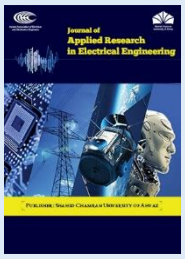
Shahid Chamran University of Ahvaz

Iranian Association of Electrical and Electronics Engineers

Journal of Applied Research in Electrical Engineering

E-ISSN: 2783-2864
P-ISSN: 2717-414X
Homepage: https://jaree.scu.ac.ir/



Research Article

Deep Learning-based Underwater Object Detection and Recognition for Multi-Robot Systems

ValiMohammad Nazarzehi Had [1,*] and Zobeir Raisi [1]

[1] *Chabahar Maritime University, Chabahar, Iran*

** Corresponding Author: v.nazarzehi@cmu.ac.ir*

Abstract: This paper introduces an innovative method for underwater object detection and recognition using a deep learning framework and autonomous underwater vehicles. The proposed method uses a deep learning framework with a network of underwater mobile robots operating within three-dimensional environments. To navigate these environments effectively, a random decentralized algorithm guides autonomous underwater vehicles (AUVs) in detecting objects within confined three-dimensional spaces. The deployment strategy involves utilizing the vertices of a cubic grid to optimize the positioning of AUVs across the search area. To enhance the object detection capabilities of mobile underwater robots within these environments and address the complexities of similar objects concealed in their surroundings, deep learning based on the YOLOv6 framework is refined to achieve improved performance and efficiency metrics. Specifically, the proposed Camouflaged Sea Edge and Attention Module (CSEA) is integrated into the YOLO framework, resulting in a substantial (8%) enhancement in mean average precision (mAP) performance, with a negligible impact (approximately 3%) on efficiency when evaluated on the publicly available TrashCan benchmark dataset. Additionally, to accelerate the training convergence and to better detect small objects within the TrashCan dataset, a distinct loss function based on Distance Intersection over Union (DIoU) is employed in the final proposed model. Several ablation studies are conducted to compare various YOLO models and evaluate the performance of the proposed CSEA module on the TrashCan dataset.

Keywords: Mobile Robots, AUVs, Distributed Control, Deep Learning, Object Detection

Article history
Received 22 October 2024; Revised 12 May 2025; Accepted 12 July 2025; Published online November 9, 2025.
© 2025 Published by Shahid Chamran University of Ahvaz & Iranian Association of Electrical and Electronics Engineers (IAEEE)

How to cite this article
V. Nazarzehi Had and Z. Raisi, "Deep learning-based underwater object detection and recognition for multi-robot systems," *J. Appl. Res. Electr. Eng.*, vol. 4, no. 1, pp. 24-33, 2025. DOI: 10.22055/jaree.2025.48211.1142

Check for updates

1. INTRODUCTION

In recent years, the progress in marine robot development has created fresh possibilities for ocean exploration. When combined with advanced machine vision techniques, self-driving underwater vehicles (AUVs) have proven very effective in studying the underwater world. Using a network of AUVs can significantly increase the speed and efficiency of certain tasks. This is especially valuable in tasks with tight deadlines or search and rescue operations. Because in tasks that require extensive sensing or data collection over a large area, multiple robots can cover more ground simultaneously. Moreover, having various robots provides redundancy. If one robot fails, others can continue the task, and this improves system reliability. In addition, multi-robot systems can be more easily scaled to handle larger workloads by adding more robots to the fleet [1].

By integrating deep learning techniques into such networks, researchers utilize these robots with the ability to autonomously process and interpret a vast array of underwater data, ranging from diverse marine species to submerged structures. Furthermore, to keep waste out of the ocean, it's beneficial to use self-operating marine robots to clear debris from the water, which depends on having a robust and accurate system for detecting objects [2].

Recently, deep learning models have demonstrated remarkable success in various applications, including object detection [3], visual place recognition [4], segmentation [5], classification [6], and video processing [7].



Fig. 1: Sample camouflaged objects in underwater images.

Deep learning architectures that are designed specifically for object detection are categorized into two-stage and one-stage frameworks. The one-stage architectures, such as YOLO [8] (You Only Look Once) and SSD [9] (Single Shot Detector), perform object detection in a single pass. On the other hand, two-stage object detection architectures, like Faster R-CNN [10] or Mask-RCNN [11], employ a two-step process. They typically involve a first stage that generates region proposals and a second stage that refines these proposals by predicting more accurate bounding boxes and object classes, enabling high-precision object detection but at the cost of increased computational complexity and slightly slower inference times, which makes them unsuitable for detecting objects in real-time in underwater environments.

With the advancement of deep learning techniques, several methods for detecting objects in underwater environments have been introduced [12-15]. While these algorithms take advantage of both the one- and two-stage deep learning frameworks to detect underwater objects, they still have difficulties in detecting challenging scenarios, including poor video quality, low visibility, contrast, and distortions in sharpness, non-uniform lighting, and reflections from suspended particles. As shown in Fig. 1 , one of the common challenges in underwater images is that the object has high similarity in hiding with its surrounding environment, which affects the final performance of the utilized detector. Moreover, the effectiveness of a group of autonomous underwater vehicles (AUVs) in searching for objects in situations where the presence of targets is uncertain and the environment is unpredictable relies on the efficiency of the search strategy.

In this paper, we extend the distributed random search algorithm presented in [16-18] by combining the proposed distributed search approach with deep learning object identification methods to improve the AUV's object detection abilities in the search environment. Using this algorithm, the AUVs move on the vertices of a cubic grid to detect objects in the entire search area. The vertices in the cubic grid will enhance the distribution of deployed autonomous underwater vehicles (AUVs). Consequently, they optimize the object detection task by efficiently spreading AUVs in the search area. In this paper, unlike the previous studies, it is assumed that the AUVs are equipped with object detection capabilities. Moreover, we discuss the pre-processing steps necessary to enhance underwater images, the architecture of convolutional neural networks tailored to underwater scenarios, the importance of annotated datasets for training, and the various challenges and ongoing research efforts in this field. We apply a newly designed module, namely CSEA, and analyze its performance on the baseline YOLOv6 [19] framework using the publicly available benchmark dataset [20]. Due to the lack of natural light and the blurry condition in underwater images, objects in the taken frames are close to the background. The main advantage of the designed CSEA module in combination with the YOLOv6 architecture is the capability to distinguish these types of objects. The main contributions of this paper can be summarized as follows:

1- This work proposes an extension to the distributed random search algorithm presented in [16-18] by integrating deep learning object detection methods, aiming to enhance the AUV's object detection capabilities in underwater search scenarios.
2- We used the YOLOv6 architecture [19] as the baseline and applied the proposed Camouflaged Sea Edge and Attention (CSEA) module to enhance the detection of camouflaged objects in underwater images.
3- To determine the most suitable YOLO architecture for underwater object detection, we trained and compared various YOLO detection models.
4- We conducted a comprehensive quantitative and qualitative ablation experiment to evaluate the performance of the proposed CSEA module when integrated with the baseline architecture.
5- To evaluate the effectiveness of our proposed model, we applied it to the well-known TrashCan dataset [20] and compared its performance with state-of-the-art (SoTA) methods, including [12-15,20].

2. Related Work

Underwater object identification has attracted remarkable research interest in recent years due to its extensive utility in fields such as marine engineering. Currently, advanced techniques for detecting underwater objects are based on deep learning frameworks, which can be summarized as follows.

2.1. Object Detection Architectures

Single Shot Detector (SSD) [9] and YOLO (You Only Look Once) [8] are well-known one-stage pipelines that aim to perform object detection and localization in a single pass through the network. These architectures divide the image into a grid of cells and predict bounding boxes and class scores directly for each cell. Typically, one-stage methods are faster due to their single-pass nature. They may need help with accurately localizing small objects or handling complex scenes with overlapping objects and often have lower accuracy compared to two-stage methods, especially on challenging datasets. Nevertheless, two-stage approaches like Faster-RCNN [21] and Mask-RCNN [22] pipelines are more accurate, especially in scenarios with small objects and complex backgrounds. These methods break the object detection process into the region proposal and object classification stages. Firstly, they produce region proposals (potential object bounding boxes) using techniques like

Region Proposal Networks (RPN). They then classify the proposed regions and refine the bounding box coordinates. However, two-stage methods are computationally more intensive due to the two-step process and may not be as applicable for applications that require real-time outputs as one-stage methods.

The one-stage and two-stage deep-learning object detection frameworks have been used in many SoTA underwater object detection algorithms [12-15] on benchmark datasets and achieved good performances. For example, Ref. [14] modified the Mask-RCNN [22] architecture by applying dilated convolution to improve the feature maps outputted from the feature pyramid network backbone, and an attention mechanism that led to adaptive learning for the network. These improvements make their proposed pipeline more suitable for detecting small garbage underwater. Ref. [20] introduced the benchmark TrashCan dataset and established a foundational detector using the Faster-RCNN [21] and Mask-RCNN [22] frameworks. This dataset and detector combination have become a significant reference point for subsequent research in trash detection, providing a standardized benchmark for evaluating detection performance and advancing the state of the art. Ref. [23] utilized the capabilities of YOLOv3 to identify underwater marine life and floating debris on the ocean's surface. Their study suggests that we can effectively use autonomous robots like drones and underwater vehicles as viable tools for the continuous monitoring and assessment of the marine environment. Ref. [24] combined the idea of deep-learning techniques and autonomous underwater vehicles (AUVs) to detect submerged objects in underwater environments effectively, which achieved promising results in underwater object detection. Ref. [24] proposed the idea of using a single autonomous underwater vehicle to identify submerged marine debris by analyzing Forward-Looking Sonar (FLS) imagery. To that effect, they first trained a deep learning Convolutional Neural Network (CNN) to perform image classification to recognize distinct objects within the FLS images. Then, the trained model is fed as an object detector by using a sliding window technique to identify and localize underwater objects. This early method highlights the capability of deep learning techniques and autonomous underwater vehicles in underwater object detection.

The above-mentioned innovative strategies represent significant contributions to the domain of detecting underwater objects using mobile robots, which forms the core focus of this paper. However, all of the mentioned studies utilized the idea of one AUV in their research. By using the potential of multiple autonomous underwater vehicles (AUVs), we can significantly enhance our ability to search and cover a wider range of underwater domains than using only one AUV, which increases efficiency, improves coverage, and enhances object detection capabilities in underwater environments.

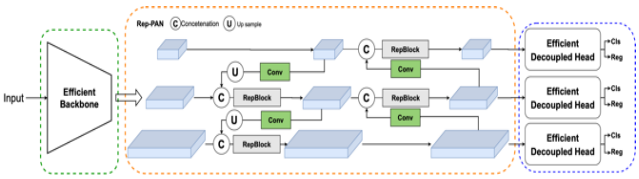


Fig. 2: The overall architecture of the baseline YOLOv6 [19].

2.2. YOLO Architectures for Object Detection

YOLO architectures [8], [25], [26] are single-stage object detection frameworks that do not require the Regional Proposal Network as used in two-stage frameworks like Faster-RCNN [21] and Mask-RCNN [22]. The YOLO architectures are also well-known for working well in multi-processing with powerful GPUs, and their networks not only reduce delays caused by hardware but also expand the range of applications in industrial settings.

After the success of the first version of YOLO introduced by [8], multiple improved versions of a framework based on YOLO have been proposed [8, 25, 26]. Inspired by ResNet and [27] Feature Pyramid Network (FPN) [28], the Ref. [26] improved the performance of the previous generation by modifying the feature extraction backbone via using a skip connection like ResNet (Darknet-53) and utilizing more prediction heads like FPN [28]. Ref. [25] improved the accuracy and inference time of the previous model by introducing a decoupled prediction head and modifying the feature extraction backbone. YOLOv6 [19] also introduced a new backbone called EfficientRep and modified the middle parts of the architecture by introducing Rep-PAN Neck, which led to a more efficient pipeline in terms of FPS and better performance regarding precision compared to the previous frameworks. It is worth mentioning that the researcher has developed other versions of YOLOs, and their performance and efficiency are slightly better in terms of Precision and FPS compared to the previous architectures. The models and their implementation of these models, such as YOLOv5 [29], YOLOR [30], YOLOv8 [31], and YOLONAS [32] can be found on GitHub.

YOLOv6 [19] is a recent architecture from the YOLO world. Its framework has several advantages in object detection; it is designed to be hardware-efficient and fast while still maintaining high accuracy, making the YOLOv6 a well-suited object detection model for a wide range of applications. Furthermore, the YOLOv6 head is an anchor-free head, which means that it does not rely on predefined anchors to predict bounding boxes. Instead, the head predicts the center and size of each object directly, which makes the YOLOv6 head faster and more efficient than anchor-based heads like Faster-RCNN [21]. Fig. 2 illustrates an overview of the YOLOv6 pipeline, which is composed of three main components: Backbone, neck, and Head, which can be summarized as follows. Firstly, the Efficient-backbone module extracts features from the input image.

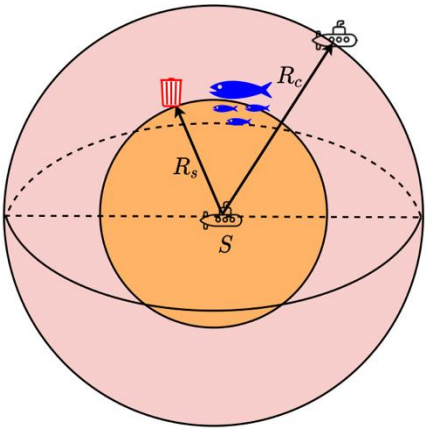


Fig. 3: The sensing and communication ranges are represented by R_s and R_c, respectively.

Then, the elements from the backbone are passed through the Rep-PAN neck to fuse them and enhance the detection features. Afterward, the combined features are passed through the anchor-free head to predict the center and size of each object. Finally, the predicted bounding boxes are post-processed to remove duplicates and filter out low-confidence detections.

Although the YOLOv6 architecture has shown remarkable performance in general object detection datasets [33], its performance may decline on challenging ocean or underwater images due to the complexity of the objects in these environments and the low-resolution images. One of the challenges that affects the performance of the baseline YOLO6 model is that the object has high similarity with its background, which is abundant underwater. These types of images, as shown in Fig. 1 make the baseline model perform poorly on these images. Inspired by the recent progress in this field and aiming to enhance the accuracy of the baseline model for these images, we incorporated a novel module known as the Camouflaged Sea Edge and Attention (CSEA) module onto the YOLOv6 architecture. This addition aids the model in more effectively identifying images of this nature.

3. METHODOLOGY

In this paper, we consider the challenge of object detection by a network of AUVs equipped with deep-learning object detection algorithms in bounded three-dimensional environments. Assume that mobile underwater robots are deployed to detect objects. Note that, in real-world scenarios, mobile robots can exhibit a diverse range of movement patterns. In this study, we specifically examine a system where mobile robots move randomly within the search area. We assume that the AUVs are using sensors with limited-range object detection capabilities, denoted as R_s. The AUVs can only identify objects located inside their detection zone, which is a sphere of radius R_s. Additionally, we consider an object as identified when it falls within the sensing range of a mobile underwater robot.

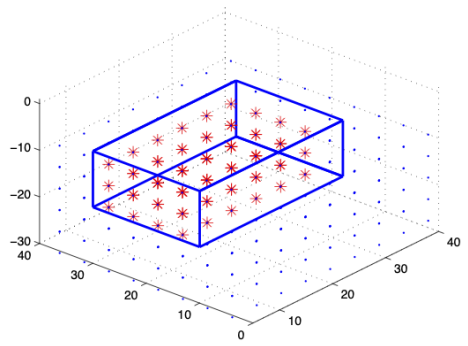


Fig. 4: The specified target region (T) is depicted as a cube, $\hat{\mathrm{V}}$ denoted by *,V represented by '.' [17].

To guide an autonomous underwater vehicle (AUV) back to the search area, upon reaching the boundary of the search region, it is given a new random heading that steers it back into the search area. The objective of the search is to detect objects in the given search area. We also adopt a spherical communication model for each AUV, denoted as $R_c > 0$, as shown in Fig. 3. This implies that AUV i can gather information about its neighbors within a sphere with a radius R_c defined by:

$$S_{i,R_c} = \{p \in R^3; \| \left(p - p_i(K)\right) \| \leq R_c\} \quad (1)$$

Here, we utilize a random search algorithm based on a cubic grid to effectively disperse the autonomous underwater vehicles (AUVs) throughout the search area, enabling object detection within a confined 3D space. Consider a cubic grid that divides the target region T into uniform cubes, each having sides of length. $\frac{2R_S}{\sqrt{3}}$. Denote V as the infinite set comprising the centers of all cubes within this grid. The set $\hat{\mathrm{V}} = \mathrm{V} \cap \mathrm{T}$ is referred to as a cubic covering set for T, as depicted in Fig. 4. In the initial phase of this algorithm, the AUVs create a shared cubic grid that covers the entire search area. Subsequently, they execute the object detection task by navigating randomly among the grid vertices.

As illustrated in Fig. 5, autonomous underwater vehicles (AUVs) are represented as spheres with a sensing radius denoted as R_s, and they navigate randomly within the designated search area. Throughout the search process, these mobile underwater robots retain information about detected objects and share this data with other robots within their communication range. In the depicted scenario, AUVs 1 and 5, as well as AUV 1 and 3, fall within each other's communication ranges, facilitating the exchange of search information between them. Consequently, AUV 3 and 5 establish indirect communication through AUV 1, optimizing the search task by avoiding redundant searches for objects already detected by their counterparts. Note that, to always keep the mobile underwater robot inside the search area, the algorithm calculates how far the robot can move in the next step without going outside the boundary. If the calculated distance would take the robot beyond the search area, a new distance is chosen. This repeats until a valid distance is found, ensuring the robot stays within the designated region.

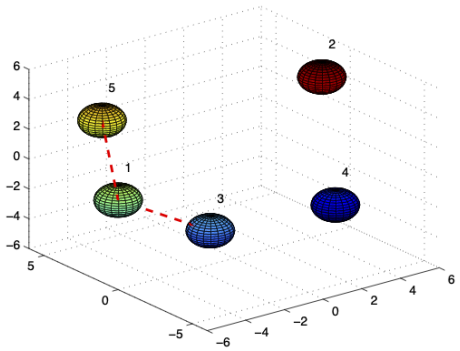


Fig. 5: Spheres denote the AUVs, -- depicts communication between the neighboring AUVs [18].

The search operation is conducted by the mobile underwater robots, which navigate randomly to the vertices of a cubic grid using random movements. Initially, it is assumed that autonomous underwater vehicles (AUVs) lack a common covering grid and a shared coordinate system. Consequently, consensus variables are employed to establish a unified covering grid, as outlined in [29]. This covering grid consists of the centers of cubes that tessellate the search area.

$$V_1 = \left(x + \alpha_1 \frac{2R_s}{\sqrt{3}}, y + \alpha_2 \frac{2R_s}{\sqrt{3}}, z + \alpha_3 \frac{2R_s}{\sqrt{3}}\right) \quad (2)$$

Vector V_1 denotes the positions of cube centers utilized to tessellate the search area. In this context, α_1, α_2, and α_3 belong to the set of integers Z. The algorithm takes the sensing range R_s and coordinates (x, y, z) as inputs, serving as a seed for the 3D grid's growth. For establishing a shared covering grid, it is assumed that each AUV possesses consensus variables $x_i(k)$, $y_i(k)$, and $z_i(k)$ within its coordinate system.

The sensing range R_s and a point $q_i(k) = \big(x_i(k), y_i(k), z_i(k)\big)$ uniquely define a 3D grid. Hence, $q_i(k)$ and R_s uniquely determine a 3D grid in T. The consensus variables $q_i(k) = [x_i(k) \quad y_i(k) \quad z_i(k)]$ along with scalar parameters α_1, α_2, α_3 and R_s describe the positions of the grid's vertices. The AUVs initiate with different values for coordination variables $x_i(0)$, $y_i(0)$, and $z_i(0)$, ultimately converging to consensus values x_0, y_0, and z_0, establishing a common coordinate system for the AUVs. Let $p_i(k)$ represent the coordinates of AUV i at time k, and $C[q, R_s]$ defines the nearest vertices of the 3D grid to point p. The following equations demonstrates the rules for updating the consensus variables and the coordinates of AUVs, which can be commutated as:

$$x_i(k+1) = \frac{x_i(k) + \sum_{j \in N_i(k)} x_j(k)}{1 + |N_i(k)|}$$
$$y_i(k+1) = \frac{y_i(k) + \sum_{j \in N_i(k)} y_j(k)}{1 + |N_i(k)|} \quad (3)$$
$$z_i(k+1) = \frac{z_i(k) + \sum_{j \in N_i(k)} z_j(k)}{1 + |N_i(k)|}$$

$$p_i(k+1) = C[q_i(k), R_s]\big(p_i(k)\big) \quad (4)$$

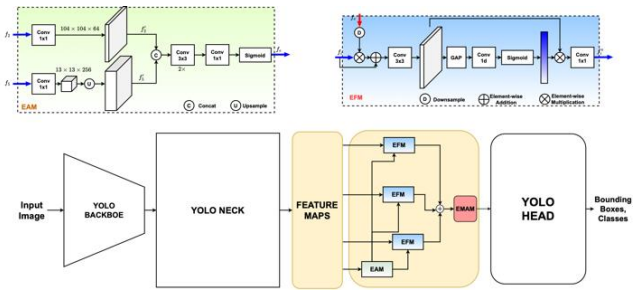


Fig. 6: The overall diagram of the proposed underwater object detection framework. The EAM (up-right) and EFM (up-left) modules were redesigned from [36] and [37], respectively.

Here, $N_i(k)$ represents the set of neighboring AUVs j (where $j \neq i$ and $j \in 1,2,\ldots,n$) within its communication range, and $|N_i(k)|$ denotes the number of neighbors at time k. Following the above equations, each underwater robot updates its consensus variables by averaging its values with those of its neighbors within its communication range. After building the common grid, the AUVs travel between grid vertices until they identify all objects using the introduced deep learning method.

3.1. Baseline Model

In this work, inspired by the YOLOv6 [19] and the camouflage object detection [34,35] architectures, we design a new framework, as shown in Fig. 6 to improve the baseline YOLOv6's (see Fig. 2) performance in underwater object detection. As seen, the network first extracts the features from the backbone, and then the extracted features to enhance the features are fed into the Rep-PAN neck to further improve the receptive field and the localization of multi-level feature maps. The resulting multi-level features are provided in the proposed module (CSEA), which consists of three main sub-blocks (more details in section 3.2) for processing to better focus on the region of interest of the object. Finally, the outputs of the CSEA features are passed through an anchor-free prediction head module that outputs the final bounding box and classes of the input image.

3.2. Camouflaged Sea Edge and Attention (CSEA) Module

Inspired by [36], after extracting multi-level features from the backbone and the neck of the YOLOv6 architecture, we integrated a new module (as shown in yellow Fig. 6) to help the model effectively distinguish the object structure and details. The integrated module is composed of three sub-blocks: edge-aware module (EAM), edge-guidance feature module (EFM), and edge mask attention module (EMAM).

The EAM sub-block combines the low and high-level features coming from the neck of YOLOv6, which effectively extracts object boundary-related edge semantics, which later help the upcoming sub-bloks to better capture edge features that are important in close to background object.

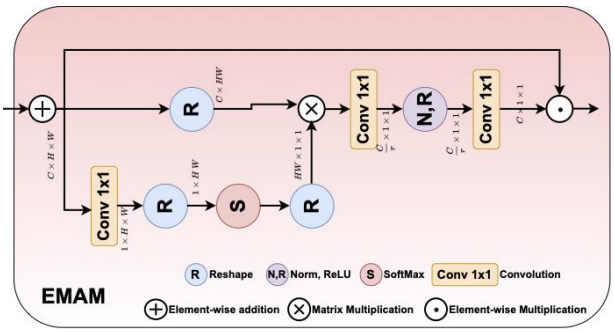


Fig. 7: The utilized Edge Mask Attention Module (EMAM) [38].

To that effect, as seen in Fig. 7, both the low-level and high-level features pass through a two 1x1 convolution layer to output an edge feature (f_e) that is obtained by concatenation and convolution operations, followed by the Sigmoid function.

The EFM sub-block works similar to an attention module in a vision transformer [37] pipeline. It enhances the representation learning in object detection by incorporating boundary-related edge cues into feature representations. This module uses a channel attention mechanism locally to improve cross-channel interaction and identify important alerts among different feature channels. The process involves element-wise multiplication between the input features (f_i) and edge features (f_e), followed by a skip-connection and a 3×3 convolution to produce fused components.

Finally, we utilize a new attention module, namely edge mask attention module (EMAM), to enable the model to obtain more contextual semantics and focus accurately on different regions of the feature maps more likely to contain objects [38]. The proposed sub-block is shown in Fig. 4. Assume F be the fused multi-level feature maps coming from the EAM and EFM sub-blocks. As shown, the global attention map ($\mathcal{G}$) in the EMAM sub-block can be computed as follows:

$$\mathcal{G} = \hat{F} \otimes \left(c_\gamma \sum_{\forall i} \frac{\exp^{c_\alpha f_i}}{\sum_{\forall j} \exp^{c_\alpha f_j}} F_i \right) \quad (5)$$

where c_α and c_γ are 1x1 convolutions that are learnable. The $\hat{F}$ is the 1-dimensional (1D) version of F. The i and j in (5) mean the enumeration for all the positions of the 2D input feature maps, and the $\otimes$ operation denotes matrix multiplication. Then to further achieve channel-wise dependencies of (5), the global attention maps transformed ($\mathcal{T}$) as follows:

$$\mathcal{T} = \sigma(c_1(R(\mathrm{L_N}(c_2(S))))) \quad (6)$$

where σ denotes the sigmoid function. c_1 and c_2 represent the trainable 1x1 convolutions. R and $\mathrm{L_N}$ show the RelU function and layer normalization, respectively.

The final output ($\mathcal{Y}$) of EMAM sub-block can be calculated as follows using (5) and (6):

$$\mathcal{Y} = \mathcal{G} \odot F \quad (7)$$

where $\odot$ shows the element-wise multiplication.

3.3. Loss Function

The head of the YOLOv6, as shown in Fig. 2, uses two loss functions for the final prediction: the classification loss that predicts the category of the object and the bounding box regression loss that outputs the coordinates of the bounding box. For the classification, an improved combination version of cross entropy and focal loss (FL) [39] called Varifocal Loss ($\mathrm{V_{fl}}$) [40] is used, which is defined as follows:

$$V_{fl}(D_i, G_j) = \begin{cases} -G_j(G_j log(D_i) + (1 - G_j) log(1 - D_i), & G_j > 0 \\ -\alpha D_i^{\gamma}(1 - D_i), & G_j = 0 \end{cases} \quad (8)$$

Where D_i is the predicted class score by the model, and G_j denotes the score of the ground truth in the dataset. We use the Distance-IoU (DIoU) Loss [41] for bounding box regression. DIoU is an improved version of GIOU [42] loss, which converges faster during training and works better on small and dense object detection.

$$\mathcal{L}_{DIoU} = 1 - IoU + \frac{\rho^2(D, G)}{c^2}. \quad (9)$$

where $\rho(\cdot)$ illustrates Euclidean distance. c represents the diagonal length of the smallest bounding box that covers both detected and ground-truth bounding boxes. IoU is the intersection over a union computed as:

$$IoU = \frac{|D \cap G|}{|D \cup G|}, \quad (10)$$

where $|.|$ denotes the area, $G = (x', y', w', h')$ is the ground-truth, and $D = (x, y, w, h)$ is the predicted box. The final loss of the model, which considers both the classification of (8) and (9), can be defined as :

$$\mathcal{L} = \lambda_b \mathcal{V}_{fl} + \lambda_v \mathcal{L}_{DIoU} \quad (11)$$

where λ_b and λ_v are hyperparameters that are set to 0.8 and 1 during our implementation.

4. Experimental Results

4.1. Implementation Details

The proposed model is fine-tuned until it reaches convergence (~ 150 epochs) by using the transfer learning technique. To optimize the parameters associated with the loss functions, the AdamW optimizer [43] is employed, accompanied by the poly-learning policy. This optimization algorithm ensures that the model learns and adapts effectively throughout the training process. The training phase is carried out on a 24 GB memory of the NVIDIA RTX 3090 GPU. We set the batch size to 16 during training.

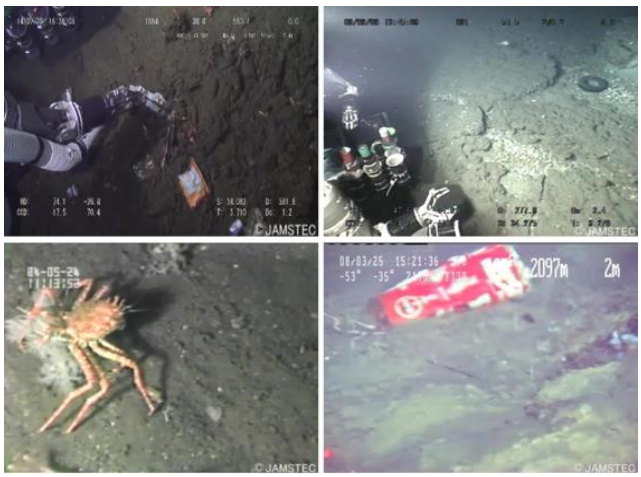


Fig. 8: Sample images of TrashCan Dataset [20].

Different augmentation strategies are applied during training, including image resizing, random rotations, and horizontal flipping. The augmentation process enables the final model to acquire a more comprehensive understanding of the TrashCan dataset and be more generalizable and robust in real-world, unseen scenarios.

4.2. TrashCan Dataset

The TrashCan dataset [20] is designed specifically for underwater object detection. The images of this dataset are collected using remotely operated underwater vehicles (ROVs), which are obtained from the J-EDI dataset [44].

The TrashCan dataset includes 7212 images of different underwater objects (22 diverse classes), mainly from the Sea of Japan. The resolution of images in this dataset changes from 480x270 pixels to 480x360 pixels, where they are annotated in the COCO dataset [33] annotation format using rectangular bounding boxes and segmentation labels. Fig. 8 illustrates some sample images of this dataset.

4.3. Evaluation Metrics

The Average Precision (AP) metric relies on precision-recall measures and works with the classes of multiple objects. It defines a positive prediction by using Intersection over Union (IOU). Precision measures the accuracy of the model for positive predictions. The Recall metric checks how many actual positive cases the model correctly recognizes. The precision (P) and Recall (R) can be calculated as follows:

$$P = \frac{TP}{TP + FP} \tag{12}$$

$$R = \frac{TP}{TP + FN} \tag{13}$$

where TP and FP denote true positives and false positives, respectively. False Negatives (FN) are the number of object regions that the model did not correctly predict.

Let G_j be the j^{th} ground-truth in the given image, and D_i be the i^{th} detection bounding box (D_i). Therefore, we can calculate the precision and recall based on the intersection over union (IoU) metric as follows:

Table 1: Comparing the proposed method's performance with SOTA methods [12-15] on the TrashCan [20] dataset. The best performance is highlighted in bold.

Model-Name	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
RISDU [12]	0.38	0.49	0.32	0.38	—	—
GCC-Net [13]	0.41	0.61	—	—	—	—
Hong et al. [20]	0.29	0.51	0.28	0.21	0.30	0.40
Deng et al. [14]	0.44	0.65	**0.48**	0.35	**0.48**	**0.66**
MLDeT [20]	0.42	0.63	0.46	0.37	0.42	0.57
Proposed	**0.45**	**0.66**	0.47	**0.38**	0.46	0.63

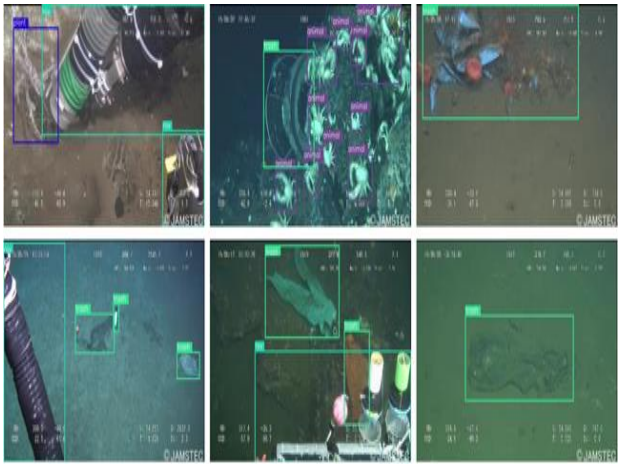


Fig. 9: Qualitative results of the proposed model on challenging samples of the TrashCan Dataset [20].

$$\text{IoU} = \frac{\text{Area}(G_j \cap D_i)}{\text{Area}(G_j \cup D_i)} \tag{14}$$

In the end, for handling multiple object categories, the mAP can be formulated as follows:

$$mAP = \frac{1}{N}\sum_{i=1}^{N} A\,P_i \tag{15}$$

where AP_i denotes the average precision of the i −th class.

4.4. Quantitative Results

In this section, we carried out a thorough evaluation of underwater object detection methods using the mean Average Precision (mAP) is a widely accepted metric in the object detection community. Table 1 presents a quantitative comparison of our proposed model with SoTA techniques on the TrashCan dataset. Our model outperforms other methods [12-15] on the TrashCan dataset, achieving the highest mAP.

4.5. Qualitative Results

Fig. 9 illustrates the qualitative results of the proposed model on some challenging images of the TrashCan dataset. As seen, the proposed model accurately detects objects in the images, which confirms the effectiveness of the proposed model.

Table 2: Comparison of the map performances of different deep learning models [19, 25, 26] on the TrashCan dataset.

Model	mAP
YOLOv3	0.33
YOOX	0.35
YOLOv6[Baseline]	**0.37**

Table 3: The effect of the utilized module on the baseline model. The best performance is highlighted in bold. "FPS" demonstrates the frames per second.

Model	Module	mAP	FPS
Baseline	-	0.37	**68**
Proposed	**CSEA**	**0.45**	65

It is worth mentioning that there are certain limitations to the proposed architecture, which may fail in some sample images. The proposed method has difficulty in the detection of low-resolution images and objects that are heavily occluded. These challenges are common and unresolved issues in the object detection community. One potential solution for addressing these challenges is augmentation techniques during training and using a higher number of instance images that contain low-resolution and occluded objects.

4.6. Ablation Study

We evaluate the performance of several SOTA deep-learning models using mean Average Precision (mAP) and Frames Per Second (FPS) metrics. To that effect, we first evaluated different YOLO models on the TrashCan dataset. As shown in Table 2, the YOLOv6 model has the best performance in terms of mAP, which makes it a good candidate as our baseline detection model for further investigation. Building upon this foundation, we applied the introduced CSEA module, which we integrated into the YOLOv6 architecture, as demonstrated in Table 3. The incorporation of the CSEA module on the baseline leads to a remarkable performance boost, signifying a substantial improvement in object detection accuracy. The successful detection of camouflage objects in Fig. 10 also confirms the effectiveness of the utilized module in the baseline architecture. We also experimented with the efficiency of our models. As seen in Table 3, the baseline model achieved a good efficiency in terms of FPS (~ 68), which confirms it can be used in real-time applications like underwater object detection using mobile robots. It can also be seen that applying the designed CSEA module does not affect the efficiency of the final model, and it obtains a close speed to the baseline model.

5. Discussion and Future Work

Deep learning models are renowned for their requirement of substantial data for training, which often leads to enhanced inference performance and more generalization capabilities on novel datasets. Our work falls into this category as well.

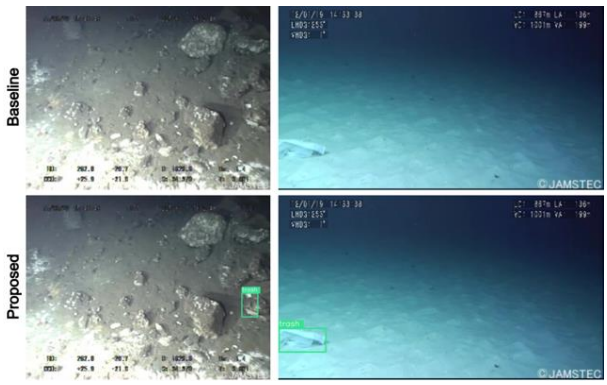


Fig. 10: Comparing qualitatively the proposed model with the baseline on objects that have a color close to the background.

As evident from the results presented in Table 1. While our proposed model achieved the SoTA performance on the TrashCan dataset, with an average precision of 0.45, these results fall short of practical application in real-world scenarios. However, this performance can be improved by utilizing a larger dataset, which could be obtained from real-world annotated datasets that are costly or can be synthetically created in limited environments to further investigate the illumination variation and generalization capability of the model, such as the pool that we intend to explore in future research. Ultimately, our objective is to apply the findings of deep learning detections to real-world autonomous underwater vehicle robots for practical and meaningful applications.

6. Conclusions

In this study, we have presented the idea of integrating distributed random search techniques with deep learning techniques to identify challenging cases in underwater images using AUVs within a 3D search environment. After experimenting with different baseline object detection frameworks, we have selected the YOLOv6 pipeline as the baseline, which has better efficiency and accuracy performance. We have designed a new module, namely Camouflaged Sea Edge and Attention Module (CSEA), and integrated it with YOLOv6 architecture to improve the detection of objects hidden in the background. We have compared our model with SoTA models on the benchmark TrashCan dataset that is designed for underwater object detection and recognition. The quantitative results have shown that the final model achieved competitive performance in comparison with SOTA methods on the TrashCan using the Average Precision metric and a real-time performance (~ 65) in terms of FPS. The CSEA module in the architecture boosted the mAP by a large margin (~ 8%) than the baseline and had a minor effect (~ 3) on the FPS. We have also provided qualitative results to show the performance of the

model and the designed CSEA module on challenging underwater images.

CREDIT AUTHORSHIP CONTRIBUTION STATEMENT

ValiMohammad Nazarzehi Had: Investigation, Methodology, Visualization, Roles/Writing - original draft, Writing - review & editing. **Zobeir Raisi**: Conceptualization, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Roles/Writing - original draft, Writing - review & editing.

DECLARATION OF COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper. The ethical issues; including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, redundancy has been completely observed by the authors.

REFERENCES

[1] V. Nazarzehi and A. V. Savkin, "Decentralized three-dimensional formation building algorithms for a team of nonholonomic mobile agents," *International Journal of Control, Automation and Systems*, vol. 17, pp. 1283–1292, 2019.

[2] J. Li, G. Zhang, C. Jiang, and W. Zhang, "A survey of maritime unmanned search system: theory, applications and future directions," *Ocean Engineering*, vol. 285, p. 115359, 2023.

[3] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," *Proceedings of the IEEE*, 2023.

[4] Z. Raisi and J. Zelek," Visual place recognition from end-to-end semantic scene text features," *Frontiers in Robotics and AI*, vol. 11, p.1424883 2024.

[5] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7, pp. 3523–3542, 2021.

[6] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural computation*, vol. 29, no. 9, pp. 2352–2449, 2017.

[7] M. Ghafari, A. Amirkhani, E. Rashno, and S. Ghanbari, "Smart AI-based Video Encoding for Fixed Background Video Streaming Applications," *Journal of Applied Research in Electrical Engineering*, 2023 vol. 2, no. 1, pp. 37-44.

[8] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. on Comp. Vision and Pattern Recognition*, pp. 779– 788, 2016.

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Eur. Conf. on Comp. Vision*, pp. 21–37, Springer, 2016.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[11] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2961–2969, 2017.

[12] B. C. Corrigan, Z. Y. Tay, and D. Konovessis, "Real-time instance segmentation for detection of underwater litter as a plastic source," *Journal of Marine Science and Engineering*, vol. 11, no. 8, p. 1532, 2023.

[13] L. Dai, H. Liu, P. Song, and M. Liu, "A gated cross-domain collaborative network for underwater object detection," *arXiv preprint arXiv:2306.14141*, 2023.

[14] H. Deng, D. Ergu, F. Liu, B. Ma, and Y. Cai, "An embeddable algorithm for automatic garbage detection based on complex marine environment," *Sensors*, vol. 21, no. 19, p. 6391, 2021.

[15] D. Ma *et al.*, "Mldet: Towards efficient and accurate deep learning method for marine litter detection," *Ocean & Coastal Management*, vol. 243, p. 106765, 2023.

[16] V. Nazarzehi, A. V. Savkin and A. Baranzadeh, "Distributed 3D dynamic search coverage for mobile wireless sensor networks," *IEEE Communications Letters*, vol. 19, no. 4, pp.633-636, April 2015.

[17] V. Nazarzehi and A. V. Savkin. "Distributed self-deployment of mobile wireless 3D robotic sensor networks for complete sensing coverage and forming specific shapes," *Robotica*, vol. 36, no. 1, pp. 1-18, 2018.

[18] V. Nazarzehi and A. Baranzadeh, "A distributed bio-inspired algorithm for search of moving targets in three-dimensional spaces," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Zhuhai, China, 2015, pp. 2507-2512.

[19] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, *et al.*, "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.

[20] J. Hong, M. Fulton, and J. Sattar, "Trashcan: A semantically segmented dataset towards visual detection of marine debris," *arXiv preprint arXiv:2007.08097*, 2020.

[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. in Neural Info. Process. Sys.*, pp. 91–99, 2015.

[22] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. on Comp. Vision*, pp. 2961–2969, 2017.

[23] J.-I. Watanabe, Y. Shao, and N. Miura, "Underwater and airborne monitoring of marine ecosystems and debris," *Journal of Applied Remote Sensing*, vol. 13, no. 4, Art. no. 044509, 2019.

[24] M. Valdenegro-Toro, "Submerged marine debris detection with autonomous underwater vehicles," in *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*, pp. 1–7, IEEE, 2016.

[25] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, "Yolox: Exceeding yolo series in 2021," *arXiv preprint arXiv:2107.08430*, 2021

[26] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770– 778, 2016.

[28] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

[29] G. Jocher, K. Nishimura, T. Mineeva, and R. Vilarino, "Yolov5 (2020)," *GitHub repository. [Online]. Available: https://github.com/ultralytics/yolov5, 2020.*

[30] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "You only learn one representation: Unified network for multiple tasks," *arXiv preprint arXiv:2105.04206*, 2021.

[31] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics yolov8," *[Online]. Available: https://github.com/ultralytics/ultralytics*, 2023.

[32] J. Terven and D. Cordova-Esparza, "A comprehensive review of yolo: From YOLOv1 to YOLOv8 and beyond," *arXiv preprint arXiv:2304.00501*, 2023.

[33] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Com-´ mon objects in context," in *Computer Vision–ECCV 2014: 13th European Conference*, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740–755, Springer, 2014.

[34] D.-P. Fan, G.-P. Ji, G. Sun, M.-M. Cheng, J. Shen, and L. Shao, "Camouflaged object detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2777–2787, 2020.

[35] H. Mei, G.-P. Ji, Z. Wei, X. Yang, X. Wei, and D.-P. Fan, "Camouflaged object segmentation with distraction mining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8772–8781, 2021.

[36] Y. Sun, S. Wang, C. Chen, and T.-Z. Xiang, "Boundary-guided camouflaged object detection," *arXiv preprint arXiv:2207.00794*, 2022.

[37] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," arXiv preprint arXiv:2010.11929, 2020.

[38] Z. Raisi and J. Zelek, "Investigation of deep learning optimization algorithms in scene text detection," *International Journal of Industrial Electronics Control and Optimization*, vol. 6, no. 3, pp. 171–182, 2023.

[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980–2988, 2017.

[40] H. Zhang, Y. Wang, F. Dayoub, and N. Sunderhauf, "Varifocalnet: An iou-aware dense object detector," in *CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8510–8519, 2020.

[41] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-iou loss: Faster and better learning for bounding box regression," *in Proceedings of the AAAI conference on artificial intelligence*, vol. 34, pp. 12993–13000, 2020.

[42] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 658–666, 2019.

[43] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[44] C. H. Lucas et. al., "Gelatinous zooplankton biomass in the global oceans: geographic variation and environmental drivers," *Global Ecology and Biogeography*, vol. 23, no. 7, pp. 701–714, 2014.

BIOGRAPHY



Valimohammad Nazarzehi Had received his Ph.D. degree in 2016 from the University of New South Wales, Australia. Currently, he is an assistant professor in the Department of Electrical Engineering, Chabahar Maritime University, Iran. His research interests include decentralized control, marine control systems, control of mobile robots, robotics, and image processing



Zobeir Raisi received his Ph.D. degree in Systems Design Engineering from the University of Waterloo, Waterloo, Canada. His graduate research focused on advanced techniques in text detection and recognition in wild images, object detection, and segmentation.

Following his Ph.D., he served as a Postdoctoral Fellow in the Department of Systems Design Engineering at the University of Waterloo, collaborating with the Vision and Image Processing Lab on several high-impact academic-industrial projects. These projects spanned areas of computer vision, image processing, and machine learning, and engaged notable partners, including Apple Inc., Baltimore Orioles Inc., and ATS Automation Inc.

Dr. Raisi's research interests are computer vision, deep learning, and artificial intelligence. He currently holds a faculty position as an Assistant Professor at Chabahar Maritime University.

Copyrights
2025 Licensee Shahid Chamran University of Ahvaz, Ahvaz, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution –NonCommercial 4.0 International (CC BY-NC 4.0) License (http://creativecommons.org/licenses/by-nc/4.0/).