IUST
Iran University of Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Enhanced Lightweight YOLO Model for Efficient Vehicle Detection in Satellite Imagery

Mohamad Haniff Junos*(C.A.), Anis Salwa Mohd Khairuddin**, Elmi Abu Bakar*, and Ahmad Faizul Hawary*

Abstract: Vehicle detection in satellite images is a challenging task due to the variability in scale and resolution, complex background, and variability in object appearance. One-stage detection models are currently state-of-the-art in object detection due to their faster detection times. However, these models have complex architectures that require powerful processing units to train while generating a large number of parameters and achieving slow detection speed on embedded devices. To solve these problems, this work proposes an enhanced lightweight object detection model based on the YOLOv4 Tiny model. The proposed model incorporates multiple modifications, including integrating a Mix-efficient layer aggregation network within its backbone network to optimize efficiency by reducing parameter generation. Additionally, an improved small efficient layer aggregation network is adopted in the modified path aggregation network to enhance feature extraction across various scales. Finally, the proposed model incorporates the Swish function and an extra YOLO head for detection. The experimental results evaluated on the VEDAI dataset demonstrated that the proposed model achieved a higher mean average precision value and generated the smallest model size compared to the other lightweight models. Moreover, the proposed model achieved real-time performance on the NVIDIA Jetson Nano. These findings demonstrate that the proposed model offers the best trade-offs in terms of detection accuracy, model size, and detection time, making it highly suitable for deployment on embedded devices with limited capacity.

Keywords: Lightweight architecture, Modified YOLO, Satellite Image, Vehicle Detection.

1 Introduction

IN Recent Decades, There Have Been Significant Improvements in Remote Sensing Technology, leading to remarkable growth in the research field focused on ground object detection from satellites. Additionally, the rapid retrieval of information from ground stations [1], along with the increasing availability of publicly accessible vehicle data [2], has further advanced research in this field. Modern technological advancements have played a significant role in fostering the growth of interest in Intelligent Transportation Systems (ITS). ITS integrates cutting-edge technologies into transportation systems to improve efficiency, safety, and sustainability. The study of traffic scenes and vehicle detection draws interest from researchers aiming to address numerous challenges, including identifying and counting vehicles [3], assessing vehicle speed [4], forecasting traffic patterns [5], and identifying traffic congestion [6]. These tasks are particularly complex because satellite images, which differ visually from natural images, present unique challenges. Specifically, vehicle detection in satellite images is hindered by resolution limitations, occlusion, environmental conditions, and cluttered backgrounds. Besides, variances in vehicle appearance, including size, shape, and color, pose challenges for creating a universal detection model.

Iranian Journal of Electrical & Electronic Engineering, 2025.
Paper first received 08 Dec. 2024 and accepted 20 Feb. 2025.
* The author is with the School of Aerospace Engineering, Universiti Sains Malaysia, Engineering Campus, 14300 Nibong Tebal, Penang, Malaysia.
E-mail: haniffjunos@usm.my, meelmi@usm.my, aefaizul@usm.my.
** The authors are with the Department of Electrical Engineering, Faculty of Engineering, Universiti Malaya, 50603 Kuala Lumpur, Malaysia.
E-mails: anissalwa@um.edu.my,
Corresponding Author: Mohamad Haniff Junos.

The advancement of deep learning technology in machine vision applications has led to significant enhancements in deep convolutional neural networks (CNNs), notably elevating their effectiveness in object detection with respect to both accuracy and efficiency [7]. Deep learning facilitates the automatic extraction of multi-scale image features through self-learning from extensive datasets. Currently, two main approaches to object detection using deep learning techniques are candidate region-based and regression-based models. Candidate region-based models operate in two phases, with the initial phase generating an area of interest and the subsequent phase extracts features from each candidate box to accomplish bounding box determination and classification [8]. Despite achieving high accuracy in localizing and identifying objects, the candidate region-based model is limited by its slow processing speed, rendering it unsuitable for real-time use. Conversely, object detection is treated as a unified regression problem in the regression-based approach, handling region detection and classification simultaneously within the network [9]. You Only Look Once (YOLO) model is considered state-of-the-art in regression-based models due to its remarkable real-time performance of powerful GPU. However, the complex nature of the network requires longer training times, leading to the creation of a large-sized model that is unsuitable for implementation in embedded systems.

Furthermore, detecting vehicles using satellites necessitates deploying an object detection model locally on embedded devices near the data source. This is done to achieve faster data processing and to avoid unreliable data exchange with a remote server, as well as security and privacy issues. However, these devices typically have limited hardware resources. As a result, it is essential to conduct a reasonable trade-off analysis between accuracy and efficiency. Therefore, a fast and lightweight detection model is required for accurate vehicle detection in satellite images. Currently, several modified YOLO-based models have been developed to improve the trade-off performance in object detection tasks in satellite images. Satellite images can capture a wide range of information across different spectral bands and sensor types. A comprehensive representation of the observed scene can be obtained by fusing data from multiple modalities. The multimodal fusion method was integrated into the YOLO network, enabling improved object detection performance by leveraging complementary information from various data sources [10]-[11]. However, combining multiple modalities requires extra computational resources, involving processing various data types and integrating them into the model. This can result in longer training and inference times, increasing the resource intensity of the system. Besides, a dilated convolution is integrated into the YOLO, which applies a filter over a larger input field without increasing the number of parameters or computation [12]. This module allows the YOLO model to capture features at multiple scales by adjusting the dilation rates, enabling the detection of vehicles of different sizes in low-resolution images. While the dilated convolution module enhances feature extraction and increases the receptive field, it also introduces challenges related to design complexity and computational overhead.

Channel pruning is a popular technique for removing redundant and less significant channels, thus preserving the overall efficacy of the YOLO network. This approach effectively decreases the total number of parameters in the network [13]-[14]. However, at times, this method might remove essential features required for precise object detection, which can lead to a decline in performance. On the other hand, adding extra detection layers to the YOLO model can improve its ability to detect objects of various scales more effectively [13], [15]. Different layers can focus on different feature resolutions, enabling the identification of both small and large objects within the same image. However, it's important to note that incorporating more detection layers will increase the computational load, potentially slowing down the inference time. Furthermore, residual blocks (ResNets) have been incorporated into the backbone structure of YOLO, facilitating the training of deeper networks while reducing the number of parameters [16]. ResNets help address the issue of vanishing gradients and contribute to stability during the optimization process. However, these advantages come at the cost of increased computational demands, potentially impacting the model's efficiency. Moreover, MobileNet is known for its lightweight and efficient real-time object detection on resource-constrained devices. The network requires fewer parameters with less computational power and can achieve faster inference times. A MobileNetv3 replaced the original backbone of the YOLOv4 model to simultaneously reduce the parameter complexity [17]. Despite being lightweight, the network's compact design may limit its ability to capture all the necessary features, especially when dealing with complex datasets that are large-scale or highly diverse, which could lead to potential decreases in accuracy. Based on the related works, there is a lack of studies focused on enhancing the accuracy and efficiency of the YOLO model for detecting vehicles in satellite images. This is mainly due to the complexities involved in working with satellite images, which pose challenges in developing a highly accurate and efficient detection model.

Based on the abovementioned shortcomings, an efficient YOLO model is proposed for detecting vehicles in satellite imagery. The Improved Tiny YOLO model aims to accurately detect multi-class vehicles in complex satellite images while maintaining an efficient and

lightweight structure for real-time performance on embedded devices. The proposed model includes three major improvements. First, a novel Mix-efficient layer aggregation network (MixELAN) module is integrated into the feature extraction network by combining the mixed-depthwise convolutions and efficient layer aggregation network to efficiently improve the model's structure complexity. Next, a small, efficient layer aggregation network (SELAN) is fused into the modified path aggregation network (PANet) to promote a more comprehensive utilization of features from various levels of the network. Finally, the Swish activation function is utilized in every convolutional layer, and an extra detection layer is added to the prediction network to improve detection accuracy.

The remainder of this article is structured as follows. Section 2 outlines the comprehensive methodology behind the Improved Tiny YOLO model developed for vehicle detection in satellite imagery, which also includes the dataset, performance evaluation, and experimental setup. Experimental results are analyzed and discussed in Section 3, and the conclusion of the study is presented in Section 4.

2 Methodology

The Improved Tiny YOLO model has been developed based on the modifications applied on the YOLOv4 Tiny model to enhance the accuracy of detecting small objects while maintaining its lightweight and fast performance for object detection in satellite images. This section provides an outline of the methodology used in this study. Firstly, Section 3.1 gives a brief overview of the VEDAI datasets used in the experiments. Then, Section 3.2 discussed the proposed methodology in detail. Finally, Sections 3.3 and 3.4 describe the standard evaluation metric and experimental setup.

2.1 Dataset

This study utilized publicly available RGB satellite imagery datasets known as the VEDAI dataset [2]. The dataset consists of a wide variety of small vehicle types depicted in different environmental contexts, such as varied orientations and backgrounds, lighting conditions, and occlusions. In total, there are 1125 images in the training set and 125 images in the validation set. Each image measures 1024 × 1024 pixels and has a spatial resolution of 12.5 cm. Additionally, the dataset contains nine categories of vehicles, such as 'truck,' 'boat,' 'tractor,' 'plane,' 'car,' 'pickup,' 'van,' 'camping car,' and 'others.' The 1250 images in the dataset are split into a 90% train set and a 10% test set.

2.2 The proposed Improved Tiny YOLO model

An optimized YOLO model based on the YOLOv4 Tiny model is proposed to achieve an optimal balance between accuracy and efficiency. The proposed model integrated several significant modifications in the backbone and neck section. The backbone network employs the novel MixELAN module to effectively improve the feature extraction and reduce the model's parameters. Then, the small efficient layer aggregation network (SELAN) is adopted in the modified PANet in the neck section. Finally, the Swish function and additional YOLO head detection are integrated into the proposed model. Figure 1 illustrates the proposed architecture of the Improved Tiny YOLO model.

2.2.1 The backbone network

The backbone architecture employs two 3×3 Convolution-Batchnorm-Swish (CBS) blocks with stride 2 and three MixELAN modules replacing the three CSP blocks from the YOLOv4 Tiny model to improve accuracy while reducing the number of parameters generated within the network. These modules are a fusion of the efficient layer aggregation network (ELAN) modules mixed with a depthwise convolution module (MixConv), representing a memory-efficient block. The ELAN module addressed the diminishing convergence observed in deep models as they scale. ELAN focuses on establishing an efficient network by managing the shortest and longest gradient paths in each layer. This approach enables deeper networks to achieve convergence more effectively [18]. The proposed model comprises of a simplified ELAN containing three 1×1 CBS blocks with a stride of 1. The MixConv module substitutes single convolutional kernels with a mix of different sizes to improve accuracy and efficiency [19]. This is achieved by segmenting channels and applying various kernel sizes to each group. Larger kernels capture high-resolution patterns, whereas smaller kernels concentrate on low-resolution patterns, thus enhancing model accuracy and efficiency. This layer enhances the model's capability to extract features from inputs with high dimensions, thereby improving its expressive power.

The proposed model adopted two different architectures of MixELAN modules with two different kernel sizes, namely MixELAN5 and MixELAN7 modules. For the MixELAN5 module, the first two layers used a 1×1 CBS block with a 32 filter size. Subsequently, the second layer employed the MixConv layer using larger kernel sizes to effectively capture a wide range of patterns at different resolutions. The MixConv layer comprises two Depthwise convolution-Batchnorm-Swish (DWCBS) blocks, using a mix of convolutional kernel sizes of 3×3 and 5×5 with 32 filter size and stride 1. Then, the feature maps generated from the four earlier blocks are concatenated to obtain feature maps of 128 size. Finally, a 1×1 CBS block of 64 filter size is performed. The second MixELAN5 block utilized the same structure with a different filter size of 64 for the first four layers and a filter size of 128 for the last

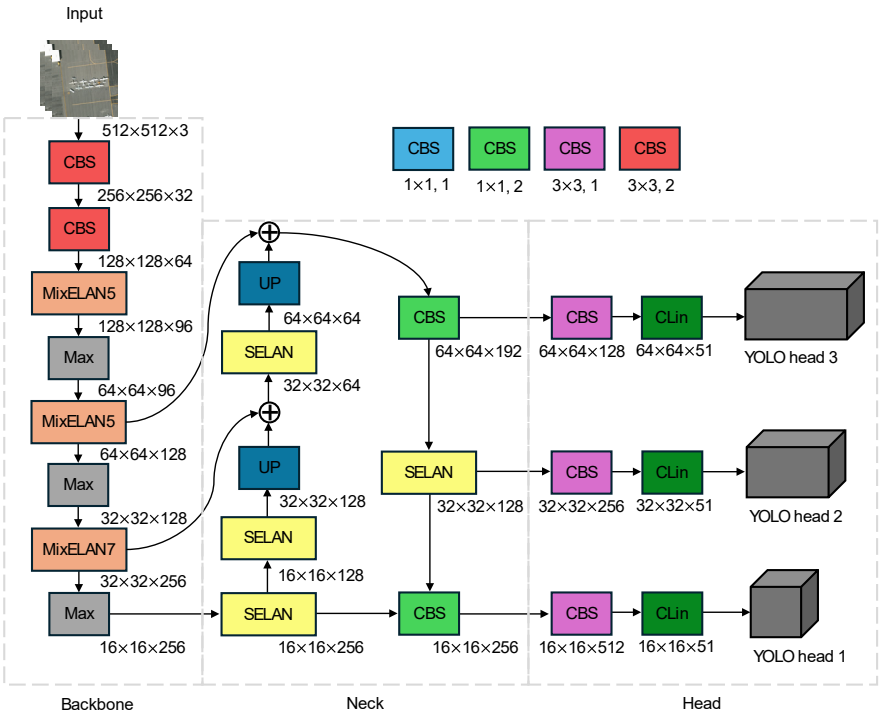

Fig. 1 Detailed architecture of Improved Tiny YOLO model.

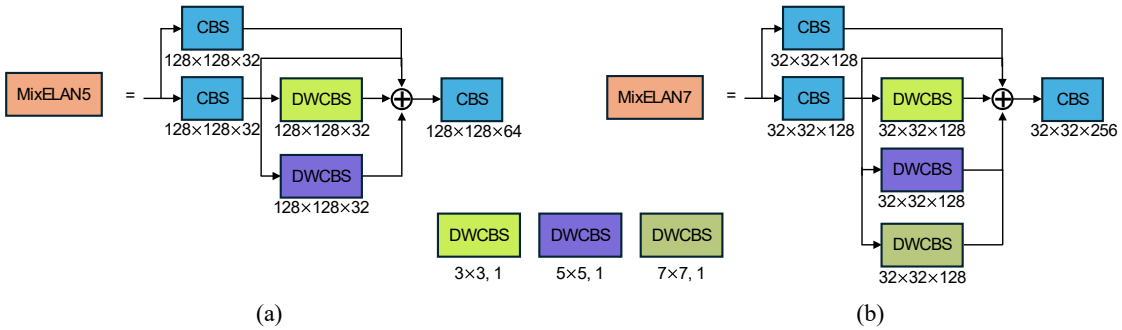

Fig. 2 Mix efficient layer aggregation network**.**

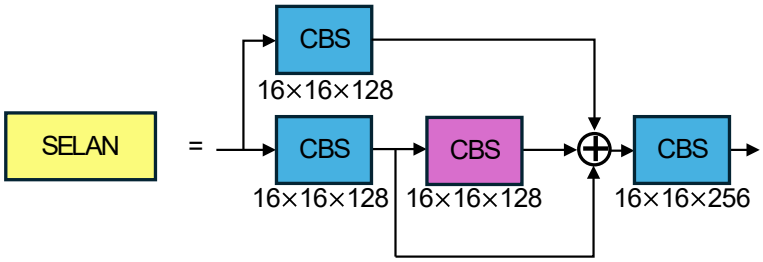

Fig. 3 Small efficient layer aggregation network.

layer. The detailed architecture for the first MixELAN5 module for the first block is shown in Figure 2 (a). Next, the proposed module adopted MixELAN7 modules to further improve the model's ability to extract features from high-dimensional inputs. The module utilizes the same structure as the MixELAN5 but adds a 7×7 DWCBS block with stride 1. The MixELAN7 block utilized a filter size of 128 for the first five layers and 256 for the last layer. The detailed architecture of the MixELAN7 is depicted in Figure 2 (b).

2.2.2 The neck network

The proposed model adopted a simplified Path Aggregation Network (PANet) [20] in the neck network to improve the feature extraction and information flow. The PANet uses the bottom-up path, which helps to

better propagate lower-level features upwards, ensuring that fine-grained details are preserved and utilized in higher layers. This architecture aggregates features from different network layers, combining low-level and high-level features to improve the model's ability to detect objects at various scales and improve the representation of the context around objects. The PANet also enhances the information flow, allowing the network to leverage more comprehensive feature representations, leading to better object detection performance.

The first layer of the simplified PANet utilizes three small, efficient layer aggregation networks (SELAN) with two up-sample operations. The SELAN comprises two 1×1 CBS blocks and one 3×3 CBS block. The feature maps from these three blocks are concatenated and followed by 1×1 CBS blocks. The example architecture of the first SELAN is illustrated in Figure 3. The first SELAN is connected to the feature map with a filter of 256 from the backbone, producing feature maps of 16×16×256. The second and third SELANs undergo up-sample operation followed by a concatenation process from feature maps with the filters of 128 and 64 from the backbone, which produced feature maps of 16×16×128 and 32×32×64. The up-sample operation is used to double up the dimension of the feature maps. The second layer comprises two 1×1 CBS blocks and one SELAN. CBS block with a stride of 2 is used to reduce the feature map dimension in the second layer. Finally, the detection layers utilize the feature maps generated in the second layer.

2.2.3 The detection network

The object detection process takes place in the prediction network, using the last three feature maps from the neck network. Each network consists of a 3×3 CBS layer with stride 1, a 1×1 convolution-linear layer, and a YOLO prediction layer. The CBS layer combines the feature maps from the previous layer, while the convolution-linear (CLin) layer generates the final prediction feature map. The YOLO output prediction layers generate a vector that includes the relevant class label, confidence score, and the predicted bounding box's coordinates. Each YOLO prediction layer calculates the loss separately, and these individual losses are combined and summed up for the backpropagation process.

2.3 Performance evaluation

Various performance metrics were employed to assess the effectiveness of the proposed model. These metrics include recall, precision, precision-recall curve, F1-score, mean average precision (mAP), average IoU, and detection speed. Recall measures the likelihood of correctly identifying true objects, whereas precision evaluates the accuracy of predictions. The precision-recall curve effectively demonstrates the balance between precision and recall. The definitions of recall and precision are provided in Equations 1 and 2, respectively. Furthermore, the F1-score in Equation 3 merges precision and recall into a harmonic mean, with a score of 1 representing the highest accuracy. A true positive (TP) describes an object that has been correctly identified. Conversely, objects incorrectly identified are called false positives (FP), and those not identified are called false negatives (FN). The area beneath the precision-recall curve across different detection thresholds is measured by average precision (AP), and the mAP assesses the mean accuracy over various classes. The AP and mAP are computed by Eqs. 4 and 5, respectively. Moreover, the detection speed is measured based on the inference speed and the frame per second (FPS). Furthermore, the computational performance is determined through various criteria, including computation duration, model size, number of parameters, and floating point operations (FLOPs). The model's size reflects its storage requirements, which are determined by the number of trainable parameters in the network. Additionally, FLOPs are used to measure the computational complexity of the model.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$F_1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{3}$$

$$AP = \int_0^1 Precision(Recall)\, dRecall \tag{4}$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{5}$$

2.4 Experiment setup

The experiments were conducted using a Windows 10 64-bit OS equipped with an Intel Core i5-9300H @ 2.4G Hz CPU, NVIDIA GeForce GTX 1650 Ti, 8 GB of RAM, and a graphic card with 4 GB of VRAM. The Darknet framework was applied to train the YOLO-based models. The real-time performance was assessed using an NVIDIA Jetson Nano 4GB embedded device. Standardized hyperparameters were applied across all YOLO-based models to maintain uniformity in the experimental results. The input images were resized to 512×512 pixels, and a batch size of 64, with a subdivision of 32, was selected. Evaluation metrics were determined using a standard IoU threshold of 0.5. The initial learning rate was set at 0.001, with adjustments to 0.0001 and 0.00001 after achieving 80% and 90% of the training steps. Furthermore, adjustments were made to

the decay weight and momentum at 0.0005 and 0.9. The training process spanned over 30,000 steps.

3 Results and Discussions

This section evaluates the performance of the Improved Tiny YOLO model, focusing on its detection performance and computational efficiency. The model is benchmarked against several lightweight YOLO models as well as studies conducted in the past.

3.1 Comparison of detection performance

Table 1 provides a comparative assessment of the detection efficiency of the proposed Improved Tiny and several other detection models. The results demonstrate that the Improved Tiny YOLO model outperforms the other tiny models, achieving a mAP of 50.41%, notably higher than the mAPs obtained by YOLOv7 Tiny (47.84%), YOLOv4 Tiny (47.25%), and YOLOv3-tiny (37.63), %respectively. Additionally, the Improved Tiny model exhibits a significant mAP improvement of 6.69% over the baseline YOLOv4 Tiny model. Despite the YOLOv7 Tiny model featuring a more complex architectural depth, it achieves a slightly lower mAP than the proposed model. The simpler designs of YOLOv3 Tiny and YOLOv4 Tiny models are associated with their lower recall values, primarily due to difficulties in detecting smaller objects. In terms of average IoU, the proposed model obtained a higher value compared to the YOLOv7 Tiny and YOLOv3 Tiny models by 46.92%. While the value is slightly lower than the YOLOv4 Tiny model, it is considered a good overlap percentage between ground truth and predicted bounding boxes, especially considering it exhibited a higher overall mAP value. The proposed model achieved a higher precision value than the YOLOv7 Tiny and YOLOv3 Tiny models but slightly lower than the original YOLOv4 Tiny model at 0.6. However, the recall value of the proposed model is significantly increased by 0.2 compared to the YOLOv4 Tiny model, resulting in an F1-score of 0.59, which is the highest among the other tiny models. These results show that the Improved Tiny YOLO model excels in detecting more vehicles, especially in the case of smaller objects. The comparison of the precision-recall curve between the lightweight models is depicted in Figure 4. It can be noticed that the curve for Improved Tiny YOLO is shifted slightly better, leading to a marginally larger area under the curve and corresponding to a higher mAP value.

These results show that the proposed modifications have significantly improved the ability of the Improved Tiny YOLO models to detect small objects, especially vehicles, in satellite images. Firstly, integrating MixConv and ELAN modules in the backbone network leverages the strengths of both approaches to create an efficient network architecture, enhancing multi-scale feature representation, optimizing computational efficiency, and improving detection accuracy. The ELAN module effectively merges features from various layers within the network, enabling the capture of low-level and high-level information for more precise predictions. This combination of features allows the model to utilize information from various stages of the network, hence improving its capability to identify objects of different sizes, shapes, and appearances. Besides, the MixConv module relies on its mixed-depthwise convolutions and channel-mixing operations to improve efficiency and performance by enhancing feature interactions across different channels. These operations enable the model to capture richer and more diverse feature representations by mixing information from different channels. This design allows the model to efficiently capture features at multiple scales, leading to improved feature representation learning.

Secondly, the neck section, composed of a modified PANet, enhances detection performance by improving feature fusion and information flow. This network leverages a bottom-up pathway to reduce the distance of information transfer between lower and higher layers, thereby retaining more spatial details from the initial layers. The integration of the modified ELAN module helps to combine features across various layers, which enables the model to capture more complex details and contexts from various input images. Finally, the proposed model incorporates three YOLO head layers for object detection. This enables the model to capture fine details and features at various scales, allowing it to detect objects of different sizes more effectively. Moreover, additional detection layers improve the feature representation by combining information from different stages of the network, leading to richer and more discriminative features.

A comprehensive analysis compares the proposed model to several established past studies to assess its effectiveness, as detailed in Table 1. It is evident that both YOLO-S [16] and Modified YOLO-PmA [21] demonstrated inferior mAP in comparison to the Improved Tiny YOLO model. Notably, YOLO-S utilizes feature fusion and a pass-through module in its architecture, whereas modified YOLO-PmA is distinguished by its novel hypermetropic architecture. In the case of YOLO-S, the categorization approach involved merging cars and pickups into one class and similarly merging camping cars with vans into a single van class, which resulted in an mAP of 43.60%. Conversely, the YOLO-PmA analysis considered all categories individually and achieved an mAP of 37.91%. Compared to the Improved Tiny YOLO model, YOLO-Fine [13], SuperYOLO [10], and YOLOrs [11] have shown superior accuracy levels, achieving 68.18%, 72.49%, and 57.00%, respectively. YOLO-Fine enhances the YOLOv3 model through channel pruning to reduce the number of network parameters. In contrast,

SuperYOLO leverages multimodal data fusion with super-resolution learning for improved detection across various scales at high resolution, while YOLOrs integrates ResNet's modular residual blocks with YOLOv3's multi-head detection approach. Their notably high mAP can be attributed to the models not evaluating all nine classes. Specifically, YOLO-Fine evaluates only seven classes, combining the plane class with others, whereas both SuperYOLO and YOLOrs omit any class with fewer than 50 instances, such as motorcycles, planes, and buses.

Table 1 Comparison of detection performance.

Model	Input size	Precision	Recall	F1-score	Average IoU (%)	mAP (%)
YOLO-S [16]	416	0.78	0.69	0.73	-	43.60
Modified YOLO-PmA [21]	416	-	-	-	-	37.91
YOLO-Fine [13]	512	0.67	0.70	0.69	-	68.18
SuperYOLO [10]	512	-	-	-	-	72.49
YOLOrs [11]	512	0.47	0.74	-	-	57.00
YOLOv3 Tiny	512	0.51	0.50	0.51	36.52	37.63
YOLOv4 Tiny	512	0.69	0.39	0.50	54.87	47.25
YOLOv7 Tiny	512	0.55	0.59	0.57	43.86	47.84
Improved Tiny YOLO	512	0.60	0.59	0.59	46.92	50.41

Table 2 Comparison of individual class performance.

Model	Truck (%)	Car (%)	Van (%)	Pickup (%)	Tractor (%)	Plane (%)	Camping car (%)	Boat (%)	Other (%)	mAP (%)
YOLO-S [16]	32.60	80.60	44.50	-	27.30	74.20	-	25.90	19.90	43.60
Modified YOLO-PmA [21]	-	-	-	-	-	-	-	-	-	37.91
YOLO-Fine [13]	63.45	76.77	77.91	74.35	78.12	-	64.74	70.04	45.04	68.18
SuperYOLO [10]	68.55	90.30	70.30	53.86	79.48	-	76.69	58.08	53.86	72.49
YOLOrs [11]	50.65	85.25	38.92	72.93	76.77	-	70.31	18.65	42.67	57.00
YOLOv3 Tiny	21.45	62.94	48.17	59.27	49.93	8.33	32.12	38.54	17.93	37.63
YOLOv4 Tiny	60.70	39.32	16.68	50.10	34.07	62.68	71.48	54.84	19.68	47.25
YOLOv7 Tiny	18.83	82.25	77.59	67.09	41.92	16.66	77.64	37.71	10.85	47.84
Improved Tiny YOLO	25.29	81.91	56.23	66.79	64.92	46.50	56.23	34.35	21.60	50.41

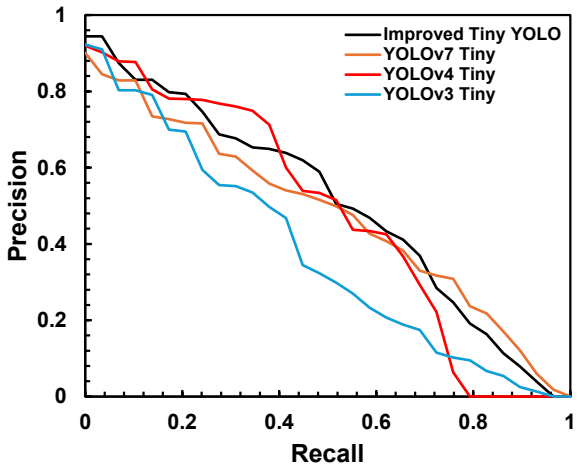

Fig. 4 Comparison of precision-recall curve.

Table 2 compares the detection accuracy per class for several object detection models. It can be observed that the Improved Tiny YOLO model showcased superior performance by achieving the highest AP in two distinct classes, 'tractor' and 'other,' with 64.92% and 21.60%. On the contrary, the YOLOv7 Tiny excelled in recognizing 'car,' 'camping car,' 'van,' and 'pickup,' while the YOLOv4 Tiny outperformed in detecting 'track,' 'plane,' and 'boat.' Importantly, the proposed model demonstrated a remarkable consistency in AP across all considered classes, resulting in a slightly higher mAP when compared to other tiny models. This underlines its effectiveness in distinguishing diverse objects with diverse features. Despite obtaining the highest AP for four out of nine classes, the YOLOv7 Tiny model recorded the least AP for the 'other' and 'truck' classes when compared to other tiny models. This led to a marginally reduced overall mAP compared to the proposed model.

A qualitative analysis was performed on three different images comprised of vehicles in diverse backgrounds, including (a) varying size, (b) occlusion, and (c) complex background. Figure 5 depicts the comparison of visual detection across various lightweight YOLO models. The proposed model and YOLOv7 Tiny model can successfully detect vehicles of different sizes, with only one false positive, as shown in Figure 5 (a). In contrast, YOLOv4 Tiny struggles to detect large

vehicles, while YOLOv3 Tiny misses some detections. In the case of occlusion, both Improved Tiny YOLO and YOLOv7 Tiny models were able to detect most of the occluded vehicles, while there were some false negatives obtained from the YOLOv4 Tiny and YOLOv3 Tiny, as shown in Figure 5 (b). Finally, the Improved Tiny YOLO model showcased its capability by accurately detecting all vehicles, even in the challenging environments of housing areas characterized by a variety of shapes and features. While the YOLOv4 tiny model recognized a cabin as a camping car, both the YOLOv4 Tiny and YOLOv3 Tiny experienced some inaccuracies, with certain vehicles being misidentified. These outcomes are visually demonstrated in Figure 5(c).

Fig. 5 Comparison of visual detection on test images. (a) Varying sizes (b) Occlusion (c) Complex background.

Table 3 Comparative analysis of computational performance and detection time.

Model	Layers	Training time (h)	Trainable parameter (Million)	Model size (MB)	Inference time (ms)		Frame per second (FPS)	
					GeForce GTX 1650	Jetson nano	GeForce GTX 1650	Jetson nano
YOLOv3 Tiny	23	17.76	8.25	33.0	5.71	1593.30	63.4	17.3
YOLOv4 Tiny	37	18.34	5.63	22.5	5.90	1630.45	63.3	16.6
YOLOv7 Tiny	98	26.89	5.78	23.1	7.44	1715.78	63.5	14.3
Improved Tiny YOLO	78	24.24	3.05	12.2	6.95	1692.22	63.4	15.1

The test images demonstrate the capabilities of the proposed model, achieving improved detection accuracy in a range of image conditions.

3.2 Comparison of speed and computational performance

Table 3 provides a detailed comparison focusing on detection speed and computational performance. The findings show that the proposed model operates at a slightly higher training time compared to YOLOv7 Tiny but slower than the speed of both YOLOv4 Tiny and YOLOv3 Tiny models. This is mainly attributed to the complexity of the architecture, which corresponds to the number of layers. Deeper architectural designs necessitate longer computation times because they process each image through a more extensive series of operations. Besides, the Improve YOLO model has achieved an impressive reduction in size, resulting in a remarkably tiny model size of only 12.2 MB. This represents a significant decrease of 47.2%, 45.8%, and 63.0% compared to the YOLOv7 Tiny, YOLOv4 Tiny, and YOLOv3 Tiny models, respectively. This size reduction was made possible by implementing various modifications that have substantially reduced the total number of trainable parameters in the network. The ELAN structure used in the backbone and neck network helps to minimize the computational overhead related to feature aggregation. This module efficiently combines features from different scales, which helps in reducing redundant computations. In addition, MixConv presents a flexible approach to depthwise convolution, adapting the sizes of the convolution kernels in response to the dimensions of the feature maps. This method effectively minimizes computational demands while maintaining the performance of the model.

The models were evaluated on NVIDIA Jetson Nano and GeForce GTX 1650 in order to analyze their real-time performance. The Improved Tiny model showed a marginal improvement in inference time compared to YOLOv7 Tiny but slower than the other tiny models on both hardware. Additionally, the proposed model reached a frame rate of 63.4 FPS on the GeForce GTX 1650, which is on par with other tiny models. On the Jetson Nano, the model recorded an FPS of 15.1, which is marginally slower than that achieved by the YOLOv3 Tiny and YOLOv4 Tiny models, yet it outperforms the YOLOv7 Tiny model. This indicates that the proposed model can operate at near real-time speeds on embedded devices.

The distribution of FLOPs values for an input size of 512×512 is compared to analyze the computational complexity across different network sections as detailed in Figure 6. Notably, the Improved Tiny YOLO model demonstrated the lowest FLOPs value among the lightweight models, with 5.917 FLOPs. This represents a 42.6% reduction compared to the original YOLOv4 Tiny model. This illustrates that the proposed model is less complex than the YOLOv4 Tiny model. In the backbone network, the proposed model achieved a FLOPs value of 1.917, which is the smallest compared to the other models. This value is approximately 3.9 times smaller than the conventional YOLOv4-tiny model and about half the size of the YOLOv3 Tiny and YOLOv7 Tiny models. This demonstrates the effectiveness of integrating MixELAN modules into the backbone network. These modules not only improve the accuracy of the proposed model but also decrease the parameters generated within the network.

The inclusion of the modified PANet in the neck section has increased the FLOPs value by 0.447 FLOPs compared to the YOLOv4 tiny model. This increase results from the modified PANet, which consists of two layers with four SELAN modules aimed at providing a richer and more detailed feature representation through feature propagation within the network. Additionally, the integration of SELAN blocks helps to reduce computational complexity further. SELAN helps streamline the processing pipeline and reduce redundant computations by efficiently aggregating features at different scales. The YOLOv4 Tiny model employed only two Convolution-Batchnorm-Leaky (CBL) blocks, leading to a very low FLOPs value. On the other hand, the FLOPs value is slightly higher than the YOLOv3-tiny model but lower than the YOLOv7 Tiny model. The YOLOv3 Tiny model incorporated only two CBL blocks in the neck network, while the YOLOv7 Tiny model utilized a complex PANet architecture comprising multiple CBL and ELAN blocks. The FLOPs value in the prediction network is the highest compared to other lightweight models because of the additional prediction layer. Besides, the modified PANet structure in the neck section has extracted three feature maps of three different sizes, which slightly increases the FLOPs value in the network.

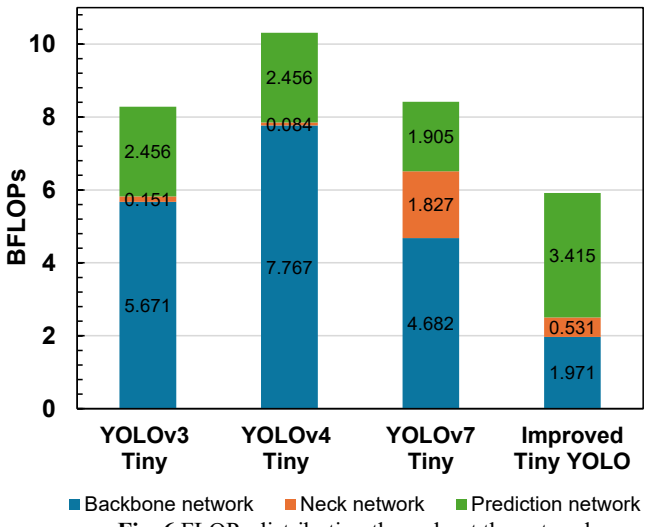

Fig. 6 FLOPs distribution throughout the network.

3.3 Future works and applications

In the future, it would be beneficial to incorporate satellite imagery with additional multimodal data fusion such as synthetic aperture radar data, infrared imagery, or LiDAR data. This integration could improve detection accuracy, especially in challenging conditions such as poor lighting. Additionally, the proposed model can be further enhanced by implementing attention mechanisms, which would allow the model to concentrate on relevant parts of the image. Lastly, the proposed model can be optimized for deployment on edge devices, enabling real-time processing of satellite images directly on-board satellites or UAVs. The developed model proposed in this paper is beneficial for numerous real-time applications, such as traffic monitoring and management. It can help optimize traffic flow and reduce congestion in urban areas. Additionally, it can be utilized for security and surveillance, such as monitoring unauthorized vehicle movements in restricted areas. Furthermore, the model can be deployed for rapidly assessing vehicle movement and density in disaster-stricken areas, helping to efficiently allocate rescue and relief efforts.

4 Conclusion

This study proposed an enhanced YOLO-based detection model, called Improved Tiny YOLO, to detect multi-class vehicles in satellite images. Several modifications were integrated into the original YOLOv4 Tiny model in order to improve the accuracy and efficiency of the proposed model. First, three MixELAN modules were employed to replace the original CSP blocks to improve accuracy while reducing the number of parameters generated within the network. Then, the PANet was restructured using SELAN to promote more comprehensive utilization of features from various levels of the network. Finally, an additional detection layer is added to the prediction network, and the Swish activation function is utilized, replacing the Leaky ReLu function to further enhance the detection performance. The proposed model has shown significant improvement in terms of accuracy, achieving an mAP of 50.41% with an increment of 6.69% from the baseline model of the YOLOv4 Tiny model. Besides, the proposed model is significantly less complex than the other tiny models, with a BFLOP value of 5.917, indicating lower computational requirement for the training process. Moreover, the proposed model obtained a remarkably small size of 12.2MB and achieved near real-time speed on Jetson Nano. Based on these advantages, the proposed model is well-suited for real-time vehicle detection on satellite images using embedded devices.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

M.H.J: Writing original draft, Methodology development, Experiments and data analysis, Conceptualization, Software utilization, Investigation, Data visualization. A.S.M.K: Conceptualization, Writing-review & editing, Investigation, Validation. E.A.B: Writing-review & editing, Validation. A.F.H: Writing-review & editing, Validation.

Informed Consent Statement

Not applicable.

Acknowledgment

This work was supported in part by the Short-Term Research Grant by Universiti Sains Malaysia with project number R501-LR-RND002-0000000130-0000.

References

[1] H. V. Koay, J. H. Chuah, C. O. Chow, Y. L. Chang, and K. K. Yong, "YOLO-RTUAV: Towards real-time vehicle detection through aerial images with low-cost edge devices," *Remote Sens.*, vol. 13, no. 21, pp. 1–26, 2021, doi: 10.3390/rs13214196.

[2] S. Razakarivony and F. Jurie, "Vehicle detection in aerial imagery: A small target detection benchmark," *J. Vis. Commun. Image Represent.*, vol. 34, pp. 187–203, 2016, doi: 10.1016/j.jvcir.2015.11.002.

[3] A. Froidevaux *et al.*, "Vehicle Detection and Counting from VHR Satellite Images: Efforts and Open Issues," in *IEEE nternational Geoscience and Remote Sensing Symposium*, 2020, pp. 256–259, doi: 10.1109/IGARSS39084.2020.9323827.

[4] L. Pulvirenti, L. Rolando, and F. Millo, "Energy management system optimization based on an LSTM deep learning model using vehicle speed prediction," *Transp. Eng.*, vol. 11, no. January, p. 100160, 2023, doi: 10.1016/j.treng.2023.100160.

[5] M. Abbasi, A. Shahraki, and A. Taherkordi, "Deep learning for network traffic monitoring and analysis (NTMA): A survey," *Comput. Commun.*, vol. 170, no. January, pp. 19–41, 2021, doi: 10.1016/j.comcom.2021.01.021.

[6] Z. Yang and L. S. C. Pun-Cheng, "Vehicle detection in intelligent transportation systems and its applications under varying environments: A review," *Image Vis. Comput.*, vol. 69, pp. 143–154, 2018, doi: 10.1016/j.imavis.2017.09.008.

[7] L. Liu *et al.*, "Deep Learning for Generic Object Detection: A Survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, 2020, doi: 10.1007/s11263-019-01247-4.

[8] L. Jiao *et al.*, "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, no. 3, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once : Unified, real-time object detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
[10] J. Zhang, J. Lei, W. Xie, Z. Fang, Y. Li, and Q. Du, "SuperYOLO: Super resolution assisted object detection in multimodal remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–14, 2023, doi: 10.1109/TGRS.2023.3258666.
[11] M. Sharma *et al.*, "YOLOrs: Object detection in multimodal remote sensing imagery," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 14, pp. 1497–1508, 2021, doi: 10.1109/JSTARS.2020.3041316.
[12] Q. Xu, Y. Li, and Z. Shi, "LMO-YOLO: A Ship Detection Model for Low-Resolution Optical Satellite Imagery," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, pp. 4117–4131, 2022, doi: 10.1109/JSTARS.2022.3176141.
[13] M. T. Pham, L. Courtrai, C. Friguet, S. Lefèvre, and A. Baussard, "YOLO-fine: One-stage detector of small objects under various backgrounds in remote sensing images," *Remote Sens.*, vol. 12, no. 15, pp. 1–26, 2020, doi: 10.3390/RS12152501.
[14] M. F. Humayun, F. A. Nasir, F. A. Bhatti, M. Tahir, and K. Khurshid, "YOLO-OSD: Optimized Ship Detection and Localization in Multiresolution SAR Satellite Images Using a Hybrid Data-Model Centric Approach," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 5345–5363, 2024, doi: 10.1109/JSTARS.2024.3365807.
[15] A. Momin, M. Haniff, J. Anis, and S. Mohd, "Lightweight CNN model : automated vehicle detection in aerial images," *Signal, Image Video Process.*, 2022, doi: 10.1007/s11760-022-02328-7.
[16] A. Betti and M. Tucci, "YOLO-S : A lightweight and accurate YOLO-like network for small target detection in aerial imagery," *Sensors*, vol. 23, p. 1865, 2023.
[17] Y. Yang, Z. Miao, H. Zhang, B. Wang, and L. Wu, "Lightweight Attention-Guided YOLO With Level Set Layer for Landslide Detection From Optical Satellite Images," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 17, pp. 3543–3559, 2024, doi: 10.1109/JSTARS.2024.3351277.
[18] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, "Designing network design strategies through gradient path analysis," 2022, doi: 10.6688/JISE.202307_39(4).0016.
[19] M. Tan and Q. V. Le, "MixConv: Mixed depthwise convolutional kernels," 2019.
[20] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path Aggregation Network for Instance Segmentation," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8759–8768, doi: 10.1109/CVPR.2018.00913.
[21] A. N. Amudhan and A. P. Sudheer, "Lightweight and computationally faster Hypermetropic Convolutional Neural Network for small size object detection," *Image Vis. Comput.*, vol. 119, p. 104396, 2022, doi: 10.1016/j.imavis.2022.104396.

Mohamad Hanif Junos was born in Perak, Malaysia. He received a B.S. degree in Aerospace Engineering from Universiti Sains Malaysia, in 2013 and an M.S. degree from Universiti Teknologi Malaysia in 2016. He obtained his Ph.D. degree in Electrical Engineering from University Malaya. He is currently working as a senior lecturer at the School of Aerospace Engineering, Universiti Sains Malaysia. His research interests include computer vision, aerial object detection, and lightweight detection models.

Anis Salwa Mohd Khairuddin received a Bachelor of Electrical Engineering from the Universiti Tenaga Nasional, Malaysia, in 2008, and a Master of Computer Engineering from the Royal Melbourne Institute of Technology (RMIT), Australia in 2010. She graduated with her Ph.D. degree in Electrical Engineering from Universiti Teknologi Malaysia, in 2014. She is currently working as a senior lecturer at the Department of Electrical Engineering, Faculty of Engineering, Universiti Malaya, Malaysia. Her research interests are in the areas of Expert systems (machine learning, optimization), Agriculture robotics and automation (classification, control system, intelligent system) and image processing.

Ahmad Faizul Hawary earned his Ph.D. in Flight Control Engineering from the University of Southampton, in 2016. Prior to that, he completed his M.Sc. in Robotics and Automation at the University of Salford, in 2008, and his B.Sc. in Mechatronic Engineering from Universiti Teknologi Malaysia, in 1999. He has been a faculty member at the School of Aerospace Engineering, Universiti Sains Malaysia since 2008. Before joining USM, he gained extensive industrial experience. Throughout his academic career, he has been actively involved in numerous consultancy projects for private companies. He has also contributed significantly to the advancement of drone technology, collaborating with local companies to design and produce drones for the Malaysian market. Dr. Ahmad Faizul's expertise spans embedded control systems, robotics and automation, mobile robots, and drone

technology, and he continues to make impactful contributions to both academia and industry.

Elmi Abu Bakar is an Associate Professor in the School of Aerospace Engineering, Universiti Sains Malaysia. He is also serving as deputy dean of research at USM. He received all his higher education from Japan (Dip. Eng Mechanical at Kisarazu, Bachelor Engineering Mechanical at Iwate, Master Engineering Production System at Toyohashi, and PhD certificates in Electronics and Information at Toyohashi, Japan). His research interests include control systems and robotics, machine vision (image-based measurement), abnormal detection using signal processing methods, shape classification and analysis, CAD, tool & die quality inspection, and computer-aided inspection.