Highlights

Edge-Guided Oceanic Scene Element Detection

Keke Xiang,Xingshuai Dong,Weibo Wang,Xinghui Dong

- Introducing a novel computer vision task, i.e., oceanic scene element detection.
- Making the first effort on oceanic scene element image collection and annotating 10,040 images.
- Proposing a Multi-scale Edge-Guided Module, to guide the backbone toward learning edge characteristics.
- Designing an Edge-Guided Oceanic Scene Element Detection framework and generating a set of benchmarks.

Edge-Guided Oceanic Scene Element Detection*

Keke Xiang[a], Xingshuai Dong[b], Weibo Wang[a] and Xinghui Dong[a,*]

[a]School of Computer Science and Technology, Ocean University of China, China
[b]School of Systems and Computing, University of New South Wales, Canberra, Australia

ARTICLE INFO

Keywords:
Object Detection
Edge-Guided Object Detection
Oceanic Scene Element Detection
Multi-task Learning
Deep Learning

ABSTRACT

Oceanic environmental monitoring is critical to environmental protection. As a core technique, Oceanic Scene Element Detection (OSED) plays an important role. Existing oceanic object detection approaches are usually focused on a single category. Therefore, a multi-category OSED data set is demanded. Considering oceanic scene elements normally present large-scale complicated structures, the edge cue is particular useful for representation of these elements. However, none of existing object detection methods take this cue into account. To address the two problems, we first collect and annotate three OSED data sets, which comprise a total of 10,040 images and 60 categories. Then we propose a generic Multi-scale Edge-Guided Module (MSEGM), which can be inserted into an object detection network, for guiding the backbone toward learning edge characteristics. An Edge-Guided Oceanic Scene Element Detection (EG-OSED) framework is built on top of this module and a base object detector, which can be end-to-end trained using a multi-task learning scheme. A series of experiments are conducted on the three OSED data sets. The results demonstrate that the EG-OSED framework normally outperforms its base object detector which does not utilize edges. We believe that these promising results should be due to the importance of edges to representation of oceanic scene elements.

1. Introduction

The ocean covers around 71% of the surface of the Earth and contains rich animal, plant and mineral resources. With the reduction of land resources, more and more emphasis has been put on ocean exploration activities. But these activities may impair the oceanic environment. In this case, oceanic environmental monitoring is critical to oceanic environmental protection. Detection of ordinary objects has received much attention in the field of computer vision [1–3]. Object detection has also been widely applied to anomaly detection [4–6], autonomous driving [7, 8] and ocean engineering [9, 10]. However, this is not the case for detection of oceanic scene elements. Since Oceanic Scene Element Detection (OSED) (see Fig. 1) is useful for analyzing and understanding oceanic scene images or videos, this task plays an important role in monitoring the oceanic environment.

Object detection aims to detect semantic objects, which belong to a certain category, shown in images and videos. Traditional object detection methods have been applied to oceanic object detection. Schwegmann et al. [11] detected ships in Synthetic Aperture Radar (SAR) images using a Constant False Alarm Rate ship pre-screening method and a cascade classifier with Haar-like features. Alsahwa et al. [12] extracted HOG features from sliding-windows and classified these windows using an SVM classifier. However, the performance of those methods is dependent on the choice of the features and classifiers.

Fig. 1. The results of the proposed oceanic scene element detection method on two sea stack ("ShiLaoRen" or "Old Stone Man", located at Qingdao, China) images.

With the rapid development of deep learning techniques [2, 3, 13–19], in particular, Convolutional Neural Networks (CNNs) [13, 14], object detection has made great progress as these techniques are able to learn features from training samples directly. A large number of deep learning-based object detection methods were developed, including the R-CNN series [3, 17, 20], the YOLO series [18, 19] and the Single Shot MultiBox Detector (SSD) series [2].

Deep learning techniques have also been widely used for oceanic object detection. Liu and Wang [21] performed marine benthos detection using the Faster R-CNN [17] and Feature Pyramid Network (FPN) [22] methods. In [23], both YOLOv3 [18] and attention mechanism were used for small ship detection. Ye et al. [24] detected ships on top of YOLOv5 [25] and BiFPN [26]. To detect objects from ship safety plans, a CNN was proposed [9]. In [27], oceanic animal detection was carried out. Besides, oceanic object detection was conducted using radar images [28, 29], infrared images [24] and underwater images [10].

However, none of the above-mentioned studies were conducted for the OSED task. Existing oceanic object detection studies were normally focused on a single category of objects. In this case, a multi-category OSED data set

*Source code and data sets are available at: https://indtlab.github.io/projects/EG-OSED.

*Corresponding author
xkk9068@stu.ouc.edu.cn (K. Xiang); xingshuai.dong@adfa.edu.au (X. Dong); wwb@stu.ouc.edu.cn (W. Wang); xinghui.dong@ouc.edu.cn) (X. Dong)

is required. On the other hand, oceanic scene elements, for example, bay, beach, buoy, cargo port, headland, intertidal zone, island and so on, normally present large-scale complicated structures. Since edges encode both the shape and semantic information of an image, they are important characteristics of the image. The usefulness of the edge cue to representation of shapes and scenes has been demonstrated [30]. Although edges have been used in many computer vision tasks, such as salient object detection [31, 32], camouflage detection [33] and semantic segmentation [34], existing object detection approaches normally do not take this important cue into consideration.

To address the two issues, we first select oceanic scene elements from three publicly available data sets, including FOSD [35], Places365 [36] and SUN [37]. In total, 60 categories and 10,040 images are collected. These images are annotated in accordance with the format of Pascal VOC [38]. Thanks to those images, an object detection network can be trained for the OSED task. Then we propose a generic Multi-scale Edge-Guided Module (MSEGM), which can be inserted into an object detection network, to guide the backbone toward learning edge characteristics. An Edge-Guided Oceanic Scene Element Detection framework, namely, EG-OSED, is further introduced. This framework contains the MSEGM and a base object detector, which can be end-to-end trained using a multi-task learning scheme. To our knowledge, none of the OSED data sets and the EG-OSED framework have been explored for object detection before.

The main contributions of this study can be summarized as fourfold.

- We make the first attempt to perform a novel task on detection of a wide range of oceanic scene elements. This task is key to oceanic environmental monitoring.
- We collect and annotate three oceanic scene element data sets, containing 60 categories and 10,040 images in total. To our knowledge, this work is the first effort made on oceanic scene element image collection. Those data sets can be used to train an object detector for the OSED task.
- We develop a generic Multi-scale Edge-Guided Module (MSEGM), which can be incorporated into an existing object detection network and is able to guide the backbone network toward learning edge characteristics, to boost the performance of object detection.
- An Edge-Guided Oceanic Scene Element Detection (EG-OSED) framework is proposed. This framework comprises a base object detector and the MSEGM, which can be end-to-end trained using a multi-task learning scheme. The EG-OSED framework is tested on the three data sets that we collect. The results provide the community with a series of benchmarks.

The rest of this paper is organized as follows. We review the related work in Section 2. In Section 3, we describe the data sets that we collect. The proposed EG-OSED framework is introduced in Section 4. Experimental setup and results are reported in Sections 5 and 6 respectively. Finally, our conclusion is drawn in Section 7.

2. Related Work

2.1. Object Detection

Due to the re-emergence of CNNs around 2012 [13], object detection algorithms can be divided into two groups: traditional and deep learning-based algorithms. Traditional object detection algorithms, such as Viola Jones detector [39], the HOG detector [40] and the Deformable Part-Based Model [41], normally rely on manual feature extraction. Since this process has a large amount of computational redundancy, it is less effective in complex scenes and multi-class object detection tasks.

Deep learning-based detection algorithms are normally designed on top of Deep Neural Networks (DNNs). These networks can perform not only feature extraction but also classification and/or regression. Those algorithms can be further categorized into two-stage and one-stage detection methods. The former first generate a set of region proposals which probably contain objects, and then classify and regress these proposals. The latter do not produce region proposals and perform detection directly.

Representative two-stage detection methods include the R-CNN [3, 17, 20] series, SPP-Net [42] and FPN [22]. Compared with one-stage detectors, these methods usually produce the higher accuracy but the detection speed is relatively slower. Classical one-stage detection approaches comprise the YOLO [1] series, SSD [2] and DETR [43]. With the development of object detection techniques, one-stage detectors have gradually surpassed two-stage detectors in terms of both the detection accuracy and speed.

2.2. Oceanic Object Detection Using Deep Learning

Deep learning techniques have also been applied to oceanic object detection. Similarly, these approaches can also be divided into two classes, including two-stage detection and one-stage detection.

2.2.1. Two-Stage Detection

Mou et al. [44] proposed an oceanic object detection method based on Faster R-CNN [17]. To solve the sample imbalance problem, the focal loss [45] function was utilized. In [46], a ship detection method was introduced, which was used for optical remote sensing images. A set of candidate regions were first obtained using a lightweight local Candidate Scene Network (L^2 CSN). Then ship detection was performed on these regions using a Visual Attention DSOD (VA-DSOD) [47] based approach.

Wu et al. [48] developed a ship detection method for high-frequency surface wave radar images. The Region of Interest (RoI) was first obtained from an image using the Extreme Value Detector (ED). A lightweight CNN (LW-CNN) was then used to extract features. Finally, a fast classification operation was conducted using the Extreme Learning Machine (ELM) [49]. In [50], the SK module [51]

was added into the backbone, to learn the better feature representation. The FPN structure was also used to improve the accuracy of dense marine bentho detection. Kong et al. [9] applied Faster R-CNN to ship safety plan detection. In [52], Mask R-CNN [53] was utilized for marine debris detection.

2.2.2. *One-Stage Detection*

Hass and Jokar Arsanjani [54] adopted a YOLOv3-based method for distinguishing icebergs and ships in SAR images. In [55], a ship detection method was introduced on top of YOLOv3 [18] and the Focal Binary Cross-Entropy (FBCE) loss function [55]. YOLOv3 was also employed for ship detection in optical remote sensing images [23]. The interference of clouds and waves was reduced by emphasizing the target region using a dilated attention module. Al Muksit et al. [56] proposed an underwater fish detection network on top of YOLOv3. The small object information loss was reduced by increasing the upsampling size in the Neck of YOLOv3. In [57], a fish detection network was developed based on YOLOv4 [19]. In [24], an infrared ocean ship detection method was designed on top of YOLOv5 [25], referred to as CAA-YOLO.

Oceanic object detection was also conducted using other object detection techniques. Wang et al. [58] built a lightweight marine object detection network using both the Context Attention Enhancement FPN (CAE-FPN) and improved ShuffleNet [59]. Yu et al. [60] further improved the accuracy of marine object detection by bringing together the coordinate competing attention, spatial supplement attention and path aggregation networks. In [61], both the cross-stage multiple-branching block and shuffle attention were introduced for underwater object detection. Jia et al. [62] utilized both the SSD [2] and YOLOv5 [25] for Floating Production Storage and Offloading (FPSO) crack detection.

2.3. Computer Vision Methods Using Edges

An edge can be considered the discontinuity in the distribution of image features. Edges can be found between an object and the background or between two objects. In essence, edges carry the majority of the semantic and shape information of an image. Dong and Chantler [30] investigated the importance of edges to the representation of images and designed a new image descriptor. Edges were also utilized in order to boost visual word based image descriptors by modeling the Spatial Layout of Words (SLoW) [63]. Other vision tasks, such as semantic segmentation [34], camouflage detection [33] and salient object detection [31, 32], also benefited from edges.

For the purpose of segmenting defects, Yeung and Lam [34] introduced a transformer-based boundary-aware method, in which the boundary data and the context information were integrated. Sun et al. [33] designed an edge-directed camouflage detection algorithm. Specifically, an edge-aware module was used to extract edge maps while an edge-directed feature module was used to weight object features using the edge maps. In [31], a one-to-one guidance module was developed for salient object detection. This module fused salient edge features with multi-resolution salient object features. Liu et al. [32] improved the details of salient objects using an additional edge prediction branch. Dong et al. [64] used edge attention features to guide the monocular depth estimation task.

Table 1
Comparison of the existing deep learning-based object detection approaches and the proposed method.

	Approach	Object Categories	Use of Edges
Two-Stage	Girshick et al. [20]	Ordinary Objects	No
	Girshick [3]	Ordinary Objects	No
	Ren et al. [17]	Ordinary Objects	No
	He et al. [42]	Ordinary Objects	No
	Lin et al. [22]	Ordinary Objects	No
	Mou et al. [44]	Marine Object	No
	Bi et al. [46]	Ship	No
	Wu et al. [48]	Ship	No
	Liu and Wang [50]	Marine Bentho	No
	Kong et al. [9]	Ship	No
	Sánchez-Ferrer et al. [52]	Marine Debris	No
One-Stage	Redmon et al. [1]	Ordinary Objects	No
	Liu et al. [2]	Ordinary Objects	No
	Carion et al. [43]	Ordinary Objects	No
	Hass and Jokar Arsanjani [54]	Iceberg and Ship	No
	Liu et al. [55]	Ship	No
	Chen et al. [23]	Ship	No
	Al Muksit et al. [56]	Underwater Fish	No
	Zhu et al. [57]	Fish	No
	Ye et al. [24]	Ship	No
	Wang et al. [58]	9 Marine Objects	No
	Yu et al. [60]	Echinus, Holothurian, Scallop and Starfish	No
	Li et al. [61]	Underwater Object	No
	Jia et al. [62]	FPSO	No
	Ours	60 Oceanic Scene Elements	Yes

2.4. Summary

In Table 1, we compared the existing object detection approaches that we reviewed in Sections 2.1 and 2.2 with the proposed method. As can be seen, the existing approaches normally utilized ordinary objects or a small number of oceanic object categories, which are not suitable for the OSED task. Furthermore, none of these approaches paid attention to the importance of the edge cue to representation of the shape and semantic information. In contrast, we collected and annotated three oceanic scene element data sets, which contain a total of 60 categories and 10,040 images. Inspired by the studies surveyed in Section 2.3, we proposed a Multi-scale Edge-Guided Module (MSEGM), which can be used to guide an object detection network toward learning edge characteristics, to improve the detection performance.

3. Oceanic Scene Element Data Sets

Existing oceanic object data sets normally contain a single category, such as fishes [57], ships [11] or oceanic mammals [27]. Hence, the detector trained using one of these data sets can only be used to detect the objects within a specific category. Moreover, some data sets consist of radar [29, 54] or infrared [24] images. The detector trained using these data sets may be not applicable to RGB images.

In this study, we focused on performing oceanic scene element detection across a relatively wide range of categories. To this end, we constructed three oceanic scene element

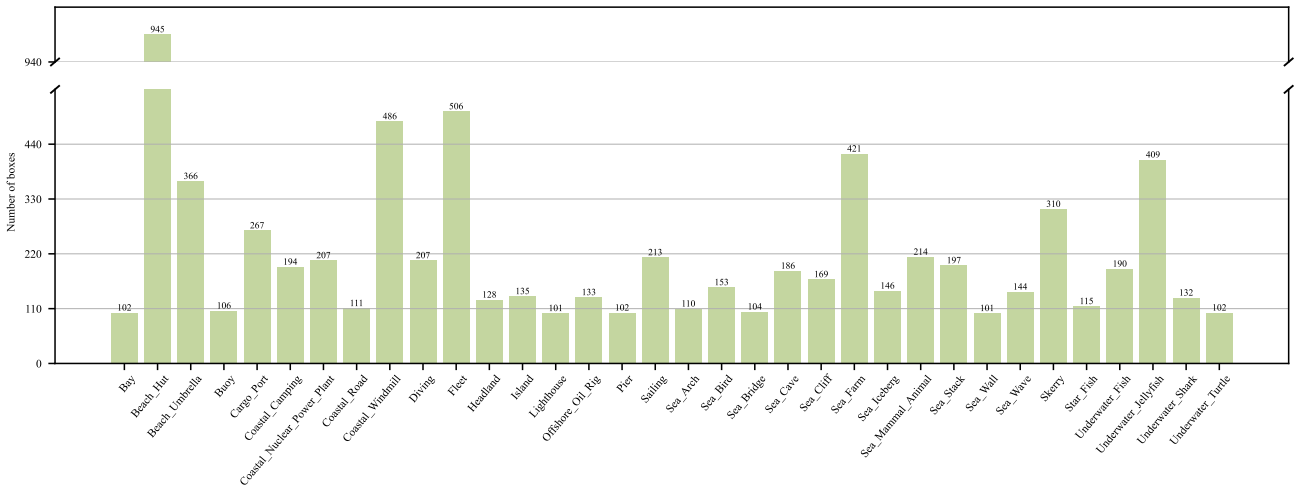


Fig. 2. The number of bounding boxes contained in each of the 34 categories of the FOSD_OD data set.

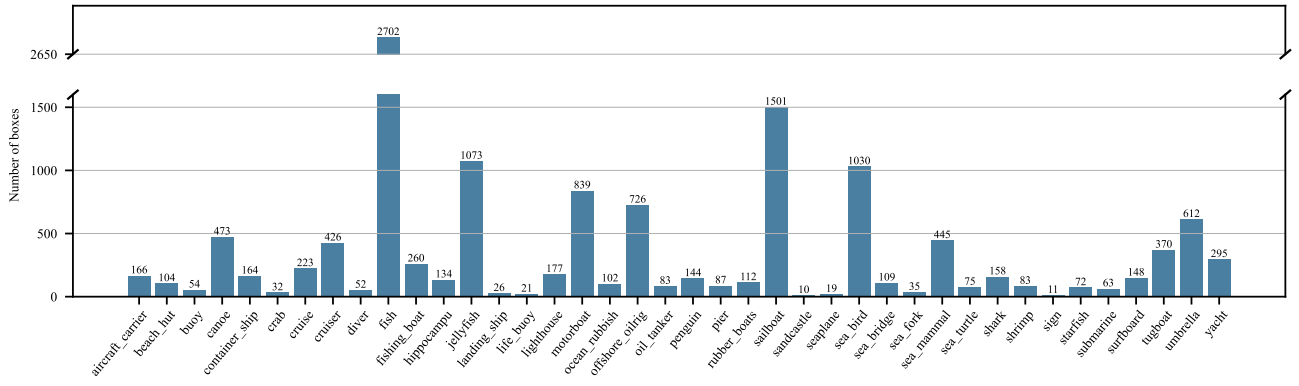


Fig. 3. The number of bounding boxes contained in each of the 40 categories of the Places365_OD data set.

data sets, referred to as the FOSD_OD, Places365_OD and SUN_OD data sets, from three publicly available data sets, including FOSD [35], Places365 [36] and SUN [37]. In total, 60 categories and 10,040 images were selected.

We manually annotated these images in the format of Pascal VOC [38] using the tool that Tzutalin [65] provided. To our knowledge, this work is the first effort made on collecting such a wide range of oceanic scene element images. Regarding the three data sets, each category and the number of bounding boxes contained in this category are shown in Figs. 2, 3 and 4 respectively.

3.1. FOSD_OD

The Flickr Oceanic Scene Dataset (FOSD) [35] contains 45 oceanic scene categories. Each category comprises 100 images. We selected 34 categories from the FOSD. As a result, a total of 3,367 images were selected, which were comprised of the FOSD_OD data set. An example image of each category in the FOSD_OD data set is shown in Fig. 5.

3.2. Places365_OD

The Places365 [36] data set is a large scene recognition data set, which comprises 365 unique scene categories. In

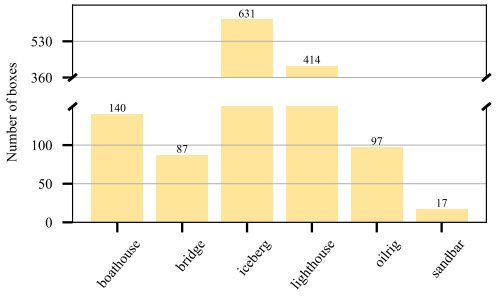


Fig. 4. The number of bounding boxes contained in each of the six categories of the SUN_OD data set.

total, eight million images are included in this data set. We selected 40 oceanic scene categories, containing a total of 5,567 images, from the Places365 data set. These images were comprised of the Places365_OD data set. An example image of each category within this data set is shown in Fig. 6.

3.3. SUN_OD

The SUN [37] data set is a scene recognition data set, which consists of 899 categories and 130,519 images in

Fig. 5. An example image is shown for each of the 34 categories contained in the FOSD_OD data set.

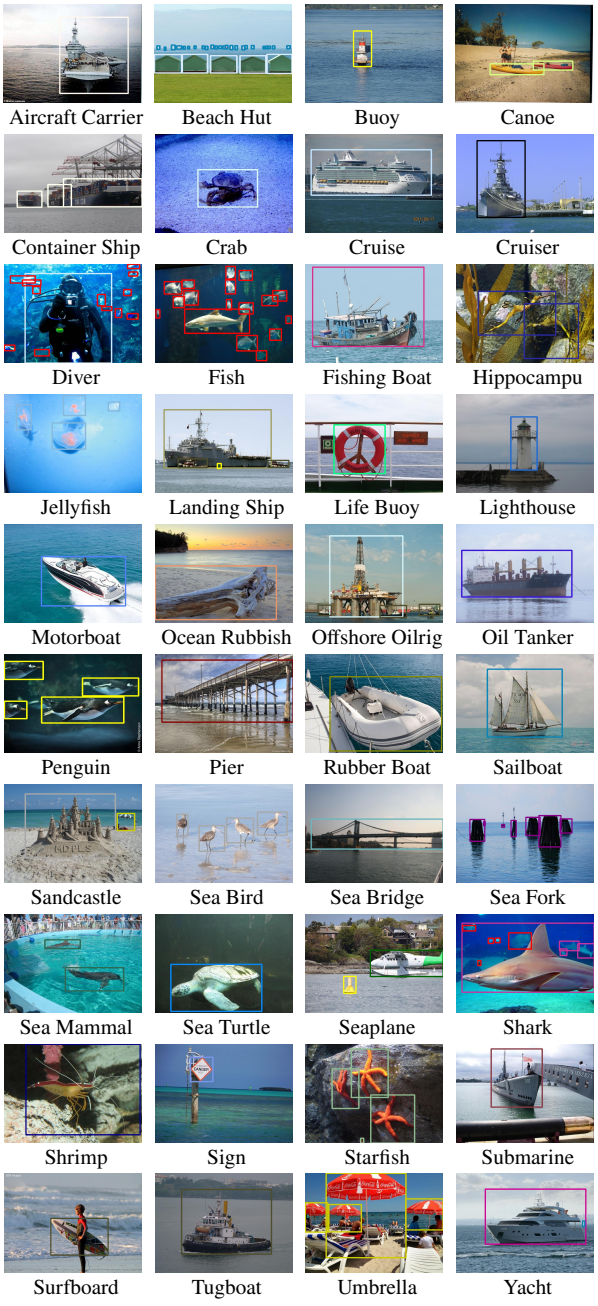


Fig. 6. An example image is shown for each of the 40 categories comprised in the Places365_OD data set.

total. We selected six oceanic scene categories. A total of 1,106 images were included in these categories, which were comprised of the SUN_OD data set. Regarding each category contained in this data set, an example image is shown in Fig. 7.

Fig. 7. An example image is shown for each of the six categories included in the SUN_OD data set.

4. Edge-Guided Oceanic Scene Element Detection Framework

The importance of edges to representation of images has been demonstrated using a user study [30]. Edges have been utilized in different vision tasks, such as image recognition [63], semantic segmentation [34], camouflage detection [33]

and salient object detection [31, 32]. In this study, we are motivated to propose an Edge-Guided Oceanic Scene Element Detection (EG-OSED) framework. This framework contains two main components, i.e., a base object detector and a Multi-scale Edge-Guided Module (MSEGM).

The base object detector can be an existing object detection network, such as Faster R-CNN [17], YOLOv4 [19] and SSD [2]). The MSEGM is described in Algorithm 1. Both the base object detector and the MSEGM are combined. They can be end-to-end trained using a multi-task learning scheme. In this section, we will describe the EG-OSED framework in detail. Given that YOLOv4 [19] is used as the base object detector, the instance of the EG-OSED framework is referred to as EG-OSED-YOLOv4. The architecture of the EG-OSED-YOLOv4 is shown in Fig. 8.

Algorithm 1: Multi-scale Edge-Guided Module

Input: f_i ($i \in \{1,2,3,4,5\}$): the features extracted using the i-th block of the backbone;
foreach i *in* $\{1,2,3,4,5\}$ **do**
  $f'_i \leftarrow Conv_{1\times1}(f_i)$; // *expect* f_1;
  $f_{ij} \leftarrow Split(f_i)$;
  $f''_{i_1} \leftarrow Conv^{d_1}_{3\times3}(f'_{i_1})$;
  $f''_{i_j} \leftarrow Conv^{d_j}_{3\times3}(f''_{i_{j-1}} + f'_{i_j}), j \in \{2,3,4\}$;
  $f_{i\_ms} \leftarrow f'_i + Conv_{1\times1}(Concat(f''_{i_1}, f''_{i_2}, f''_{i_3}, f''_{i_4}))$;
end
$f^e_5 \leftarrow \lambda_5(Upsample(Conv_{1\times1}(f_{5\_ms})))$;
foreach i *in* $\{2,3,4\}$ **do**
  $f^e_i \leftarrow \lambda_i(Upsample(Conv_{1\times1}(f^e_{i+i} + f_{i\_ms})))$;
end
$f^e_1 \leftarrow f^e_2 + \lambda_1(f_{1\_ms})$;
$E \leftarrow \sigma(Conv_{1\times1}(Conv_{3\times3}(f^e_1)))$;
Result: E: the resultant edge map with $1/2$ size of the input image.

4.1. Base Object Detector

Redmon et al. [1] originally proposed the You Only Look Once (YOLO) object detector in 2015. This detector can be end-to-end trained by taking the entire image as the input, partitioning a grid at the output layer, and performing the position and class regression on the grid directly. YOLOv4 [19] inherited this idea but improved the detection performance by exploiting many optimization strategies. For example, the mosaic data augmentation method, the CSPDarknet53 [66] backbone and the CIOU [67] loss function. Considering YOLOv4 has the higher detection accuracy than the prior YOLO variants and the real-time performance, we use YOLOv4 as the base object detector of the proposed EG-OSED framework.

We extract five sets of side feature maps from the backbone at five different blocks respectively. The sizes of these feature maps are equal to $1/2$, $1/4$, $1/8$, $1/16$ and $1/32$ of the size of the input image. All the five sets of features are used for the MSEGM, to derive an edge map of the input image. Among these sets of features, those extracted from the last three blocks are used for object detection. In the neck part of the object detection network, the 1×1 convolutional layer and the 3×3 convolutional layer followed by batch normalization and the Leaky-ReLU activation function are denoted as CBL1 and CBL3, respectively. For simplicity, three and five consecutive CBL1 and/or CBL3 layers are denoted as CBL*5 and CBL*3 respectively.

4.2. Multi-scale Edge-Guided Module

In practice, it is struggling to extract edges from different scales of objects using the features computed at a single scale. Multi-scale image representation has been applied to many computer vision tasks. In this study, we design the Multi-scale Edge-Guided Module (MSEGM) by exploiting the features extracted at different scales for the sake of performing edge extraction. The MSEGM consists of five Scale-Correlated Pyramid Convolution (SCPC) [68] blocks.

Given that the features extracted using the i-th block of the backbone is denoted as f_i ($i \in \{1,2,3,4,5\}$), f_i ($i \in \{2,3,4,5\}$) is first processed using a 1×1 convolution. This convolution is used to change the number of channels of the input features to $1/4$ of the original number of channels, which can be expressed as

$$f'_i = Conv_{1\times1}(f_i), i \in \{2,3,4,5\}, \tag{1}$$

where $Conv_{1\times1}(\cdot)$ is a 1×1 convolution. It should be noted that $f'_1 = f_1$ and the number of reduced channels should not be less than the number of channels of f'_1.

Then f'_i is fed into an SCPC block. Here, the features are divided into four groups along the channel dimension, which can be expressed as

$$f'_{i_1}, f'_{i_2}, f'_{i_3}, f'_{i_4} = Split(f'_i), \tag{2}$$

where $Split(\cdot)$ means the average split operation. Different groups of features are processed using the dilated convolutions with different dilation rates. It can be expressed as

$$f''_{i_1} = Conv^{d_1}_{3\times3}(f'_{i_1}), \tag{3}$$

$$f''_{i_j} = Conv^{d_j}_{3\times3}(f''_{i_{j-1}} + f'_{i_j}), j \in \{2,3,4\}, \tag{4}$$

where $Conv^{d_j}_{3\times3}$ is a 3×3 dilated convolution with a dilated rate d_j. In this study, we set the dilation rate d to $\{1,2,4,8\}$ in the first three SCPC blocks and $\{1,2,3,4\}$ in the last two SCPC blocks.

All the features processed are concatenated and sent to a 1×1 convolution. The resultant features are added with the features processed by the first 1×1 convolution, which can be formulated as

$$f_{i\_ms} = f'_i + Conv_{1\times1}(Concat(f''_{i_1}, f''_{i_2}, f''_{i_3}, f''_{i_4})), \tag{5}$$

where $Concat(\cdot)$ means the concatenation operation.

The application of the SCPC block can be expressed as

$$f_{i\_ms} = SCPC(f'_i), \tag{6}$$

where $SCPC(\cdot)$ denotes the SCPC block. In terms of the five sets of features f'_1, f'_2, f'_3, f'_4 and f'_5, the five SCPC blocks produce five sets of features f_{1_ms}, f_{2_ms}, f_{3_ms}, f_{4_ms} and f_{5_ms}, respectively.

Furthermore, a Weighted Feature Fusion Module (WFFM) is used to fuse the multi-scale features generated by the five SCPC blocks using different weights. All the features f_{i_ms} ($i \in \{1,2,3,4,5\}$) are fed into the WFFM. The number of channels of the features f_{i_ms} ($i \in \{2,3,4,5\}$) extracted at

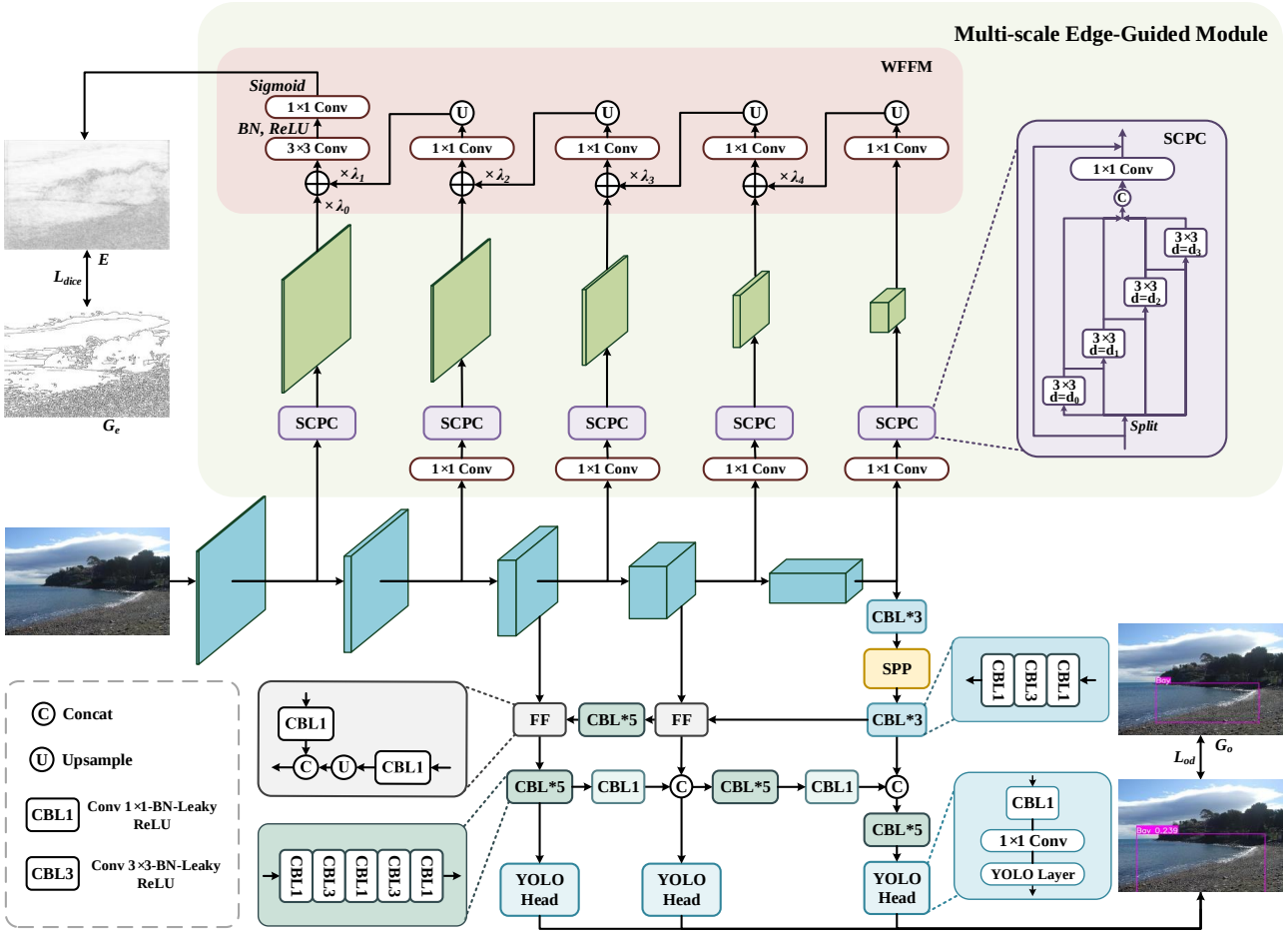


Fig. 8. The architecture of the proposed EG-OSED-YOLOv4 method, in which YOLOv4 [19] is used as the base object detector.

the last four blocks are converted to the number of channels of f_{i-1_ms} using a 1×1 convolution, while each feature map extracted at these blocks is upsampled to the size of an f_{i-1_ms} feature map.

Next, the up-sampled feature set f_i^e $(i \in \{2,3,4,5\})$ is weighted using a factor λ_i. Specifically, the weighted features and the features extracted from the previous SCPC block are fused by the pixel-wise addition operation before the features extracted from the previous SCPC block are sent to a convolutional layer. This process can be formulated as

$$f_5^e = \lambda_5(Upsample(Conv_{1\times1}(f_{5\_ms}))), \tag{7}$$

$$f_i^e = \lambda_i(Upsample(Conv_{1\times1}(f_{i+i}^e + f_{i\_ms}))), i \in \{2,3,4\}. \tag{8}$$

The feature set f_{1_ms} extracted from the first SCPC block is weighted using λ_1 and fused with f_2^e, which can be formulated as

$$f_1^e = f_2^e + \lambda_1(f_{1\_ms}). \tag{9}$$

Finally, a 3×3 convolution (followed by batch normalization and ReLU), a 1×1 convolution and the Sigmoid function are applied to f_1^e. As a result, an edge map with 1/2 size of the input image is produced. The edge extraction operation can be expressed as

$$E = \sigma(Conv_{1\times1}(Conv_{3\times3}(f_1^e))), \tag{10}$$

where E is the resultant edge map, $\sigma(\cdot)$ means the Sigmoid function and $Conv_{3\times3}(\cdot)$ and $Conv_{1\times1}(\cdot)$ denote the 3×3 and 1×1 convolutional layers, respectively.

4.3. Multi-task Loss Function

Given a set of training images, it has been demonstrated that multi-task learning is capable of learning shared representations from these images for different tasks, while it is likely to produce the better result for these tasks [69]. To end-to-end train both the base object detector and the MSEGM, we are inspired to utilize a multi-task learning scheme. We built a multi-task loss function on top of the object detection loss function L_{od} and the edge extraction loss function L_{dice}.

In terms of the object detection loss function L_{od}, it comprises three terms. The bounding box regression loss L_{box} is first computed using both the Binary Cross-Entropy Loss (BCELoss) and the Mean Squared Error Loss (MSELoss) functions, rather than the CIOULoss [67] that YOLOv4 used. The reason for this choice is due to that the better detection performance was obtained using BCELoss and

MSELoss in our experiments. Then we use BCELloss to calculate the confidence loss L_{obj} and the class loss L_{cls}. Given the three loss terms, the object detection loss function can be computed as

$$L_{od} = L_{box} + L_{obj} + L_{cls}. \tag{11}$$

Regarding the edge extraction loss, the Dice Loss [70] L_{dice} is utilized for edge supervision. This loss function is defined as

$$L_{dice} = \frac{2 \mid G \mid \cap \mid P \mid}{\mid G \mid + \mid P \mid}, \tag{12}$$

where G and P represent the ground-truth and extracted edge maps, while $\mid G \mid$ and $\mid P \mid$ denote the number of pixels in G and P respectively.

Finally, the multi-task loss for the EG-OSED framework is defined as

$$L = L_{od} + L_{dice}. \tag{13}$$

Due to this loss function, the EG-OSED framework can be end-to-end trained for both the object detection and the edge extraction tasks simultaneously.

5. Experimental Setup

We will introduce the experimental setup in this section, including the evaluation criterion used in our experiment and the implementation details of our method.

5.1. Evaluation Criterion

As a common practice, the Mean Average Precision (mAP) has been used to measure the performance of object detection methods. We also used mAP as a metric to evaluate the performance of oceanic scene element detectors. The mAP value was computed using the method which had been used to calculate the mAP on the PASCAL VOC [38] segmentation data set. Specifically, the area under the Precision-Recall curve was computed to obtain the Average Precision (AP) value. Given that a set of AP values were computed for n categories, the mAP can be calculated as:

$$mAP = \frac{1}{n} \sum_{i=1}^{n} AP_i. \tag{14}$$

5.2. Implementation Details

All networks were implemented using the PyTorch library. Experiments were performed on a GeForce RTX 3080 GPU. Given that YOLOv4 [19] was used as the base object detector, the size of input images was set to 608×608 pixels. Each network was trained for 300 epochs. The mini-batch size and subdivision were set to 64 and 8 respectively. We set the initial learning rate and the warm-up step to 0.01 and 1000 respectively. The learning rate was divided by 10 after 80% and 90% of the total iterations had been performed.

The total iterations were calculated separately for different data sets as follows:

$$sub\_batch = \frac{batchsize}{subdivision}, \tag{15}$$

$$iterations = \lfloor \frac{\lfloor \frac{train\_num}{sub\_batch} \rfloor}{subdivision} \rfloor, \tag{16}$$

Table 2
Comparison between the detection results (mAP (%)) derived using the proposed EG-OSED-YOLOv4 and YOLOv4 [19] on the FOSD_OD, Places365_OD and SUN_OD data sets.

Method	FOSD_OD	Places365_OD	SUN_OD
YOLOv4	61.56	**39.52**	67.57
EG-OSED-YOLOv4	**63.91**	39.11	**70.50**

where $batchsize$ was the number of images contained in a mini-batch, $subdivision$ meant that the training images in a mini-batch were fed into the network in $subdivision$ times (i.e., each time the network was fed with sub_batch images), $train\_num$ was the number of training images. Other hyper-parameter settings remained the same as those of YOLOv4.

Specifically, we used the ResNet-50 [14] model pre-trained on the Places365 [36] data set as the backbone in all the YOLOv4-based experiments. Besides, we randomly selected 70% of images from each category of a data set as the training set while the remaining images were utilized as the testing set. The edge maps extracted using the Canny [71] edge detector are used as the ground-truth data to guide the backbone toward learning edge characteristics.

6. Experimental Results

In this study, we conducted a series of experiments using the setup introduced in Section 5 on the three data sets that we collected. The results will be reported in this section.

6.1. Using EG-OSED-YOLOv4

We investigated the performance of the proposed EG-OSED framework along with the YOLOv4 [19] detector, i.e., EG-OSED-YOLOv4. We compared the results derived using the EG-OSED-YOLOv4 and YOLOv4 detectors quantitatively and qualitatively.

6.1.1. Quantitative Evaluation

Table 2 shows the results produced by the proposed EG-OSED-YOLOv4 method and YOLOv4 [19] on the three data sets. It can be seen that our method outperformed the YOLOv4 detector on the FOSD_OD and SUN_OD data sets and achieved the comparable result on the Place365_OD data set. This finding demonstrated that the proposed MSEGM boosted the performance of the YOLOv4 detector when they had been trained together using the multi-task learning scheme. Specifically, the mAP values derived using our EG-OSED-YOLOv4 were 2.35% and 2.93% higher on the FOSD_OD and SUN_OD data sets, respectively, while the value obtained using our method was only 0.41% lower on the Places365_OD data set, compared with YOLOv4.

6.1.2. Qualitative Evaluation

In terms of each of the three data sets, three sets of results obtained using EG-OSED-YOLOv4 and YOLOv4 [19] are visualized in Fig. 9. As can be seen, our EG-OSED-YOLOv4 performed the oceanic scene element detection task better than YOLOv4. According to the second sets of results derived on the FOSD_OD and SUN_OD data sets,

for example, it can be observed that EG-OSED-YOLOv4 was able to accurately detect small objects (e.g., sea arches and lighthouses). As can be seen from the first set of results obtained on the Places365_OD data set, the EG-OSED-YOLOv4 was capable of detecting the objects (e.g., wood) which have the similar color to the background. From the second set of results obtained on the Places365_OD data set, we can find that our EG-OSED-YOLOv4 generated the more accurate object classification.

Furthermore, three sets of feature maps extracted using the backbones of EG-OSED-YOLOv4 and YOLOv4 are visualized in Fig. 10. We can find that the feature maps extracted using the backbone of EG-OSED-YOLOv4 contain the more and sharper object edge characteristics than those extracted using the backbone of YOLOv4, which are important to oceanic scene element detection. This finding holds true across all the three data sets.

The above results show that the use of the proposed MSEGM, which was designed to guide the backbone of an existing object detector toward learning edge characteristics, improved the performance of the detector, in particular, in classification and localization of objects.

6.2. Ablation Experiments

To examine the impact of different components of the EG-OSED framework on the oceanic scene element detection task, ablation experiments were also conducted on the three data sets. We will report the results in this subsection.

6.2.1. Impact of Different Edge Utilization Strategies

To exploit the edge cue for the purpose of boosting the performance of oceanic scene element detection, we investigated two different strategies: applying the features extracted from edge maps to the neck of the detection network and guiding the backbone of the detection network toward learning edge characteristics.

Regarding the first strategy, we constructed a two-branch backbone network in which an image and the corresponding edge map were fed into the two branches respectively. Two sets of features, extracted from both the branches respectively, were fused. The resultant features were sent to the neck of the object detection network.

Two different approaches were used to extract features from the edge map. (1) The two branches shared the weights of the ResNet-50 [14] trained on the Places365 [36] data set. In other words, features were extracted from the image and edge maps using the same sets of weights. (2) The weights of the ResNet-101 pre-trained on the Im4Sketch [72] data set were used for the branch which took the edge map as the input. The two approaches are denoted as Edge_Feat_P and Edge_Feat_S respectively. In addition, three methods were utilized for feature fusion, including addition, concatenation and the cross-attention [73] mechanism.

In terms of the second strategy, we used the MSEGM (see Section 4.2) to guide the backbone of the object detection network toward learning edge characteristics, implemented by performing multi-task learning. For simplicity, we used MSEGM to denote the second strategy. In essence, this method is the proposed EG-OSED-YOLOv4.

Table 3
Comparison of the detection results (mAP (%)) derived using YOLOv4 [19] along with different edge utilization methods on the FOSD_OD, Places365_OD and SUN_OD data sets. Here, "None" means YOLOv4 which does not use edges or the fusion operation, while "Add", "Cat" and "CA" stand for the addition, concatenation and the cross-attention mechanism operations respectively.

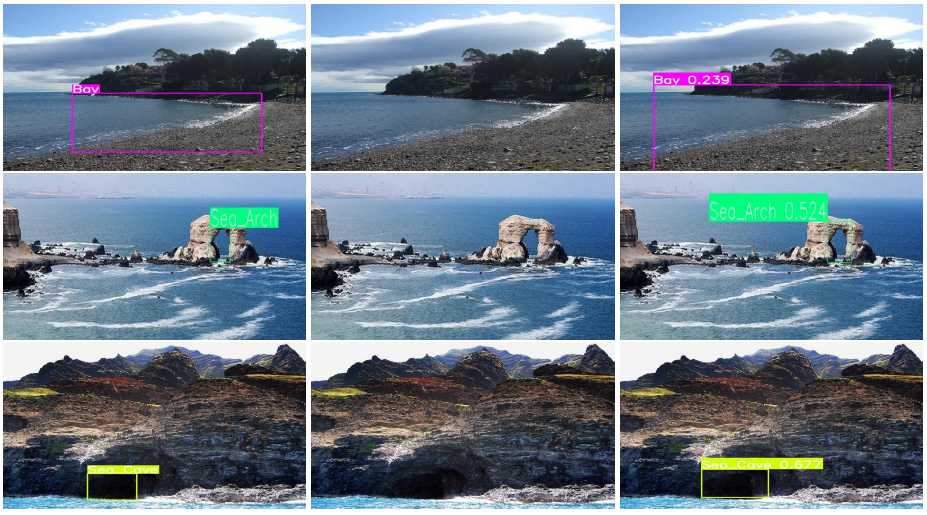
Edge Utilization Method	Fusion Method	FOSD_OD	Places365_OD	SUN_OD
None	None	61.56	39.52	67.57
Edge_Feat_P	Add	60.42	37.70	66.26
	Cat	61.52	38.47	65.72
	CA	57.35	35.42	61.00
Edge_Feat_S	Add	60.06	37.93	64.52
	Cat	61.23	36.68	62.41
	CA	57.56	35.39	62.25
MSEGM	None	**63.91**	**39.11**	**70.50**

The results obtained using the three methods are shown in Table 3. Compared with the method which did not use edges (i.e., YOLOv4 [19]), it can be seen that the two methods which applied the features extracted from edge maps to the neck of the detection network did not produce the better results. However, the proposed method, which used the MSEGM to guide the backbone toward learning edge characteristics, outperformed its counterparts on the FOSD_OD and SUN_OD data sets. Compared with the YOLOv4 [19] detector, our method improved the mAP value by 2.35% and 2.93% on the FOSD_OD and SUN_OD data sets, respectively, while slightly reduced this value by 0.41% on the Places365_OD data set.

6.2.2. Impact of Different Edge Feature Fusion Methods

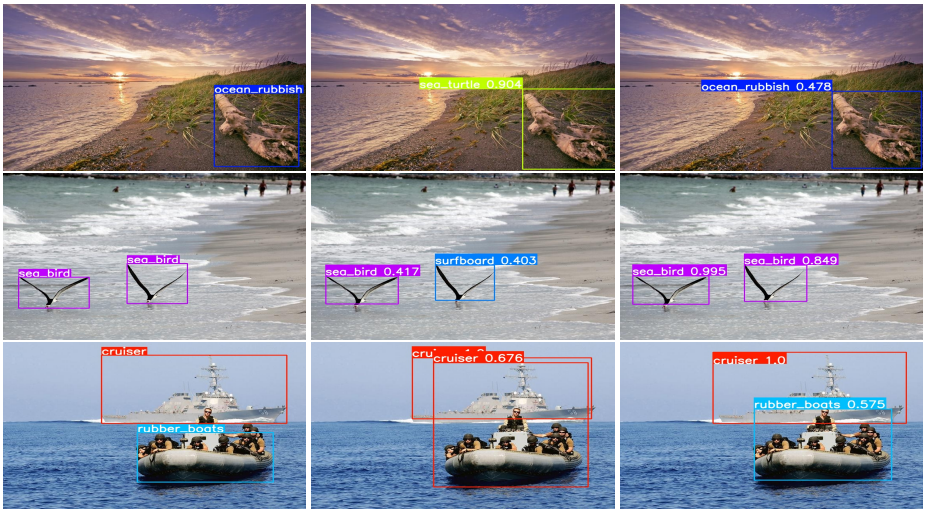
With regard to the fusion of the features extracted using five SCPC blocks (see Fig. 8), two different strategies were tested, including the cascade fusion and the direct upsampling fusion. For the cascade fusion strategy, we progressively fused the features with those generated by the previous SCPC block. Regarding the direct upsampling fusion strategy (see Section 4), the feature maps extracted using the following SCPC blocks were upsampled to the size of the feature maps produced by the first SCPC block. All the five sets of features were fused (see Fig. 8). In terms of both the strategies, the concatenation and pixel-wise addition operations were used for feature fusion.

Regarding λ_i ($i \in \{1, 2, 3, 4, 5\}$) values, we set all of these to 1.0. We refer to this setting as "1.0-1.0". Since shallow features encode more edge details and it is likely that they are more beneficial to edge extraction [74, 75], we set the larger weights for the shallower features, to boost the performance of edge extraction. To be exact, we set the λ_i values to $\{1.2, 1.1, 1.0, 0.9, 0.8\}$. This setting is referred to as "1.2-0.8". In addition, we conducted an experiment by setting the λ_i values as a set of learnable parameters, which were initialized to 1.0. We refer to this setting as "Learnable". The results produced by the proposed EG-OSED-YOLO4



Ground-truth YOLOv4 EG-OSED-YOLOv4

(a) FOSD_OD



Ground-truth YOLOv4 EG-OSED-YOLOv4

(b) Places365_OD

Ground-truth YOLOv4 EG-OSED-YOLOv4

(c) SUN_OD

Fig. 9. Visualization of the ground-truth data and the detection results derived using the proposed EG-OSED-YOLOv4 and YOLOv4 [19] on the FOSD_OD, Places365_OD and SUN_OD data sets. (Zoom in for better view).

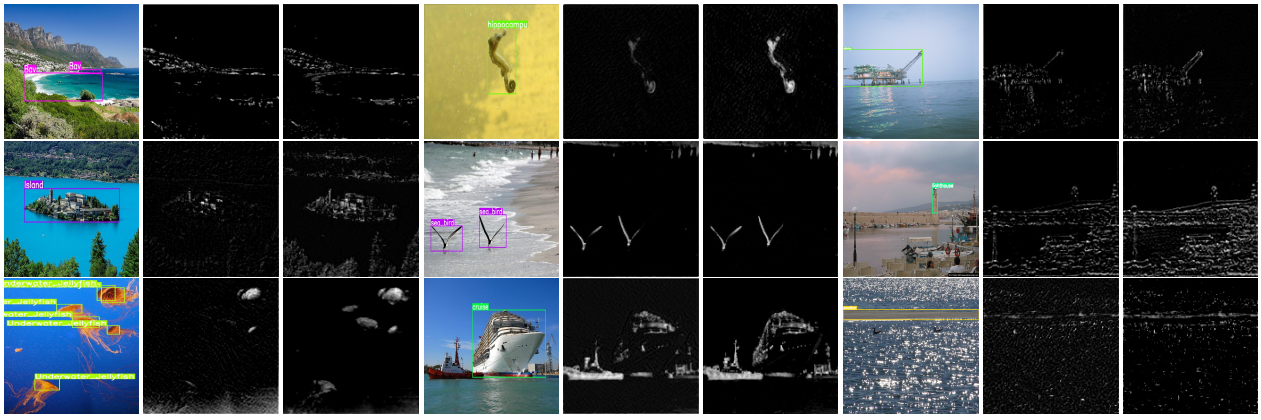


Fig. 10. Visualization of the ground-truth data and the two feature maps extracted using the backbones of the proposed EG-OSED-YOLOv4 and YOLOv4 [19], respectively, on the FOSD_OD, Places365_OD and SUN_OD data sets. (Zoom in for better view).

Table 4
Comparison of the detection results (mAP (%)) derived using EG-OSED-YOLO4 along with different edge feature fusion methods on the FOSD_OD, Places365_OD and SUN_OD data sets. Here, "None" denotes the YOLOv4 object detector, which does not use edges. "Cascade_Concat" and "Cascade_Add" represent the cascade concatenation and cascade pixel-wise addition fusion methods, respectively, while "Direct_Concat" and "Direct_Add" denote the direct upsampling followed by the concatenation and pixel-wise addition operations, respectively.

Edge Feature Fusion Method	FOSD_OD	Places365_OD	SUN_OD
None	61.56	39.52	67.57
Direct_Concat	61.11	39.55	67.88
Direct_Add_1.0-1.0	61.43	38.80	67.97
Direct_Add_1.2-0.8	62.42	**40.34**	69.63
Direct_Add_Learnable	**64.16**	38.31	70.34
Cascade_Concat	61.23	39.19	67.91
Cascade_Add_1.0-1.0	60.44	38.45	67.29
Cascade_Add_1.2-0.8	59.85	40.21	66.28
Cascade_Add_Learnable	63.91	39.11	**70.50**

method together with different fusion methods are shown in Table 4.

It can be seen that the "1.0-1.0" setting nearly did not gain the improvement across the three data sets in terms of both the fusion strategies. Given that the cascade fusion strategy was used, the large weight used for the shallow features took effect on the Places365_OD data set. This was also the case for the direct upsampling fusion strategy across all the three data sets. The direct upsampling fusion strategy along with the pixel-wise addition ("1.2-0.8") method achieved improvements on the three data sets. For both the strategies, however, the largest improvements were achieved on the FOSD_OD and the SUN_OD data sets, respectively, when the λ_i values were set as learnable parameters. With the cascade fusion strategy, there was only a slight decrease in the mAP value on the Places365_OD data set when the λ_i values were set as learnable parameters.

Table 5
Comparison of different instances of the proposed EG-OSED framework against the corresponding base object detectors in terms of the mAP (%) value obtained on the FOSD_OD, Places365_OD and SUN_OD data sets. Note that the *confidence_thresh* was set to 0.001 for all the detectors here, while it was set to 0.1 by default (e.g., Tables 2 to 4).

Object Detector	FOSD_OD	Places365_OD	SUN_OD
SSD-512	**68.26**	49.73	70.88
EG-OSED-SSD-512	68.19	**51.85**	**72.12**
YOLOv3	64.80	46.50	69.60
EG-OSED-YOLOv3	**66.50**	**46.60**	**72.20**
YOLOv4	65.50	**42.35**	70.65
EG-OSED-YOLOv4	**67.51**	42.25	**73.14**
YOLOv5	63.20	43.20	**69.50**
EG-OSED-YOLOv5	**64.30**	**44.40**	68.30
YOLOv7	64.10	50.80	**68.40**
EG-OSED-YOLOv7	**68.20**	**52.10**	65.20

6.2.3. Impact of Different Object Detectors

To investigate the generalization ability of the MSEGM, we combined it with four deep learning-based object detectors, including SSD-512 [2], YOLOv3 [18], YOLOv5 [25] and YOLOv7 [76]. As a result, four additional instances of the EG-OSED framework were derived, which are referred to as EG-OSED-SSD-512, EG-OSED-YOLOv3, EG-OSED-YOLOv5 and EG-OSED-YOLOv7, respectively.

We compared the instances of the EG-OSED framework against the corresponding object detectors in terms of the mAP value obtained on the three data sets. All the detectors were trained using the three data sets from scratch. Considering that YOLOv3 cannot work properly when the *confidence_thresh* used for the mAP calculation is greater than 0.001[1], we set this parameter to 0.001 for the experiment conducted in this subsubsection. It should be noted that the *confidence_thresh* was set to 0.1 by default in this study.

The results are reported in Table 5. As can be seen, EG-OSED-YOLOv3 achieved the better performance than

[1]https://github.com/ultralytics/yolov3

the corresponding base object detectors YOLOv3 on all the three data sets. This is also the case for EG-OSED-SSD-512, EG-OSED-YOLOv4, EG-OSED-YOLOv5 and EG-OSED-YOLOv7 except that they performed worse than their counterpart base object detectors on one data set. In this context, the generalization ability of the proposed MSEGM has been demonstrated. Furthermore, we visualize the results derived using four instances of the EG-OSED framework and the corresponding base object detectors on the FOSD_OD, Places365_OD and SUN_OD data sets in Figs. 11, 12, 13 and 14, respectively. Again, the generalization ability of the MSEGM can be intuitively observed.

6.3. Failure Cases

Since the edge map generated by the Canny [71] algorithm contains the edges of both the object and background, the MSEGM will guide the backbone towards learning all the edge characteristics within an image. When the background is complicated, the edges of the object may be mixed up by our method with those of the background. As a result, the object cannot be accurately localized or is even missed. Three cases of the failed detection produced by the proposed EG-OSED-YOLOv4 on each of the FOSD_OD, Places365_OD and SUN_OD data sets are shown in Fig. 15.

7. Conclusion

To perform the oceanic scene element detection task, we first selected and annotated three oceanic scene element data sets, including FOSD_OD, Places365_OD and SUN_OD. In total, 10,040 images and 60 categories were included. Since oceanic scene elements normally manifest large-scale complicated structures, the edge cue is particular important to the representation of these elements. Then we introduced a generic Multi-scale Edge-Guided Module (MSEGM), which can be inserted into an object detection network, to guide the backbone of this network toward learning edge characteristics. An Edge-Guided Oceanic Scene Element Detection (EG-OSED) framework was further adopted, which comprised the MSEGM and a base object detector. The two networks can be end-to-end trained using a multi-task learning scheme. We carried out a series of experiments on the three data sets. The results showed that the proposed EG-OSED framework normally outperformed the counterpart detector which did not use the MSEGM. We believe that these promising results should be due to the importance of the edge cue to the representation of oceanic scene elements.

Considering the MSEGM guides the backbone towards learning all edge characteristics, the edges extracted from the background may interfere the detection of oceanic scene elements when the background is complicated. To reduce the interference of the background, attention modules, which are able to enable the network to focus on the object, can be incorporated into the proposed method. However, our key contribution should be the introduction of the oceanic scene element detection task. This work may encourage more studies in this direction.

CRediT authorship contribution statement

Keke Xiang: Methodology, Validation, Data curation, Writing - Original Draft, Visualization, Formal analysis. **Xingshuai Dong:** Data curation. **Weibo Wang:** Methodology. **Xinghui Dong:** Conceptualization, Data curation, Methodology, Supervision, Writing - Review & Editing, Resources, Funding acquisition.

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.

Acknowledgements

The work described in this paper was in part supported by the National Natural Science Foundation of China (NSFC) (No. 42176196) and was in part supported by the Young Taishan Scholars Program (No. tsqn201909060).

References

[1] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[2] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.

[3] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1440–1448.

[4] J. Su, Y. Su, Y. Zhang, W. Yang, H. Huang, Q. Wu, Epnet: Power lines foreign object detection with edge proposal network and data composition, Knowledge-Based Systems 249 (2022) 108857.

[5] Y. Hu, J. Zhan, G. Zhou, A. Chen, W. Cai, K. Guo, Y. Hu, L. Li, Fast forest fire smoke detection using mvmnet, Knowledge-Based Systems 241 (2022) 108219.

[6] B. Chaitra, P. B. Reddy, An approach for copy-move image multiple forgery detection based on an optimized pre-trained deep learning model, Knowledge-Based Systems 269 (2023) 110508.

[7] K. Chen, J. Liu, H. Zhang, Igt: Illumination-guided rgb-t object detection with transformers, Knowledge-Based Systems 268 (2023) 110423.

[8] G. Li, Y. Yang, X. Qu, D. Cao, K. Li, A deep learning based image enhancement approach for autonomous driving at night, Knowledge-Based Systems 213 (2021) 106617.

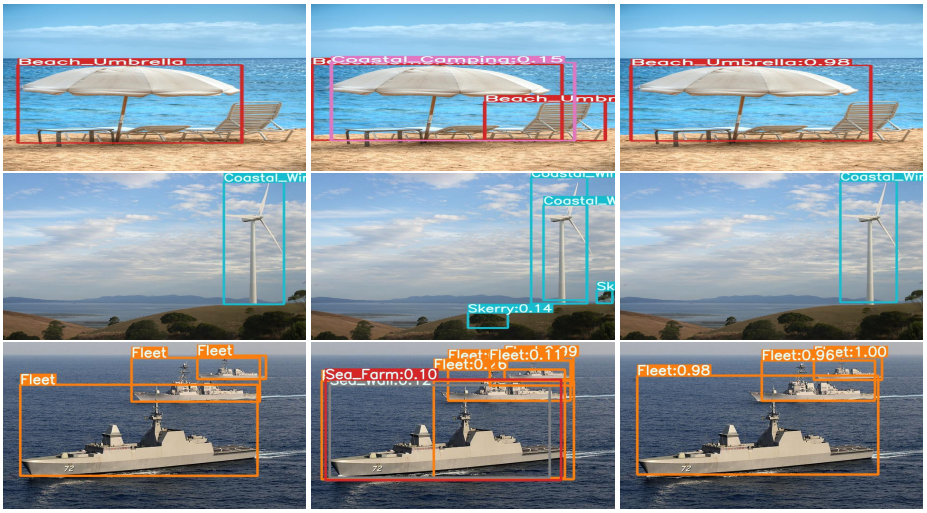
[9] M.-C. Kong, M.-I. Roh, K.-S. Kim, J. Lee, J. Kim, G. Lee, Object detection method for ship safety plans using deep learning, Ocean Engineering 246 (2022) 110587.

[10] H.-S. Jin, H. Cho, H. Jiafeng, J.-H. Lee, M.-J. Kim, S.-K. Jeong, D.-H. Ji, K. Joo, D. Jung, H.-S. Choi, Hovering control of uuv through underwater object detection based on deep learning, Ocean Engineering 253 (2022) 111321.

[11] C. P. Schwegmann, W. Kleynhans, B. P. Salmon, Ship detection in south african oceans using sar, cfar and a haar-like feature classifier, in: 2014 IEEE Geoscience and Remote Sensing Symposium, IEEE, 2014, pp. 557–560.

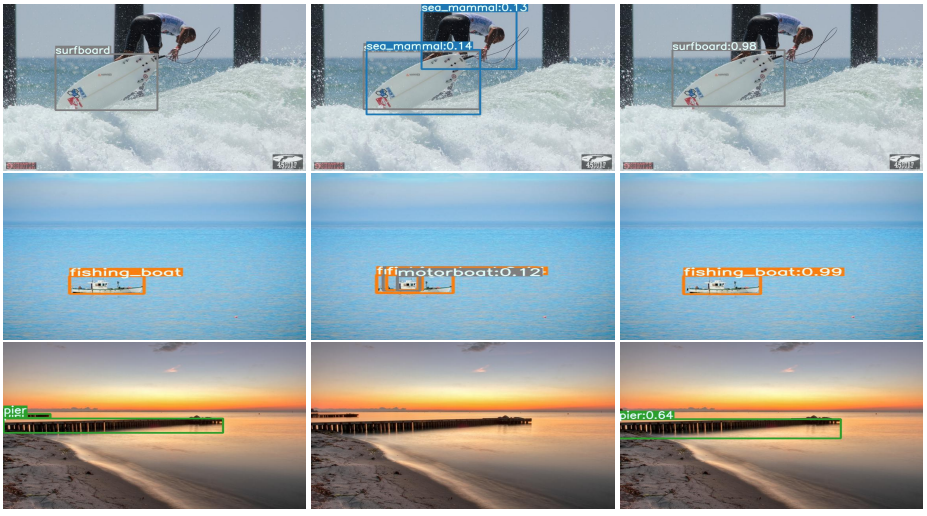
[12] B. Alsahwa, F. Maussang, R. Garello, A. Chevallier, Marine life airborne observation using hog and svm classifier, in: OCEANS 2016 MTS/IEEE Monterey, IEEE, 2016, pp. 1–5.

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25 (2012).



Ground-truth SSD EG-OSED-SSD

(a) FOSD_OD



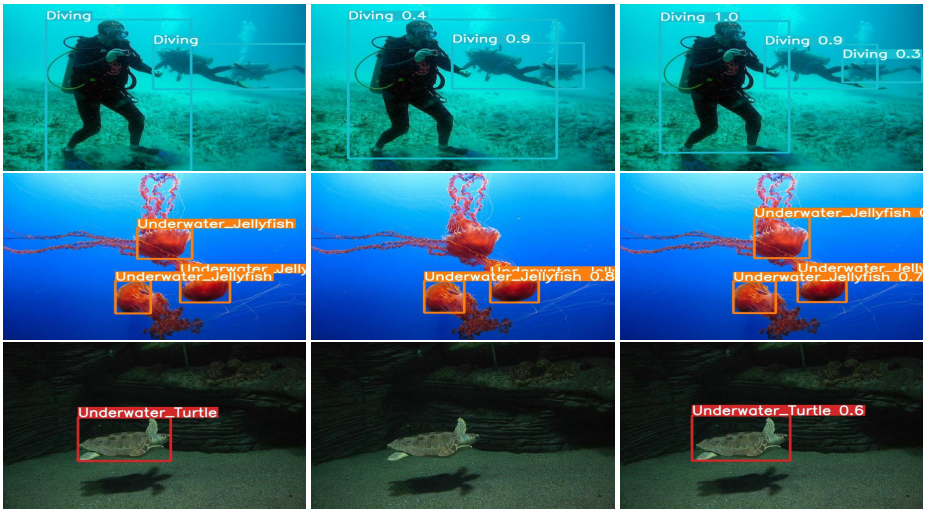
Ground-truth SSD EG-OSED-SSD

(b) Places365_OD

Ground-truth SSD EG-OSED-SSD

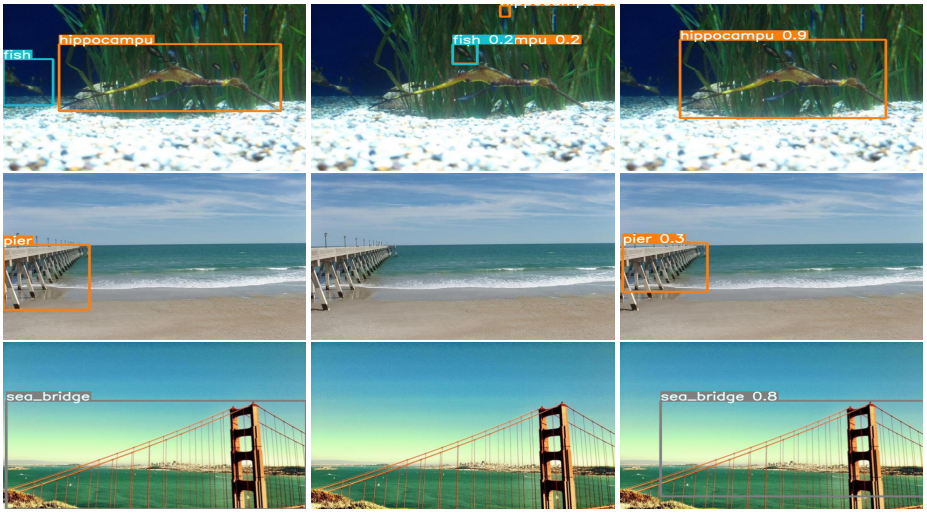
(c) SUN_OD

Fig. 11. Visualization of the ground-truth data and the detection results derived using EG-OSED-SSD and SSD [2] on the FOSD_OD, Places365_OD and SUN_OD data sets. (Zoom in for better view).



Ground-truth YOLOv3 EG-OSED-YOLOv3

(a) FOSD_OD



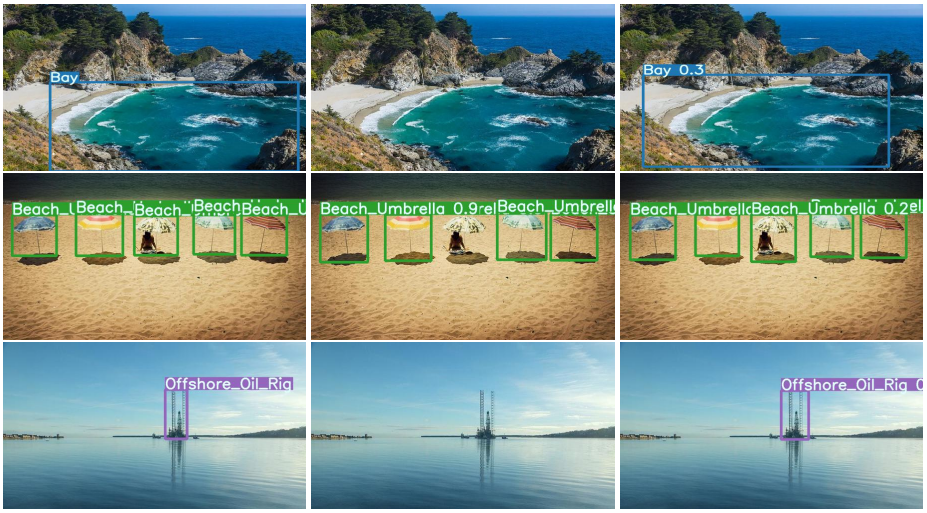
Ground-truth YOLOv3 EG-OSED-YOLOv3

(b) Places365_OD

Ground-truth YOLOv3 EG-OSED-YOLOv3

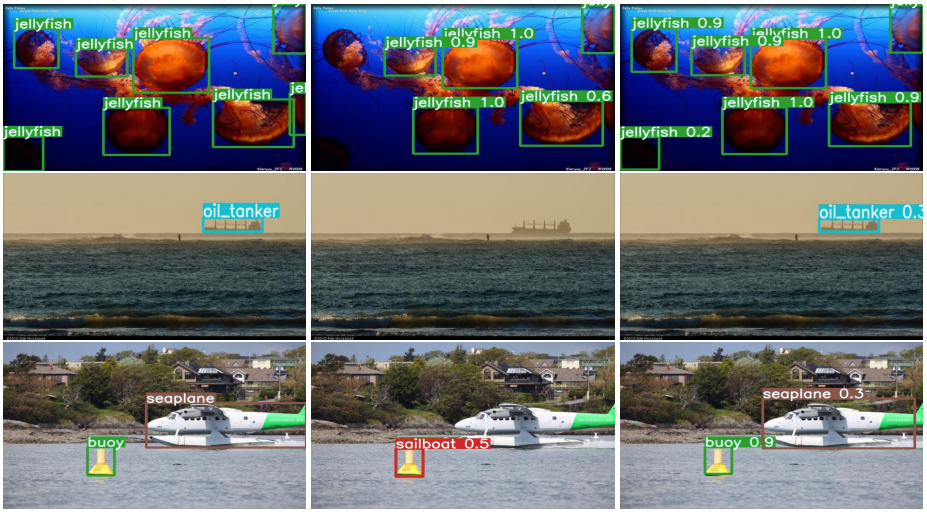
(c) SUN_OD

Fig. 12. Visualization of the ground-truth data and the detection results derived using EG-OSED-YOLOv3 and YOLOv3 [18] on the FOSD_OD, Places365_OD and SUN_OD data sets. (Zoom in for better view).



Ground-truth YOLOv5 EG-OSED-YOLOv5

(a) FOSD_OD



Ground-truth YOLOv5 EG-OSED-YOLOv5

(b) Places365_OD

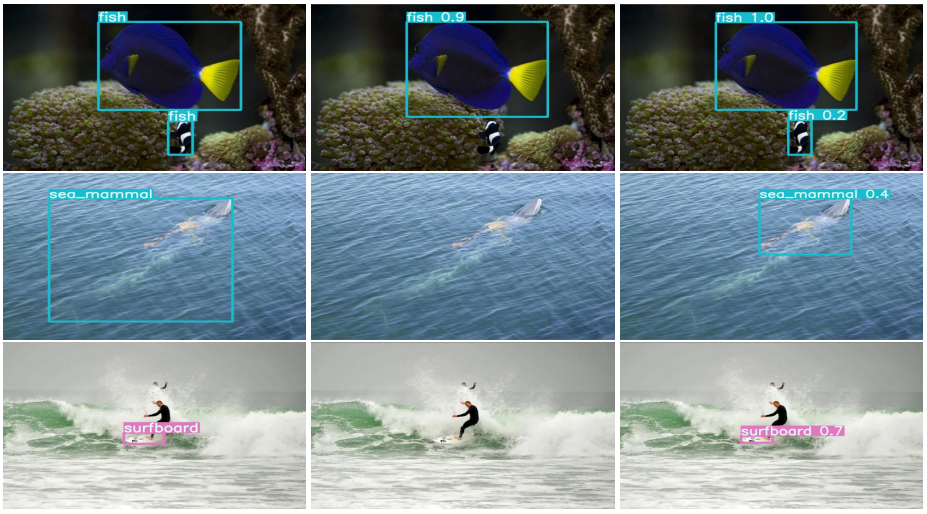
Ground-truth YOLOv5 EG-OSED-YOLOv5

(c) SUN_OD

Fig. 13. Visualization of the ground-truth data and the detection results derived using EG-OSED-YOLOv5 and YOLOv5 [25] on the FOSD_OD, Places365_OD and SUN_OD data sets. (Zoom in for better view).

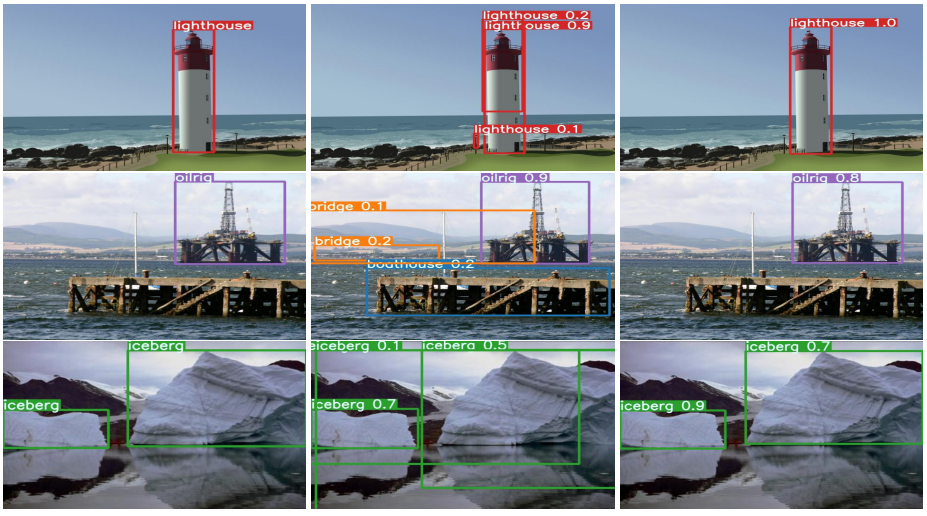
Ground-truth YOLOv7 EG-OSED-YOLOv7

(a) FOSD_OD



Ground-truth YOLOv7 EG-OSED-YOLOv7

(b) Places365_OD



Ground-truth YOLOv7 EG-OSED-YOLOv7

(c) SUN_OD

Fig. 14. Visualization of the ground-truth data and the detection results derived using EG-OSED-YOLOv7 and YOLOv7 [76] on the FOSD_OD, Places365_OD and SUN_OD data sets. (Zoom in for better view).

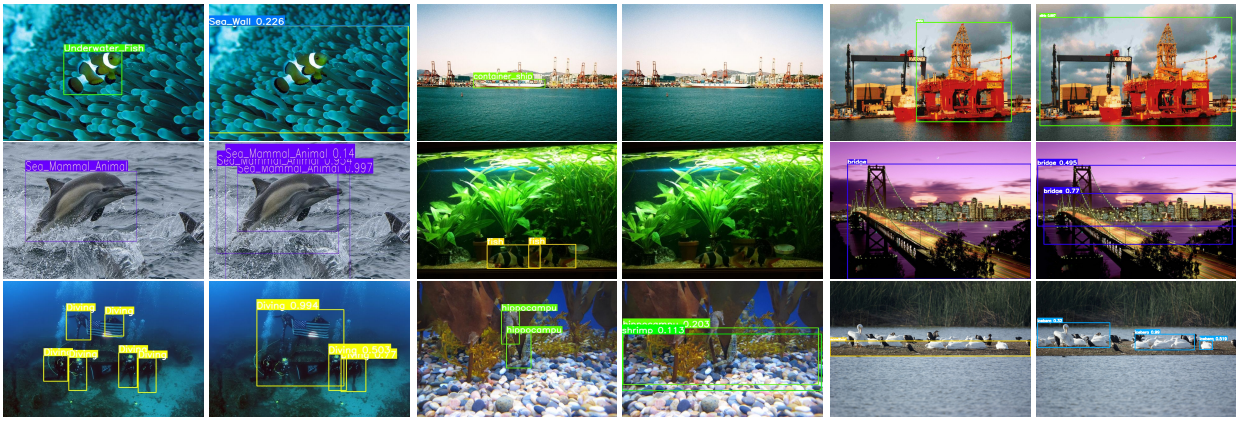


Fig. 15. Visualization of the ground-truth data and the failed detection results produced by the proposed EG-OSED-YOLOv4 on the FOSD_OD, Places365_OD and SUN_OD data sets. (Zoom in for better view).

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

[15] H. C. Altunay, Z. Albayrak, A. N. Özalp, M. Çakmak, Analysis of anomaly detection approaches performed through deep learning methods in scada systems, in: 2021 3rd International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), IEEE, 2021, pp. 1–6.

[16] H. C. Altunay, Z. Albayrak, A hybrid cnn+lstm-based intrusion detection system for industrial iot networks, Engineering Science and Technology, an International Journal 38 (2023) 101322.

[17] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, Advances in neural information processing systems 28 (2015).

[18] J. Redmon, A. Farhadi, Yolov3: An incremental improvement, arXiv preprint arXiv:1804.02767 (2018).

[19] A. Bochkovskiy, C.-Y. Wang, H.-Y. M. Liao, Yolov4: Optimal speed and accuracy of object detection, arXiv preprint arXiv:2004.10934 (2020).

[20] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2014, pp. 580–587.

[21] Y. Liu, S. Wang, A quantitative detection algorithm based on improved faster r-cnn for marine benthos, Ecological Informatics 61 (2021) 101228.

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

[23] L. Chen, W. Shi, D. Deng, Improved yolov3 based on attention mechanism for fast and accurate ship detection in optical remote sensing images, Remote Sensing 13 (2021) 660.

[24] J. Ye, Z. Yuan, C. Qian, X. Li, Caa-yolo: Combined-attention-augmented yolo for infrared ocean ships detection, Sensors 22 (2022) 3782.

[25] G. Jocher, YOLOv5 by Ultralytics, 2020. URL: https://github.com/ultralytics/yolov5.

[26] M. Tan, R. Pang, Q. V. Le, Efficientdet: Scalable and efficient object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10781–10790.

[27] P. Berg, D. Santana Maia, M.-T. Pham, S. Lefèvre, Weakly supervised detection of marine animals in high resolution aerial images, Remote Sensing 14 (2022) 339.

[28] X. Chen, X. Mu, J. Guan, N. Liu, W. Zhou, Marine target detection based on marine-faster r-cnn for navigation radar plane position indicator images, Frontiers of Information Technology & Electronic Engineering 23 (2022) 630–643.

[29] L. Si, G. Li, C. Zheng, F. Xu, Self-supervised representation learning for the object detection of marine radar, in: Proceedings of the 8th International Conference on Computing and Artificial Intelligence, 2022, pp. 751–760.

[30] X. Dong, M. J. Chantler, Perceptually motivated image features using contours, IEEE Transactions on Image Processing 25 (2016) 5050–5062.

[31] J.-X. Zhao, J.-J. Liu, D.-P. Fan, Y. Cao, J. Yang, M.-M. Cheng, Egnet: Edge guidance network for salient object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 8779–8788.

[32] J.-J. Liu, Q. Hou, M.-M. Cheng, J. Feng, J. Jiang, A simple pooling-based design for real-time salient object detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 3917–3926.

[33] Y. Sun, S. Wang, C. Chen, T.-Z. Xiang, Boundary-guided camouflaged object detection, arXiv preprint arXiv:2207.00794 (2022).

[34] C.-C. Yeung, K.-M. Lam, Attentive boundary-aware fusion for defect semantic segmentation using transformer, IEEE Transactions on Instrumentation and Measurement (2023).

[35] X. Dong, J. Dong, Oceanic scene recognition using graph-of-words (gow), in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 1122–1130.

[36] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence (2017).

[37] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, A. Torralba, Sun database: Large-scale scene recognition from abbey to zoo, in: 2010 IEEE computer society conference on computer vision and pattern recognition, IEEE, 2010, pp. 3485–3492.

[38] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, International journal of computer vision 88 (2010) 303–338.

[39] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, volume 1, Ieee, 2001, pp. I–I.

[40] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: 2005 IEEE computer society conference on computer vision and pattern recognition, volume 1, 2005, pp. 886–893.

[41] P. Felzenszwalb, D. McAllester, D. Ramanan, A discriminatively trained, multiscale, deformable part model, in: 2008 IEEE conference on computer vision and pattern recognition, Ieee, 2008, pp. 1–8.

[42] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE transactions on pattern analysis and machine intelligence 37 (2015) 1904–1916.
[43] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European conference on computer vision, 2020, pp. 213–229.
[44] X. Mou, X. Chen, J. Guan, B. Chen, Y. Dong, Marine target detection based on improved faster r-cnn for navigation radar ppi images, in: 2019 International Conference on Control, Automation and Information Sciences (ICCAIS), IEEE, 2019, pp. 1–5.
[45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
[46] F. Bi, J. Hou, L. Chen, Z. Yang, Y. Wang, Ship detection for optical remote sensing images based on visual attention enhanced network, Sensors 19 (2019) 2271.
[47] Z. Shen, Z. Liu, J. Li, Y.-G. Jiang, Y. Chen, X. Xue, Dsod: Learning deeply supervised object detectors from scratch, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 1919–1927.
[48] M. Wu, J. Niu, L. Zhang, Q. J. Wu, C. Shi, J. Zhang, Target detection for rd images of hfswr based on cnn-elm model, in: OCEANS 2021: San Diego–Porto, IEEE, 2021, pp. 1–4.
[49] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (2006) 489–501.
[50] Y. Liu, S. Wang, A quantitative detection algorithm based on improved faster r-cnn for marine benthos, Ecological Informatics 61 (2021) 101228.
[51] X. Li, W. Wang, X. Hu, J. Yang, Selective kernel networks, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 510–519.
[52] A. Sánchez-Ferrer, J. J. Valero-Mas, A. J. Gallego, J. Calvo-Zaragoza, An experimental study on marine debris location and recognition using object detection, Pattern Recognition Letters 168 (2023) 154–161.
[53] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.
[54] F. S. Hass, J. Jokar Arsanjani, Deep learning for detecting and classifying ocean objects: application of yolov3 for iceberg–ship discrimination, ISPRS International Journal of Geo-Information 9 (2020) 758.
[55] R. W. Liu, W. Yuan, X. Chen, Y. Lu, An enhanced cnn-enabled learning method for promoting ship detection in maritime surveillance system, Ocean Engineering 235 (2021) 109435.
[56] A. Al Muksit, F. Hasan, M. F. H. B. Emon, M. R. Haque, A. R. Anwary, S. Shatabda, Yolo-fish: A robust fish detection model to detect fish in realistic underwater environment, Ecological Informatics 72 (2022) 101847.
[57] J. Zhu, W. He, W. Weng, T. Zhang, Y. Mao, X. Yuan, P. Ma, G. Mao, An embedding skeleton for fish detection and marine organisms recognition, Symmetry 14 (2022) 1082.
[58] B. Wang, P. Jiang, J. Gao, W. Huo, Z. Yang, Y. Liao, A lightweight few-shot marine object detection network for unmanned surface vehicles, Ocean Engineering 277 (2023) 114329.
[59] X. Zhang, X. Zhou, M. Lin, J. Sun, Shufflenet: An extremely efficient convolutional neural network for mobile devices, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 6848–6856.
[60] H. Yu, X. Li, Y. Feng, S. Han, Multiple attentional path aggregation network for marine object detection, Applied Intelligence 53 (2023) 2434–2451.
[61] S. Li, Y. Liu, S. Wu, S. J. Zhang, Mdm-yolo: Research on object detection algorithm based on improved yolov4 for marine organisms, Computing and Informatics 42 (2023) 210–233.
[62] Z. Jia, X. Su, G. Ma, T. Dai, J. Sun, Crack identification for marine engineering equipment based on improved ssd and yolov5, Ocean Engineering 268 (2023) 113534.
[63] X. Dong, J. Dong, The visual word booster: A spatial layout of words descriptor exploiting contour cues, IEEE Transactions on Image Processing 27 (2018) 3904–3917.
[64] X. Dong, M. A. Garratt, S. G. Anavatti, H. A. Abbass, J. Dong, Lightweight monocular depth estimation with an edge guided network, in: 2022 17th International Conference on Control, Automation, Robotics and Vision (ICARCV), IEEE, 2022, pp. 204–210.
[65] Tzutalin, Labelimg, https://github.com/tzutalin/labelImg, 2015.
[66] C.-Y. Wang, H.-Y. M. Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, I.-H. Yeh, Cspnet: A new backbone that can enhance learning capability of cnn, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops, 2020, pp. 390–391.
[67] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-iou loss: Faster and better learning for bounding box regression, in: Proceedings of the AAAI conference on artificial intelligence, volume 34, 2020, pp. 12993–13000.
[68] Y.-H. Wu, Y. Liu, L. Zhang, M.-M. Cheng, B. Ren, Edn: Salient object detection via extremely-downsampled network, IEEE Transactions on Image Processing 31 (2022) 3125–3136.
[69] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, L. Van Gool, Multi-task learning for dense prediction tasks: A survey, IEEE transactions on pattern analysis and machine intelligence 44 (2021) 3614–3633.
[70] F. Milletari, N. Navab, S.-A. Ahmadi, V-net: Fully convolutional neural networks for volumetric medical image segmentation, in: 2016 fourth international conference on 3D vision, 2016, pp. 565–571.
[71] J. Canny, A computational approach to edge detection, IEEE Transactions on pattern analysis and machine intelligence (1986) 679–698.
[72] N. Efthymiadis, G. Tolias, O. Chum, Edge augmentation for large-scale sketch recognition without sketches, in: 2022 26th International Conference on Pattern Recognition, 2022, pp. 3595–3602.
[73] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Advances in neural information processing systems 30 (2017).
[74] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, X. Bai, Richer convolutional features for edge detection, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3000–3009.
[75] S. Xie, Z. Tu, Holistically-nested edge detection, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1395–1403.
[76] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 7464–7475.