The Indonesian Journal of Computer Science
www.ijcs.net
Volume 14, Issue 2, April 2025
https://doi.org/10.33022/ijcs.v14i2.4635

A Sunken Litter Detection using Dual Receptive Excitation Module

Tomi Heri Julianus Todingan[1], Imanuel Kutika[2], Vicky Nolant Setyanto Lahimade[3], Alwin M. Sambul[4], Oktavian A. Lantang[5], Muhamad Dwisnanto Putro[6]
tomitodingan026@student.unsrat.ac.id[1], imanuelkutika026@student.unsrat.ac.id[2], vickylahimade026@student.unsrat.ac.id[3], asambul@unsrat.ac.id[4], oktavian_lantang@unsrat.ac.id[5], dwisnantoputro@unsrat.ac.id[6]
[1,2,3,4,5,6] Department of Electrical Engineering, Sam Ratulangi University

Article Information

Received : 10 Jan 2025
Revised : 14 Apr 2025
Accepted : 21 Apr 2025

Keywords

Sunken Litter, Attention Module, YOLOv10, Object Detection, Deep Learning.

Abstract

Sunken litter poses a severe ecological challenge, threatening marine life and global ecosystems. Plastic litter is particularly concerning as it could disrupt the food chain, impacting the biodiversity and ecosystem. Over time, without intervention, this issue poses a severe threat to global food security, economic stability in coastal communities, and overall environmental balance. Addressing this problem requires effective monitoring systems for detection. This study enhances the YOLOv10 architecture with a novel Dual Receptive Excitation (DRE) module to improve sunken litter detection. The DRE module uses a dynamic dual-kernel approach to balance spatial and channel-wise processing in Convolutional Neural Networks, adaptively adjusting the receptive field, and capturing critical patterns across scales. Evaluations on the challenging Trash-ICRA19 dataset, sourced from J-EDI, demonstrate the model's robustness under diverse underwater conditions. The proposed system achieves a mean average precision (mAP) of 47.4% and processes 19.60 frames per second, outperforming other studies.

A. Introduction

Sunken litter management has become a major challenge, especially with the growing amount of litter in the oceans. Around 1.15 to 2.41 million tons of plastic litter enter the oceans through rivers yearly, making them a key source of pollution [1]. Sunken litter poses threats by endangering marine and terrestrial species through entanglement, ingestion, and starvation [2]. Over time these particles can accumulate in the food web, ultimately affecting human health through seafood consumption. They also contribute to the degradation of marine ecosystems, impacting biodiversity and reducing the ocean's ability to regulate the climate. Without intervention this issue poses a severe threat to global food security, economic stability in coastal communities, and overall environmental balance. The measures of detection of sunken litter are critical for protecting marine ecosystems because a small fraction of plastic litter could eventually rival the number of fish [3]. The effort focused on surface-level plastic litter, while advanced technologies to address plastic litter on the ocean floor remain limited. The monitoring system for marine litter can help to detect sunken litter. It could become a preventive measure to reduce ocean litter [4].

The need for monitoring requires a system to detect sunken litter [5]. Object detection technology based on artificial intelligence with deep learning algorithms [6], has proven effective in identifying and counting litter objects in both aquatic and coastal locations in real-time. Real-time detection introduces additional complexities when deployed in natural environments such as oceans and beaches. Problems such as unstable network connections, limited-capability mobile devices, and high energy consumption for data transmission underscore the importance of localized and energy-efficient processing [7]. Research shows that object detection systems can replace some of manual work required in marine surveys, thus speeding up the data collection process and reducing labor costs [8]. Furthermore, using diverse datasets, these systems can recognize various types of litter, though detection accuracy may vary depending on the object type. Nonetheless, issues like changes in lighting, wave interference, and natural litter continue to impact performance in complex marine environments [9].

To overcome these challenges, Convolutional Neural Networks (CNNs) have been essential in extraction features, advancing object detection [10]. Models like VGG [11] and ResNet [12] achieve high accuracy by identifying complex features but require significant computational resources, limiting their use in resource-constrained settings. More efficient CNNs have been developed to achieve high performance on mobile and edge devices, which is crucial for localized litter detection systems [13]. The YOLO series, recognized for its combination of speed and accuracy, has become a popular choice for real-time applications such as robotics, environmental monitoring, and litter detection. Among these, YOLOv10 [14] shows great potential for enhancing sunken litter detection. YOLOv10 is a highly effective model for real-time object detection, particularly suitable for sunken litter detection. Removing non-maximum suppression (NMS) and using a dual assignment strategy, improves speed, accuracy, and efficiency. Features such as decoupled downsampling, rank-guided block design, large-kernel convolution, and partial self-attention enhance its performance with minimal additional cost.

Additionally, integrating attention mechanisms can improve performance by allowing the model to focus on the most critical features [15].

Attention mechanisms [16] are vital in modern deep learning, dynamically highlighting important information while suppressing less useful details. In object detection, they enhance accuracy by improving the model's ability to prioritize relevant features across spatial and channel dimensions. This capability is particularly beneficial in underwater environments, where variations in lighting, waves, and object sizes can hinder detection. One notable implementation of this approach is the Selective Kernel Convolution [17] (SKConv), which adapts the receptive field sizes of neurons through a dynamic process. SKConv achieves this using three steps: splitting features into multiple scales, fusing them to form a comprehensive representation, and selecting the most relevant features using soft attention. In this work, we propose a novel sunken litter detection system utilizing an improved YOLOv10-nano model with an additional enhanced receptive block. This block enhances detection precision by distinguishing essential features from irrelevant information. It discriminates features using two distinct spatial areas, allowing for a comparative analysis of spatial information. Furthermore, this module effectively boosts feature extraction performance. The contributions of this work can be summarized as follows:

1. This work introduces litter detection in underwater environments using deep learning models, addressing challenges unique to this domain.
2. Performance improvements are achieved by integrating a selective kernel convolutional module into the primary feature extractor and transition block. This module enables the network to robustly capture essential features while suppressing less relevant information.
3. A comprehensive mean average precision (mAP) evaluation was conducted using the Trash-ICRA19 dataset. The results demonstrate improved precision over the original YOLOv10-nano model and superior performance compared to previous approaches.

B. Related Works

The YOLO algorithm, introduced by Redmon et al., [18] as one of the foundations for real-time object detection by directly regressing target boxes. YOLOv5 was later developed using Pytorch which is more accessible and flexible for the research community than the previous version using Darknet. YOLOv5 improved efficiency with the addition of the C3 lightweight feature extraction framework using variants—S, M, L, and X—to adapt with different hardware and performance needs [19]. However, the use of anchor-based prediction methods creates redundant boxes, making them not suitable for lightweight computing. YOLOv8 launched in 2023 to incorporate the C2f structure to improve gradient flow and adopted an anchor-free approach, reducing computational time and resources [20]. Despite these improvements, post-processing steps were still required. To overcome this, Wang et al. [14] introduced YOLOv10, which eliminates the need for post-processing by using a dual assignment strategy, achieving faster detection speeds. YOLOv10 has been applied in various detection tasks. For example, a YOLOv10-based algorithm, combined with FasterNet, was used to detect dead fish, a significant source of ocean pollution, while reducing model complexity [21].

Underwater object detection using UM-YOLOv10 that used residual attention module R-AM [22]. Another application of YOLOv10 focuses on small object detection using drones, where a modified YOLOv10 model, LD-YOLOv10, integrates RGLAN. This feature uses re-parameterized convolutions and the Conv-Tiny structure for efficiency [23]. Another modification using BGF-YOLOv10 by incorporating BotNet and GhostConv to enhance detection performance for small objects [24].

Study by [25] explores underwater image processing and object detection using Convolutional Neural Networks (CNNs) to improve detection accuracy in degraded underwater images for robots. Similarly, A work [26] presents a two-stage framework combining object detection and image quality restoration for unmanned underwater vehicles (UUVs). This method addresses challenges such as power constraints and visual distortions caused by underwater light conditions. Additionally [27], enhanced the Faster R-CNN algorithm for underwater object detection, including species like holothurians and starfish, by replacing the network backbone with Res2Net101, improving the receptive field's expressive capability.

The demand for efficient plastic litter detection in waterways has also driven advances in computer vision [9]. Niu et al [28]. improved sunken litter detection models by integrating attention mechanisms and architectural changes, resulting in better performance than previous approaches. Harada [29] et al. focused on lightweight models for real-time edge-device use, incorporating GhostBlockNeck to reduce computational overhead without sacrificing accuracy in sunken litter detection. Zhu [9] et al. improved sunken litter detection accuracy by introducing the C2f-Faster module and Efficient Multiscale Attention, achieving a 5% improvement in mean average precision on the TRASH-ICRA19 dataset. Fulton [18] et al. evaluated various convolutional neural network models using the TRASH-ICRA19 dataset, providing important benchmarks for sunken litter detection. These studies highlight the effectiveness of model improvements and dataset utilization in boosting detection accuracy and efficiency, laying the groundwork for future advancements.

C. Research Method

Sunken Litter detection is a critical technology in oceanic exploration. However, the challenges posed by the complex underwater environment and the presence of numerous small targets often hinder the effectiveness of conventional detection systems. These systems frequently fail to meet desired performance standards and are often too large in model size, making them unsuitable for deployment on ROVs with limited memory capacity. To overcome these issues, we have designed and enhanced a real-time underwater target detection model based on YOLOv10, which outperforms existing technologies in terms of both detection speed and accuracy. The proposed model incorporates backbone and neck layers specifically optimized for underwater conditions, alongside C2f modules tailored for improved performance. This modified YOLOv10 algorithm excels in detecting small underwater objects, achieving exceptional accuracy while meeting the real-time detection requirements. Additionally, the optimized model incorporates a dynamic selection mechanism in its CNN architecture, enabling each neuron to adaptively adjust its receptive field size based on multiple scales of input information. This

innovation enhances the model's ability to capture diverse feature representations, improving its compatibility with lightweight detection systems and underwater wireless sensor platforms. In this work, we will discuss the following components of the proposed method in detail:

Research methods can be supplemented by tables, graphs (pictures), and/or charts. The table does not contain vertical (upright) lines. Horizontal (flat) lines in the table are only found at the beginning and end of the table. Example of table format:

1. Backbone

Backbone is fundamental in distinguishing meaningful features from irrelevant ones, forming the core of object detection networks.

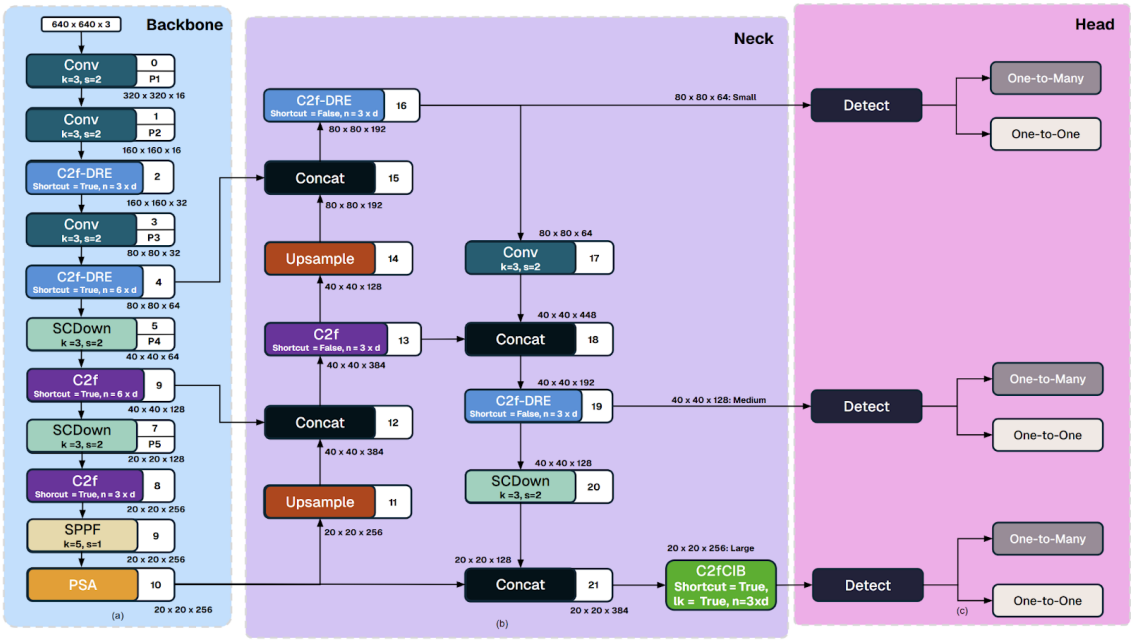

Figure 1. Proposed Architecture Module Attention DRE as an Improved Model of YOLOv10 : Backbone of YOLOv10 by replacing C2f to C2f-DRE on P2 and P3(a), Neck of YOLOv10 by replacing C2f to C2f DRE on P16 and P19(b), Head on YOLOv10(c).

This process relies on convolutional layers, which efficiently extract and refine essential patterns while discarding noise. By leveraging multi-kernel weighting, these layers effectively focus on target information. During training, the network iteratively updates the weights of its filters to ensure precise and reliable predictions. Our proposed architecture was an improved model from YOLOv10 architecture, and the feature extraction layers, referred to as the backbone in Figure 1(a), were designed to hierarchically process input data to generate low-level, mid-level, and high-level features. These hierarchical features are critical to the detection process, as they enable the model to identify objects of varying sizes and complexities. The backbone's architecture is tailored to support the three detection heads, which operate at different scales. This setup ensures that small objects are detected in fine-grained low-level features, while larger and more complex objects

are captured using high-level contextual features. By organizing the backbone in this manner, our proposed architecture achieves robust multi-scale detection with high accuracy and efficiency.

2. C2f

The C2f block in the YOLOv10 architecture is designed to extract rich feature representations from input images while maintaining computational efficiency, contributing to a balance between speed and accuracy. However, a key limitation of the original C2f block in Figure 2(a) is its relatively straightforward structure, which lacks additional mechanisms to enhance feature refinement and focus on important regions. Specifically, after the concatenation step, the output directly passes through the final convolutional layer without further enhancement, which may restrict its ability to capture fine-grained details and complex patterns in challenging tasks.

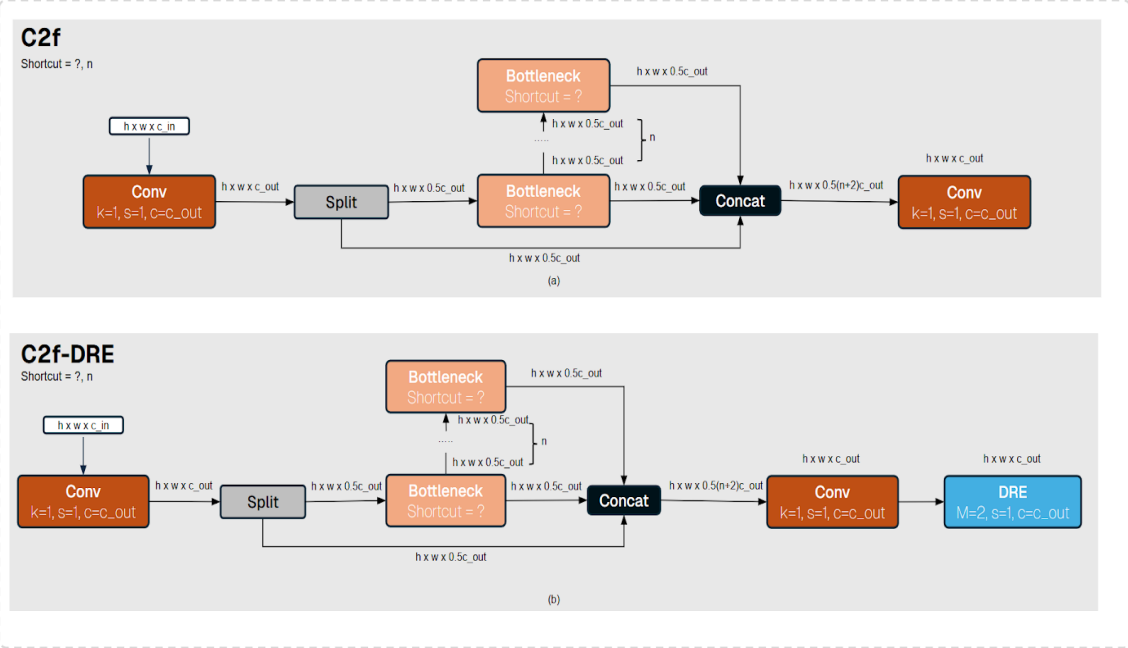

Figure 2. C2f Block in YOLOv10(a), Modified C2f Block using module DRE after the process of C2f to enhance performance(b).

To address this weakness, the proposed C2f-DRE block in Figure 2(b) introduces a more advanced structure by integrating an additional convolutional layer and the Dual Receptive Excitation (DRE) module immediately after the concatenation step.

The DRE module is strategically designed to amplify feature representation by emphasizing critical regions and incorporating multi-scale information, enabling the model to capture more nuanced details and handle complex scenarios more effectively.

Furthermore, while the original C2f relies on two simple paths—one through the Bottleneck block and another direct bypass—for basic feature extraction, the C2f-DRE expands these capabilities with the DRE module, offering a more sophisticated and comprehensive approach to feature extraction. This improvement

leads to enhanced performance and better detection accuracy, particularly in scenarios requiring deeper and more complex feature representations.

3. SPPF

The Spatial Pyramid Pooling Fast (SPPF) block is an advanced technique utilized in the YOLOv10 model, representing an improvement over the traditional Spatial Pyramid Pooling (SPP) method. It is designed to address challenges related to scaling and computational efficiency when handling objects of varying sizes within an image.

The process begins with a 1×1 convolution, followed by the concatenation of three stacks of 2D max pooling operations with kernel sizes of 5, 9, and 13. The pyramid-shaped arrangement of the kernel sizes in the max pooling stacks aims to achieve a broader receptive field for both local and global features without compromising speed. This versatility enables the SPPF block to effectively adapt to objects of different sizes across various scenarios.

4. PSA

The Partial Self-Attention (PSA) module in YOLOv10 is introduced to enhance the model's global representation learning ability while maintaining computational efficiency. PSA is an optimized self-attention mechanism tailored for CNN-based object detection tasks. In YOLOv10, PSA represents a significant enhancement to the model's architecture by incorporating global modeling capabilities without imposing heavy computational demands. By strategically designing and placing PSA, YOLOv10 achieves better accuracy-efficiency trade-offs compared to earlier versions.

The PSA in YOLOv10 is designed to improve the model's ability to learn global representations, which has been a limitation in CNN-based architectures. This PSA mechanism allows the model to capture global relationships between features in an image without adding excessive computational costs. With this approach, YOLOv10 enhances object detection capabilities, especially in handling complex contexts or objects of varying sizes. The strategic placement of PSA within the architecture ensures the model can effectively utilize attention mechanisms, achieving an optimal balance between accuracy and efficiency compared to previous versions.

5. Proposed Module

In Figure 3, The proposed module was named Dual Receptive Excitation Module (DRE Block) as it introduces a dynamic dual-kernel approach to enhance feature extraction in convolutional neural networks (CNNs). This module leverages the complementary properties of 1×1 and 3×3 kernels, effectively balancing spatial and channel-wise information processing. By adopting a dynamic selection mechanism, the module allows neurons to adaptively adjust their receptive field size based on input features, ensuring that critical patterns across varying scales are captured effectively.

The combination of these kernels allows the proposed module to process multi-scale information efficiently while maintaining a low computational footprint. The dynamic selection mechanism ensures that the module prioritizes features most relevant to the task, enhancing both accuracy and efficiency. This module is seamlessly integrated into the network's architecture, contributing to improved detection performance, particularly in scenarios requiring the recognition of small and intricate objects.

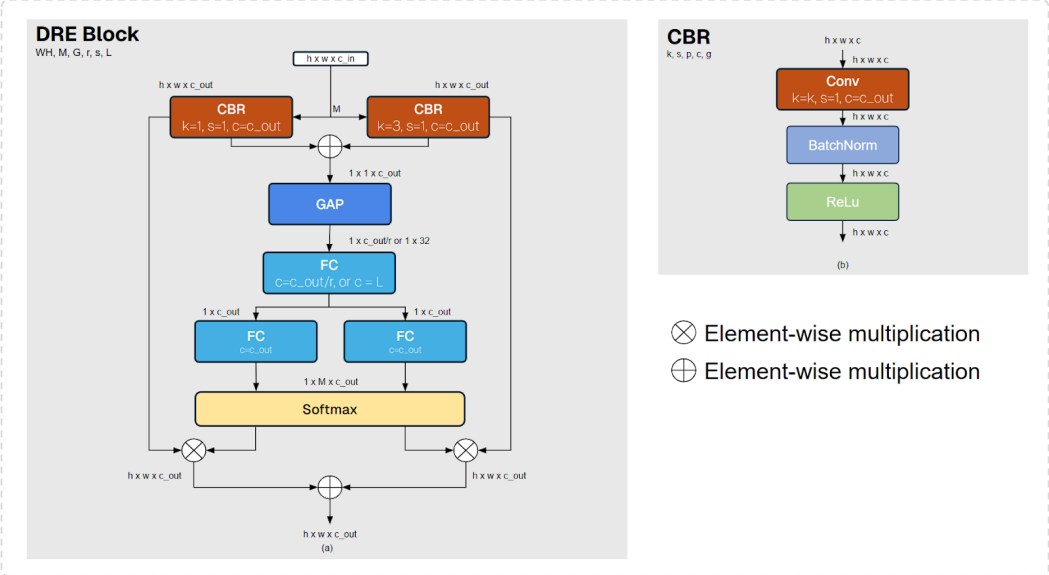

Figure 3. DRE Block for balancing spatial and channel wise processing(a), Convolution-BatchNorm-ReLu(CBR) Block to extract and normalize features while ensuring robustness with non linear activation(b).

In Equation 1, the input feature map X is processed through two separate convolution layers with kernels of receptive field size 1×1 and 3×3, respectively. Each convolution operation is followed by the application of an activation function σ, specifically the ReLU function, which introduces non-linearity to the model. The output of the activation function is then normalized using Batch Normalization (BN) to stabilize training, improve generalization, and ensure consistent feature scaling across different layers. σ

$$R_1 = BN(\sigma(Conv_{1\times1}(X)),\ \ R_2 = BN(\sigma(Conv_{3\times3}(X)). \tag{1}$$

Equation 2 computes the intermediate feature representation R_T by combining the outputs R_1 and R_2 using element-wise addition $\oplus$, followed by a Global Average Pooling (GAP) operation to aggregate spatial information into a channel-wise descriptor. The pooled features are then scaled using a learnable weight matrix $W_{c\_in/r}$ where W_{c_in} is Fully Connected operation with the channel same as the input, and r is a reduction ratio by 2 that controls the dimensionality of the channel, reducing computational complexity and emphasizing channel-wise dependencies.

$$R_T = W_{\frac{c_{in}}{r}}\big(GAP(R_1 \oplus R_2)\big). \tag{2}$$

In equation 3, the compressed feature descriptor R_T is transformed into two separate sets of weights R_{W1} and R_{W2} through learnable weight matrices W. These weights are used to modulate the importance of the respective feature maps R_1 and R_2, enabling the model to adaptively select and emphasize features across different kernel sizes.

$$R_{W1} = W_{c\_in}(R_T),\ R_{W2} = W_{c\_in}(R_T). \quad (3)$$

Equation 4 applies attention weights to the feature maps R_{S1} and R_{S2}. The weights R_{W1} and R_{W2} are normalized by their sum to generate soft attention masks. These masks are then applied to R_1 and R_2 via element-wise multiplication $\otimes$, dynamically scaling the feature maps based on their relevance to the input.

$$R_{S1} = \frac{e^{R_{W1}}}{e^{R_{W1}} + e^{R_{W2}}} \otimes R_1,\ R_{S2} = \frac{e^{R_{W2}}}{e^{R_{W1}} + e^{R_{W2}}} \otimes R_2. \quad (4)$$

Finally in equation 5, the scaled feature maps R_{S1} and R_{S2} are fused using element-wise addition $\oplus$ to produce the final output feature map DRE. This combination enables the model to leverage complementary information captured by different kernel sizes, enhancing its ability to represent multi-scale features effectively.

$$DRE = R_{S1} \oplus R_{S2}. \quad (5)$$

By incorporating the design principles of our proposed module and refining its implementation with a focus on efficient kernel utilization, the proposed module provides a robust solution for adaptive feature extraction, aligning with the demands of real-time object detection applications, more importantly in sunken litter detection.

6. Neck

The neck part of the deep learning architecture shown in Figure 1(b) is an important component that processes and combines features from different levels of the backbone. In this diagram, the neck uses several mechanisms like the Conv layer, C2f layer, C2f-DRE layer, C2f-CIB layer, Upsample, Concat, and SCDown to organize information from high to low resolution. The purpose of this structure is to align spatial and contextual information, resulting in rich features for the detection process in the head model. The main advantage of the neck design in Figure 1(b) is its flexibility in efficiently combining multi-resolution features. With different pathways that blend features from small to large scales, the model can better understand objects of various sizes. This is especially important in tasks like object detection, where objects can appear at different scales within the same image. The combination of these elements creates highly optimized feature representations for use by the head in the final detection process.

7. Detection Layer and Loss

The head part shown in Figure 1(c) consists of a classification head that identifies the class of each object while estimating the probability for each class with a total of the sum of probabilities is one and a regression head that predicts the bounding box coordinates for detected objects including the center coordinates, width and height while providing a confidence score. The heads are using Dual label assignment that is composed of One-To-Many Head that retains the original structure and optimization objective of the model to make several predictions and One-to-One Head that uses a matching strategy for label assignment, ensuring each ground truth is matched with a single prediction. Both heads were used

simultaneously during inference, allowing the backbone and neck of the model to leverage the comprehensive supervision from the one-to-many assignments that improved model's learning and accuracy.

This model utilizes CIoU loss, which enhances bounding box accuracy compared to the standard IoU. The key factors in CIoU include the area overlap between ground truth boxes, the Euclidean distance between the center points of the predicted and ground truth boxes, and the aspect ratio, which measures the similarity between the height-to-width ratios of the predicted and ground truth boxes, thereby improving the alignment and reducing the loss. Additionally, the model employs classification loss to evaluate errors in classifying objects within the predicted bounding boxes, and distributive focal loss to handle small or hard-to-classify objects by predicting a distribution of confidence scores.

D. Implementation Setup

To evaluate the proposed method, we prepared the implementation setup and dataset to achieve optimal performance while balancing computational efficiency and accuracy of the model. The experiments were conducted using a high-performance computing environment to ensure reproducibility and scalability. Specific configurations and datasets were carefully selected to align with the research objectives and optimize the training process. Details of the implementation will be explained in the section below.

1. Training and Testing Configuration

As shown in Table 1 the training phase of the proposed research was implemented using the Kaggle platform, which provides an accessible and efficient environment for deep learning experiments. The training was performed on Kaggle's GPU, specifically the G100 model, which offers substantial computational power for handling complex models. The input images were resized to 640×640 pixels to maintain a balance between computational efficiency and retaining important spatial features.

Table 1. Training and Testing Configuration

Parameters	***Setup***
Platform/device	Kaggle
GPU	P100
Image Size	640 x 640 pixels
Epochs	300
Batch Size	32
Optimizer	Stochastic Gradient Descent (SGD)
Learning Rate	0,01

The training process spanned 300 epochs, ensuring that the model was provided with sufficient iterations to converge. A batch size of 16 was chosen, balancing memory constraints with training stability. The Stochastic Gradient Descent (SGD) optimizer was employed due to its robustness and efficiency in handling large datasets. The learning rate was initialized at 0.01 to achieve a steady convergence rate without overshooting the minima.

Table 2. Inference Configuration

Parameters	*Setup*
Operation System	Ubuntu
Compiler	Python 3.9.20
Network construction method	Pytorch 2.0
CPU	AMD Ryzen 5 4500 6-Core
Image Size	640 x 640 pixels

For the Inference phase, the model inference and evaluation were performed on a local machine using a CPU setup. This approach allows for testing the model's deployment capabilities in environments with limited computational resources.

2. Datasets

The dataset utilized for this research is the Trash-ICRA19 dataset, sourced from the J-EDI marine litter dataset, as described by Fulton et al. [30] This dataset provides a diverse set of 5,720 training data images, 820 validation data images, and 1145 testing data images as shown in Table 3 extracted from real-world underwater video footage, which varies significantly in quality, depth, and lighting conditions. Each image is annotated with bounding boxes to label instances of litter, biological objects (e.g., plants and animals), and remotely operated vehicles (ROVs).

Table 3. Dataset Configuration

Parameters	*Setup*
Training Data	5.720 images
Validation Data	820 images
Testing Data	1145 images

The dataset reflects challenging real-world conditions, such as varying states of decay, occlusion, and overgrowth of objects, as well as changes in water clarity and light quality between videos. This variability makes the dataset particularly suitable for developing robust detection models. The ultimate goal of this dataset is to facilitate research in autonomous litter detection and removal, a critical step toward addressing sunken litter issues.

Figure 4. Sample Sunken Litter from dataset Trash ICRA-19 that shows the litters under diverse underwater conditions.

The variety and configuration of the images make the dataset ideal for evaluating the performance of adaptive methods such as the proposed model. A sample of the dataset is shown in Figure 4.

E. Result and Discussion

This section provides a detailed evaluation of the proposed underwater object detection model. The analysis is divided into three key aspects: the Ablation Study, which investigates the impact of individual components of the model architecture on its performance; the Evaluation on Datasets, which demonstrates the model's detection accuracy and reliability under real-world underwater conditions; and the Runtime Efficiency, which assesses the computational speed and suitability of the model for real-time applications. These evaluations collectively highlight the effectiveness and practicality of the proposed model in addressing challenges in underwater object detection.

1. Ablation Study

An ablation study is a crucial step in evaluating the effectiveness of specific improvements in a model. By analyzing the impact of individual components or modifications, researchers can confirm which changes lead to better performance. This ensures that the proposed model's enhancements are meaningful and not coincidental.

Table 4. Ablation Table with related works using dataset Trash-ICRA19

Model	*GFLOPS*	*Parameter*	*mAP 50%*
YOLOv10n	8.4	2,7	46.3
YOLOv10 C2f-DRE (Ours)	11.2	3,1	**47.4**

In Table 4, a comparison between the YOLOv10n and the proposed YOLOv10 C2f-DRE model is provided, showcasing the effectiveness of the latter in terms of computational efficiency and detection accuracy. The metrics used include GFLOPS (measuring computational complexity), the number of parameters (indicating model size), and mAP@50% (mean average precision at 50% Intersection over Union). These metrics highlight the trade-off between computational efficiency and accuracy.

From the table, it can be observed that the proposed YOLOv10 C2f_DRE model achieves better detection accuracy, with an mAP@50% of 47.4%, compared to 46.3% for the YOLOv10n model. While the proposed model requires slightly more computational power and has a higher parameter count, the improvement in accuracy demonstrates the value of the modifications.

Table 5. Comparison Table with related works using dataset Trash-ICRA19

Model	*GFLOPS*	*Parameter*	*mAP 50%*
YOLOv5-ghost [8]	1.5	1,5	46.6
YOLOv8[9]	8.1	3,01	45.5
YOLOv8–C2f Faster EMA [9]	6.5	-	47.2
YOLOv10 C2f-DRE (Ours)	11.2	3,1	**47.4**

As for Table 5, it presents the performance of the proposed YOLOv10 C2f-DRE model compared to other models using the Trash-ICRA19 dataset. The YOLOv5-ghost model achieves an mAP@50% of 46.6%, showing solid performance among the earlier models. YOLOv8, however, records a slightly lower mAP@50% of 45.5%. The YOLOv8-C2f Faster EMA improves detection accuracy further, achieving an mAP@50% of 47.2%.

On the other hand, our proposed model achieves an accuracy of 47.4 mAP at 50%, which is higher than that of previous models at best with YOLOv8-C2f Faster EMA[9] with gap 0.2 at mAP 50% with 4,7 GFLOPS increased. This result highlights the model's effectiveness in accurately detecting sunken litter in complex environments, outperforming all other models.

2. Evaluation on Datasets

The performance evaluation of using the proposed model on the Testing Dataset, as illustrated in Figure 5, demonstrates notable improvements in detection precision and reliability compared to the original YOLOv10 model. In (a), the original YOLOv10 encounters challenges in accurately detecting plastic objects, especially small or partially obscured items. Some objects are either missed entirely or not clearly localized, which compromises the model's ability to perform effectively in underwater environments characterized by cluttered backgrounds and varying object scales.

In contrast, (b) highlights the superior detection precision of the proposed model. It consistently identifies objects with clearer and more accurate bounding boxes, ensuring better localization of small and partially visible plastic litter. The proposed model also shows greater resilience in differentiating objects from complex underwater backgrounds, reducing missed detections and improving the overall detection quality.

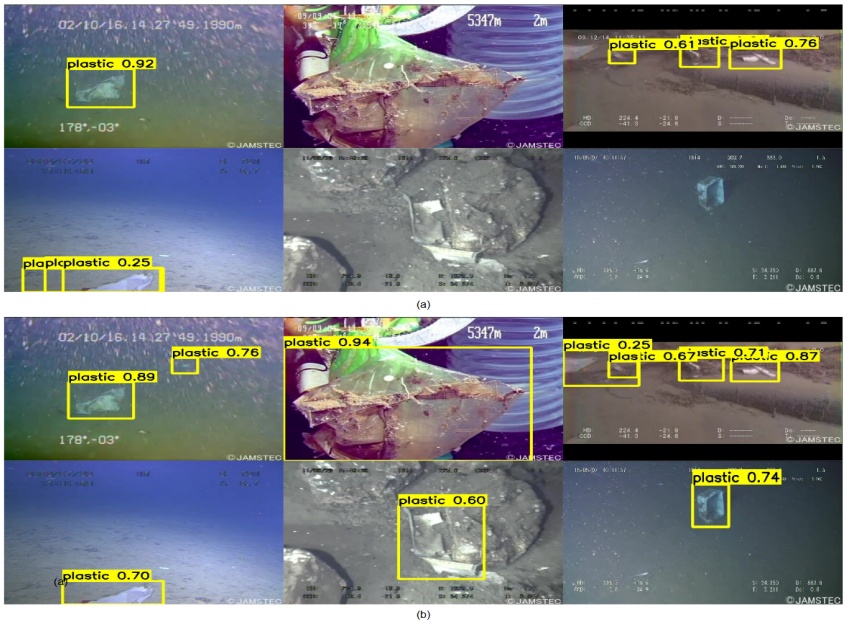

Figure 5. Comparison of sunken litter detection models : YOLOv10 original detection shows lack of detection in several objects(a), Proposed Model detection shows capabilities to detect various objects of sunken litters(b).

These observations underscore the capability of the proposed model to detect sunken litter with greater precision, particularly in scenarios involving small targets and challenging environmental conditions. This improvement enhances its suitability for real-time underwater monitoring and environmental conservation tasks.

3. Runtime Efficiency

Runtime efficiency is a critical metric in evaluating the practical usability of object detection models, particularly for applications requiring real-time performance on resource-limited devices, such as edge computing or mobile platforms. It primarily measures how effectively a model balances computational demands with inference speed, typically represented by metrics such as latency and frames per second (FPS). Table 6 provides a comparative analysis of the runtime efficiency of YOLOv10n and the proposed YOLOv10 C2f-DRE (Ours) models, evaluated using a CPU.

Table 6. Speed Table by CPU

Model	*GFLOPS*	*Latency/ms*	*FPS*
YOLOv10n	8.4	66.2	15.22
YOLOv10 C2f-DRE (Ours)	11.2	51.4	**19.60**

As observed in Table 6, the YOLOv10 C2f-DRE demonstrates significant improvements in runtime efficiency compared to YOLOv10n. Despite a slight increase in computational complexity, as reflected by a rise in GFLOPS from 8.4 to 11.2, the proposed model achieves a remarkable reduction in latency, decreasing from 66.2 ms to 51.4 ms. This improvement translates into a higher inference speed, with the FPS increasing from 15.22 to 19.60. These results indicate that the

enhancements introduced in YOLOv10 C2f-DRE, such as the integration of the DRE module, not only improve detection accuracy but also optimize processing speed, making it better suited for real-time applications while maintaining computational efficiency.

F. Conclusion

In this study, we proposed an advanced sunken litter detection system based on the YOLOv10 architecture, enhanced with the C2f-DRE module and selective kernel convolution. The system demonstrated significant improvements in detection accuracy and efficiency for underwater environments, as evaluated on the TRASH-ICRA19 dataset. The integration of dynamic receptive fields and optimized feature extraction techniques allowed the model to effectively handle challenges such as occlusion, variable lighting, and complex backgrounds. The proposed model achieved a notable mean average precision (mAP) of **47.4%**, outperforming prior models such as YOLOv8-C2f Faster EMA by a margin of 0.2%, and demonstrated exceptional runtime efficiency with a frame processing rate of **19.60 FPS**, significantly higher than YOLOv10n's 15.22 FPS. Our results indicate that the proposed model achieves superior performance compared to existing methods, with improvements in mean average precision (mAP) and processing speed, validating its suitability for real-time applications. These advancements contribute to the broader goal of environmental monitoring and marine conservation by providing a robust tool for detecting sunken litter. Future work could focus on integrating the system with autonomous underwater vehicles for litter removal and extending its capabilities to identify additional types of marine pollutants. This research highlights the potential of deep learning in addressing critical ecological challenges and underscores the importance of continued innovation in underwater detection technologies.

G. Acknowledgment

Special thanks to all members of the AIVision research group for their contributions in making this work possible.

H. References

[1] L. C. M. Lebreton, J. van der Zwet, J.-W. Damsteeg, B. Slat, A. Andrady, and J. Reisser, "River plastic emissions to the world's oceans," *Nat. Commun.*, vol. 8, p. 15611, 2017.

[2] D. Parasar, S. R. Vadalia, S. S. Chavan, K. R. Bhere, F. Nabi, and A. Z. Patel, "Waste detection and water quality assessment in aquatic environments: A comprehensive study using YoloV8 and XGBoost," 2024.

[3] N. Maximenko, A. P. Palacz, L. Biermann, J. Carlton, L. Centurioni, M. Crowley, and C. Zabin, "An integrated observing system for monitoring marine debris and biodiversity," *Oceanography*, vol. 34, no. 4, pp. 52-59, 2021.

[4] A. Redmond, "Real-time detection of marine debris using YOLOv3," *Journal of Marine Science and Engineering*, vol. 8, no. 5, pp. 345-356, 2020.

[5] K. M. Raju, S. Banuri, H. S. Abdussami, S. Kowdi, M. S. Mashkour, Manjunatha, N. Singh, and A. Kumar, "IoT-based smart garbage monitoring system and

advanced disciplinary approach," E3S Web of Conferences, vol. 507, no. 01031, pp. 1–7, 2024. doi: 10.1051/e3sconf/202450701031.
[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *arXiv preprint arXiv:1506.02640v5 [cs.CV]*, May 9, 2016. [Online].
[7] S. Rajput and T. Sharma, "Benchmarking Emerging Deep Learning Quantization Methods for Energy Efficiency," *2024 IEEE 21st International Conference on Software Architecture Companion (ICSA-C)*, Hyderabad, India, 2024, pp. 238-242, doi: 10.1109/ICSA-C63560.2024.00049.
[8] R. Johnson and L. Wang, "Application of deep learning for marine debris detection," *Environmental Monitoring and Assessment*, vol. 192, no. 4, pp. 245-258, 2020.
[9] J. Zhu, T. Hu, L. Zheng, N. Zhou, H. Ge, and Z. Hong, "YOLOv8-C2f-Faster-EMA: An improved underwater trash detection model based on YOLOv8," Sensors, vol. 24, no. 8, pp. 2483, Apr. 2024. doi: 10.3390/s24082483
[10] M. Córdova, A. Pinto, C. C. Hellevik, S. A.-A. Alaliyat, I. A. Hameed, H. Pedrini, and R. da S. Torres, "Litter Detection with Deep Learning: A Comparative Study," *Sensors*, vol. 22, no. 2, p. 548, Jan. 2022.
[11] Y. Zhou, H. Chang, Y. Lu, X. Lu and R. Zhou, "Improving the Performance of VGG Through Different Granularity Feature Combinations," in *IEEE Access*, vol. 9, pp. 26208-26220, 2021, doi: 10.1109/ACCESS.2020.3031908.
[12] C. S. Wickramasinghe, D. L. Marino and M. Manic, "ResNet Autoencoders for Unsupervised Feature Learning From High-Dimensional Data: Deep Models Resistant to Performance Degradation," in *IEEE Access*, vol. 9, pp. 40511-40520, 2021, doi: 10.1109/ACCESS.2021.3064819.
[13] S.-A. Bergies, P. T.-T. Nguyen, and C.-H. Kuo, "Cleaning Robot Vision System Based on RGBD Camera and Deep Learning YOLO-based Object Detection Algorithm," *International Journal of iRobotics*, vol. 4, no. 4, pp. 23-29, Dec. 2021.
[14] Wang, A.; Chen, H.; Liu, L.; Chen, K.; Lin, Z.; Han, J.; Ding, G. YOLOv10: Real-Time End-to-End Object Detection. arXiv 2024, arXiv:2405.14458.
[15] T. Shi, W. Zhu and Y. Su, "Improved Light-Weight Target Detection Method Based on YOLOv5," in *IEEE Access*, vol. 11, pp. 38604-38613, 2023, doi: 10.1109/ACCESS.2023.3267965.
[16] G. Brauwers and F. Frasincar, "A General Survey on Attention Mechanisms in Deep Learning," in *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 4, pp. 3279-3298, 1 April 2023, doi: 10.1109/TKDE.2021.3126456.
[17] X. Li, W. Wang, X. Hu, and J. Yang, "Selective Kernel Networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 510–519.
[18] Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the Computer Vision & Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016.
[19] Jocher, G.; Stoken, A.; Borovec, J.; Chaurasia, A.; Changyu, L.; Hogan, A.; Hajek, J.; Diaconu, L.; Kwon, Y.; Defretin, Y. Ultralytics/Yolov5: V5. 0-YOLOv5-P6 1280 Models, AWS, Supervise. Ly and YouTube Integrations. Zenodo 2021.
[20] Jocher, G. Ultralytics YOLOv8: V6. Available online: https://Github.Com/Ultralytics/Ultralytics (accessed on 23 October 2023).

[21] Q. Tian, Y. Huo, M. Yao, and H. Wang, "A method for detecting dead fish on large water surfaces based on improved YOLOv10," *arXiv preprint arXiv:2409.00388*, 2024.
[22] J. Wang and R. Mai, "Um-Yolov10: An Underwater Object Detection Algorithm for Marine Environment Based on Yolov10 Model," *SSRN*.
[23] X. Qiu, Y. Chen, W. Cai, M. Niu, and J. Li, "LD-YOLOv10: A lightweight target detection algorithm for drone scenarios based on YOLOv10," *Electronics*, vol. 13, no. 16, p. 3269, 2024.
[24] Mei and W. Zhu, "BGF-YOLOv10: Small Object Detection Algorithm from Unmanned Aerial Vehicle Perspective Based on Improved YOLOv10," *Sensors*, vol. 24, no. 21, pp. 6911, 2024.
[25] Underwater Object Detection Method Based on Improved Faster RCNN. Appl. Sci. 2023, 13, 2746. https:// doi.org/10.3390/app13042746 Academic Editor: Sungho Kim Received: 24 January 2023 Revised: 11 February 2023 Accepted: 15 February 2023 Published: 20 February 2023.
[26] Sheezan Fayaz, S. A. Parah, G. J. Qureshi, J. Lloret, J. Del Ser, and K. Muhammad, "Intelligent Underwater Object Detection and Image Restoration for Autonomous Underwater Vehicles," *IEEE Trans. Veh. Technol.*, vol. 73, no. 2, pp. Feb. 2024.
[27] H. Wang and N. Xiao, "Underwater Object Detection Method Based on Improved Faster RCNN," *Applied Sciences*, vol. 13, no. 4, p. 2746, Feb. 2023.
[28] Jinxing Niu, Shaokui Gu, Junmin Du, Yongxing Hao, "Underwater Waste Recognition and Localization Based on Improved YOLOv5," Tech Science Press, 2023. Available:TSP_CMC_40489.pdf (techscience.cn).
[29] R. Harada, T. Oyama, K. Fujimoto, T. Shimizu, M. Ozawa, J. S. Amar, and M. Sakai, "Trash detection algorithm suitable for mobile robots using improved YOLO," J. Adv. Comput. Intell. Intell. Inform., vol. 27, no. 4, pp. 622-631, 2023, doi: 10.20965/jaciii.2023.p0622.
[30] M. Fulton, J. Hong, M. J. Islam, and J. Sattar, "Robotic detection of marine litter using deep visual detection models," 2019 International Conference on Robotics and Automation (ICRA), Montreal, Canada, 2019, pp. 5752-5758. doi: 10.1109/ICRA.2019.8794182.