Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Underwater object detection by fusing features from different representations of sonar data*

Fei WANG[†1], Wanyu LI[1], Miao LIU[2], Jingchun ZHOU[†‡1], Weishi ZHANG[†1]

[1]*College of Information Science and Technology, Dalian Maritime University, Dalian 116026, China*

[2]*College of Transportation Engineering, Dalian Maritime University, Dalian 116026, China*

[†]E-mail: feiwang@dlmu.edu.cn; zhoujingchun@dlmu.edu.cn; teesiv@dlmu.edu.cn

Received Oct. 3, 2022; Revision accepted Feb. 2, 2023; Crosschecked June 1, 2023

Abstract: Modern underwater object detection methods recognize objects from sonar data based on their geometric shapes. However, the distortion of objects during data acquisition and representation is seldom considered. In this paper, we present a detailed summary of representations for sonar data and a concrete analysis of the geometric characteristics of different data representations. Based on this, a feature fusion framework is proposed to fully use the intensity features extracted from the polar image representation and the geometric features learned from the point cloud representation of sonar data. Three feature fusion strategies are presented to investigate the impact of feature fusion on different components of the detection pipeline. In addition, the fusion strategies can be easily integrated into other detectors, such as the You Only Look Once (YOLO) series. The effectiveness of our proposed framework and feature fusion strategies is demonstrated on a public sonar dataset captured in real-world underwater environments. Experimental results show that our method benefits both the region proposal and the object classification modules in the detectors.

Key words: Underwater object detection; Sonar data representation; Feature fusion
https://doi.org/10.1631/FITEE.2200429 **CLC number:** TN911.73; TP391.41

1 Introduction

Oceans cover more than 70% of the Earth with abundant natural resources that can be used by humans. To explore the unknown secrets of oceans and facilitate the development of marine resources, many researchers have focused on the study of underwater object detection algorithms and computer vision technology (Zhou et al., 2022a). Underwater object detection plays a major role in marine fields, such as the fishing industry and military (Zhou et al., 2022b). Sonar (sound navigation and ranging) is a technique that uses sound propagation to measure distance. It is insensitive to illumination changes and is especially suitable for object detection in harsh underwater environments. Therefore, many studies have focused on underwater object detection with sonar data.

Recently, a number of deep models have been proposed to improve the accuracy of underwater object detection with sonar data. Pu et al. (2021) proposed a scene-adaptive-evolution unsupervised video object detection algorithm based on the object group concept. It consists of a prototype dictionary generation strategy and a graph-based group information propagation strategy for mining positive samples from the unlabeled new scene dataset. Finally, the new positive samples with pseudo-labels act as the training data to fine-tune the detection model for detecting the new scene. Tian et al. (2022) proposed a garbage detection

‡ Corresponding author

* Project supported by the National Natural Science Foundation of China (No. 62103072) and the Postdoctoral Science Foundation of China (No. 2021M690502)

ORCID: Fei WANG, https://orcid.org/0000-0002-3973-6037; Jingchun ZHOU, https://orcid.org/0000-0002-4111-6240; Weishi ZHANG, https://orcid.org/0000-0003-0519-8397

method based on modified You Only Look Once v4 (YOLOv4), allowing high-speed and high-precision object detection. It converts the original YOLOv4 (Bochkovskiy et al., 2020) to four-scale YOLOv4 and then performs model pruning on four-scale YOLOv4. The detection speed is up to 66.67 frames/s with a mean average precision (mAP) of 95.10% on the sonar object detection dataset provided by the 2021 China Underwater Robot Professional Contest. A marine target detection method (Chen XL et al., 2022) based on the Marine-Faster region-based convolutional neural network (R-CNN) algorithm was proposed in the case of complex background and target characteristics. To improve the accuracy of detecting marine targets and reduce the false alarm rate, Faster R-CNN was optimized as the Marine-Faster R-CNN in five respects: new backbone network, anchor size, dense target detection, data sample balance, and scale normalization. Ben Tamou et al. (2021) proposed two new fusion approaches that exploit two convolutional neural network (CNN) streams to merge both appearance and motion information for automatic fish detection. When training a deep model, existing sonar datasets are relatively small. Three data augmentation methods (Huang et al., 2019) are dedicated to underwater imaging, including the inverse process of underwater image restoration, perspective transformation, and illumination synthesis.

However, existing methods have focused mainly on improving the model structure and training procedure. The influence of sonar data representation on the detection results has seldom been investigated. Thus, we first introduce three representations of sonar data in this paper, including polar image, Cartesian image, and point cloud, as shown in Fig. 1. In the polar image representation, it can be seen that the shape of the cylinder in the image changes, and the angle of the cube is stretched from the right angle to a sharp angle, while the angles of the cube in the Cartesian image and point cloud representations remain right. Geometric distortion is caused by using different data representations. Motivated by this observation, we demonstrate that the representation of sonar data does have a great influence on object detection results.

This paper focuses on how sonar data representation affects object detection results and attempts to find a solution to fuse features from different representations to improve the accuracy of detection results. To this end, a detailed summary of three representations of sonar data is presented. Theoretical and experimental comparisons among different representations are given as well. Two kinds of distortion in sonar data are discussed, including projection distortion and representation distortion. The first is unrecoverable due to the limitation of sonar sensors, while the second is introduced by data representations.

Our main contributions are summarized as follows:

1. We present a detailed summary of the acquisition and representations of sonar data and a concrete analysis of the geometric features of different data representations for underwater object detection.

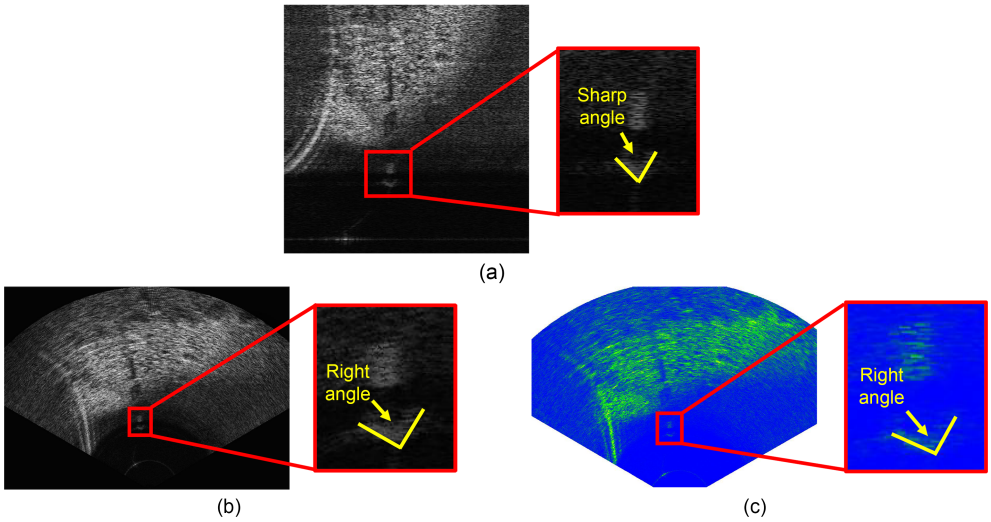

Fig. 1 Three representations of the same sonar data: (a) polar image; (b) Cartesian image; (c) point cloud

2. We propose a feature fusion framework with different sonar data representations for underwater object detection. Our framework fuses the intensity features extracted from the distorted polar image representation and the undistorted geometric features extracted from the recovered point cloud representation to improve the accuracy of the detection results. In addition, our feature fusion strategies can be easily integrated into the state-of-the-art detectors.

3. We conduct a series of experiments on a public sonar dataset to demonstrate the effectiveness of our framework. Three feature fusion strategies are tested to investigate the effect of feature fusion on different components of the detection pipeline.

2 Related works

Many studies in the field of object detection have been published in recent years, which laid the foundation of our research. This section will focus on deep-learning-based object detection research, especially the methods for underwater object detection with sonar data.

2.1 Object detection based on deep learning

Current deep-learning-based object detectors can be divided into two categories: two-stage detection frameworks and one-stage detection frameworks. Two-stage detection frameworks, also known as the region-based methods (Girshick et al., 2016), include R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), Faster R-CNN (Ren et al., 2015), sparse R-CNN (Sun et al., 2021), and dynamic R-CNN (Zhang et al., 2020). These methods divide the detectors into two parts: region proposal and object classification. One-stage detection methods can directly obtain predictions without a region proposal stage, which is also known as the region-free technique, such as YOLO series algorithms (Redmon et al., 2016; Redmon and Farhadi, 2018; Ge et al., 2021), Task-aligned One-stage Object Detection (TOOD) (Feng et al., 2021), and Disentangle Dense Object Detector (DDOD) (Chen ZH et al., 2021).

To enhance the feature extraction ability for multiscale object detection, Lin et al. (2017) proposed feature pyramid networks (FPNs), which are used to fuse low-resolution high-level and high-resolution low-level semantic information features. It achieves significant improvements over several strong baselines. Neural Architecture Search (NAS)-FPN (Ghiasi et al., 2019) seeks an effective feature fusion mode in FPN and refines features through repeated superposition. The merging cell proposed by Ghiasi et al. (2019) is used to generate the new feature map as the input layer in subnetworks. To extract important features, Yang et al. (2021) proposed a module based on the FPN structure with parallel feature fusion named PFF-FPN. PFF-FPN uses three different FPNs to extract features and fuses the corresponding layer features to reinforce the focused layer feature information. Due to feature redundancy, ambiguity, and inaccuracy on small targets, Liu and Cheng (2021) proposed the spatial refinement model (SRM)-FPN feature fusion method, which can effectively improve the detection effect of small targets in the image. Specifically, SRM is used to learn the localization information of feature points according to the context features between adjacent layers and content. Moreover, it uses the adaptive channel merging method of the attention mechanism to optimize feature fusion.

2.2 Underwater object detection with sonar data

Forward-looking sonar can provide high-resolution images that can be used for different tasks in underwater environments. To improve the accuracy of object detection on sonar images and solve the problems raised by the complex and changeable underwater environment, several deep-learning-based approaches have been proposed. CNN is used for object recognition on forward-looking sonar images (Valdenegro-Toro, 2016). It shows that CNNs can provide better performance while keeping a low parameter count. Kim and Yu (2016) applied CNN to an agent vehicle system to enhance underwater manipulation. The method uses the sonar images of a moving agent obtained by forward-looking sonar as input. The YOLO detector is applied to detect the location of the agent in the sonar images. Song et al. (2021) presented an automatic real-time object detection method based on a self-cascade convolutional neural network (SC-CNN). It is superior to typical CNN and unsupervised methods. SC-CNN is used for high-accuracy side-scan sonar (SSS) image segmentation, which is robust to speckle noise and intensity inhomogeneity. Moreover,

the segmentation results of the SC-CNN are used to initialize the Markov random filed (MRF) to obtain the final segmentation maps. Deep learning has shown excellent performance in image feature extraction and has been extensively used in image object detection and instance segmentation. These methods use only the direct output of sonar sensors (polar image) as input, which is low resolution and distorted. Therefore, the shape of the same object projected on different images is not fixed. It can cause a decrease in detection accuracy.

Recently, a feature fusion algorithm for underwater sonar image object detection has been proposed and has become a trend for improving the accuracy of object detection. For sonar datasets with small effective samples and low signal-to-noise ratios (SNRs), an improved YOLOv3 algorithm for real-time detection called YOLOv3-DPFIN was proposed (Kong et al., 2020). It not only conducts efficient feature extraction via the dual-path network (DPN) module and the fusion transition module, but also adopts a dense connection method to improve multiscale prediction. It can complete precise object classification and localization. With the deepening of the network, it increases the number of parameters. The large number of parameters will make model training difficult, increase the computational complexity, and lead to overfitting. To accurately and quickly segment different targets in a sonar image, Wang et al. (2022) proposed a multichannel fusion convolutional neural network (MCF-CNN). MCF-CNNs use a deep separable residual module to extract multiscale features and the multichannel fusion operation to enhance feature representation on different scales. However, repetitive feature fusion operations at different stages can increase the number of calculations and parameters.

Existing methods focus mainly on improving the model structure and training procedure. The influence of sonar data representation on the detection results has seldom been investigated. The polar image of sonar data is low resolution and distorted during sonar imaging. Many methods commit mainly to work on multiscale feature fusion and multichannel fusion operations to enhance feature representation. In this paper, to reduce the influence of object distortion in sonar data, we first transform the raw sonar data in a polar coordinate system into a Cartesian coordinate system. Then, the point cloud representation of sonar data can be easily generated by the Cartesian image. Our proposed fusion strategies fuse mainly the intensity features (extracted from the distorted polar image representation) and the geometric features (extracted from the recovered point cloud representation) at different components. The fusion features fused by using our proposed strategies can include more details of different sonar data representations. Fusing these features can fully use the advantages of both representations to overcome the distortion in a single representation.

3 Acquisition and representation of sonar data

3.1 Acquisition of sonar data

Sonar is a technique that can be widely used for underwater environment perception. It sends out an acoustic pulse in water and measures distances in terms of the time for the echo of the pulse to return. The resulting sonar data consist of distances, azimuth angles, elevation angles, intensities, and spectra of acoustic signals reflected by objects. Sonars include mainly forward-looking sonars, side-scan sonars, and synthetic aperture sonars according to the measuring strategy. In addition, forward-looking sonars, which are famous for multifrequency, low energy consumption, and small size, can be divided into single-beam forward-looking sonar and multibeam forward-looking sonar in terms of the number of beams radiated at the same time.

In Fig. 2, the measurement processing of a forward-looking sonar is presented. The sonar sends out a bunch of beams that are similar to scanlines to measure the visible area. The radiated acoustic pulses will bounce back when they encounter objects, and then after a period of time, the sonar will detect the beam reflection. The distance between the object and the sonar is accurately calculated by the product of the time from radiating to receiving and the known speed of beams in water. By establishing the coordinate system with the location of the sonar as the origin, it is clear to know the location of the object while knowing the distance between the two points. The raw sonar image depicting the edge outline of the object is created by innumerable beam reflections that are projected into the sonar.

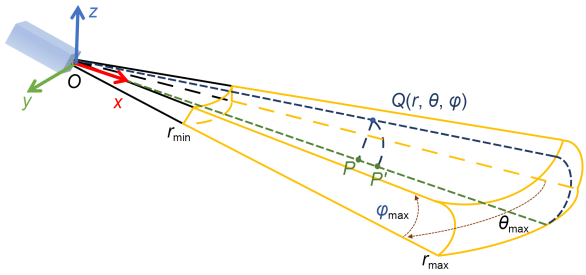

Fig. 2 Sonar data acquisition

$$\boldsymbol{Q}=\begin{bmatrix} r \\ \theta \\ \varphi \end{bmatrix}=\begin{bmatrix} \sqrt{x^2+y^2+z^2} \\ \arctan\left(y/x\right) \\ \arctan\left(z/\sqrt{x^2+y^2}\right) \end{bmatrix}. \tag{1}$$

The visible area is defined by the minimum and maximum ranges $r_{\min}$ and $r_{\max}$, the maximum azimuth angle $\theta_{\max}$, and the maximum elevation angle $\varphi_{\max}$. Assuming Q is a measurement point in three-dimensional (3D) spaces with polar coordinates (r, θ, φ), the relationship between its polar coordinates and Cartesian coordinates (x, y, z) can be formulated as Eq. (1). The polar coordinates of a sonar measurement point can be calculated when its Cartesian coordinates are provided, and vice versa.

However, the elevation angle φ of a sonar measurement point is unsensible due to the limitation of sonar devices. As a result, 3D measurement points are projected onto a two-dimensional (2D) plane, and the acquired sonar data are restricted to 2D points rather than real 3D points. Cartesian coordinates of the acquired sonar data are calculated as follows:

$$\boldsymbol{P'}=\begin{bmatrix} x' \\ y' \end{bmatrix}=\begin{bmatrix} r\cos\theta \\ r\sin\theta \end{bmatrix}. \tag{2}$$

This projection will lead to unrecoverable geometric distortion, as shown in Fig. 2. Q is a 3D measurement point. The orthographic projection point of Q onto the 2D plane is P. However, due to the lack of the elevation angle, the real acquired point is P', where the distance of OP' equals that of OQ. Many geometric details of the point cloud representation have already been lost when 3D points are projected onto a plane, but this additional distortion will worsen the observed geometric characteristics of objects in sonar data.

3.2 Representations of sonar data

This subsection will introduce three kinds of sonar data representations, including polar image, Cartesian image, and point cloud. Analyses of the geometric characteristics of different data representations are also given.

Polar images are a direct representation of sonar data. Two coordinate axes in a polar image correspond to the r and θ axes of the polar coordinate system in 3D scanning spaces. The value of each pixel represents the sonar signal at each scanning point. The relationship and pixel value can be calculated as Eq. (3), where $\mathrm{Image}_{\mathrm{polar}}(i, j)$ represents the pixel value with coordinates (i, j) in the polar image, $\mathrm{valueAt}_{\mathrm{polar}}(ir_{\mathrm{res}}, j\theta_{\mathrm{res}})$ represents the sonar signal value at the scanning point with the polar coordinate $(ir_{\mathrm{res}}, j\theta_{\mathrm{res}})$, and r_{res} and θ_{res} are the resolutions of the distance and azimuth angle, respectively. For example, assuming that the resolution of the scanning distance is 10 cm and the resolution of the azimuth angle is 0.1°, the value of pixel (20, 30) in a polar image is the sonar signal value of the scanning point with polar coordinates (2 m, 3°).

$$\mathrm{Image}_{\mathrm{polar}}\left(i, j\right)=\mathrm{valueAt}_{\mathrm{polar}}\left(ir_{\mathrm{res}}, j\theta_{\mathrm{res}}\right). \tag{3}$$

Cartesian images use the real-world coordinate system of the projection plane to form the image plane. During Cartesian image generation, raw sonar data in the polar coordinate system are first transformed into the Cartesian coordinate system. Since the elevation angle of each scanning point cannot be measured, the transformed point has only two axes' values. Then, a resolution is selected manually to perform pixelization. For example, if the spatial resolution is set to 1 cm^2, then a pixel in the resulting Cartesian image corresponds to a 1-cm^2 area in the projection plane. Finally, the mean sonar signal of each small area is set to be the pixel value in the Cartesian image.

$$\mathrm{Image}_{\mathrm{cartesian}}\left(i, j\right)=\mathrm{mean}\left(\mathrm{valueIn}_{\mathrm{cartesian}}\left(ix'_{\mathrm{res}}, jy'_{\mathrm{res}}\right)\right), \tag{4}$$

where $\mathrm{Image}_{\mathrm{cartesian}}(i, j)$ represents the pixel value at (i, j) in the Cartesian image and $\mathrm{valueIn}_{\mathrm{cartesian}}(ix'_{\mathrm{res}}, jy'_{\mathrm{res}})$ represents the sonar signal value of the point whose Cartesian coordinates are $(ix'_{\mathrm{res}}, jy'_{\mathrm{res}})$.

Point clouds have been never used as the above two representations in the domain of sonar data processing; therefore, the use of point clouds in this paper is innovative. The point cloud representation of sonar data can be easily obtained by transforming raw polar-coordinate sonar data into Cartesian-coordinate sonar data. However, the 3D point cloud representation of sonar data cannot be generated due to the lack of φ (angle of pitch) in the original data. Therefore, this paper uses an $N\times2$ matrix to represent the 2D coordinates of N points transformed from the polar coordinate system. Only the points with a sonar signal greater than a threshold are restored in the matrix; other points are considered as background and ignored.

$$\text{Point}(i)=\left(x_i',y_i'\right). \tag{5}$$

Analyses: (1) Polar images are the direct outputs of sonar sensors, requiring no further data transformation. Bounding boxes are easily defined along each scanline. However, there is a clear geometric distortion when using polar image representation. It can be observed from Figs. 1 and 3. This distortion is caused by using the 2D polar coordinate system of the projection plane to form coordinates in an image frame. This representation distortion will lead to unstable geometric characteristics of the same object in a polar image and may affect the identification of the object class. (2) There is no additional representation distortion in the Cartesian image representation and the point cloud representation since these two representations use the real-world coordinate system of the projection plane. According to that, we can infer that object classification with these two representations would be much easier than that in polar images. (3) During the generation of Cartesian images, a spatial resolution must be selected manually. This resolution also affects the performance of object detection. A larger resolution results in a smaller Cartesian image; thus, more geometric details are lost. A smaller resolution can not only restore more details but also result in a larger image size, which will slow down the detection pipeline. (4) The ground truth bounding box of the point cloud representation is harder to label than that in polar and Cartesian image representations, which makes training a 3D detector more difficult. Further experimental results and analyses on different representations are given in Section 5.

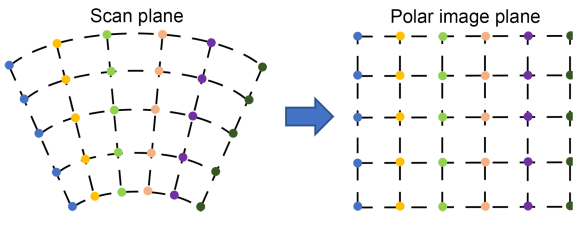

Fig. 3 Representation distortion in a polar image

4 Approach

4.1 Overview

Based on the above analyses, we present a framework to fuse features from polar image and point cloud representations of sonar data for object detection. Fusing these features can fully use the advantages of both polar image and point cloud representations to overcome the distortion in a single representation. The structure of our framework is shown in Fig. 4. Faster R-CNN is selected as the baseline structure since it is a typical two-stage detector, in which the effect of feature fusion on different components of the whole pipeline can be clearly investigated. However, our framework is not limited to this two-stage detector. The feature fusion strategies can be easily integrated into other detectors, such as the YOLO series.

The baseline structure uses ResNet-50 for feature extraction from the polar image, FPN to fuse features at different scales, the region proposal network (RPN) to generate object proposals, and the region of interest (ROI) pooling and ROI head of the R-CNN component to detect the location and class of each candidate box. Moreover, features extracted from the point cloud representation of the raw sonar data are used for detection. We present three strategies to fuse point cloud features: input fusion, fusing before RPN (RPN fusion), and fusing before the ROI pooling and ROI head of the R-CNN (R-CNN fusion). By these feature fusions, we want to use point cloud features to overcome the distortion in the polar image.

4.2 Point cloud generation and filtering

Eqs. (1) and (2) are used to calculate the Cartesian coordinates of each scanning point. The resulting point cloud is stored in an $N\times2$ matrix, where N is the total number of scanning points. Each element in the matrix is defined as the sonar signal value at

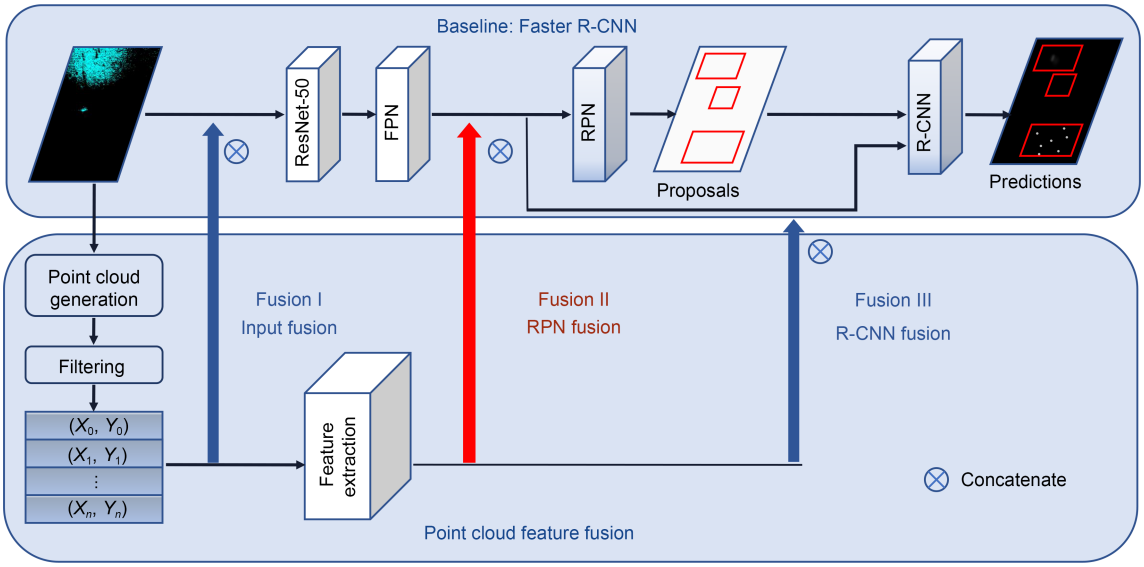

Fig. 4 Structure of our feature fusion framework for object detection of sonar data

the corresponding scanning point. If there is an obstacle at the scanning point, then the sonar signal is close to the maximum value, and the scanning point is considered occupied; otherwise, the signal is close to the minimum value, and the scanning point is considered unoccupied.

Unoccupied scanning points correspond to free space, and are useless for object detection. Therefore, preprocessing is introduced to filter out these background points. It can be observed that the number of occupied scanning points is relatively small, and the difference in average signal values between occupied and unoccupied points is relatively large. Therefore, it can be formulated as an image binarization problem. The Otsu method (Otsu, 1979) is adopted to calculate a proper threshold for sonar signals. Then, only points with a signal value larger than the threshold are kept (Eq. (6)). In this way, the unoccupied scanning area is filtered out from the point cloud. This helps to reduce the number of points, which is the input to the remaining detection pipeline.

$$\mathrm{PC}_{\mathrm{filtered}} = \begin{cases} \mathrm{Point}(i), & \text{if } \mathrm{valueAt}_{\mathrm{point}}(x_i', y_i') > \mathrm{threshold}, \\ (0,0), & \text{otherwise.} \end{cases} \tag{6}$$

4.3 Feature extraction

ResNet-50 and FPN are used to extract features at different scales from the polar image, as shown in Eqs. (7) and (8). For feature extraction of a set of point clouds, various 3D deep models, such as PointNet (Charles et al., 2017), can be adopted by simply reducing 3D input points to 2D points. However, when using these simplified deep models, those extracted features may not be easily fused to polar image features due to the lack of correspondence between the two representation features. Thus, in this paper, a similar feature extraction model is used as the polar image instead. The point cloud is stored in a $W\times H\times 2$ matrix, where W and H are the width and height of the polar image, respectively. Two values of each element in the matrix are (1) Cartesian coordinates of the corresponding scanning point if the point is occupied and (2) set to be (0, 0) otherwise. In this way, the polar image and point cloud have the same input shape except for the input channels. Therefore, a separate ResNet-50 with a two-channel input and FPN structure is used to extract features from a set of point clouds.

$$f_{\mathrm{image}} = \mathrm{FPN}\big(\mathrm{ResNet}\,(\mathrm{Image})\big), \tag{7}$$

$$f_{\mathrm{pointcloud}} = \mathrm{FPN}'\big(\mathrm{ResNet}'(\mathrm{PC}_{\mathrm{filtered}})\big). \tag{8}$$

4.4 Feature fusion strategies

Three strategies to fuse features from polar image and point cloud representations of sonar data are presented, including input fusion, fusing before RPN

(RPN fusion), and fusing before the ROI pooling and ROI head of the R-CNN (R-CNN fusion). Details of each strategy are described as follows.

Input fusion is a data-level fusion strategy. The polar image representation is a gray image in general. Each pixel in a polar image has only one channel, where the value is the intensity of the reflected sonar signal. In point cloud representation, as mentioned in the previous part, a $W\times H\times 2$ matrix is used to store the Cartesian coordinates of each sonar point. In this way, polar image and point cloud representations share the same spatial size except for the number of channels. Thus, the input fusion strategy fuses the input data by directly concatenating these two matrices along the third dimension (Eq. (9)). The result is a $W\times H\times 3$ matrix, where each element forms (x, y, I), including the 2D Cartesian coordinates (x, y) of a measurement point and the intensity I of the corresponding sonar signal. The input fusion strategy is a simple and straightforward way to fuse features from different data representations. The fused data will be used as inputs to a detector. Features are learned from the fused data and affect the whole detection pipeline.

$$\text{Input} = \text{concatenate}\left(\text{Image}_{\text{polar\_gray}}, \text{PC}_{\text{filtered}}\right). \quad (9)$$

RPN fusion is a strategy that fuses features from polar image and point cloud representations before RPN. It can be formulated as Eq. (10). In this strategy, the intensity features are extracted from the polar image representation, and the geometric features are extracted from the point cloud representation. Then, the intensity features and the geometric features are fused and fed into the RPN components, which are used for region proposal generation. Next, the features mapped from the RPN components are fed into the ROI pooling and ROI head of the R-CNN for classification and regression tasks. Therefore, this fusion strategy affects the generation of region proposals and final predictions of object classes and bounding boxes. Using separate feature learning models for different data representations decouples the entire feature fusion framework, helping to increase the scalability of the proposed method. This strategy investigates the impact of intensity–geometric feature fusion on both region proposals and object predictions.

$$f_{\text{RPN}} = \text{concatenate}\left(f_{\text{image}}, f_{\text{pointcloud}}\right). \quad (10)$$

R-CNN fusion is a strategy that fuses features from polar image and point cloud representations before the ROI pooling and ROI head of the R-CNN component. Fused features are used only for object classification and bounding box prediction in R-CNN. Region proposals are generated based on the intensity features extracted from the polar image representation. This strategy is motivated by the observation that the representation distortion in a polar image does damage the stability of geometric characteristics of the same object but does not change its boundary. Using the intensity features extracted from the polar image is able to predict candidate object regions. Those geometric features learned from the undistorted point cloud are fused and fed into the R-CNN to provide more stable geometric features of different objects and help to make a better object classification. It can be formulated as follows:

$$f_{\text{R-CNN}} = \text{concatenate}\left(f_{\text{image}}, f_{\text{pointcloud}}\right). \quad (11)$$

Common feature fusion operations include concatenation and add operations. The concatenation operation merges the input matrix along the feature dimension. The number of feature dimensions increases, while the feature information of each dimension does not change. This operation will not cause any loss of input feature information. The feature fused by the concatenation operation may improve the accuracy of object detection, but this operation will lead to an increase in parameters and computations. The add operation is an elementwise add operation on the features of each dimension without changing the number of dimensions. It requires inputs to share the same shape. After the add operation, the feature information of each dimension increases. However, the original features from different inputs cannot be distinguished from the fused features. Thus, all three feature fusion strategies in this paper use the concatenation operation rather than the add operation.

5 Experiments and results

To demonstrate the effectiveness of our feature fusion framework, we test it on a public dataset of real-world underwater environments with various kinds of target objects. A series of experiments are conducted to investigate the three feature fusion strategies

presented in our framework. Both qualitative and quantitative results are provided in this section.

5.1 Datasets

The public dataset used in our experiments is provided by the 2021 China Underwater Robot Professional Contest. The dataset consists of 4000 labelled polar sonar images, covering typical underwater scene environments. It contains eight categories of typical objects, including cube, ball, cylinder, human body, tyre, circle cage, square cage, and metal bucket. In our experiments, the entire dataset of 4000 images is split into three sets: training set (approximately 60%), validation set (approximately 20%), and test set (approximately 20%). The split is performed not according to the number of images but to the number of each kind of objects that exists in the set. Assuming there are 1000 images that include a single object of ball and 500 images that include a ball and some other objects, we first use a random sampling method to select 600 images from the 1000 images into the training set, 200 images into the validation set, and 200 images into the test set; then, we randomly sample 300 images from 500 images into the training set, 100 images into the validation set, and 100 images into the test set. As a result, there are 600+200 images for training, 200+100 images for validation, and 200+100 images for test, where each image contains a single target object or multiple objects. If there are overlapped images between multiple objects, those duplicate images will be removed. This random sampling is performed three times. All experimental results in this section are mean values on these three randomly generated datasets. Details of the datasets are shown in Table 1.

In addition, the above random sampling method can ensure the following points: (1) The data distribution of the original datasets is essentially in agreement with the training set, validation set, and test set. (2) It is basically the same proportion of images that include the single object and multiple objects in the training set, validation set, and test set to make the difficulty of object detection in the three subsets almost the same. (3) As we performed three random sampling partitions, we compared the experimental results on three-partition datasets in subsequent experiments, which comprehensively validated the effectiveness of our method.

5.2 Metrics

In our experiments, two evaluation metrics are used to compare the performance of different approaches: mAP and average recall (AR). Precision refers to the proportion of samples that are predicted to be positive in all positives, including true positives and false positives, and recall refers to the proportion of samples that are predicted to be positive in true positives and false negatives. mAP is a widely used metric for object detection that is the mean precision among all categories, which is used to evaluate the overall performance of different detection approaches. AR is used to quantify the results of region proposals, which is used to evaluate the effectiveness of feature fusion on region proposals.

5.3 Implementation details

In this paper, MMDetection (Chen K et al., 2019), an open-sourced object detection framework, is used to implement the proposed feature fusion approaches. Codes are written in Python 3.7 and PyTorch 1.8.1. The server operating system of all experiments is Ubuntu 18.04, CUDA Toolkit 11.1, CUDNN 8.0, and MMDetection 2.12, while the hardware includes Intel Core i9-10980XE (3.0 GHz), 32 GB RAM, and NVIDIA GeForce 3090 (24 GB memory).

The polar image is duplicated along the pixel channel to form an image with three channels. Then, the pretrained ResNet-50 model is used for intensity

Table 1 Number of images with single object/multiple objects in each subset

Dataset	Number of images with single object/multiple objects							
	Ball	Square cage	Tyre	Circle cage	Cube	Metal bucket	Human body	Cylinder
Original dataset	82/1861	24/631	14/838	11/375	1194/557	20/383	208/476	215/187
Training set	50/1003	16/351	10/482	7/207	718/279	12/214	126/253	129/100
Validation set	16/421	4/136	2/189	2/75	238/138	4/84	41/113	43/38
Test set	16/437	4/144	2/167	2/93	238/140	4/85	41/110	43/49

feature extraction, and we fine-tune the parameters during training. For the geometric feature extraction of the point cloud representation, another ResNet-50 model structure is used with the parameter initialized from a normal distribution with 0 mean and 0.001 standard deviation (std). RPN and R-CNN are also randomly initialized from normal distributions, while Xavier initialization is adopted for FPN. When feature fusion strategies are used in these approaches, the number of channels of components involved in fused features is doubled compared to the original Faster R-CNN. All models are optimized by stochastic gradient descent (SGD) with an initial learning rate of 0.02 and momentum of 0.9. Linear warmup is adopted every 500 iterations. Weight decay and random flip are used to prevent overfitting.

5.4 Comparison with different data representations

Qualitative comparison: Fig. 5 illustrates the typical detection results with different representations of sonar data on the given datasets. For the polar image representation, the ball in the first scene is misclassified as a human body; false positive predictions are also observed, such as the predicted ball in the second scene and the tyre and cube in the third scene. Compared with other representations, more misclassifications and false positive predictions are observed in the polar image representation. This is caused mainly by the representation distortion in the polar image. This additional distortion makes it harder for detectors to learn the distinguishable geometric features of different objects. In the point cloud representation, objects are detected correctly with only an exception of false positive prediction in the second scene. With the increase of spatial resolution in Cartesian images, the confidence of predictions of ball, circle cage, and square cage increases, and the detection results of these regularly shaped objects improve. However, the prediction of irregular objects, such as the human body, becomes worse when high resolutions are used. The reason is that a high spatial resolution results in a small image size, and many details will be lost in the Cartesian image.

Quantitative comparison: Tables 2 and 3 show the overall and classwise evaluations of the detection results from different data representations on the given datasets, respectively. mAP@[.5, .95] means calculating mAP with intersection over union (IOU) between 0.5 and 0.95, mAP@0.5 means calculating mAP with IOU greater than 0.5, and mAP@0.75 means calculating mAP with IOU greater than 0.75. S, M, and L indicate the size of the region proposals (small, medium, and large) for the objects in the image. It can be seen that detection results with polar image and point cloud representations share a similar performance. Cartesian image (1 cm) representation achieves +0.6% compared with polar image and point cloud representations. The reason is that there is no distortion in the Cartesian image, and more details of objects can be preserved.

In addition, it can be observed that the accuracy shows a clear downward trend in the performance of mAP@[.5, .95], mAP@0.5, and mAP@0.75 with increasing spatial resolution of the Cartesian image. The reason is that different spatial resolutions lead to different image sizes. Similar to the raster processing in the field of point clouds, the larger the spatial size of a grid is, the less the amount of data obtained, and consequently, the outline of the object becomes more indistinctive.

It can also be observed that the precisions of small and medium predictions show a clear upward trend as the spatial resolution of the Cartesian image increases. For instance, compared with Cartesian image (1 cm), the mAP(M) of Cartesian image (5 cm) increases to 70.1%. The reason is that the size of the image generated with a resolution of 1 cm is larger than that generated with a resolution of 5 cm. Detection in a large image with small proposals will have a relatively low accuracy. In the extreme case, where the proposal size is too large for the generated Cartesian image, the mAP becomes invalid. Among these different approaches, the mAP[.5, .95] of the Cartesian image (1 cm) increases to 51.4% and achieves the best performance of object detection.

5.5 Comparison with different fusion strategies

Qualitative comparison: Fig. 6 illustrates the typical detection results with different feature fusion strategies of sonar data on the given datasets. The results from the Faster R-CNN with a single polar image or point cloud input are also presented to give a better comparison. Fig. 6 shows that the overall performance of our feature fusion methods outperforms

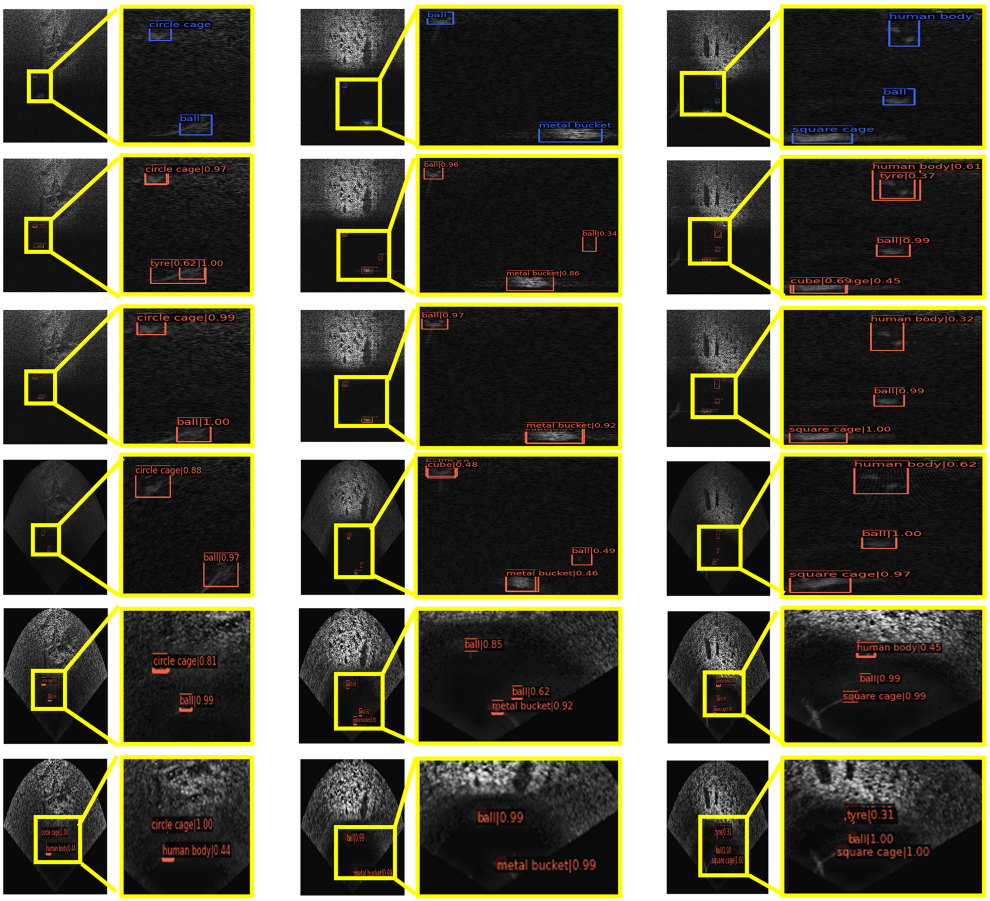

Fig. 5 Typical detection results of three scenes by using different data representations

From top to bottom: ground truth annotation, polar image, point cloud, Cartesian image with a resolution of 1 cm, Cartesian image with a resolution of 5 cm, and Cartesian image with a resolution of 10 cm

Table 2 Mean average precision of approaches with different data representations

Representation	Mean average precision (%)					
	mAP@[.5, .95]	mAP@0.5	mAP@0.75	mAP(S)	mAP(M)	mAP(L)
Polar image	50.8	93.8	**49.7**	18.5	51.0	45.5
Point cloud	50.8	**95.4**	48.4	24.4	50.9	**46.2**
Cartesian image (1 cm)	**51.4**	94.7	49.0	20.0	51.4	43.1
Cartesian image (5 cm)	48.2	94.6	40.6	**48.1**	**70.1**	–
Cartesian image (10 cm)	43.9	92.3	31.4	43.9	–	–

Best results are in bold

Table 3 Classwise precision of approaches with different data representations

Representation	Classwise precision (%)							
	Ball	Circle cage	Human body	Square cage	Cube	Cylinder	Tyre	Metal bucket
Polar image	**52.1**	52.3	52.2	46.9	51.9	44.3	**53.5**	53.4
Point cloud	51.9	52.4	**53.0**	47.3	53.6	46.2	51.7	50.5
Cartesian image (1 cm)	50.4	**53.7**	50.9	**51.1**	**56.2**	**47.7**	45.0	**56.2**
Cartesian image (5 cm)	50.5	48.2	46.7	46.2	50.6	38.7	49.1	55.4
Cartesian image (10 cm)	42.7	47.3	49.3	42.3	39.2	33.5	46.7	50.1

Best results are in bold

that of detectors with a single representation input. In the first two scenes, the precision of irregularly shaped objects (such as human body and metal bucket) in the RPN fusion method is higher than that in the other two fusion strategies. However, in the third scene, it can be observed that the precision of regularly shaped objects (such as ball) is almost the same among the three fusion methods. Noting those regions highlighted by yellow squares, it can be seen that the RPN fusion method performs best among our three feature fusion strategies.

Quantitative comparison: Tables 4 and 5 show the overall and classwise precision of detection approaches with different feature fusion strategies on the given datasets. In this experiment, we compare three feature fusion strategies with two baseline approaches (Faster R-CNN with polar image representation and point cloud representation). From Table 4, it can be seen that our proposed input fusion, RPN fusion, and R-CNN fusion strategies outperform the baseline methods by +1.0% mAP, +1.4% mAP, and +0.7% mAP on average, respectively. Therefore, each of our proposed feature fusion strategies has the potential to improve the overall accuracy of object detection with sonar data. In RPN fusion, intensity features and geometric features are learned from polar image and point cloud representations separately, alleviating the difficulty of robust feature learning; then, features are fused and fed into RPN and R-CNN for final predictions. It seems that the undistorted geometric features not only

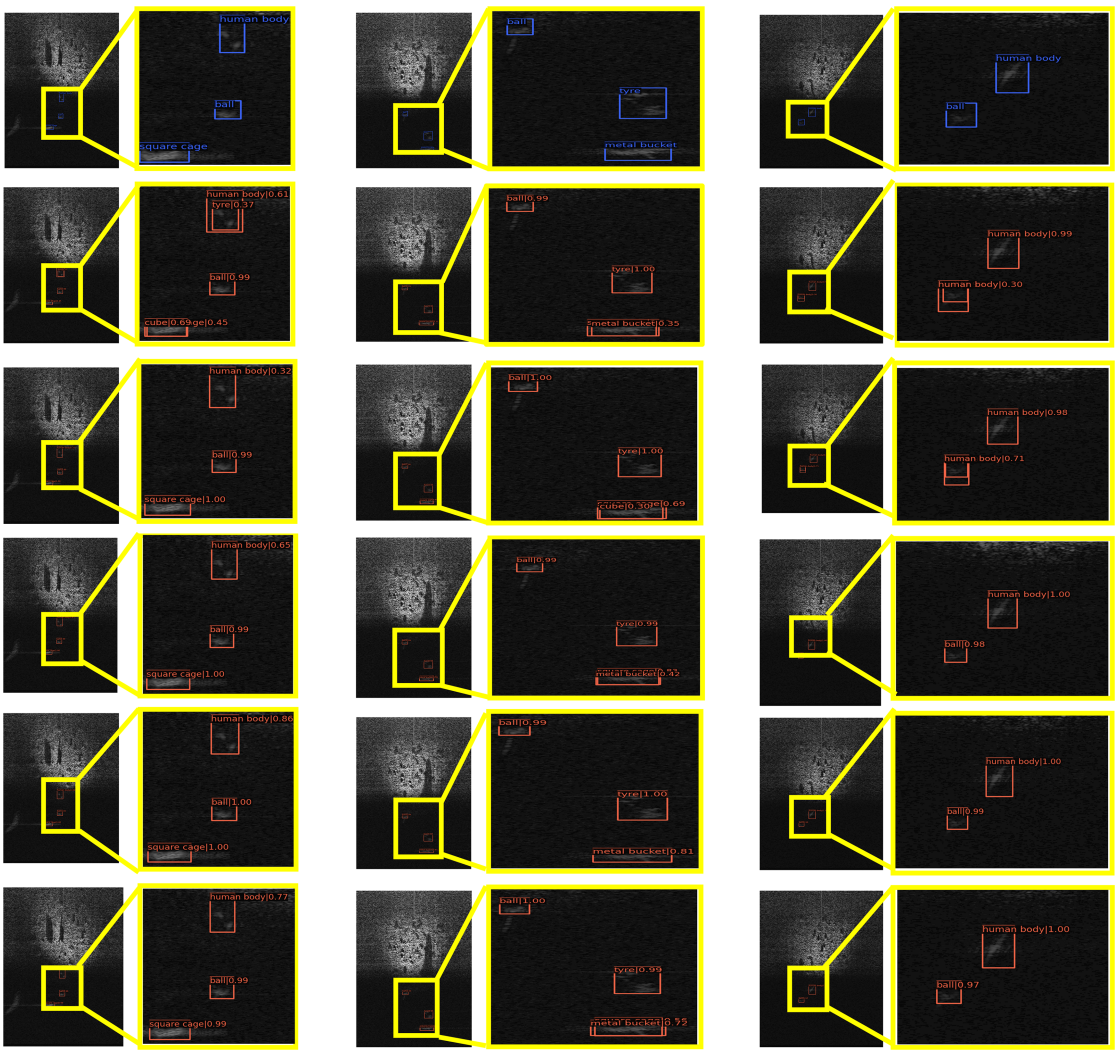

Fig. 6 Typical detection results of three scenes by using different feature fusion strategies

From top to bottom: ground truth annotation, polar image, point cloud, input fusion, RPN fusion, and R-CNN fusion. References to color refer to the online version of this figure

make a better classification but also help to generate better object proposals. Compared with RPN fusion, the input fusion strategy shows the second-best performance. Fused features are also fed into both RPN and R-CNN, resulting in a better prediction. However, the fused input data make the model struggle to learn effective feature representations, resulting in a −0.4% mAP on average. For R-CNN fusion, since the geometric features are fused only for R-CNN, the overall performance is relatively low. However, the mAP of the R-CNN fusion still shows a +0.7% mAP compared with detection approaches without feature fusion.

Table 6 shows the comparison results among different fusion operations. In this table, input fusion, RPN fusion, and R-CNN fusion indicate feature fusion using the concatenation operation; RPN fusion_add and R-CNN fusion_add indicate feature fusions using the add operation. AR_{1000} means that the maximum number of object proposals that can be detected in the image is 1000. In this experiment, we compare five approaches in terms of floating-point operations per second (FLOPs), number of parameters, AR, and mAP. From Table 6, it can be seen that RPN fusion and R-CNN fusion strategies outperform RPN fusion_add and R-CNN fusion_add strategies by +0.7% mAP and +0.7% mAP on average, respectively. Moreover, RPN fusion and R-CNN fusion strategies with concatenation operation show +0.2% AR and +0.3% AR on average compared with RPN fusion_add and R-CNN fusion_add strategies, respectively. Therefore, the feature fusion strategy with the concatenation operation has more potential to improve the overall accuracy of object detection with sonar data than the add operation. However, the FLOPs and the number of parameters of the RPN fusion and R-CNN fusion are much more than those of the RPN fusion_add and

Table 4 Mean average precision of approaches without/with feature fusion

Representation	Mean average precision (%)					
	mAP@[.5, .95]	mAP@0.5	mAP@0.75	mAP(S)	mAP(M)	mAP(L)
Polar image*	50.8	93.8	49.7	18.5	51.0	45.5
Point cloud*	50.8	95.4	48.4	24.4	50.9	46.2
Input fusion#	51.8	95.6	50.0	24.4	51.8	48.7
RPN fusion#	**52.2**	**95.8**	**51.4**	**26.1**	**52.1**	**49.3**
R-CNN fusion#	51.5	94.3	50.5	24.4	51.6	47.5

Best results are in bold. * without feature fusion; # with feature fusion

Table 5 Classwise precision of approaches without/with feature fusion

Representation	Classwise precision (%)							
	Ball	Circle cage	Human body	Square cage	Cube	Cylinder	Tyre	Metal bucket
Polar image*	52.1	52.3	52.2	46.9	51.9	44.3	53.5	53.4
Point cloud*	51.9	**52.4**	**53.0**	47.3	**56.3**	46.2	51.7	50.5
Input fusion#	52.4	50.2	52.2	**48.0**	54.9	**47.7**	**56.5**	52.4
RPN fusion#	**52.5**	52.3	52.9	47.7	55.9	47.1	55.0	53.9
R-CNN fusion#	52.2	51.8	51.4	**48.0**	54.1	45.3	54.1	**54.9**

Best results are in bold. * without feature fusion; # with feature fusion

Table 6 Comparison of different fusion operation approaches with feature fusion

Representation	FLOPs (G)	Number of parameters (M)	AR_{1000}	mAP@0.5	mAP@0.7	mAP@[.5, .95]
Input fusion*	**206.7**	**41.4**	**59.5%**	95.6%	50.0%	51.8%
RPN fusion*	411.9	81.2	59.4%	**95.8%**	**51.4%**	**52.2%**
R-CNN fusion#	361.7	80.6	58.3%	94.3%	50.5%	51.5%
RPN fusion_add#	348.8	67.8	59.2%	95.3%	50.3%	51.5%
R-CNN fusion_add#	348.8	67.8	58.0%	94.2%	48.3%	50.8%

Best results are in bold. * without feature fusion; # with feature fusion; G: giga; M: million

R-CNN fusion_add strategies. The feature fused by the concatenation operation can improve the accuracy of object detection, but this operation will lead to an increase in computation quantity and parameter quantity. The FLOPs and the number of parameters of the input fusion are the lowest among the five approaches because it is a data-level fusion strategy.

5.6 Impact of feature fusion on proposal generation

From the previous experiments, we observe that the best fusion strategy is RPN fusion instead of R-CNN fusion. It seems that fused features help both region proposal and bounding box predictions in two-stage detectors. Therefore, this experiment focuses on the investigation of how feature fusion affects region proposals. To this end, we compare the accuracy of proposals generated by Faster R-CNN with or without feature fusion. The results are summarized in Table 7. From the table, it can be seen that our input fusion strategy and RPN fusion strategy show +0.6% mAP@[.5, .95] and +1.3% mAP@[.5, .95] on average compared with detectors using polar image representation. It clarifies that our proposed feature fusion methods undoubtedly have the ability to improve the accuracy of region proposals. In addition, RPN fusion still shows better performance than the input fusion strategy. This is because RPN fusion uses two separate ResNet-50 structures to learn the intensity features and geometric features from the polar image and point cloud representations, respectively. Each of the two ResNet-50 structures responses for one kind of feature learning. This releases the burden of learning robust features.

The comparison of the AR of proposals generated by Faster R-CNN with or without feature fusion is shown in Table 8. From the table, AR_{100} has the same value as AR_{300} and AR_{1000}. AR_{100} means the maximum number of object proposals that can be detected in the image is 100. Since there are only a few objects in the sonar image, the number of 100 object proposals is sufficient to predict the location of objects in the image. The value of AR(S) is much lower than AR(M) and AR(L). Since the size of the objects in the sonar image is larger than that in the normal image, the performances of AR(M) and AR(L) are better than that of AR(S). Moreover, it can be seen that our input fusion strategy and RPN fusion strategy show +0.6% AR_{100} and +0.4% AR_{100} on average compared to detectors using polar image representation. In addition, the input fusion shows a slightly better performance on proposal generation than the RPN fusion strategy. This is because the input fusion uses the data of the polar image and the point cloud representations as the input to extract features.

5.7 Extension to modern detectors

To validate the applicability of our feature fusion module, we extend our module to state-of-the-art detectors and make a comparison between the extended and original detectors. Both one-stage detectors (including DDOD, TOOD, YOLOv3, and YOLOX) and two-stage detectors (including sparse R-CNN and

Table 7 Precision of proposal generation without/with feature fusion

Representation	mAP@[.5, .95]	mAP@0.5	mAP@0.75	mAP(S)	mAP(M)	mAP(L)
Polar image*	52.2%	96.6%	50.0%	24.6%	52.2%	55.2%
Point cloud*	52.0%	96.9%	49.6%	32.4%	52.0%	54.2%
Input fusion#	52.8%	**97.0**%	51.1%	32.0%	52.8%	57.8%
RPN fusion#	**53.5**%	**97.0**%	**53.0**%	**34.6**%	**53.3**%	**58.2**%

Best results are in bold. * without feature fusion; # with feature fusion

Table 8 Average recall of proposal generation without/with feature fusion

Representation	AR_{100}	AR_{300}	AR_{1000}	AR(S)	AR(M)	AR(L)
Polar image*	61.1%	61.1%	61.1%	24.4%	61.3%	61.4%
Point cloud*	60.8%	60.8%	60.8%	32.2%	61.0%	60.8%
Input fusion#	**61.7**%	**61.7**%	**61.7**%	32.2%	**61.7**%	**64.9**%
RPN fusion#	61.5%	61.5%	61.5%	**34.4**%	61.6%	62.4%

Best results are in bold. * without feature fusion; # with feature fusion

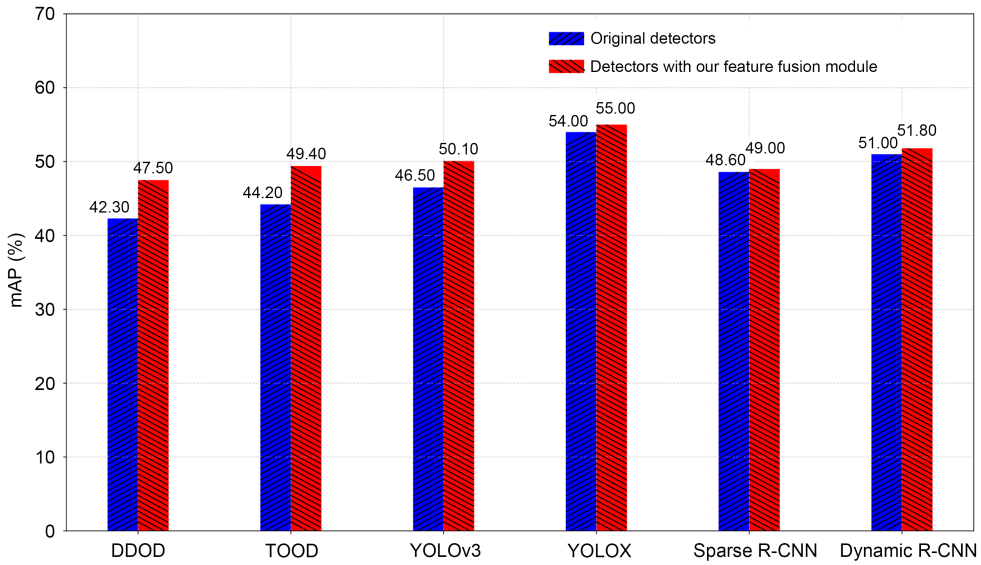

Fig. 7 Comparison of the state-of-the-art detectors without/with our feature fusion module

dynamic R-CNN) are chosen for experiments. The original detectors and our extended detectors are implemented under an open-sourced detection framework MMDetection. Both detectors are trained by the same parameter settings. The experimental results are shown in Fig. 7. It can be seen that (1) both one-stage detectors and two-stage detectors show better performance when extended with our feature fusion module and (2) one-stage detectors with our module show more significant improvement than two-stage detectors. The mAP of the one-stage models with the feature fusion module shows approximately +1.0%–5.2%; the mAP of the two-stage models shows approximately +0.4%–0.8%. Therefore, our feature fusion module can be easily and effectively extended to state-of-the-art detectors and show better performance on this competitive benchmark.

6 Conclusions

This paper focuses on the impact of various sonar data representations on object detection. A summary of different representations for sonar data was first presented, including polar image, Cartesian image, and point cloud. The geometric characteristics of each data representation were analyzed. Two kinds of geometric distortion in sonar data were discussed, including projection distortion and representation distortion. The first distortion is caused by the limitation of sonar sensors, while the second is due to the representation of sonar data. Therefore, a feature fusion framework from different sonar data representations for underwater object detection was proposed. Three feature fusion strategies were presented in the framework to investigate the impacts of feature fusion on different components of the detection pipeline. RPN fusion strategy showed the best performance among all the approaches. In addition, our feature fusion strategies can be easily integrated into other detectors, such as YOLO series. A series of experiments were conducted on a public sonar dataset. Experimental results showed the validity and practicality of our proposed framework and feature fusion strategies.

Contributors

Fei WANG and Jingchun ZHOU designed the research. Fei WANG, Wanyu LI, and Miao LIU processed the data. Fei WANG drafted the paper. Weishi ZHANG helped organize the paper. Fei WANG and Jingchun ZHOU revised and finalized the paper.

Compliance with ethics guidelines

Fei WANG, Wanyu LI, Miao LIU, Jingchun ZHOU, and Weishi ZHANG declare that they have no conflict of interest.

Data availability

Due to the nature of this research, participants of this study did not agree for their data to be shared publicly, so supporting data are not available.

References

Ben Tamou A, Benzinou A, Nasreddine K, 2021. Multi-stream fish detection in unconstrained underwater videos by the fusion of two convolutional neural network detectors. *Appl Intell*, 51(8):5809-5821. https://doi.org/10.1007/s10489-020-02155-8

Bochkovskiy A, Wang CY, Liao HYM, 2020. YOLOv4: optimal speed and accuracy of object detection. https://arxiv.org/abs/2004.10934

Charles RQ, Su H, Mo KC, et al., 2017. PointNet: deep learning on point sets for 3D classification and segmentation. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.77-85. https://doi.org/10.1109/CVPR.2017.16

Chen K, Wang JQ, Pang JM, et al., 2019. MMDetection: open MMLab detection toolbox and benchmark. https://arxiv.org/abs/1906.07155

Chen XL, Mu XQ, Guan J, et al., 2022. Marine target detection based on Marine-Faster R-CNN for navigation radar plane position indicator images. *Front Inform Technol Electron Eng*, 23(4):630-643. https://doi.org/10.1631/FITEE.2000611

Chen ZH, Yang CHY, Li QF, et al., 2021. Disentangle your dense object detector. Proc 29th ACM Int Conf on Multimedia, p.4939-4948. https://doi.org/10.1145/3474085.3475351

Feng CJ, Zhong YJ, Gao Y, et al., 2021. TOOD: task-aligned one-stage object detection. Proc IEEE/CVF Int Conf on Computer Vision, p.3490-3499. https://doi.org/10.1109/ICCV48922.2021.00349

Ge Z, Liu ST, Wang F, et al., 2021. YOLOX: exceeding YOLO series in 2021. https://doi.org/10.48550/arXiv.2107.08430

Ghiasi G, Lin TY, Le QV, 2019. NAS-FPN: learning scalable feature pyramid architecture for object detection. Proc IEEE/CVF Conf on Computer Vision and Pattern Recognition, p.7029-7038. https://doi.org/10.1109/CVPR.2019.00720

Girshick R, 2015. Fast R-CNN. Proc IEEE Int Conf on Computer Vision, p.1440-1448. https://doi.org/10.1109/ICCV.2015.169

Girshick R, Donahue J, Darrell T, et al., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.580-587. https://doi.org/10.1109/CVPR.2014.81

Girshick R, Donahue J, Darrell T, et al., 2016. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans Patt Anal Mach Intell*, 38(1):142-158. https://doi.org/10.1109/TPAMI.2015.2437384

Huang H, Zhou H, Yang X, et al., 2019. Faster R-CNN for marine organisms detection and recognition using data augmentation. *Neurocomputing*, 337:372-384. https://doi.org/10.1016/j.neucom.2019.01.084

Kim J, Yu SC, 2016. Convolutional neural network-based real-time ROV detection using forward-looking sonar image. Proc IEEE/OES Autonomous Underwater Vehicles, p.396-400. https://doi.org/10.1109/AUV.2016.7778702

Kong WZ, Hong JC, Jia MY, et al., 2020. YOLOv3-DPFIN: a dual-path feature fusion neural network for robust real-time sonar target detection. *IEEE Sens J*, 20(7):3745-3756. https://doi.org/10.1109/JSEN.2019.2960796

Lin TY, Dollár P, Girshick R, et al., 2017. Feature pyramid networks for object detection. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.936-944. https://doi.org/10.1109/CVPR.2017.106

Liu D, Cheng F, 2021. SRM-FPN: a small target detection method based on FPN optimized feature. Proc 18th Int Computer Conf on Wavelet Active Media Technology and Information Processing, p.506-509. https://doi.org/10.1109/ICCWAMTIP53232.2021.9674107

Otsu N, 1979. A threshold selection method from gray-level histograms. *IEEE Trans Syst Man Cybern*, 9(1):62-66. https://doi.org/10.1109/TSMC.1979.4310076.

Pu SL, Zhao W, Chen WJ, et al., 2021. Unsupervised object detection with scene-adaptive concept learning. *Front Inform Technol Electron Eng*, 22(5):638-651. https://doi.org/10.1631/FITEE.2000567

Redmon J, Farhadi A, 2018. YOLOv3: an incremental improvement. https://doi.org/10.48550/arXiv.1804.02767

Redmon J, Divvala S, Girshick R, et al., 2016. You Only Look Once: unified, real-time object detection. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.779-788. https://doi.org/10.1109/CVPR.2016.91

Ren SQ, He KM, Girshick R, et al., 2015. Faster R-CNN: towards real-time object detection with region proposal networks. Proc 28th Int Conf on Neural Information Processing Systems, p.91-99.

Song Y, He B, Liu P, 2021. Real-time object detection for AUVs using self-cascaded convolutional neural networks. *IEEE J Oceanic Eng*, 46(1):56-67. https://doi.org/10.1109/JOE.2019.2950974

Sun PZ, Zhang RF, Jiang Y, et al., 2021. Sparse R-CNN: end-to-end object detection with learnable proposals. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.14449-14458. https://doi.org/10.1109/CVPR46437.2021.01422

Tian MJ, Li XL, Kong SH, et al., 2022. A modified YOLOv4 detection method for a vision-based underwater garbage cleaning robot. *Front Inform Technol Electron Eng*, 23(8):1217-1228. https://doi.org/10.1631/FITEE.2100473

Valdenegro-Toro M, 2016. Object recognition in forward-looking sonar images with convolutional neural networks. Proc OCEANS MTS/IEEE Monterey, p.1-6. https://doi.org/10.1109/OCEANS.2016.7761140

Wang Z, Guo JX, Huang WZ, et al., 2022. Side-scan sonar image segmentation based on multi-channel fusion convolution neural networks. *IEEE Sens J*, 22(6):5911-5928. https://doi.org/10.1109/JSEN.2022.3149841

Yang GY, Wang ZY, Zhuang SN, 2021. PFF-FPN: a parallel feature fusion module based on FPN in pedestrian detection. Proc Int Conf on Computer Engineering and Artificial Intelligence, p.377-381. https://doi.org/10.1109/ICCEAI52939.2021.00075

Zhang HK, Chang H, Ma BP, et al., 2020. Dynamic R-CNN: towards high quality object detection via dynamic training. Proc 16th European Conf on Computer Vision, p.260-275. https://doi.org/10.1007/978-3-030-58555-6_16

Zhou JC, Zhang DH, Ren WQ, et al., 2022a. Auto color correction of underwater images utilizing depth information. *IEEE Geosci Remote Sens Lett*, 19:1504805. https://doi.org/10.1109/LGRS.2022.3170702

Zhou JC, Yang TY, Chu WS, et al., 2022b. Underwater image restoration via backscatter pixel prior and color compensation. *Eng Appl Artif Intell*, 111:104785. https://doi.org/10.1016/j.engappai.2022.104785