VEHICLE DETECTION METHOD BASED ON OPTIMISED YOLOV7 LIGHTWEIGHT MODEL

JOHAN LELA ANDIKA BIN JOHAN BUDIMAN

FACULTY OF ENGINEERING
UNIVERSITI MALAYA
KUALA LUMPUR

2024

VEHICLE DETECTION METHOD BASED ON OPTIMISED YOLOV7 LIGHTWEIGHT MODEL

JOHAN LELA ANDIKA BIN JOHAN BUDIMAN

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING SCIENCE

FACULTY OF ENGINEERING
UNIVERSITI MALAYA
KUALA LUMPUR

2024

UNIVERSITI MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: **JOHAN LELA ANDIK BIN JOHAN BUDIMAN**
Matric No: **22058586/1**
Name of Degree: **MASTER OF ENGINEERING SCIENCE**
Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"):
VEHICLE DETECTION METHOD BASED ON OPTIMISED YOLOV7 LIGHTWEIGHT MODEL

Field of Study:

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the Universiti Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature Date: 20/08/2024

Subscribed and solemnly declared before,

Witness's Signature Date: 20/08/2024

Name:

Designation:

VEHICLE DETECTION METHOD BASED ON OPTIMSED YOLOV7 LIGHTWEIGHT MODEL

ABSTRACT

The advancement of unmanned aerial vehicles (UAVs) has encouraged researchers to update object detection algorithms for better accuracy and computational performance. Previous works that apply deep learning models for object detection applications required high graphics processing unit (GPU) computational power. Generally, object detection models suffer trade-off between accuracy and model size where the relationship is not always linear in deep learning models. Various factors such as architectural design, optimization techniques, and dataset characteristics can significantly influence the accuracy, model size and computational cost in adopting object detection models for low-cost embedded devices. Hence, it is crucial to employ lightweight object detection models for real-time object identification for the solution to be sustainable. This work proposes modifications on the head and backbone architecture of YOLOv7-tiny model. Firstly, efficient long-range aggregation network for vehicle detection (ELAN-VD) is incorporated in backbone layer. Secondly, the (UPSAMPLE-VD) on head architecture is improvised resolution to improve the detection accuracy of small vehicles in the aerial image. This study shows that the proposed method yields mean average precision (mAP) of 77.47 %, which is higher than the conventional YOLOv7-tiny of 48.89 %. In addition, the proposed model shown significant performance when compared to previous works, making it viable for application in low-cost embedded devices.

Keywords: Image, Object detection, Vehicle detection, YOLO

KAEDAH PENGESAHAN KENDERAAN BERDASARKAN MODEL RINGAN YOLOV7 YANG DIOPTIMUMKAN

ABSTRAK

Kemajuan dalam pesawat udara tanpa juruterbang atau *unmanned aerial vehicles* (UAVs) telah mendorong penyelidik untuk mengemaskini algoritma pengesanan objek untuk ketepatan dan prestasi pengkomputeran yang lebih baik. Kajian terdahulu menggunakan model pembelajaran mendalam untuk aplikasi pengesanan objek memerlukan kuasa pengkomputeran unit pemprosesan grafik (GPU) yang tinggi. Secara umum, model pengesanan objek mengalami masalah keseimbangan antara ketepatan dan saiz model di mana hubungan tersebut tidak selalu linear dalam model pembelajaran mendalam. Pelbagai faktor seperti reka bentuk seni bina, teknik pengoptimuman, dan ciri-ciri set data dapat mempengaruhi ketepatan, saiz model, dan kos pengkomputeran dalam mengguna pakai model pengesanan objek untuk peranti terbenam kos rendah. Oleh itu, adalah penting untuk menggunakan model pengesanan objek ringan untuk mengesan objek secara masa nyata agar penyelesaian itu lestari. Kajian ini mencadangkan pengubahsuaian pada seni bina kepala dan asas model YOLOv7-tiny. Pertama, rangkaian pengagregatan jarak jauh yang cekap untuk pengesanan kenderaan (ELAN-VD) disertakan dalam lapisan asas. Kedua, (UPSAMPLE-VD) pada seni bina kepala ditingkatkan resolusi untuk meningkatkan ketepatan pengesanan kenderaan kecil dalam imej udara. Kajian ini menunjukkan bahawa kaedah yang dicadangkan menghasilkan purata ketepatan purata (mAP) sebanyak 77.47%, yang lebih tinggi daripada YOLOv7-tiny konvensional sebanyak 48.89%. Selain itu, model yang dicadangkan menunjukkan prestasi yang signifikan berbanding dengan kajian terdahulu, menjadikannya sesuai untuk aplikasi dalam peranti terbenam kos rendah.

Keywords: Gambaran, Pengesahan Objek, Pengesahan Kenderaan, YOLO

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to Allah the Almighty for His abundant blessings and guidance, which have enabled me to successfully complete this thesis.

The success of this thesis is heavily reliant on the support and guidance provided by countless individuals. I would like to express my sincere appreciation to all those who played a part in the successful completion of this thesis.

My greatest gratitude and appreciation are addressed to my research supervisor, Associate Professor. Dr. Anis Salwa Binti Mohd Khairuddin, for her encouragement, guidance, and criticism throughout this research. Her generosity, enthusiasm, motivation, and patience have also profoundly inspired me. It was an honour and privilege to work and study under her great supervisor. She provided valuable assistance in various aspects of research, including conducting high-quality research, crafting an impactful technical paper, and delivering engaging presentations. I would also like to extend my sincere gratitude for her constant support and care during the challenging times of my academic journey. The precious time she spent with me throughout the process is really appreciated. Special thanks also to another supervisor, Professor Ir. Dr. Harikrishnan A/L Ramiah for his kind and thoughtful advice regarding my research direction. Thanks to the invaluable support and interest of those involved, this thesis was successfully completed.

I would like to extend my gratitude to LSF Technology Sdn. Bhd. for providing financial support throughout this research education. Special thanks to their representatives, Mr. Ammar and Ms. Rafieza, for their time and assistance throughout the research.

Furthermore, I would like to extend my appreciation to Universiti Malaya, a renowned institution in Malaysia, for granting me the opportunity to contribute to its academic community. I have gained a lot of new knowledge that could prepare me for my future career.

Lastly, I would like to extend my heartfelt appreciation to my dear late parents, as well as my aunt, her family, and my siblings, for their unwavering love, selflessness, prayers, and unwavering support. I owe a great deal to the guidance and support I received from those in the academic community. Their invaluable assistance has played a crucial role in shaping my current position. Furthermore, I would like to dedicate the success to my supportive friends, Dr. Ellie, Rony and Aini, fore being my good friend throughout this wonderful journey. I am genuinely thankful for their help and for always being there by my side.

Finally, I have a great expectation that my research will be beneficial for anyone interested in reading this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

CBL	:	Convolution + BN layer + Leaky ReLU
CIoU	:	Complete Intersection over Union
CNN	:	Convolutional Neural Networks
Conv	:	Convolution
DNN	:	Deep Neural Networks
ELAN	:	Efficient Layer Aggregation Network
ELAN-VD	:	Efficient Layer Aggregation Network - VEDAI
FLOPs	:	Floating point operations
FPS	:	Frames per second
GPU	:	Graphics processing unit
ITS	:	Intelligent transportation system
mAP	:	Mean Average Precision
MOT	:	Multi-object tracking
NMS	:	Non-Maximum suppression
PAFPN	:	Path Aggregation Feature Pyramid Network
PANet	:	Path Aggregation Network
SSD	:	MultiBox detector
UAV	:	Unmanned aerial vehicles
YOLO	:	You Only Look Once

CHAPTER 1: INTRODUCTION

1.1 Background study

Advancement in Unmanned Aerial Vehicles (UAVs) and imaging technologies play significant role in the areas of computer vision and image processing. Due to unique viewpoint and large field of view, image dataset has become an essential source for many applications such as mapping, intelligent monitoring, precision agriculture, infrastructural inspection, and traffic management (Junos et al., 2022). Generally, the visual appearances of aerial images are different from natural images. Besides that, aerial images suffer from variation in object scale, highly occlusion and truncation conditions. Hence, these issues contribute to challenges in developing accurate real-time object detection system.

YOLO (You Only Look Once) model has been a significant advancement method in object detection, especially in real-time detection applications. YOLO model processes images in a single step, as opposed to traditional methods that require multiple passes through an image, by dividing them into a grid and instantly estimating bounding boxes and class probabilities from this grid. This remarkable method significantly improves inference speed and efficiency. YOLO models are developed based on deep learning and convolutional neural network. The advantages of YOLO models are fast speed for real-time applications and accurate detection. Currently, YOLOv7 model has shown significant speed and accuracy in object detection applications which is between 5 to 160 frames per second. YOLOv7 model can balance speed and accuracy well which makes the model ideal in various object detection applications (Diwan et al., 2023). For example, previous works adopted YOLOv7 model in defect detection (Wang et al., 2022), object detection (Zheng et al., 2023), vehicle detection (Zheng et al., 2023), and pedestrian crosswalk detection (Zhang et al., 2023). The work by (Wang et al., 2022), develops a steel surface defect detection based on improved YOLOv7 to solve the problems of low

detection speed and low detection accuracy of traditional steel surface defect detection methods. Meanwhile, the work by (Zheng et al., 2023), develops a multi-object tracking method (YOLO-BYTE) based on YOLOv7 model to detect and track cows in complex environments. In addition, the work by (Zhang et al., 2023) proposes YOLOv7-RAR model in vehicle detection on urban roads by solving weak perception of small, and insufficient feature extraction. A modified YOLOv7 network was developed for the automatic detection of a pedestrian crosswalk in an urban road network, designed from both pedestrian and vehicle perspectives (Kaya et al., 2023). Finally, the work by (Liang et al., 2023) explores various methods for quick and accurate detection of radar target against complicated background. Previous works have demonstrated recent breakthroughs for one-stage detector models based on YOLOv7. Albeit the high accuracy reported in the previous works, most YOLOv7 models require high computation resources. YOLOv7 models tend to have deep network structures and require large network parameters. Hence, the detection methods require high graphics processing unit (GPU) computation power. Due to limited memory and processing capacity on low-cost embedded devices, it is a key challenge to develop high accuracy and real-time object detection system. As such, YOLOv7 is less practical for low practical for low-cost embedded devices for real-time detection. Therefore, lightweight YOLO model is developed to overcome the abovementioned limitations with the aim to improve detection speed while sustaining good detection accuracy.

Researchers have actively developed lightweight YOLO models in object detection applications by incorporating factorizing convolutions, group convolutions, depth-wise separable convolution, bottleneck design, and neural architecture search (Li et al., 2019). These lightweight YOLO models have simpler and effective network structures with lesser parameters. Consequently, the lightweight YOLO models require less memory and computing resources, making them more feasible to be implemented in low-cost

embedded devices (Chen et al., 2019; Mandal et al., 2019; Razakarivony & Jurie, 2016; Sapitri et al., 2023; Wang et al., 2023; Zarei et al., 2023). This development resulted in the creation of YOLOv7-tiny, an iteration designed with specific advantages to cater practical deployment scenarios. The primary benefits of YOLOv7-tiny model include compact architecture, efficient design, and high-speed model to ensure real-time performance even on hardware with limited computational capabilities. These features are essential for applications that require real-time object detection. YOLOv7-tiny, in the context of autonomous vehicles, can play a critical role in recognising pedestrian, cars, and road obstacles, thereby contributing to safe navigation. YOLOv7-tiny model has shown its significant contributions in real-time object detection works as tabulated in Table 2.1 (Bai et al., 2024; Chen et al., 2023; Hong et al., 2020; Hua et al., 2023; Jiao et al., 2023; Kumar, 2023; Liu & Wang, 2022; She et al., 2023; Ye & Wang, 2023; Yu et al., 2023)

1.2 Problem statement

Object detection identifies and locates objects in images or videos. Challenges include limited capacity on embedded devices, poor image quality, varying object sizes, occlusions, and real time processing. Occlusion obstructs objects, blur reduces boundaries, poor quality degrades performance, and different sizes require specialized algorithms. Limited resources in embedded devices hinder complex models, making lightweight models crucial. These models are efficient in balancing accuracy and resource usage. They address the challenges, enabling effective object detection on embedded devices without overwhelming their limitations. The YOLO lightweight models have limitations that can affect their performance such as reduced accuracy in detecting small objects, low contrast object due to fewer convolutional layers and challenges in handling occlusion, which can lead to inaccurate detection and localization. The YOLOv7-tiny

lightweight model can optimize object detection problems such as dataset augmentation, transfer learning, hyperparameter tuning, resembling and post preprocessing techniques that can be employed. The ELAN and UPSAMPLE blocks in YOLOv7-tiny have major issues that affect how well they detect things and how fast they work. The ELAN block is good at combining features, but it makes things too complicated and uses too many parameters. This makes it hard for the model to work well on devices with limited resources. The complexity also makes it tough for the model to spot small objects, as the many connections might not catch or keep important small details. The UPSAMPLE block has trouble keeping the right level of detail needed to detect things small or overlapping objects. This can cause a loss of detail in the upscaled feature maps, which makes the model less accurate at finding smaller targets. These techniques can enhance the model's performance, improving accuracy, robustness, and occlusion handling capabilities.

1.3 Research objectives

This study aims to develop an accurate and efficient YOLO-based object detection model using a lightweight deep convolutional neural network. The research is conducted based on these objectives:

I. To design a multiclass object detection method for vehicle detection under various dynamic background;
II. To optimise YOLOv7-tiny model capable of effectively detecting and classifying objects in real time; and
III. To evaluate the performance of proposed model on low-cost embedded device

1.4 Research scope

This research is conducted based on the following scopes:

I. This work focuses on object detection, including image classification to classify an object into a certain category and image localization to identify the location of a single object in the given image.

II. This work focuses on improving and modifying the structure of the ELAN-VD and UPSAMPLE-VD one-stage object detection model based on the YOLO model.

III. The trade-off between the detection and computational performance of the proposed models is evaluated using several evaluation metrics. First, the detection performance is evaluated using average precision, precision, recall, F1-score while GFLOPs, detection speed, training time, model size and the number of parameters for the computational performance.

IV. This work evaluates the performance of the proposed models using a novel VEDAI dataset and UA-DETRAC dataset.

1.5 Significance of study

This research proposes novel lightweight object detection models based on YOLOv7-tiny model. Firstly, the model's configurations and network structures were developed by maintaining the cascade-based model scaling technique while enhancing the effectiveness of the long-range aggregation network (ELAN) to achieve higher detection accuracy using small parameters and quicker detection times. Secondly, the UPSAMPLE on head architecture was improvised resolution from lower to higher resolution using reflected ray method on each other block unit to get better resolution. The significant contributions of this research are as follows: Firstly, the developed models achieved better detection performance over the original and state-of-the-art models evaluated on two different datasets. Secondly, the developed models are lighter and less complex leading to low computational costs. Finally, the developed models can be implemented in real-time embedded devices such as NVIDIA Jetson Nano. These advantages are beneficial

for real-time applications such as precision agriculture and remote sensing that operate in constrained environments with limited memory capacity. Additionally, the findings presented in this research will provide valuable information for future research investigating alternative lightweight networks capable of achieving real-time performance.

1.6 Organizational of the thesis

This thesis reports the research on developing a lightweight object detection method based on the YOLOv7 model. This study is divided into five chapters. The chapters are arranged in the following manner:

i. Chapter 1 – Introduction

This chapter provides a general introduction to object detection in computer vision. Firstly, the traditional and modern approaches are discussed. Besides, the challenges for current DNNs methods are highlighted. Then, the research problem statement is stated, which led to the precise research objectives. Next, the research activities required to accomplish the objectives are included in the scope of study. Finally, the significance of the study clearly mentioned.

ii. Chapter 2 – Literature review

This section presents works on object detection based on single-stage methods. In addition, the use of YOLOv7 models in remote sensing applications is discussed. Finally, this chapter discusses about previous works that had been done to achieve significant method on lightweight by applying various application that motivate and encourage this research.

iii. Chapter 3 – Methodology

This chapter introduces the novel datasets adopted in this study. This chapter then explains the proposed research methodology and processes to improve the YOLO-based object model.

iv. Chapter 4 – Results and discussion

The experimental results and discussion from the experiment analysis are presented in this chapter. The first part discusses the detection performance by comparing them to state-of-the-art models and previous works. Then, the second part discusses the computational performance.

v. Chapter 5 – Conclusion and recommendations

This chapter concludes that research objectives and proposes recommendations for future research.

CHAPTER 2: LITERATURE REVIEW

Vehicle detection is a critical task that involves identifying and localizing vehicles in a traffic scenario. More researchers are interested in improving the object detection techniques on remote sensing and UAVs images to solve the challenges encountered by the emerging future of intelligent transportation system (ITS). The ITS tries to manage these issues appropriately. In this system, vehicles are accurately identified and counted, and not only are traffic accidents prevented, but by analysing the route of the vehicles and their numbers, the accidents that have occurred are well investigated, and traffic control is done more efficiently. This achievement is previous in the traffic control system, and the vehicle detection algorithm is one of the most fundamental issues in the ITS system. Vehicle recognition plays a vital role in many applications, such as remote sensing, traffic modelling, traffic monitoring, environmental monitoring, and road planning, for estimation or simulation of air and noise pollution (Chen et al., 2019; Mandal et al., 2019; Razakarivony & Jurie, 2016; Wang et al., 2023; Zarei et al., 2023).

2.1 One-stage detection method

On the contrary, one-stage detection address object detection as a simple regression problem that takes the entire image as input and simultaneously generates class probabilities and multiple bounding boxes (Le & Lin, 2019). This has made the model much faster than the two-stage object detectors. This model achieves an optimal balance between accuracy and speed. Several examples of these models are OverFeat (Sermanet et al., 2013), Single Shot Detector (SSD) (Liu et al., 2016), RetinaNet (Li & Ren, 2019), YOLO series models, including YOLOv1(Redmon et al., 2016), YOLOv2 (Redmon & Farhadi, 2017), YOLOv3 (Redmon & Farhadi, 2018), and YOLOv4 (Bochkovskiy et al., 2020). Unlike the Faster R-CNN network, the YOLO network eliminates the need for a proposal region and simplifies the detection problem by converting it to a regression

problem. It uses regression to generate bounding box coordinates and probabilities for each class simultaneously, which helps to increase the detection speed.

The YOLO network works by splitting the image into equal dimension regions of S x S grids. Each grid is responsible for detecting the target if the center of the ground truth target falls within it. Correspondingly, each grid predicts B bounding box coordinates, the class label (C) and probability of the probability of locating the object within the grid. Finally, the confidence score can be calculated using Equation 2.1.

$$Confidence = Pr\,(Object) \times IoU_{pred}^{truth} \quad (2.1)$$

P_r (Object) value is equal to 1 if the grid contains a target, or else it is equals 0. $IoU_{prediction}^{truth}$ represents the intersection over union (IoU) between the ground truth and the predicted bounding box. The confidence value indicates if a grid contains targets and the accuracy of the predicted bounding box that contains targets. YOLO chooses the best bounding box using the non-maximum suppression (NMS) approach in a condition where multiple bounding boxes detect the same target. The general architecture of the YOLO model is illustrated in Figure 2.1.

Figure 2.1 General architecture of the YOLO model

YOLO series models have produced remarkable results in numerous real-world applications due to their generalize object detection. Various enhanced versions of the YOLO model have been proposed, such as YOLO-L for vehicle license plate detection (Min et al., 2019), YOLOv3-MobileNet for detection of the electronic components (Huang et al., 2019), TF-YOLO for detection of multiple objects from aerial images (He et al., 2019), YOLO-CA for car accident detection (Tian et al., 2019), and YOLO-UA for traffic flow monitoring (Cao et al., 2019). These proposed techniques modified the network model to solve object detection problems in respective datasets and applications.

2.2 Application of YOLO model in object detection

2.2.1 Remote sensing

The potential remote sensing has become more reliable source for research area to discover many types of application by using machine vision applications for centuries.

On the other hand, the improved satellite technology has made UAVs more expendably and extreme sharp detection within nano pixel.

Due to its distinct perspective and wide coverage, the dataset of aerial images has become a crucial resource for various practical applications, including mapping. (Elkhrachy, 2021), precision agriculture (Dijkstra et al., 2019), catastrophe control management (Gupta et al., 2021), vehicle detection (Froidevaux et al., 2020; Wu et al., 2020), and others. Typically, aerial images are captured from drones, aeroplanes, or satellites, providing a unique top-down perspective that sets them apart from natural images. The unique characteristics of aerial object detection present several challenges. These challenges include variations in object scale, imbalanced numbers of objects across different categories, high levels of occlusion and truncation, relatively small objects accounting for a larger percentage, and objects with similar appearances. Object detection on aerial images poses a significant challenge due to the diverse range of deployment environments. Nevertheless, the challenges posed by non-board memory and computation, along with the delicate balance between accuracy and speed, have rendered the task of creating precise and reliable object detection algorithms exceedingly intricate.

One stage detector has received far less attention than the expanding number of research that relies on two stage detection methods for small scale remote sensing objects. This comparatively low level of interest may be attributed to the fact that, while YOLO detectors are commonly utilised in real-time computer vision applications, with a focus on achieving high detection accuracy is much lower than the region-based detectors, particularly when identifying small objects. Numerous studies have been suggested, with a particular focus on utilising YOLO-based frameworks for object detection in remote sensing images. Examples include aircrafts (Zhao et al., 2020), ships (Chen et al., 2021), and building footprints (Xie et al., 2020)

The improved algorithms mentioned earlier have demonstrated superior performance compared to the current state-of-the-art model in terms of detection accuracy. Nevertheless, the proliferation of intricate network structures has resulted in a significant expansion of parameters within the network, resulting in a substantial increase in model size. Due to hardware limitations in practical applications, the complexity of the improved networks has emerged as the primary challenge. Despite their limited use in remote sensing applications, the high efficiency of the YOLO detector lies in its suitability for real-time applications.

2.3 Lightweight models

The balance between accuracy and speed has been extensively studied in terms of computational expenses for speed and memory usage. Many of the current deep CNN structures have intricate network architectures and produce a larger quantity of network parameters. Typically, these tasks demand significant GPU computational power and consume a considerable amount of energy. Given the constraints of memory and processing power on non-GPU and embedded devices, achieving accurate and real-time object detection poses a significant challenge. In relation to this matter, the focus is on developing networks that are both efficient and compact, aiming to enhance detection speed without compromising detection accuracy. Throughout the years, academics in the field have explored various methods to create a more efficient model. These methods include factorising convolutions, group convolution, depth-wise separable convolution, bottleneck design, and neural architecture search (Zou et al., 2023). These mathematical models possess network frameworks that are simpler and more efficient, requiring fewer variables. Due to their efficient use of memory and computing resources, they are well-suited for applications on embedded and mobile devices. Object detection models often incorporate lightweight CNN methods in the field of object detection. Additionally, Tiny-SSD (Womg et al., 2018) and lightweight YOLO series such as YOLOv2-tiny (Redmon

& Farhadi, 2017), YOLOv3-tiny (Redmon & Farhadi, 2018), YOLOv4-tiny (Bochkovskiy et al., 2020) were developed replacing the original components in SSD or YOLO networks with lightweight backbones and detecting heads. A lightweight object detection methods have developed practical applications in various fields, such as identifying vehicles (Chen et al., 2020; Zhang et al., 2019), face detection (Luo et al., 2021), crop detection (Koirala et al., 2019), etc. Although there has been a notable increase in detection speed, there is still room for improvement in terms of accuracy of detection. This can be attributed to the challenge of effectively handling tiny objects. The influence of speed and accuracy on mission objectives for individual engineering applications must be considered.

Although significant progress has been achieved in recent years for lightweight object detection models, there is still a major speed gap, notably for the detection of small targets.

2.4 Vehicle detection methods based on Computer Vision

It is widely recognised that lighter models tend to be smaller and can achieve faster detection, albeit at the expense of lower detection accuracy. In order to address this issue, numerous methods have been suggested to enhance both the precision and efficiency of the model, as well as to reduce its size and detection time. In recent years, academics have explored various network configurations to create a lightweight object detection model that works on resource-limited channels. Regarding the YOLO-based model, numerous alterations to the framework of networks have been suggested.

2.4.1 Compression method

The works by (Chen et al., 2019; Junos et al., 2022; Mandal et al., 2019) applied VEDAI dataset (Razakarivony & Jurie, 2016) for automated multi-class vehicle detection. In the study conducted by (Junos et al., 2022), a more advanced CNN model,

built upon YOLOv4-tiny, was utilised to effectively detect smaller objects of interest. Model pruning, such as mobile inverted bottleneck, was used in the backbone to reduce the computational complexity of the model. However, the mean average precision (mAP) was 53.11% for the VEDAI dataset. In the work by (Mandal et al., 2019), the challenges to detect aerial images using the existing CNN models were addressed, and AVDNet model was introduced. Nevertheless, the AVDNet algorithm is not suitable for embedded devices due to heavy computational parameters in the architecture. An improved CNN model was proposed in the work by (Chen et al., 2019) on the VEDAI dataset to detect small target objects. On the other hand, a normalization-based attention module (NAM) was used for enhancing the YOLOv5s algorithm in the study by (Wang et al., 2023) to detect small target objects for vehicle detection on the UA-DETRAC dataset. However, the developed algorithms provided low performance for detecting multiple target objects. Automatic vehicle detection is highly demanding due to dynamic environment. In addition, the constraints of on-board memory and computation, along with the need to balance accuracy and speed, have posed significant challenges in the development of precise and reliable object detection algorithm development.

Nowadays, researchers have performed various network configurations on lightweight YOLO models for use on platforms with low resources. In the realm of computer vision and object detection methodologies, an array of innovative approaches has been devised to address diverse challenges, such as the intricate task of shadow detection exemplified by SEAT-YOLO, a novel architecture integrating Squeeze-Excite and Spatial Attentive mechanisms (Kumar, 2023) within the You Only Look Once (YOLO) framework. The application of μCT technology in tandem with the specialized R-YOLOv7-tiny model has further extended the capabilities of computer vision to discern minute endosperm cracks in soaked maize, showcasing the prowess of this tailored approach in agricultural research (Jiao et al., 2023). Meanwhile, the domain of traffic sign detection has seen

remarkable advancements with the introduction of an improved traffic sign detection model hinging on the YOLOv7-tiny architecture, contributing to enhanced road safety and traffic management systems (She et al., 2023). Furthermore, in the maritime arena, an innovative SAR ship detection method has been developed, leveraging an improved YOLOv7-tiny model to efficiently identify and locate ships in synthetic aperture radar (SAR) imagery, showcasing the adaptability of YOLO-based architectures across varied domains (Liu & Wang, 2022). Venturing into underwater environments, U-YOLOv7 emerges as a specialized network meticulously designed for the detection of underwater organisms, thereby facilitating ecological studies and environmental monitoring with its unique capabilities (Yu et al., 2023). Agricultural landscapes, too, have witnessed a tailored solution in the form of a peanut and weed detection model, aptly named BEM-YOLOv7-tiny, exemplifying the versatility of YOLO-based architectures in precision agriculture (Hua et al., 2023). Lastly, within the critical domain of power transmission line safety, an optimized YOLOv7-tiny model has been fine-tuned for smoke detection, exemplifying its utility in early fire detection and prevention, thereby contributing to the robustness of power infrastructure (Chen et al., 2023). Table 2.1 summarizes the application of YOLOv7-tiny models in previous works.

Table 2.1 Application of lightweight YOLOv7 model in various works

Work	Method	Application	Number of Classes	Accuracy (%)
(Kumar, 2023)	Three squeeze-excite blocks within feature extraction network	Shadow detection	1	59.53
(Jiao et al., 2023)	The SiLU activation function replaced the LeakyReLU activation function, the GhostConv module replaced the Conv module, and the	Detecting endosperm cracks	6	92.10

	CoT block and C3_TR module was added to the model's neck and backbone sections.			
(She et al., 2023)	Down-sampling module: Slice-Sample	Traffic Sign Detection	1	93.47
(Liu & ang, 2022)	Improve SPP of the backbone network, additional coordinate attention mechanism and well-designed SIoU	SAR Ship Detection	1	92.15
(Yu et al., 2023)	1.Combination CrossConv + efficient squeeze-excitation module. 2. Content-Aware ReAssembly of FEatures (CARAFE). 3. A decoupling head using hybrid convolutional	Underwater organism detection	6	84.4
(Hua et al., 2023)	1.ECA and MHSA modules focus on predicted targets. 2.The BiFPN module and SIoU loss function increase the convergence speed and efficiency,	Peanut and weed detection	2	92.4
Chen et al., 2023)	1.Multiple parameter free attention modules. 2.Spd-Conv (Space to depth Conv) to improve detection accuracy and speed.	Smoke detection	1	79.47
(Bai et al., 2024)	1.Swin Transformer prediction was construct 2.The GS-Elan Optimisation Module for neck section	Flower and fruits on strawberry seedlings detection	2	92.1

Hong et al., 2020)	1.The three-level feature MAPs generated by the backbone network 2.A feature encoding and decoding structure module is proposed	Traffic Surveillance	4	68.01
e & Wang, 2023)	1.RT-YOLO algorithm enhanced the multi-scale fusion strategy based on YOLOv7. 2. Added NAM into the backbone and neck-network to identify the region of interest.	Crowded Pedestrian	10	94.3
Yang et al., 2023)	Deformable convolution, the Biformer dynamic attention module mechanism, new decoupling head	Tea Tree Pest Detection	5	93.23

As reported in previous works, YOLOv7-tiny model has shown good detection performance and precision when applied on aerial images for vehicle detection applications. This work endeavours to bridge existing gaps in the literature by modifying the YOLOv7-tiny to address the limitations observed in the conventional YOLOv7-tiny model. The proposed improvements are tailored to enhance scaling methodology within the model architecture, thereby optimizing precision by effectively leveraging features across different scales within the input data. Hence, the performance of the object detection can be improved.

2.5 General structure lightweight YOLOv7 based model

YOLOv7-tiny is a variant of YOLOv7 (You Only Look Once) object detection framework that targets lightweight and real-time applications, where computational

efficiency and quick response time are critical. It was specifically created for situations where lightness and speed are essential in its applications. YOLOv7-tiny can be deployed on devices that have limited resources such as mobile phones, or embedded systems. The main design idea of YOLOv7 remains the same: it uses a one-stage object detector to predict simultaneously bounding boxes positions and class scores for objects found in an image. However, the tiny version simplifies its by decreasing the number of layers, filters, and parameters. As a result, it achieves smaller model size with faster inference time while maintaining competitive accuracy. The network is made up of two parts; a backbone which performs feature extraction using few convolutional layers on average and a head that does detection through a series of convolutional layers predicting object classes and bounding boxes at multiple scales. Besides it also applies advanced methods like multi-scales features fusion and attention mechanism though in compact form to balance between speed versus accuracy trade-offs. The YOLOv7-tiny model works well in situations where it needs to detect objects or perform detection tasks quickly in edge computing or other low latency settings. YOLOv7-tiny, similar to YOLO models consists of two parts: the backbone and the head. As shown in figure 2.2 General structure for improvement on the YOLO-based model.

2.5.1 Feature Extraction Network

The core component, in YOLOv7-tiny is responsible for extracting features from the input image. Is designed to be light yet powerful. It typically uses a series of layers with reduced depth and fewer filters compared to its counterparts to minimize computational load and memory usage. These layers are structured to capture features of the image starting from edges and textures in the early layers to more complex patterns and object parts in the deeper layers. The core may incorporate methods like connections to enhance feature propagation and learning efficiency without increasing the networks complexity.

The improvement process includes integrating the Head (DyHead) module, which enhances the model's capability to detect objects by improving scale and spatial awareness. The DyHead effectively captures object sizes making it especially valuable in applications that require detection of small or intricate objects, such as in printed circuit boards or steel defect detection tasks. Furthermore these enhancements often involve pruning techniques to reduce the models size and computational requirements making YOLOv7-tiny an efficient choice, for real time applications (Zhang et al., 2024).

2.5.2 Feature Fusion Network

On the other hand, YOLOv7-tiny is designed head, head for object recognition and works by taking the feature maps produced by spines and using multiple variable layers to set bounding boxes, objectless scores and class probabilities at scale the prophecy of the various. Thus, often uses anchor boxes, which are predefined shapes that the network changes as it is being optimized using for well-known objects YOLOv7-small heads are streamlined, focusing on speed and consistent performance effectively, thus forgoing the more complex dynamics found in larger images but still including some important features such as multi-scale predictions. There were potential attenuation mechanisms to refine the search process the careful design of the spine and head in YOLOv7-tiny ensures that the model is balanced in maintaining real-time performance to achieve high detection accuracy, making it suitable for applications where speed and resources are loaded use it properly is most important

These YOLOv7-tiny models have been modified recently to improve its performance in a variety of specialized applications. One way is to optimize the UPSAMPLE block, which is important to maintain the resolution of the feature maps during the upsampling process of the model. This variable is especially important for tasks that require the detection of small objects in images such as aerial imagery or surveillance scenarios

For times, in a have a look at targeted on detecting smoke in energy transmission traces, the modified YOLOv7-tiny model turned into optimized to decorate detection accuracy while maintaining real-time performance. This optimization possibly worried adjustments to the upsampling manner to better maintain spatial facts, which is vital for detecting small and first-rate details like smoke in complex backgrounds. Additionally, any other examine evolved a version of YOLOv7-tiny tailored for aerial automobile pixel, in which similar changes to the upsampling block have been vital for improving the detection of small and remote (Nguyen et al., 2023) (Chen et al., 2023). These modifications generally aim to enhance the model ability to handle the trade-off between maintaining high detection accuracy and the computational efficiency required for real-time application.

2.5.3 Prediction Network

The prediction network in YOLOv7-tiny is an exceptional development of the YOLOv7-tiny system, designed for conditions requiring high accuracy in object detection, especially in complicated or important environments. Predictive links construct on this through at once integrating the knowledge into the prediction method. This integration can also contain domain-specific rules, constraints, or models representing physical or recognized objects, shapes, or motions which can be unique to the found environment, such as geographical styles in satellite imagery or unique anatomical implants (Bayram et al., 2022). The network complements YOLOv7-tiny widespread characteristic extraction and prediction abilities by way of using advanced techniques like multi-scale feature fusion, which lets in the model to combine records from one-of-a-kind layers and higher detect embedded device of various sizes, specifically small or much less object. It can also attention mechanisms to cognizance the model's potential at the maximum applicable features, improving detection accuracy wherein conventional models might falter. By integrating those insights and advanced strategies, the prediction

network on YOLOv7-tiny not only retains the unique model pace and performance however additionally substantially boost its accuracy and robustness, making it especially suitable for packages that require unique and reliable object detection.

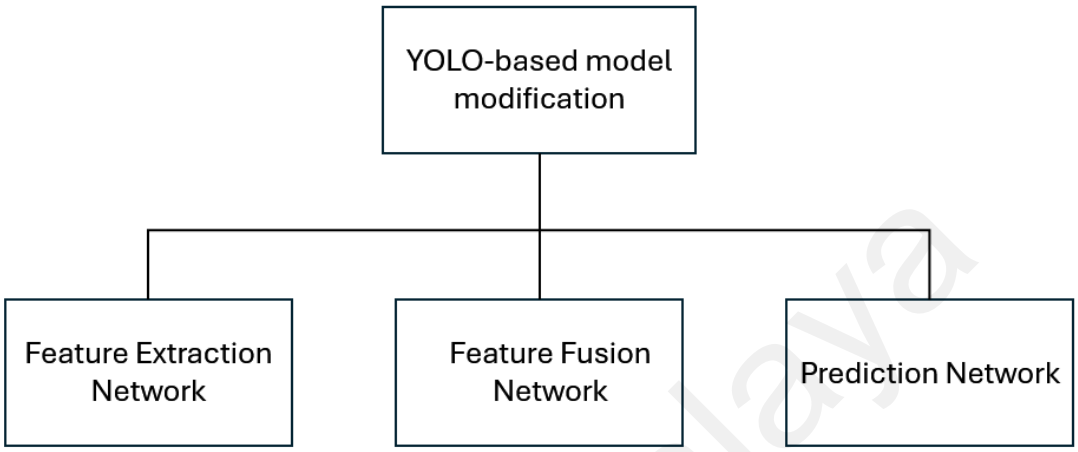


Figure 2.2 General structure for improvement on the YOLO-based model.

2.6 Summary

The literature shows that improvements were made in the three sections of the YOLO model, including the backbone, neck, and prediction layer. Each of these section plays an important role in the performance of object detectors. In order to develop an accurate and efficient one-stage object detection model, several optimization techniques were proposed in each of these sections. Generally, the backbone network is responsible for approximately 90% of the storage and calculation of the detectors. Hence, lightweight networks were adopted in the backbone section to reduce the GLOPs value and number of parameters and strengthen the extraction of feature maps. In addition, several researchers remove unnecessary layers by using channel pruning and compression method. Despite generating a more efficient model, the detection accuracy is declined substantially. The methods adopted works have shown significant improvement in the

lightweight YOLO model in terms of accuracy and efficiency. However, the detection accuracy is still considered low and unsuitable for implementation in a critical application that requires precise object recognition, such as traffic monitoring, precision agriculture, surveillance, robot vision and others. However, some improvements can be made to further improve the model's accuracy while adopting these modules. Therefore, this has contributed to the motivation of this work to develop a lightweight object detection model that has a good balance in terms of accuracy and efficiency.

CHAPTER 3: METHODOLOGY

3.1 Introduction

The detection model was formulated through three primary stages. Initially, the datasets were obtained from various sources and the objects in the image were marked with bounding boxes. Next, the YOLO network underwent optimisation and training using the relevant datasets. Furthermore, evaluation metrics were calculated to validate the performance of the detection. After careful evaluation, the most optimal model was chosen for object detection. Figure 3.1 displays the pipeline of the proposed methodology for object detection.

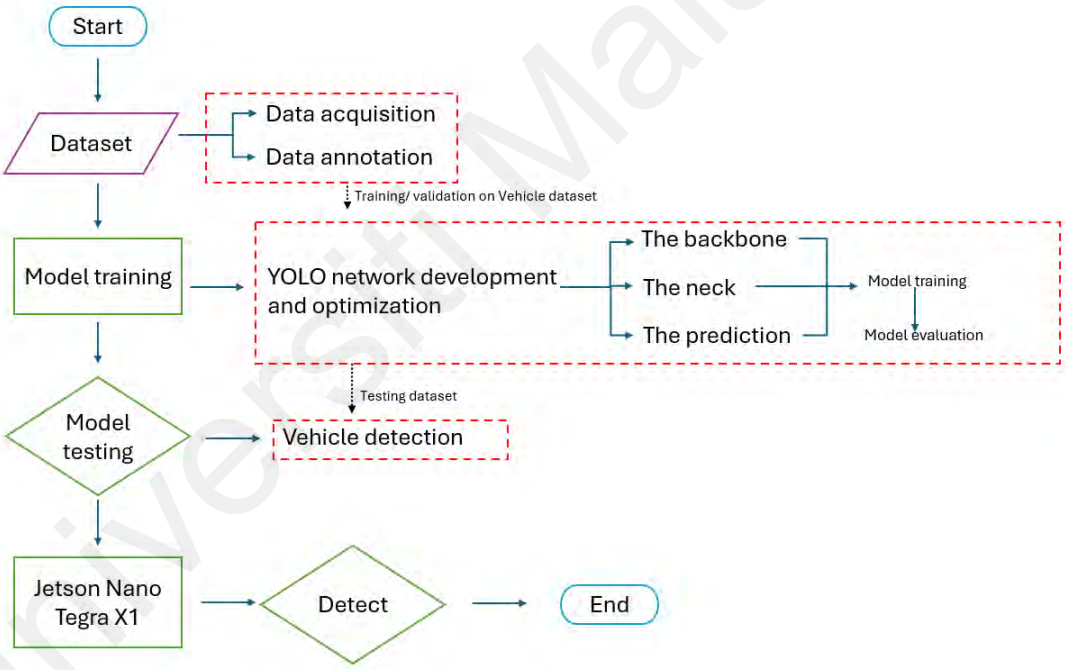


Figure 3.1 General structure of the proposed method

3.2 Dataset to use for experimental

This research incorporated two distinct datasets. VEDAI dataset and UA-DETRAC dataset were obtained from a public dataset. Table 3.1 provides a detailed of each dataset. All dataset was annotation by manually one by one images using labelImg application in

YOLO-series to annotate square box out specific vehicle type and size each time before train on Pytorch.

Table 3.1 Detailed description of each dataset

Dataset	Number of classes	Training images	Test Images	Total
VEDAI	12	1250	15	1250
UA-DETRAC (NIGHT)	4	15200	20	15200
UA-DETRAC (RAINY)	4	12687	20	12687

3.2.1 VEDAI dataset

In this study, the UAVs VEDAI dataset was chosen to test the developed CNN model. For the training and testing phase, a total of 1250 images were utilised from the VEDAI dataset. A total of 1250 annotated images were utilised for both training and testing purposes in this object detection experiment. The objects were classified into 12 different categories for training purposes. These categories included car, truck, van, tractor, pickup, camping car, plane, boat, and other. Given the nature of the dataset, the images contain small objects belonging to various classes. As shown in figure 3.2 example VEDAI dataset images.

Figure 3.2 VEDAI dataset images

3.2.2 UA-DETRAC dataset

University at Albany Detection and Tracking (UA-DETRAC) (Wen et al., 2020) is a public dataset which was created by researchers for multi-object tracking (MOT) (Sun et al., 2022). The dataset was created using 100 real world traffic videos on different weather conditions for detecting and tracking purposes. The videos were captured from different altitudes and orientations to enrich the dataset augmentation. In this study, two sets of data from UA-DETRAC dataset were used: Night and Rainy. These were four classes of vehicles in that dataset which are car, Bus, Van, and Others. Motorcycles and Humans were not categorized as annotation from the dataset. For this study, a total of 15,200 images from the Night dataset and 14,580 images from the Rainy dataset were utilised for training and testing. As shown in figure 3.3 and figure 3.4 example UA-DETRAC dataset images.

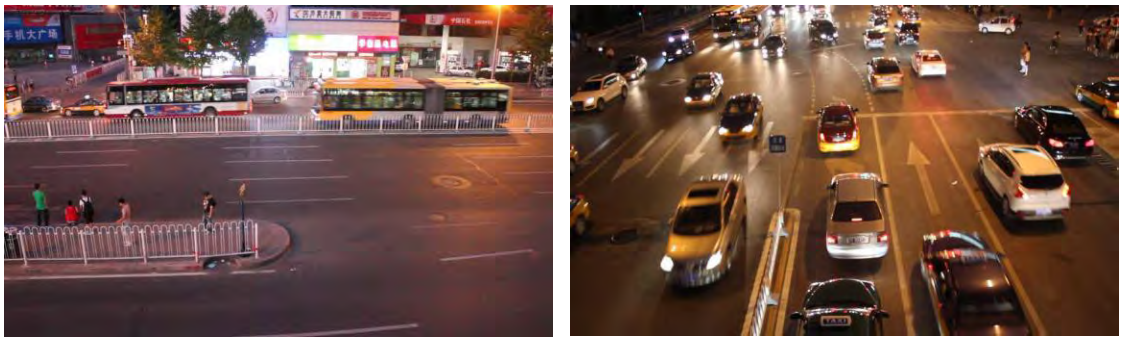


Figure 3.3 Night images from UA-DETRAC dataset

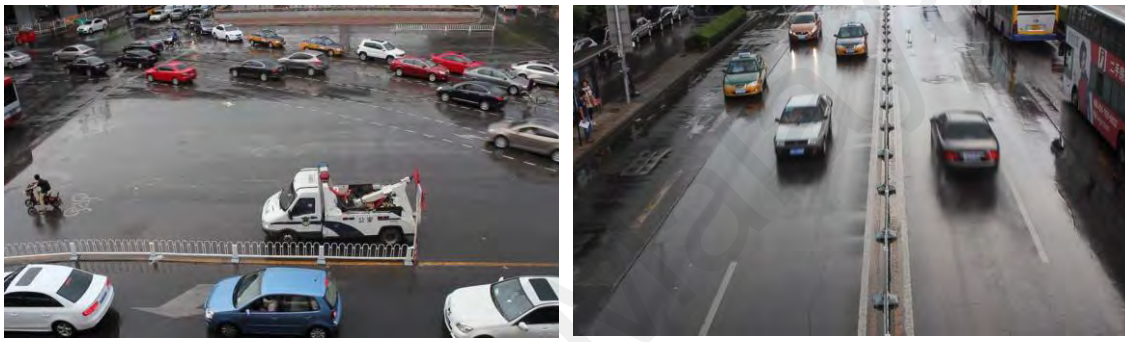


Figure 3.4 Rainy images from UA-DETRAC dataset

3.2.3 Data annotation

An open-source annotation tool called LabelImg (García-Aguilar et al., 2023) was used for the annotation process of images in the VEDAI and UA-DETRAC datasets. For this study, the VEDAI dataset and UA-DETRAC dataset underwent meticulous annotation. Precise bounding boxes were meticulously drawn around the objects in the images, and each object was meticulously classified into its respective category. Each of the apparent object in each image was labelled with a bounding box that represents the object's location. Images with insufficient or ambiguous pixel areas were not labelled.

3.3 Preliminary methodologies

In this work, the proposed model aims to improve small object identification accuracy by modifying the conventional YOLOv7-tiny architecture. VEDAI dataset use in this work will be name on proposed model as generally to memorize future improvement.

According to YOLOv7, the tiny algorithm YOLOv7 maintains the cascade-based model scaling technique while enhancing the effectiveness of the long-range aggregation network (ELAN-VD) to achieve higher detection accuracy using smaller parameters and quicker detection time and the (UPSAMPLE-VD) on head architecture is improvised resolution to improve the detection accuracy of small vehicles in the aerial image. The fundamental basis for YOLOv7-tiny algorithm includes feature extraction network, feature fusion network, and prediction network.

3.4 Development of the improved algorithm

A few adjustments are made to get a decent balance between detection time and accuracy. The design of the proposed model includes two proposed features: (1) an enhanced efficient long range aggregation network module VEDAI (ELAN-VD) and (2) a modification improvised resolution (UPSAMPLE-VD) model. As shown in figure 3.5 displays the block diagram of the proposed model.

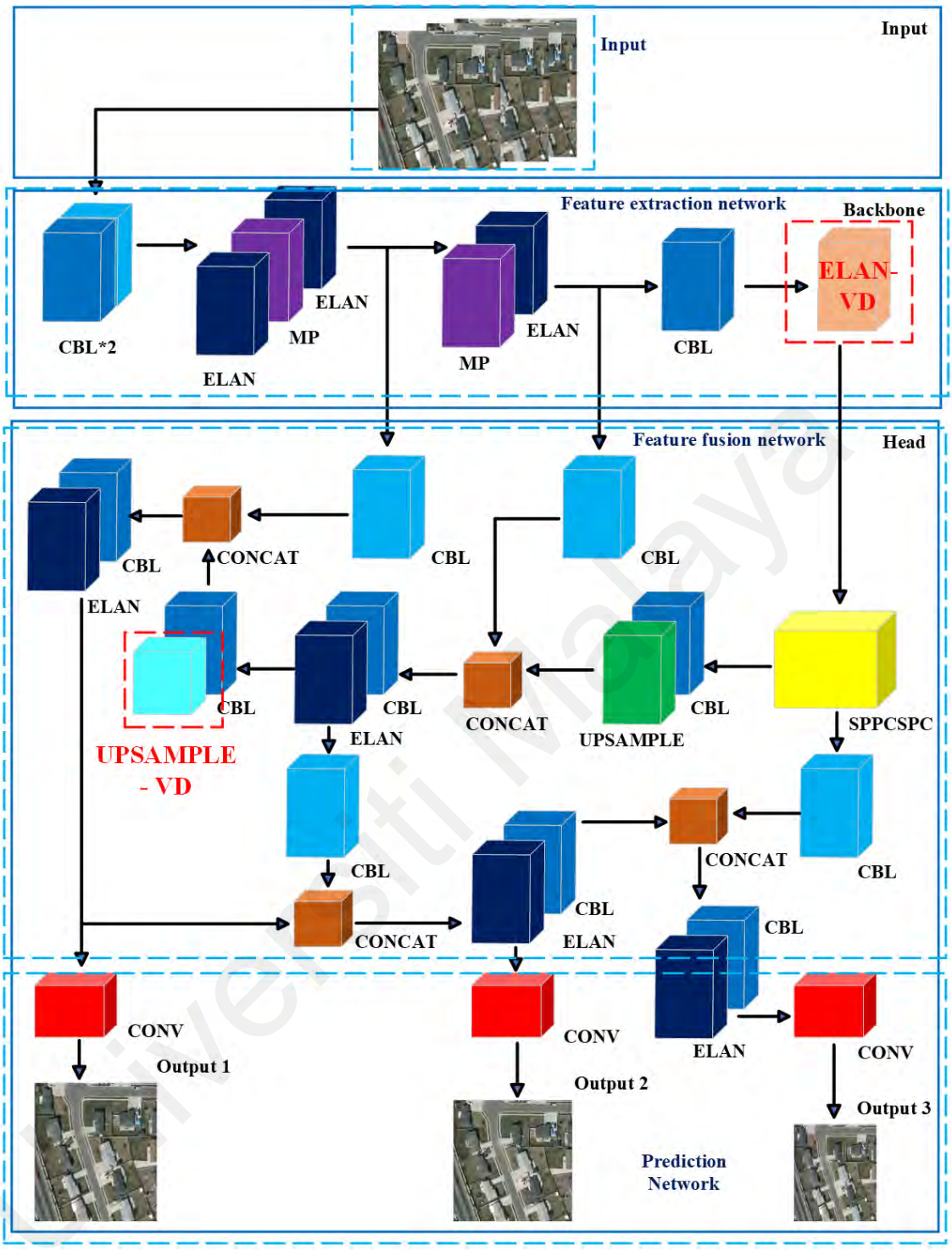


Figure 3.5 Block diagram of the proposed method

3.4.1 Improved Feature Extraction Network

The differences between feature extraction network (FEN) and feature fusion network (FFN) are their objectives and architecture strategies. FEN is applied to extract features

from the vehicle images by adopting hierarchical layers to transform input data into increasingly abstract representations. The emphasis is on extracting intrinsic feature from each vehicle image. On the other hand, FFN focuses on integrating the features from various scales to enhance overall detection performance by combining information from multiple CNN layers.

Generally, the feature network aims to accomplish multi-scale learning with robust semantic data from the highest level of the feature pyramid network (FPN) (Tan & Le, 2019; Zhang et al., 2018). However, the nearest neighbour interpolation up-sampling is insufficient to strike a suitable balance within the detection task speed and accuracy thresholds. Beside that, YOLOv7-tiny feature fusion network tensor splicing might not be sufficiently thorough to fuse feature details from neighbouring layers. Furthermore, feature information loss may occur if, to a sufficient degree, the fusion network does not priorities the desired feature information. Nevertheless, the YOLOv7-tiny present detection header uses conventional convolution, which can lead to unfocused feature fusion outcomes that necessitate a targeted strategy to improve the efficacy of tiny target identification.

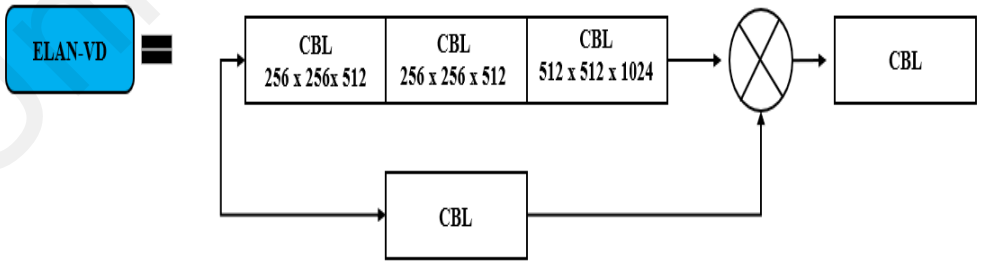


Figure 3.6 Structure of the proposed ELAN-VD

Therefore, this work proposes modifications on the YOLOv7-tiny algorithm to improve the detection accuracy. The feature extraction network includes CBL

convolutional blocks, an improved efficient long-range aggregation network (ELAN-VD) layer, and MP Conv convolutional layer. The ELAN-VD is proposed in the feature extraction network which consists of convolutional blocks of CBL. The structure of the proposed ELAN-VD is shown in Figure 3.6. In ELAN-VD, two CBL blocks of 256x256x512 pixels are connected to one CBL block of 512x512x1024 pixels. On this method, it utilizes in the feature extraction module to replace normal convolutional block to gain capability on high precision training. As shown in figure 3.6 Structure of the proposed ELAN-VD.

In addition, the neck adopts multi-scale feature fusion model of the Path Aggregation Network (PAN) structure to incorporate low-level spatial information and high-level semantic features. This will then preserve more information to improve the vehicle detection accuracy of small targets. The multi-scale features refer to representation of the input signal, which is the vehicle image, that encompasses information extracted at various levels of granularity or resolution. The concept involves capturing details ranging from fine to coarse scales to enhance the vehicle detection model's ability to understand complex patterns. Scale pyramids involve constructing representations of the input at various levels, enabling the extraction of features at both local and global contexts. These features are then integrated to form a comprehensive and multi-scale understanding of the input data which will aid the detection process. The fused features will contribute to the network effectiveness in detecting vehicles of different sizes, including small targets.

Moreover, multi-scale feature fusion enhances the network’s contextual understanding by leveraging both local and global contextual information which is helpful for identifying vehicles from complex backgrounds. This is because, fusing features from multiple scales enable the network to learn more discriminative representations which will improve the robustness against scale variations and occlusions. Meanwhile,

upscaling aids the alignment and combination of features extracted from smaller scales to correspond the size of features from larger scales. Thus, this approach can lead to a more comprehensive understanding of the scene, especially in detecting small-target vehicles.

3.4.2 Improved Feature Fusion Network

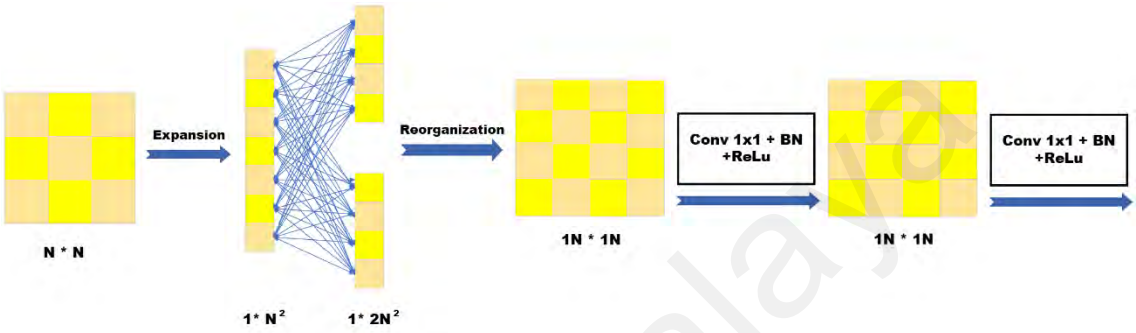


Figure 3.7 Structure of the proposed UPSAMPLE-VD

In YOLOv7, the UPSAMPLE block is used to increase the spatial resolution of feature maps from deeper layers, following them to be combined with higher resolution features from shallower layers. This process enhances the network ability to detect of varying sizes, particularly small objects, by focusing detailed spatial information with rich sematic features. The UPSAMPLE block typically involves bilinear interpolations, followed by concatenation and convolution, and is crucial for achieving high detection accuracy in the network's multi-scale detection network.

On the other hand, this work also proposes modifications on the Feature Fusion Network to improvised higher resolution features within shallow layers. The feature fusion network includes several element UPSAMPLE block that is deconvolution, ReLU, 3x3 Conv and PA Block which can apply to refine the features and adjust the channel dimensions. This helps to smooth and enhance the upsampled features, ensuring they are well-suited for detection. The 3x3 convolution preserves spatial context while reducing noise, which is crucial for maintaining high detection accuracy across different scales by

adding more channel dimension it will increasing the refine layer on 3x3 Conv block so that more robustness and high-density during training model time. The UPSAMPLE-VD is proposed in the feature fusion network which consists of 3-unit 3 x 3 convolutional blocks. The structure of the proposed UPSAMPLE-VD is shown in Figure 3.7. In UPSAMPLE-VD, 3x3 Conv blocks basically make the. In this result show in Chapter 4 have significant boost the precision percentage 19.53% from first proposal model result.

In meantime, when apply toward to low embedded device will get increased chance on detection real-time high precision output result. It has shown how great improvements this modification when using with any device to boost from lower quality to highly quality pixel.

3.5 Performance evaluation

The evaluation metrics applied in this study are precision (P), recall (R), mean average precision (mAP), F1 score, and frames per second (FPS). Precision is defined as the number of correct predictions made when the projected bounding boxes match the ground truth boxes. On the other hand, recall determines the probability of correctly detecting ground truth objects. True positive (TP) signifies the correctly detected objects. False positive (FP) and false negative (FN) indicate the wrongly and miss detected objects respectively.

$$Precision\ (P) = \frac{TP}{TP + FP} \qquad (1)$$

$$Recall\ (R) = \frac{TP}{TP + FN} \qquad (2)$$

Meanwhile, F1-score is a combined measure of precision and recall with 1 being the highest value. For each image, the Intersection over Union (IoU) between the predicted

bounding box and the actual bounding box can be computed by finding the ratio of the overlap area over the union area. The Average IoU is calculated by averaging the IoU over the number of data images. The area under the precision-recall curve at different detection thresholds is defined by average precision (AP). The mAP determines the mean accuracy for n classes of objects where n is the number of all categories, and the denominator is the sum of the APs of all categories. In a nutshell, AP and F1-score are important metrics to assess the performance of object detectors.

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (3)$$

$$AP = \int_0^1 P(R)\, dR \quad (4)$$

$$mAP = \frac{1}{n} \times \sum_{k=1}^{k=n} AP_k \quad (5)$$

3.6 Summary

In this study, two separate datasets are used to assess the proposed models. First dataset is acquired using an open dataset and contains aerial images with nine object classes meanwhile second dataset have only four object classes. In addition, two single-stage object detection models are proposed, which are inspired by the YOLOv7-tiny models. First proposed model: Improved Feature Extraction Network and second proposed model: the head architecture is improvised resolution to improve the detection accuracy of small vehicles in the aerial image. Furthermore, the effectiveness of the proposal methods has been benchmarked by the presented result and discussion in the next chapter.

CHAPTER 4: RESULTS AND DISCUSSION

4.1 Introduction

A series of extensive experiments were carried out in this section to assess the dependability of the proposed object detection model on various diverse datasets. First, the experimental settings were introduced. Then, the comparative experiment results were thoroughly analysed and discussed.

4.2 Experiment setup

For this study, the experiments were conducted on a Windows 10 64-bit operating system. A deep learning framework called PyTorch was utilised to train the models based on YOLO. The learned model was subsequently put into use on the NVIDIA Jetson Nano 4 GB to assess the detection speed of the simulation on embedded system devices. Table 4.1 shown the hardware and software setup.

Table 4.1 Hardware and software setup for VEDAI dataset testing and training

Model	Lenovo ThinkPad ThinkPad P15 Gen2	NVIDIA® Jetson Nano Developer Kit P3450
Processor	11th Gen Intel Core™ i7-11800H 2.30GHz	Quad-Core ARM Cortex-A57
Memory	48 GB DDR4 2933 MHz	4 GB LPDDR4
GPU	NVIDIA RTX A2000 4 GB	NVIDIA Tegra X1
DISK	KINGSTON SNVS2000GB	KINGSTON SDCS2/128GB
CUDA	11.6	11.3
PyTorch	1.12.1	1.11.0
Python	4.1.1	3.6

Standardising the hyperparameters for all YOLO-based models ensured that the results could be compared effectively across different experimental configurations. The images in each mini batch were sent to the GPU for computation. The computation process was to be train by 300 epochs as default to get efficient result and best precision-recall during training time. Using a larger number of images for averaging could enhance the training process, although it came with the drawback of increased memory demands. The VEDAI

datasets (Zhong et al., 2017) were trained for 300 epochs while the UA-DETRAC datasets (Wen et al., 2020) also were trained for 300 epochs. On the other hand, VEDAI dataset was original being taken on imagery satellite with pixel original to be testing and can be download throughout open-source platform. Meanwhile, UA-DETRAC were taken by local scientific to be use as real-time experiment on vehicle detection. This training step was well-suited for achieving optimal convergence of average loss and mAP, taking into account the dataset's image count and class count. In addition, the network parameters were adjusted by incorporating a momentum of 0.8 and a decay weight of 0.0005. This approach helps to regulate and minimise significant weight fluctuations during different epochs. During the initial phase of the training process, it was crucial to have a high learning rate as there was no prior information available. As the network gathered a substantial amount of data, it became necessary for the learning rate to decrease gradually. At last, the experiment was carried out with an input size of 512 x 512 pixels for both the VEDAI and UA-DETRAC datasets. Figure 4.1 shown YOLOv7-tiny convention structure.

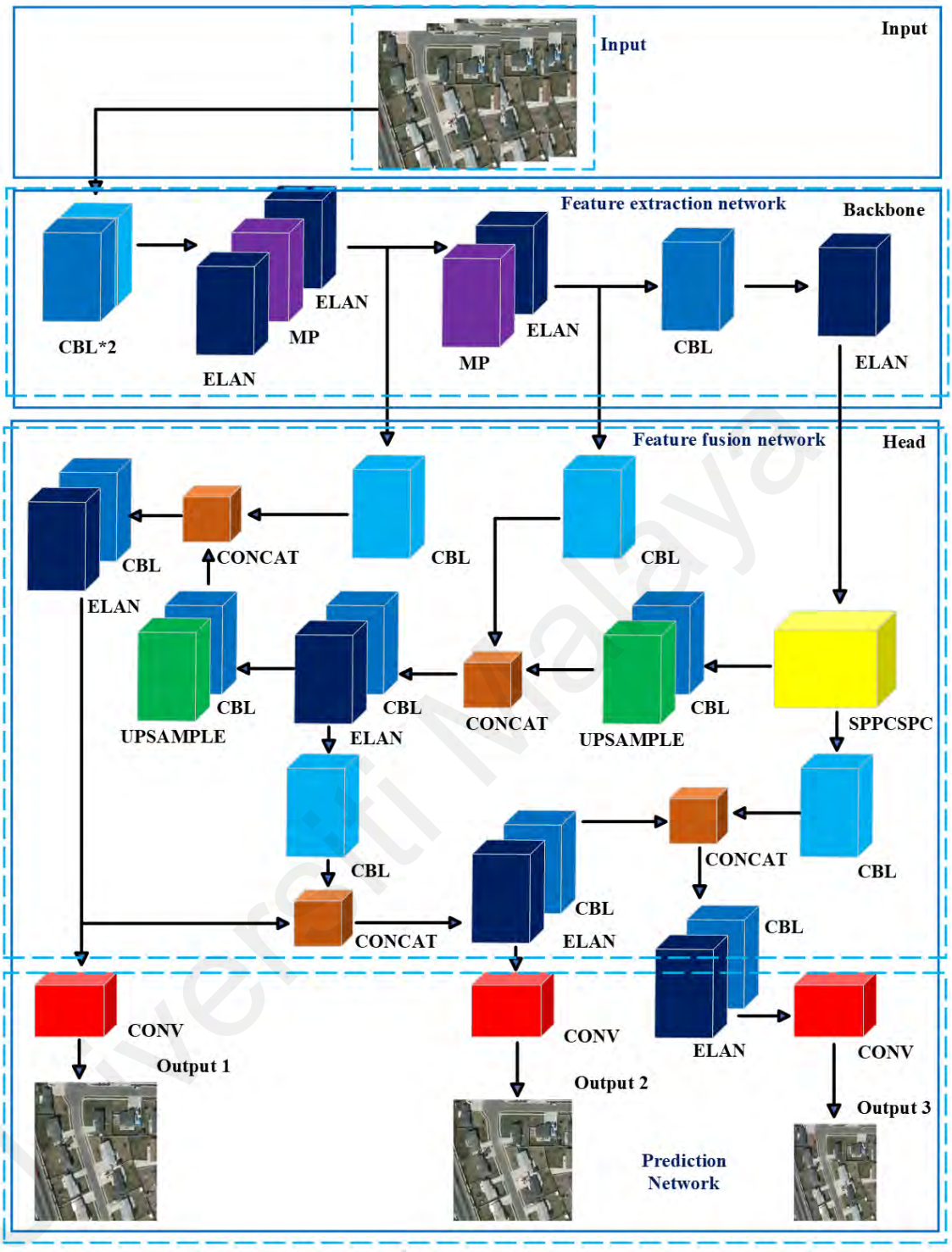


Figure 4.1 YOLOv7-tiny architecture

Table 4.2 Summary of the experimental dataset

Dataset	Number of classes	Training images	Test Images	Parameters	IoU
VEDAI	12	1250	15	6044754	0.50
UA-DETRAC (NIGHT)	4	15200	20	11531202	0.50
UA-DETRAC (RAINY)	4	12687	20	11531202	0.50

4.3 Detection performance on different modifications

4.3.1 First proposed model (ELAN-VD)

These experiments were conducted to investigate upscale method on the performance of the model. The first proposed model is developed through modification in ELAN block. The baseline mode was the YOLOv7-tiny model. The comparison of the experimental results evaluated on the VEDAI dataset is highlighted in Table 4.3. As shown in the table, that the conventional YOLOv7-tiny model achieved the lowest mAP among the other models with 48.89 %. This is mainly due to the shallow backbone structure network with several layer detection only leading to more false-positive and false-negative objects.

The feature extraction network includes CBL convolutional blocks, an improved efficient long-range aggregation network (ELAN-VD) layer, and MP Conv convolutional layer. The ELAN-VD was proposed in the feature extraction network which consists of convolutional blocks of CBL. The structure of the proposed ELAN-VD is shown in method 3.4.1. In ELAN-VD, two CBL blocks of 256x256x512 pixels are connected to one CBL block of 512x512x1024 pixels. On each block contains multiple increasing upscaling compare with original that was two CBL blocks of 128x128x256 pixels are connected to one CBL block of 256x256x512 pixels by doing so can obtain significantly boost resolution precision detection. Although the ELAN-VD layer speeds the extraction of features, it may decrease feature extraction capacity because, it removes two feature

computational blocks from the conventional YOLOv7-tiny algorithm. The result shows that the mAP increases by nearly 9 % compared to the YOLOv7-tiny model. The use of a feature reuse mechanism allowed the extraction of richer information from each layer in the network. However, the first proposed model size increases by 0.1 times while the detection time increased from 0.573 ms to 0.46 ms. This caused by a higher number of convolution layers in the network.

Finally, to further improve the accuracy, the first proposed model used CIoU as the localization loss function. As a result, the mAP increases by 9.05 % with significant change in model size and detection time. Overall, the first proposed model has the advantage of good detection performance at considerably small model size and improved inference time.

4.3.2 Second proposed model (UPSAMPLE-VD)

Table 4.3 shows the experimental results of several ablation experiments evaluated on the VEDAI dataset. The baseline model of the YOLOv7-tiny model was optimized in the VEDAI dataset, including the backbone, neck, and detection scale. An additional detection scale was included in second proposed model while retaining the other network section. The results shows that the mAP increased to 77.47 %, with an increment of 28.58 % from the original model. The multi-scale detection helped to improve the detection of different sizes of targets by extracting multi-scale features from different sizes of the receptive field. Besides, the inference time also increased by approximately 5.5 %.

This shows the great contribution of the 3x3 Conv block layer to the improvement pf the overall performance of the model. Finally, the second proposed model adopted the UPSAMPLE module that linked the feature fusion network to the second detection layer to further improve the detection accuracy. As the result, the model achieved an increment in mAP by 77.47 % compared to the original YOLOv7-tiny model. Nonetheless, the

detection time is slightly increased by 28.58 % than the original model. Overall, the second proposed model achieved an excellent real-time detection performance at a small model size.

4.4 Performance evaluation on different datasets

In this phase, the models proposed were compared and analysed against the top performing single stage detection models and previous studies to validate their effectiveness in terms of detection accuracy. The one-stage detector has opted for YOLOv7 and YOLOv7-tiny. The evaluation metrics were calculated using a standard IoU value of 0.5. The experiment results were evaluated on two distinct datasets: VEDAI and UA-DETRAC dataset.

4.4.1 VEDAI dataset

The Utah, USA, spring of 2012 yielded the VEDAI dataset, which offers a variety of small vehicle types in various environments conditions. The VEDAI dataset contains objects with different attributes, such as multiple orientations, varying illuminations, and occlusions. In addition, this database includes a variety of backgrounds, such as urban, peri-urban, rural, and other diversified settings. This experiment employed an image with dimensional of 1024 by 1024 pixels and a spatial resolution of 12.5 cm.

4.4.2 Performance evaluation on VEDAI dataset

This experiment utilised 512 x 512-pixel input pictures. Table 4.4 compares the proposed model detection performance with several other models using the VEDAI dataset. Based on the findings of the experiment, the proposed model achieved a mean average precision (mAP) of 77.47 %, which is higher than other lightweight models but lower than the conventional YOLOv7-tiny model (48.89%). In addition, the proposed model shows a 28.58 % increase in mAP compared to the conventional YOLOv7-tiny model. This study demonstrates the effectiveness of the additional ELAN-VD module

and UPSAMPLE-VD module by incorporating both local and global characteristics to expand the reception field and significantly segregate important content aspects.

In addition, the connections between the layers in the proposal model have greatly enhanced its ability to accurately identify smaller objects. Therefore, this led to a notable increase in the accuracy of detection, especially when identifying targets that have significant variations in size within an image. Compared to conventional YOLOv4-tiny, conventional YOLOv7-tiny exhibited a notable improvement in mAP. As a result of their lower detection and convolution layers, these two models, however, built for quick detection performance, required help identifying tiny objects. Conversely, the suggested strategy was not as effective as the usual YOLOv4 and YOLOv7, with mAP of 73.38% and 68.75% respectively. However, the YOLOv4 and YOLOv7 conventional models with developed using intricate backbone topologies, with YOLOv4 using left over connections as a DarkNet53 backbone. In the CSPDarknet backbone, YOLOv4 concurrently employs a cross-stage partial connections method. Therefore, these two traditional approaches are inappropriate to use with constrained computing resources.

However, this overlap percentage between bounding boxes and ground truth is considered reasonable, considering the higher mAP number. The first proposed model recall value increased dramatically to 0.59 and precision value of 0.64 meanwhile second proposed model achieved more than 0.70 and precision value of 0.67. This suggests that the proposed model can identify more tiny targets compared to the conventional YOLOv7-tiny model. The F1-score value of 0.67 was higher on the second proposed model rather than F1-score found on first proposed model as value of 0.61 due to the vital balance between recall and accuracy.

Table 4.3 Results of comparison for the VEDAI dataset

Model	Number of classes	Input size	Precision	Recall	F-1 score	Avg IoU (%)	mAP (%)
Modified cascade CNN (Zhong et al., 2017)	6	-	-	-	0.32	-	54.6
Modified CNN (Ju et al., 2019)	9	512	-	-	-	-	47.8
YOLOv3 (Junos et al., 2022)	9	512	0.7	0.71	0.69	54.45	61.79
YOLOv3 tiny (Junos et al., 2022)	9	512	0.51	0.5	0.51	36.52	37.63
YOLOv4 (Junos et al., 2022)	9	512	0.71	0.77	0.74	56.58	73.38
YOLOv4 tiny (Junos et al., 2022)	9	512	0.53	0.54	0.54	40.51	47.8
Modified YOLOv4 tiny (Junos et al., 2022)	9	512	0.62	0.57	0.59	49.73	53.11
Modified YOLOv4 tiny (Momin et al., 2022)	9	512	0.59	0.53	0.56	42.89	52.61
YOLOv7	12	512	0.8	0.62	0.69	-	68.75
YOLOv7 tiny	12	512	0.62	0.43	0.50	-	48.89
First proposed model (ELAN-VD)	**12**	**512**	**0.64**	**0.59**	**0.61**	**-**	**57.94**
Second proposed model (ELAN-VD + UPSAMPLE-VD)	**12**	**512**	**0.83**	**0.70**	**0.67**	**-**	**77.47**

Figure 4.1 illustrate how recall and accuracy measurements relate to each other for every detection model. The proposal model has a slightly steeper P-R curve than the standard YOLOv7-tiny. This results in a larger area beneath the graph, which raises the mAP value.

The second proposed model achieved a higher mAP of 77.47 %, suggesting that it outperforms the models proposed in previous studies, as discussed in (Ju et al., 2019; Junos et al., 2022; Momin et al., 2022; Zhong et al., 2017). The suggested model produced superior results compared to earlier studies, with all twelve classes detected and a respectively detection performance of 77.47 %.

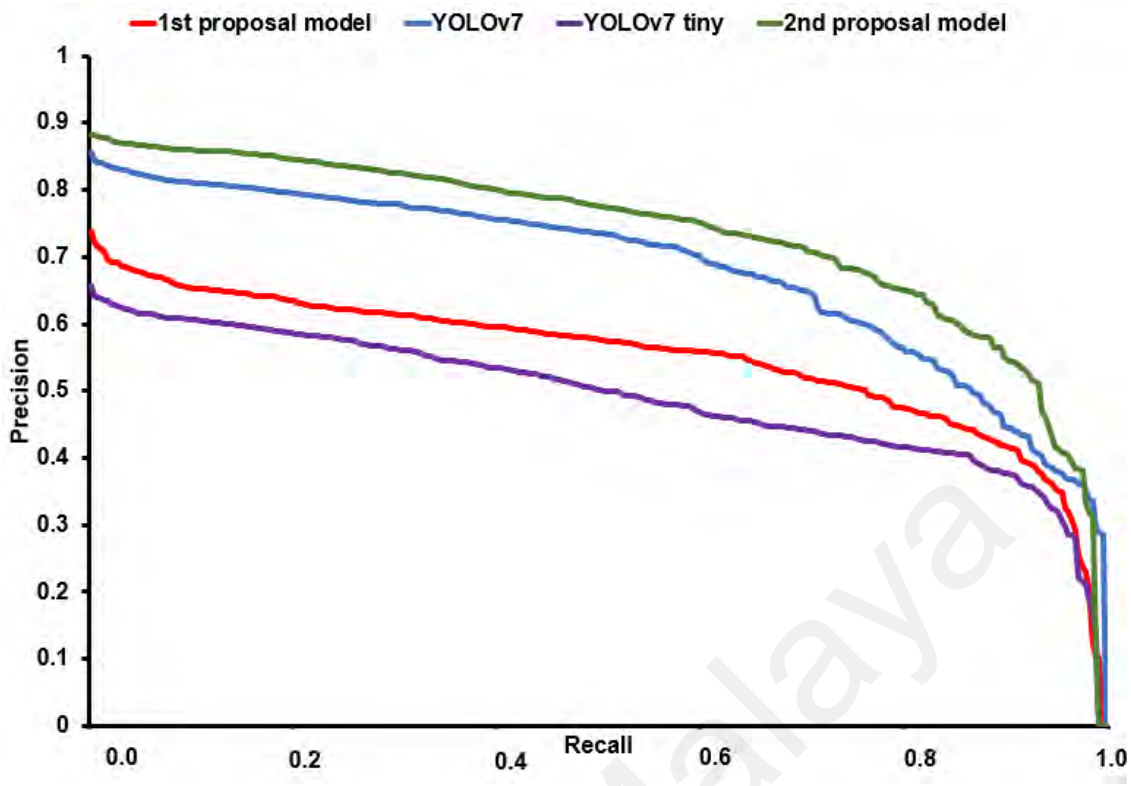


Figure 4.2 An analysis of precision-recall curves for the VEDAI dataset

Table 4.4 shows The VEDAI dataset detection accuracy for every class of objects. Remarkably, compared to conventional YOLOv7-tiny, the proposed model achieved 9 out of the 12 types, the highest average precision. These categories include car, truck, pickup, tractor, camping car, boat, van, plane and other. It is also important to note that the most types of cars that produced the most remarkable results were the car with 88% and others, which outperformed the average precision found in YOLOv7 and YOLOv4 and another earlier research. The proposed model has demonstrated a notable rise in average precision, particularly for the automobile that represents tiny cars, indicating its capacity to identify a more significant number of small objects. Cars, tractors, pickup trucks, camping gear, aeroplanes, boats, and other comparable interclass elements might increase misclassification.

However, there was a significant increase in the average precision for the tractor, plane, boat, and other object. This indicates that the proposed model has shown a remarkable ability to differentiate between the various deep features within these seven groups. Figure 4.2 illustrates the detection results on VEDAI and UA-DETRAC images. Comparing these images to the conventional YOLOv7-tiny model reveals difference in the surroundings and situations of the automobiles. Based on the evaluation images, it is clear that the proposed model has achieved higher levels of detection accuracy. The proposed model, as seen in Figure 4.3 (a), could accurately identify the vehicles with comparable interclass characteristics. However, the YOLOv7-tiny model could not detect the camping vehicle as shown in Figure 4.3 (b). On the other hand, the UA-DETRAC images in Figure 4.3 (c)(d) illustrates images under different conditions, such as night and rainy conditions. Table 4.5 shows that the proposed model outperformed previous works in detecting vehicles when using UA-DETRAC dataset under night and rainy conditions. The results show that the proposed model could detect vehicles correctly under challenging environment conditions which justifies the robustness of the proposed model in practical applications as shown in Figure 4.3 (i)(g)(k) that utilize using first proposal model and Figure 4.3 (j)(h)(m) applied using second proposal model.

Table 4.4 Evaluating the performance of each category on the VEDAI dataset

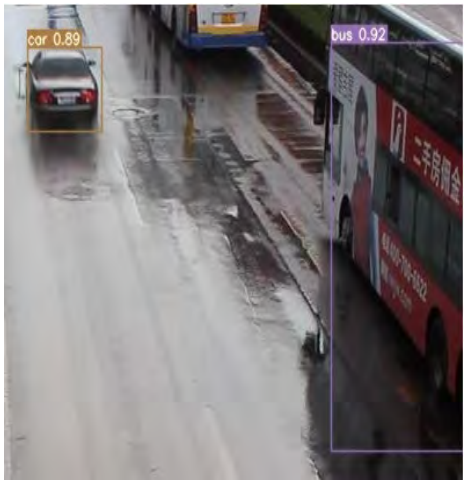
Model Presented	Car-%	Truck-%	Van-%	Tractor-%	Pickup-%	Camping-%	Plane-%	Boat-%	Other-%	mAP-%
Modified cascade CNN (Zhong et al., 2017)	-	-	-	-	-	-	-	-	-	54.6
Modified CNN (Ju et al., 2019)	60	24.5	56.6	37	52	54.8	100	26.5	19	47.8
YOLOv3 (Junos et al., 2022)	54.68	63.49	54.78	50	75.79	67.57	87.5	45.14	29.32	61.79
YOLOv3-tiny (Junos et al., 2022)	62.94	21.45	48.17	49.93	59.27	32.12	8.33	38.54	17.93	37.63
YOLOv4 (Junos et al., 2022)	63.37	89.04	65.62	82.75	51.49	86.87	100	76.36	35.36	73.38
YOLOv4-tiny (Junos et al., 2022)	39.32	60.7	16.68	34.07	50.1	71.48	62.68	54.84	19.68	47.25
Modified YOLOv4 (Junos et al., 2022)	82.15	38.2	55.94	53.9	66.14	54.06	85.94	28.16	13.52	53.11
Modified YOLOv4 (Momin et al., 2022)	74.76	36.14	55.54	56.77	56.72	58.37	85.52	22.78	26.98	52.61
YOLOv7	91.4	83.1	81	79.4	86.6	80.5	95.1	76.5	69.8	68.75
YOLOv7-tiny	79.1	40.4	50.5	54.6	63.3	63.1	41.6	37.9	37.9	48.89
First proposed model (ELAN-VD)	**81**	**44.6**	**55.7**	**72**	**63.9**	**70.2**	**78.2**	**67.7**	**54.5**	**57.94**
Second proposed model (ELAN-VD + UPSAMPLE-VD)	**88.8**	**88.6**	**85.1**	**88.8**	**85.9**	**89.4**	**99.5**	**88.4**	**90.2**	**77.47**

Figure 4.3 Visual illustration of vehicle detection in conventional YOLOv7-tiny: Sunny (a, b) Night (c) Rainy (d), first proposal model Sunny (e, f) Night (g) Rainy (h) and second proposal model Sunny (i, j) Night (k) Rainy (m)

(a)

(b)

(c)



(d)


pickup 0.82
pickup 0.50

(e)


camping car 0.71
camping car 0.63
pickup 0.40

(f)


car 0.92
car 0.91
bus 0.91

(g)


car 0.90
bus 0.93

(h)


pickup 0.93
van 0.78

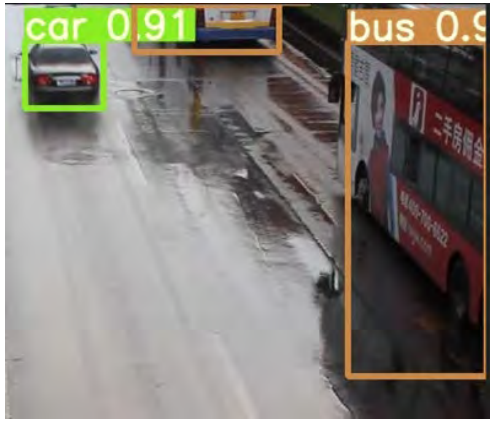
(i)


camping car 0.87
camping car 0.65
pickup 0.86

(j)


car 0.95
car 0.96
bus 0.94

(k)


car 0.91
bus 0.9

(m)

4.4.3 UA-DETRAC dataset

The UA-DETRAC dataset comprises videos captured by a Canon EOS 550D camera at various locations, showcasing diverse traffic patterns and conditions such as urban highways, traffic crossings, and T-junctions. The dataset was carefully selected from hours of image sequences. The videos are recorded at 25 frames per seconds (fps) and with resolution 960 x 540 pixels. The labelled types included cars, buses, trucks and etc more.

4.4.4 Performance evaluation on UA-DETRAC dataset

This experiment utilised 512 x 512-pixel input pictures. Table 4.5 compares the proposed model detection performance with several other models using the UA-DETRAC dataset. Based on the findings of the experiment, the proposed model achieved a mean average precision (mAP) Night of 99.71 % and Rainy of 99.70 % on the first proposed model, which was higher than other lightweight models but lower than the conventional YOLOv7-tiny model (99.70%). In addition, when applying the second proposed model that shows a 0.03 % increase in mean average precision (mAP) Night of 99.73 % and Rainy of 99.70 %. This study demonstrates the effectiveness of the additional ELAN-VD module and UPSAMPLE-VD module by incorporating both local and global characteristics to expand the reception field and significantly segregate important content aspects.

Table 4.5 Results of comparison for the UA-DETRAC dataset

Method	Night mAP (%)	Rainy mAP (%)
YOLOv3 (Hong et al., 2020)	67.50	50.16
Improved YOLOv3 (Hong et al., 2020)	69.53	70.89
YOLOv3-MT (Freudenberg et al., 2022)	67.65	50.18
Lateral CNN (Zhao et al., 2021)	74.36	55.77

Cas-FESSD (Zhao et al., 2021)	74.38	69.14
YOLOv7-tiny	**99.70**	**99.71**
First proposed model (ELAN-VD)	**99.71**	**99.70**
Second proposed model (ELAN-VD + UPSAMPLE-VD)	**99.73**	**99.70**

4.5 Comparison of computational performance

This work analyses the computing times, detection time per image, and GFLOPs, as presented in Table 4.6 and 4.7 by comparing the proposal model with conventional YOLO models. As presented in Table 4.6, the proposed model's real-time performance was tested on two hardware namely on the RTX A2000 and Jetson Nano. The average inference time of the proposed model was lower when compared to the YOLOv7 and YOLOv7-tiny models. In addition, the model we proposed achieved a higher performance compared to both YOLOv7 and YOLOv7-tiny models on the RTX A2000, with a frame rate of 40 frames per second (FPS) slightly better speed compared to convention YOLOv7 of 32 frames per second (FPS). The proposed model also obtained 3 FPS on an inexpensive embedded device namely Jetson Nano, which was slightly faster than the YOLOv7-tiny model. This performance demonstrates that the suggested model is capable of functioning on affordable devices jetson nano with nearly real-time speed of convention YOLOv7-tiny is 39.721 average inference time second (s) compared to first and second proposed model is 8.925 and 13.056 average inference time second (s). The performance improvement was generated from increment of the network's layers in the proposed model namely ELAN-VD module and UPSAMPLE-VD module, which impacted the training and detection process.

Meanwhile, as presented in Table 4.7, YOLOv7-tiny model showed lower detection accuracy than the proposed model. Albeit the highest detection of 68.75 mAP, YOLOv7 model has the highest model size of 74.9 MB because of its profound backbone network which makes it less feasible to be implemented in low-cost device for real-time applications. Inputs, feature extraction networks, feature fusion networks, and output residual connections structures make up YOLOv7 integrated. The networks parameters have increased due to their intricate arrangements, resulting in bigger models. Hence, the network's complexity in YOLOv7 model impacted the calculation and detection times for every image. Compared to the lightweight models, YOLOv7 model consumed more time to train, raising the computational cost. In a nutshell, the average model size of the proposed model is only 23.4MB, which is a reduction of 77.47% mAP from the conventional YOLOv7 model. The network's improved feature extraction network module created fewer trainable parameters, which led to an incredibly tiny model size. By default, convolutional YOLOv7-tiny produce 6219709 of parameters and when modified second proposed model can obtain reducing layer the result outcome 5320117 of parameters as shown in table 4.7.

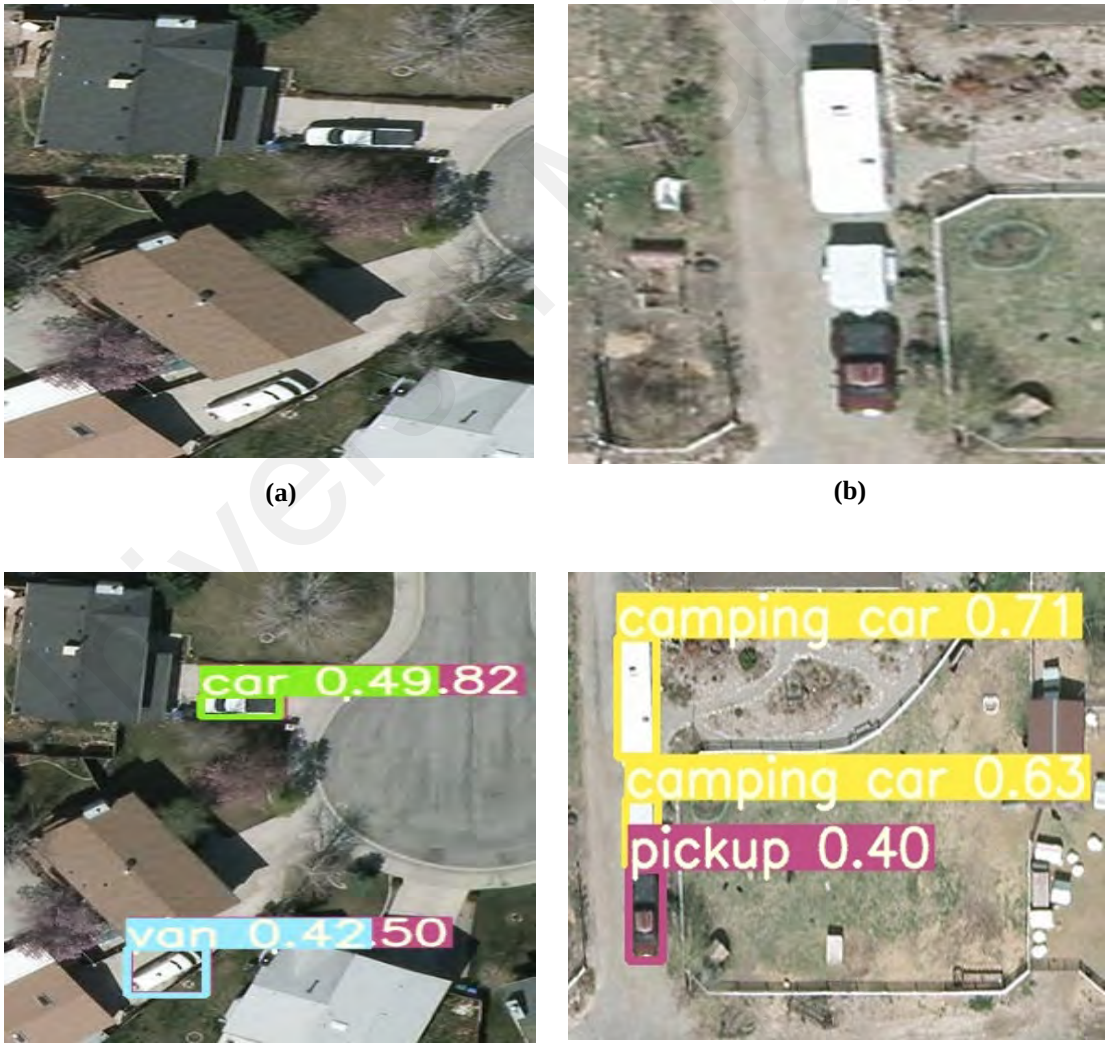
Table 4.6 Comparing performance based on average inference time and frames per second (FPS) using VEDAI dataset

Vehicle detection method	Average inference time second (s)		Frames per second (FPS)	
	Jetson Nano Tegra X1	RTX A2000	Jetson Nano Tegra X1	RTX A2000
YOLOv7	NA	0.599	NA	32
YOLOv7-tiny	39.721	0.573	2	41
First proposed model (ELAN-VD)	**8.925**	**0.46**	**3**	**43**
Second proposed model (ELAN-VD + UPSAMPLE-VD)	**13.056**	**0.518**	**3**	**40**

Table 4.7 Comparing performance based on model size and weight file using VEDAI dataset

Model	metrics/mAP_0.5	GFLOPs	Weight file (MB)	Parameters
YOLOv7	**68.75**	**105.3**	**74.9**	**-**
YOLOv7-tiny	**48.89**	**13.3**	**12.3**	**6219709**
First proposed model (ELAN-VD)	**57.94**	**17.7**	**23.3**	**11543922**
Second proposed model (ELAN-VD + UPSAMPLE-VD)	**77.47**	**22.3**	**23.4**	**11539826**

Figure 4.4 Visual illustration of vehicle detection on Jetson Nano Tegra X1 using YOLOv7-tiny (a, b), first proposal model (c, d) and second proposal model (e, f)



(a) **(b)**

(c) **(d)**

(e)

(f)

CHAPTER 5: CONCLUSION AND RECOMMENDATIONS

5.1 Conclusions

This study presents a lightweight detection model that utilises computer vision to detect objects in real-time applications on embedded devices. The proposed model draws inspiration from the YOLOv7-tiny models, which are known for their fast detection speed and relatively good detection performance. To enhance the precision of detection, Firstly, efficient long-range aggregation network for vehicle detection (ELAN-VD) was incorporated in backbone layer. Secondly, the (UPSAMPLE-VD) on head architecture was improvised resolution to improve the detection accuracy of small vehicles in the aerial image.

The experiment results, evaluated on the VEDAI and UA-DETRAC dataset, demonstrate that the proposed ELAN-VD and UPSAMPLE-VD has achieved a satisfactory level of detection performance compared to the current leading models. However, the detection performance of the proposed ELAN-VD and UPSAMPLE-VD on the VEDAI dataset was quite satisfactory, with a mean average precision (mAP) of 77.47% and an F1-score of 67%. These results outperformed the YOLOv7-tiny model. Based on the evaluation conducted on the UA-DETRAC dataset, it is evident that the YOLOv7-tiny model demonstrates exceptional resilience when faced with difficult datasets. It has achieved an impressive highest mean Average Precision (mAP) score of 99.70%. The ELAN-VD module and UPSAMPLE-VD module achieved an impressive second-highest mAP of 99.73% and an F1-score of 99%. In addition, models offer substantial enhancements in terms of computational performances. The model sizes generated by the proposed model were 23.4 MB, with GFLOPs values of 22.3. It achieved a frame rate of 3 FPS on the Jetson Nano and 40 FPS on the RTX A2000.

The extensive findings demonstrate the correlation between the accuracy of detection, speed of detection, and size of the model in the one stage object detection model, particularly in the case of the YOLO-based model. It is worth noting that a model with a simple and shallow network generates a small number of parameters and GFLOPs value which occupies a small storage size and performs fast detection. In this research, the proposed model demonstrated the best trade-off between the performance metric with a less complex network structure against its preceding models. Based on the proposed network structure, the ELAN-VD provided high accuracy with faster detection time and the (UPSAMPLE-VD) on head architecture is improvised resolution to improve the detection accuracy of small vehicles in the aerial image. According to these significant advantages, the proposed model can facilitate real-time performance on embedded devices.

5.2 Future works

Future work will continue to refine the proposed model to enhance the speed of detection and improve the overall accuracy. Based on the findings of the study, it has been observed that there exists a trade-off between the accuracy and efficiency of the YOLO detection model. However, there are still numerous enhancements that can be suggested to achieve a more balanced performance.

Object detection in multispectral and hyperspectral images has significant advantages in remote sensing applications. The study utilised RGB images to assess the effectiveness of the suggested model. Thus, further exploration of this study could involve utilising multispectral and hyperspectral datasets for training and evaluation purposes, to examine the dependability and resilience of the suggested model. Lastly, this study will be expanded to create a counting algorithm.

REFERENCES

Bai, Y., Yu, J., Yang, S., & Ning, J. (2024). An improved YOLO algorithm for detecting flowers and fruits on strawberry seedlings. *Biosystems Engineering*, *237*, 1-12.

Bayram, A. F., Gurkan, C., Budak, A., & KARATAŞ, H. (2022). A Detection and Prediction Model Based on Deep Learning Assisted by Explainable Artificial Intelligence for Kidney Diseases. *Avrupa Bilim ve Teknoloji Dergisi*(40), 67-74.

Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.

Cao, C.-Y., Zheng, J.-C., Huang, Y.-Q., Liu, J., & Yang, C.-F. (2019). Investigation of a promoted you only look once algorithm and its application in traffic flow monitoring. *Applied Sciences*, *9*(17), 3619.

Chen, C., Yuan, G., Zhou, H., Ma, Y., & Ma, Y. (2023). Optimized YOLOv7-tiny model for smoke detection in power transmission lines. *Mathematical biosciences and engineering*, *20*(11), 19300-19319.

Chen, C., Zhong, J., & Tan, Y. (2019). Multiple-oriented and small object detection with convolutional neural networks for aerial image. *Remote Sensing*, *11*(18), 2176.

Chen, D., Sun, S., Lei, Z., Shao, H., & Wang, Y. (2021). Ship target detection algorithm based on improved YOLOv3 for maritime image. *Journal of Advanced Transportation*, *2021*, 1-11.

Chen, L., Ding, Q., Zou, Q., Chen, Z., & Li, L. (2020). DenseLightNet: A light-weight vehicle detection network for autonomous driving. *IEEE Transactions on Industrial Electronics*, *67*(12), 10600-10609.

Dijkstra, K., van de Loosdrecht, J., Schomaker, L. R., & Wiering, M. A. (2019). Hyperspectral demosaicking and crosstalk correction using deep learning. *Machine Vision and Applications*, *30*(1), 1-21.

Diwan, T., Anirudh, G., & Tembhurne, J. V. (2023). Object detection using YOLO: Challenges, architectural successors, datasets and applications. *Multimedia Tools and Applications*, *82*(6), 9243-9275.

Elkhrachy, I. (2021). Accuracy assessment of low-cost unmanned aerial vehicle (UAV) photogrammetry. *Alexandria Engineering Journal*, *60*(6), 5579-5590.

Freudenberg, M., Magdon, P., & Nölke, N. (2022). Individual tree crown delineation in high-resolution remote sensing images based on U-Net. *Neural Computing and Applications*, *34*(24), 22197-22207.

Froidevaux, A., Julier, A., Lifschitz, A., Pham, M.-T., Dambreville, R., Lefèvre, S., Lassalle, P., & Huynh, T.-L. (2020). Vehicle detection and counting from VHR satellite images: Efforts and open issues. IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium,

García-Aguilar, I., García-González, J., Luque-Baena, R. M., & López-Rubio, E. (2023). Automated labeling of training data for improved object detection in traffic videos by fine-tuned deep convolutional neural networks. *Pattern Recognition Letters*, *167*, 45-52.

Gupta, A., Watson, S., & Yin, H. (2021). Deep learning-based aerial image segmentation with open data for disaster impact assessment. *Neurocomputing*, *439*, 22-33.

He, W., Huang, Z., Wei, Z., Li, C., & Guo, B. (2019). TF-YOLO: An improved incremental network for real-time object detection. *Applied Sciences*, *9*(16), 3225.

Hong, F., Lu, C.-H., Liu, C., Liu, R.-R., & Wei, J. (2020). A traffic surveillance multi-scale vehicle detection object method base on encoder-decoder. *IEEE Access*, *8*, 47664-47674.

Hua, Y., Xu, H., Liu, J., Quan, L., Wu, X., & Chen, Q. (2023). A peanut and weed detection model used in fields based on BEM-YOLOv7-tiny. *Mathematical biosciences and engineering: MBE*, *20*(11), 19341-19359.

Huang, R., Gu, J., Sun, X., Hou, Y., & Uddin, S. (2019). A rapid recognition method for electronic components based on the improved YOLO-V3 network. *Electronics*, *8*(8), 825.

Jiao, Y., Wang, Z., Shang, Y., Li, R., Hua, Z., & Song, H. (2023). Detecting endosperm cracks in soaked maize using μCT technology and R-YOLOv7-tiny. *Computers and Electronics in Agriculture*, *213*, 108232.

Ju, M., Luo, J., Zhang, P., He, M., & Luo, H. (2019). A simple and efficient network for small target detection. *IEEE Access*, *7*, 85771-85781.

Junos, M. H., Khairuddin, A. S. M., & Dahari, M. (2022). Automated object detection on aerial images for limited capacity embedded device using a lightweight CNN model. *Alexandria Engineering Journal*, *61*(8), 6023-6041.

Kaya, Ö., Çodur, M. Y., & Mustafaraj, E. (2023). Automatic Detection of Pedestrian Crosswalk with Faster R-CNN and YOLOv7. *Buildings*, *13*(4), 1070.

Koirala, A., Walsh, K., Wang, Z., & McCarthy, C. (2019). Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'. *Precision Agriculture*, *20*, 1107-1135.

Kumar, A. (2023). SEAT-YOLO: A Squeeze-Excite and Spatial Attentive You Only Look Once Architecture for Shadow Detection. *Optik*, 170513.

Le, T.-T., & Lin, C.-Y. (2019). Deep learning for noninvasive classification of clustered horticultural crops–A case for banana fruit tiers. *Postharvest Biology and Technology*, *156*, 110922.

Li, Y., Han, Z., Xu, H., Liu, L., Li, X., & Zhang, K. (2019). YOLOv3-lite: A lightweight crack detection network for aircraft structure based on depthwise separable convolutions. *Applied Sciences*, *9*(18), 3781.

Li, Y., & Ren, F. (2019). Light-weight retinanet for object detection. *arXiv preprint arXiv:1905.10011*.

Liang, S., Chen, R., Duan, G., & Du, J. (2023). Deep learning-based lightweight radar target detection method. *Journal of Real-Time Image Processing*, *20*(4), 61.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14,

Liu, Y., & Wang, X. (2022). SAR Ship Detection Based on Improved YOLOv7-Tiny. 2022 IEEE 8th International Conference on Computer and Communications (ICCC),

Luo, Y., Zhang, Y., Yan, J., & Liu, W. (2021). Generalizing face forgery detection with high-frequency features. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Mandal, M., Shah, M., Meena, P., Devi, S., & Vipparthi, S. K. (2019). AVDNet: A small-sized vehicle detection network for aerial visual data. *IEEE Geoscience and Remote Sensing Letters*, *17*(3), 494-498.

Min, W., Li, X., Wang, Q., Zeng, Q., & Liao, Y. (2019). New approach to vehicle license plate location based on new model YOLO-L and plate pre-identification. *IET Image Processing*, *13*(7), 1041-1049.

Momin, M. A., Junos, M. H., Mohd Khairuddin, A. S., & Abu Talip, M. S. (2022). Lightweight CNN model: automated vehicle detection in aerial images. *Signal, Image and Video Processing*, 1-9.

Nguyen, H. H., Nghiem, V. Q., Hoang, M. S., Nghiem, T. K., & Dang, N. M. (2023). A Novel Variant of Yolov7-Tiny for Object Detection on Aerial Vehicle Images. International Conference on Communication and Intelligent Systems,

Razakarivony, S., & Jurie, F. (2016). Vehicle detection in aerial imagery: A small target detection benchmark. *Journal of Visual Communication and Image Representation*, *34*, 187-203.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. Proceedings of the IEEE conference on computer vision and pattern recognition,

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. Proceedings of the IEEE conference on computer vision and pattern recognition,

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Sapitri, A. I., Nurmaini, S., Rachmatullah, M. N., Tutuko, B., Darmawahyuni, A., Firdaus, F., Rini, D. P., & Islami, A. (2023). Deep learning-based real time

detection for cardiac objects with fetal ultrasound video. *Informatics in Medicine Unlocked*, *36*, 101150.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

She, F., Hong, Z., Zeng, Z., & Yu, W. (2023). Improved Traffic Sign Detection Model Based on YOLOv7-Tiny. *IEEE Access*, *11*, 126555-126567.

Sun, Y., Zhao, Y., & Wang, S. (2022). Multiple traffic target tracking with spatial-temporal affinity network. *Computational intelligence and neuroscience*, *2022*.

Tan, M., & Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. International conference on machine learning,

Tian, D., Zhang, C., Duan, X., & Wang, X. (2019). An automatic car accident detection method based on cooperative vehicle infrastructure systems. *IEEE Access*, *7*, 127453-127463.

Wang, J., Dong, Y., Zhao, S., & Zhang, Z. (2023). A high-precision vehicle detection and tracking method based on the attention mechanism. *Sensors*, *23*(2), 724.

Wang, Y., Wang, H., & Xin, Z. (2022). Efficient Detection Model of Steel Strip Surface Defects Based on YOLO-V7. *IEEE Access*, *10*, 133936-133944.

Wen, L., Du, D., Cai, Z., Lei, Z., Chang, M.-C., Qi, H., Lim, J., Yang, M.-H., & Lyu, S. (2020). UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. *Computer Vision and Image Understanding*, *193*, 102907.

Womg, A., Shafiee, M. J., Li, F., & Chwyl, B. (2018). Tiny SSD: A tiny single-shot detection deep convolutional neural network for real-time embedded object detection. 2018 15th Conference on computer and robot vision (CRV),

Wu, Y., Abdel-Aty, M., Zheng, O., Cai, Q., & Zhang, S. (2020). Automated safety diagnosis based on unmanned aerial vehicle video and deep learning algorithm. *Transportation research record*, *2674*(8), 350-359.

Xie, Y., Cai, J., Bhojwani, R., Shekhar, S., & Knight, J. (2020). A locally-constrained YOLO framework for detecting small and densely-distributed building footprints. *International Journal of Geographical Information Science*, *34*(4), 777-801.

Yang, Z., Feng, H., Ruan, Y., & Weng, X. (2023). Tea Tree Pest Detection Algorithm Based on Improved Yolov7-Tiny. *Agriculture*, *13*(5), 1031.

Ye, H., & Wang, Y. (2023). Residual Transformer YOLO for Detecting Multi-Scale Crowded Pedestrian. *Applied Sciences*, *13*(21), 12032.

Yu, G., Cai, R., Su, J., Hou, M., & Deng, R. (2023). U-YOLOv7: A network for underwater organism detection. *Ecological Informatics*, 102108.

Zarei, N., Moallem, P., & Shams, M. (2023). Real-time vehicle detection using segmentation-based detection network and trajectory prediction. *IET Computer Vision*.

Zhang, G., Liu, S., Nie, S., & Yun, L. (2024). YOLO-RDP: Lightweight Steel Defect Detection through Improved YOLOv7-Tiny and Model Pruning. *Symmetry*, *16*(4), 458.

Zhang, X., Wang, H., Xu, C., Lv, Y., Fu, C., Xiao, H., & He, Y. (2019). A lightweight feature optimizing network for ship detection in SAR image. *IEEE Access*, *7*, 141662-141678.

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. Proceedings of the IEEE conference on computer vision and pattern recognition,

Zhang, Y., Sun, Y., Wang, Z., & Jiang, Y. (2023). YOLOv7-RAR for Urban Vehicle Detection. *Sensors*, *23*(4), 1801.

Zhao, M., Zhong, Y., Sun, D., & Chen, Y. (2021). Accurate and efficient vehicle detection framework based on SSD algorithm. *IET Image Processing*, *15*(13), 3094-3104.

Zhao, Y., Zhao, L., Li, C., & Kuang, G. (2020). Pyramid attention dilated network for aircraft detection in SAR images. *IEEE Geoscience and Remote Sensing Letters*, *18*(4), 662-666.

Zheng, Z., Li, J., & Qin, L. (2023). YOLO-BYTE: An efficient multi-object tracking algorithm for automatic monitoring of dairy cows. *Computers and Electronics in Agriculture*, *209*, 107857.

Zhong, J., Lei, T., & Yao, G. (2017). Robust vehicle detection in aerial images based on cascaded convolutional neural networks. *Sensors*, *17*(12), 2720.

Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. *Proceedings of the IEEE*.