SINERGI Vol. 30, No. 1, February 2026: 185-196
http://publikasi.mercubuana.ac.id/index.php/sinergi
http://doi.org/10.22441/sinergi.2026.1.017

An effective and efficient vehicle detection using ER-EMA-YOLOv10n

Imanuel Kutika*, Vicky Nolant Setyanto Lahimade, Tomi Heri Julianus Todingan, Hebron Prasetya, Steven Ray Sentinuwo, Muhamad Dwisnanto Putro
Master Program of Informatics, Postgraduate Program, Sam Ratulangi University, Indonesia

Abstract

Vehicle detection plays a key role in automating traffic analysis, a field that continues to advance rapidly. Vision-based systems identify vehicle types and sizes, but achieving high accuracy and efficiency remains a challenge. Reliable real-world deployment requires optimized models that balance performance and computational cost. YOLOv10n, the most efficient version of the YOLO family, offers a solid foundation for lightweight feature extraction. To improve its detection performance, this study proposes an enhanced version of YOLOv10n by incorporating a scale-aware attention mechanism. We proposed the Expanded Refinement Efficient Multi-Scale Attention (ER-EMA) module, which enhances feature encoding by capturing vehicle characteristics across multiple receptive fields. ER-EMA consists of two core components: the Expanded Converted Inverted Block (ECIB) and the Convolutional Refinement Block (CRB). These components use diverse convolutional kernels to extract and refine multi-frequency spatial features. Integrating ER-EMA into the YOLOv10n framework produces a more compact and accurate detection model. Experimental results show that the proposed model increases mAP@50 by 1%, while reducing the number of parameters by 0.1M and computation by 0.1 GFLOPS on the Vehicle-COCO dataset. On the UA-DETRAC benchmark, it achieves a 4% improvement in mAP@50:95, with a reduction of 0.2M in parameters and 0.4 GFLOPS in computational efficiency—outperforming the original YOLOv10n and prior methods in both performance and computational efficiency.

This is an open-access article under the CC BY-SA license.

Keywords:
Deep Learning;
Efficient Model;
Vehicle Detection;
YOLOv10;

Article History:
Received: May 9, 2025
Revised: July 12, 2025
Accepted: July 24, 2025
Published: January 16, 2026

Corresponding Author:
Imanuel Kutika
Master Program of Informatics, Postgraduate Program, Sam Ratulangi University, Indonesia
Email: imanuelkutika026@student.unsrat.ac.id

INTRODUCTION

Urbanization and population growth have significantly increased vehicle use in both urban and rural areas, contributing to traffic congestion and longer commuting times [1][2]. In 2024, individuals around the world spent an estimated 88 hours annually, on average, stuck in traffic congestion [3]. It is urgent to address these challenges by developing accurate, efficient vehicle detection systems that adapt to dynamic road conditions and traffic scenarios.

Object detection, a subfield of Vision Intelligence, becomes significant in intelligent transportation systems. Modern detection methods, primarily based on CNN frameworks, have achieved considerable success by learning discriminative features directly from visual data [4, 5, 6]. In this context, the YOLO variant models have gained prominence for adjusting detection accuracy with computational efficiency. The recent introduction of YOLOv10 offers improvements over its predecessors through strategies such as model pruning, architecture simplification, and the novel “Consistent Dual Assignment” mechanism, which excludes the need for Non-Maximum Suppression (NMS) [7][8].

YOLOv10 includes various scaled versions, with the YOLOv10n model tailored for low latency and edge device deployment. Although YOLOv10n is efficient, it may still underperform in scenarios with scale variation, partial occlusion, or cluttered scenes. To enhance performance under these conditions, attention mechanisms such as the Efficient Multi-scale Attention (EMA) block have been introduced to improve spatial feature representation by capturing relationships across different resolutions [9]. However, current EMA-based approaches often lose essential fine-grained details due to limited feature refinement capabilities during extraction.

Recent studies highlight a growing interest in optimizing YOLO-based architectures across various application domains. For instance, YOLOv8 variants have been adapted for specialized tasks such as fish and sea turtle detection in marine biology [10, 11, 12], as well as tomato and wheat spike recognition in agricultural environments [13][14]. In aerial surveillance, BGF-YOLOv10 and LD-YOLOv10 have been developed to detect small objects from unmanned aerial vehicles, focusing on enhancing detection accuracy while maintaining low computational cost [15][16]. Within the transportation domain, YOLOv3-tiny and YOLOv4-tiny have been applied to vehicle detection tasks, including traffic violation monitoring and autonomous driving systems [17][18]. Other models, such as LittleYOLO-SPP, ShortYOLO-SPP, YOLOv5-NAM, and YOLOv5-IPA_MSSCR [19, 20, 21, 22], further illustrate efforts to balance real-time performance with detection robustness, particularly under challenging conditions involving occlusion or small objects.

While recent studies have demonstrated significant progress in adapting various YOLO-based models, most of these works focus on lightweight optimization and domain-specific enhancements. However, these models often struggle to handle scale variation, occlusion, and dense object configurations. YOLOv10 represents the latest evolution of the YOLO framework, introducing key innovations, namely the Consistent Dual Assignment strategy and improved architectural efficiency. Despite these advancements, limited research has explored the synthesis of refined multi-scale attention techniques within the YOLOv10 framework to enhance performance in complex environments, such as vehicle detection under dense traffic, across varying object scales, and under partial occlusion. This gap underscores the need to enhance YOLOv10 with attention-based modules further to improve its robustness in real-world, high-density detection tasks.

To address this gap, this study proposes an enhanced vehicle detection framework that elevates YOLOv10n with a novel feature-extraction module for lightweight detectors. The model integrates the ER-EMA module into the backbone, improves feature quality through better spatial representation, and scale adaptability. By doing so, it aims to achieve higher detection accuracy without compromising speed or resource efficiency. Designed for edge-device compatibility, this method supports real-time applications on low-budget hardware. The core contributions of this work are:

- Introducing a novel vehicle detection framework to localize vehicles optimally using modified YOLOv10-nano, which achieves better performance and efficient cost.
- Expanded Refinement of Efficient Multi-Scale Attention (ER-EMA) aims to enhance feature extraction performance while maintaining efficiency by combining it with the original EMA.
- The Expanded Convolution Inverted Block (ECIB) increases channel capacity by incorporating normalization and activation functions, enabling more effective feature extraction.
- Convolutional Refinement Block (CRB) to refine and optimize feature representations.
- Comprehensive performance analysis, runtime efficiency, and ablation studies were conducted on the proposed architecture and compared against several efficient object detection models from previous work and attention modules.

METHOD

In this section, the proposed architecture is explained in detail. This section focuses on improving vehicle detection performance. The Pyramid-based Spatial Pooling in YOLOv10 is designed to expand the receptive field during feature extraction, enabling it to adapt to objects with diverse dimensions. Figure 1 shows the operation of spatial max pooling in 2D with different window sizes (5, 9, 13). The Partial Self-Attention (PSA) block, adapted from the self-attention mechanism, enhances global feature representation while maintaining computational efficiency, as illustrated in Figure 1. By focusing on a subset of feature channels, PSA captures global context with lower overhead than full self-attention, balancing accuracy and efficiency. Both SPPF and PSA were modified by adding an ER-EMA block at the end to enrich feature extraction.

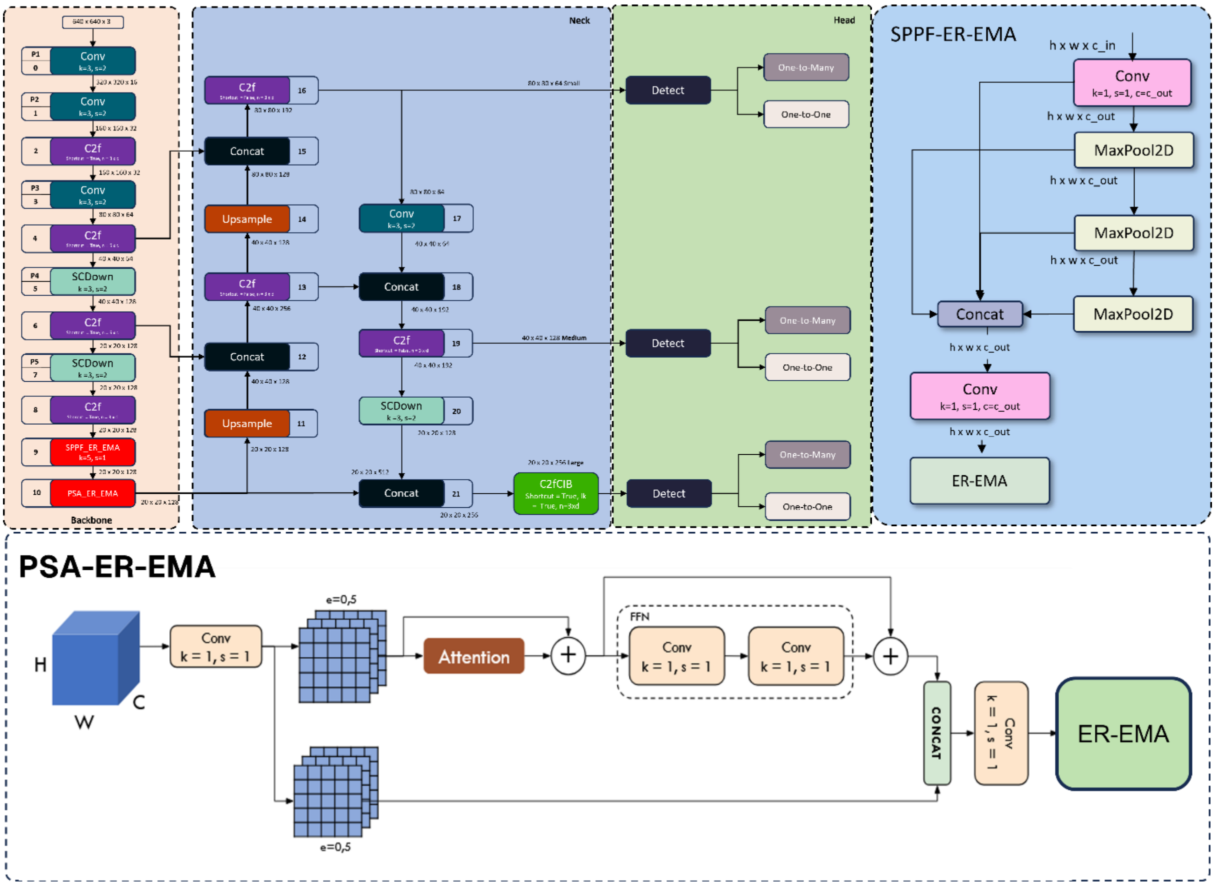


Figure 1. ER-EMA-YOLOv10n architecture with SPPF and PSA Block. It consists of a backbone, neck, and head to detect, distribute, and extract information using the ER-EMA on the SPPF and PSA block.

ER-EMA-YOLOv10n Architecture

The backbone of the ER-EMA-YOLOv10n architecture serves as the core feature extractor, leveraging multi-kernel convolutional layers to learn object patterns through iterative weight updates during training. The Pyramid-based Spatial Pooling in YOLOv10 is designed to expand the receptive field during feature extraction, enabling it to adapt to objects with diverse dimensions. It operates on spatial max pooling in 2D with different window sizes (5, 9, 13).

The Partial Self-Attention (PSA) block, adapted from the self-attention mechanism, enhances global feature representation while maintaining computational efficiency. By focusing attention on a subset of feature channels, PSA effectively captures global context with lower overhead than full self-attention, offering a balance between accuracy and efficiency. Block SPPF and PSA were both modified by adding an ER-EMA block at the end to enrich the extraction features.

The neck in the YOLOv10n architecture is a crucial component that integrates features at multiple levels and scales from the backbone, aligning the resolution between the head and the backbone. The head module is designed to classify each detected object and calculate the probability for each object class. It is a regression predicting each bounding box, including the One-To-Many head that sustains the native form and adjustment goal to produce a collection of forecasts. The model also incorporates a One-to-One head with an alignment-based label-matching strategy, ensuring a unique correspondence between each ground-truth label and a single prediction. It uses CIoU loss to measure convergence between matched prediction–ground-truth pairs, accounting for both spatial overlap and aspect ratio [23].

Expanded Refinement of Efficient Multi-Scale Attention (ER-EMA)

To create a robust detection model, simultaneous detection of multiple vehicle types is required. This module seeks to improve EMA (Efficient Multi-Scale Attention) by boosting performance and enhancing feature extraction across different scales. This module also enhances the network's ability to identify vehicles of varying sizes. ER-EMA module is located in each block of the SPPF and PSA. This work includes several block components working together to relate features at different frequencies within the proposed method.

ER-EMA consists of the EMA attention module as the context, and the ECIB and CRB blocks, as shown in Figure 2. EMA starts with feature grouping to partition X into G feature subsets spanning the stream dimensions direction for learning and extracting different semantics with G // C, learnt emphasis weight descriptor sets to strengthen the feature encoding of interest region in each subset as described as follows:

$$X = [X_0, X_i, \dots, X_{G-1}], X_i \in R.^{C//G \times H \times W} \quad (1)$$

Parallel subnetworks in EMA associate two features with the image height and share a 1×1 convolution, without decreasing dimensionality in the 1×1 path stream, using a process resembling Coordinate Attention (CA). After the 1×1 convolution, the feature map is split into two branch vectors, proceeding with the application of the Sigmoid activation function to model the binary distribution of the 2D convolutional output [24]. EMA enables distinctive trans-channel associations between the two symmetrical sub-networks at the 1×1 branch, where the feature maps from both channels are combined element-wise within the model architecture. Conversely, the 3×3 convolution branch acquires proximal trans-channel feature cross-talk, thereby expanding the representation. This module adjusts the magnitudes of diverse channels and maintains exact spatial data blocks within them.

Cross-spatial learning is a strategy for encoding holistic context and modeling broad associations. The primary spatial attention representation is obtained by applying a matrix product to the synchronous times' outputs. Similarly, 2D holistic mean pooling is used in the section to extract holistic spatial data and generate a subsequent spatial weight representation that preserves spatial address details. Activation maps for clusters are calculated, combining the two-weight points. The 2D holistic pooling operation is as follows:

$$Z_C = \frac{1}{H \times W} \sum_j^H \sum_i^W X_C \ (i, j). \quad (2)$$

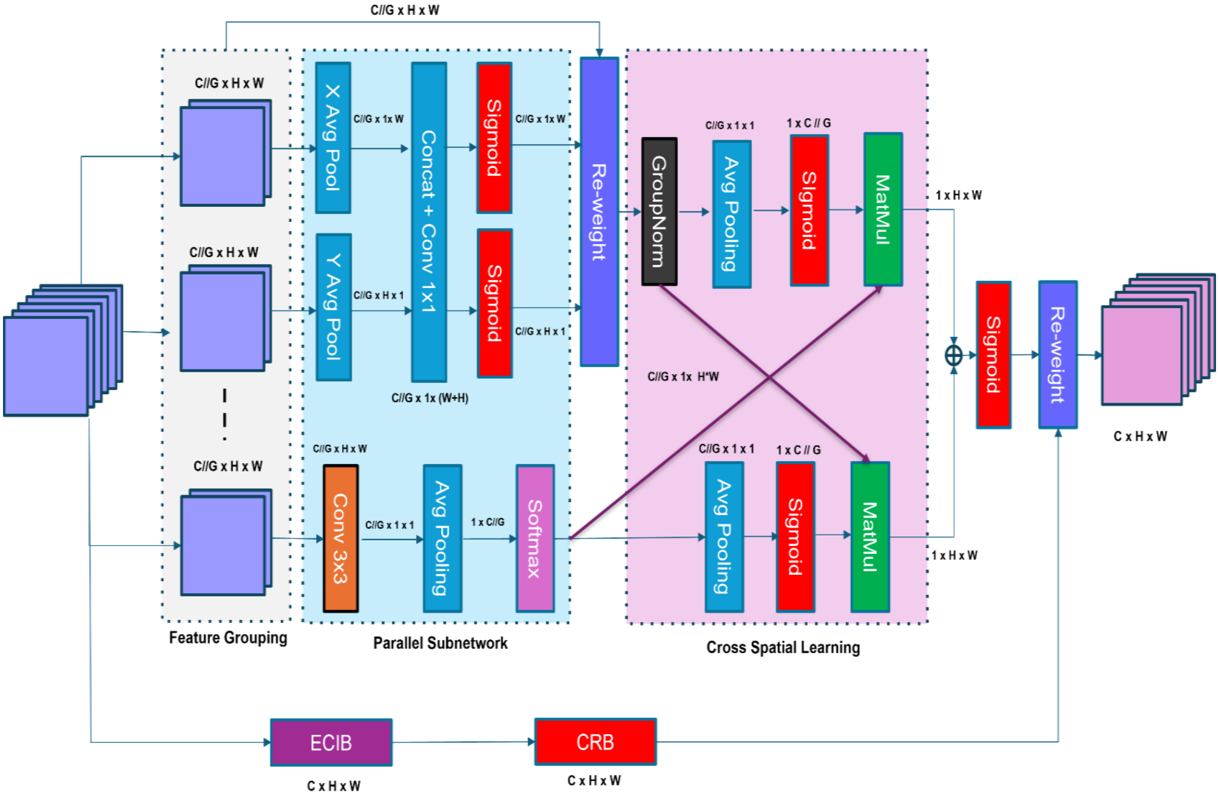


Figure 2. The ER-EMA attention module is integrated into both the SPPF and PSA blocks within the ER-EMA-YOLOv10n, enhancing multi-scale feature extraction and expanding contextual representation

This work introduces a new block in the new branch that provides a representation of the existing feature map, as shown in Figure 3. This block has strong potential to be further developed into a unified module that enhances performance while maintaining computational efficiency. The ECIB begins with the input features and applies 3×3 depthwise convolutions, referred to as CDW1, as shown:

$$ECIB = CDW_5(C_4(CDW_3(C_2(CDW_1(x))))), \quad (3)$$

$$CDW_1 = GELU\,(GN\,(C_{DW(3\times3)}(x))) \quad (4)$$

In the subsequent step, Group Normalization (GN) [25] is applied to the input data, producing more stable feature representations and facilitating faster training following the convolution operation. GELU (Gaussian Error Linear Unit) is applied, providing a smoother representation and preserving information by offering more stable gradients for complex learning [26]. The process continues with 1×1 convolution as C2, as shown:

$$C_2 = GELU\,(GN\,(C_{(1\times1)}(CDW_1))). \quad (5)$$

$$REP = SILU(\,(C_{DW3\times3}(C_2)) + (C_{DW7\times7}(C_2))). \quad (6)$$

REP is a REPVGGDW block that performs depthwise convolutions with 3×3 and 7×7 kernels to obtain a broad feature representation with SILU (Sigmoid Linear Unit), improving the smoothness of gradient flow, preventing the loss of features. After REP, the process continues on GN and GELU activation, resulting in C3. Then, on C4, following a similar process on C2. It continues and concludes at CDW5, which applies a 3×3 convolution followed directly by GELU activation, as shown:

$$CDW_3 = GELU\,(GN\,(REP)). \quad (7)$$

$$C_4 = GELU\,(GN\,(C_{(1\times1)}(CDW_3))). \quad (8)$$

$$CDW_5 = GELU\,(GN\,(C_{DW(3\times3)}(C_4))). \quad (9)$$

CRB begins by performing a 3×3 convolution, which effectively captures fine details such as the edges and textures of small shapes. This process continues with batch normalization to maintain convergence on data features. The sigmoid function is used as the activation to determine the probability values of the generated feature map as follows:

$$CRB = \sigma\,(BN\,(C_{(3\times3)}(x))). \quad (10)$$

In Figure 3, the process ends with a 1×1 convolution that mixes the features to produce a better representation. The result will be concatenated with EMA modules to improve ER-EMA performance.

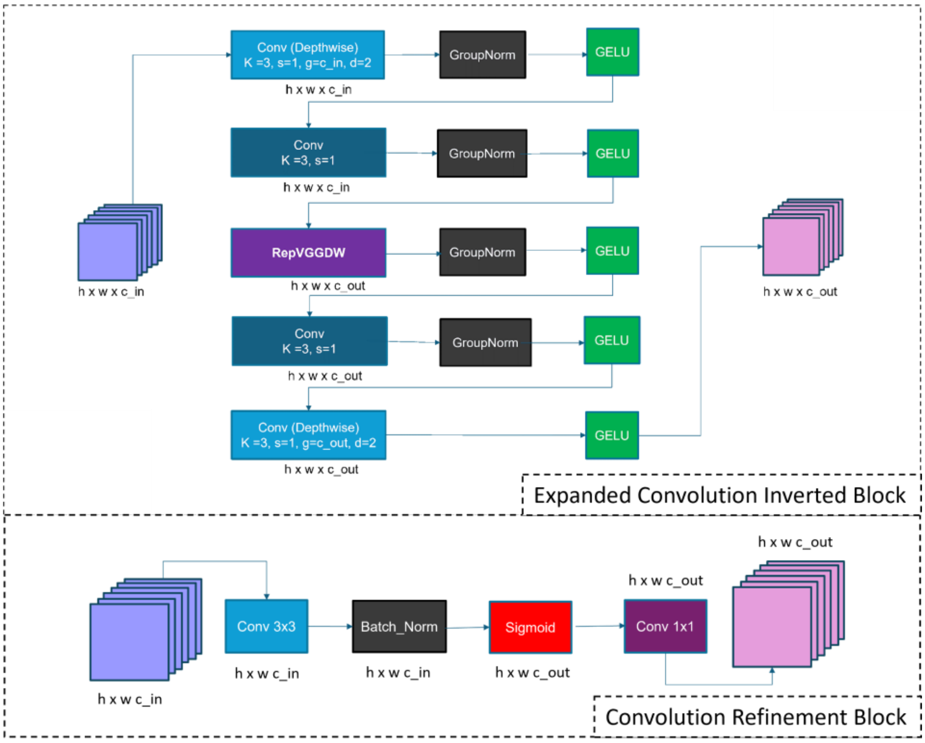


Figure 3. ECIB and CRB block on the ER-EMA attention module

RESULTS AND DISCUSSION

Training Configuration

Table 1 outlines the proposed research using a high-specification computer for deep learning experiments. Training data is processed on an NVIDIA Colorful GPU with 12 GB of RAM, which is well-suited for training complex models. The feed graphic is resized to 640 × 640 for efficiency, without sacrificing spatial information accuracy.

Training was conducted for 300 epochs on the Vehicle-COCO dataset, then for 100 epochs on the UA-DETRACT dataset, with the exact requirements applied to both. Training data using a batch size of 16 to ensure stable training. The Stochastic Gradient Descent (SGD) solver was used owing to its robustness in handling large datasets. The learning rate was set at 0.01 to achieve a reach fitting rate. These configurations help achieve optimal results in vehicle detection. The method and evaluation are executed on the CPU, as shown in the inference stage in Table 2. This setup allows for evaluating the architecture's competence under limited computational resources.

Evaluation of the Dataset

The dataset is part of the MS COCO dataset family and contains 18,998 images, with 13,300 for training and 3,788 for validation, as shown in Table 3. The images are labelled into four classes: car, bus, truck, and motorcycle. This dataset was used in previous research by Chauduri, as shown in Figure 4(a) [17].

Another dataset used in this work is UA-DETRACT. It consists of 100 surveillance traffic-tracking videos collected from 24 locations under various weather conditions. It was partitioned into 60 group timelines for the training phase and 40 for validation [27], as detailed in Table 4. It annotates four vehicle types: car, bus, van, and others, as shown in Figure 4(b). This dataset was used in previous research by Yuhai Wang et al. [22]. The Vehicle-COCO dataset provides a diverse set of vehicle types and complex scenes for general object detection. UA-DETRAC represents real-world traffic scenarios. Together, they demonstrate the robustness across varied scenes, backgrounds, and viewpoints.

The dataset evaluation demonstrates the model's accuracy and reliability across various road conditions and highlights its robustness in detecting vehicles. The ablation study is an important phase for specifically evaluating the model's development by analyzing each component of the architecture. This allows for confirming which changes contributed to the performance improvements.

Table 5 compares the ER-EMA-YOLOv10n architecture with previous methods. The outcome demonstrates that the designed framework delivers enhanced accuracy along with efficiency. Specifically, ER-EMA-YOLOv10n exhibits enhanced capability in detecting various vehicle types, outperforming other frameworks in terms of precision, comparison, and resource usage. Compared to ShortYOLO-CSP, the framework increases mAP@50 by 0.8% while reducing the parameter count by 6.1 million and lowering the computational load by 4.6 GFLOPS.

Table 1. Training and Evaluation Setup

Properties	**Deployment**
Device	AMD Ryzen 5 3500 6-Core
GPU	NVIDIA RTX Colorful 12 GB
Image Size	640 × 640 pixels
Epochs	300(Vehicle-COCO), 100 (UA-DETRACT)
Batch Size	16
Optimizer	SGD
Learning Rate	0.01

Table 2. Inference Setup

Properties	***Deployment***
System Software	Ubuntu
Compiler	Python 3.9.20
Framework	Pytorch 2.0
CPU	AMD Ryzen 5 3500 6-Core
Visual Dimension	640 × 640 pixels

Table 3. Vehicle-Coco Dataset properties

Properties	***Visual data***
Training Data	13,300
Validation Data	3,788

Table 4. UA-DETRACT Dataset properties

Properties	***Visual data***
Training Data	83,791
Validation Data	56,340

Table 5. Evaluation of the proposed architecture versus other models on the Vehicle-COCO dataset

Model	***GFLOPS***	***Para Meter (M)***	***mAP 50 %***	***mAP 50:95 %***
YOLOv3-tiny [19]	12.9	8.7	46.1	-
LittleYOLO-SPP [19]	12.9	8.7	52.9	-
ShortYOLO-CSP [20]	12.9	8.7	63.3	-
YOLOv5-NAM [21]	16.0	7.0	56.1	-
ER-EMA-YOLOv10n	8.3	2.6	**64.5**	**45.2**

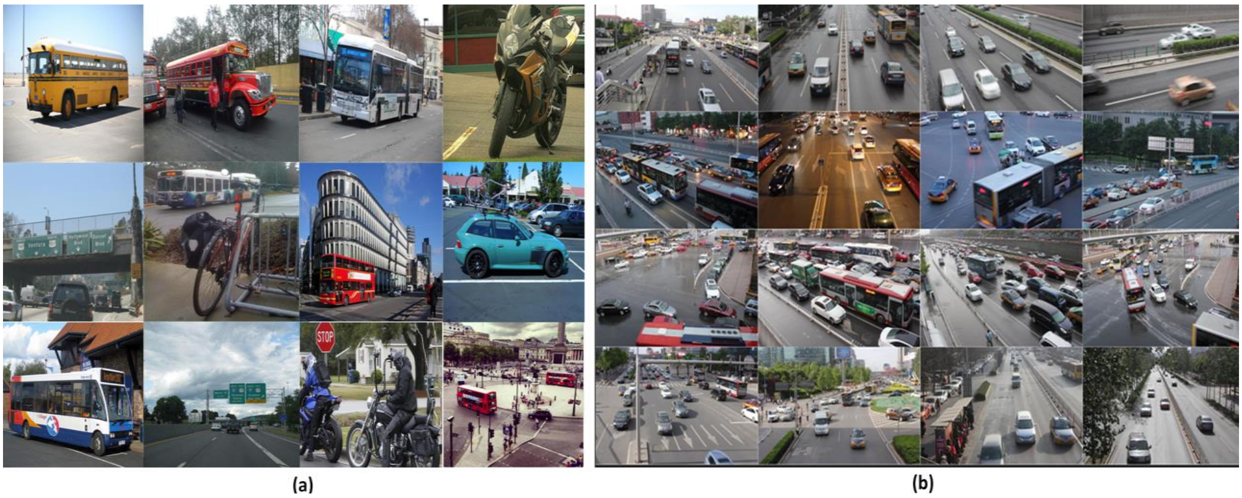


Figure 4. (a) Vehicle COCO Dataset, (b) UA-DETRACT Dataset

Table 6 highlights the architecture's performance in classifying and distinguishing the diversity of vehicles on the traffic road. ER-EMA-YOLOv10n demonstrates superior performance in detecting various vehicle types, outperforming other models in both accuracy and resource usage. It performs better than previous works such as YOLOv5-IPA-MSCCR, achieving an improvement of 8.4% in mAP@50. In addition, this model reduces the parameter count by 3.9 million and decreases the computational cost by 7.3 GFLOPS, making it more efficient.

Table 7 presents the comparison of the proposed model with several attention modules used in the YOLOv10n. In this experiment, YOLOv10n was augmented with various attention modules under identical conditions and tested on a 20-second video, with inference spanning 420 frames. This demonstrates that the proposed model outperforms other attention-based variants in both accuracy and efficiency, with only a minimal compromise in inference speed.

The slight decrease in average FPS is attributed to the inclusion of proposed blocks, which introduce an additional processing branch, thereby affecting detection speed. Compared to CBAM, the proposed model achieves superior accuracy: 64.5% mAP@50 vs. 63.7%, and 45.2% mAP@50:95 vs. 44.4%. In terms of computational efficiency, the proposed model also maintains a more favorable balance between accuracy and computational cost, with 0.1 GFLOPS less and 0.1 M fewer parameters than the original YOLOv10n.

In terms of detection comparison, Figure 5 (a) shows that the original YOLOv10 encounters difficulties in detecting vehicle objects with precision across different types, with some vehicles not detected in (a). Conversely, Figure 5(b) shows higher accuracy and consistent object detection, enabling the capture of even small vehicle objects.

The evaluation results for the proposed architecture, tested on the UA-DETRACT validation subset datasets, are shown in Figure 6. It demonstrates significant upgrades in precision and efficiency compared to the YOLOv10n variant. Figure 6(a) shows that the original YOLOv10n version has difficulty detecting vehicles with high precision across different types. In contrast, Figure 6(b) shows improved precision with consistent object detection, yielding accurate bounding boxes that improve recognition.

Table 6. Evaluation of the proposed architecture with another framework on UA-DETRACT

Model	*GFLOPS*	*Para Meter (M)*	*mAP 50 %*	*mAP 50:95 %*
YOLOv3-tiny-FPGA [18]	5444	8.7	62.5	-
YOLOv5-NAM [21]	16.0	7.0	51.2	-
YOLOv6n [22]	11.3	4.6	53.6	-
YOLOv7-tiny [22]	13.0	6.0	47.2	-
YOLOv5-IPA-MSCCR [22]	15.6	6.4	56.5	-
ER-EMA-YOLOv10n	8.3	2.5	**64.1**	**48.2**

Table 7. Evaluation of the novel model alongside other attention modules on the Vehicle-COCO dataset

Model	*GFLOPS*	*Para Meter (M)*	*mAP 50 %*	*mAP 50:95 %*	*Avg. FPS*
YOLOv10n	8.4	2.7	63.7	44.2	23.1
YOLOv10n-EMA	7.7	1.9	63.8	44.3	22.4
YOLOv10n-CA	7.7	1.9	63.8	44.6	24.3
YOLOv10n-ECA	7.7	1.9	63.5	44.6	23.7
YOLOv10n-ELA	7.7	1.9	63.8	44.6	22.6
YOLOv10n-SE	7.7	1.9	63.6	44.6	24.2
YOLOv10n-CBAM	8.2	2.5	63.7	44.6	22.2
ER-EMA-YOLOv10n	**8.3**	**2.6**	**64.5**	**45.2**	**20.8**

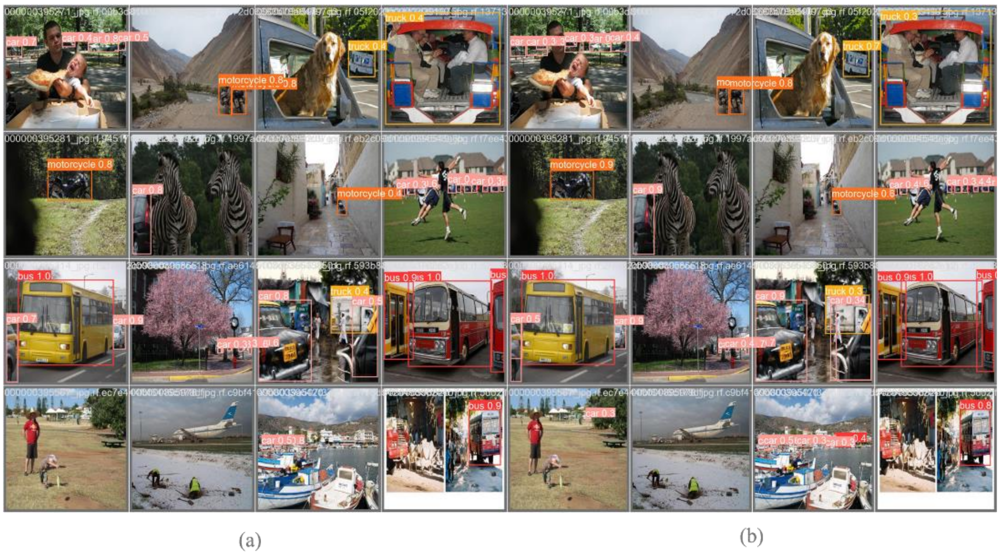


Figure 5. Comparisons on Vehicle-COCO dataset: (a) YOLOv10n, (b) ER-EMA-YOLOv10n

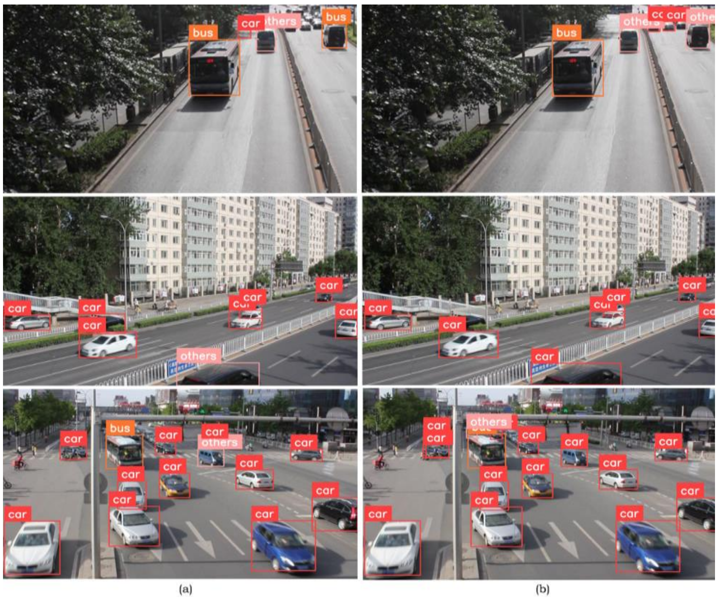


Figure 6. Comparison on UA-DETRAC dataset: (a) YOLOv10n original, (b) ER-EMA-YOLOv10-n

Heatmap results of the ER-EMA-YOLOv10n model on the COCO dataset, focusing on the detection of different vehicle types with varying scale features in Figure 7(a). Moderate-activation areas (green, yellow, and red) contribute to general object recognition, while low-activation areas (dark blue) represent backgrounds and irrelevant regions. The proposed model improves detection accuracy compared to the YOLOv10n base model. When compared, the metrics show mAP@50 is 64.5% versus 63.7%, and mAP@50-95% is 45.2% versus 44.2%. Accuracy gained 07 – 1%.

On Figure 7(b), the heatmap of the ER-EMA-YOLOv10n model on the UA-DETRACT dataset, focusing on the detection of different vehicle types on the road traffic. Moderate-activation areas (green, yellow, and red) contribute to general object recognition, while low-activation areas (dark blue) represent backgrounds and irrelevant regions. The proposed model improves performance over the original, reducing background noise. The accuracy shows improvement, indicated by mAP@50-95% of 48.2%, compared to the original mAP@50-95% of 44.2%. These evaluations show that ER-EMA-YOLOv10n performs better at feature extraction, improves accuracy, and offers greater computational efficiency than the original YOLOv10n. Because it focuses on important visual features while ignoring less relevant areas, it makes it a more robust and optimized vehicle detection.

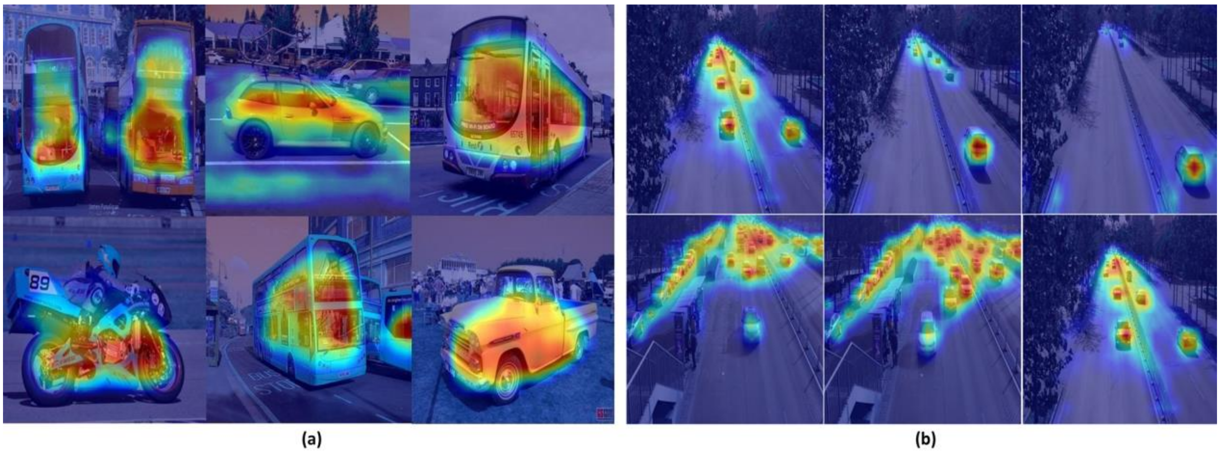


Figure 7. (a) Heatmaps on various vehicle types in Vehicle-COCO dataset, (b) Heatmap on traffic condition in UA-DETRACT dataset.

Ablation Study

Both datasets were evaluated and analyzed in ablation studies. The ablation study focuses on the impact of each component of the model architecture and its contribution to improved performance. The dataset evaluation demonstrates the accuracy and reliability of the vehicle detection model across various road conditions and highlights its robustness in addressing the challenges of vehicle detection. The ablation study is a crucial phase for specifically evaluating model developments by analyzing each component of the architecture. It allows for confirming which changes contributed to the performance improvements.

Table 8 presents the results of the ablation study conducted across various modifications of the YOLOv10n model. It examines the individual contributions of the ECIB and CRB components, as well as their combined effect. It highlights the performance improvements enabled by integrating these modules while maintaining computational efficiency. The proposed module combination yields a 1% increase in mAP@50:95, indicating a notable enhancement in detection performance. In terms of model efficiency, the approach reduces the number of parameters by 0.1 million and the computational cost by 0.1 GFLOPs.

Table 9 presents the ablation study comparing the baseline YOLOv10n, combined EMA-YOLOv10n, and the ER-EMA-YOLOv10n. The proposed method demonstrates a significant performance improvement, achieving a 4% increase in mAP@50:95. In terms of computational efficiency, the framework also reduces the parameter count by 0.2 million and the computational load by 0.4 GFLOPs. As a note, experiments conducted on the UA-DETRAC architecture were slightly modified by discarding the CRB block and using only the ECIB block. To further optimize the receptive field, the ECIB block was adapted with a dilation factor of 2, effectively enlarging it without increasing the parameter count or computational complexity. This adjustment is important given the nature of the UA-DETRAC dataset, which consists of extensive video footage captured from road surveillance, resulting in a high volume of similar or near-identical images.

Table 8. Ablation study about the proposed model on the Vehicle-COCO dataset

Model	*GFLOPS*	*Para Meter (M)*	*mAP 50 %*	*mAP 50:95 %*
YOLOv10n	8.4	2.7	63.7	44.2
YOLOv10n_EMA	8.5	2.7	64.1	44.9
YOLOv10n_EMA_ECIB	8.0	2.6	63.5	44.1
YOLOv10n_EMA_CRB	8.0	2.6	63.8	44.7
ER-EMA-YOLOv10n	8.3	2.6	**64.5**	**45.2**

Runtime Efficiency

Runtime efficiency is a critical metric in evaluating the usability of vehicle objects. It measures model efficiency in terms of speed, parameter efficiency, and computational demand. It provides a comparative analysis of the runtime between YOLOv10n and ER-EMA-YOLOv10n using a CPU.

Table 9. Ablation study about the proposed model on the UA-DETRACT dataset

Model	*GFLOPS*	*Para Meter (M)*	*mAP 50 %*	*mAP 50:95 %*
YOLOv10n	8.4	2.7	58.9	44.2
EMA-YOLOv10n	8.5	2.7	59.3	44.3
ER-EMA-YOLOv10-n	8.0	2.5	**64.1**	**48.2**

As shown in Table 7, the ER-EMA-YOLOv10n model shows a significant improvement in runtime efficiency. There is a decrease of 0.1 M in the number of parameters on the Vehicle-COCO dataset and 0.2 M in the UA-DETRACT dataset. There is also an efficiency of 0.1 GFLOPS in the Vehicle-COCO dataset and 0.4 GFLOPS in the UA-DETRACT dataset. These improvements can be attributed to the depthwise technique and to pruning the number of channels in the backbone, resulting in a more efficient architecture.

The ER-EMA module shows a slight decrease in inference speed due to the additional branch introduced by the ECIB and ERB blocks, which increases memory access and computational overhead. Although depthwise convolutions reduce the number of parameters, they also increase CPU memory usage. As a result, the ER-EMA-YOLOv10n achieves 2.3 FPS lower performance compared to the original YOLOv10n.

Table 10. Real-time Setup on Embedded System

Properties	***Deployment***
Device	NVIDIA Jetson Nano B-01 4 GB
CPU	Quad-Core ARM ® Cortex ®-A57 MPCore Processor
GPU	128-core NVIDIA Maxwell™ architecture

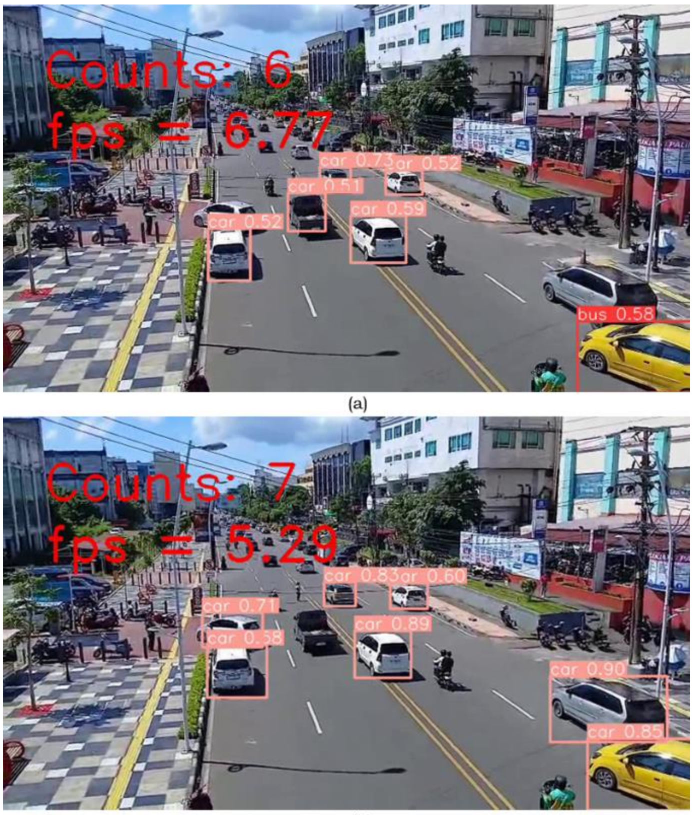
Real-World Application

Real-world application testing demonstrates the model's performance on embedded systems. The Jetson Nano served as the deployment platform, with the setup detailed in Table 10.

The proposed ER-EMA-YOLOv10n model exhibits a slight reduction in inference speed, with a decrease of 0.87 FPS on the Vehicle-COCO dataset and 3.03 FPS on the UA-DETRAC dataset, as shown in Table 11. This decline is primarily attributed to the additional computations introduced by the ER-EMA module. As illustrated in Figure 8, the model was deployed in a real-world system, demonstrating improved vehicle detection performance. Despite the trade-off in speed, the improvement in detection accuracy and robustness justifies the added complexity, rendering the model suitable for real-time applications, including those in resource-constrained environments.

Table 11. Comparison of FPS between the proposed model and the original on a real-world application

Model	***Dataset***	***FPS***
YOLOv10n	Vehicle-COCO	7.69
ER-EMA-YOLOv10n	**Vehicle-COCO**	**6.82**
YOLOv10n	UA-DETRACT	9.02
ER-EMA-YOLOv10n	**UA-DETRACT**	**5.99**



(a)

(b)

Figure 8. Real-World application on vehicle detection. (a) YOLOv10n, (b) ER-EMA-YOLOv10n

CONCLUSION

This research presents approaches to improving the performance of vehicle detection models, first by introducing ER-EMA-YOLOv10n models that use YOLOv10n as the main architecture and ER-EMA, a cutting-edge attention module. ER-EMA consists of two novel blocks: the Expanded Convolution Inverted Block (ECIB) and the Convolutional Refinement Block (CRB). This model is trained on two datasets: Vehicle-COCO and UA-DETRACT. Our method significantly improves performance over the baseline model and previous work. The ER-EMA-YOLOv10n version achieves 64.5% mAP 50% accuracy, surpassing the original model by 0.7%, and 45.2% mAP 50-95% accuracy, exceeding the original by 1% on the Vehicle-COCO dataset.

Significant performance improvement is also demonstrated on the UA-DETRACT dataset. It produces a mAP of 50% (64.1%), surpassing the original by 5.2%, and a mAP of 50-95% (48.2%), exceeding the original by 4%. In addition, our model reduces the parameter count and GFLOPS, improving compute resource efficiency. On the Vehicle-COCO dataset, the model has 2,646,552 parameters, 62,032 fewer than the original model, and 8.3 GFLOPS, which is 0.1 more efficient than the original model. Similar efficiency is observed on the UA-DETRACT dataset, with 2,572,040 parameters, 136,644 fewer than the original model, and 8.0 GFLOPS, which is 0.4 more efficient than the original. These findings demonstrate that ER-EMA-YOLOv10n offers a balanced improvement in both precision and computational efficiency, making it more robust for vehicle detection.

Future work plans could focus on elevating the performance of the YOLOv10n architecture by adapting and deploying the proposed model in real-world applications, such as vehicle counting for intelligent traffic management systems. Plans will focus on improving runtime efficiency, as the current model performs more slowly than previous work, and will also direct further efforts toward real-world application deployment.

ACKNOWLEDGMENT

The authors are sincerely thankful to the AIVISION research team for their immense support and expertise in computer vision and deep learning, for providing the necessary resources.

REFERENCES

[1] M. Kruszyna, P. Śleszyński, and J. Rychlewski, “Dependencies between demographic urbanization and the agglomeration road traffic volumes: Evidence from Poland,” *Land*, vol. 10, no. 1, Art. no. 47, 2021, doi: 10.3390/land10010047

[2] Y. Gao and J. Zhu, "Characteristics, Impacts and Trends of Urban Transportation," *Encyclopedia*, vol. 2, pp. 1168–1182, 2022, doi: 10.3390/encyclopedia2020078.

[3] S. Xu, C. Sun, and N. Liu, “Road congestion and air pollution - Analysis of spatial and temporal congestion effects,” *Sci. Total Environ.,* vol. 945, p. 173896, Oct. 2024, doi: 10.1016/j.scitotenv.2024.173896.

[4] S. Dwijayanti, R. Agam, and B. Y. Suprapto, “Comparative study of CNN techniques for tuberculosis detection using chest X-ray images from Indonesia,” *SINERGI*, vol. 29, no. 2, pp. 485–496, Jun. 2025, doi: 10.22441/sinergi.2025.5.018.

[5] M. D. Putro, D. -L. Nguyen and K. -H. Jo, "A Fast CPU Real-Time Facial Expression Detector Using Sequential Attention Network for Human–Robot Interaction," in *IEEE Transactions on Industrial Informatics*, vol. 18, no. 11, pp. 7665-7674, Nov. 2022, doi: 10.1109/TII.2022.3145862.

[6] A. Suryanto, D. H. Wicaksono, A. Mulwinda, M. Harlanu, and M. N. Syah, “Car seatbelt monitoring system using a real-time object detection algorithm under low-light and bright-light conditions,” *SINERGI*, vol. 29, no. 3, pp. 771–778, Oct. 2025, doi: 10.22441/sinergi.2025.3.108.

[7] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Mach. Learn. Knowl. Extr.*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[8] M. Hussain and R. Khanam, "In-depth review of YOLOv1 to YOLOv10 variants for enhanced photovoltaic defect detection," *Solar*, vol. 4, no. 3, pp. 351–386, Jun. 2024.M. M. Taye, "Theoretical Understanding of Convolutional Neural Network: Concepts, Architectures, Applications, Future Directions," *Computation*, vol. 11, p. 52, 2023, doi: 10.3390/solar4030016.

[9] D. Ouyang et al., "Efficient Multi-Scale Attention Module with Cross-Spatial Learning," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* Rhodes Island, Greece, 2023, pp. 1-5, doi: 10.1109/ICASSP49357.2023.10096516

[10] R. Bharath, H. U. Ujwal, K. Chandan, R. C. Ravindranath, R. Chandramma and H. R.

Rakshith, "Advancing Marine Species Detection: A Comparative Analysis of YOLOv5 and YOLOv8 in Challenging Underwater Environment," *2024 First International Conference for Women in Computing (InCoWoCo)*, Pune, India, 2024, pp. 1-8, doi: 10.1109/InCoWoCo64194.2024.10863206.

[11] Zhang and S. Chen, "Research on Improved Lightweight Fish Detection Algorithm Based on Yolov8n," *J. Mar. Sci. Eng.*, vol. 12, p. 1726, 2024, doi: 10.3390/jmse1210172.

[12] M. D. Putro, D. G. S. Ruindungan, R. Syahputra, T. -H. Oh, I. Y. Chun and V. C. Poekoel, "An Efficient and Effective Sea Turtle Detection Using Positioning Enhancement Module," *2024 International Workshop on Intelligent Systems (IWIS)*, Ulsan, Korea, Republic of, 2024, pp. 1-6, doi: 10.1109/IWIS62722.2024.10706029.

[13] A. Li *et al.*, "D3-YOLOv10: Improved YOLOv10-Based Lightweight Tomato Detection Algorithm Under Facility Scenario," *Agriculture*, vol. 14, p. 2268, 2024, doi: 10.3390/agriculture14122268.

[14] S. Guan *et al.*, "Real-time Detection and Counting of Wheat Spikes Based on Improved YOLOv10," *Agronomy*, vol. 14, p. 1936, 2024, doi: 10.3390/agronomy14091936.

[15] J. Mei and W. Zhu, "BGF-YOLOv10: Small Object Detection Algorithm from Unmanned Aerial Vehicle Perspective Based on Improved YOLOv10," *Sensors*, vol. 24, p. 6911, 2024, doi: 10.3390/s24216911.

[16] X. Qiu *et al.*, "LD-YOLOv10: A Lightweight Target Detection Algorithm for Drone Scenarios Based on YOLOv10," *Electronics*, vol. 13, p. 3269, 2024, doi: 10.3390/electronics13163269.

[17] A. Chaudhuri, "Smart traffic management of vehicles using faster R-CNN based deep learning method," *Scientific Reports*, vol. 14, no. 1, p. 10357, 2024, doi: 10.1038/s41598-024-60596-4.

[18] J. Zhai, B. Li, S. Lv and Q. Zhou, "FPGA-based vehicle detection and tracking accelerator," *Sensors*, vol. 23, no. 4, p. 2208, 2023, doi: 10.3390/s23042208.

[19] E. Rani et al., “LittleYOLO-SPP: A delicate real-time vehicle detection algorithm,” *Optik*, vol. 225, p. 165818, 2021, doi: 10.1016/j.ijleo.2020.165818.

[20] P. E. Rani and S. S. Jamiya, "ShortYOLO-CSP: A decisive incremental improvement for real-time vehicle detection," *J. Real-Time Image Process.*, vol. 20, no. 1, p. 3, 2023, doi: 10.1007/s11554-023-01256-0

[21] J. Wang, Y. Dong, S. Zhao and Z. Zhang, "A high-precision vehicle detection and tracking method based on the attention mechanism," *Sensors*, vol. 23, no. 2, p. 724, 2023, doi: 10.3390/s23020724

[22] Y. Wang *et al.*, "Lightweight vehicle detection based on improved YOLOv5s," *Sensors*, vol. 24, no. 4, p. 1182, 2024, doi: 10.3390/s24041182

[23] S. Du, B. Zhang, P. Zhang and P. Xiang, "An Improved Bounding Box Regression Loss Function Based on CIOU Loss for Multi-scale Object Detection," *2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML)*, Chengdu, China, 2021, pp. 92-98, doi: 10.1109/PRML52754.2021.9520717.

[24] S. R. Dubey, S. K. Singh and B. B. Chaudhuri, "Activation Functions in Deep Learning: A Comprehensive Survey and Benchmark," *Neurocomputing*, vol. 503, pp. 92–108, 2022, doi: 10.1016/j.neucom.2022.06.111.

[25] L. Huang, J. Qin, Y. Zhou, F. Zhu, L. Liu and L. Shao, "Normalization Techniques in Training DNNs: Methodology, Analysis and Application," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 8, pp. 10173-10196, Aug. 2023, doi: 10.1109/TPAMI.2023.3250241.

[26] A. Nguyen, K. Pham, D. Ngo, T. Ngo and L. Pham, "An Analysis of State-of-the-art Activation Functions for Supervised Deep Neural Network," *2021 International Conference on System Science and Engineering (ICSSE)*, Ho Chi Minh City, Vietnam, 2021, pp. 215-220, doi: 10.1109/ICSSE52999.2021.9538437.

[27] L. Wen *et al.*, "UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking," *Comput. Vis. Image Understand.*, vol. 193, p. 102907, 2020, doi: 10.1016/j.cviu.2020.102907.