Jurnal Rekayasa Elektrika Vol. 20, No. 3, September 2024, page 116-124
ISSN. 1412-4785; e-ISSN. 2252-620X, Accredited by Kemendikbudristekdikti No. 105/E/KPT/2022
DOI: 10.17529/jre.v20i3.35236

Streamlining Deep Learning Network for Real-time Sea Turtle Detection

Muhamad Dwisnanto Putro[1*], Yuliana Mose[2], Alex Copernikus Andaria[2], Jane Litouw[1], Vecky Canisius Poekoel[1], and Xaverius Najoan[1]
[1]Department of Electrical Engineering, Faculty of Engineering, Sam Ratulangi University
Kampus Bahu, Manado, Indonesia 95112
[2]Department of Computer Engineering, Faculty of Science, Technology and Teaching, Trinita University
Malalayang, Manado, Indonesia 95163
*e-mail: dwisnantoputro@unsrat.ac.id

***Abstract*—Monitoring turtle behavior is a conservation effort to preserve its habitat, and the detection process is a vital initial stage. On the other hand, robotics demands a deep learning network that can operate in real-time to detect the presence of sea turtles automatically. The need for increased model speed in the inference stage has led to many lightweight vision-based detectors. This work proposes a novel turtle detection to localize multiple sea turtles using a deep learning method. A lightweight primary extractor is applied to distinguish crucial features without producing a substantial computational. An excited group attention is offered as an enhancement module that can capture essential turtle components in multi-level convolutional patches. A new turtle dataset is proposed that contains lighting, blur, occlusion, and complex background challenges. The evaluation results show that the proposed model performs higher accuracy than other lightweight object detection models. High-efficiency benefits models that can be implemented on low-end devices in terms of real-time data processing speed.**

Keywords: sea turtle detection, deep learning, vision system, efficient model, underwater robot

I. Introduction

Indonesia plays a significant role in providing crucial breeding and foraging habitats for numerous marine species. Several rare species of turtles are found in Indonesia and live spread across millions of islands. Knowing the location of turtles is also essential to understanding the behavior, distribution, and even population of these turtles. Turtle is a rare underwater animal that can maintain the balance of marine life ecosystems whose habitat is currently threatened. This animal lives and is active at sea depths of ± 100 meters. Illegal hunting and marine damage threaten these animals' lives. Conservation can reduce the negative impact of the extinction of these animals. Observation and research activities on marine animals are popular underwater research topics and have a rapid development trend [1]. Monitoring and observing their life can help researchers study their behavior and habits.

In the current era, the utilization of robotic technology and the recognition of objects through image processing hold significant importance. The capability to accurately detect marine items in its application is beneficial for autonomous marine robots. Numerous studies demonstrate that using deep learning in image processing to detect marine objects implemented on underwater robots or vehicles yields significant findings for research. The implementation of robotics involves an automated underwater robot that can be used to monitor sea turtles' behavior while observing their habitat's development. It replaces the life-threatening work of divers due to dynamic weather and environmental conditions [2]. This robot requires a turtle detector to localize and detect the turtle's presence first, which is the initial process of analyzing the behavior. The turtle detection is vital to finding the location of this animal, which is the basic information of the robot to observe and learn the activities of the turtle automatically. Object detection can be utilized by discriminating against turtle features from other marine objects.

Along with object detection in different fields, it can be applied as a vision system for underwater robots. The challenge is that marine objects are difficult to detect with refraction from lighting [3]. Turtles are almost extinct animals and are among the most protected animals in the world. Therefore, turtle detection is urgently needed to help preserve this endangered animal so that it can be protected due to extinct habitat caused by natural changes and illegal hunting. There are various types of famous turtles in Indonesian islands, for example, the olive ridley turtle, hawksbill turtle, and green turtle. Turtle detection is applied to recognize the characteristics and texture of turtles so that it can be used in more effective conservation monitoring and studying their behavior and migration patterns. Automatic detection using vision methods can contribute to conserving living creatures and their fragile ecosystems [4].

Modern methods present convolutional neural networks

Received 18 November 2023; Revised 26 December 2023; Accepted 02 January 2024

(CNN) as a feature extraction method that effectively discriminates object features from the background [5]. The filter operation can find a typical characteristic of the target object by applying trained weights. The self-learning encourages the network to independently determine kernel weights by estimating prediction errors. It is proven to reliably highlight important information from objects, although it requires expensive computational energy when employing deep convolutional layers. Even some work [6] and [7] apply this method to recognize an object from an input image. Instead of getting saturation in the training process, they achieved satisfactory performance.

In its development, the CNN algorithm can also be applied to find distinctive features of marine animal and plant objects [8]. The challenges of blur, object scale, and minimal lighting do not reduce the performance of this approach to finding distinctive features from marine objects. Studies [3], [4], [9] apply this method to find the location of marine animals that have a fundamental role in the underwater environmental conservation process. The detection object relies on feature extraction to find valuable information. So, this part plays an important role in achieving high precision. The appearance of marine objects tends to be contaminated by blur and low illumination. These challenges make it difficult for simple object detection to achieve satisfactory performance.

The efficacy of employing deep learning techniques with substantial datasets has been empirically demonstrated across various domains, facilitating accurate identification and classification tasks. Using an intelligent algorithm in the context of underwater items involved the development of a region-convolutional neural network (R-CNN) or Faster R-CNN model [3], [4]. This model aimed to minimize the duration required for identifying and classifying marine objects, specifically fish, since both the R-CNN and Fast R-CNN algorithms utilize selective search to identify region proposals, a time-consuming process affecting the network's performance. Practical applications emphasize the development of an efficient vision system that can operate rapidly on a robotic device. The system must be designed to be efficient without compromising its performance. Robot vision systems generally utilize low-cost devices with low computational consumption. It can increase the economic value of the design and production technology. Therefore, this issue is unfavorable for vision methods with slow data processing. On the other hand, machine vision work applies live streams with the help of cameras directly capturing frames. A slow system results in time delays that slow down the decision response. It is detrimental to robotic automation systems that demand real-time monitoring of marine objects.

The greatness of the CNN method encourages this study to apply a convolutional-based architecture that considers efficiency factors. It motivates this work to design a low-cost architecture operating on low-end devices. A lightweight backbone quickly extracts turtle features assisted by an attention module. The attention module can strengthen extractor performance that can capture specific features of the target object [10]. In the development of object detection, YOLOv8 has shown impressive performance in detecting objects under low illumination and image quality [11]. However, this network must improve its efficiency to be implemented in real applications. In this paper, we propose a vision system for detecting and localizing sea turtles that can operate with low computing power. This system supports the capabilities of underwater robots that can run in real time on low-cost devices. This study offers a novel enhancement module to boost feature extraction performance and employs three detection layers. It improves the feature extractor's performance without significantly reducing the network efficiency. The attention module can be utilized easily by plugging and playing on convolutional neural networks. The contribution of the work can be summarized as follows.

1. An efficient deep learning model is proposed to build novel real-time turtle detection that can operate quickly for implementation on an underwater robot. The proposed architecture improves the architecture of YOLOv8, which focuses on increasing efficiency to operate smoothly on robotic devices.
2. An excited group attention module is proposed to strengthen the backbone performance by capturing turtle-specific features inserted on the last backbone part. This novel module assigns grouped convolution and grouped attention to discriminate object features at various scales of the capture area. Separate assignments of each feature map can increase the variety of target information without compromising the savings of computational energy and parameters.
3. Comprehensive experimental results show that the proposed network achieves state-of-the-art performance compared to the lightweight YOLO family detector on the proposed turtle dataset.

The content of this paper is as follows. Section II describes the related research that summarizes previous research related to turtle detection. The proposed architecture, an improvement of YOLOv8n, is explained in detail in Section III. Section IV describes the configuration of training, testing, and the dataset. The experiments and evaluation of the proposed detector are presented in Section V. Lastly, the conclusions and future work is described in Section VI.

II. Related Works

Sea turtles have several evolutionary changes, resulting in speciation and sub-speciation. This factor causes the sea turtle population to decline. Global conservation is required to save the population, followed by an in-depth analysis with modern advances in detecting marine animals and sea turtles. It involves a machine learning and computer vision approach, which aids observation efforts [12]. Neural networks have gained traction as a powerful approach to automatically recognize marine objects across data domains. A work [13] used this

method for unoccupied aircraft systems (UAS) footage that produces fresh population-level insights. The combined approach estimates the population numbers of various marine creatures. It shows that the automation inherent in both approaches will soon enable monitoring at hitherto unrealistic geographical and temporal scales. Deep learning has demonstrated accurate performance for extracting features and localizing target objects. Feature learning from sample data discriminates important information from trivial parts. In addition, the neural network helps to generate trained weights and implements convolution operations to find valuable and useful features for predicting object location and category. A CNN-based animal identification was developed to achieve automated identification of sea turtle species [14]. It utilizes a robust extractor feature that inputs images into the network for automated categorization and recognition. This work uses preprocessing to optimize the training performance. The result was presented that the common turtles achieved satisfying accuracy.

On the other hand, the study [15] employs a multibeam imaging sonar as the primary sensor to identify turtles. The proposed approach has been implemented on the HATTORI AUV. An autonomous underwater vehicle tracks turtles without adding tags to them. Under natural conditions, the AUV followed a turtle for 270 seconds in shallow waters.

III. Architecture Detector

In this section, the proposed architecture is explained in detail and focuses on an offered module to improve the performance of turtle detection. Turtles tend to have texture information. The contours and shapes are different from other marine animals, making it easier for feature extractors to get the characteristics of the model.

Table 1. The difference between YOLOv8 nano and modified network

Component	Original YOLOv8n	Modified Network
Maximum number of channels	256	128
Attention module	Not used	EGA module

Therefore, the proposed architecture emphasizes extracting turtle information and increasing energy efficiency, which encourages this vision system to be implemented on low-cost devices for the development needs of underwater robots.

Globally, our architecture is inspired by YOLOv8 [11], which was developed to be lighter and consists of a primary extractor, attention module, path aggregate network, and detection layer. The general architecture of our detector is presented in Figure 1. This study modifies the nano version of YOLOv8 by increasing the data processing speed without neglecting the detection performance. It focuses on improving the efficiency of the backbone model by streamlining the number of feature map channels of the convolution operation and adding a reinforcement module called excited group attention (EGA) that increases the precision of turtle location prediction. These modifications are shown in Table 1.

A. Primary Extractor Module

Feature extraction plays a vital role in separating valuable features from trivial features. This process becomes the central core of an object detection network, which employs convolutional layers. This operation robustly discriminates target information by applying weighted multi-kernel. The learning process drives each filter to update weights and impose an accurate final prediction.

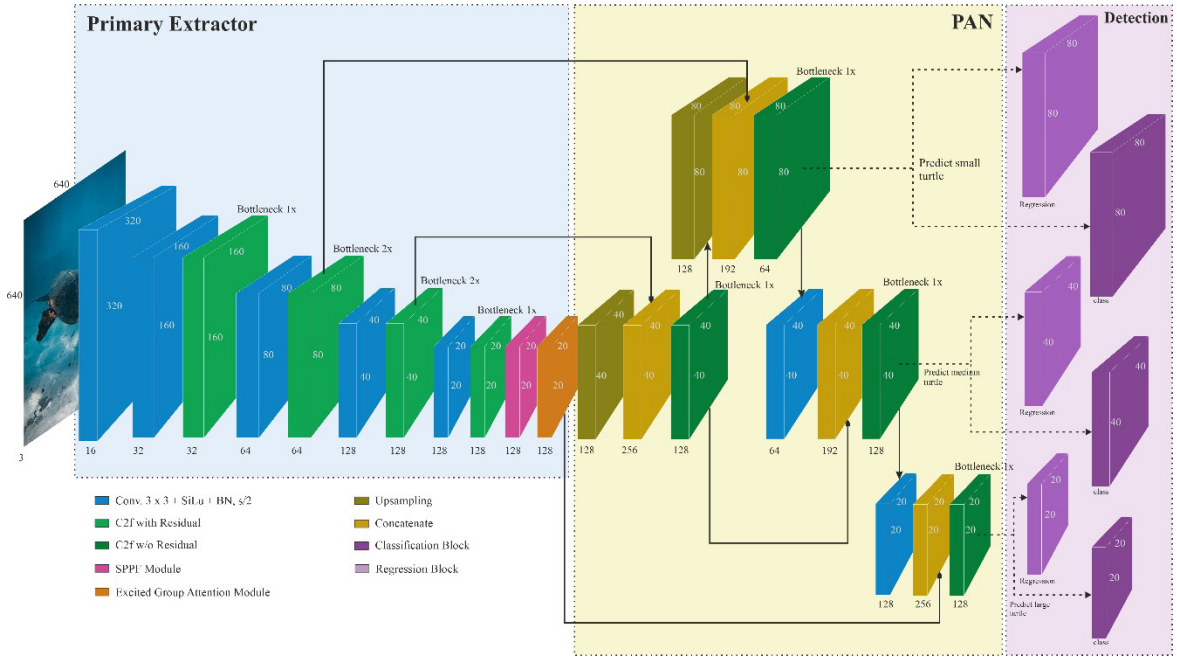

Figure 1. The general architecture of turtle detection. The YOLOv8 structure inspires the proposed architecture with lightweight operation encouraging efficient networking. An excited group attention is inserted on the backbone's last part, marked with orange. Best viewed in color.

The proposed architecture uses a primary extractor to extract essential features while reducing the feature map dimensions to reduce floating-point computation. Besides, it simultaneously increases the number of channels that can accommodate the extracted features. This structure is typical of all CNN architectures, which tend to be small in dimension and fat on the network.

This backbone adopts the YOLOv8 structure, which applies 33 convolutional layers with a stride of 2 at the initial stage to five, which can reduce map dimensions. This operation is more effective than pooling. The backbone still uses faster convolutional module called coarse-to-fine (C2F) to distinguish between turtle features and other marine components efficiently. This module utilizes two convolution layers at the beginning and end of the module and applies a bottleneck module to a split feature map. We apply the same number of bottlenecks as the nano version of YOLOv8 on each C2F module to emphasize efficiency, but we limit the maximum number of channels to 128.

This modification generates a version of YOLOv8 that is lighter than the nano version. Instead of generating massive parameters and computations, this module is slim and can operate quickly. A residual technique is used to prevent information loss due to over-extraction. In addition, each split map is combined with concatenation operations to enrich the information

Furthermore, a spatial pyramid pooling-fast (SPPF) is applied after the last C2F module on the backbone to select spatial features with cascade pooling. This module can capture the maximum value in different coverage areas for each level. This technique impacts a processing speed superior to the standard version. This sequentially applies max-pooling with a window of size 5×5, pushing the final pooling to get a receptive field of 13×13 from the input features. Combining features from each pooling layer emphasizes increasing feature variety. On the other hand, the 1×1 convolution operation is applied at the beginning and end of the module to reconstruct the number of channels.

B. Path Aggregate Network

The neck of the network determines the quality of features in object detection by relating them to multi-level features in the backbone. Feature fusion improves the relationship between elements of each convolution stage using an up-sampling and down-sampling approach to equalize the map dimensions. YOLOv8 implements a path aggregate network (PAN) that generates three levels using a bottom-up merger information path approach. It employs a C2F extractor to filter aggregated information. This module ignores residuals due to the different dimensions between the input and output maps.

The nearest method is applied to double the size of the map dimensions in the first and second stages. It generates the largest feature map on the detection layer. On the other hand, two 3×3 convolutional blocks with a stride of 2 are applied sequentially after the second stage to generate medium and small maps. The interconnection of information with backbone features applies the concatenate operation, which is needed to strengthen combination features before being forwarded to the detection module.

C. Detection Layer

Object detection networks commonly predict objects' location, size, and class. This work places the detection layer at the end of the network. The proposed turtle detection used the detection part of YOLOv8 that applies a decoupled head. It employs two branch blocks generated by two 3×3 convolutional layers and a 1×1 convolutional layer. It is applied to each branch sequentially. These two branches predict the bounding box regression and the class probability of the object. Instead of applying anchors to the regression part, it uses an anchor-free approach to avoid assignment complexity. This architecture implements three detection layers with different scale assignments. Large feature maps predict small objects, small feature maps estimate large objects, and medium feature maps detect medium objects.

Furthermore, a binary cross entropy (BCE Loss) is employed to compute the prediction error of the class probabilities. In addition, distributed focal loss (DFL) and complete intersection over union (CIoU) are used in the regression task. These two complex functions guarantee an optimal network obtains penalty for errors in predicting the location and size of the bounding boxes. A task-aligned assigner dynamically assigns instances to enhance the detection performance.

D. Excited Group Attention Module

An attention module can enhance the performance of backbone modules by capturing the essential information and reducing the unimportant elements. Several works use this model in specific areas of networks to save computational power [16], [17]. The proposed network offers EGA to automatically select the interest of turtle components using a group map, as illustrated in Figure 2. It has the motivation to divide computations into specific groups rather than calculating them across the entire channel map.

A 2D-convolutional with 1×1 kernel is applied to construct the output channel at the beginning of the process. The proposed module consists of three essential parts, including grouped convolution, shuffle, and attention. The first module generates feature maps with multi-scale representations of different receptive fields for four groups of feature maps (Gc_1, Gc_2, Gc_3, Gc_4). It applies different convolutional group splits to each of them. This operation can capture information from different map areas, which encourages variations in the formation of feature variations. This module can be formulated as follows.

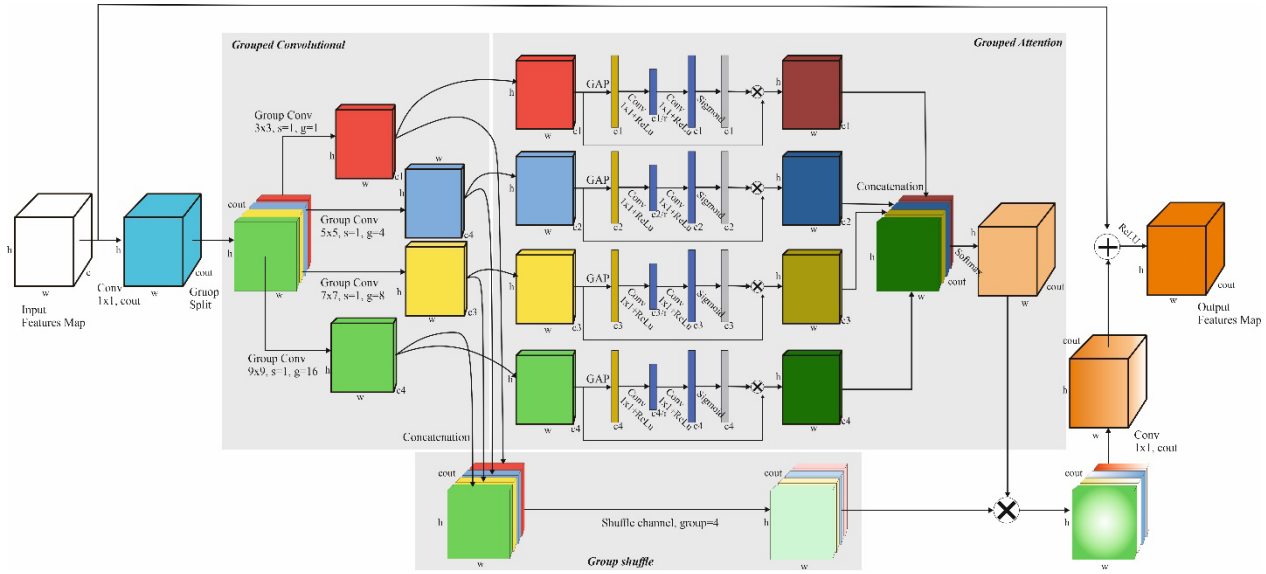

Figure 2. An excited group attention module. This module contains grouped convolutional, group shuffle, and grouped attention blocks. Best viewed in color.

$$Gc_1(x_i^1) = Conv_{3\times3}^{g=1}(x_i^1), \quad Gc_2(x_i^2) = Conv_{5\times5}^{g=4}(x_i^2),$$
$$Gc_3(x_i^3) = Conv_{7\times7}^{g=8}(x_i^3), \quad Gc_4(x_i^4) = Conv_{9\times9}^{g=16}(x_i^4), \tag{1}$$

$$[x_i^1, x_i^2, x_i^3, x_i^4] = x_i. \tag{2}$$

where x_i is the input featuressplitted to produce patches into four groups ($x_i^1, x_i^2, x_i^3, x_i^4$) with equal portions. Then, it applies different group and kernel convolutions, including 3×3 filter without group ($Conv_{3\times3}^{g=1}$), 5×5 filter with group = 4 ($Conv_{5\times5}^{g=4}$), 7×7 filter with group = 8 ($Conv_{7\times7}^{g=8}$), and filter 9×9 with group = 16 ($Conv_{9\times9}^{g=16}$).

This approach produces map output with different feature representations that strengthen distinct information. Furthermore, this module applies grouped attention to each patch that has been convolved. It adopts the SE module [10] approach that captures the specific features of each map with different information representations. This module updates the feature map with a self-learning process that generates a weighted probability vector, which is expressed as

$$SE(x_i) = \sigma\left(W_2\left(W_1\left(GAP(x_i)\right)\right)\right) \otimes x_i. \tag{3}$$

where global average pooling (GAP) summarizes input information (x_i) by taking the average score from each channel map, W_1 and W_2 are 1×1 convolution followed by ReLU activation, and σ is sigmoid activation.

This attention block captures the vital information of the object by selecting the channel-based feature map obtained from the weighted vector representation. This extraction process ensures that each patch focuses on obtaining specific turtle features by selectively increasing valuable information. To reduce the use of massive computation and parameters, it applies channel ratio to the first convolution operation. We combine all patch outputs with the concatenation operation of these block outputs as a powerful group feature representation. This map is used to improve grouped convolutional features.

A shuffle operation is applied to the groped convolutional map to blend extracted information with different receptive fields. The technique improves updating performance for respective channel elements. This mixing emphasizes ordering the channel map by randomizing the four groups. The grouped attention is operated to update the shuffle map that produces a new feature map. Then, a 1×1 convolution is employed to reconstruct channels and mix the soft information of the updated map. It applies a residual approach to ensure that essential features are lost due to absurd feature extraction.

The excited group attention is offered to improve the performance of the backbone by attaching it to the end of the backbone. Moreover, this attention module effectively increases turtle features and relieves other marine object information. This module helps improve turtle localization performance by finding distinctive features different from other marine objects. In addition, group convolutional and multi-region representation modules can avoid using abundant computation and parameters from the network, thus producing a vision detector that can operate on low-end devices.

IV. Dataset and Implementation Setup

A. Dataset

The dataset of our experiments was carefully annotated for the specific task of turtle detection. It includes underwater images sampled from videos in underwater locations with natural conditions, such as coral reefs, open water, and coastal areas. We obtained the video from the internet and agency tours and still left the label watermark attached. The entire video contains turtles typical of Bunaken National Marine Park, consisting of hawksbill, Olive Ridley, and green turtle species.

We annotate the turtle instances using Boobs Master

Table 2. Dataset configuration

Category	Train	Validation	Testing
Image	2,670	667	837
Resolution	640640 pixels		
Augmentation	Mosaic images, random color distortion, random brightness, random contrast, random cropping, and random horizontal flipping		
Acquiring	Internet and agency tours videos		
Annotation tools	Boobs Master tools [18]		

tools [18] and ignore the background that helps to provide detector knowledge precisely. This process replicates real-world scenarios and ensures the quality and accuracy of the bounding box ground truth through manual annotation. We carefully annotated each image by identifying and delineating bounding boxes around each turtle instance. This customized dataset plays a crucial role in training and evaluating the performance of our real-time turtle detection system, which incorporates the proposed network architecture. This effort produced 2,670 images for the training stage, 667 for validation, and 837 for testing. It uses the dataset configuration that adopts the work [11], which is presented in detail in Table 2.

B. Configuration Training and Testing

In this experiment, the workstation utilized Ubuntu 18.04 with 16 GB of RAM, powered by an Intel Core i5-2320 CPU and equipped with an NVIDIA GTX 1080 GPU as an accelerator graphics card. The proposed network was simulated using PyTorch and took advantage of CUDA 10.1 for efficient GPU acceleration. The training process was conducted over 300 epochs, ensuring the model underwent sufficient training iterations to detect turtles in various underwater environments.

The image size used for training and evaluation was set at 640×640 pixels. A learning rate of 0.001 and 64 batch size were employed during the training process. Then, it adopts the loss function [11] to calculate the difference between the bounding box prediction and the object class and ground truth. In the testing and deployment stage, its performance was measured on the NVIDIA Jetson Nano with 4 GB of 64-bit LPDDR4 RAM. It allowed us to evaluate the model's compatibility and efficiency in real-world scenarios with limited computational resources. The performance model is evaluated on mean average precision (mAP) that calculates the intersection over union (IoU) on 0.5 and 0.5:0.95.

V. Experiment and Result

In this section, the performance of the proposed model is evaluated on the Bunaken turtle dataset and is compared to other lightweight YOLOv8-based object detection. It utilizes the performance of YOLO that can localize objects in real-time [19]. It also analyzes the efficiency cost and measures the model speed on Jetson Nano.

A. Evaluation on Dataset

The proposed model evaluates the performance on mean average precision under IoU of 0.5 (mAP@50) and average of 0.5 until 0.95 IoU (mAP@50:95). Since this vision approach is aimed at robot systems, it compares with efficient detectors such as the lightweight YOLO family (YOLOv3 tiny, YOLOv5n, YOLOv6n, YOLOv7 tiny, and YOLOv8n). Several works have tested that these lightweight detectors perform satisfactorily for localizing marine objects on low-cost devices [20], [21], [22]. In contrast, YOLOv1 and YOLOv2 have low performance in motion blur and lighting conditions, which is a challenge for underwater video [23]. Hence, it ignores both detectors.

Based on Table 3, YOLOv5n achieves a mAP of 52%, which results in a lower performance than YOLOv8n. The quantitative results also present that our network overcomes YOLOv3 tiny. It also indicates that the YOLOV8 can achieve higher performance than previous versions. The latest version improves the convolutional backbone that generates robust extractor features. Our improvement of this model produces an efficient version, namely YOLOv8p, that generates lower precision than YOLOv8s. The improved model achieves a mAP of 58.3%. Using the YOLOv8p backbone that installed an EGA Block attention module has the best results compared to other attention modules, such as SE Block, with a difference of 9.6%. When it was compared to CBAM Block, it differed by 5.7%. The performance of the transformer module is lower than our proposed attention module, with a difference of 3.1%. The combination of the improvement backbone and the proposed attention block reaches the highest precision scores, achieving a mAP of 61.6%. Based on the confusion matrix in Figure 3, the proposed detector obtains a true positive value for predicting turtles of 0.87 on the test dataset.

The proposed detector can precisely discriminate turtle features, even under marine environments affected by different lighting, colour and texture similarities, and orientations. The proposed network uses a more robust backbone than the previous YOLO version. C2F splits the

Table 3. Comparison between the performance of the proposed model and other networks

Model	mAP@50	mAP@50:95
YOLOv3 tiny	0.837	0.533
YOLOv5n	0.824	0.520
YOLOv6n	0.772	0.481
YOLOv7 tiny	0.657	0.325
YOLOv8s	0.863	0.587
YOLOv8n	0.868	0.567
YOLOv8p	0.870	0.563
YOLOv8p-SE	0.816	0.520
YOLOv8p-CBAM	0.867	0.559
YOLOv8p-Transformer	0.909	0.585
YOLOv8p-EGA (proposed)	0.909	0.616

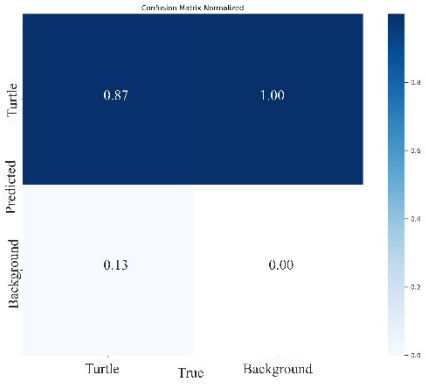

Figure 3. Confusion matrices in mAP@0.5 on test dataset

feature map, which can ease the computational process. It benefits the network by sequentially extracting features with convolution operations that can find essential turtle features precisely. In comparison, YOLOv7 tiny and YOLOv6n only gradually apply several single convolution blocks. Testing on the turtle dataset shows that these competitors perform less than our model.

On the other hand, these competitors apply 3 x 3 convolution on the main block, which increases the number of operations. YOLOv5n applies a three-convolutional (C3) block that divides the feature map by convolution operations. The backbone of this network is more efficient than the backbone of YOLOv8n. However, this competitor does not effectively strengthen the performance of the path aggregate network that interconnects features of different frequencies.

Furthermore, YOLOv3 tiny implements the DarkNet mini as the backbone, producing abundant feature map channels. Although it achieves better mAP than YOLOv5n, the competitor's extractor cannot drive more true positives than the YOLOv8p backbone. On the other hand, our backbone is suitable with a feature-level relation module involving PAN, which can strengthen the feature extraction more effectively than other competitors.

The EGA module is installed on YOLOv8p, which plays an essential role in capturing information of interest. The grouped attention module represents different spatial areas of information, increasing the variety of coverage extraction and generating the trained weights of each piece of information. This difference in information relates the object features to the background, which determines the importance value. Therefore, this module can help the extractor network to focus on turtle-specific features and strictly distinguish valuable elements from trivial information.

B. Visualization of Model Detection

In order to observe the performance by visualization, it presents the qualitative result of detection that collaborates with the class activation map. This result applies eigen-CAM [24] to display the vital feature that impacts prediction. This method computes and presents the principal elements of the learned features from the specific convolutional blocks. Therefore, it effectively obtains the representation of the target feature [25].

Figure 4 shows that the heatmap of prediction is on the top side, and the detection result is on the bottom. We also evaluate the performance of turtle detection by observing the bounding box prediction results. This experiment was carried out on a testing dataset unrelated to the training data. As a result, our model accurately recognizes the location of the turtle component, where the red area presents valuable features. On the other hand, the proposed model can find turtle locations precisely.

Typical Bunaken turtles tend to live in coral reef environments with variations in colour and texture. It is a challenge to separate turtle-specific elements from the complex background. In addition, our detection system satisfactorily detects multiple objects with various size scales and object positions. Even though the network built is lightweight, the proposed model was supported by a robust feature extractor that discriminates against turtle components. Blurred objects due to water interference did not weaken our detector from locating turtles precisely in a frame. The combination of the backbone module and attention module can filter trivial features that distinguish turtle objects without being disturbed by the noise.

C. Analyzing of Model Efficiency

Model efficiency describes the ability of a detector to be implemented in real-application cases. Even the issue demands a vision system to operate on low-cost devices [26]. The propensity of robots to use these devices gave rise to the idea of testing and installing a turtle detection system on the Jetson Nano. This device has been widely used in robotics as a main system for processing computations generated by an algorithm. In addition, this subsection also analyses the number of parameters and computational complexity cost generated by the model and compares them with other architectures.

The structure of YOLOv8p achieves higher efficiency than YOLOv8n and YOLOv5n, as shown in Table 4. Although the precision of YOLOv8p is 3% lower than that of YOLOv8n, the proposed module obtains parameters 1.6 times lower than the competing network. Additionally, YOLOv8p generates a computing cost that is lighter at 1 GFLOPs. To enhance the detection performance of YOLOv8p, an EGA module is embedded in the backbone, which produces accuracy that outperforms YOLOv8n and YOLOv8s. These results do not produce much computational overhead and parameters, so the proposed turtle detection architecture is still more efficient than other architectures. Furthermore, the model speed was investigated using video with a resolution of 640 x 480. It produced a data processing speed of 14.91 FPS, faster than the YOLOv5n and YOLOv8n models. We also evaluated the speed of the YOLOv6n and YOLOv3 tiny detectors. Both detectors are slower than YOLOv8n and the proposed detector. These results prove that the proposed detector can operate quickly with satisfactory performance and is

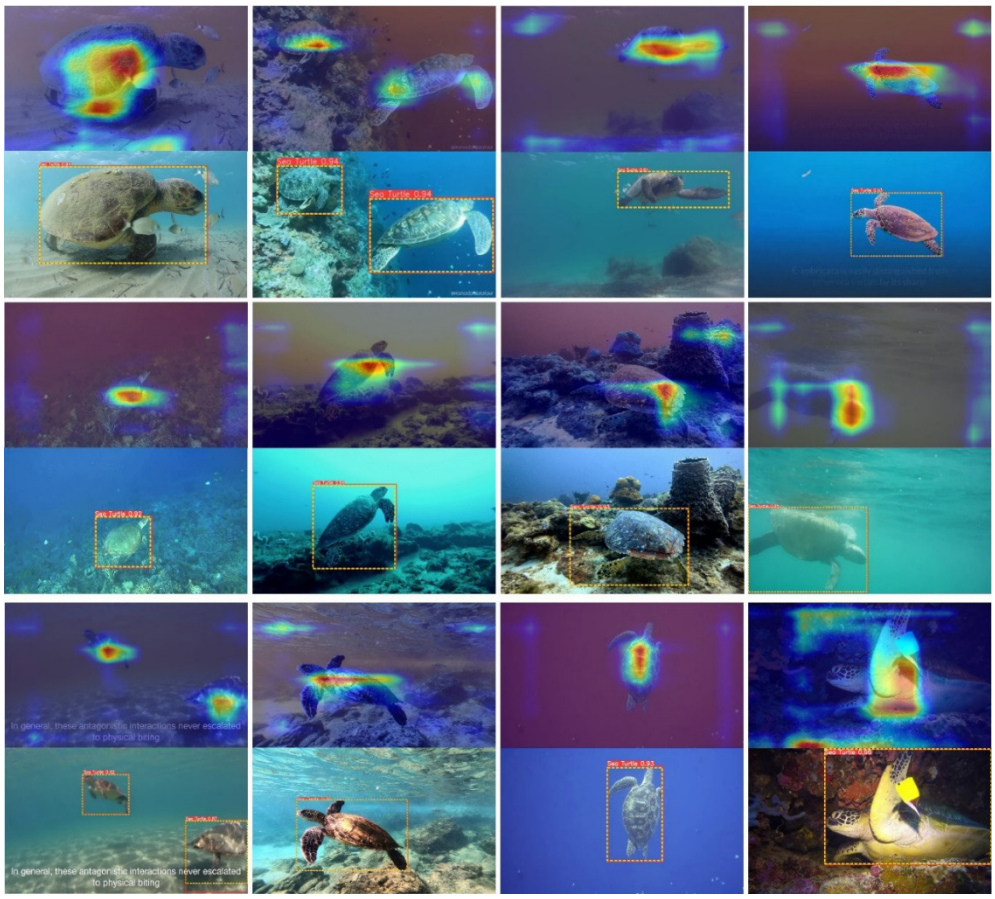

Figure 4. Class activation map and detection results. The red box is the prediction results, and the yellow dashed box indicates the ground truth. Best viewed in colour

Table 4. Comparison runtime efficiency of the proposed model and other networks. Testing speed was conducted on Jetson Nano.

Model	Parameters	GFLOPs	mAP@50:95	FPS on Jetson Nano
YOLOv5n	2,508,643	7,2	0.520	13,24
YOLOv8s	11,135,987	28.6	0.587	11,07
YOLOv8n	3,011,043	8.2	0.567	13,52
YOLOv8p	1,870,179	7.2	0.563	15,73
YOLOv8p-SE block	1,983,187	7,2	0.520	15,73
YOLOv8p-CBAM block	1,997,749	7,2	0.559	15,38
YOLOv8p-Transformer block	1,940,371	7,2	0.585	15,37
YOLOv8p-EGA block	2,123,125	7.3	0.616	14,91

suitable for underwater robot vision systems. This system can play a role in monitoring turtle behaviour for the needs of underwater environmental conservation analysis.

VI. Conclusion

Turtle habitat conservation encourages an underwater robot to be able to visually observe the behaviour of these animals, prompting an automatic vision system that can detect these objects. A novel sea turtle detection is built that implements streamlining deep convolutional layer operations. A lightweight backbone is implemented to extract object features quickly with high performance. In addition, the EGA module improves detector performance through the ability to capture turtle-specific elements. It divides the feature map into four patches with different receptive regions and can increase the variety of target features. As a result, the proposed model achieves a mAP of 61.6%, which outperforms the YOLOv5n, YOLOv8s and YOLOv8n architectures. Additionally, our model achieves higher efficiency than its competitors, which can operate at a real-time speed of 15 FPS on low-end devices.

REFERENCES

[1] H. Y. Lin, S. L. Tseng, and J. Y. Li, "SUR-Net: A deep network for fish detection and segmentation with limited training data," *IEEE Sens J*, vol. 22, no. 18, pp. 18035–18044, Sep. 2022.

[2] Z. Zhou, J. Liu, and J. Yu, "A survey of underwater multi-robot systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 1, pp. 1–8, Jan. 2022.

[3] F. Han, J. Yao, H. Zhu, and C. Wang, "Marine organism detection and classification from underwater vision based on the deep CNN method," *Math Probl Eng*, vol. 2020, no. 3937580, pp. 1–11, Feb. 2020.

[4] E. M. Ditria, S. Lopez-Marcano, M. Sievers, E. L. Jinks, C. J. Brown, and R. M. Connolly, "Automating the analysis of fish abundance using object detection: optimizing animal ecology with deep learning," *Front Mar Sci*, vol. 7, no. 429, pp. 1–9, Jun. 2020.

[5] H. Leo, F. Arnia, and K. Munadi, "Fine tuning CNN pre-trained model based on thermal imaging for obesity early detection," *Jurnal Rekayasa Elektrika*, vol. 18, no. 1, pp. 1–9, Apr. 2022.

[6] A. Sani and S. Rahmadinni, "Deteksi gestur tangan berbasis pengolahan citra," *Jurnal Rekayasa Elektrika*, vol. 18, no. 2, pp. 115–124, Jun. 2022.

[7] H. Leo, F. Arnia and K. Munadi, "Fine tuning CNN pre-trained model based on thermal imaging for obesity early detection," *Jurnal Rekayasa Elektrika*, vol. 18, no. 1, pp. 53–60, Mar. 2022.

[8] M. Pedersen, J. B. Haurum, R. Gade, and T. B. Moeslund, "Detection of marine animals in a new underwater dataset with varying visibility," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Apr. 2020, pp. 18–26.

[9] K. Ali, M. Moetesum, I. Siddiqi, and N. Mahmood, "Marine object detection using transformers," in *Proc. IEEE 19th International Bhurban Conference on Applied Sciences and Technology*, Aug. 2022, pp. 951–957.

[10] Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Trans Pattern Anal Mach Intell*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.

[11] F. Wang, H. Wang, Z. Qin, and J. Tang, "UAV target detection algorithm based on improved YOLOv8," *IEEE Access*, vol. 11, pp. 116534–116544, Oct. 2023.

[12] A. J. Paul, "The need and status of sea turtle conservation and survey of associated computer vision advances," in *Proc. IEEE 8th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering*, Jan. 2021, pp. 1–8.

[13] P. C. Gray, A. B. Fleishman, D. J. Klein, M. W. McKwon, V. S. Bézy, K. J. Lohmann, and D. W. Johnston, "A convolutional neural network for detecting sea turtles in drone imagery," *Methods Ecology and Evolution*, vol. 10, no. 3, pp. 345–355, Mar. 2019.

[14] J. Lu and J. Wei, "Turtle species identification design based on CNN," *Journal of Physics: Conference Series*, vol. 1345, no. 2, pp. 022067, Nov. 2019.

[15] T. Maki, H. Horimoto, T. Ishihara, and K. Kofuji, "Autonomous tracking of sea turtles based on multibeam imaging sonar: toward robotic observation of marine life," *IFAC-PapersOnLine*, vol. 52, no. 21, pp. 86–90, Dec. 2019.

[16] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. of the European Conference on Computer Vision*, Sept. 2018, pp. 3–19.

[17] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Nov. 2021, pp. 13708–13717.

[18] A. M. Malik. (View Mar. 2021). YOLO annotation tool. GitHub. [Online]. Available: https://github.com/ManzarIMalik/YOLO-Annotation-Tool.

[19] T. Shi, Y. Ding and W. Zhu, "YOLOv5s-2E: Improved YOLOv5s for aerial small target detection," *IEEE Access*, vol. 11, pp. 80479–80490, Aug. 2023.

[20] Y. Liu, H. Chu, L. Song, Z. Zhang, X. Wei, M. Chen, and J. Shen, "An improved tuna-YOLO model based on YOLO v3 for real-time tuna detection considering lightweight deployment," *Journal of Marine Science and Engineering*, vol. 11, no. 3, pp. 542–558, Mar. 2023.

[21] L. Zhao, Q. Yun, F. Yuan, X. Ren, J. Jin, and X. Zhu, "YOLOv7-CHS: An emerging model for underwater object detection," *Journal of Marine Science and Engineering*, vol. 11, no. 10, pp. 1949–1969, Oct. 2023.

[22] R. Shankar and M. Muthulakshmi, "Comparing YOLOv3, YOLOv5 & YOLOv7 architectures for underwater marine creature detection," in *Proc. IEEE International Conference on Computational Intelligence and Knowledge Economy*, May 2023, pp. 25–30.

[23] A. Nazir and M. A. Wani, "You Only Look Once - object detection models: A review," in *Proc. IEEE 10th International Conference on Computing for Sustainable Global Development*, May 2023, pp. 1088–1095.

[24] M. B. Muhammad and M. Yeasin, "Eigen-CAM: Class activation map using principal components," in *Proc. IEEE International Joint Conference on Neural Networks*, July 2020, pp. 1–7.

[25] N. Hnoohom, P. Chotivatunyu, S. Yuenyong, K. Wongpatikaseree, S. Mekruksavanich, and A. Jitpattanakul, "Object identification and localization of visual explanation for weapon detection," in *Proc. Research, Invention, and Innovation Congress: Innovative Electricals and Electronics*, Oct. 2022, pp. 144–147.

[26] M. D. Putro, L. Kurnianggoro, and K. H. Jo, "High performance and efficient real-time face detector on central processing unit based on convolutional neural network," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4449–4457, July 2021.