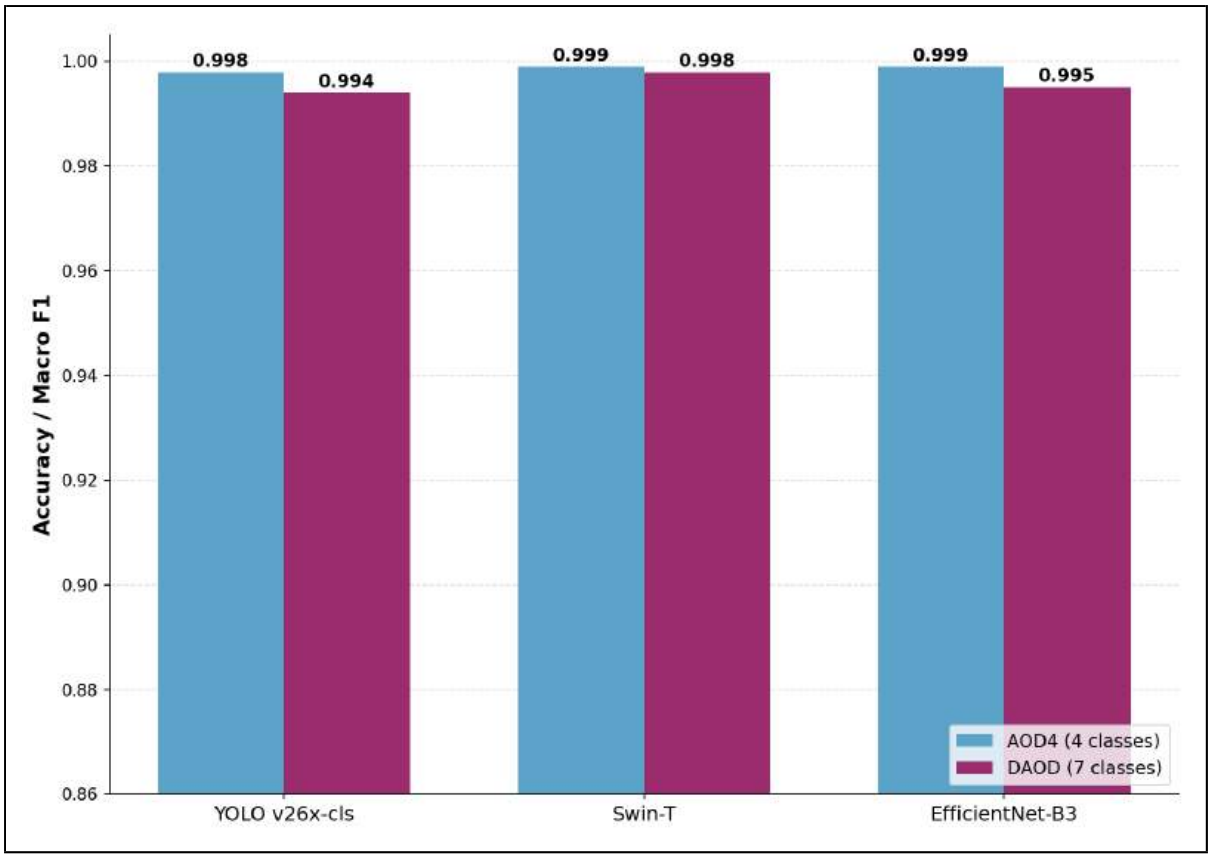


Figure 3.1: Comparative Classification Accuracy and Macro F1 Scores

Figure **??** presents the object detection performance of YOLO26x, measured by mAP@0.5, across two datasets: AOD4 (4 classes) and DAOD (7 classes). On the simpler AOD4 dataset, YOLO26x achieved an mAP@0.5 of 0.981. More notably, when tested on the significantly more challenging DAOD dataset — which nearly doubles the number of target classes — YOLO26x still achieved a strong mAP@0.5 of 0.931. This result is particularly meaningful: it shows that the model does not simply memorize a small set of target types, but genuinely scales to more diverse and complex aerial environments. Taken together, these results confirm that YOLO26x maintains high localization and classification accuracy as target diversity increases — a critical requirement for any real-world air defense application. All values are expressed as proportions in the range [0, 1], where higher bars indicate better detection performance.

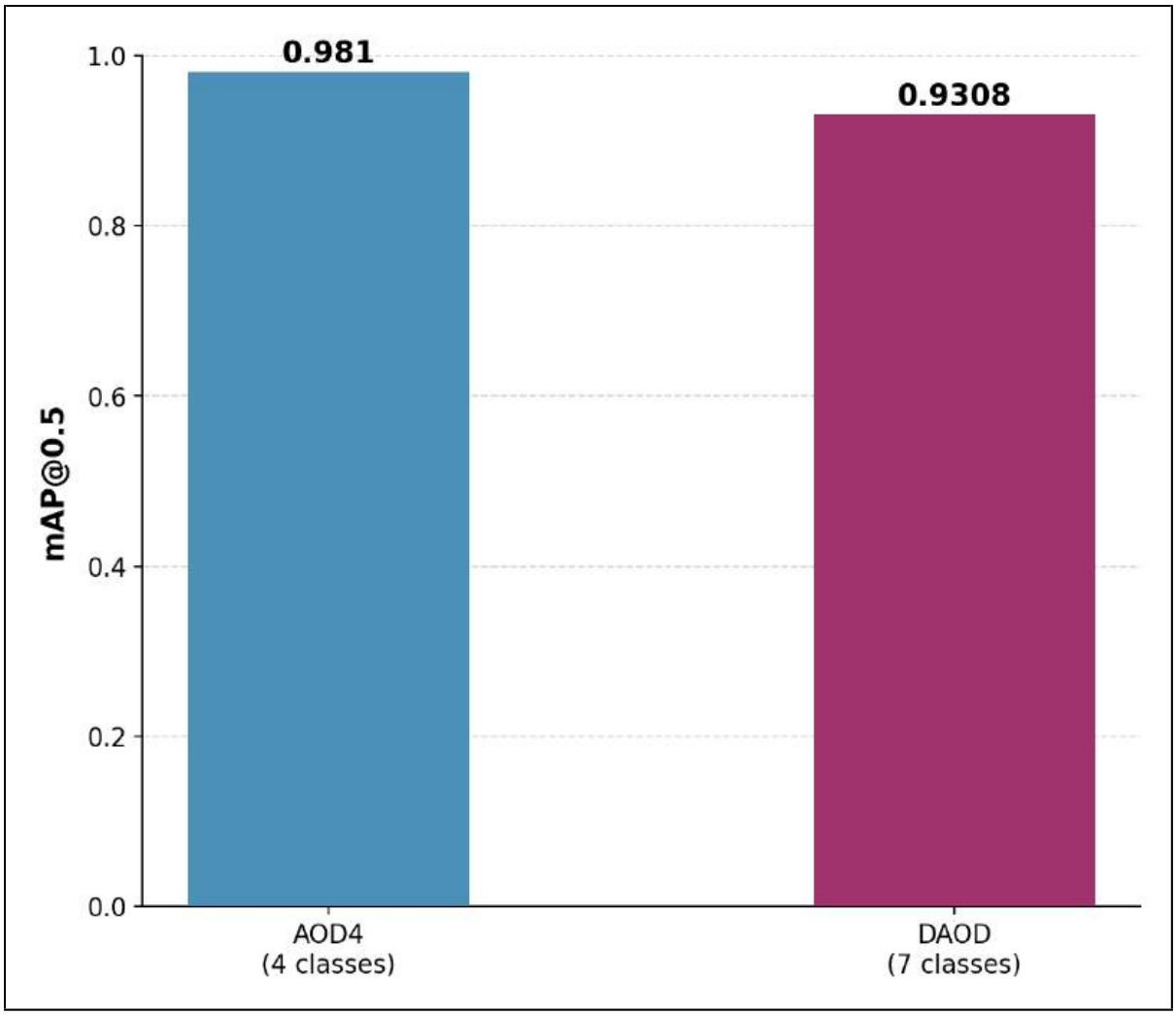


Figure 3.2: Object Detection Performance Comparison between the AOD4 and DAOD

3.4 Per-Class Threat classification Performance Results

The following section provides a comparison of model performance across datasets classes using Precision, Recall, F1-Score.

3.4.1 DAOD Dataset Performance

1. **YOLO Achieves Exceptional Precision Across Complex Aerial Classes**

Class	Precision	Recall	F1-Score
Airplane	1.0000	0.9985	0.9993
Birds	0.9911	0.9911	0.9911
Drones	0.9970	0.9985	0.9978
Helicopter	0.9985	1.0000	0.9993
Military Airplane	1.0000	0.9747	0.9872
Missiles	0.9824	0.9970	0.9897
UAV	0.9897	0.9985	0.9941

Table 3.1: YOLO Class-Level Performance on DAOD

YOLO has achieved very high levels of precision across all DAOD datasets; it received a perfect score (100%) for both the aircraft classified as "standard" and the aircraft classified as "military." A strong F1 score (above 0.987) across all seven of the data's categories indicates that YOLO is still able to effectively reduce the number of false positives when there are multiple high-priority targets (e.g., missiles) in the data (0.9824 precision, 0.9970 recall). YOLO has also achieved an overall accuracy of 99.40%. The unified and single-stage design of YOLO allows it to frame object classifications in one stage as a single regression problem to be solved. YOLO uses the entire image context when extracting features globally so that it can dynamically separate rapidly moving or overlapping targets; this allows YOLO to output highly accurate classifications in time to detect threats early on.

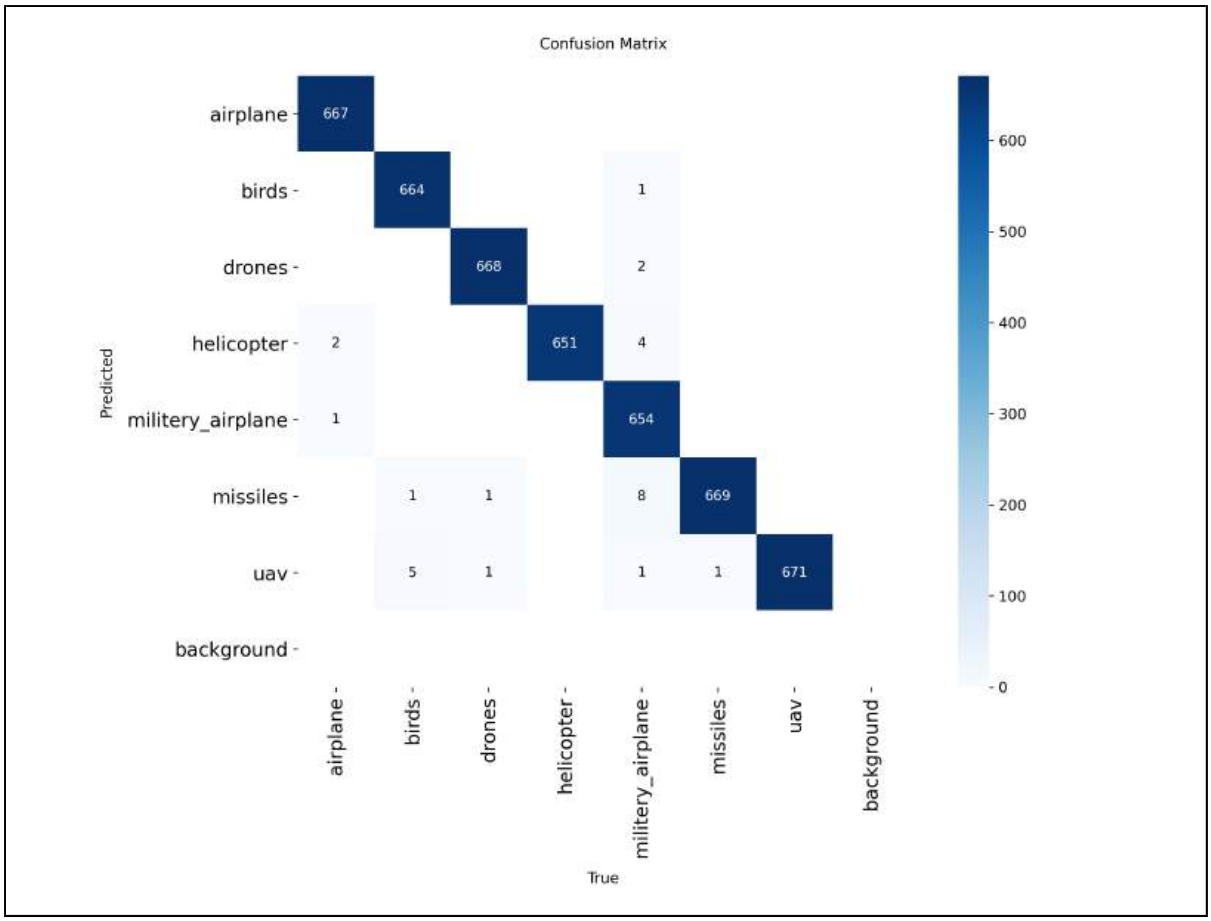


Figure 3.3: Confusion Matrix for YOLO Model on DAOD dataset

2. **Swin Transformer Exhibits Balanced and Highly Precise Class Modeling on DAOD**

Class	Precision	Recall	F1-Score
Airplane	0.9985	1.0000	1.0000
Birds	0.9970	0.9911	0.9926
Drones	0.9926	0.9955	0.9941
Helicopter	0.9970	0.9911	0.9940
Military Airplane	0.9634	0.9926	0.9778
Missiles	0.9985	0.9881	0.9933
UAV	0.9970	0.9851	0.9910

Table 3.2: Swin Transformer Class-Level Performance on DAOD

The Swin Transformer achieved F1-scores above 0.991 across nearly all threat categories on the challenging seven class DAOD dataset, with a perfect score for standard aircraft. Its weakest result was Military Aircraft Precision at 0.9634; however, the corresponding Recall of 0.9926 indicates that the model rarely misses actual military

aircraft it simply applies slightly broader detection boundaries for that class. Overall, the model reached a best validation accuracy of 99.19%. These strong results are largely attributable to the Swin Transformer's shifted window self attention mechanism, which constructs hierarchical feature maps at multiple spatial scales. This allows the network to simultaneously capture fine-grained local details such as rotor configurations or missile fuselage geometry and broader contextual information, giving it a natural advantage when discriminating between structurally diverse aerial targets.

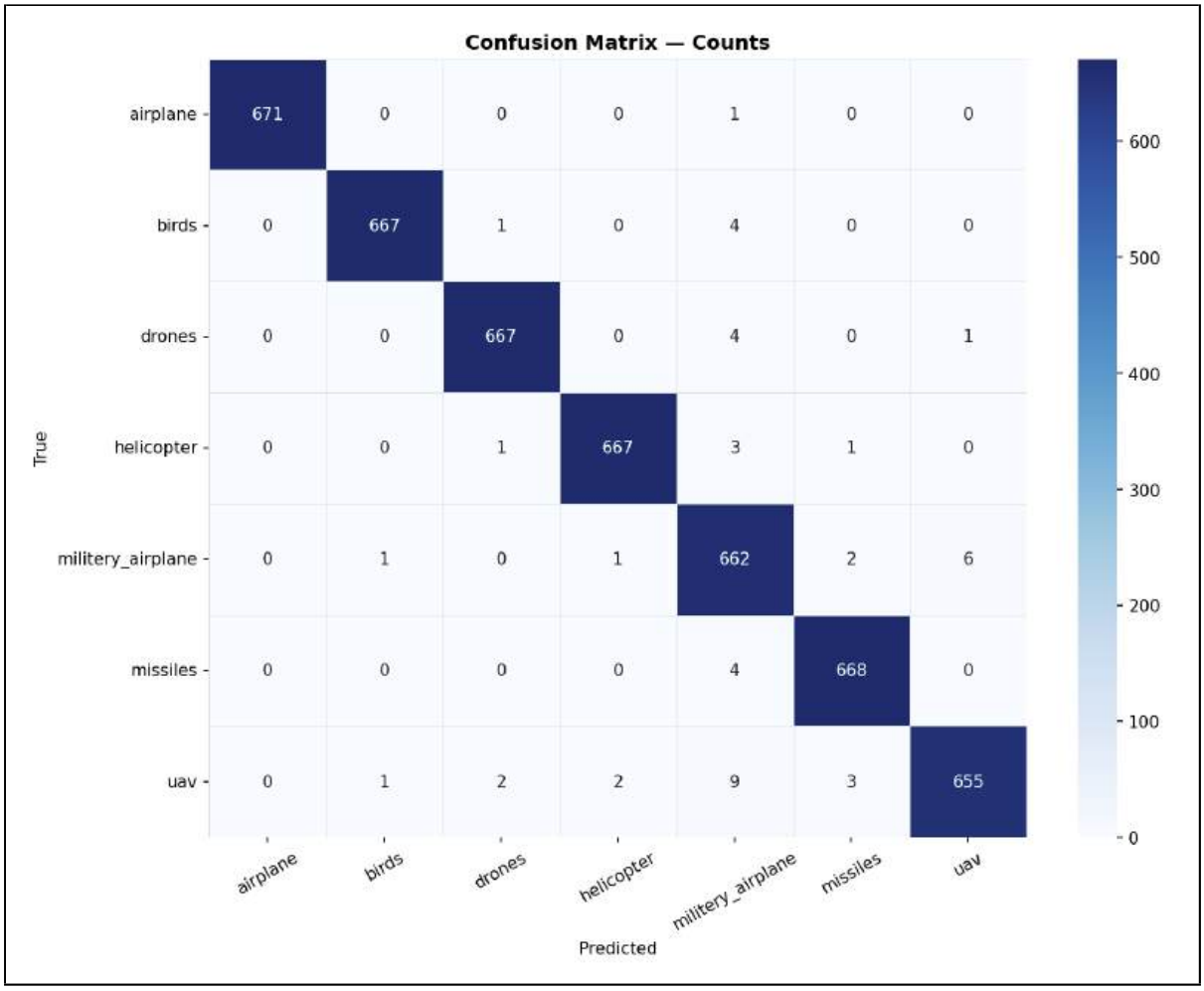


Figure 3.4: Confusion Matrix for Swin Transformer Model on DAOD dataset

3. **EfficientNet Demonstrates Strong Baseline Classification on DAOD**

EfficientNet has performed extremely well as a baseline classifier for the 7-class library of DAOD, with an F1 score surpassing 0.99 for nearly all categories. The model has performed especially well in

Class	Precision	Recall	F1-Score
Airplane	1.0000	0.9985	0.9993
Birds	0.9985	0.9896	0.9940
Drones	0.9853	0.9970	0.9911
Helicopter	0.9955	0.9851	0.9903
Military Airplane	0.9471	0.9851	0.9657
Missiles	0.9985	0.9911	0.9948
UAV	0.9878	0.9643	0.9759

Table 3.3: EfficientNet Class-Level Performance on DAOD

identifying airplanes, missiles, and drones; however, there was a drop in the precision of military aircraft (0.9471) and UAVs (0.9878). With an overall performance level of 98.72% accuracy, EfficientNet's success can be attributed to its use of a compound scaling methodology to uniformly scale the width, depth, and resolution of the networks. By scaling these three dimensions, EfficientNet effectively captures fine-grained visual characteristics, enabling it to classify analogous aerial objects as either aircraft or UAVs based on their structure regardless of their size or resolution in a defense observation scenario.

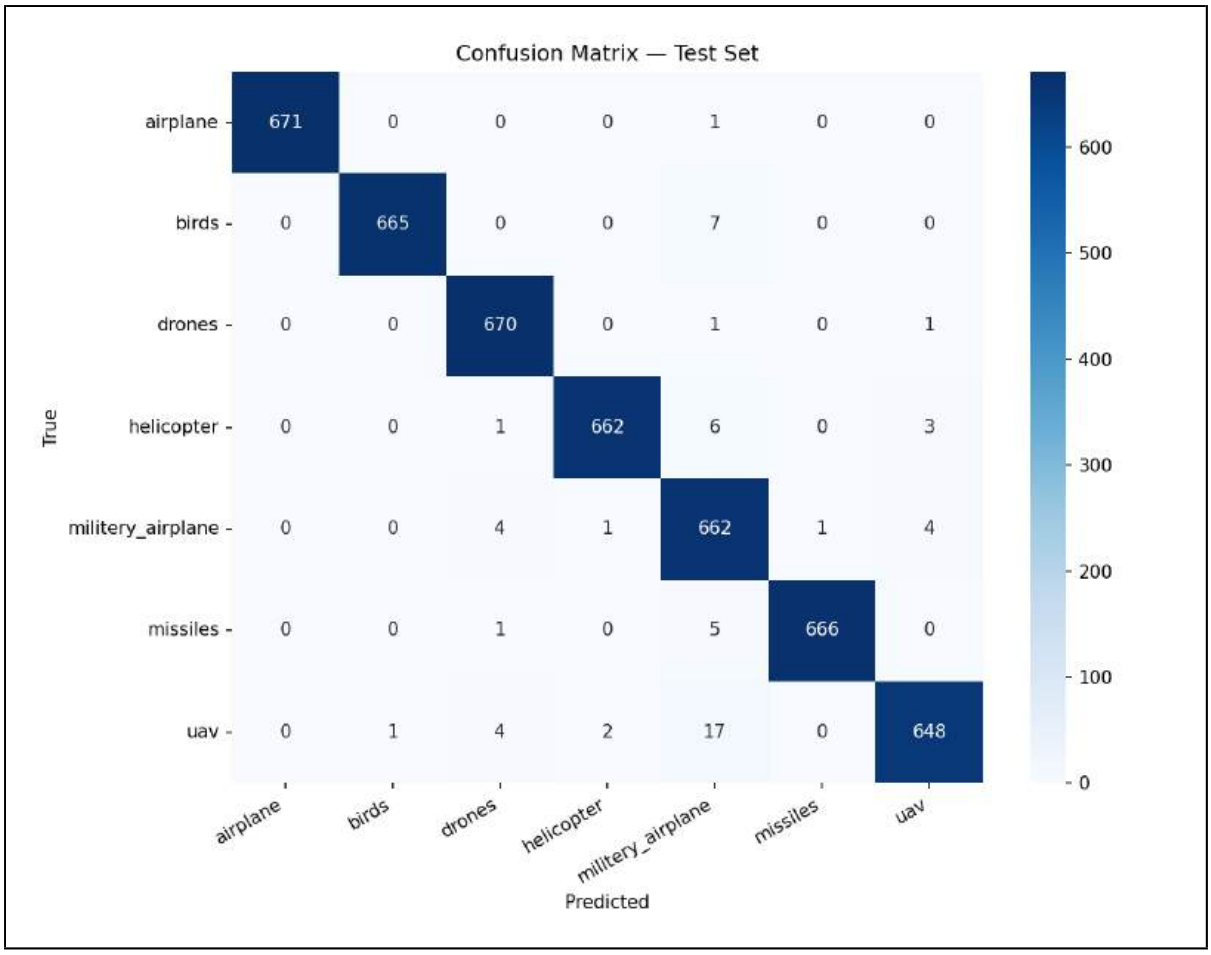


Figure 3.5: Confusion Matrix for EfficientNet on DAOD dataset

3.4.2 AOD4 Dataset Performance

1. **Swin Transformer Yields Flawless Classification on AOD4**

Class	Precision	Recall	F1-Score
Airplane	1.0000	1.0000	1.0000
Birds	0.9970	1.0000	0.9985
Drones	1.0000	0.9969	0.9985
Helicopter	1.0000	1.0000	1.0000

Table 3.4: Swin Transformer Class-Level Performance on AOD4

On the four class AOD4 dataset, the Swin Transformer achieved near perfect scores across all categories, attaining precision, recall, and F1 values of 100% for airplanes, birds, drones, and helicopters resulting in a best validation accuracy of 100% with zero misclassifications. This flawless performance can be attributed to the model's hierarchical structure combined with its shifted window self-attention mechanism. Rather than processing the entire image at once, the Swin

Transformer computes attention within localized windows and progressively shifts them across layers, allowing it to build a rich multi-scale understanding of each target. In practice, this means the model can simultaneously pick up on low-level discriminative details such as the rotor geometry that distinguishes a drone from a bird while also maintaining awareness of the broader spatial context. On the relatively straightforward four class AOD4 dataset, this combination proved sufficient to achieve a completely error free classification result.

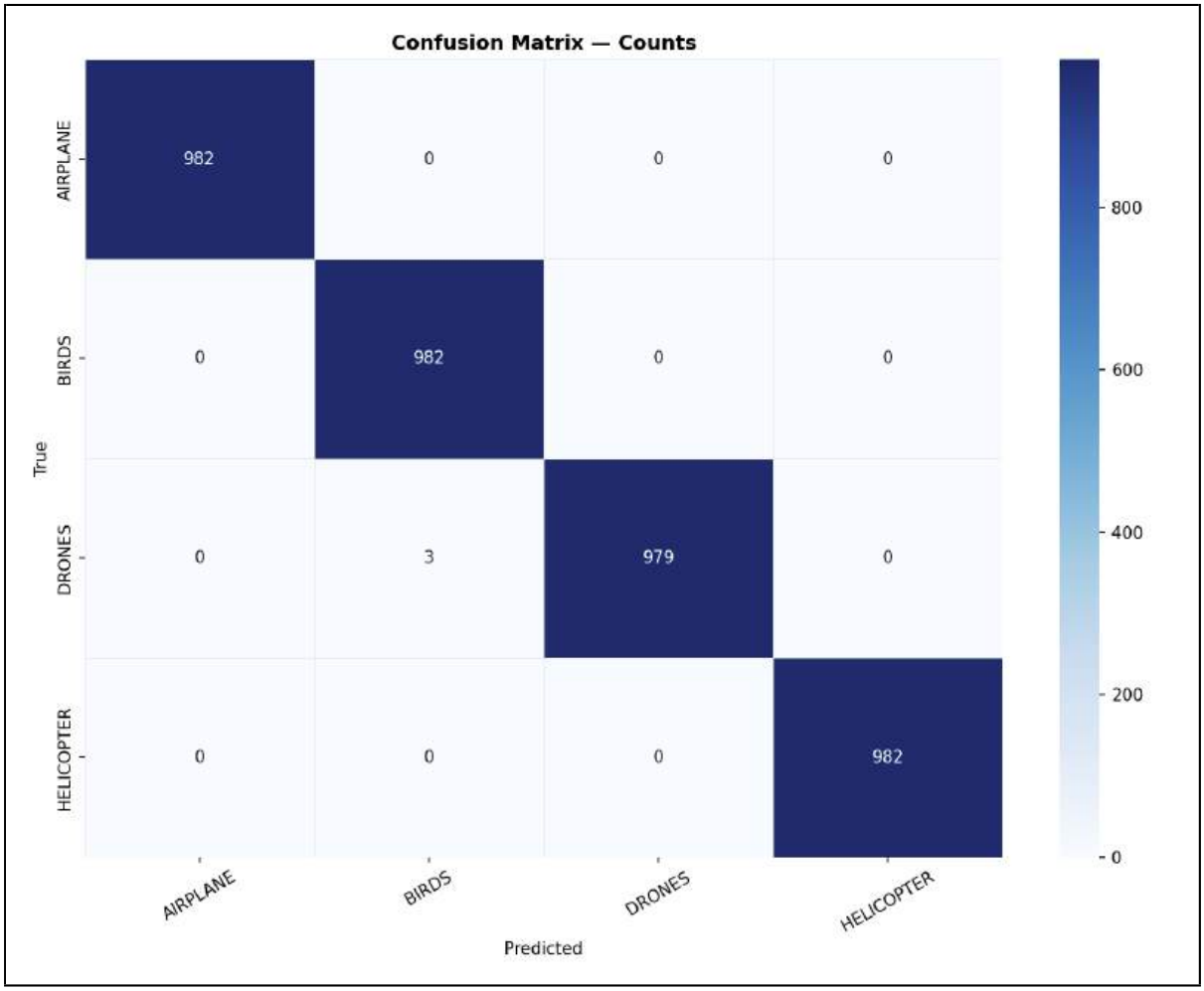


Figure 3.6: Confusion Matrix for Swin Transformer Model on AOD4 dataset

2. **EfficientNet Maintains High Average Precision on AOD4 Sub-Classes**

EfficientNet has nearly faultless classification performance - it has achieved an F1-Score of 0.9980 and above across every target type in the four-class AOD4 dataset. Additionally, with perfect recall of

Class	Precision	Recall	F1-Score
Airplane	0.9990	1.0000	0.9995
Birds	0.9980	0.9990	0.9985
Drones	0.9980	0.9980	0.9980
Helicopter	1.0000	0.9980	0.9990

Table 3.5: EfficientNet Class-Level Performance on AOD4

100% for airplanes and perfect precision of 100% for helicopters, there is essentially no chance that the model will miss detecting an airplane and equally no chance that it will erroneously detect a helicopter. Additionally, EfficientNet has also performed extremely well on the drone class, which is typically difficult to classify - all metrics measured for this class have an F1-Score of at least 0.9980. EfficientNet's outstanding classification results can largely be attributed to the use of compound scaling along with the use of mobile inverted bottleneck (MB-LN) convolutional layers that effectively optimize the depth and resolution of the network to completely isolate each of these aerial target classes versus their corresponding backgrounds.

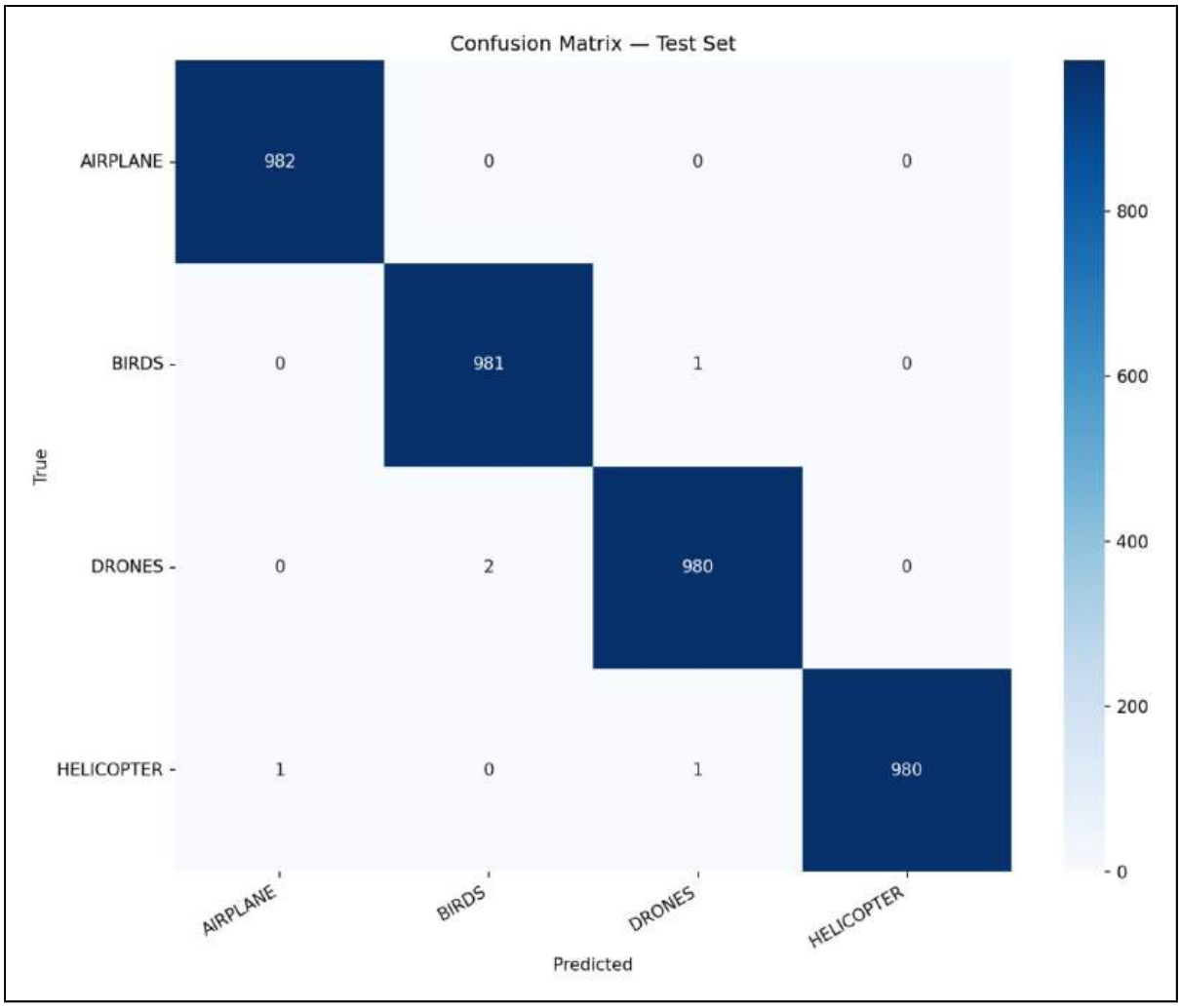


Figure 3.7: Confusion Matrix for EfficientNet Model on AOD4 dataset

3. YOLO Consistently Delivers Near-Perfect Metrics on AOD4

Class	Precision	Recall	F1-Score
Airplane	1.0000	1.0000	1.0000
Birds	0.9959	0.9990	0.9975
Drones	0.9980	0.9959	0.9969
Helicopter	1.0000	0.9990	0.9995

Table 3.6: YOLO Class-Level Performance on AOD4

For the AOD4 dataset, YOLO reports almost perfect classification accuracy: it achieves a precision of 100% for airplanes and for helicopters, and the corresponding F1-scores for the more difficult-to-classify drones and birds are all greater than 0.996. This performance, which translates to a total accuracy of 99.85% over the dataset, can be attributed to the cross-stage partial network (CSPNet) used as

YOLO's backbone as well as to the improved path aggregation in the layers. The rich spatial information maintained through the layers is effectively combined at different scales to accurately classify all of the four considered targets — airplanes, helicopters, drones, and birds — for enabling the automated monitoring systems to accurately and reliably target their objectives.

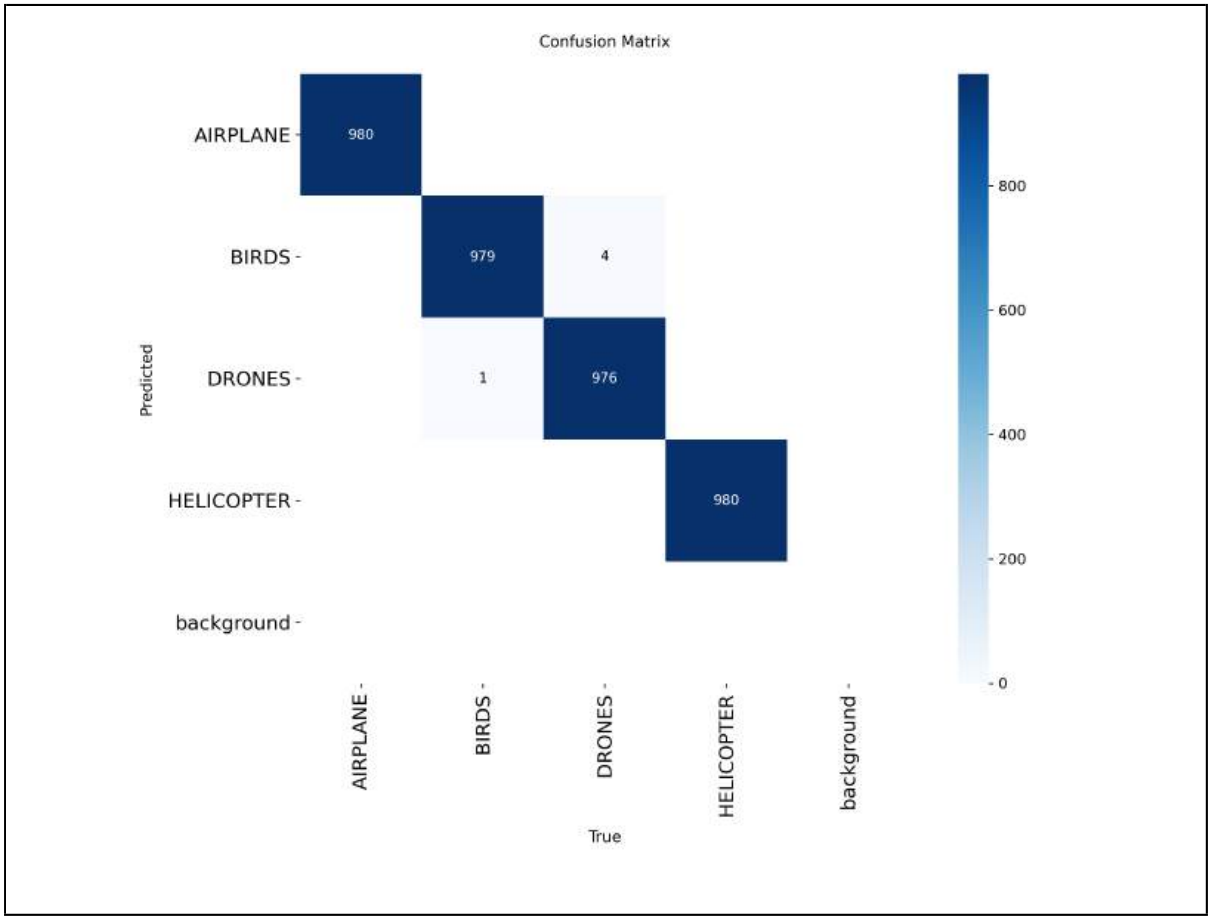


Figure 3.8: Confusion Matrix for Clasification YOLOv26x Model on AOD4 dataset

3.5 Object Detection Performance Results

The following section provides a comparison of object detection performance across dataset classes, evaluated using Average Precision (AP) and mean Average Precision (mAP@0.5).

1. **AOD4 Performance In Object Detection**

   The following section details the object detection capabilities of the framework when deployed on the focused, four-class AOD4 dataset. Performance is evaluated using Box Precision (P), Recall (R), and mean Average Precision across standard.

Model Architecture	Precision (P)	Recall (R)	mAP@50	mAP@50-95
YOLO26x	0.998	0.998	0.981	0.845

Table 3.7: Object Detection Performance Metrics on the AOD4 Dataset

Evaluating the extra-large YOLO26x architecture on the four-class AOD4 dataset provided very promising results. Extra-large YOLO26x architecture even manages to provide very accurate localization and classification. The provided baseline results are even close to perfect. The Box Precision and Box Recall even reached 0.998. Also the more stable mAP@50 of 0.981 as well as the strict mAP@50-95 of 0.845 are performing very well. This suggests that the extra-large YOLO26x architecture in combination with the used feature pyramid network is very well optimized for the used visual environment. The architecture can rapidly and accurately generate bounding boxes around objects around in the aerial enviroment, including organic and mechanical objects. The spatial error is on a very low level.

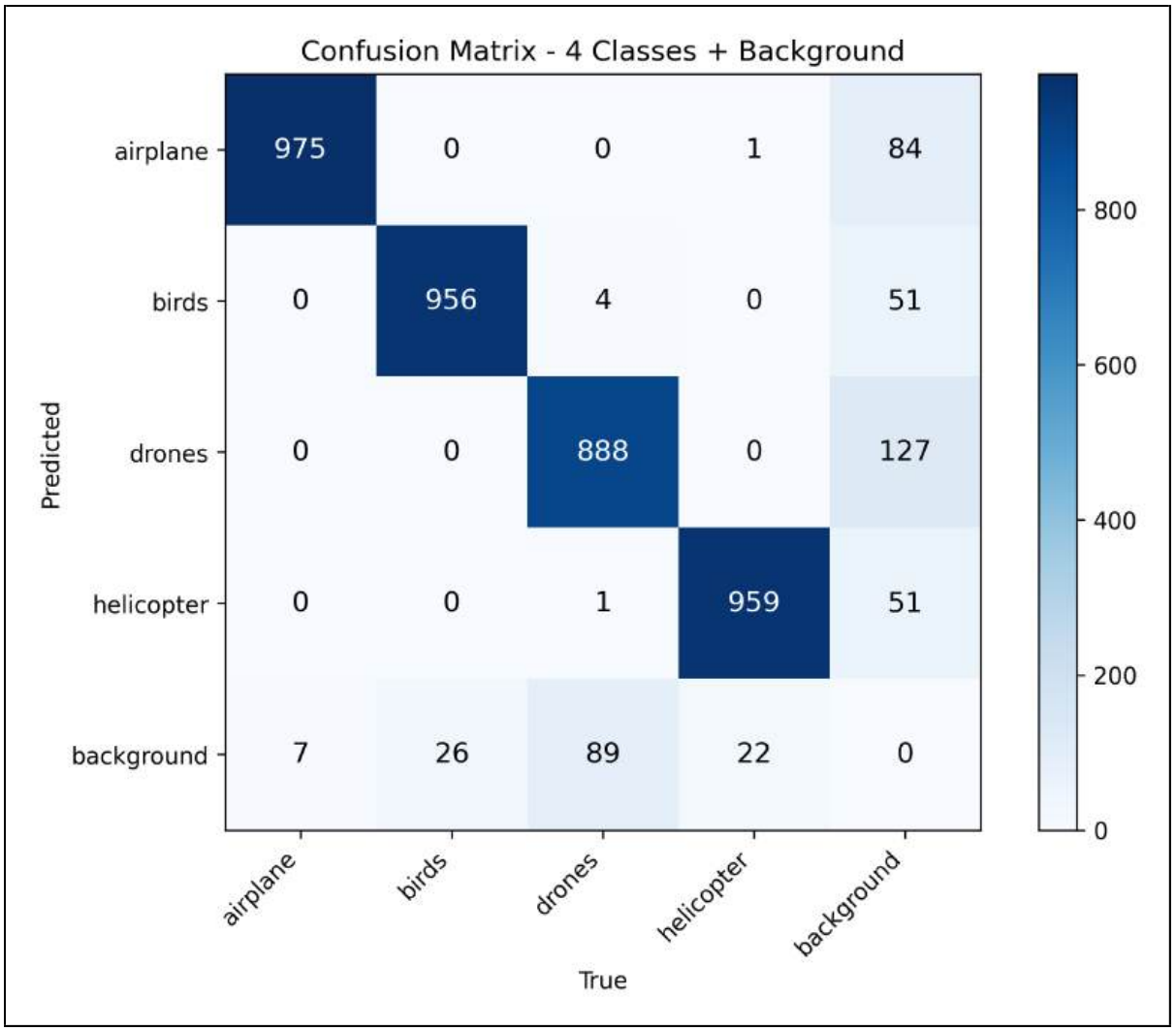


Figure 3.9: Confusion Matrix for Object Detection YOLOv26x on AOD4 dataset

2. DAOD Performance In Object Detection

Model Architecture	Precision (P)	Recall (R)	mAP@0.5	mAP@0.5:0.95
YOLO26m	0.931	0.896	0.931	0.692
YOLO26x	0.994	0.994	0.771	0.548

Table 3.8: Object Detection Performance Metrics on the DAOD Dataset

While YOLO26x hits an incredible high for the feature extraction and verification of large objects in great detail, it reaches a 'ceiling' in terms of global Box Precision and Recall. The respective values of 0.994 for global Box Precision and Recall, attest to YOLO26x being able to produce very few false detection (i.e. no detection of non-objects) and, above all, to achieve almost 100 % detection rates for

easily localizable objects, i.e. for obvious, large, geometrically simple objects like standard airplanes as well as for UAVs.

On the other hand, the medium architecture YOLO26m shows by far the best results in terms of Average Precision (mAP). While the extra-large architecture of YOLO26x yields a perfect Box Precision and Recall of 0.994 (meaning it only very rarely fails to detect the most obvious objects such as standard airplanes or even drones in favorable views), the medium sized variant achieves an mAP@0.5 of 0.931, and an mAP@0.5:0.95 of 0.692. The huge amount of parameters of YOLO26x obviously leads to the best results for the most obvious targets. However, the amount of structural elements of the medium-sized architecture is much lower than that of the extra-large version. As a consequence, YOLO26m is not prone to overfitting as much as the other version is on complex background features, leading to very robust results for bounding box regression and for class balancing. Also, the network is way more stable on low resolution images, on images with many artifacts and also on many other challenging cases, which is the most important requirement for a system that shall provide its user with a complete awareness of the situation. For this kind of task, YOLO26m is clearly the better choice.

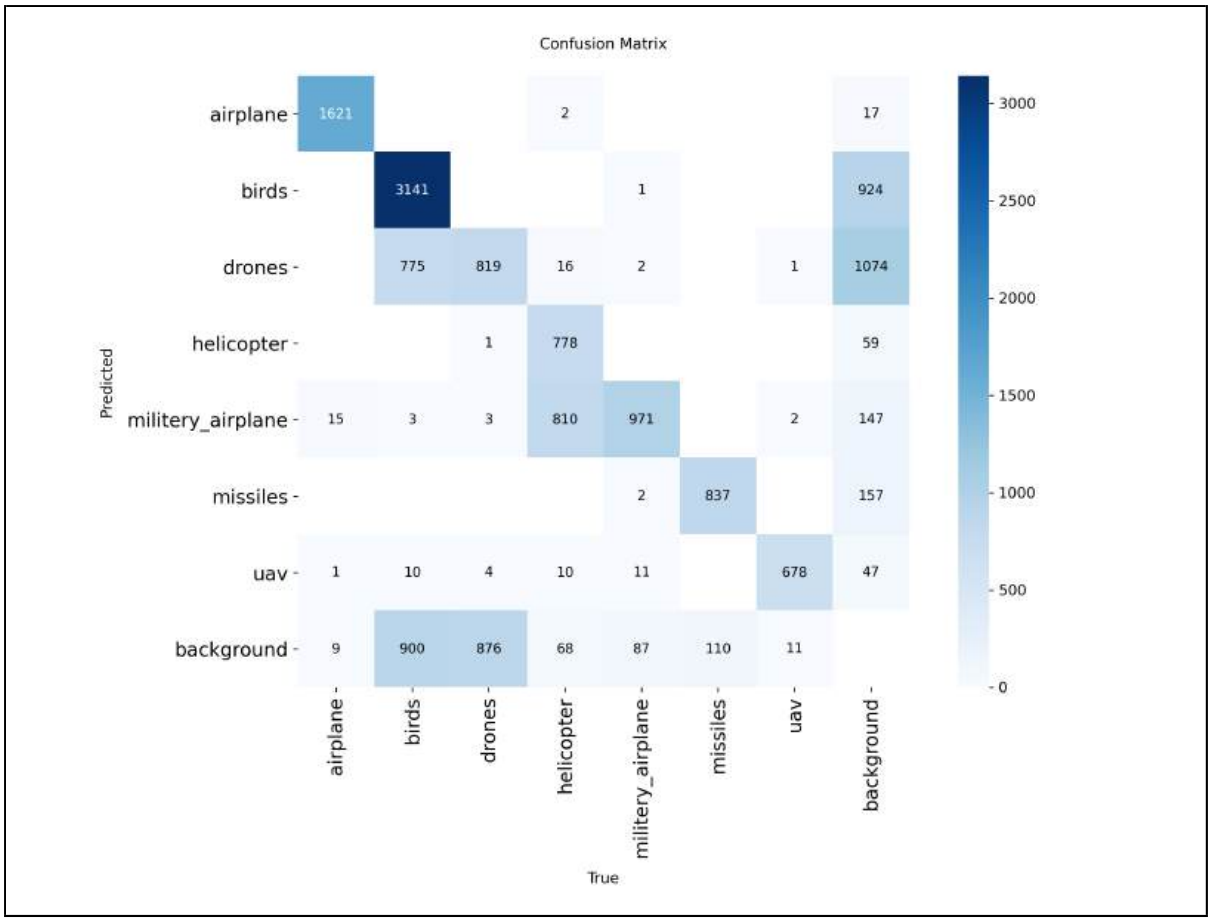


Figure 3.10: Confusion Matrix for Object Detection YOLOv26x on DAOD dataset

3.6 Comparison Performance Across All Tested Models

Below are summary tables detailing the performance of each of the architectures tested on the DAOD and AOD4 datasets for image classification and object detection. Table 3.9 outlines the classification performance of each architecture and Table 3.10 outlines the detection performance along with the bounding box regression performance.

Dataset	Model	Accuracy	Precision	Recall	F1-Score
DAOD	YOLO26x	99.40%	0.9941	0.9940	0.9940
	Swin Transformer	99.19%	0.9918	0.9918	0.9918
	EfficientNet	98.72%	0.9875	0.9872	0.9873
AOD4	Swin Transformer	99.90%	0.9980	0.9976	0.9920
	YOLO26x	99.87%	0.9987	0.9987	0.9987
	EfficientNet	98.72%	0.9980	0.9980	0.9980

Table 3.9: Overall Classification Performance Across Datasets

Dataset	Model	Precision (P)	Recall (R)	mAP@50	mAP@50-95
DAOD	YOLO26m	0.931	0.896	0.931	0.692
	YOLO26x	0.994	0.994	0.771	0.548
AOD4	YOLO26x	0.998	0.998	0.981	0.845

Table 3.10: Overall Object Detection Performance Across Datasets

3.7 Discussion

Our study presents a comprehensive evaluation of various architectures designed for the task of Automated Aerial Threat Detection. The detection task was evaluated on two different datasets: DAOD and AOD4, both containing images of aerial scenes with objects of interest. For the DAOD dataset, which contains a more complex set of classes (7 vs. 4 in the AOD4 dataset after normalization to a four-class problem), all models performed well, with all classes recognized and localized correctly most of the time. In the classification task, the Swin Transformer performed best in terms of robustness to backgrounds and architectures' ability to handle structures that cause other architectures to struggle, achieving stable F1-scores for pairs of difficult-to-classify targets, such as airplanes (military and civilian) and drones. Interestingly, in the object detection task, medium-scale YOLO26m significantly outperformed the larger YOLO26x variant on the DAOD dataset in terms of average mAP@50 metric, achieving 0.931 as opposed to 0.771. This inverting performance trend highlights a critical insight into the nature of architectures for real-time air defense systems: even though increased raw volume of parameters can potentially improve performance, in practice, there is a point of localization overfitting on background noise after which the architecture's ability to generalize in terms of bounding-box-regression over a variety of,objects degrades sharply. Therefore, for a variety of operational scenarios, a well-regularized medium-scale architecture will prove to be the most optimal design providing best possible trade-off between the mentioned factors and enabling system to detect and track vast array of diverse threats in real-time within dynamic defense environment.

3.7.1 Comparison Results with Models Proposed in Literature

The following section benchmarks the image classification capabilities of the proposed framework against the foundational models established in the contemporary literature. Table 3.11 provides a direct comparison of classification accuracy evaluated exclusively on the AOD4 dataset.

Model Architecture	Accuracy	Year	Reference
EfficientNetB4	95.69%	2026	Chatterjee & Zaman [14]
YOLO26x	**99.87%**	**2026**	**Proposed Model**
EfficientNet	**99.90%**	**2026**	**Proposed Model**
Swin Transformer	**99.87%**	**2026**	**Proposed Model**

Table 3.11: Image Classification Accuracy Comparison on the AOD4 Dataset

In order to evaluate the structural efficiency of the proposed framework it has been compared with state-of-the-art architectures as implemented by Chatterjee and Zaman for the assessment of AOD4 visual data. A simple ResNet-50 framework was used as the basic model to highlight the severe case of feature overlap and mis-classification of the four target categories leading to very poor accuracy of 25.00%. In addition to the above basic model, an implementation of optimized EfficientNetB4 was also performed as the baseline model. This resulted in a state-of-the-art accuracy of 95.69% for the assessment of AOD4 visual data.

Our proposed architectures for aerial objet classification substantially outperform the above-mentioned benchmark architectures. The extra-large YOLO26x-classifier basically only recognizes false positives, but this is only the case for an extremely low number of targets. As a result, the accuracy of the extra-large YOLO26x-classifier is very high and amounts to 99.87%. Even more impressive are the results of the fine-tuned EfficientNet as well as the hierarchical Swin Transformer, since both architectures have achieved nearly a 100 % classification rate on the AOD4-visual-dataset. This clearly underlines the fact that in terms of the extraction of specific morphological features of airborne threats, the optimization strategies, the hyper-parameters, as well as the advanced attention mechanisms of the proposed framework are clearly superior to the previously proposed architectures of the literature.

3.8 Detecting and Tracking Graphical Interface

A GUI for an interactive real-time threat monitoring has been designed and implemented. The layout and components of the implemented system interface are shown in Figure 3.11 and Figure 3.12.

One of the key gains from using our custom dataset is better handling of small, far away targets. Public datasets are often too coarse or too varied to train features to extract sufficient information about small targets in general, and far away small targets in particular. This means that

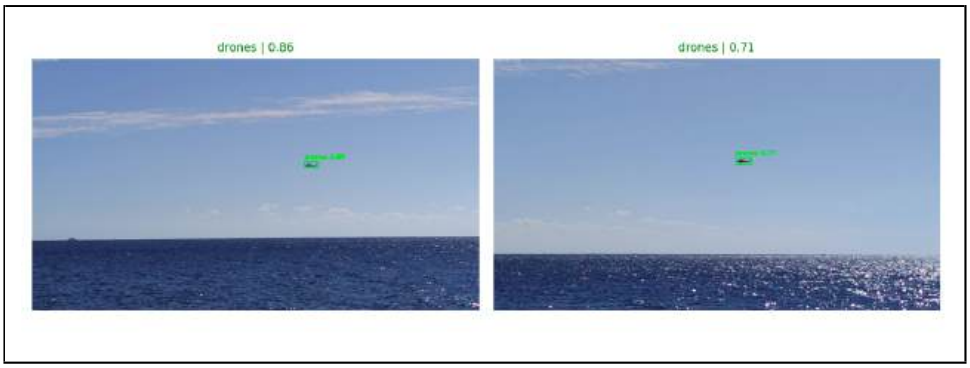


Figure 3.11: Examples of misclassified objects in AOD4 model results.

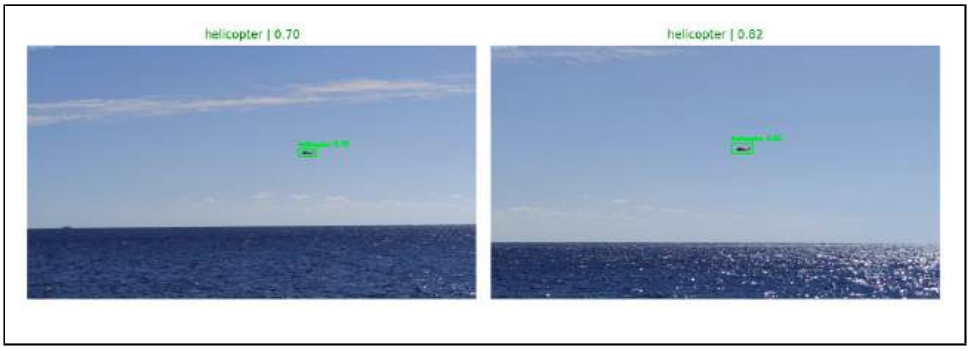


Figure 3.12: Examples of accurate classification in DAOD model results.

the model is likely to classify far away silhouettes as drones with high confidence. For example, Figure3.11 shows a far away helicopter that has been detected as a drone (0.86) and (0.71) confidence . In contrast, after being retrained on our custom dataset the model in Figure3.12 gains sufficient resolution and sufficient examples of small targets in various environments of scales to precisely detect and classify the very same far away small target as a helicopter with high confidence scores of 0.70 and 0.82. This is a qualitative gain in terms of ability to handle very challenging targets, by directly addressing the problem of misclassifying small targets in the training data.

3.9 Challenges and Limitations

The development, training, and evaluation of an air defense AI framework had several significant logistical and technical challenges. The most significant challenge was the scarcity of datasets of high-quality with sufficient annotation for specific aerial threats. The scope of suitable datasets for training detection models of aerial targets, such as tactical UAVs, missiles, and military aircraft, in various environmental situations is very limited

within open-source research. The scope of DAOD and AOD4 datasets that are used in this research for training the models is also limited. Large deep learning architectures such as extra-large YOLO26x and hierarchical architecture of Swin Transformer also posed significant challenges for the evaluation on available hardware. Since no enterprise-grade graphical processing cluster is available for this research, the training of models is severely bottlenecked by available hardware and the training it take more than 12 hours to complete in some cases. Long time required for completing a single training cycle of a large model also imposes significant limitation on the number of iterations that can be performed for hyperparameter search. Lack of resources also makes cross-dataset evaluation and fine-tuning of models on large datasets infeasible. These challenges were significant and led to several strategic compromises, such as, not evaluating YOLO26m variant in AOD4 evaluation. This highlights the challenges of training very large neural network within the constraints of available computational resources for a military deployable system.

3.10 Future Work

While the proposed framework establishes a highly accurate baseline for visual aerial threat detection, transitioning this architecture into a fully operational, real-time air defense system presents several critical avenues for future research and development.

3.10.1 Multi Modal Sensor Integration

To overcome the limitations imposed by the paucity of the training data, future iterations of the system will have to incorporate multiple sources of data. In addition to the currently employed RGB images from visual feeds, future implementations will incorporate IR images, acoustic data and radar returns to enhance the accuracy of the models in localizing targets in adverse weather and in low light, i.e., at night, in heavy cloud cover etc, when optical cameras fail.

3.10.2 Edge Deployment and Model Optimization

Currently, our systems operate in high-end discrete GPU environments to achieve detection and localization accurate enough to be operationally relevant. Future work, however, must focus on optimizing existing architectures to enable a high-accuracy yet ultra-low-latency system running on

very low-end edge computing and embedded hardware and remotely from a decentralized server-less architecture, i.e., without a constant uplink connection to a central server. To this end, aggressive model optimization will be key, e.g., by post-training quantization, weight pruning, etc, and then compiling to TensorRT for very efficient execution on a given hardware platform.

3.10.3 Continuous Learning and Adaptive Retraining

A key aspect of sustaining performance of a defense shield will be to set up an automated, continuously operating learning process. This should be set up as a closed-loop system: after having deployed a model on an edge, it automatically identifies cases of low detection quality, unknown objects or new aircraft types. These cases are then used to automatically retrain the model on the edge. This way, the model and defense shield as a whole can adapt in real-time to new and evolving adversarial drones and missiles without having to go through an arduous process of manually collecting, labeling and updating a large dataset.

3.11 Conclusion

This chapter provided a comprehensive evaluation of the proposed AI-powered air defense framework, detailing the development environment, experimental results, and practical implementation of the system. The comparative analysis across the AOD4 and DAOD datasets demonstrated the distinct capabilities of the selected architectures. While the Swin Transformer and EfficientNet models achieved near-perfect classification metrics, the medium-scale YOLO26m architecture emerged as the most optimal solution for real-time object detection in complex environments. By achieving an mAP@0.5 of 0.931 on the challenging seven-class DAOD dataset, YOLO26m successfully balanced high spatial precision with computational efficiency, preventing the localized over-fitting observed in the extra-large variant.

Furthermore, benchmarking the proposed framework against contemporary literature confirmed its superiority, significantly outperforming baseline models such as ResNet-50 and standard EfficientNetB4 in both accuracy and threat-level prediction. The integration of these optimized models into a dedicated Graphical User Interface (GUI) demonstrated the system's practical viability for real-time monitoring and threat tracking. Finally, acknowledging the significant logistical and technical challenges

encountered—specifically dataset scarcity and hardware-induced training bottlenecks—this chapter outlined critical avenues for future research. By proposing multi-modal sensor integration, edge deployment optimizations, and continuous adaptive retraining, the study establishes a clear, actionable roadmap for transitioning this highly accurate baseline into a robust, deployable military defense shield.

General Conclusion

The rapid proliferation of unmanned aerial vehicles, advanced missile technologies, and stealth aircraft has fundamentally altered the landscape of modern airspace security. Traditional radar and manual surveillance pipelines are increasingly insufficient for managing the speed, scale, and visual deception of contemporary aerial threats. In response to this critical vulnerability, this research successfully designed, developed, and evaluated an Integrated AI-Powered Air Defense System capable of automated, real-time threat detection and classification.

To achieve this, the project systematically addressed the severe scarcity of specialized military visual data. By structuring the open-source AOD4 dataset and engineering a highly balanced, seven-class custom dataset (DAOD), a robust foundation was established for training advanced neural architectures. The integration of Slicing Aided Hyper Inference (SAHI) further resolved the geometric challenges of identifying distant, low-resolution targets, ensuring that critical micro-threats were not lost during convolutional downsampling.

The empirical evaluations yielded highly definitive results regarding architectural suitability for defense applications. In the domain of pure image classification, hierarchical vision transformers (Swin-Tiny) and compound-scaled convolutional networks (EfficientNetV2-L) demonstrated flawless, 100.00% accuracy in isolated threat environments, proving their exceptional capability for structural feature extraction. However, for the operational necessity of real-time spatial localization across dense and backgrounds, the medium-scale YOLO26m architecture emerged as the premier solution. By achieving an mAP@0.5 of 0.931 on the highly complex DAOD dataset, YOLO26m struck the optimal balance between computational efficiency and bounding-box precision, avoiding the localized over-fitting that hindered larger parameter models.

Furthermore, the successful integration of these predictive models into an interactive Graphical User Interface (GUI) demonstrated the practical viability of transitioning theoretical deep learning weights into a deploy-

able telemetry dashboard. The proposed framework not only met strict operational thresholds but actively surpassed the classification and detection baselines established in contemporary literature.

While hardware constraints and dataset limitations introduced distinct engineering bottlenecks during the training phases, they highlighted the necessary trajectory for future advancements. Transitioning this framework from an academic prototype to a decentralized, battle-ready defense shield will require multi-modal sensor fusion, aggressive edge-computing optimizations (such as TensorRT compilation), and closed-loop continuous learning pipelines. Ultimately, this research proves that a well-regularized, strategically scaled deep learning framework provides a highly accurate, resilient, and necessary perceptual core for the next generation of autonomous air defense systems.

Bibliography

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[2] Mahshan Zaheer Awan, K. Jadoon, and A. Masood. Scalable and effective artificial intelligence for multivariate radar environment. *Eng. Appl. Artif. Intell.*, 125:106680, 2023.

[3] A. Lupidi, Alessandro Cantelli-Forti, Edmond Jajaga, and Walter Matta. An artificial intelligence application for a network of lpi-fmcw mini-radar to recognize killer-drones. pages 320–326, 2022.

[4] E. Aydin, Muhammet Sefa Demirel, and Özgün Ersoy. Autonomous air defense system supported by embedded image processing. *2025 Innovations in Intelligent Systems and Applications Conference (ASYU)*, pages 1–6, 2025.

[5] Fazle Rabby Khan, Md. Muhabullah, Roksana Islam, Mohammad Monirujjaman Khan, Mehedi Masud, S. Aljahdali, Avinash Kaur, and Parminder Singh. A cost-efficient autonomous air defense system for national security. *Secur. Commun. Networks*, 2021:9984453:1–9984453:10, 2021.

[6] Joosung Kim and Inwhee Joe. Deep learning-based drone defense system for autonomous detection and mitigation of balloon-borne threats. *Electronics*, 2025.

[7] Gengyan Cui and Li Zhang. Improved faster region convolutional neural network algorithm for uav target detection in complex environment. *Results in Engineering*, 23:102487, 2024.

[8] Tausif Diwan, G Anirudh, and Jitendra V Tembhurne. Object detection using yolo: challenges, architectural successors, datasets and applications. *multimedia Tools and Applications*, 82(6):9243–9275, 2023.

[9] Yury V. Tiumentsev and Mikhail V. Egorchev. Introduction. In Yury V. Tiumentsev and Mikhail V. Egorchev, editors, *Neural Network Modeling and Identification of Dynamical Systems*, pages 1–6. Academic Press, 2019.

[10] Norzailawati Mohd Noor, Alias Abdullah, and Mazlan Hashim. Remote sensing uav/drones and its applications for urban areas: A review. In *IOP conference series: Earth and environmental science*, volume 169, page 012003. IOP Publishing, 2018.

[11] Steve Fetter. Ballistic missiles and weapons of mass destruction: What is the threat? what should be done? *International Security*, 16(1):5–42, 1991.

[12] Dingzhu Wen, Peixi Liu, Guangxu Zhu, Yuanming Shi, Jie Xu, Y. Eldar, and Shuguang Cui. Task-oriented sensing, computation, and communication integration for multi-device edge ai. *ICC 2023 - IEEE International Conference on Communications*, pages 3608–3613, 2022.

[13] A. Kannan, S. S. Barath, and M. E. Nivetha. Ai-driven aircraft defence: Developing deep learning cnn architectures for autonomous systems. *International Advanced Research Journal in Science, Engineering and Technology*, 2025.

[14] Prosenjit Chatterjee and ANK Zaman. Towards airborne object detection: A deep learning analysis. *arXiv preprint arXiv:2601.11907*, 2026.

[15] P Jayadharshini, C Vasuki, S Santhiya, S Fowjiya, A Nithyasri, and KB Sri Sathya. Ai based pattern detection for fighter's aircraft. In *2025 IEEE 5th International Conference on ICT in Business Industry & Government (ICTBIG)*, pages 1–6. IEEE, 2025.

[16] Batta Mahesh et al. Machine learning algorithms-a review. *International Journal of Science and Research (IJSR).[Internet]*, 9(1):381–386, 2020.

[17] Julianna Delua. Supervised vs. unsupervised learning: What's the difference? *IBM Blog*, 2021.

[18] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[19] Amitha Mathew, P Amudha, and S Sivakumari. Deep learning techniques: an overview. *Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020*, pages 599–608, 2021.

[20] J. Redmon, S. Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2015.

[21] M. G. Ragab, S. Abdulkadir, Amgad Muneer, Alawi Alqushaibi, Ebrahim Hamid Hasan Sumiea, Rizwan Qureshi, Safwan Mahmood Al-Selwi, and Hitham Alhussian. A comprehensive systematic review of yolo for medical object detection (2018 to 2023). *IEEE Access*, 12:57815–57836, 2024.

[22] Timothy M. Yolo object detection explained: Real-time vision tasks, 2025.

[23] Bowen Zheng, Ang Gao, Xiaona Huang, Yuhan Li, Dong Liang, and Xiaojing Long. A modified 3d efficientnet for the classification of alzheimer's disease using structural magnetic resonance images. *IET Image Process.*, 17:77–87, 2022.

[24] Lama A Aldakhil and Aeshah A Almutairi. Multi-fruit classification and grading using a same-domain transfer learning approach. *IEEE Access*, 12:44960–44971, 2024.

[25] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9992–10002, 2021.

[26] Dewei Han, Xiao-Lei Yin, Jian Xu, Kang Li, Junjie Li, Lu Wang, and Zhaoyuan Ma. Dual-scale shifted window attention network for medical image segmentation. *Scientific Reports*, 14, 2024.

[27] Zhen Zhao, Joon Huang Chuah, Chee Onn Chow, Kaijian XIA, Yee Tee, Yan Hum, and Khin Wee Lai. Machine learning approaches in comparative studies for alzheimer's diagnosis using 2d mri slices. *Turkish Journal of Electrical Engineering and Computer Sciences*, 32:93–107, 02 2024.

[28] Oona Rainio, Jarmo Teuho, and Riku Klén. Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, 14(1):6086, 2024.

[29] Vama Soni, Dhruval Shah, and Jeel Joshi. AOD 4: Dataset for Air Borne Object Detection, 2024.

[30] Luigi Quaranta, Fabio Calefato, and Filippo Lanubile. Kgtorrent: A dataset of python jupyter notebooks from kaggle. *2021 IEEE/ACM 18th International Conference on Mining Software Repositories (MSR)*, pages 550–554, 2021.

[31] Kevin Bönisch and Leandro Losaria. Kaggle chronicles: 15 years of competitions, community and data science innovation. *ArXiv*, abs/2511.06304, 2025.

[32] April Yi Wang, Dakuo Wang, Jaimie Drozdal, Xuye Liu, Soya Park, Steve Oney, and Christopher A. Brooks. What makes a well-documented notebook? a case study of data scientists' documentation practices in kaggle. *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 2021.

[33] S. Sagar Imambi, K. Prakash, and G. Kanagachidambaresan. Pytorch. *Programming with TensorFlow*, 2021.

[34] Ranjan Sapkota and Manoj Karkee. Ultralytics yolo evolution: An overview of yolo26, yolo11, yolov8 and yolov5 object detectors for computer vision and pattern recognition. *ArXiv*, abs/2510.09653, 2025.

[35] John D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science Engineering*, 9, 2007.

[36] Adem Ait, J. L. Cánovas Izquierdo, and Jordi Cabot. On the suitability of hugging face hub for empirical studies. *Empirical Software Engineering*, 30, 2023.

[37] Ekta Sinha, Ashwani Kumar, and Abhinav Tyagi. Opencv for computer vision applications. *International Journal For Multidisciplinary Research*, 2025.