VISION-BASED MULTI-SCALE UAV DETECTION

BY

ADNAN MUNIR

A Thesis Presented to the

DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fufillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER ENGINEERING

DECEMBER, 2023

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

This thesis, written by **ADNAN MUNIR** under the direction of his thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER ENGINEERING**.

Thesis Committee

Dr. Abdul Jabbar Siddiqui (Advisor)

Dr. Aiman Helmi El-Maleh (Member)

Dr. Ayaz Ul Hassan Khan (Member)

Dr. Aiman Helmi El-Maleh
Department Chairman

Dr. Suliman Al-Homidan
Dean of Graduate Studies

Date

I express my heartfelt dedication in this thesis to my dearest Father, Mother, Wife, and entire family, whose unwavering support has been my pillar in every circumstance. Furthermore, I extend this dedication to my homeland, Pakistan, which granted me nationality and paved the way for me to explore educational opportunities globally.

ACKNOWLEDGMENTS

I extend my heartfelt gratitude to the Almighty Lord for bestowing upon me the ability, talent, enthusiasm, and strength to undertake dedicated research for my thesis. My appreciation goes to several individuals whose sincere guidance and motivation have been instrumental in my thesis journey. I am deeply thankful to my supervisor, Dr. Abdul Jabbar Siddiqui, for their unwavering support, continuous guidance, and invaluable feedback throughout my research. Special thanks to Dr. Aiman Helmi El-Maleh, Dr. Ayaz H. Khan, and Dr. Saeed for their expert feedback, fruitful discussions, advice and access to lab resources. I am indebted to my father, Mr. M. Munir, for being a constant source of inspiration and unwavering support, and to my undergraduate supervisor, Dr. Ajmal Khan, for insightful discussions and guidance in prioritizing tasks. Lastly, I express my gratitude to my mother, siblings, and wife for their encouragement and love, which played a pivotal role in sustaining my morale during the challenges and triumphs of my research journey. I also dedicate this thesis to my dear friends Ibrahim A. Albulushi and Hamed A. al-Mehdar, whose unwavering friendship has added richness and joy to my academic journey. Their support, encouragement, and shared moments of both challenges and celebrations have made this research endeavor even more meaningful.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACBD	Augmented Complex-Background Dataset
ANTD	Additive Noise Test Dataset
AWGN	Additive White Gaussian Noise
CBD	Complex Background Dataset
CNN	Convolutional Neural Networks
IoU	Intersection over Union
mAP	Mean Average Precision
MBTD	Motion Blurred Test Dataset
MWTS	Multi-WeatherVision Tes Set
RTD	Rainy Test Dataset
SBD	Sky Background Dataset
SimAM	A Simple, Parameter-Free Attention Module
UAV	Unmanned Aerial Vehicle
YOLO	You Only Look Once

THESIS ABSTRACT

NAME: ADNAN MUNIR

TITLE OF STUDY: Vision-based Multi-Scale UAV Detection

MAJOR FIELD: Computer Engineering

DATE OF DEGREE: December, 2023

Unmanned Aerial Vehicle (UAV) detection using computer vision and deep learning algorithms is a rapidly growing topic of research that attempts to detect UAVs in real-time. Concerns have been raised about the potential risks of using UAV in civilian and military applications, such as potential privacy violations, safety concerns, and security threats. To address the concerns, vision-based object detection methods have been developed for UAV detection. However, UAV detection in images with complex backgrounds and weather artifacts like rain, noise, and motion blur has yet to be reasonably studied. Hence, for this purpose, we curated two custom training datasets. The first dataset has the sky as its background and is called the Sky Background Dataset (SBD). The second training dataset has more complex scenes (with diverse backgrounds) and is named the Complex Background Dataset (CBD). Furthermore,

to facilitate a more comprehensive evaluation of this model's effectiveness in UAV detection, we have curated a complex background dataset and introduced three distinct test sets affected by adverse weather conditions. These three test sets comprise the Rainy Test Dataset (RTD), the Additive White Gaussian Noise (AWGN) Test Dataset (ANTD), and the Motion Blurred Test Dataset (MBTD). This work not only focuses on benchmarking state-of-the-art object detection models, but also introduces a novel model YOLO-RAW, and to the best of our knowledge, it is the first to investigate the performance of recent and popular vision-based object detection methods for UAV detection under challenging conditions such as complex backgrounds, varying UAV sizes, and low-to-heavy rainy, motion blur, and noisy conditions. The findings presented in the work shall help provide insights concerning the performance of the selected models for UAV detection under challenging conditions and pave the way to develop more robust UAV detection methods.

ملخص الرسالة

الاسم: عدنان منير

عنوان الدراسة: الطائرات بدون طيار متعددة النطاقات المعتمدة على الرؤية الكشف د

التخصص: كلية هندسة الحاسوب

تاريخ الدرجة العلمية: أدخل الشهر والسنة

يعد اكتشاف الطائرات بدون طيار باستخدام رؤية الكمبيوتر وخوارزميات التعلم العميق موضوعًا سريع النمو للبحث الذي يحاول اكتشاف الطائرات بدون طيار في الوقت الفعلي. لقد أثيرت مخاوف بشأن المخاطر المحتملة لاستخدام UAV في التطبيقات المدنية والعسكرية، مثل انتهاكات الخصوصية المحتملة، والمخاوف المتعلقة بالسلامة، والتهديدات الأمنية. ولمعالجة هذه المخاوف، تم تطوير طرق الكشف عن الكائنات القائمة على الرؤية للكشف عن UAV. ومع ذلك، لم يتم بعد دراسة اكتشاف UAV في الصور ذات الخلفيات المعقدة ومؤثرات الطقس مثل المطر والضوضاء وضبابية الحركة بشكل معقول. ولهذا الغرض، قمنا بإعداد مجموعتين من البيانات التدريبية. تحتوي مجموعة البيانات الأولى على السماء كخلفية لها وتسمى مجموعة بيانات خلفية السماء (SBD). تحتوي مجموعة بيانات التدريب الثانية على مشاهد أكثر تعقيدًا (بخلفيات متنوعة) وتسمى مجموعة بيانات الخلفية المعقدة (CBD). علاوة على ذلك، لتسهيل إجراء تقييم أكثر شمولاً لفعالية هذا النموذج في الكشف عن الطائرات بدون طيار، قمنا برعاية مجموعة بيانات خلفية معقدة وقدمنا ثلاث مجموعات اختبار متميزة تتأثر بالظروف الجوية السيئة. تشتمل مجموعات الاختبار الثلاثة هذه على مجموعة اختبار المطر (RTD)، ومجموعة اختبار AWGN (ANTD)، ومجموعة اختبار عدم وضوح الحركة (MBTD). لا يركز هذا العمل فقط على قياس نماذج الكشف عن الأشياء الحديثة، ولكنه يقدم أيضًا نموذجًا جديدًا YOLO-RAW، وعلى حد علمنا، فهو أول من قام بالتحقيق في أداء الرؤية الحديثة والشائعة. طرق الكشف عن الكائنات القائمة على اكتشاف الطائرات بدون طيار في ظل ظروف صعبة مثل الخلفيات المعقدة، وأحجام UAV المتفاوتة، وظروف الأمطار المنخفضة إلى الغزيرة. يجب أن تساعد النتائج المقدمة في العمل في تقديم رؤى تتعلق بأداء النماذج المختارة

لاكتشاف UAV في ظل ظروف صعبة وتمهيد الطريق لتطوير طرق كشف أكثر قوة UAV.

CHAPTER 1

INTRODUCTION

In recent years UAVs have become increasingly popular for aerial photography, delivery services, and surveillance purposes. Yet, the proliferation of UAVs has generated privacy, security, and safety issues. UAVs can be used for harmful objectives such as carrying illegal goods or performing uninvited monitoring. There is an increasing demand for effective UAV detection and identification technologies to address these security and privacy concerns. Image and video-based analysis, which uses computer vision algorithms to detect and track UAVs in real time, is one of the most promising UAV detection methods. This system has various advantages over standard radar-based approaches, including improved accuracy, cheaper costs, and the potential to gather visual evidence for prosecution [6].

The methods used to detect, identify, and locate unmanned aerial vehicles (UAVs) in specific airspace are referred to as UAV detection methods. To detect and track UAVs, several sensors such as radar, acoustic sensors, and electro-optical/infrared (EO/IR) cameras can be used. Radar systems may identify UAVs by detecting radio wave reflections, whereas acoustic sensors detect the specific sound signature of UAVs.

UAVs may be detected using EO/IR cameras that recognize visual features such as form, size, and color [7]. Furthermore, some UAV detection systems analyze communication data and discover patterns related to UAV activity using artificial intelligence and machine learning algorithms. The efficiency of UAV detection technologies is determined by a variety of criteria, including sensor range, size, and speed.

Several image and video-based UAV detection methods, such as object detection, motion analysis, and machine learning-based algorithms, have been presented in the literature. Object identification techniques recognize the UAV as an object in photos or videos, whereas motion analysis techniques detect the UAV's movement patterns. Machine learning techniques use algorithms such as convolutional neural networks (Convolutional Neural Networks (CNN)s) to train the system to detect the visual aspects of the UAV [8].

The UAV detection methods in the literature have not been well investigated for multi-scale UAV detection and there is limited research on UAV detection with real-time devices such as jetson-nano and Raspberry Pi. This motivates us to study how deep learning-based cutting-edge object detection methods would perform for the task of detecting UAVs of varying scales, in varying backgrounds, and on real-time edge devices. In this study, we will look at the newest strategies and tools for detecting UAVs based on their visual features. We will also examine the technology's problems and limits and potential future research areas.

1.1 Background

This section describes the background and different UAV detection methods, followed by the main motivation of this research. As discussed in the section 1, the technique of detecting the existence of unmanned aerial vehicles (UAVs) or UAVs in a certain region is referred to as UAV detection. UAV detection may be accomplished using a variety of technologies, including radar systems, acoustic sensors, and cameras. Several systems combine various technologies to create a more complete picture of the airspace. After a UAV has been spotted, appropriate action can be taken, such as informing authorities, tracking the UAV's flight route, or if required, disabling or intercepting the UAV.

Different UAV Detection Methods

Acoustic(Sound)-based Detection

This type of detection mostly UAVs consists of multiple microphones to capture the sound signatures of UAVs. The reasoning for this strategy is that brushless direct current motors in UAVs create a characteristic hissing high-frequency sound of about 40 kHz. P. Casabianca et al. [9] proposed a UAV detection system based on acoustic features and trained the convolutional neural network (CNN) with a mel-spectrogram as input. This system achieves reasonable accuracy but this type of detection always fails in urban areas, where the sound features of UAVs can be mixed with noise. Audio-based detection also has the disadvantage of not being suitable for the surveillance of patrol UAVs since it would gather the UAV's sound.

Radar-Based Detection

The typical radar approach recognizes big aircraft but fails to distinguish small quadcopters. This type of detection mainly considers radar cross-section features and trains the machine/deep learning algorithm by passing these features as input. It is sometimes difficult to detect and differentiate UAVs from other objects with this type of detection like UAVs and birds. This study [10] discusses the design and experimental findings of a commercial-UAV detection system demonstrator using a widely available frequency-modulated continuous wave (FMCW) radar system operating at 8.75 GHz (X-band). This system achieves reasonable accuracy, and radar could be mounted on patrolling UAVs.

Radio Frequency(RF)-Based Detection

The integration of radio frequency (RF) sensing and deep learning algorithms has been proposed as a solution for UAV detection. However, the lack of databases containing UAV RF signals has been a hindrance in this field [11]. In this paper [12], the authors address this limitation by creating a distinctive open-source database that encompasses RF signals from multiple UAVs captured under various flight conditions. They utilize this database to evaluate a UAV detection and recognition system based on deep neural networks. UAVs are detected utilizing radio waves, which can only travel a limited distance. As a result, the system's detecting range is restricted, and it may miss UAVs that are too far away. For RF-based UAV detection, a clear line of sight (LOS) between the UAV and the detecting entity is frequently necessary. This indicates that the system may be rendered useless in areas where barriers or obstacles block the line of sight.

Vision-Based Detection

The use of cameras and other visual sensors to detect the presence of UAVs in a specific region is known as vision-based UAV detection. This technology is becoming increasingly crucial as the commercial and recreational usage of UAVs grows, bringing significant dangers to security, privacy, and safety [13]. Vision-based detection systems analyze visual data from cameras, detecting UAVs based on distinguishing characteristics such as shape, size, and movement patterns. Machine learning is used in certain systems to improve accuracy and reduce false positives. By using a monocular camera, the authors [14] proposed a system for autonomously detecting and following collaborating UAVs. With a dataset of 26,000 images, the work of [15] presents a modified YOLOv4 for recognizing four types of UAVs distinguishing them from birds, and achieving high accuracy. Gliding birds, on the other hand, provide difficulty for these procedures since they produce unsatisfactory outcomes. Cameras are also very reactive to changes in illumination, and their usefulness is restricted to things in their immediate field of vision.

1.2 Motivation

UAV detection research is motivated by a number of factors, including:

1. UAVs pose a risk to public safety and security, particularly when flown in pro-

Table 1.1: Summary of UAV Detection Techniques

Detection Method	**Target theory**	**Pros & Cons**
Acoustic-Based	Hissing at a high frequency of roughly 40kHz	-Fails in noisy urban area -Range 25-30 ft.
RF-Based	RF control and video transmission	-High altitude does not support -Range<1400 ft.
Radar-Based	Doppler and Micro-Doppler effect	Can not detect small UAVs
Visual-Based	Image object detection	-Range<350 ft. -Hard to distinguish between UAVs and bird

hibited areas or near key infrastructure such as airports, government buildings, and military sites. UAV detection systems may help identify and locate unauthorized UAVs, as well as launch necessary countermeasures.

2. UAVs with cameras have the potential to be used for undesired monitoring, possibly infringing on privacy rights. UAV detection systems can assist in identifying and locating such UAVs.

3. Adversaries may utilize UAVs to launch attacks or spy operations. UAV detection systems can assist in the detection and location of such UAVs, allowing necessary action to be taken to avoid or remove any possible adversarial threats.

The primary motivation for UAV detection research is to ensure that UAVs are used in a safe, responsible, and authorized way, while also preserving the public's well-being and privacy and assuring their security. In comparison to other suggested technologies such as RADAR, acoustics, and RF signal analysis, computer vision-based is frequently used to recognize UAVs autonomously due to its robustness. In this research, we will survey intensively aforementioned UAV detection techniques with their pros and cons. We proposed a custom dataset based on the integration of four well-known and newly used datasets in UAV detection which will be discussed in the dataset section 2.3. Then, eight different one-stage and two-stage object detection algorithms will be surveyed, trained, and tested on the proposed dataset.

1.3 Problem Statement

Unmanned aerial vehicles (UAVs) are increasingly being used in a variety of sectors. UAVs might be utilized in the military to conduct tactical operations in battle strategy. It is vital from a security perspective to detect UAVs since they may be packed with explosives or to avoid security and privacy breaches. Yet, real-time detection of UAVs involves various challenges, such as protecting the camera from heavy rain, direct sunlight, physical damage, and so on. Another issue might be the distance between the camera and the UAV. If the UAV is too far away from the camera, the system will struggle to detect it. Similarly, flying birds may bring a new problem. Computer Vision and Deep Learning-based systems for detecting and tracking objects are becoming increasingly popular. Many state-of-the-art (SOTA) object detectors are producing good results. This thesis focusses on the problem of investigating and enhancing the performance of vision-based UAV detection under challenging conditions such as adverse weather. The specific research questions this thesis seeks to answer are described in the next subsection.

1.4 Research Questions

1. How does the performance of one-stage detectors compare with that of two-stage detectors for UAV detection?

2. How robust and accurate are state-of-the-art detectors for real-time UAV detection?

3. What is the impact of complex backgrounds (e.g., urban scenarios) and adverse weather conditions on the performance of state-of-the-art detectors for UAV detection?

4. How to develop an innovative model designed to exhibit superior resilience to adverse weather conditions compared to state-of-the-art?

1.5 Contribution

The aims of UAV detection relate to the precise goals that an organization or individual intends to achieve by establishing a UAV detection system. As per the following literature review, many different methods and algorithms have been applied and tested. Surprisingly, none of the recent studies have explored the impact of adverse weather conditions on UAV detection. In this work, we make the following contributions:

- We curated and built a dataset of UAVs against intricate backgrounds, and subsequently we extended the original dataset by adding Rainy, Noisy, and

Motion Blurred images by artificially introducing rain, noise, and motion blur effects.

- We investigated and benchmarked the performance of both one-stage and two-stage object detectors, such as YOLOv5, You Only Look Once (YOLO)v8, YOLONASs, RetinaNet, and Faster-RCNN, using our novel dataset containing both clean and adverse-weather training images, and separate weather affected test datasets.

- To the best of our knowledge, this is the first work to benchmark object detection methods for UAV detection in adverse weather conditions and study the performance gains through training with weather-affected images.

- This thesis proposes a novel UAV detection method, i.e., YOLO-RAW, that facilitates multi-scale UAV detection and exhibits robustness against adverse weather conditions, eliminating the need for retraining with images affected by weather artifacts.

- To the best of our knowledge, this is the first work to offer insights into UAV detection methods operating under adverse weather conditions and also to develop a novel model that can achieve UAV detection with improved robustness to adverse weather effects when compared with state-of-the-art.

The work accomplished through this thesis has resulted in the following publications:

- **A. Munir**, A. J. Siddiqui, and S. Anwar, *"Investigation of UAV detection in images with complex backgrounds and rainy artifacts,"* in Proceedings of the

IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops, November 2023.

- **A. Munir**, A. J. Siddiqui, A. El-Maleh, and Khan, AH, *"Benchmarking and Enhancing Robustness By Data-Augmentation of Object Detection Models for UAV Detection"* submitted to IEEE Access, 2023.

- **A. Munir**, A. J. Siddiqui, A. El-Maleh, *"YOLO-RAW: Advancing UAV Detection Robustness in Varied Weather Conditions"* to be submitted.

1.6 Thesis Outline

The thesis is organized as follows:

Chapter 2 provides a comprehensive review of vision and radar-based UAV detection systems, and available datasets.

Chapter 3 describes the details of adverse weather conditions and proposed datasets.

Chapter 4 provides a concise overview of UAV benchmarking techniques and demonstrates how incorporating data augmentation enhances the accuracy of the model.

Chapter 5 details the novel-introduced "YOLO-RAW" model, outlining its modules, presenting results, engaging in discussions, and conducting ablation studies pertinent to this model.

Chapter 6 wraps up by summarizing the accomplishments of this thesis and outlines future avenues of research for UAV detection.

CHAPTER 2

LITERATURE REVIEW

We did a thorough evaluation of relevant papers to better understand the current state of research and datasets on our issue, which we offer in this section.

2.1 Vision-Based UAV Detection

Zheng et al. [16] first introduced the Det-Fly dataset, which contains images of a flying target UAV taken by a different flying UAV. The dataset encompasses a diverse range of real-world situations, featuring different backgrounds, viewing perspectives, distances between objects, flying heights, and lighting circumstances. In addition to evaluating Det-Fly's performance, the authors also assessed eight alternative deep-learning systems using this dataset. Det-Fly comprises a collection of 13,271 images, each depicting a specific micro-UAV target - the DJI Mavic. They fine-tuned and evaluated eight popular deep learning-based object detection techniques: SSD [17], RetinaNet [18], YOLOv3 [19], RefineDet [20], Faster R-CNN [21], FPN [22], Cascade R-CNN [23], and Grid R-CNN [24]. These techniques can be categorized into two types: one-stage networks and two-stage networks. One-stage networks aim to rapidly detect objects by directly performing classification and regression on the feature map. The one-stage networks include SSD, RetinaNet, YOLOv3, and RefineDet. The two-stage networks they evaluated include a region proposal network (RPN) that suggests potential bounding boxes, followed by a classification and regression network that precisely identifies and localizes objects. Techniques like Faster R-CNN, FPN, Cascade R-CNN, and Grid R-CNN fall into the category of two-stage networks. Cascade R-CNN, in particular, outperforms other models, including RetinaNet, RefineDet, FPN, Faster R-CNN, Grid R-CNN, SSD512, and YOLOv3, achieving higher average precision. The dataset of [16] only considers one type of UAV. The methods of [16] performed better on their dataset and not on the MIDGARD dataset of [25], which shows that the MIDGARD dataset is a more challenging dataset due to its variation of backgrounds in the dataset.

In recent years, deep convolutional neural networks (CNNs) for object detection have shown strong performance. Nevertheless, many of the existing CNN-based methods are unable to effectively balance detection performance and model size. In [26], a modest iterative backbone network called TIB-Net, a UAV detection network, was designed to address the issue. They suggest a structure in this network called a cyclic pathway that improves the capacity to extract valuable data from small objects and incorporates it into an effective technique called Extremely Tiny Face Detector (EXTD) [27]. Furthermore, the researchers incorporated a spatial attention module into the suggested network backbone. This module attempts to focus on acquiring useful information about small objects to improve the detection performance of small

UAVs. Nanjing University collected the dataset that was used in this investigation in real-time. Including images with multi-rotor UAVs and fixed-wing UAVs, the UAV dataset contains 2860 photos of various UAV types. Each image's resolution is 1920 x 1080 pixels. This structure's significance is shown by the fact that the addition of cyclic pathways with the no updating method increases UAV detection's mean average precision (mAP). It also raises the mAP when the spatial attention module is used. This proposed system takes more time to process one image than other state-of-the-art algorithms.

The study in [28], considers the challenge of tracking UAVs while giving detailed information like position and trajectory. With more than 300 video pairings consisting of over 580k personally annotated bounding boxes, their study presents a new benchmark called Anti-UAV to aid research on the subject. This benchmark opens a potential approach for long-distance UAV tracking. For evaluating the system performance, the authors collected 318 RGB-T video pairs, consisting of RGB and thermal videos. Three steps, coarse annotation, fine annotation, and inspection and correction are adopted for getting good annotation quality. The dual-flow semantic consistency (DFSC) method is suggested for UAV tracking as a straightforward but efficient method. The 318 RGB-T video pairs with labels in various situations make up the Anti-UAV dataset, "RGB" represents visible light spectrum images in red/green/blue channels, and "T" denotes thermal infrared. The training data consists of 160 sequences and 294.4k bounding boxes, while the testing data consists of 91 sequences and 168.4k bounding boxes. The pre-trained GlobalTrack model is used as the starting model for the Anti-UAV visible dataset. The infrared UAV tracker is trained across 12 epochs, beginning with a learning rate of 0.02. The Anti-UAV infrared dataset is being utilized for training. For visible UAV tracker training, the pre-trained GlobalTrack model is trained using a mix of MS COCO, LaSOT, and GOT-10k datasets. On their proposed dataset, the authors assessed the performance of the top twenty tracker algorithms. On the infrared validation set, SiamRCNN [29] had the highest accuracy score and the maximum success score. GlobalTrack [30] also performs well. However, further research could be done on multi-modal tracking using unaligned data, which might improve tracker accuracy.

Deep learning techniques have been employed to recognize small UAVs in videos. Nalamati et al. [31] investigated different CNN models such as ResNet-101 [32], Inception [33] with Faster RCNN [21], and Single Shot Detector (SSD) [17]. The tests that combined the ResNet-101 basic architecture with Faster-RCNN produced the best results. Their work used the region-based detectors Faster RCNN, Resnet101, and SSD with Inception v2 as the basic architectures. The authors trained the network with transfer learning from publicly accessible pre-trained COCO models to hasten convergence. The UAV vs Bird [34] dataset was used in their study. Faster-RCNN [21] with ResNet101 achieved higher accuracy than with InceptionV2m [33] and SSD [17].

Singha et al. [35] proposed a UAV detection system based on the Birds vs UAV Dataset. The model is evaluated then by using their custom-collected dataset from two different UAV videos. YOLOv4 [36] architecture is fine-tuned on their custom dataset. By using transfer learning, they make the Darknet CNN framework [37] compatible with their own proposed system. The last three YOLO and convolutional layers are fine-tuned on two classes. The Darknet CNN framework used 255 filters and the authors changed their filters to 21 based on the formula $(classesnumber + 5) \times 3$. The custom dataset used in their study was collected from different UAV videos with different sunlight and background conditions. YOLOv4 pre-trained on the MS-COCO [38] dataset was employed in their study, fine-tined on a custom dataset. However, their work is only limited to YOLOv4 and no other object detector was studied and evaluated.

Muhammad. et al. [39] investigated several network topologies based on Convolutional Neural Networks (CNNs), such as Zeiler and Fergus (ZF) from Visual Geometry Group (VGG16) [40]. Due to the scarcity of training data, transfer learning was used to train networks utilizing pre-trained models. Their work focuses on strategies that make use of deep CNNs. The authors used the Caffe deep learning package and Faster RCNN to carry out their tests. Transfer learning from ImageNet is used for model fine-tuning. The ZF architecture is composed of five convolutional layers and three fully connected layers. Similarly, the VGG16 architecture comprises of thirteen convolutional layers and three fully connected layers. The Bird vs UAV dataset was used in their study. A distinct size, vantage position, and lighting are used as the UAV enters the picture. The models were trained on a 64-batch NVIDIA Quadro P6000 GPU with a learning rate of 0.0001. The VGG16 showed the best performance in their work.

In video sequences with the presence of birds and other distracting items as well as movements in the background or front, the work of [41] identifies one or more UAVs that emerge at a specific point in the videos. The deep learning approaches are proposed and evaluated on the birds vs UAV dataset. There are 77 distinct videos in the training set for the challenge. The average number of frames in the films is 1384. The videos are taken in different places with different background conditions. Seven types of UAVs were considered in their work. The standard cascade R-CNN architecture was trained with the COCO dataset with standard anchors. Second, with the UAVs vs Birds dataset, the RetinaNet is used with custom anchors. Third, a novel technique is described that dynamically modifies these anchors depending on the input picture size, the number of feature pyramid levels, and their corresponding strides, trained with all available datasets. Their RetinaNet-based model achieved higher precision and outperformed the other two methods. In this study, [42] the deep learning object detector YOLOv3 is trained with pre-trained weights and transfer learning to detect the UAVs. With the input image of size 416*416, the proposed method achieves good accuracy. To detect small UAVs effectively, the authors also captured images of UAVs at different altitudes, ranging from 10 to 50 meters, with intervals of 10 meters. A total of 1500 UAV photographs were carefully evaluated to remove extraneous images, yielding a final batch of 1435 images. The study discovered that with the YOLOv3 custom detector, UAVs may be spotted with good accuracy. Their work was not tested in a real-time environment.

Lv et. al. [43] proposed a detection technique combining background differences and the lightweight network SAG-YOLOv5s to address the issue of the system. To decrease computational overhead, most of the background is first removed from high-resolution images before possible UAV targets are extracted. The attention mechanism and the Ghost module are integrated into the SAG-YOLOv5s lightweight object detection network that is proposed in their study. The images dataset is produced based on the publicly available dataset UAVs vs Birds [34] dataset. The 4k videos are selected with a total of 9410 images of 3840 $\times$ 2160 pixels. The authors combined their 4K video picture sequences with 1876 annotated images (6240 $\times$ 4120) pixels from field-collected videos to create a high-resolution UAV dataset of 11,286 images. It contains many images having tiny objects of resolution (30 $\times$ 30) or less. The proposed system after utilizing the background difference showed a 73.3% to 97.0% increase in mean average precision (mAP) values. However, Intersection over Union (IoU) is not discussed which is crucial for any object detection model.

The YOLOv5[44] model has also been used for detecting and classifying small UAVs such as in the work of [45]. Their model was trained on the Det-Fly [46] dataset paired with one of the publicly accessible UAV air-to-air datasets [47] with complex backdrops and lighting scenarios. On the 10% randomly selected data from the entire dataset, YOLOv5 and Faster R-CNN are tested and evaluated. The ResNet50 as a backbone network, together with an FPN detector neck, was employed with the Faster R-CNN model. The YOLOV5x pre-trained network from the YOLOV5 repository was utilized and trained via transfer learning. No proper results are discussed for Faster-RCNN. Once the UAV is obscured or flies in front of certain hidden backdrops, such as a tree canopy or a structure, it cannot be seen. When a flock of birds flies in front of the camera, it is possible to misidentify them as UAVs in addition to insects. A further issue with the Faster R-CNN+FPN was the repetitious detection, which led to an excessive number of false negatives.

Mahnoor et al. [48] proposed a multifeatured and advanced UAV detection network for SafeSpace. The authors adjusted the backbone and neck of the YOLOv5 network to build a new model known as MFNet to increase the speed and precision of recognizing tiny objects. The researchers changed the kernel size in both the backbone and neck to improve the YOLOv5s model's feature sensitivity and scalability. This change was made to modify the size of the derived feature maps, which resulted in increased accuracy and performance on unfamiliar data. The authors presented three MFNet versions, MFNet-S, MFNet-M, and MFNet-L, which varied in the kernel and feature map sizes, corresponding to small, medium, and large settings. To train their MFNet model, the researchers collected over 5105 photos of UAVs and birds from publicly available open-source datasets on Roboflow. They selected 2605 UAV images and 2500 bird images from a total of 5105 images. The suggested model's performance is measured using many metrics, including precision (P), recall (R), mean average precision (mAP), and intersection over union (IoU). In comparison to the YOLOv5s model, the suggested MFNet-M obtained higher mAP than other MFNet versions. Birds obtained the lower precision using MFNet-M than UAVs. However, the Kite in the background is also detected as a UAV.

By collecting a UAV-to-UAV (UAV2UAV) tracking dataset in the study of this work [49], the authors investigate the issues of UAV-to-UAV tracking. This dataset has 44 videos in total (23k frames). To comprehend the functionality of existing tracking techniques and set a foundation for future research, the authors conduct a comprehensive analysis of 19 state-of-the-art trackers, encompassing both manually crafted feature-based and deep-learning-based methodologies. This research scrutinizes a range of advanced trackers, incorporating those reliant on hand-crafted features and those leveraging deep-learning features. The study introduces a novel UAV2UAV tracking algorithm, utilizing data augmentation within a correlation filter. This real-time tracker is based on the histogram of oriented gradient (HOG) features. Its CNN-based design allows it to perform comparably with cutting-edge techniques. Each tracker is measured using one-pass evaluation (OPE). There are several methods for correlation filter tracking are used. The comparison of the suggested technique with the hand-crafted feature-based methods revealed that the proposed method based on the HOG feature performed better.

In [50] a method for identifying and monitoring autonomous UAVs is described, which employs two types of cameras: a stationary wide-angle camera and a lower-angle camera mounted on a spinning turret. The authors introduce a novel multi-frame deep learning detection approach that merges the frames captured by a wide-angle static camera with those from a zoomed camera. This integration optimally utilizes both memory and time, facilitating the simultaneous identification of minute aerial intruders on the primary image plane and their detection on the magnified image plane. The cameras offer a resolution of 2000×1700 pixels, operating at a frame rate of approximately 25 FPS. The wide-angle camera is equipped with a 16 mm focal length lens, providing an expansive field of view of around 110 degrees. The revolving turret's camera features a 300 mm lens, giving it a diagonal field of view of around 8.2 degrees. The modified lightweight YOLOv3 model is used for detection. Due to privacy considerations, the data generated or evaluated during the study is not available to the public. The lightweight YOLO is being implemented, tested, and compared with other models such as the Cascade Haar feature classier and a Gaussian mixture model (GMM) background subtraction algorithm. However, tracking and detection are just focused on the YOLO algorithm, and only one UAV type is considered.

Ali et. al. [51] created a custom dataset and applied YOLOv2 and YOLOv3 algorithms to detect unmanned aerial vehicles (UAV). YOLO (You Only Look Once) is a popular method for recognizing things in real-time. It works by applying a single neural network to the entire image, separating it into regions, and then predicting bounding boxes and probabilities for each region. The custom dataset is created and annotated with the BBOX label tool. Only UAV and no UAV classes are considered. There are 12 images in the no-UAV category, whereas the UAV category contains 137 images. The 100 images are added as training and the remaining are utilized as validation. The supplied image is scaled down to 416 x 416 pixels and then divided into a 5 x 5 grid using YOLO. If the center of an item lies on a grid, the grid's job is to forecast the object by calculating the bounding box confidence score and conditional probability. YOLOV3s performs better than YOLOV2s However, YOLOv3s training time was significantly longer than YOLOv2s since darknet-53 is a deep network with 53 convolutional layers. Simple two algorithms are used, the dataset is very limited, and accuracies could be increased. No false alarm was considered.

In their work, [52] authors proposed a detection algorithm for moving and still objects. To detect different types of UAVs, the YOLOv3 model is trained using a suitable dataset and trained for 150 epochs only. A convolutional neural network using contemporary object identification techniques demonstrates an outstanding way for real-time UAV detection. For feature extraction from images, YOLO used a convolutional neural network (CNN). For better and quicker performance, it makes a few minor adjustments to them. In YOLOv3, DarkNet53 is employed, which uses 53 convolutional layers to extract features from the images. The dataset is created after collecting images and videos from the internet. More than 10,000 images are collected for different types of UAVs. At several iterations, the model's performance is examined. The mean average precision (mAP) value is computed to assess the detection performance. Based on the data, the model performs best at the 150th epoch with an mAP of 0.74. A very basic dataset and techniques are applied. Only precision and recall are considered. No false alarm is considered.

Wisniewski et al. [53] proposed a convolutional neural network (CNN) for different UAV model identification. This synthetic dataset is used for training while real-life video data are used for testing. The classification includes the DJI Phantom, DJI Mavic, and DJI Inspire UAV models. The suggested technique minimizes the time spent manually labeling UAVs. The method to generate a dataset synthetically is the main part of this paper. The synthetically created 1000 images dataset is used for training. ImageNet is used to pre-train the CNNs. The parameters include a learning rate of 0.0001, a momentum of 0.9, and a batch size of 32. The CNN is trained for 100 epochs and to avoid overfitting, a dropout of 0.25 is used. The anti-UAV [54] dataset is used for testing that contains the videos and ground truth labels. According to [55], freezing the layers during neural network training increased accuracy for synthetic to real transfer. When layers are not frozen, a network's accuracy rises to 92.4% from 73.0% when layers are frozen. The mean average precision of the proposed system is not discussed. Only DenseNet201 is trained and tested.

Pawelczy et al. [56] created a new object detection dataset that is specially tailored to train machine learning systems for binary object recognition. This dataset was generated to enable the automatic detection of several UAV objects in an industrial scenario utilizing video feed. The authors started by creating a test set and a subset of the final training set to fine-tune pre-trained CNNs for generating "UAV proposal" bounding boxes. These bounding boxes can then be utilized to accelerate the tagging process without requiring object-tracking methods. The authors then presented and successfully implemented this innovative way of getting bounding box tags for object detection. This study used Haar Cascade-based object detectors to incorporate models for edge devices. Finally, the MobileNet model is fine-tuned. The custom dataset is generated having up to 51446 images. The training and testing datasets were created using a total of 56821 images and 55539 bounding boxes. The images are taken from the internet, and some are collected in real-time by authors. A total of 51446 images are scaled down from different resolutions (varying from 640 $\times$ 480 to 4K) to 640 $\times$ 480 making up the whole training dataset. The 51445 images have 52676 UAV bounding boxes altogether and 1 negative picture (not containing any UAVs). On the other hand, the testing set consists of 5375 images, out of which 2750 images are negatives (not containing any UAV). The best model achieved 67% accuracy, 61% F1 score, 0.56 AUC, 52% mAp, 73 precision, and 52% recall. The given dataset is not compared with any benchmark dataset. Even with transfer learning (MobileNet), they achieved a 60.2% score. So, with some finetuning, it could be enhanced.

A. Husodo et al. [57] proposed a radar-like sensor tracker that comprises on 2D camera and GPS (Geo Positioning Sensor) for the detection and tracking of UAVs. The intruder's presence is detected by a 2d camera, and then the intruder UAV position coordinates are sent to the specific server. The UAV tracker locates the target, and then follows and tracks it using a visual-based tracking strategy. A real-time visual-based target localization system is created by the UAV tracker to determine the target's precise location in the outside environment. The long-term object tracking dataset is considered [58]. UAV-tracker used an Adaboost machine learning classifier for intruder detection in an image. The target position is also located by the UAV tracker in the picture frame coordinates (x and y). The Bayesian-based tracking algorithm is used. 5. This work is not based on real-time. This system could be implemented with a Raspberry Pi camera and GPS sensor in real-time.

With the help of 241 videos containing 331,486 images, the authors [59] applied and tested four detection and three tracking algorithms. The overall 98.6% mean average precision (mAP) is achieved by the detector algorithm while 98.7% MOTA is obtained by the tracking algorithm. The authors considered the Faster-RCNN, YOLOv3, SSD, and DETR benchmark algorithms for object detection. The SORT, DeepSORT, and Tracktor algorithms are utilized for object tracking. The researchers used three datasets to conduct the study: MAV-VID (Multirotor Aerial Vehicle VID), Anti-UAV, and UAVs-vs-Bird. The MAV-VID dataset is made up of 53 movies, each with 29,500 images for training, and 11 videos, each with 10,732 photos for validation. In MAV-VID, the average object size is 136 77 pixels. The UAV vs. bird collection includes 77 films and 104,760 pictures. The 6 different types of UAVs flying with different lighting conditions are considered by the anti-UAv dataset. It consists of an average object of size 52 $\times$ 29 pixels. Although Faster RCNN consistently earns the greatest mAP for the category of identifying small UAVs YOLOv3 overall produces the best precision (as high as 0.986 mAP). As a detection backbone for tracking systems for tiny objects, DETR works well with cross-modal videos. Further evidence that the Tracktor architecture is the most effective way to track UAVs is provided (MOTA: 0.987). They used already proposed datasets. Table 2.1 presents some recent studies that used methods, their datasets, their contribution, and overall achieved precision of their models.

Table 2.1: Summary of vision-based UAV detection papers

Ref	Model Tested	Dataset Used	Comparison with another dataset	Task	Custom Dataset	Mean Avrage Precision (mAP)
[16]	SD, RetinaNet, YOLOv3, RefineDet, Faster R-CNN, FPN, Cascade R-CNN, Grid R-CNN	Det-Fly	MIDGARD	Detection	✓	Grid R-CNN: 82.4%
[60]	Dual-flow semantic, consistency (DFSC) method	Anti-UAV	×	Tracking	×	SiamRCNN:95.70% GlobalTrack : 87.13%
[31]	ResNet-101 with Faster-RCNN, InceptionV2 with Faster-RCNN, Single Shot Detector (SSD)	UAV vs Bird dataset	×	Detection	×	ResNet + Faster-RCNN:0.49% Inception+ Faster-RCNN0.35% SSD:0.15%
[41]	YOLOv4	UAV vs Bird dataset	×	Detection	✓	YOLOv4: 74.36%
[35]	Cascade-RCNN, RetinaNet, Dynamic Anchor Modification	UAV vs Bird dataset Custom	UAV vs Bird dataset	Detection	✓	Dynamic Anchor Modification: 100%
[42]	YOLOv3	Private	×	Detection	✓	YOLOv3: 88.9%
Ours	YOLO-RAW	Proposed Dataset	×	Detection	✓	95.2%

2.2 Radar-Based Detection

The traditional constant false alarm rate (CFAR) detector based on radar signal has demonstrated benefits in the identification of moving targets. These detectors do not perform well on small moving objects. This study [61] comprised a Neural networks classifier that predicts the presence of the target of the input range-doppler map patch, and a Neural network regressor for checking the offset between the target and the patch center. A non-maximum suppression (NMS) method is used to eliminate false alarms and control them efficiently. This approach has three components depending on the network's output: probability-based initial recognition, distribution densitybased recognition, and voting-based regression. Their system's backbone architecture consists of three convolutional layers, two pooling layers, and two fully linked layers. Two sibling completely linked layers are used for class recognition and offset regression. This study considers the Realtime data and simulated data and tested their model on both datasets. The proposed study achieved a 0.985 target detection rate and 0.80 false alarm rate for simulation data while a 0.66 target detection rate and 0.839 false alarm rate are achieved on the real-time dataset at 20db SNR. The suggested method's training and testing phases take a long time due to a series of patches obtained from range-doppler radar, which restricts its practical applicability.

UAV detection systems have been created because of an increase in UAV accidents and misuse. The non-coherent S-band radar is used in this study [62] for two classification approaches of flying UAVs. They collected radar datasets of UAVs, ships, and birds in various backgrounds. The neural networks are used in both classification methods. Using features that are extracted from track data, the first method is trained. While using radar images the second method is trained. The main part of this paper is to detect the DJI Phantom UAV. An automated dependent surveillance-broadcast system (ADS-B) was used to gather and label data. For analysis, the study used two alternative methodologies. The first approach entailed extracting eight characteristics such as track area, speed, movement, and amplitude. These characteristics were fed into a shallow neural network, which classified the object as a bird, ship, or UAV. To evaluate the data, the second approach combined Long Short-Term Memory (LSTM) with Convolutional Neural Networks (CNN). A series of radar scan images is passed as the CNN-LSTM architecture's input. After flattening the image, with

220 hidden neurons, LSTM is performed. The image sequence is classified as a final step using the SoftMax function and a fully connected layer. The overall accuracy of UAV detection with method 1, is 81.16%, and method 2 is 89.3%. The deep learning method clearly achieved 8% better performance than feature-based classification. But this method needs more data in the training phase and needs high power and computation.

Detecting and classifying small items is frequently desirable in security applications. For detecting and classifying small UAVs, this study [63] used radar micro-doppler. Doppler shifts known as micro-Doppler are caused by small motions of the objects. They used time velocity graphs (TVDs) of birds and UAVs generated by radar. With the help of a deep learning classifier, they differentiate between UAVs and birds and types of UAVs. Their system architecture consists of three convolutional with max-pooling and two dense layers with dropout. The Rectified linear unit (Relu) is used in hidden layers and categorical cross entropy is used as a loss function. The dataset of TVDs is divided into five groups and 5-fold cross-validation is used. The two dense layers have linear activation functions. Their proposed systems achieved 98% accuracy.

R. Akter et al. proposed a residual neural network-based UAV detection surveillance system [64]. The suggested system's processing unit is organized into four blocks, each of which is responsible for extracting advanced characteristics. In addition, the network uses traditional convolution layers to extract generic information. A real-time radar dataset including micro-doppler signatures of autos, persons, and UAVs is used to evaluate the system. Based on the dataset, the model starts with an input layer of size 11613. Two convolutional layers are used, with kernel sizes of 33 and 11, 32 channels, and a stride of 11. The generic features are extracted by the first two convolutional layers, while the unique features are extracted by the processing unit layers, which comprise four blocks of the deep feature processing unit. Then batch normalization and max pooling are performed. The RDRD dataset is used, which is publicly available. The dataset consists of 5065 samples of UAVs, 6700 samples of people, and 5720 samples of cars. Different deep learning models are applied and compared with the proposed architecture. The proposed study achieved 95.92% accuracy, while the classification accuracy of XGBoost, MLP, and ResNet50 is 81.73%, 70.83%, and 91.8% respectively.

From images of an ultra-wideband (UWB) radar range profile this study [65] describes a convolutional neural network (CNN) model. The research focuses on five different types and styles of UAVs: Matrice 600, 3DR Solo, Phantom 3, Mavic Pro, Bebop UAV, and Bionic Bird. Radar pre-processing methods are used before the implementation of CNN. These approaches entail converting received power to radar cross section (RCS), eliminating clutter, and creating radar pictures for use as input to the CNN model. The highest accuracy is achieved by Matrice 600 because of differences in shape and size. The accuracies of each class's 3DR solo, Phantom 3, Mavic Pro, Bebop UAV, and Bionic Bird are 78.5%, 87.5%, 95.5%, 88.0%, and 98.0% respectively. The suggested method was not tested with changing UAV flying postures, and it did not account for the spatial connection between the radar antenna and the target UAV.

To enable the development of the most efficient approach for detecting and identifying UAVs, Radio Frequency (RF) signals are used which are remotely detected and recorded. In this research [66], authors collected different UAVs' RF signals in different modes such as off, on hovering, and video recording. The presence, type, and mode of an intruding UAV are classified with three different deep neural networks. The signal transformation and multi-class classification are the two main parts of this study. Signal transformation techniques are used to alter complicated RF signals in order to uncover hidden information that can aid in efficient detection and identification. The modified RF signals are then sent into deep neural networks, which are used to categorize the invading UAV. An RF database stores the RF characteristics of three unique UAVs operating in various modes. The RF data is tagged with a Binary Unique Identifier (BUI), which includes information about the kind, size, and mode of the UAV. Deep neural networks are trained and tested using a UAV RF database that has been generated. The first DNN recognizes whether a UAV is present or not. According to the results, the average F1 score is 99.5%, the average accuracy is 99.7%, and the average error is 0.3%. A second DNN checks the presence and type of UAV. Results show an average F1 score of 78.8% and an average accuracy of 84.5%, with an average error of 15.5%. Finally, the third DNN detects the presence, type, and mode of the UAV. It shows an average accuracy of 46.8%, an average error of 53.2%, and an F1 score of 43%.

Amateur UAVs (ADRs) can cause some serious security threats, that need proper investigation and instant detection of ADRs. In their study [67], the authors introduced a Machine Learning framework for detecting UAVs, birds, and airplanes based on their acoustic sounds. Sounds are recorded through arrays of microphones and passed to the ground control station. Raw sound could affect the accuracy of the system, for that reason they extracted important features from sound by applying Machine Learning techniques. The study utilizes feature extraction techniques such as Mel frequency cepstral coefficients (MFCC) and linear predictive cepstral coefficients (LPCC). They trained their system by applying Support Vector Machine (SVM) with different kernels. The 138 sound samples are used for training and 34 sound samples are used for testing. 10-fold cross-validation is used to enhance system accuracy. They registered 85% accuracy with the MFCC approach with various kernels under a noisy environment. Overall, their system achieved 97% detection accuracy with acoustic features. A comparison of different SVM kernels is performed. The cubic kernel with MFCC provides higher accuracy than other SVM classifier kernels. However, the dataset used in this study is limited and did not test with deep learning approaches.

F. Svanstrom et al. [68] come up with a multi-sensor UAV detection system. The researchers combined numerous sensors, including a video camera, audio sensors, and a thermal camera, to develop a durable system. To aggregate the data from various sensors and develop a robust detection system, sensor fusion techniques were used. UAV detection was largely accomplished using the thermal infrared camera (IRcam) and the video camera. A system that broadcasts the aircraft's position, velocity vectors, and identifying data is required in the event of cooperative aircraft. Using the characteristic noises, a UAV or a helicopter is identified. Since main cameras have a small field of view, the fishy-eye lens camera is installed to cover the 180° horizontally and 90° vertically Field of View (FoV). All computations are carried out by laptop and the sensor's reading is monitored by MATLAB Graphical User Interface

(GUI). The servo controller and GPS sensor are used to update the GUI script about the target. Both the thermal infrared camera (IRcam) and the video camera (Vcam) utilize the trained YOLOv2 detector and classifier, which are fundamentally similar. The main GUI script receives information from these cameras, including the identified target's class, detection confidence, and the horizontal and vertical offsets in degrees from the center of the image. On the other hand, the acoustic sensor extracts MFCC features, which are then inputted into a Long Short-Term Memory (LSTM) classifier. The customized dataset is used, the input layer of YOLOv2 is 256×256 and the four output classes are considered. The visible video detector performed similarly to the infrared detector, with an F1-score of 0.7601. The F1-score for the audio classifier is 0.9323. Still, the False Alarm Rate (FAR) is not discussed in this paper.

In [69] micro-unmanned aerial vehicles (UAVs) are detected and classified using Radio Frequency (RF) signal features, transmitted from the controller to the UAVs. This study is divided into two subtasks. Firstly, to check whether Micro-UAV is present in the RF signal or not. Second, to classify the micro-UAV based on its RF signatures. Using a predefined wavelet tree, the RF signals are converted into the wavelet domain. The customized approach was chosen because it aids in RF signal detection even in low SNR environments. The existence of a UAV in each frame is determined using a naïve Bayes method that is based on Markov models created separately for UAV and non-UAV classes. This proposed study used the energy-transient signal for micro-UAV classification because this method can work with different modulation techniques and is efficient with noise. The extraction of the RF fingerprints occurs once the energy transient has been identified. These statistical moments serve as the energy transient's "fingerprints," or physical characteristics. Kurtosis (k), skewness (γ), variance (σ), and energy spectrum entropy (H) are the characteristics that were extracted. To keep the algorithm's computing cost low, significant features are chosen using neighborhood component analysis (NCA). These features are fed to k-nearest neighbor (KNN), Support Vector Machine (SVM), and Neural Networks (NN) classifiers to classify the UAV type. The k-nearest neighbor (kNN) classification method successfully detects every micro-UAV with an average accuracy of 96.3%. the deep learning models were not tested, which could be more accurate than machine learning models.

The authors of this work [70] assess a Convolutional Neural network (CNN) classifier's capacity to distinguish UAVs from non-UAV targets, such as birds when those targets are absent from the training set. The training is performed on both real and synthetic data. The authors further examine which regions of the Doppler spectrum receive greater weights in the classification stage to add some transparency to the classifier. A radar system was employed to collect datasets from various sources, including six different UAVs, people, birds, and trihedral corner reflectors. The training set consisted of a varying number of data samples, ranging from 4028 to 10093, for each class. The remaining 1000 samples were allocated for testing, while another 1000 samples were used for validation. The CNN classifier was trained using stochastic gradient descent with adaptive moment estimation (Adam) as the optimization algorithm, employing the categorical cross-entropy loss function [71]. Their system architecture consists of three convolutional and two fully connected layers. Relu is used for the hidden layer, while SoftMax is used for the final layer. Overall system accuracy is noted as 68%. For UAVs vs non-UAVs, the mean classification accuracy achieved 88% and 93% respectively. This study is limited to the simple CNN classifier.

V. Semkin et al. [72] proposed a UAV detection and classification system by using radar cross-section (RCS) signatures. The Radar Cross Section (RCS) of a carbon fiber UAV model was evaluated at a frequency of 28 GHz. To minimize any interference caused by reflections from surrounding objects, the measurements were conducted within an echoic chamber. The UAV was rotated in the azimuth plane, then rolled in 1-degree increments around its center axis to determine its bottom hemisphere. This method enables the collection of the UAV's quasi-3D RCS signatures. The authors simulate a 4x4 uniform rectangular antenna array that operates as a 28 GHz radar to analyze the effects of an RCS-based UAV detection probability. The RCS values and probability of detection are measured at different distances such as 30,50,70, and 90 meters. As the distance increased, the detection probability of the UAV with the same Radar Cross Section (RCS) decreased. Despite the presence of a 5G base station monitoring the UAV trajectory and ensuring a high detection probability at the average RCS value, there remains a notable risk of losing track of the UAV target at certain angles because the receiver received considerable power reflection from only a certain area of a UAV.

The paper [73] introduces a UAV classification approach based on convolutional neural networks (CNN). An essential requirement for achieving accurate classification using neural networks is a large and diverse training dataset. To address this, the authors generate a database comprising micro-doppler spectrogram images of UAVs and birds captured during their flight. The identical images have been divided into two distinct datasets, one with RGB images and the other with greyscale. The training process involved using the RGB dataset to train the GoogLeNet architecture-based model. Additionally, a series of networks proposed in the paper were trained using the grayscale dataset. The input shape of GoogleNet is RGB $224 \times 224 \times 3$. For GoogleNet, 6 epochs are used, and each epoch consists of 32 iterations with a learning rate of 0.001. Two class training systems got 100% training and 99.1% validation accuracy. The validation accuracy for 4 classes is noted as 95.9% and with dropout, it was 91.3%. The model overall achieved 90% testing accuracy in both cases. For the 4-class series network, accuracy was 95.1%, for the 2-class series network, 98.5%, for the 4-class GoogleNet, and 99.0% for the 2-class GoogleNet. This shows that the models that were developed are effective enough to be included in a real-time radar-based UAV detection system.

In [74], Ignacio and Carlos proposed the UAV detection and recognition system for other targets. The optical sensors that are often needed for the recognition portion of systems like the one described employed persistent range-Doppler radar, which has fewer limitations. They create a database of UAVs, people, and cars with different backgrounds named RDRD [75]. The RDRD database is comprised of numerous CSV files, each containing an 11×61 matrix representing the power values in dBm for each cell. Based on the statistics, approximately 32.71% of the data samples correspond to cars, 28.97% to UAVs, and 38.32% to people. The classification of targets is performed using a convolutional neural network (CNN) classifier. The architecture of the classifier includes a single convolutional layer with a filter size of 3×3 and 32 filters, followed by a flattened layer and four fully connected layers. The hidden layers utilize the rectified linear unit (ReLU) activation function, while the final layers employ the SoftMax activation function. The MobileNet and NasNet are compared with the proposed system. With five-fold cross-validation, average accuracies achieved by MobileNet are 98.94%, 97.69% for NasNet, and finally 99.48% for the proposed system. Accuracy is simply one factor to consider while evaluating the network. Precision is a crucial factor to consider since a false positive for a UAV might result in unneeded and occasionally hazardous actions.

Table ‘2.2 shows the summary of radar-based UAV detection systems. It briefly shows the different methods, datasets, and precision of different most commonly used methods.

Table 2.2: Summary of Radar-based UAV detection papers

Ref	Model Tested	Dataset Used	Task	Custom Dataset	Mean Avrage Precision (mAP)
[36]	Neural Network Regressor, with Nonmaximal Suppression	Realtime and Simulation data	Detection	✓	Realtime: 98.5% Simulated: 66%
[39]	Residual Neural Network, XGBoost, MLP, ResNet50	RDRD	Detection	×	Residual Neural Network: 95.92% XGBoost, MLP, ResNet50: 81.73%, 70.83%, 91.8%
[40]	Convolutional Neural Network (CNN)	×	Detection	✓	3DR solo: 78.3% Phantom3, Mavic pro, 95.5%, 88.0%
[41]	Deep Neural Network (DNN)	×	Detection	✓	DNN: 99.5%
[49]	Convolutional Neural Network (CNN)	RDRD [50]	Detection	✓	CNN: 99.48% MobileNet: 98.94% NasNet: 97.69%

2.3 Datasets

Data is the foundation of any machine/deep learning research, and as the amount of available data grows, so does the demand for good data collection, administration, and analysis. In this study, we look at several datasets obtained from various sources and papers. This sub-section will provide an overview and analysis of the available data, identifying patterns and trends within datasets and evaluating their strengths and limits. Table 2.3 shows the different datasets with their size, shape, number of images, type of UAVs, background, and input size of the image.

Ref	**Dataset**	**# of classes**	**# of Images**	**Resolution**	**Types of UAVs**	**Background**	**Bird**	**Single or Multiple UAVs**	**Available**	**Input** *Size*
[76]	20 Video sequences 20 YouTube videos	1	4000	752×480	UAVs Aircraft	Sky	$\times$	single	$\times$	NA
[77]	30 YouTube videos (thermal and normal)	1	1000	1280×720	UAVs helicopter USC((phantom 3)	Indoor and outdoor (clean and cloudy)	$\times$	Single	✓	NA
[77]	30 YouTube videos	1	1000	1920×1080	UAVs, helicopters USC phantom 3	Indoor and outdoor (clean and cloudy)	$\times$	Single	✓	NA

[25]	MIDGARD	1	NA	480×752	UAVs (DJI F550)	Forests, Garden, Meadow, Different light	$\times$	single	✓	NA
[47]	Det-Fly-simple	1	13271	3840×2160	UAVs DJI Mavic2	Clean sky	$\times$	Single	✓	YOLOv3 $(416, 416\}$
[47]	Det-Fly complex	1	13271	3840×2160	UAVs DJI Mavic2	Field, mountain, urban	$\times$	Single	✓	YOLOv3 $[416, 416)$

[78]	Realtime data	3	1500	32×16	Hex copters, quad-copter rs, and UAVs	Sunny. cloudy	$\times$	Multiple	$\times$	NA
[79]	Kaggle data	1	1500	2200×1467	quad-copter	Sunny. cloudy		Multiple	✓	NA
[80]	UAV detection dataset	4	90 audio, 650 videos clips (IR) and visible camera)	640×512	UAVs, birds, airplanes, helicopter	Sunny. cloudy	✓	Single	✓	YOLOv2 $[256 \times 256\rangle$

[81]	NPS-UAVs	1	50 videos	(1920×1080 and 1280×760		outdoor	$\times$	Single	✓	NA
[74]	UAV data (Radar)	1	17000 samples	NA	UAVs	outdoor	✓	Single	✓	NA
[82]	UAV data (radar)	3	3400	224×224	UAVs/ No UAVs, bird, | clutter, noise	NA	✓	single	✓	NA
[83]	UAV's data	NA	51,455	640×480	Multiple types of UAVs	Outdoor with different background	$\times$	Multiple	✓	255×255

[60]	Anti-UAV	4	160 high quality, full HD video sequence	640×512	DII-Inspire, DII-Phantom4 Dji-Mavic	NA	$\times$	Multiple	✓	NA

Table 2.3: Existing Datasets for UAV Detection (vision based)

2.4 Challenges in Vision-based UAV Detection

Due to the tiny size, rapid speed, and agility of UAVs, as well as their ability to fly at low altitudes, vision-based UAV identification offers various obstacles. Some of the most significant issues in vision-based UAV detection are as follows:

1. Complex Backgrounds and Scenes: Detecting UAVs through visual analysis of the surrounding environment can be difficult in urban areas with a lot of visual clutter, like buildings, trees, and other obstacles. It can be a challenge to differentiate between UAVs and other objects in such environments.

2. Illumination Conditions Variations in illumination conditions can reduce the accuracy of vision-based UAV detection systems. UAV identification can be hampered by shadows, reflections, and glare caused by bright sunshine.

3. Multi-Scale UAVs: UAVs come in a variety of sizes, from tiny hobby UAVs to

bigger commercial UAVs. Because of their tiny size, smaller UAVs are difficult to detect, making them less apparent and simpler to avoid detection.

4. UAVs vs Birds: Many methods exhibit confusion between birds and UAVs in images captured from real-world scenarios.

5. Adverse Weather Conditions: Many UAV detection methods suffer performance degradation under challenging conditions such as rain, blur (that may be caused by winds), image noise, etc.

CHAPTER 3

PROPOSED DATASETS

This work involved the review and acquisition of several datasets from various open-source or public-domain sources. After the review, several datasets were combined to curate custom and more challenging datasets to train and test. This dataset contains a total of 12,492 distinct images, collectively referred to as the Complex Background Dataset (CBD). These images are composed of a variety of backgrounds, including those found in urban and semi-urban settings, as well as those found in indoor and outdoor settings. The lighting conditions of the backgrounds are also varied, ranging from sunny to cloudy, as illustrated in Fig. 3.2. Another category, named *Sky Background Dataset (SBD)*, comprises 3888 images, which exclusively feature a sky background, as illustrated in Fig. 3.1. We have curated multiple test sets to evaluate the proposed models which we will explain in detail. A summary of all the datasets introduced in this article is presented in Table 3.1.

3.1 Sky Background Dataset (SBD)

The SBD is formed by merging three different datasets (selecting images with the only sky as background), namely: Det-Fly[46], Wosdec challenge[34], and a real-world object detection dataset designed for quadcopters[84].

Figure 3.1: Sample images from the proposed Sky Background Dataset (SBD) are shown. The red bounding boxes show the original location of the UAV in each image.

3.2 Complex Background dataset (CBD)

In the course of examining other datasets, certain restrictions were identified. For instance, the datasets containing the birds were not included in the datasets containing the UAV model, such as the MIDGARD dataset [85], Det-Fly [46] dataset, the Ant-UAV dataset, and so on. Additionally, some of the datasets contained only a single model of UAV. The CBD is a compilation of seven datasets (only images with complex backgrounds were selected for inclusion) including the Det-Fly dataset [46], Wosdec challenge dataset [34], Real-World Object Detection dataset for quadcopters [84], MIDGARD dataset [85], and Vision-Based Anti-UAV dataset [86]. Annotations from various formats have been converted into YOLO format and JSON format. To further enhance the complexity of the datasets and to test the detectors' ability to distinguish between UAV and bird images, bird flying images were included in the dataset.

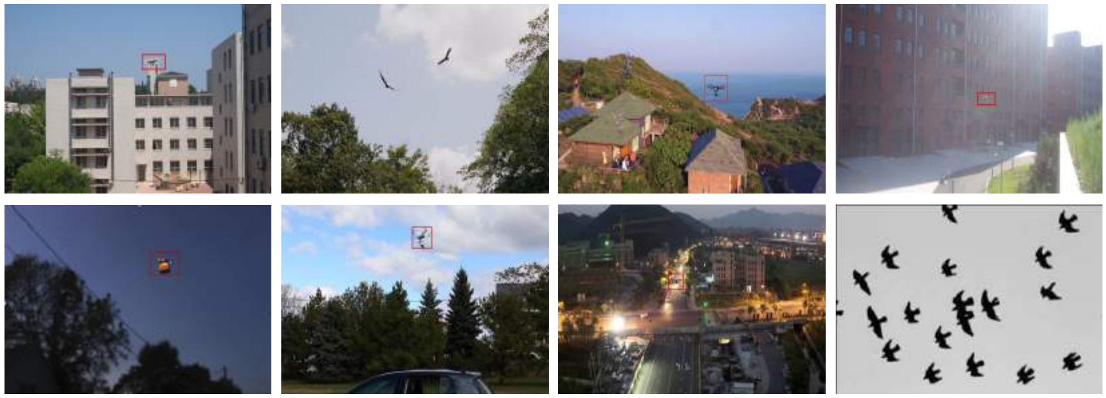


Figure 3.2: Sample images from the proposed Complex Background Dataset (CBD) are shown. The red bounding boxes show the original location of the UAV in each image.

3.3 Adverse Weather Affected Test Sets

Artificial image distortions are frequently used for a variety of objectives, including data augmentation, testing the durability of image processing methods, and evaluating the performance of computer vision systems. Many of the prior works on vision-based UAV detection focused on clear images, e.g., in the Anti-UAV [86] dataset, the adverse weather effects such as rain, noise, and motion-blurred were not considered. In this work, weather-affected test sets are built based on the Anti-UAV [86] test set, hereby referred to as the Test Set (TS), which is not included in the training/validation data of the CBD dataset.

Table 3.1: A summary of all the novel datasets used for training, validation, and testing.

Dataset	CBD	TS	Rainy Test Set (RTD)			Additive Noise Test Dataset (ANTD)			Motion Blur Test Set (MBTD)		
			Drizzle	Heavy	Torrential	Low	Medium	High	Low	Medium	high
Training	9725	-	-	-	-	-	-	-	-	-	-
Validation	2767	-	-	-	-	-	-	-	-	-	-
Testing	-	2528	2528	2528	2528	2528	2528	2528	2528	2528	2528

3.3.1 Rainy Test Dataset (RTD)

Rainy conditions may lead to a decrease in the performance of UAV detection techniques, a topic that has not been extensively explored in previous work to the fullest extent. Rain is composed of numerous drops of varying sizes and intricate shapes, which are dispersed randomly and at different speeds when falling on streets, pavements, vehicles, pedestrians, and aerial drones in the environment. Additionally, raindrops can cause pixel values to change due to the blocking of reflected light from objects in the scene, and rain streaks can cause visual data to appear less contrast and more white. To consider these challenging conditions, rain streaks were synthetically added to the test images of the Anti-UAV [86] dataset to build the Rainy Test Set (RTD) proposed in this work. The clean images of RTD are utilized to obtain baseline performance of selected models, while the rainy images of RTD are used to analyze the impact of rainy artifacts.

Three types of rain effects (drizzle, heavy, and torrential) were added to the TS dataset images and generated rainy test set drizzle ($RTD_{drizzle}$), rainy test set heavy (RTD_{heavy}), and rainy test set torrential (RTD_{tor}). A popular tool employed to generate synthetic (yet realistic) rain effects on images is the automold library of [87] as used in the literature of other domains [88], [89], [90]. It is worth mentioning that

Figure 3.3: Sample images from the proposed Complex Background Test Set (CBTS) and Rainy Test Set (RTD). Three types of rain artifacts applied are *drizzle, heavy* and *torrential*. The top row shows the original clean image (before applying artifacts).

the rain effects are added only to the test set images (and not to the training sets) to study the performance of the selected models under investigation. While one may argue that including rain effects in training images could enhance a model's accuracy on rainy test images, this paper chooses to train the selected models only on clean (no rain) data to facilitate a better understanding of the strengths and weaknesses of the preferred models in unforeseen rainy conditions.

3.3.2 Motion Blurred Test Dataset (MBTD)

In photography and computer vision, motion blur is an optical effect caused by the movement of an object or camera during the exposure period of an image. The resulting image is streaked or blurred. Motion blur can lead to a loss of detail in an image [91]. Unmanned aerial vehicles (UAVs), particularly those that are small and move quickly, can be difficult to identify even in a clear image. Motion blur can cause the distinctive characteristics of a UAV to be blurred, making them difficult to identify. To cater to this situation, we have curated three different test sets by applying motion blur to the TS dataset. The $MBTD_{low}$, $MBTD_{medium}$, and $MBTD_{high}$ datasets are obtained by executing a 2D convolution of each image with a filter of low, medium, and high sizes. Samples images are present in Fig. 3.4

3.3.3 Additive White Gaussian Noise (AWGN) Test Dataset (ANTD)

AWGN, commonly referred to as additive white Gaussian noise, adds a considerable amount of random noise to images, resulting in a significant decrease in image quality. Image clarity is of paramount importance in vision-based UAV detection, and the presence of noise in images can obscure essential details, making it difficult to accurately identify and track UAVs [92]. Many UAV detection systems are based on computer vision algorithms, AWGN can interfere with the functioning of these algorithms, resulting in a more complex UAV detection and tracking process. Three types of artificial AWGN noise effects such as medium, low, and high are added to the

Figure 3.4: Sample images from the proposed Complex Background Test Set (CBTS) and MBTD. Three types of motion blurness effects applied are *low, medium* and *high*. The top row shows the original clean image (before applying the blur effects to images).

TS dataset by multiplying the original image by different amounts of noise to curate a $ANTD_{low}$, $ANTD_{medium}$, and $ANTD_{high}$. Some sample images from the ANTD dataset are shown in Fig. 3.5.

3.3.4 Augmented Complex-Background Dataset (ACBD)

To get better insights into how state-of-the-art object detectors will perform on UAV detection systems with clean and adverse test datasets when original datasets are augmented with adverse weather effects. This dataset is based on the training set consisting of 19446 which combines images with a ratio of 33% noisy images, 33% motion blurred, and 34% rainy images and combined with 100% of CBD. These motion-blurred, rainy, and noisy images are created synthetically by using clean images.

3.3.5 Multi-WeatherVision Test Set (MWTS)

To create a diverse dataset, we have brought together various weather conditions and visual variations, including clean, motion-blurred, AWGN-affected, and rainy images. This compilation is named the Multi-Weather Test Set (MWTS). MWTS consists of 700 images in each category: clean, rainy, noisy, and motion-blurred, resulting in a total of 2,800 images. These images are taken from RTD, MBTD, and ANTD datasets.

Figure 3.5: Sample images from the proposed Complex Background Test Set (CBTS) and AWGN Test Set (ANTD). Three types of AWGN noise effects applied are *low, medium* and *high*. The top row shows the original clean image (before applying artifacts).

We have intentionally included multi-scale UAV images against complex backgrounds and various weather conditions to elevate the dataset's level of challenge and to better assess the capabilities of our proposed model.

CHAPTER 4

INVESTIGATION OF UAV DETECTION IN ADVERSE WEATHER CONDITIONS

This work seeks to investigate the impact of complex backgrounds and adverse weather conditions such as rain, motion blur, and noise on the performance of UAV detection methods and examine the advantages or disadvantages of the selected methods. To this end, a custom test dataset is curated (i.e., the RTD, MBTD, and ANTD described in the previous section), considering challenging conditions such as varying UAV scales, background variety, and other adverse weather effects effects. The object detection methods selected to be investigated in this work include critical one-stage detectors such as YOLOv5 [93], YOLOv8 [94], RetineNet [95], and a popular two-stage detector called Faster-RCNN [96].

4.1 Object Detection Models

In this section, a brief overview and description of the selected object detectors are provided.

4.1.1 YOLOv5

The YOLOv5 model (specifically, the YOLOv5s version) consists of three main components: Darknet-53 [37] backbone network, PANet [97] neck network, and the Head (for outputs). The basic structures of YOLOv5s are CBS and CSP_n. The CBS is a basic convolution module that includes a batch normalization operation and a SiLU activation function. The CSP_n consists of two branches, the first of which is a series of n Bottleneck modules, while the second is a CBS convolution block. These two branches are then stacked to increase the network depth and enhance feature extraction capabilities. The input terminal is responsible for tasks such as improving data and adapting anchor boxes. The backbone network comprises 2 CSP_1 and 2 CSP_3 structures for extracting features. By using multiple down-sampling and up-sampling the neck network combines features of the different levels. The neck network has four CSP_1 structures and combines features of different levels using multiple up-sampling and down-sampling, resulting in multi-scale output and improving the detection of small objects. The output head performs bounding box regression and NMS (Non-Maximal suppression) post-processing to achieve precise target detection [98]. The YOLOv5 model architecture is shown in Fig. 4.1. The neck network has four CSP_1 structures and combines features of different levels using multiple up-sampling and

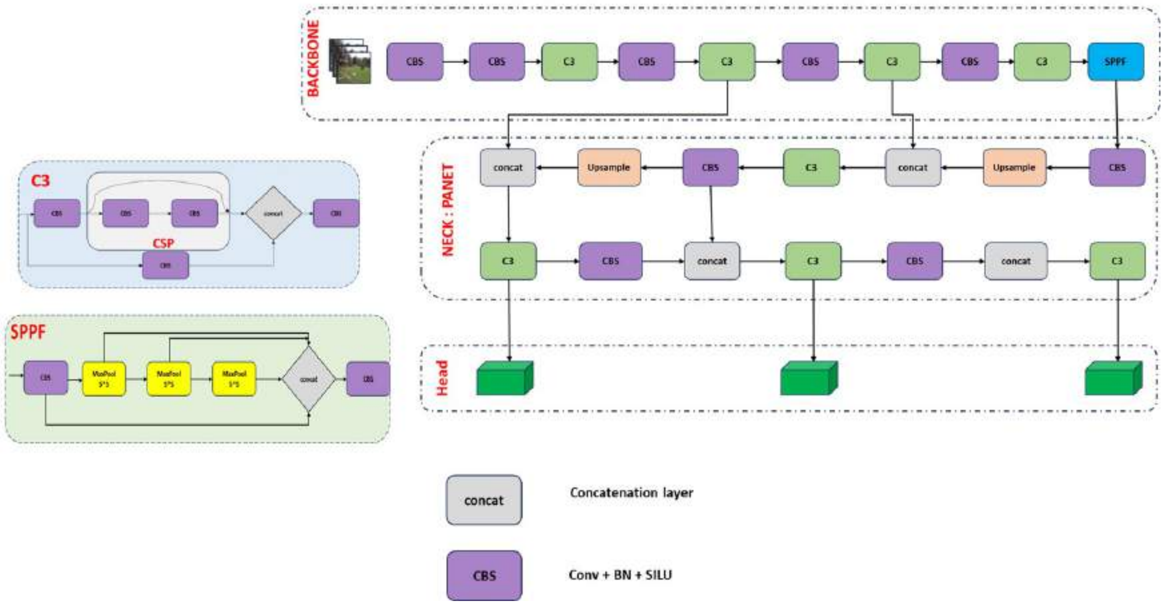


Figure 4.1: Network architecture for YOLOv5 [1]

down-sampling, resulting in multi-scale output and improving the detection of small objects. The output head performs bounding box regression and NMS (Non-Maximal suppression) post-processing to achieve precise target detection [93]. The YOLOv5 model architecture is shown in Fig. 4.1.

4.1.2 YOLOv8

YOLOv8 is an enhanced version of YOLOv5 produced by the same company Ultralytics [99] and is the most recent edition of the YOLO model used for object detection and image segmentation. This advanced model improves on its predecessors by adding new features and enhancements that boost its performance, adaptability, and effectiveness [93]. The YOLOv8 model surpasses its earlier versions by integrating advanced technologies such as a novel backbone network, an anchor-free split head, and updated loss functions. These changes empower YOLOv8 to produce superior outcomes, while still being compact and remarkably fast. Furthermore, YOLOv8 does not rely on anchors. This implies that it forecasts the center of an item directly rather than the offset from a known anchor box. The Non-Maximum Suppression (NMS) is accelerated by anchor-free detection since it minimizes the number of box predictions. In YOLOv8, the first 6x6 convolution in the backbone was replaced with a 3x3 one, and two of the later convolutions (No.10 and No.14) were removed from the YOLOv5 setup. The Fig. 4.2 presents the architecture of YOLOv8.

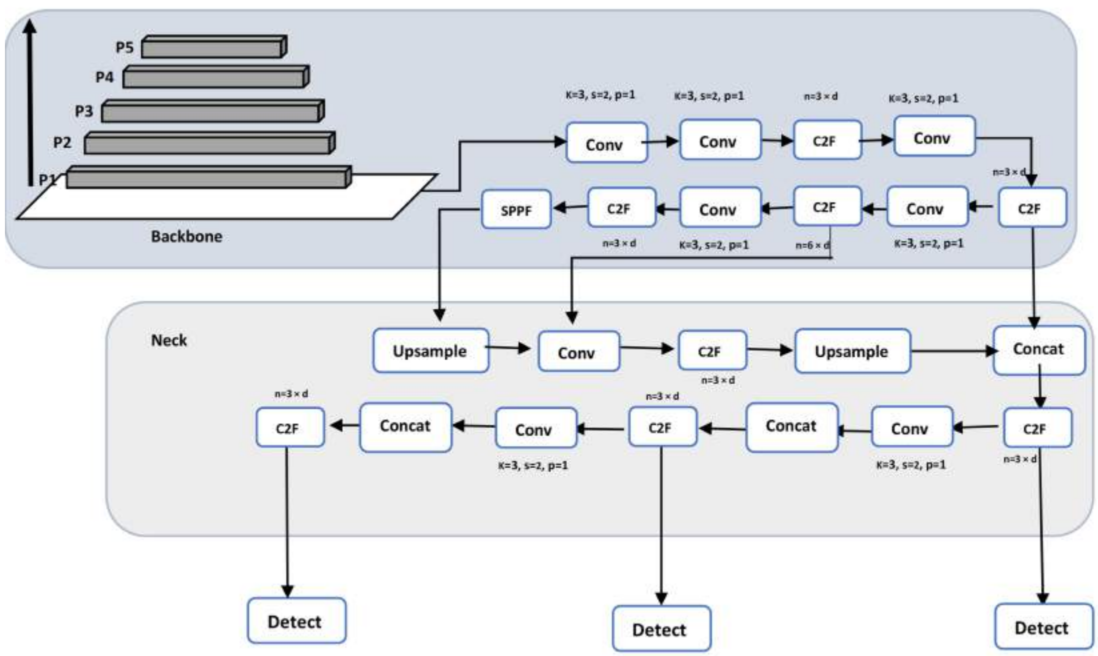


Figure 4.2: Network architecture for YOLOv8 [2]

4.1.3 RetinaNet

RetinaNet is an object detection model that includes a Feature Pyramid Network (FPN), a backbone network, classification and regression subnetworks. The backbone network is in charge of extracting features from the input images, and it employs

ResNet to generate four feature maps with varying resolutions. The FPN then combines these feature maps to form a pyramid of multi-scale feature maps, which are utilized to identify objects of varied sizes. RetinaNet is intended to manage unbalanced data and objects of various sizes, which it does through its unique architecture and usage of the Focal Loss function. It is a one-stage object detection model that is tuned for high object detection accuracy [18]. The fundamental component of its architecture is RetinaNet's Focal Loss Key. The RetinaNet model architecture is shown in Fig. 4.3.

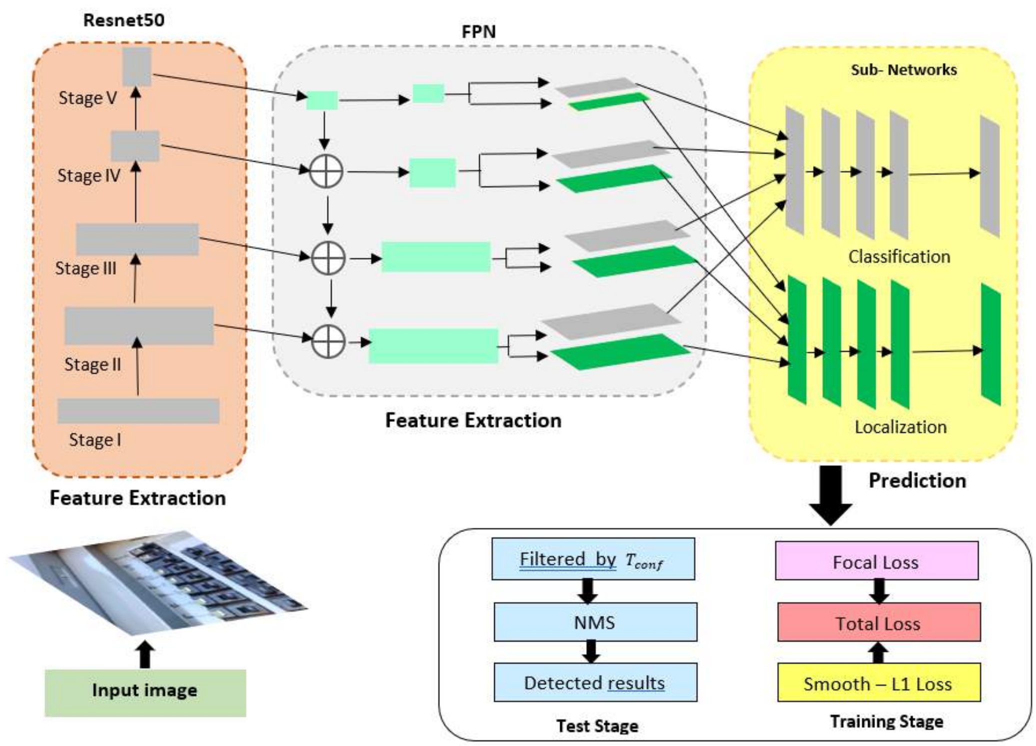


Figure 4.3: Network architecture for RetinaNet [3]

4.1.4 Faster-RCNN

Fast R-CNN is an object detection network that overcomes various shortcomings of the prior R-CNN model by increasing speed. The Region of Interest (ROI) Pooling layer in Faster R-CNN gathers feature vectors of the same length from each ROI inside an image. This is an enormous improvement over R-CNN. Faster R-CNN goes one step further by employing a single-stage network rather than R-CNN's three-stage method as shown in its architecture in Fig. 4.4. It conducts computations such as convolutional layer calculations once and then distributes them across all suggestions (such as ROIs). Because of the new ROI Pooling layer, Faster R-CNN is quicker than R-CNN. Furthermore, unlike R-CNN, Faster R-CNN does not require the storage of extracted features, which saves disk space. Faster R-CNN outperformed R-CNN in terms of accuracy [21].

4.1.5 YOLO-NAS

The Deci AI has released YOLO-NAS [100], their latest deep learning model that provides enhanced real-time object identification skills and high performance that is ready for production. Well-known datasets such as COCO[38] and Objects365 are used to pre-train this network. To improve its training process, the YOLO-NAS design integrates Knowledge Distillation (KD) and Distribution Focal Loss (DFL). The model's architecture incorporates adaptive quantization, a feature that omits quantization at specific levels based on the balance between improving latency/throughput and the loss of accuracy. Upon conversion to its INT8 quantized version, YOLO-NAS

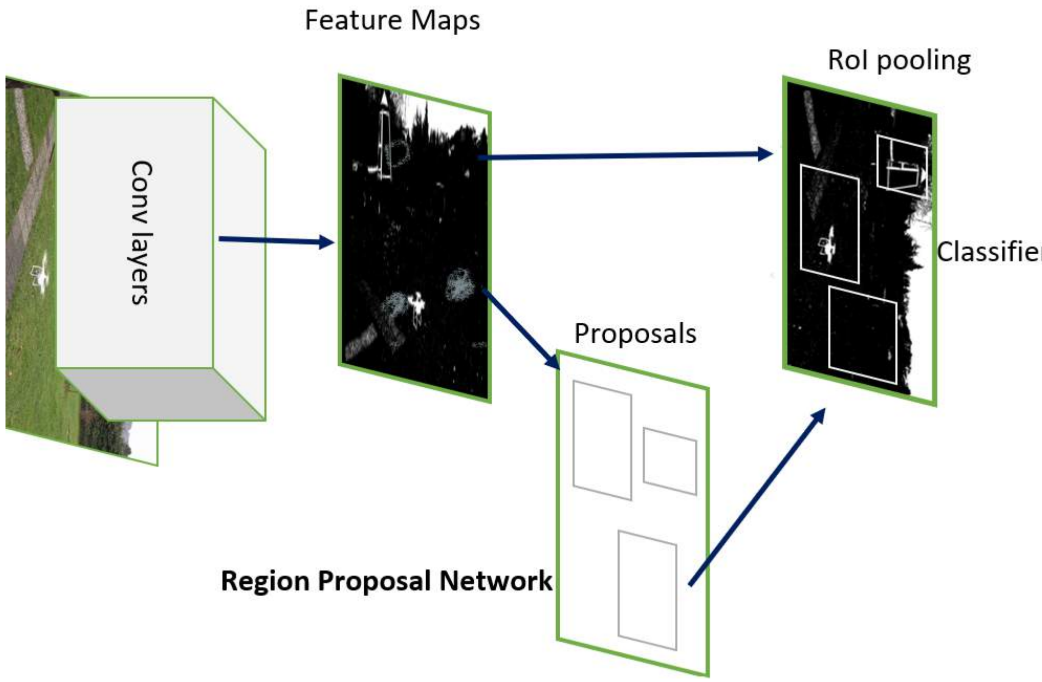


Figure 4.4: Faster R-CNN architecture [4]

experiences a smaller decline in accuracy (0.51, 0.65, and 0.45 mAP points for Small, Medium, and Large versions, respectively) compared to other models, which exhibit a greater loss of 1-2 mAP points during quantization.

4.2 Experimental Setup

This section will briefly introduce the tools and performance metrics employed in this work and the set of experiments conducted. The PyTorch [101] framework is used for the deep learning model implementations, as well as the Ultralytics [44] library. The proposed datasets described in chapter 3 are used in the experiments. The annotations follow the format of the YOLO [98] and of COCO [38] datasets. Input images were configured to have dimensions of 640×640. Training and evaluation of the baseline models and enhanced models were conducted on an NVIDIA RTX-A4500 GPU, for 100 epochs, with a batch size of 16.

4.2.1 Experiments

In this study, extensive experiments were conducted to investigate respectively the impact of:

- Complex Backgrounds
- Noisy Images (Additive White Gaussian Noise)
- Motion Blur Effects (varying severity)
- Rain Artifacts (varying severity)

Moreover, this work investigates the effectiveness of a strategy to enhance the robustness of the selected models to adverse weather conditions through the incorporation of adverse weather effects on training set images. In Sec. 4.3, results and discussions on these experiments are provided along with analysis of detection failures and Grad-CAM-based [102] analysis of selected models' performance.

4.2.2 Performance Metrics

The efficiency and accuracy of object detection systems are evaluated using performance measures. These metrics quantify how successfully a model recognizes and localizes objects inside video or image frames.

Mean Average Precision (mAP)

The mean average precision (mAP) is a frequently used performance statistic in the field of object detection. This parameter entails averaging the precision values over different recall levels ranging from 0 to 1. Because it gives a full summary of a model's detection accuracy, the mAP is a popular choice for assessing object detection methods. We may define average precision as the area beneath the precision-recall curve.

$$mAP = \frac{1}{n} * \sum(AP)$$

Where n is the number of classes.

Intersection over Union (IoU)

The Intersection over Union (IoU) measure determines the extent to which the predicted and actual bounding box coordinates overlap. A higher IoU score indicates that the projected bounding box coordinates are more closely aligned with the ground truth bounding box coordinates.

$$IoU = \frac{Area of Overlap}{Area of Union}$$

4.3 Results and Discussion

This study offers practical results that incorporate an in-depth assessment and analysis of various advanced one-stage and two-stage object detection models, including YOLOv5, YOLOv8, YOLO-NAS, Faster-RCNN, and RetinaNet for UAV detection. The dataset is divided into two segments: training, and validation, with allocation proportions of 70%, and 30% respectively. The NVIDIA RTX A4500 is utilized in our experiments. All models undergo training for 100 epochs. The validation data helps in assessing how well the model generalizes to new, previously unseen samples.

4.3.1 Detections on *CBD*

This section describes the validation and testing mean average precision (mAP) of used models with complex datasets. Table 4.1 shows that YOLOv5 outperformed all other models for this custom dataset.

The results of the Faster R-CNN model, depicted in Fig. 4.5, showcase identified UAVs through bounding boxes in the image. Similarly, the YOLO-NAS model exhibits reliable detection capabilities, as illustrated in the bounding boxes. The RetinaNet model, with its effective localization using bounding boxes, demonstrates its proficiency in identifying objects. The YOLOv5 model's detection results underscore its ability to discern and highlight items within the complex background of the image. Lastly, the YOLOv8 model's detection outcomes are presented in the final part of the figure.

Table 4.1: Results of UAV Detection for *CBD*. The YOLOV5 outperforms all other SOTA object detectors.

Model	**mAP**$_{50}$(%) **Val**	**mAP**$_{50}$(%) **Test**
YOLOV5m	**95.19**	**95.6**
YOLOV8m	91.3	91.8
YOLO-NASs	83.47	88.17
Faster-RCNN	78.16	82.41
RetinaNet	80.93	84.39

Faster-RCNN YOLO-NASs RetinaNet

YOLOv5 YOLOv8

Figure 4.5: Detection results of *CBD*

4.3.2 Detection on *ANTD*

In this section, we present the results of UAV detection when additive white Gaussian noise (AWGN) is introduced to the test set. We evaluate three separate test sets, each with a distinct noise distribution ratio. The models' performances are shown in Table 4.2. YOLOv5m has the highest mAP_{50} test results for medium and high levels of noise among the models. While YOLO-NASs performed relatively close to Faster-RCNN at a low level of noise, they performed worse at medium and high levels of noise. When the noise increases, all models' performance declines.

Table 4.2: Performance Evaluation on *ANTD*. For low noise, Faster-RCNN however for medium and high range noise, YOLOv5 outperforms the other SOTA object detectors.

Model	**mAP**$_{50}$(%) **(low)**	**mAP**$_{50}$(%) **(medium)**	**mAP**$_{50}$(%) **(high)**
YOLOV5m	64.6	**55.2**	**45.5**
YOLOV8m	62	53.1	42.7
YOLO-NASs	64.75	44.91	31.70
Faster-RCNN	**66.70**	53.94	40.35
RetinaNet	62.62	51.99	42.27

4.3.3 Detection on *MBTD*

Three types of blurred data sets are built by varying the blur filter size, which effectively increases the amount of blur. A higher filter size yields a higher blur in the image. These datasets are tested and analyzed by selected state-of-the-art models. Objects with blurred edges and deformed shapes may be misclassified or missed. Overall, motion blur in images degrades the average precision of object detection models by degrading visual attributes and changing item appearances, as shown in Table 4.3. The higher the motion blur, the greater the degradation of mAP. Motion blur effects on the selected models are shown in Table 4.3.

Table 4.3: Performance Evaluation on MBTD. The YOLOV5 outperforms all other SOTA object detectors.

Model	**mAP**$_{50}$(%) **(low)**	**mAP**$_{50}$(%) **(medium)**	**mAP**$_{50}$(%) **(high)**
YOLOV5m	**93.4**	**65.6**	**21.9**
YOLOV8m	89.7	58.7	18.6
YOLO-NASs	64.75	53.36	19.21
Faster-RCNN	74.86	38.99	10.29
RetinaNet	75.57	37.91	12.86

4.3.4 Detection on *RTD*

Synthetic rain effects at three levels were added to the test set images to examine the influence of rain on UAV detection performance. As a consequence, three distinct test sets including drizzle, heavy, and torrential were built, each corresponding to a different level of rain intensity. When the rainy images were analyzed, there was a notable decrease in both Mean Average Precision (mAP) and Intersection over Union (IoU) scores. These findings emphasize the negative impacts of rain on the accuracy and resilience of UAV identification systems, as seen by the lower performance metrics. Table 4.4 indicates the performance degradation with an increased amount of rain. However, still, some models can detect the object but with lower confidence including YOLOV5, YOLOV8, and YOLO-NAS.

Table 4.4: Performance Evaluation on Rainy Test set

Model	**mAP$_{50}$(%) (drizzle)**	**mAP$_{50}$(%) (heavy)**	**mAP$_{50}$(%) (torrential)**
YOLOV5m	**66.8**	**65.1**	47.9
YOLOV8m	65.7	60.2	42.7
YOLO-NASs	53.88	53.42	**50.92**
Faster-RCNN	50.35	28.70	28.57
RetinaNet	44.96	42.30	35.69

Table 4.5: Performance comparison with clean vs. distorted, weather-affected data

Model	**mAP$_{50}$**	**mAP $_{50}$ Rainy**			**mAP$_{50}$ Noisy**			**mAP $_{50}$ Motion Blurred**		
	Clean	**Drizzle**	**Heavy**	**Tor**	**Low**	**Medium**	**High**	**Low**	**Medium**	**High**
YOLOV5m	95.6	66.8	65.1	47.9(−50.62)	64.6	55.2	45.5(−52.40)	93.4	65.6	21.9(−77.0)
YOLOV8m	91.8	65.7	60.2	42.7(−53.23)	62	53.1	42.7(−53.23)	89.7	58.7	18.6(−79.62)
YOLO-NASs	88.17	53.88	53.42	50.92(−42.22)	87.67	44.91	31.70(−64.0)	64.75	53.36	19.21(−78.21)
Faster-RCNN	82.41	50.35	28.70	28.57(−66.58)	66.70	53.94	40.35(−52.80)	74.86	38.99	10.29(−87.96)
Retina-Net	84.39	44.96	42.30	35.69(−57.77)	62.62	51.99	42.27(−49.99)	75.57	37.91	12.86(−84.76)

Table 4.4 showcases the Mean Average Precision (mAP) at an IoU threshold of

0.50, and performance degradation in percentage for different algorithms. The performance on the original, undistorted dataset is represented by the ”clean data” column. The columns that follow indicate how each method performs when rain, noise, and blurriness are introduced into the dataset. The last column depicts the proportion of performance deterioration experienced by each method when compared to the clean data. This degradation is estimated by comparing the mAP_{50} scores of the clean and other datasets respectively. The table shows that the addition of rain, noise, and blur typically results in a fall in mAP_{50} scores for all methods. The values in brackets $(-)$ show the performance degradation of the model from the clean data model. Among the models, YOLOv5m consistently exhibits the highest mAP on all types of distorted data, followed by YOLOv8m, YOLO-NASs, Faster-RCNN, and RetinaNet. On further analysis of the table, it can be observed that Faster-RCNN exhibits the highest performance degradation when tested on rainy and motion-blurred data. Conversely, YOLO-NASs demonstrate the most significant degradation in performance when evaluated on noisy data.

4.3.5 Detection Failures

In the realm of object detection, a situation arises known as missed detection, wherein an object within an image or video frame goes undetected by the object detection model in use. This occurrence can stem from a range of factors, such as subpar image quality, instances of object obstruction, inadequate training of the model, or scarcity of data. In contexts associated with Anti-Unmanned Aerial Vehicle (Anti-UAV) endeavors, the failure to detect objects can potentially give rise to security vulnerabilities.

Fig. 4.6 depicts the detection failure situation using the YOLOv8 model, illustrating cases when it could not properly detect particular items. It also depicts the YOLO-NAS model encountering detection failures, highlighting occasions when it was unable to properly recognize particular items inside the image. The detection failures with the RetinaNet model, show circumstances when the model failed to recognize specific objects with bounding boxes. Moreover, the detection failures of the YOLOv5 model are also illustrated in Fig. 4.6, indicating regions where it failed to differentiate individual objects. The Faster-RCNN did not show any miss detection however, it shows a lot of false positives which makes its mAP lower than YOLOv5. The overall images in Fig 4.6 give information on the performance restrictions of various object detection algorithms recognition algorithms by illustrating instances where they failed to identify UAVs.

YOLOv8 YOLO-NASs RetinaNet YOLOv5

Figure 4.6: Detection Failure of each model on *CBD*.

4.3.6 Enhancing Robustness of Selected Models

Based on the discussions and analysis presented above, it becomes evident that the model's generalization ability is limited when it comes to various types of adverse conditions. In instances where adverse conditions arise, the model's performance is notably compromised and tends to deteriorate significantly.

This serves as a motivation for us to augment our training CBD by incorporating different types of adverse weather effects. Specifically, we include 33% noisy images, 33% motion-blurred images, and 34% rainy images in our training data. Subsequently, We assess the performance of the top-performing models, YOLOv5, YOLOv8, and YOLO-NASs on the test sets we introduced earlier. By conducting this analysis, we assess how effectively the models can handle and adapt to adverse conditions. The performance evaluation of the YOLOv5, YOLOv8, and YOLO-NAS models using enhanced datasets incorporating rainy, noisy, and motion-blurred data is presented in Table 4.6. The table displays the Mean Average Precision (mAP) values for various circumstances. The table shows that the model obtains high mAP scores on clean data, indicating strong performance on undistorted images. However, if the data is influenced by rain, noise, or motion blur, the mAP scores decline, indicating the model's diminished ability to recognize objects under poor conditions. Our models performed better even under unfavorable settings and blurry images after including different adverse data in our training set. This implies that using such data during training improved the models' capability as shown in 4.7, and 4.8 to handle difficult conditions, resulting in more robust and accurate detection of objects.

Fig. 4.7 shows how the updated model performed on motion-blurred, noisy, and rainy images. The model effectively detects the UAV in motion-blurred images. Similarly, the model correctly recognizes the UAV in a noisy picture in the same Fig. Finally, Fig. 4.7 shows the model's ability to recognize the UAV in a rainy scene. These findings illustrate the improved models' capacity to detect UAVs accurately and under adverse conditions.

Table 4.6: Enhanced Robustness to Adverse Weather Conditions and Image Distortions of Object Detection Models. The YOLOv5 outperforms the other models.

Model	**mAP$_{50}$**	**mAP$_{50}$ Rainy**			**mAP$_{50}$ Noisy**			**mAP$_{50}$ MotionBlurred**		
	Clean data	**Drizzle**	**Heavy**	**Torrential**	**Low**	**Medium**	**High**	**Low**	**Medium**	**High**
YOLOv5m	95.6	66.8	65.1	47.9	64.6	55.2	45.5	93.4	65.6	21.9
YOLOV5m (Enhanced)	94.6	86	86.3	83.3	91.6	88.3	84.8	94.6	89.4	66.8
YOLOV8m	91.8	65.7	60.2	42.7	62	53.1	42.7	89.7	58.7	18.6
YOLOv8m (Enhanced)	91.1	80.0	81.7	77.7	88.7	84.7	82.0	91.3	85.1	63.0
YOLO-NASs	88.17	53.88	53.42	50.92	87.67	44.91	31.70	64.75	53.36	19.21
YOLO-NASs (Enhanced)	86.23	69.61	69.52	68.86	81.06	72.30	57.12	86.67	75.59	46.38
Faster-RCNN	82.41	50.35	28.70	28.57	66.70	53.94	40.35	74.86	38.99	10.29
Faster-RCNN (Enhanced)	80.18	66.45	65.13	64.50	77.62	72.76	64.53	80	75.02	47.34
RetinaNet	84.39	44.96	42.30	35.69	62.62	51.99	42.27	75.57	37.91	12.86
RetinaNet (Enhanced)	78.84	61.65	62.21	60.50	72.52	66.20	59.83	78.64	68.39	44.10

4.3.7 Grad-CAM-based Analysis

This section will evaluate the performance of the chosen object detection model against all proposed test sets using Gradient-weighted Class Activation Mapping (Grad-CAM) [102]. This is the most popular tool used by researchers to gain insight into the intent of object detection models and results. The GradCAM focuses on the part of the image that affects the model's accuracy. We looked at the images from the *CBD, RTD, MBTD* datasets, as well as the *ANTD* dataset, to see how well the four models performed when it came to UAV detection. The results of the GradCAM for all models in different datasets are shown in Fig. 4.9. YOLOv5 had the best class score out of all datasets. Fast-RCNN also had a good class score out of the dataset of the same name *CBD*. However, for the datasets of *MBTD and RTD*, there is a clear case of

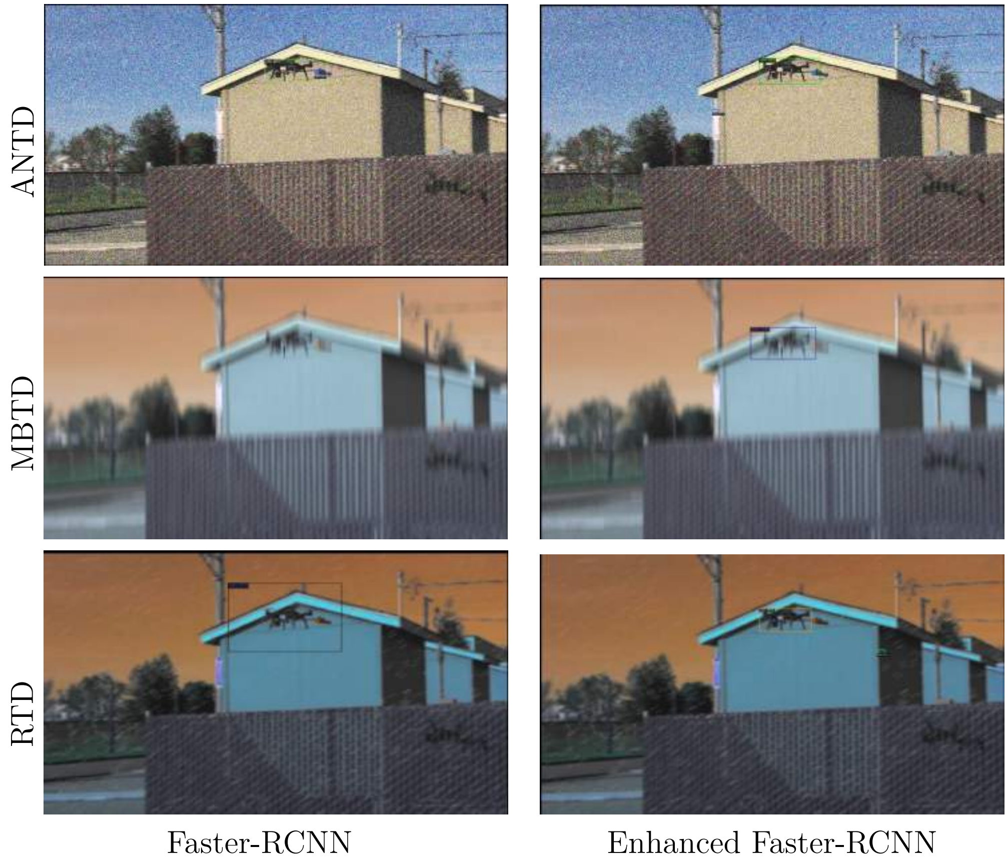
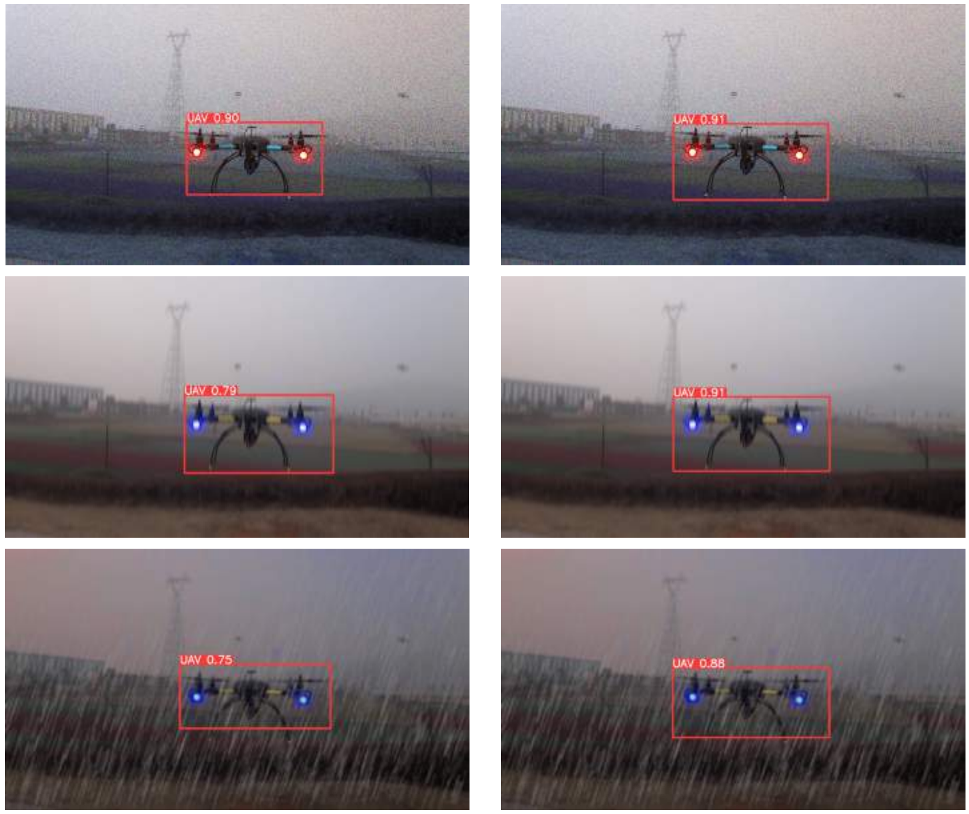


Figure 4.7: Detection Result and Improvement comparison of Faster-RCNN and Enhanced Faster-RCNN model. The Enhanced version clearly achieved higher detection results. For *MBTD* the original Faster-RCNN fails to detect the UAV.

misclassification as shown by the model's concentration in the figure. In addition, the YOLOv8 model demonstrated good overall class scores for the following categories: *CBD, RTD*. But, for *ANTD*, the model's focus is not on the UAV, resulting in a low average accuracy for the model.

4.3.8 Execution Time Analysis

This section examines the computing efficiency of the object detection models under consideration. it demonstrates the practicality of deploying these models in real-world applications by measuring the time it takes each model during the training and



ANTD

MBTD

RTD

YOLOv5 | Enhanced YOLOv5

Figure 4.8: Detection Result and Improvement comparison of YOLOv5 and Enhanced YOLOv5 model. The Enhanced version achieved higher detection results.

testing phase to evaluate a given dataset of image or video frames. Execution times were tested using a consistent hardware and software environment, including NVIDIA A4500 GPU for 100 epochs. Table 4.7 compares training and inference times for several object detection models. The training execution performance of each model is measured in terms of (iterations per second (iter/sec)), and testing execution such as inference speed is measured in (milliseconds (ms). YOLOv8 has a training time of 3.64 iterations per second, which indicates how rapidly it can be trained on the dataset. Furthermore, it has an inference time of 9.5 milliseconds. YOLOv5, on the other hand, has a somewhat higher training rate of 4.54 iterations per second. Surprisingly, this enhanced training efficiency does not come at the sacrifice of inference speed,

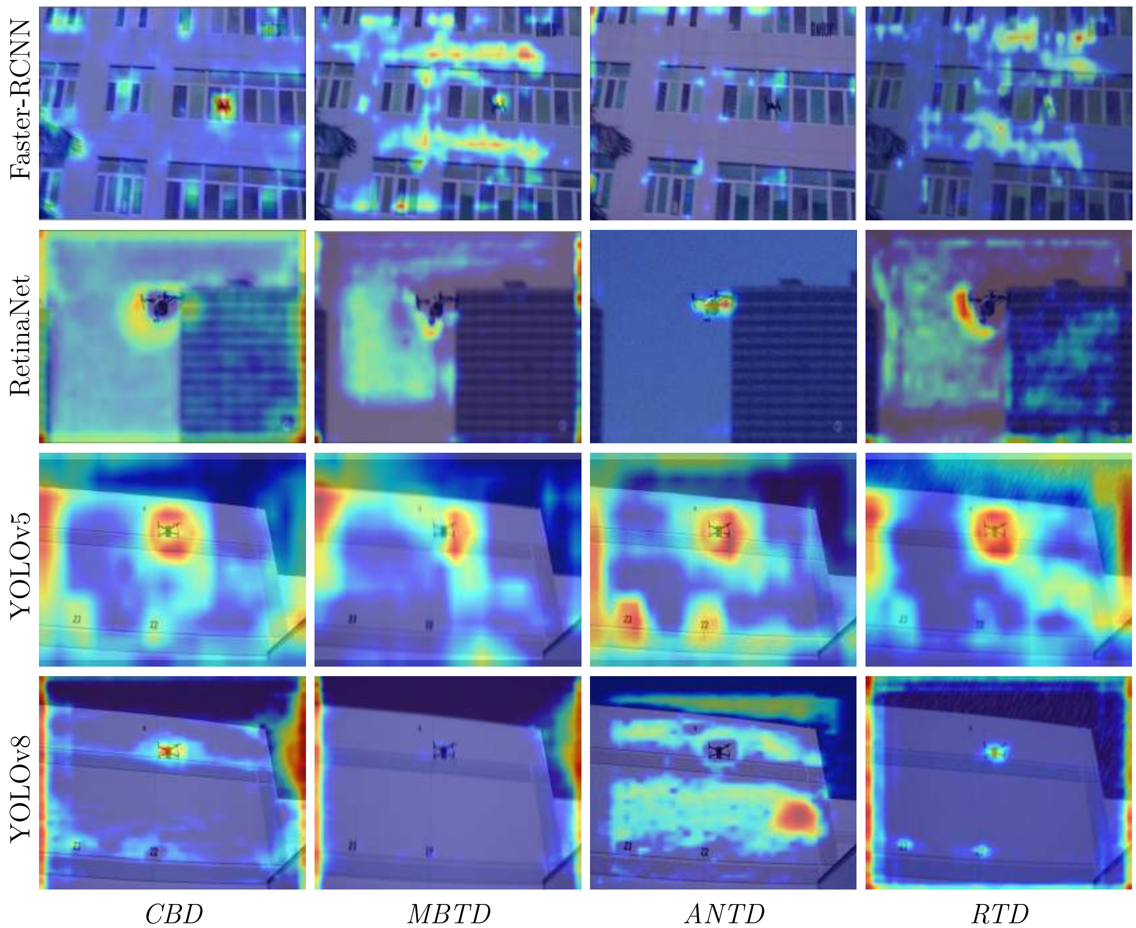


Figure 4.9: Sample GradCAM results are shown with all proposed datasets. The YOLOv5 shows high scores for the UAV class for all the mentioned datasets. However, for *RTD* the YOLOv5 focus is toward UAV and the edges of the image which can cause a False positive.

as YOLOv5 reaches an inference time of 7.1 milliseconds, demonstrating its ability to recognize objects quickly. Although Faster-RCNN has a slower training rate of 0.30 iterations per second, it has a longer inference time of 73.38 milliseconds. This disparity highlights the model's accuracy while also emphasizing its greater processing costs during the inference phase. With 0.18 iterations per second, RetinaNet falls behind in training efficiency in terms of training time, also there is a cost in terms of inference time since it takes 153.18 milliseconds to process.

Table 4.7: Execution time of different Models

Model	**Training time (itr/sec)**	**Inference time (ms)**
YOLOv8	3.64	9.5
YOLOv5	4.54	7.1
Faster-RCNN	0.30	73.38
RetinaNet	0.18	153.18
YOLO-NAS	3.92	44.158

CHAPTER 5

YOLO-RAW: ADVANCING UAV DETECTION ROBUSTNESS IN VARIED WEATHER CONDITIONS

5.1 Proposed Model

In this section, we first provide an architectural overview of the proposed *YOLO-RAW* model inspired by YOLOv5 and enhanced for *R*obustness to *A*dverse *W*eather. Subsequently, we highlight and explain the enhancements proposed in this work to address the challenges associated with robust UAV detection in adverse weather effects.

YOLO-RAW's Architecture

The key modules in the YOLO-RAW's architecture are (i) Backbone, (ii) SPP-

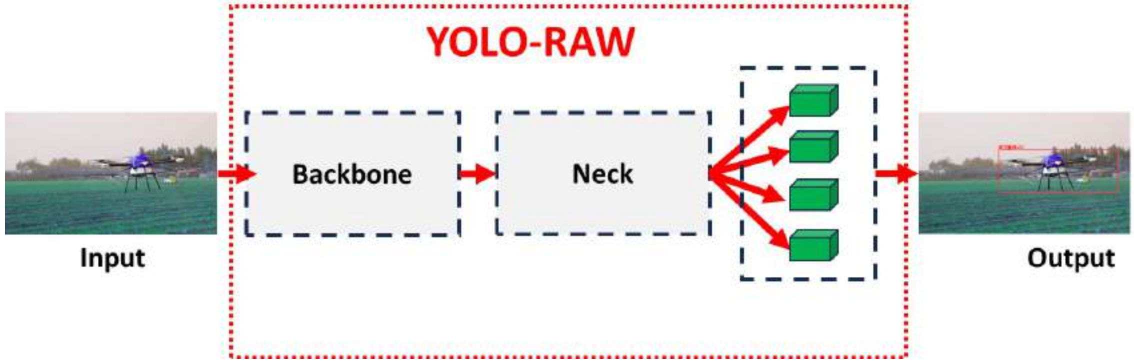


Figure 5.1: The block diagram of the proposed method.

Extended, (iii) Scale-blended Feature Aggregation Head, (iv) Parameter-free Attention Module, (v) Squeeze-and-Excitation Network (SENet) which shall be described below. The block diagram of the proposed method is shown in Fig.5.1.

Backbone

The backbone is mainly used to extract the features from the input image. The architecture of the proposed backbone is shown in Fig. 5.2. It improves on the YOLOv5 [98] model. The original YOLOv5 employs the architectural components of CSPDarknet53 as the backbone, incorporates an SPP layer, utilizes PANet as the neck, and features YOLO [103] as the detection head. The proposed backbone adds one CBS (Conv+BN+SILU) and one C3 block to get refined features from different scales. These features are passed to SPP-Extended (SPP-E).

SPP-Extended

In YOLOv5, the SPP component is composed of three branches, each with grid sizes of 5x5, 9x9, and 13x13 respectively. The original YOLOv5 failed to detect large-

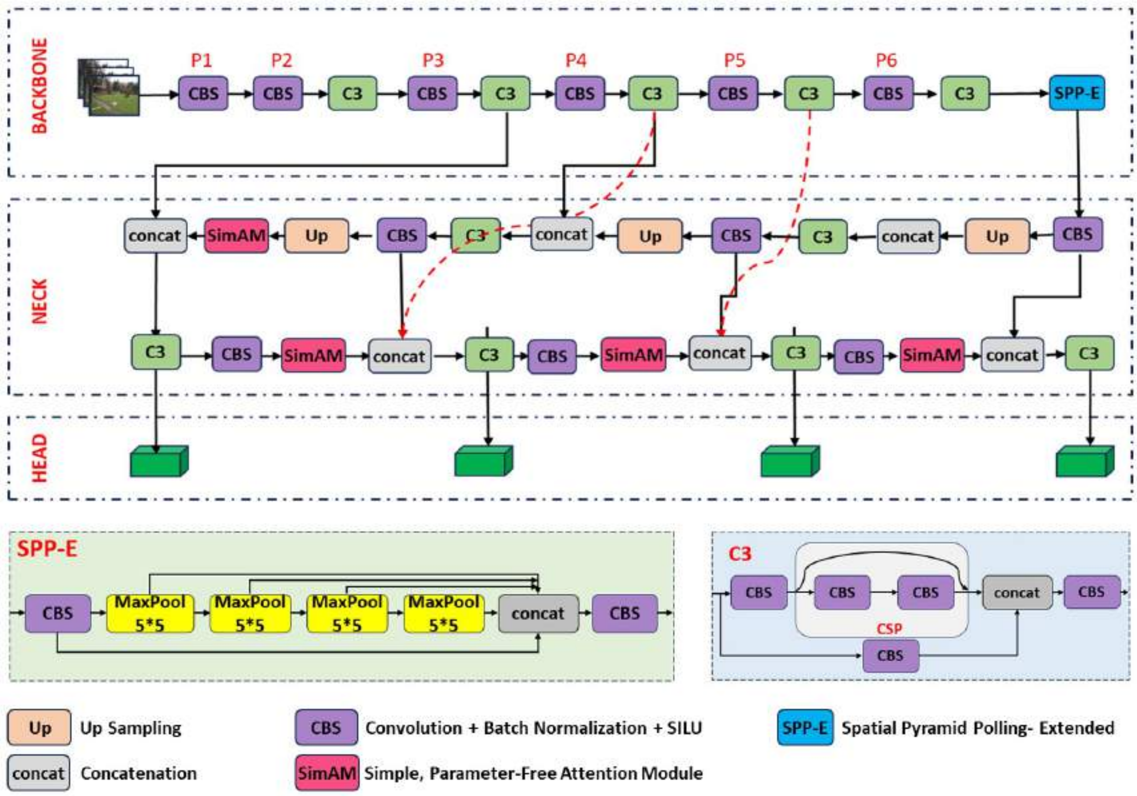


Figure 5.2: The architecture of the YOLOv5-RAW. a) CSPDarknet53 backbone with one extra CBS and C3 block, followed by SPPF-Enhanced at the end. b) The Neck uses a structure like PANet. The four SimAM (attention mechanism) are added to find a region of interest in the image. c) Four prediction heads are used in the YOLO head.

scale objects in our custom dataset so we have added one more grid size of 17x17 for capturing more context which is shown in Fig. 5.3. The information from these four branches will be combined into a single branch, and this combined feature data will be passed on to the subsequent layer of the network.

Neck with Scale-blended Feature Aggregation

The proposed Scale-blended Feature Aggregation with neck includes enhancements to account for UAVs of various scales. The proposed neck is an extended version of PANet [104], which closely relates to BiFPN [105]. The proposed neck architecture [104] is an enhanced version of FPN [22] architecture. Within the neck architecture, there are five feature fusion stages P1, P2, P3, P4, and P5 to integrate intermediate information obtained from the network in original YOLOV5 as shown in Fig. 5.2.

In our prior work of [106], performance degradation of YOLOv5 was observed for larger-scale UAVs in both clean and weather-affected images. So, the proposed model adds feature fusion stage labeled as P6. The original YOLOv5 network merges the feature maps C3 and C4 from the backbone with the up-sampled features from the neck of feature maps P3, and P4. However, C5 after passing from the SPP-E layer is merged with the same level feature map to P5 in the neck. Unlike YOLOv5, in addition to the fusion of C3 and C4 with the respective layers of the neck of feature

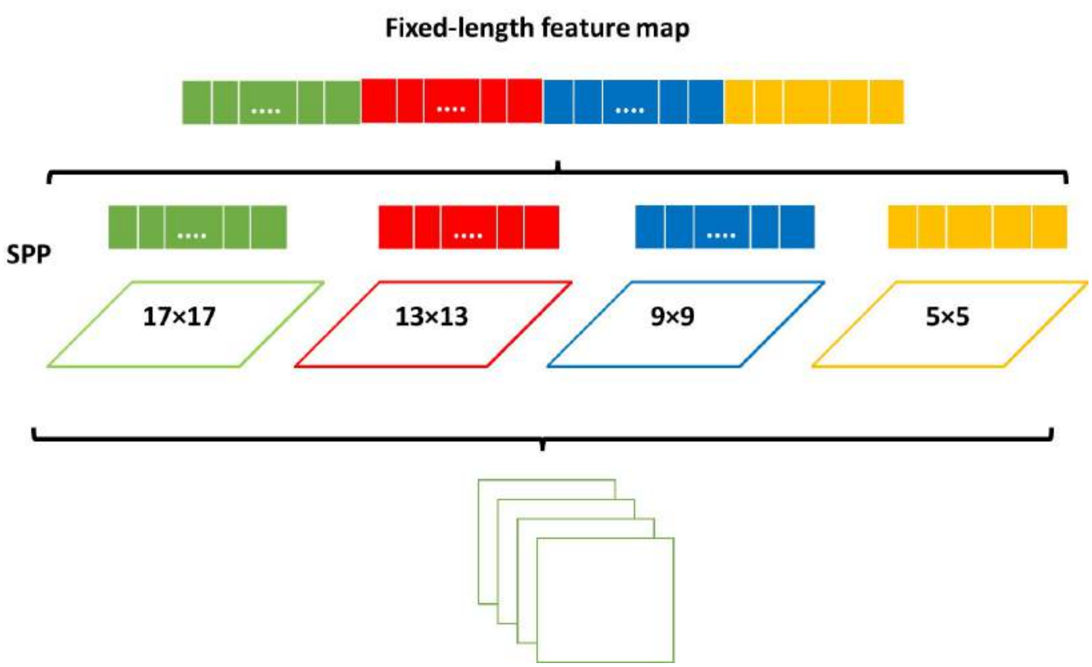


Figure 5.3: The architecture of SPP-Extended.

maps P3, and P4, the proposed model fuses the deepest feature map, i.e., C6, with the up-sampled features P6 of the neck that is coming from P5 to get the low resolution and richer semantic information map.

The proposed enhancements to the original neck structure of YOLOv5 are depicted in Fig 5.4. In the proposed model, P6 feature maps are obtained by merging feature maps from SPP-Extended and those from P5. The final network structure has four output scales. To improve overall model precision, the fourth detection head for larger-scale objects was also added. Section 5.4.2 showed the importance of an extra added head to the model. To enhance the richness of fusion features and improve the prediction accuracy for medium and large-scale objects, we incorporate the spatial information from lowest-level feature maps P4 and P5 into BiFPN [105] that is shown in Fig. 5.2 with dashed red lines.

The scale sizes of P3, P4, P5, and P6 feature maps are 80×80, 40×40, 20×20, and 10×10 respectively, and used as output to feed into the detection Heads. To optimize the detection of objects, a feature scale of 80×80 was found to be most effective for smaller objects. In the case of medium-sized objects, feature scales of 40×40 and 20×20 provide superior results, while for larger objects, feature scales of 10×10 are more suitable. However, the 100×100 or greater feature scales could be used to improve the precision for tiny objects. These four 80×80, 40×40, 20×20, and 10×10 different scales are associated with the grid sizes, and inspired by [107], we have used different anchor sizes for each scale to effectively predict objects of various sizes. The authors in [107] proposed an auto-anchor mechanism by using the "EIOU K-means" [108] algorithm which is slightly different than that of the original

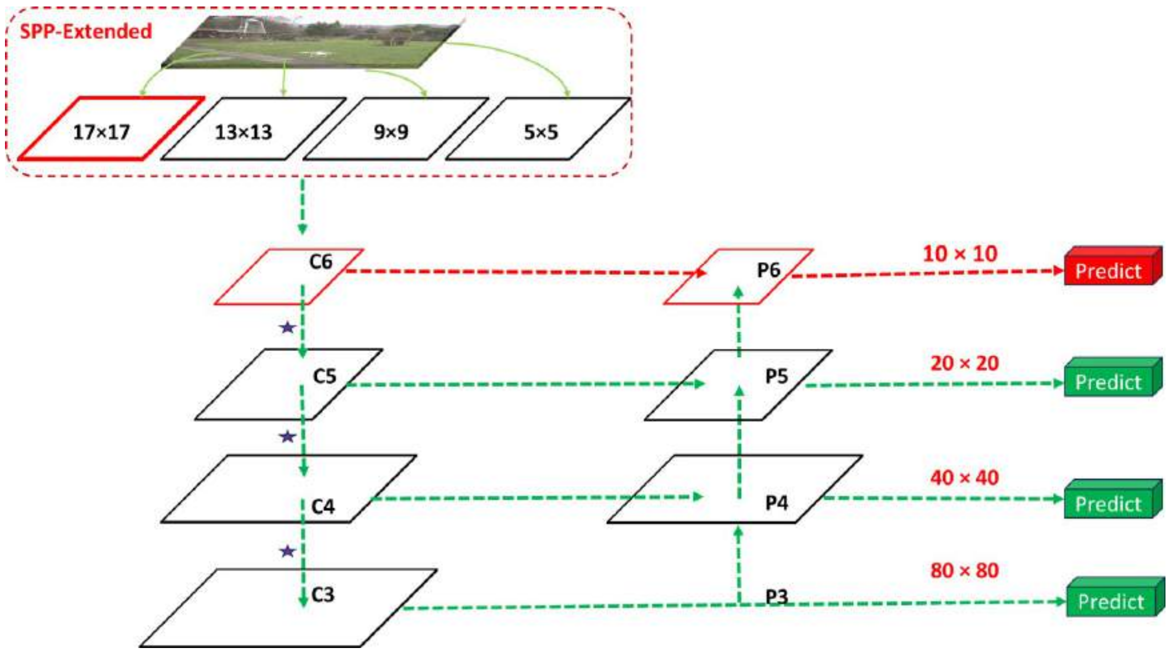


Figure 5.4: The architecture of the improved Neck (PANet). The Feature maps shown with red outlines are added to the YOLOv5 PANet structure to get the fourth scale for larger objects. The P6 is achieved by merging the P5 feature map with the feature map coming from the bottom-up approach after the SPP-Extended module. The $\star$ is representing the up-sampling.

YOLOv5 [98] auto-anchor mechanism.

Historically, experienced researchers have designed anchors during the development of two-step object detection, particularly with the anchor-based approach. The Autoanchor mechanism [98] introduced by YOLOv5 automates the setting of anchor sizes by computational calculations instead of manual specification. The mechanism becomes operational when the initially selected anchor sizes are considered unsuitable, resulting in the generation of a set of new anchor sizes to replace the initial ones. It emphasizes the need to adapt anchor sizes to particular datasets, taking into account the specific characteristics of different types and sizes of objects in different datasets. In YOLOv5, the Autoanchor mechanism is implemented through the k-means clustering [108] algorithm. This algorithm is designed to group samples with close distance to each other into the same category by evaluating the similarity between samples. The anchor sizes used in the work for each scale are shown in Table 5.1. The addition of a fourth prediction head beside the other three heads provides a four-headed detection system that assists in mitigating the adverse effects of scale variations within our data set.

Table 5.1: The Feature maps and their relative anchor sizes.

Module	**Feature Map**	**Anchor**
P3	80×80	[10,13, 16,30, 33,23]
P4	40×40	[30,61, 62,45, 59,119]
P5	20×20	[116,90, 156,198, 373,326]
P6	10×10	[153,103, 186,213, 458,426]

**Anchor*=$[x_{min}, y_{min}, x_{max}, y_{max}, x_{centre}, y_{centre}]$

Parameter Free Attention Module

In the proposed Neck architecture (see Fig. 5.2), parameter-free attention modules are added to achieve robustness to adverse weather artifacts and obtain refined features. It is essential to force the model to focus on important areas of an image that contain objects of interest (UAVs in our case) due to the large image area and the corruption caused by adverse weather artifacts.

Adding a fourth scale in SPP-Extended may dilute the features of small objects which could in turn lead to a lower overall precision for small objects (UAVs). Attention modules are often utilized in deep learning models to enhance the extraction of important features and to focus on relevant features for better detection of target objects [109]. However, most of the existing attention modules including the Convolutional Block Attention Module (CBAM) [110] typically assign weights to the channel dimension to improve model performance, which necessitates additional parameters, increasing the model complexity.

To avoid an increase in model complexity, this work employs the parameter-free *SimAM* [5] method which is a feature-based attention mechanism that does not have any parameters, allowing for the ability to adjust 3D attention weights for feature maps to enhance the model's ability to extract features. The purpose of the SimAM attention mechanism is to identify significant neurons by evaluating the degree of linear separation between them and to grant higher priority to these particular neurons. In the proposed model, the SimAM attention module is employed in the Neck architecture as shown in Fig. 5.2 to get the fine-grained feature maps and to improve the precision of the model for clean and adverse weather data. SimAM increases the proposed YOLO-RAW's robustness to adverse weather artifacts as will be illustrated in Section VII-A 5.4.1.

SimAM Details: An energy function is used in SimAM to differentiate the importance of individual neurons and to facilitate the implementation of the attention mechanism. This function indicates the degree of linear separation between a neuron, referred to as 't', and all other neurons in the same channel. For each neuron, the energy function is as follows [5]:

$$\begin{aligned} e_t(a,b,y,c) = \frac{1}{N-1} \sum_{i=1}^{N-1} \left(-1 - (\omega_t c_i + b_t)^2\right) \\ + \left(1 - (\omega_t t + b_t)^2\right) + \alpha a_t^2 \end{aligned} \tag{5.1}$$

where the c_i and t are the targeted neurons, weights, and biases represented by ω_t, and b_t respectively, those are the linear transformations of t. The spatial dimension is represented by the index i, the hyper-parameter is represented by α, and $N = HW$, which represents the number of neurons in a single channel. The weights and biases are as follows

$$\omega_t = \frac{2\left(t - \mu_t\right)}{\left(t - \mu_t\right)^2 + 2\sigma_t^2 + 2\alpha} \tag{5.2}$$

$$\mathrm{b_t} = -\frac{1}{2}\left(t + \mu_t\right)\omega_t \tag{5.3}$$

here, the mean and variance of all neurons are represented by μ_t, and σ_t^2.

$$\mu_t = \frac{1}{N-1}\sum_{i=1}^{N-1} c_i \tag{5.4}$$

$$\sigma_{\mathrm{t}}^2 = \frac{1}{N-1}\sum_{i=1}^{N-1}\left(c_i - \mu_t\right)^2 \tag{5.5}$$

By deriving the analytical solutions for ω_t, b_t and computing the mean and variance of all neurons within the channel, the minimal energy equation is as follows [5]: According to Equation 5.6, with lower energy $\mathrm{e_t^*}$, the neuron t is more easily differentiated from its neighbors, thus emphasizing its greater importance in visual processing.

$$\mathrm{e_t^*} = \frac{4\left(\hat{\sigma}^2 + \alpha\right)}{(t - \hat{\mu})^2 + 2\hat{\sigma}^2 + 2\alpha} \tag{5.6}$$

where $\hat{\mu}$ and $\hat{\sigma}^2$ stand for mean and variance:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1} c_i \tag{5.7}$$

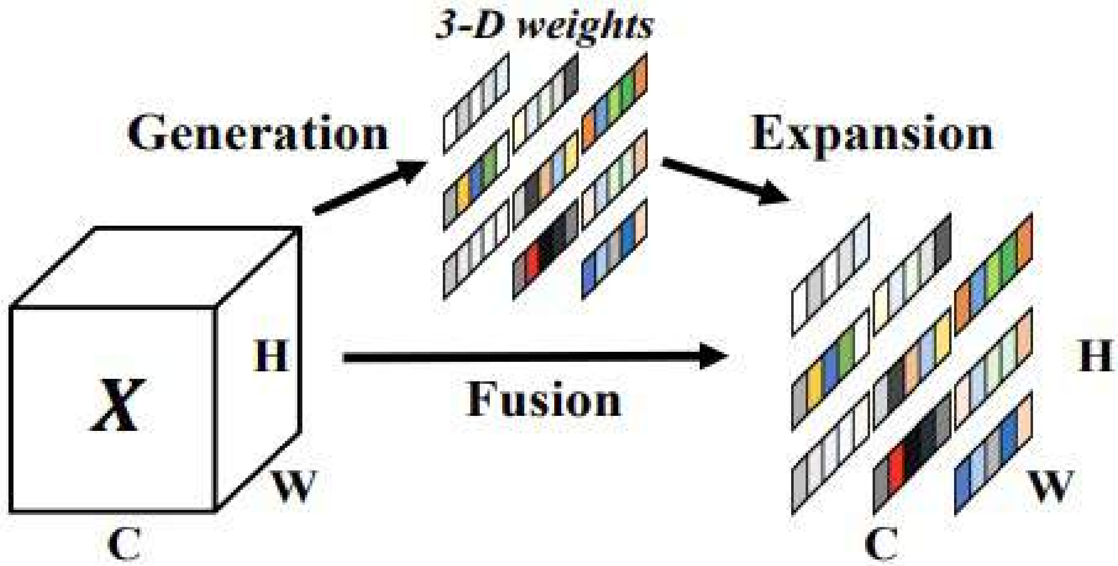


Figure 5.5: SimAM, 3-D weights for attention from feature **X**[5]. Identical color signifies that a single scalar is applied for each channel, spatial location, or individual point on that feature.

$$\hat{\sigma}^2 = \frac{1}{N}\sum_{i=1}^{N}(c_i - \hat{\mu})^2 \tag{5.8}$$

As expressed in equation 5.6, a lower value of the energy function signifies a stronger linear separation between neuron t and other neurons. The complete attention module operates based on this energy function, avoiding the need for excessive heuristic rules and adjustments. Employing full 3D attention weights was found to be superior to traditional 1D and 2D attentions [5]. SimAM enhances the features using full 3-D attention weights, as illustrated in Fig. 5.5. The implementation detail of SimAm is shown in Fig. 5.1.

In this work, we introduced the SimAM attention module in the neck at four points to extract the features efficiently from the affected adverse weather images without adding any parameters to our original model. The outcomes depicted in Fig. 5.7

```python
# Y: input feature [N, C, H, W]
def forward(Y, alpha):
    # Calculate the spatial size
    m = Y.shape[2] * Y.shape[3] - 1

    # Calculate the square of (p - q)
    e = (Y - Y.mean(dim=[2, 3])).pow(2)

    # e.sum() / m is channel variance
    r = e.sum(dim=[2, 3]) / m

    # Z_inv groups all the importance of Y
    Z_inv = e / (4 * (r + alpha)) + 0.5

    # Return attended features
    return Z_inv
```

The implementation details of SimAM attention.

and the findings outlined in Section 5.2 collectively affirm the effectiveness of our enhancements.

Squeeze-and-Excitation Networks (SENet)

The combination of squeeze and excitation makes up the SENET [111] attention mechanism. The SENet is mainly based on a channel-wise attention mechanism (CAM). SENet is designed to draw attention to the differences between different channels within feature maps and adjust the attention weights accordingly. The network diagram of SENet is shown in Fig. 5.6. In the section 5.4.2 on the ablation studies, the SENet will be employed. We will use SENet in place of the SimAM module and evaluate the effectiveness of the two attention techniques.

5.2 Experimental Setup

In this section, we will outline the specific details of our experimental implementation. We will also provide an overview of the evaluation criteria and datasets used in this work. Additionally, we will conduct tests and ablation studies to evaluate the effectiveness of our proposed improvements and compare them to alternative approaches. Lastly, we will provide an analysis of the effectiveness, complexity, and limitations of our proposed methodology.

5.2.1 Experimental Tools and Framework

In this subsection, we will describe the details and used tools of our work. The publicly available library PyTorch [101] and Ultralytics [44] are used for the programming. The proposed network is based on YOLOv5, so annotations are used in COCO [38]

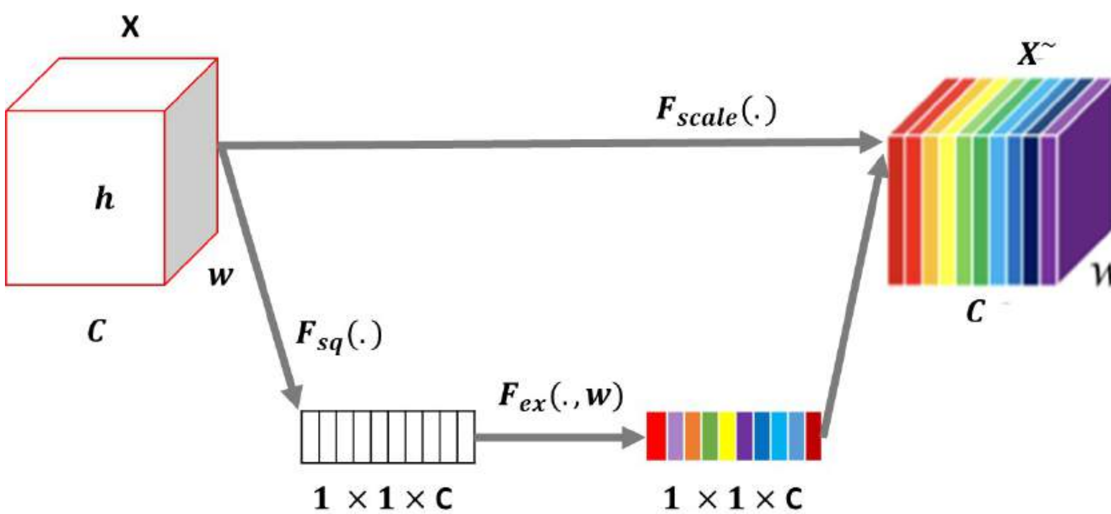


Figure 5.6: The architecture diagram of the Squeeze-and-Excitation Networks (SENet)

.

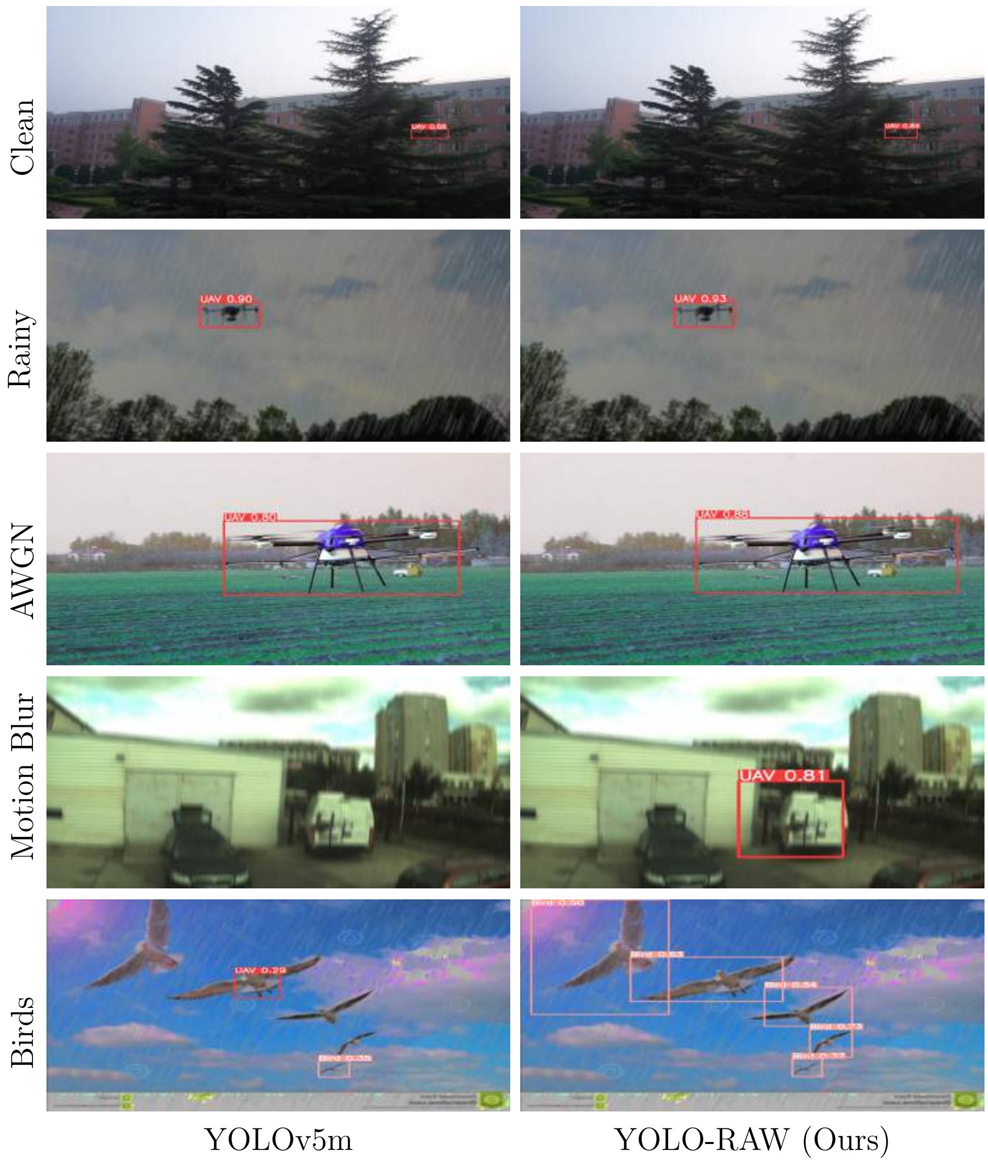


Figure 5.7: We showcase sample images from all test sets. The original YOLOv5m results are displayed on the left, while our proposed model YOLO-RAW results are presented on the right. YOLO-RAW outperforms YOLOv5m, especially in Motion Blur images where YOLOv5m fails to detect, whereas YOLO-RAW confidently identifies the objects. Additionally, YOLOv5 has challenges detecting birds with some false positives, whereas YOLO-RAW minimizes false positives and accurately detects all birds.

format. The training and evaluation of YOLO-RAW were conducted on an NVIDIA RTX A4500 GPU. Each model underwent training for 200 epochs, employing an initial learning rate of 0.01 and a batch size of 16. The proposed model utilized pre-trained weights from the COCO dataset [38]. The learning rate schedule here is based on a step decay method, which is similar to the cosine annealing method used to optimize learning rates. Input images are set to 640 × 640, the NMS IoU is set to 0.5 and the prediction box confidence is set to 1e-3. All other parameters stick to their default settings.

5.2.2 Evaluation Metrics

The detection performance of the proposed YOLO-RAW architecture is evaluated using standard evaluation metrics such as Mean Average Precision (mAP) as discussed in 4.2.2, and Intersection over Union (IoU) discussed in 4.2.2, Precision (P), and Recall (R). These metrics are as follows:

Precision (P): is determined by dividing the instances correctly predicted as positive by the total number of items identified as positive. Precision is calculated using the following formula:

$$P = \frac{TP}{TP + FP}$$

Recall (R): gauges the model's proficiency in identifying positive samples and is computed by dividing the count of correctly classified positive samples over the total number of positive samples. An elevation in recall signifies an enhanced rate of positive samples being recognized.

$$R = \frac{TP}{TP + FN}$$

where, TP, TN, FP, and FN are true positive, true negative, false positive, and false negative respectively. The TP refers to an actual UAV instance being detected by a method.

Inference Time: the inference time in object detection refers to the duration it takes for the model to process and analyze input images and generate predictions.

5.3 Results and Discussion

To assess the detection capabilities of our proposed YOLO-RAW model, we conducted extensive experiments by training (and validation) on CBD and testing one of the proposed test sets TS, RTD, MBTD, ANTD, and Multi-Weather Test Set (described in Sec 3). Moreover, we compare the performance of YOLO-RAW against that of YOLOv5 [98], YOLOv8 [94], Faster-RCNN [21], and RetinaNet [112]. So, here we will explain the results for the proposed model, and other models with the proposed dataset separately in their respective subsections.

5.3.1 Results on CBD

The proposed model YOLO-RAW performed well in medium and large-scale UAVs. The results of all four models are compared with our proposed model, as shown in Table 5.3. Our proposed model is performing overall well for the validation set, however for TS YOLOv5 performs better, and the reason is that TS has more tiny objects

Table 5.2: UAV sizes of each category.

Category	Tiny	Small	Medium	Large
UAV size ($L \times W$)	13.48×11.52	20×23.30	30.97×53.31	420×268

Figure 5.8: Number of images in Test Set (TS) according to their sizes.

than the validation set. The brackets (+) show the improvement of the model with percentage points to the YOLOv5. The TS has only fewer small and large-scale UAVs as it is shown in Fig 5.8. The distribution of the dataset into four categories including tiny, small, medium, and large is based on the area of UAVs present in the image. The cut-offs of all categories are shown in Table 5.2. A UAV of length less than 20 will be considered a tiny UAV, similarly, a UAV of length less than 420 and greater than 30 will be considered a medium UAV.

To get more insights into our proposed model and how well it performs on different

Table 5.3: The performance comparison of various object detection models for UAV detection on the CBD's Validation Set and on the Test Set (TS) for testing. YOLOv5m shows the best performance among others.

	Validation		**Test**	
Model	$\mathbf{mAP_{50}}$	$\mathbf{mAP_{50-95}}$	$\mathbf{mAP_{50}}$	$\mathbf{mAP_{50-95}}$
YOLOv5m	95	56.9	**94.9**	**59**
YOLOv8m	91.3	**58.1**	91.8	51
Faster-RCNN	78.04	41.34	82	46
RetinaNet	80.85	44.39	84.39	46.75
YOLO-RAW (Ours)	**95.3 (+0.3)**	**57.4 (+0.5)**	94.6	57.1

scales of UAVs, we divided the TS into four different categories i.e, TS_{tiny}, TS_{small}, TS_{medium}, and TS_{large} according to the size of UAVs, and tested YOLOv5 and our proposed model. The performance of YOLOv5 and YOLO-RAW of each category is shown in Table 5.4. YOLO-RAW outperformed YOLOV5 for medium and large-scale UAVs and showed comparable performance for smaller UAVs. The overall detection performance of the proposed model YOLO-RAW and YOLOv5 of each scale UAV is shown in Fig. 5.9.

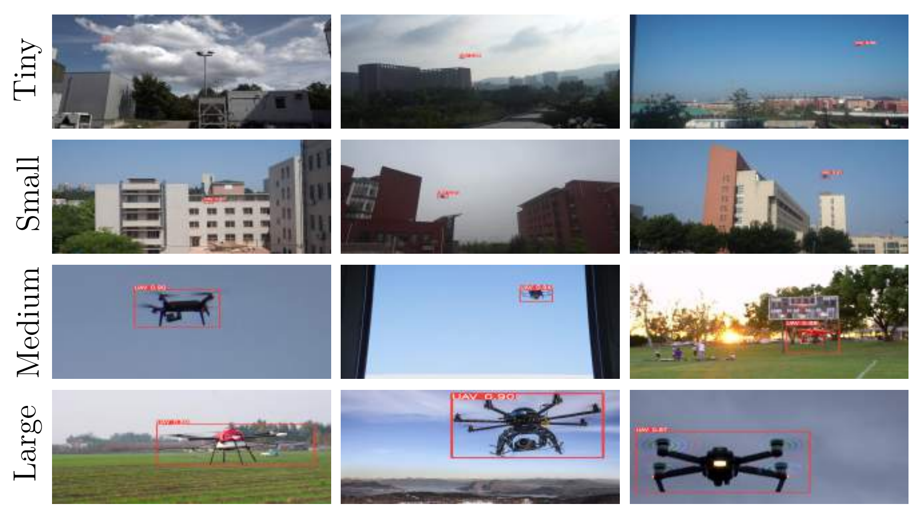


Figure 5.9: Detection results of the proposed YOLO-RAW model for UAVs at four different size categories.

Table 5.4: The performance comparison of YOLOV5, and YOLO-RAW (Ours) models for UAV detection on the *Test Set (TS)* four categories TS_{tiny}, TS_{small}, TS_{medium}, and TS_{large}. YOLO-RAW (Ours) performed better on medium and Large scale objects however it is comparable performance on tiny and small scales.

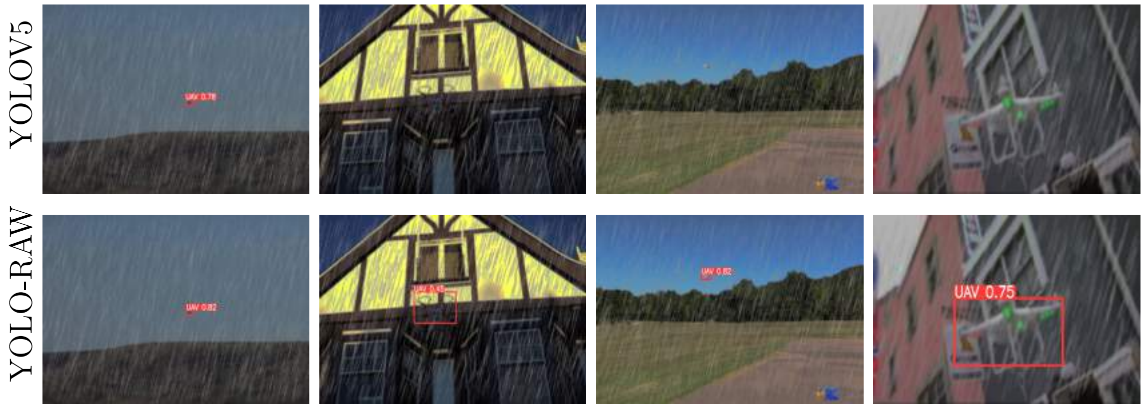
Models	Test Set (TS)											
	Mean Average Precision (mAP)				Precision (P)				Recall (R)			
	Tiny	Small	Medium	Large	Tiny	Small	Medium	Large	Tiny	Small	Medium	Large
YOLOv5	**93**	**93.8**	95.4	98	92.8	**90**	93.5	94	**91**	**92.2**	92.4	97.7
YOLO-RAW (Ours)	92.7	92	**95.8**	**99.5**	**93**	89.5	**93.7**	**98**	90	89	**93**	**100**

Table 5.5: Summary of the performance of proposed model and other object detection models for UAV detection on the three *Rainy Test Sets (RTD)*: $RTD_{drizzle}$, RTD_{heavy} and RTD_{tor} vs. the performance on clean TS images (no rain).

Model		$\mathbf{mAP_{50}}$		
	drizzle	**heavy**	**torrential**	**Inference time (ms)**
YOLOv5m	**64**	**61**	**53.5**	**2.6**
YOLOv8m	65.7	60.2	42.7	7.5
Faster-RCNN	50.35	45.25	37.40	41.3
RetinaNet	46.33	43.69	35.57	29.2
YOLO-RAW (Ours)	61.4	59	**54.6 (+1.1)**	3.9

5.3.2 Results on RTD

The outcomes of our model, YOLO-RAW, and the other four models are presented in Table 5.5. Our proposed model outperforms the others, especially as the amount of rain in the images increases. It's worth noting that while the inference time of our proposed model is slightly lower than YOLOv5, the overall detection results have improved by 1.1% compared to YOLOv5. Fig. 5.10 illustrates the detection of UAVs in images. YOLO-RAW, our proposed model, exhibits promising performance in detecting UAVs in rainy images, whereas YOLOv5 struggles with such conditions.



(a) (b) (c) (d)

Figure 5.10: Sample images from the proposed *Rainy Test Set (RTD)*. Our Model YOLO-RAW outperforms the YOLOv5 as the amount of rain increases in images. (a) Our model shows higher confidence than YOLOv5, In (b), (c), and (d) YOLOv5 misses detect the UAV however our model detects the UAV with higher confidence.

5.3.3 Results on ANTD

We conducted tests on our proposed *AWGN test set (ANTD)* which includes three categories: $ANTD_{low}$, $ANTD_{medium}$, and $ANTD_{high}$. The results for each category are detailed in Table 5.6. Notably, our proposed model outperformed all other models, particularly as the noise level in the images increased. This performance can be attributed to the presence of the SimAM attention module in our model, which refines the feature maps and directs the model's focus toward the UAV in an image. Fig. 5.11 presents sample detection results of our proposed model in comparison to YOLOv5. In Fig. 5.11 (d), it is evident that YOLOv5 produces false positives, whereas YOLO-RAW detects UAVs with higher confidence and avoids generating any false positives.

Table 5.6: Summary of the performance of the proposed model and other object detection models for UAV detection on the three *AWGN Test Sets (ANTD)*: $ANTD_{low}$, $ANTD_{medium}$ and $ANTD_{high}$ vs. the performance on clean TS images.

Model	mAP$_{50}$ Low	mAP$_{50}$ Medium	mAP$_{50}$ High	Inference time (ms)
YOLOv5m	63.9	56	47.4	**2.6**
YOLOv8m	62	53.1	42.7	7.5
Faster-RCNN	61.65	53.94	40.35	41.3
RetinaNet	62.62	51.99	42.27	29.2
YOLO-RAW (Ours)	**67.7 (+3.8)**	**57.4 (+1.4)**	**49.2 (+1.8)**	3.9

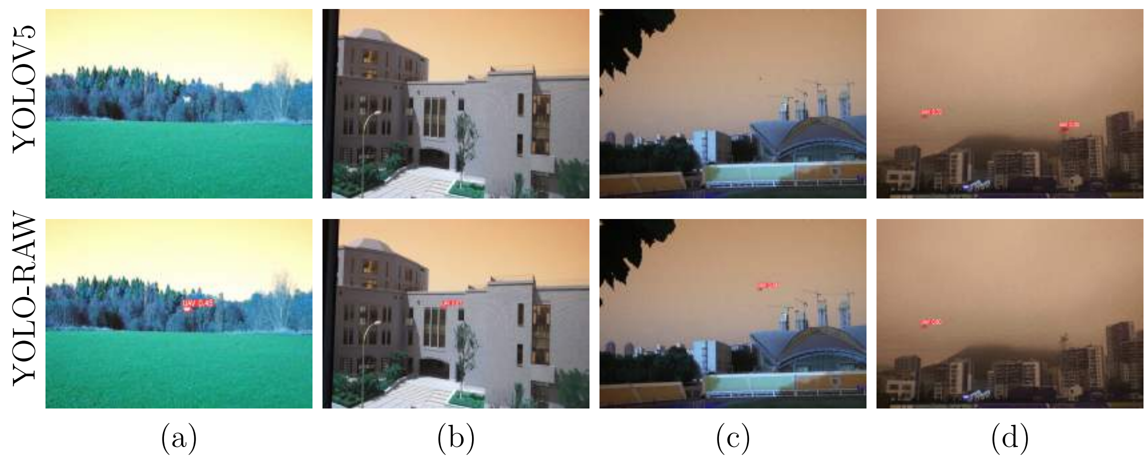


Figure 5.11: Sample images from the proposed *AWGN Test Set (ANTD)*. Our Model YOLO-RAW outperforms the YOLOv5 as the amount of noise increases in images. In (a), (b), and (c) our proposed model YOLO-RAW detects the UAV with higher confidence and YOLOv5 fails to detect. In (d) YOLOv5 detects the UAV but also generates False Positive however YOLO-RAW detects UAV with higher confidence.

5.3.4 Results on MBTD

Motion Blurred test set (MBTD) also consists of three types of datasets with low, medium, and high blurriness effects on images. These proposed test sets are evaluated on the proposed model YOLO-RAW, and the other four selected models. Table 5.7 presents the mAP of the proposed model and other models. For medium-size blurriness, our proposed model gets 0.3 percentage points better performance than YOLOv5.

Table 5.7: Summary of the performance of the proposed model and other object detection models for UAV detection on the three Motion Blurred Test Sets *(MBTD)*: $MBTD_{low}$, $MBTD_{medium}$ and $MBTD_{high}$.

Model	**Low**	**mAP$_{50}$** **Medium**	**High**	**Inference time (ms)**
YOLOv5m	**93.4**	62.7	**19.2**	**2.6**
YOLOv8m	89.7	58.7	18.6	7.5
Faster-RCNN	74.86	38.99	10.29	41.3
RetinaNet	75.57	37.91	12.86	29.2
YOLO-RAW (Ours)	92.8	**63 (+0.3)**	17	3.9

5.3.5 Results on MWTS

As we've observed in previous sections, our proposed model consistently outperformed all other selected models for each dataset we introduced. YOLOv5 exhibited the next-best performance. Therefore, for this particular test set, we will conduct a comparative analysis between YOLO-RAW and YOLOv5 models. You can find the results of this comparison in Table 5.8. Notably, YOLO-RAW surpasses the performance of the YOLOv5 model. The mAP is increased by 5.4 percentage points. An important metric to highlight is the recall value, where the proposed model outperforms YOLOv5 by 7.7 percentage points. This improvement in recall reflects the YOLO-RAW model's superior ability to minimize false negatives, signifying its proficiency in correctly identifying positive instances or elements as positive. Illustrated in Figure 5.16 is a selection of detected samples from MWTS using the YOLO-RAW model. Additionally, Figure 5.17 presents a comparative analysis of the detection performance between YOLOv5 and our proposed model, YOLO-RAW. It's important to note that our proposed model exhibits a distinctive emphasis on UAV detection, a characteristic attributed to the incorporation of the SimAM attention module. This module

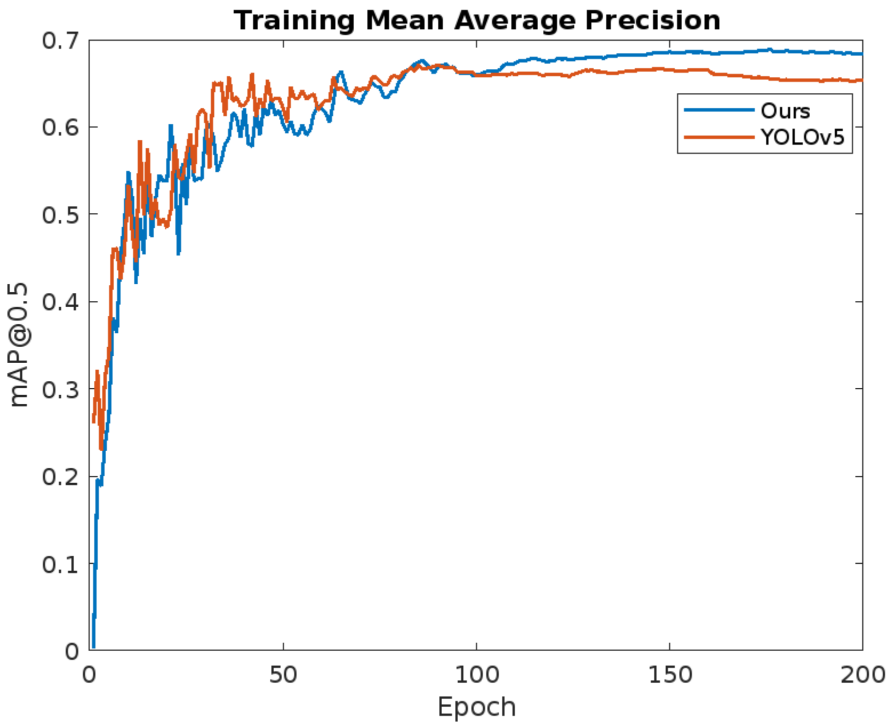


Figure 5.12: The mean average precision of the proposed YOLO-RAW model outperforms the baseline YOLOv5 model for *MWTS*.

refines the feature map within the model's neck section, enhancing its proficiency in detecting UAVs, a valuable insight into our research. The mAP_{50}, average recall (R), and training/validation loss of our proposed model and YOLOv5 are shown in Figure 5.12, Figure 5.13, and Figure 5.15, respectively. Figure clearly shows the improved performance of the proposed model for *MWTS* dataset. The improved recall of YOLO-RAW hints that the YOLO-RAW yields lower false negatives than YOLOv5. Figure 5.14 shows that training the proposed model for 200 epochs does not make our model overfit.

Figure 5.13: The mean recall is shown, and Our model outperforms the baseline YOLOv5 model for *MWTS*.

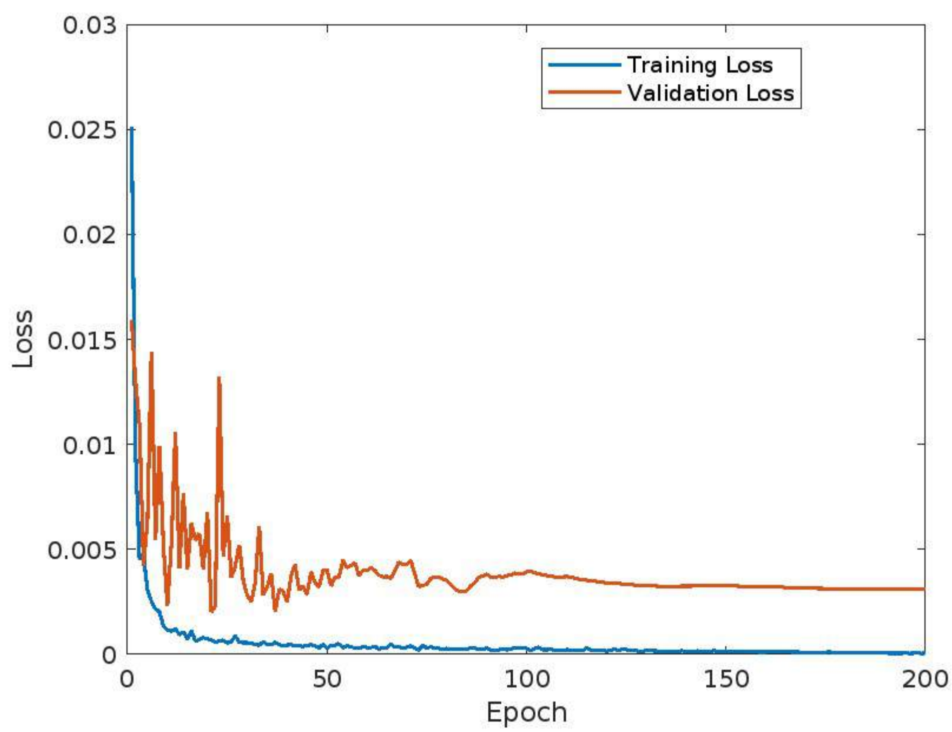


Figure 5.14: The training and validation loss of the proposed YOLO-RAW model.

Table 5.8: Summary of the performance of proposed model YOLO-RAW and YOLOv5 for *MultiWeatherVision Test Set (MWTS)*. The proposed Model outperforms the YOLOv5.

Model	$\mathbf{mAP_{50}}$	$\mathbf{mAP_{50-95}}$	P	R
Faster-RCNN	52.78	28.19	60.2	39.6
RetinaNet	58.21	31.3	63.0	41
YOLOv8m	60.7	35.3	73.1	54.7
YOLOv5m	65.7	37.6	79.4	61.3
YOLO-RAW (Ours)	**71.1 (+5.4)**	**38.1 (+0.5)**	**80 (+0.6)**	**69 (+7.7)**

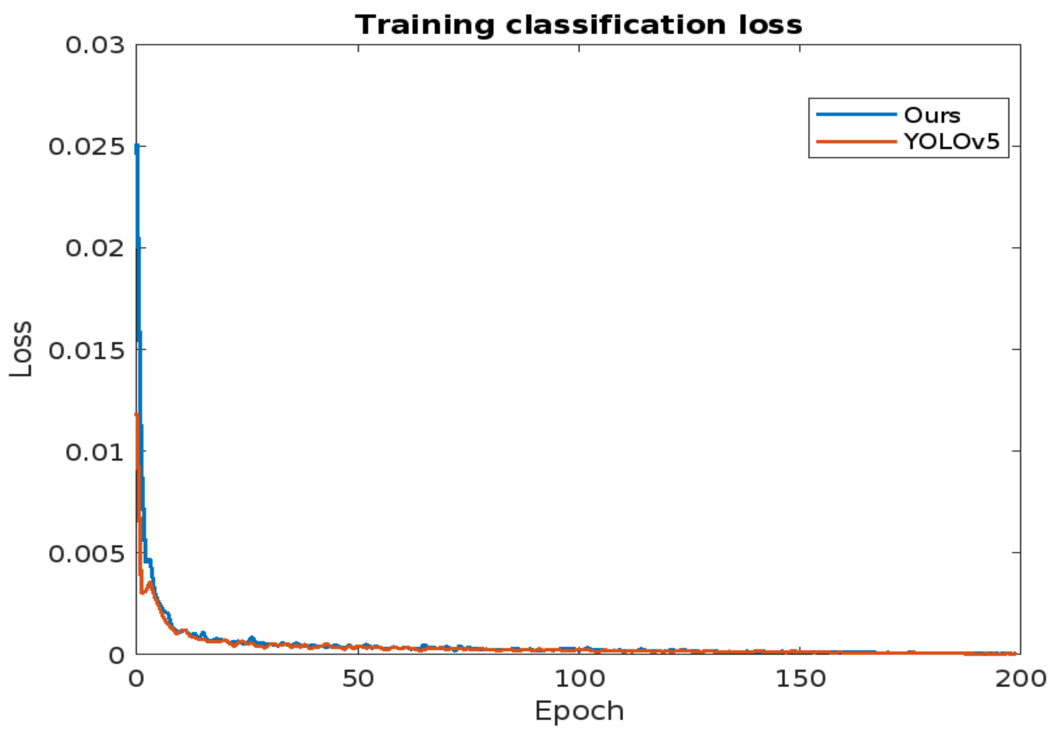


Figure 5.15: The model losses are shown in Figure. Our model outperforms the baseline YOLOv5 model for *MWTS*.

5.3.6 Enhancing YOLO-RAW with Data-Augmentation

In this phase of focused training, YOLO-RAW is trained on a diverse dataset (ACBD) to understand better and handle adverse weather scenarios. The subsequent evaluation, specifically tailored for challenging conditions (MWTS), provided a detailed view of its performance. To explain further, Table 5.9 thoroughly compares the results of YOLOv5 and our new model, YOLO-RAW, both trained with an augmented dataset ACBD. This analysis gives us a clear understanding of the unique capabilities and adaptability of our model when dealing with adverse weather conditions. The proposed model achieved better performance in precision and recall.

Table 5.9: Performance of YOLOv5 and YOLO-RAW With Data-Augmentation of ACBD, and evaluated on MWTS

Model	mAP_{50}	P	R	Inference Time (ms)
YOLOv5	91.2	90.2	88.3	2.0
YOLO-RAW (Ours)	91	**91.7 (+1.5)**	**89.2 (+0.9)**	2.9

5.3.7 Performance Comparison with GradCAM-based Analysis

To introduce an efficient detection system, researchers often use Grad-CAM [113] to examine the focal points of a model and interpret its results. To gain a more comprehensive understanding of the outputs of the proposed model YOLO-RAW, and baseline model YOLOv5, Grad-CAM is incorporated into each model to highlight the regions that have a significant influence on the predictions of the model. To evaluate the UAV detection performance of the models mentioned, images were selected from the *MWTS*, and Grad-CAM was applied.

Figure.5.17 illustrates the results of our proposed YOLO-RAW model in comparison to the baseline YOLOv5 model. The results are comprehensively elucidated, showing that our proposed model, YOLO-RAW, has a higher class score. For example, in Fig.5.17 (a), YOLOv5 displays a lack of focus on the UAV, resulting in a miss detection. On the other hand, in Fig.5.17(b), Fig.5.17(c), and Fig.5.17(d), the YOLO-

RAW model displays a focus on the UAV, attributed to the SimAM attention module. This refined feature mapping, even in unfavorable weather conditions, increases the resilience of our model to difficult weather conditions.

5.4 Ablation Studies

We do all experiments in the ablation study with the proposed model YOLO-RAW, for the *CBD* dataset. The model is tested with *MWTS* Test set.

5.4.1 Ablation Study of SimAM Modules

Firstly, we carried out a series of experiments involving our proposed models. In this investigation, we conducted an ablation study where we removed the SimAM modules one by one from the YOLO-RAW architecture. Specifically, these SimAM were removed after the third upsample module and then the second upsample module, which integrates with the P4 feature map, and P2 feature map respectively. The results of this ablation study are presented in Table 5.10. This analysis sheds light on the judicious selection of the number and placement of SimAM modules within the architecture. The values enclosed in parentheses indicate the enhancements in comparison to YOLOv5m.

Our findings underscore the significance of the SimAM module in feature refinement. Notably, we observed a decline in both mAP and P values following the removal of one SimAM module, underscoring the importance of this module in enhancing model performance. This insight is crucial in our research as it highlights the role of SimAM

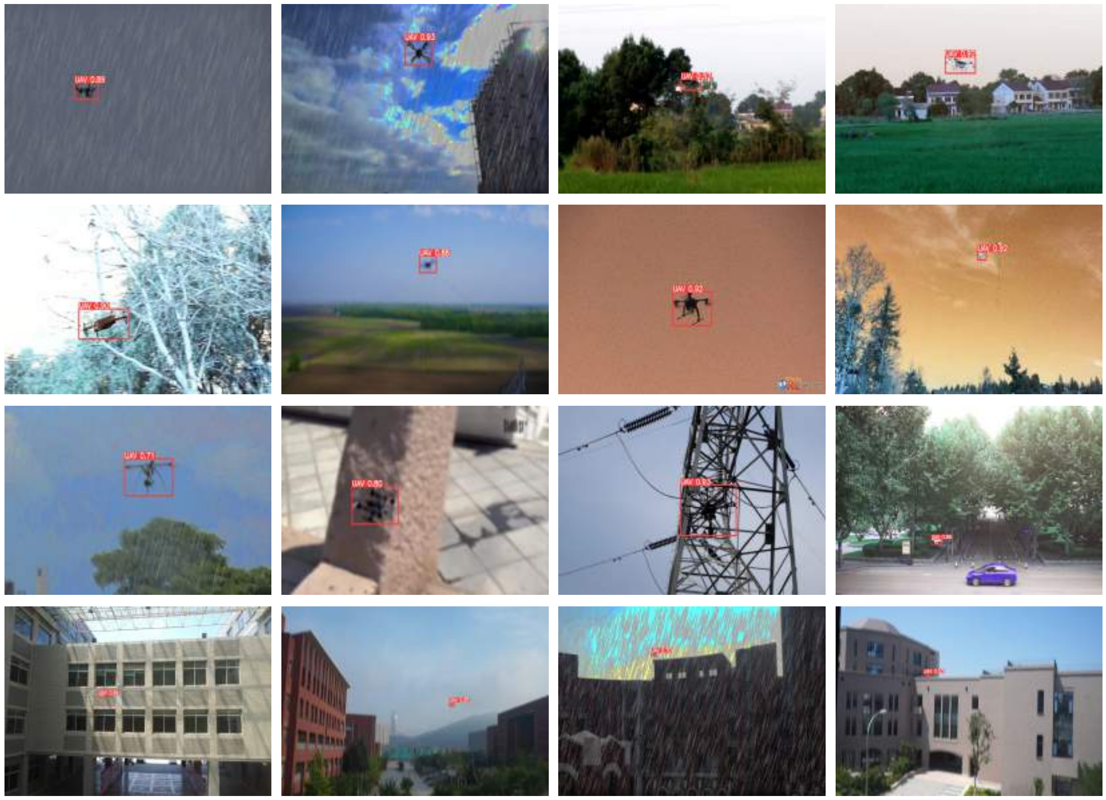


Figure 5.16: Illustrative detection outcomes using YOLO-RAW from the *MWTS*. YOLO-RAW, the proposed model, demonstrates its capability to detect UAVs in adverse weather conditions and across various scales.

in feature optimization. It's important to highlight that a stepwise removal of SimAM modules results in a reduction in the Recall (R) value. This decline signifies a diminishing capability of the model to mitigate false positives. This observation underscores the crucial role played by SimAM modules in enhancing the model's ability to manage false positive identifications.

Table 5.10: Summary of the ablation experiments while removing one SimAM module from YOLO-RAW. Brackets show the improvement and decrease of values from the YOLOv5.

Model	$\mathbf{mAP_{50}}$	$\mathbf{mAP_{50-95}}$	P	R	Parameters (M)
YOLOv5	65.7	37.6	79.4	61.3	21.2
YOLO-RAW (Removing Two SimAM)	65.2 (-0.5)	34.9 (-2.7)	76.6 (-2.8)	62.2 (+0.9)	80.2
YOLO-RAW (Removing One SimAM)	66.4 (+0.7)	35.8 (-1.8)	77.7 (-1.7)	64 (+2.7)	80.2
YOLO-RAW (Ours)	**71.1 (+5.4)**	**38.1 (+0.5)**	**80 (+0.6)**	**69 (+7.7)**	80.2

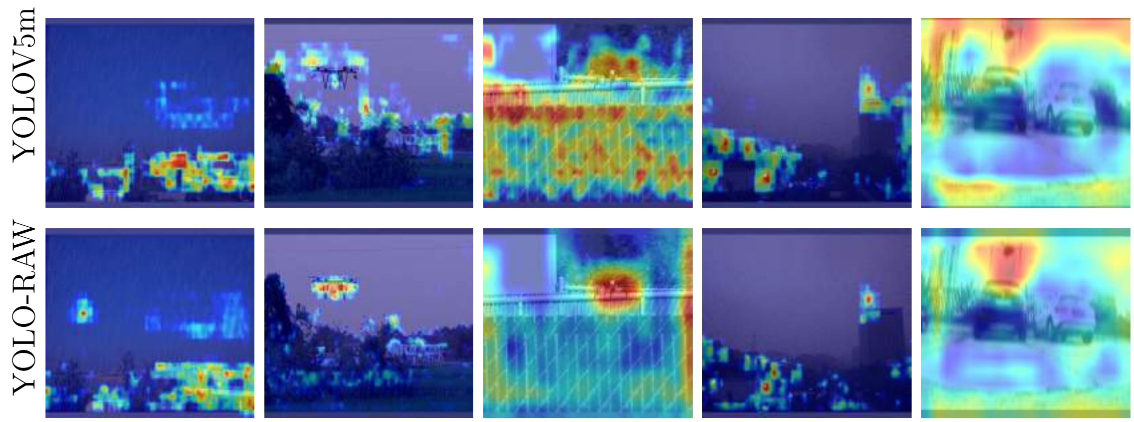


Figure 5.17: Sample images from *MWTS* tested with YOLO-RAW and compared with YOLOV5 based on Grad-CAM. The proposed model YOLO-RAW focuses on the UAV and shows a higher class score than YOLOv5.

5.4.2 Ablation Study of Head Modules

In this particular experimental setup, we opted to eliminate an additional head along with the associated SimAM module from the YOLO-RAW architecture. The outcomes of this modification are outlined in Table 5.11, shedding light on the significance of the supplementary heads. The results indicate that while a reduction in parameters can lead to favorable outcomes, it's essential to consider the trade-off. Specifically, for larger-sized UAVs, this alteration may hinder the model's ability to effectively detect UAVs in images. This observation has noteworthy implications for our research, emphasizing the impact of head and module selection on model performance.

Table 5.11: Summary of the ablation experiments while removing one Head and one SimAM module from YOLO-RAW. Brackets show the improvement of values from the YOLOv5m.

Model	$\mathbf{mAP_{50}}$	$\mathbf{mAP_{50-95}}$	P	R	Parameters (M)
YOLOv5	65.7	37.6	79.4	61.3	21.2
YOLO-RAW(Removing one Head and one SimAM)	69.9 (+4.2)	38.2 (+0.6)	79.7 (+0.3)	65.2 (+3.9)	21.2
YOLO-RAW (Ours)	**71.1 (+5.4)**	**38.4 (+0.7)**	**80 (+0.6)**	**69 (+7.7)**	80.2

5.4.3 Ablation Study of other Attention Modules

In this particular experiment, we introduced an alternative attention module, detailed in Section 5.1. We opted to replace all four SimAM modules with SENet by keeping the same place as SimAM have in the neck, and the outcomes are detailed in Table 5.12. While the overall mean Average Precision (mAP) showed improvement compared to YOLOv5, it didn't reach the mAP achieved by our proposed model. It's worth noting that the integration of the SENet module resulted in an additional 3.5 million parameters being added to the YOLO-RAW model. This insight provides valuable context for our research, emphasizing the trade-offs associated with different attention modules and their impact on model parameters and performance.

Table 5.12: Summary of the ablation experiments while replacing the SimAM module with the SENet module. Brackets show the improvement of values from the YOLOv5m.

Model	**mAP$_{50}$**	**mAP$_{50-95}$**	**P**	**R**	**Parameters (M)**
YOLOv5	65.7	37.6	79.4	61.3	21.2
YOLO-RAW(Replacing SimAM With SENet)	70 (+4.3)	38 (+0.4)	76.1 (-3.3)	67.5 (+6.2)	83.7
YOLO-RAW (Ours)	**71.1 (+5.4)**	**38.4 (+0.7)**	**80 (+0.6)**	**69 (+7.7)**	80.2

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1 Conclusion

The experiments and analysis carried out in this study shed light on a variety of insights linked to the detection of UAVs under adverse conditions. It is observed through examining the influence of factors such as motion blur, noise, and rain that these conditions have a major impact on the performance of object detection models. However, through the augmentation of the training dataset with diverse and adversely affected image data, the models' generalization abilities and performance were enhanced under unfavorable situations. In addition, in this study, we proposed YOLO-RAW, a vision-based UAV detection system designed to excel in adverse weather conditions. YOLO-RAW demonstrates notable enhancements in UAV detection precision and recall when compared to YOLOv5m. To evaluate the model's performance, we have meticulously compiled a range of training and test sets, encompassing both pristine and adverse weather conditions. In these test sets, we have deliberately introduced various adverse weather conditions, including rain, motion blurriness, and noise. Furthermore, our approach incorporates the SimAM attention mechanism, and we have conducted ablation studies by substituting it with SENet to make performance comparisons. Notably, our proposed system with SimAM attention outperforms the YOLOv5m model, achieving a remarkable improvement of 5.4 percentage points in mean average precision and 7.7 percentage points in recall. The proposed model also shows a 10.4 percentage point improvement in mean average precision and a 14.3 percentage point improvement in recall value over the YOLOv8 model.

6.2 Future Work

In this study, we introduced an additional scale in SPP and head designed for the detection of larger UAVs that were previously overlooked. For future work, we plan to add the following:

- We are considering the possibility of adding a specialized head for tiny-scale UAVs or implementing a slicing technique to enhance the precision in detecting these smaller UAVs.
- The optimized loss function can be generated to further enhance the model mean average precision and early convergence.
- The YOLO-RAW can be optimized for deployment on edge devices by using compression and knowledge distillation techniques.

REFERENCES

[1] R. Xu, H. Lin, K. Lu, L. Cao, and Y. Liu, "A forest fire detection system based on ensemble learning," *Forests*, vol. 12, p. 217, 02 2021.

[2] N. Sharma, S. Baral, M. Paing, and R. Chawuthai, "Parking time violation tracking using yolov8 and tracking algorithms," *Sensors*, vol. 23, p. 5843, 06 2023.

[3] R. Yang, Z. Pan, X. Jia, L. Zhang, and Y.-K. Deng, "A novel cnn-based detector for ship detection based on rotatable bounding box in sar images," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, pp. 1–1, 01 2021.

[4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[5] L. Yang, R.-Y. Zhang, L. Li, and X. Xie, "Simam: A simple, parameter-free attention module for convolutional neural networks," in *Proceedings of the 38th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. Meila and T. Zhang, Eds.,

vol. 139. PMLR, 18–24 Jul 2021, pp. 11 863–11 874. [Online]. Available: http://proceedings.mlr.press/v139/yang21o.html

[6] H. M. Oh, H. Lee, and M. Y. Kim, "Comparing convolutional neural network(cnn) models for machine learning-based drone and bird classification of anti-drone system," in *2019 19th International Conference on Control, Automation and Systems (ICCAS)*, 2019, pp. 87–90.

[7] I. Bisio, C. Garibotto, F. Lavagetto, A. Sciarrone, and S. Zappatore, "Unauthorized amateur uav detection based on wifi statistical fingerprint analysis," *IEEE Communications Magazine*, vol. 56, pp. 106–111, 04 2018.

[8] G. Pham and P. Nguyen, "Drone detection experiment based on image processing and machine learning," 02 2020.

[9] P. Casabianca and Y. Zhang, "Acoustic-based uav detection using late fusion of deep neural networks," *Drones*, vol. 5, no. 3, p. 54, 2021.

[10] . D. de Quevedo, F. I. Urzaiz, J. G. Menoyo, and A. A. López, "Drone detection with x-band ubiquitous radar," in *2018 19th International Radar Symposium (IRS)*, 2018, pp. 1–10.

[11] I. Guvenc, F. Koohifar, S. Singh, M. L. Sichitiu, and D. Matolak, "Detection, tracking, and interdiction for amateur drones," *IEEE Communications Magazine*, vol. 56, no. 4, pp. 75–81, 2018.

[12] M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, "Rf-based drone detection and identification using deep learning approaches:

An initiative towards a large open source drone database," *Future Generation Computer Systems*, vol. 100, pp. 86–97, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X18330760

[13] A. Barisic, M. Car, and S. Bogdan, "Vision-based system for a real-time detection and following of uav," in *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, 2019, pp. 156–159.

[14] R. Opromolla, G. Fasano, and D. Accardo, "A vision-based approach to uav detection and tracking in cooperative applications," *Sensors*, vol. 18, no. 10, 2018. [Online]. Available: https://www.mdpi.com/1424-8220/18/10/3391

[15] F. Dadrass Javan, F. Samadzadegan, M. Gholamshahi, and F. Ashatari Mahini, "A modified yolov4 deep learning network for vision-based uav recognition," *Drones*, vol. 6, no. 7, 2022. [Online]. Available: https://www.mdpi.com/2504-446X/6/7/160

[16] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao, "Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1020–1027, 2021.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector." in *ECCV (1)*, 2016.

[18] Q. Wei, X. Hu, X. Wang, and H. Wang, "Improved retinanet target detection model," in *2022 2nd International Conference on Algorithms, High Performance Computing and Artificial Intelligence (AHPCAI)*, 2022, pp. 470–476.

[19] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” 2018, cite arxiv:1804.02767Comment: Tech Report. [Online]. Available: http://arxiv.org/abs/1804.02767

[20] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, “Single-shot refinement neural network for object detection,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4203–4212.

[21] S. Ren, K. He, R. B. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks.” in *NIPS*, 2015.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” 2017.

[23] Z. Cai and N. Vasconcelos, “Cascade r-cnn: High quality object detection and instance segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1483–1498, 2021.

[24] X. Lu, B. Li, Y. Yue, Q. Li, and J. Yan, “Grid r-cnn,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 7355–7364.

[25] V. Walter, M. Vrba, and M. Saska, “On training datasets for machine learning-based visual relative localization of micro-scale uavs,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 674–10 680.

[26] H. Sun, J. Yang, J. Shen, D. Liang, L. Ning-Zhong, and H. Zhou, "Tib-net: Drone detection network with tiny iterative backbone," *IEEE Access*, vol. 8, pp. 130 697–130 707, 2020.

[27] Y. Yoo, D. Han, and S. Yun, "Extd: Extremely tiny face detector via iterative filter reuse," 2019.

[28] N. Jiang, K. Wang, X. Peng, X. Yu, Q. Wang, J. Xing, G. Li, G. Guo, J. Zhao, and Z. Han, "Anti-uav: A large multi-modal benchmark for uav tracking," 01 2021.

[29] P. Voigtlaender, J. Luiten, P. H. S. Torr, and B. Leibe, "Siam r-cnn: Visual tracking by re-detection," 2020.

[30] L. Huang, X. Zhao, and K. Huang, "Globaltrack: A simple and strong baseline for long-term tracking," 2019.

[31] M. Nalamati, A. Kapoor, M. Saqib, N. Sharma, and M. Blumenstein, "Drone detection in long-range surveillance videos," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2019, pp. 1–6.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[33] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.

[34] A. Coluccia, A. Fascista, A. Schumann, L. Sommer, A. Dimou, D. Zarpalas, M. Méndez, D. de la Iglesia, I. González, J.-P. Mercier, G. Gagné, A. Mitra, and S. Rajashekar, "Drone vs. bird detection: Deep learning algorithms and results from a grand challenge," *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2824

[35] S. Singha and B. Aydin, "Automated drone detection using yolov4," *Drones*, vol. 5, no. 3, 2021. [Online]. Available: https://www.mdpi.com/2504-446X/5/3/95

[36] A. Bochkovskiy, C. Wang, and H. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[37] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.

[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13.* Springer, 2014, pp. 740–755.

[39] M. Saqib, S. Daud Khan, N. Sharma, and M. Blumenstein, "A study on detecting drones using deep convolutional neural networks," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, pp. 1–5.

[40] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," 2013.

[41] A. Coluccia, A. Fascista, A. Schumann, L. Sommer, A. Dimou, D. Zarpalas, M. Méndez, D. de la Iglesia, I. González, J.-P. Mercier, G. Gagné, A. Mitra, and S. Rajashekar, "Drone vs. bird detection: Deep learning algorithms and results from a grand challenge," *Sensors*, vol. 21, no. 8, 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/8/2824

[42] D. T. Wei Xun, Y. L. Lim, and S. Srigrarom, "Drone detection using yolov3 with transfer learning on nvidia jetson tx2," in *2021 Second International Symposium on Instrumentation, Control, Artificial Intelligence, and Robotics (ICA-SYMP)*, 2021, pp. 1–6.

[43] Y. Lv, Z. Ai, M. Chen, X. Gong, Y. Wang, and Z. Lu, "High-resolution drone detection based on background difference and sag-yolov5s," *Sensors*, vol. 22, no. 15, 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/15/5825

[44] [Online]. Available: https://zenodo.org/records/7347926

[45] F. Dadboud, V. Patel, V. Mehta, M. Bolic, and I. Mantegh, "Single-stage uav detection and classification with yolov5: Mosaic data augmentation and panet," in *2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2021, pp. 1–8.

[46] Y. Zheng, Z. Chen, D. Lv, Z. Li, Z. Lan, and S. Zhao, "Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1020–1027, 2021.

[47] ——, "Air-to-air visual detection of micro-uavs: An experimental evaluation of deep learning," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1020–1027, 2021.

[48] M. Dil, M. U. Khan, M. Z. Alam, F. A. Orakazi, Z. Kaleem, and C. Yuen, "Safespace mfnet: Precise and efficient multifeature drone detection network," 2022.

[49] Y. Wang, Z. Huang, R. Laganière, H. Zhang, and L. Ding, "A uav to uav tracking benchmark," *Knowledge-Based Systems*, vol. 261, p. 110197, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S095070512201293X

[50] E. Unlu, E. Zenou, N. Rivière, and P.-E. Dupouy, "Deep learning-based strategies for the detection and tracking of drones using several cameras," Jun 2020. [Online]. Available: https://hal.science/hal-02863410

[51] S. A. Hassan, T. Rahim, and S. Y. Shin, "Real-time uav detection based on deep learning network," in *2019 International Conference on Information and Communication Technology Convergence (ICTC)*, 2019, pp. 630–632.

[52] D. K. Behera and A. Bazil Raj, "Drone detection and classification using deep learning," in *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, 2020, pp. 1012–1016.

[53] M. Wisniewski, Z. Rana, and I. Petrunin, "Drone model classification using convolutional neural network trained on synthetic data," *Journal of Imaging*, vol. 8, p. 218, 08 2022.

[54] N. Jiang, K. Wang, X. Peng, X. Yu, Q. Wang, J. Xing, G. Li, G. Guo, J. Zhao, and Z. Han, "Anti-uav: A large multi-modal benchmark for uav tracking," 01 2021.

[55] S. Hinterstoisser, V. Lepetit, P. Wohlhart, and K. Konolige, *On Pre-trained Image Features and Synthetic Images for Deep Learning: Munich, Germany, September 8-14, 2018, Proceedings, Part I*, 01 2019, pp. 682–697.

[56] M. . Pawełczyk and M. Wojtyra, "Real world object detection dataset for quadcopter unmanned aerial vehicle detection," *IEEE Access*, vol. 8, pp. 174 394–174 409, 2020.

[57] A. Y. Husodo, G. Jati, N. Alfiany, and W. Jatmiko, "Intruder drone localization based on 2d image and area expansion principle for supporting military defence system," in *2019 IEEE International Conference on Communication, Networks and Satellite (Comnetsat)*, 2019, pp. 35–40.

[58] M. Kristan and et.al., "The sixth visual object tracking vot2018 challenge results," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds. Cham: Springer International Publishing, 2019, pp. 3–53.

[59] B. Isaac-Medina, M. Poyser, D. Organisciak, C. Willcocks, T. Breckon, and H. Shum, "Unmanned aerial vehicle visual detection and tracking using deep neural networks: A performance benchmark," 03 2021.

[60] N. Jiang, K. Wang, X. Peng, X. Yu, Q. Wang, J. Xing, G. Li, J. Zhao, G. Guo, and Z. Han, "Anti-uav: A large multi-modal benchmark for uav tracking," 2021. [Online]. Available: https://arxiv.org/abs/2101.08466

[61] C. Wang, J. Tian, J. Cao, and X. Wang, "Deep learning-based uav detection in pulse-doppler radar," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, pp. 1–12, 2021.

[62] E. Kaya and G. B. Kaplan, "Neural network based drone recognition techniques with non-coherent s-band radar," in *2021 IEEE Radar Conference (RadarConf21)*, 2021, pp. 1–6.

[63] S. Björklund, "Target detection and classification of small drones by boosting on radar micro-doppler," *2018 15th European Radar Conference (EuRAD)*, pp. 182–185, 2018.

[64] R. Akter, M. Golam, J.-M. Lee, and D.-S. Kim, "Doppler radar-based real-time drone surveillance system using convolution neural network," in *2021 Interna-*

tional Conference on Information and Communication Technology Convergence (ICTC), 2021, pp. 474–476.

[65] D. Kawaguchi, R. Nakamura, and H. Hadama, "Evaluation on a drone classification method using uwb radar image recognition with deep learning," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, 2021, pp. 1–5.

[66] M. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, and A. Erbad, "Rf-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database," *Future Generation Computer Systems*, vol. 100, 05 2019.

[67] M. Z. Anwar, Z. Kaleem, and A. Jamalipour, "Machine learning inspired sound-based amateur drone detection for public safety applications," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 2526–2534, 2019.

[68] F. Svanstrom, C. Englund, and F. Alonso-Fernandez, "Real-time drone detection and tracking with visible, thermal and acoustic sensors," 01 2021, pp. 7265–7272.

[69] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, and I. Guvenc, "Micro-uav detection and classification from rf fingerprints using machine learning techniques," in *2019 IEEE Aerospace Conference*, 2019, pp. 1–13.

[70] A. Karlsson, M. Jansson, and M. Hämäläinen, "Model-aided drone classification using convolutional neural networks," in *2022 IEEE Radar Conference (RadarConf22)*, 2022, pp. 1–6.

[71] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.

[72] V. Semkin, M. Yin, Y. Hu, M. Mezzavilla, and S. Rangan, "Drone detection and classification based on radar cross section signatures," in *2020 International Symposium on Antennas and Propagation (ISAP)*, 2021, pp. 223–224.

[73] S. Rahman and D. Robertson, "Classification of drones and birds using convolutional neural networks applied to radar micro-doppler spectrogram images," *IET Radar, Sonar Navigation*, 05 2020.

[74] I. Roldan, C. Del-Blanco, . Duque de Quevedo, F. Ibañez Urzaiz, J. Menoyo, A. Asensio, D. Berjón, F. Jaureguizar, and N. García, "Dopplernet: a convolutional neural network for recognizing targets in real scenarios using a persistent range-doppler radar," *IET Radar, Sonar Navigation*, vol. 14, 04 2020.

[75] Ignacio, "Real doppler rad-dar database," Jun 2019. [Online]. Available: https://www.kaggle.com/datasets/iroldan/real-doppler-raddar-database

[76] A. Rozantsev, V. Lepetit, and P. Fua, "Flying objects detection from a single moving camera," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 4128–4136.

[77] Y. Chen, P. Aggarwal, J. Choi, and C.-C. J. Kuo, "A deep learning approach to drone monitoring," in *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2017, pp. 686–691.

[78] S. Singha and B. Aydin, "Automated drone detection using yolov4," *Drones*, vol. 5, p. 95, 09 2021.

[79] Dasmehdixtr, "Drone dataset (uav)," Oct 2019. [Online]. Available: https://www.kaggle.com/datasets/dasmehdixtr/drone-dataset-uav?resource=download

[80] F. Svanström, C. Englund, and F. Alonso-Fernandez, "Real-time drone detection and tracking with visible, thermal and acoustic sensors," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 7265–7272.

[81] J. Li, D. H. Ye, T. Chung, M. Kolsch, J. Wachs, and C. Bouman, "Multi-target detection and tracking from a single camera in unmanned aerial vehicles (uavs)," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4992–4997.

[82] A. Karlsson, "Radar measurements on drones, birds and humans with a 77GHz FMCW sensor." Sep. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5845259

[83] M. . Pawełczyk and M. Wojtyra, "Real world object detection dataset for quadcopter unmanned aerial vehicle detection," *IEEE Access*, vol. 8, pp. 174 394–174 409, 2020.

[84] M. Pawełczyk and M. Wojtyra, "Real world object detection dataset for quadcopter unmanned aerial vehicle detection," *IEEE Access*, vol. 8, pp. 174 394–174 409, 2020.

[85] V. Walter, M. Vrba, and M. Saska, "On training datasets for machine learning-based visual relative localization of micro-scale uavs," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 10 674–10 680.

[86] J. Zhao, J. Zhang, D. Li, and D. Wang, "Vision-based anti-uav detection and tracking," 2022.

[87] UjjwalSaxena, "Ujjwalsaxena/automold–road-augmentation-library." [Online]. Available: https://github.com/UjjwalSaxena/Automold--Road-Augmentation-Library

[88] M. Ng, D. Jagetiya, X. Gao, H. Shi, J. Gao, and J. Liu, "Real-time detection of objects on roads for autonomous vehicles using deep learning," 08 2022.

[89] M. Rahman, M. Islam, C. Holt, J. Calhoun, and M. Chowdhury, "Dynamic error-bounded lossy compression to reduce the bandwidth requirement for real-time vision-based pedestrian safety applications," *Journal of Real-Time Image Processing*, vol. 19, 02 2022.

[90] Leroux, Sam and Li, Bo and Simoens, Pieter, "Multi-branch neural networks for video anomaly detection in adverse lighting and weather conditions," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2022, pp. 3027–3035. [Online]. Available: http://doi.org/10.1109/WACV51458.2022.00308

[91] S. Zheng, Y. Wu, S. Jiang, C. Lu, and G. Gupta, "Deblur-yolo: Real-time object detection with efficient blind motion deblurring," in *2021 International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–8.

[92] Y. Zhou, J. Jiao, H. Huang, Y. Wang, J. Wang, H. Shi, and T. Huang, "When awgn-based denoiser meets real noises," 2019.

[93] "Home." [Online]. Available: https://docs.ultralytics.com/#yolo-a-brief-history

[94] S. Rath, "Yolov8 ultralytics: State-of-the-art yolo models," Mar 2023. [Online]. Available: https://learnopencv.com/ultralytics-yolov8/

[95] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[96] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.

[97] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[98] G. Jocher, "Yolov5 by ultralytics," 2020. [Online]. Available: https://github.com/ultralytics/yolov5

[99] Ultralytics, "Ultralytics/ultralytics: New - yolov8 in pytorch gt; onnx gt; coreml gt; tflite." [Online]. Available: https://github.com/ultralytics/ultralytics

[100] Deci and Deci, "Yolo-nas by deci achieves state-of-the-art performance on object detection using neural architecture search," May 2023. [Online]. Available: https://deci.ai/blog/yolo-nas-object-detection-foundation-model/

[101] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[102] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

[103] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[104] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768.

[105] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision*

and Pattern Recognition (CVPR), June 2020.

[106] A. Munir, A. J. Siddiqui, and S. Anwar, "Investigation of uav detection in images with complex backgrounds and rainy artifacts," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops*, November 2023.

[107] Z. Qu, L. yuan Gao, S. ye Wang, H. nan Yin, and T. ming Yi, "An improved yolov5 method for large objects detection with multi-scale feature cross-layer fusion network," *Image and Vision Computing*, vol. 125, p. 104518, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885622001470

[108] X. Jin and J. Han, *K-Means Clustering.* Boston, MA: Springer US, 2010, pp. 563–564. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_425

[109] S. Kardakis, I. Perikos, F. Grivokostopoulou, and I. Hatzilygeroudis, "Examining attention mechanisms in deep learning models for sentiment analysis," *Applied Sciences*, vol. 11, no. 9, 2021. [Online]. Available: https://www.mdpi.com/2076-3417/11/9/3883

[110] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," 2018.

[111] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," 2019.

[112] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[113] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 618–626.

VITAE

- Name: ADNAN MUNIR
- Nationality: PAKISTANI
- Date of Birth: 09 Nov, 1995
- Email: *adnanmunir294@gmail.com*
- Permanent Address: 2F-24 Pakistan Ordnance Factories (POF) Sanjwal Cantt Attock, Pakistan PO-BOX 43670.

- **Bachelor of Science in Computer Engineering**

\- July, 2019

\- COMSATS University Islamabad, Pakistan

- **Master of Science in Computer Engineering**

\- Dec, 2023

\- King Fahd University of Petroleum and Minerals, KSA

Publication

- **A. Munir**, A. J. Siddiqui, and S. Anwar, *"Investigation of UAV detection in images with complex backgrounds and rainy artifacts,"* in Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops, November 2023.
- **A. Munir**, A. J. Siddiqui, A. El-Maleh, and Khan, AH, *"Benchmarking and Enhancing Robustness By Data-Augmentation of Object Detection Models for UAV Detection"* submitted to IEEE Access, 2023.
- **A. Munir**, A. J. Siddiqui, A. El-Maleh, *"YOLO-RAW: Advancing UAV Detection Robustness in Varied Weather Conditions"* to be submitted.