

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,000

Open access books available

148,000

International authors and editors

185M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Chapter

Marine Robotics 4.0: Present and Future of Real-Time Detection Techniques for Underwater Objects

Meng Joo Er, Jie Chen and Yani Zhang

Abstract

Underwater marine robots (UMRs), such as autonomous underwater vehicles, are promising alternatives for mankind to perform exploration tasks in the sea. These vehicles have the capability of exploring the underwater environment with onboard instruments and sensors. They are extensively used in civilian applications, scientific studies, and military missions. In recent years, the flourishing growth of deep learning has fueled tremendous theoretical breakthroughs and practical applications of computer-vision-based underwater object detection techniques. With the integration of deep-learning-based underwater object detection capability on board, the perception of underwater marine robots is expected to be enhanced greatly. Underwater object detection will play a key role in Marine Robotics 4.0, i.e., Industry 4.0 for Marine Robots. In this chapter, one of the key research challenges, i.e., real-time detection of underwater objects, which has prevented many real-world applications of object detection techniques onboard UMRs, is reviewed. In this context, state-of-the-art techniques for real-time detection of underwater objects are critically analyzed. Futuristic trends in real-time detection techniques of underwater objects are also discussed.

Keywords: underwater marine robots, deep learning, real-time object detection

1. Introduction

In the age of Industry 4.0, revolutions based on artificial intelligence have increased by leaps and bounds in various sectors [1–3]. In the community of marine science and engineering, many underwater exploration tasks are usually executed by Underwater Marine Robots (UMRs), such as Remotely Operated Underwater Vehicles (ROVs) and Autonomous Underwater Vehicles (AUVs), as shown in **Figure 1**. These marine robots have significantly overcome many difficulties in underwater exploration tasks thanks to their distinct capability of operating round the clock. As a matter of fact, they have been widely used in the community of marine science and engineering extensively.

These UMRs, which are available in different shapes and sizes, are capable of performing a wide variety of tasks and are widely employed in many sectors. In the

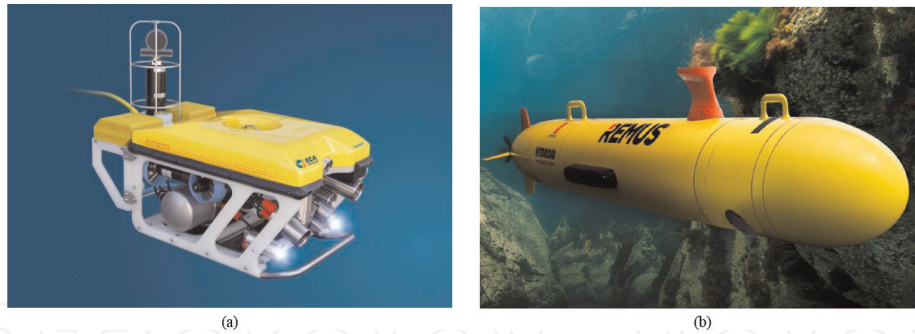


Figure 1. Underwater marine robots: (a) remotely operated underwater vehicle (ROV), (b) autonomous underwater vehicle (AUV). (images from the internet).

civilian sector, UMRs are used for aquaculture, such as providing important information for feeding, surveillance and security, and early warning of diseases [4]. Furthermore, UMRs have been exploited in seafood collection, e.g., picking holothurian, sea urchin, scallop, and other marine products, and have made significant contributions to the economy [5]. UMRs are also a promising choice to perform maintenance and cleaning works on underwater hulls, which is important to maintaining health conditions of a ship [6]. In other applications, UMRs have been employed for scientific research of the ocean, including ocean observation, underwater inspection, and monitoring of marine ecosystems. Furthermore, UMRs have been employed in the military and security sector for specific missions, such as surveillance, underwater monitoring, mine detection, and countermeasure [7].

Superior perception is highly desired for UMRs to perform assigned tasks successfully. Cameras and sonars are two kinds of sensors that UMRs typically rely on for environmental perception. There are distinct advantages and disadvantages in employing cameras and sonars for exploration tasks. However, it should be noted that both optical and sonar images share the same technology stack of processing. In recent years, flourishing development of artificial intelligence, especially deep learning, has fueled tremendous theoretical breakthroughs and practical applications [8, 9]. On one hand, development of deep learning is inseparable from exponential growth of data, which has spawned a lot of research works related to data mining [10–12]. On the other hand, artificial intelligence has been successfully applied to various fields, such as smart city [13, 14] and intelligent transportation [15, 16]. However, to our knowledge, most of these applications are on the land; underwater applications with artificial intelligence have not been fully explored yet. In the age of Industry 4.0, underwater object detection is one of the important applications that employ artificial intelligence techniques. Object detection is crucial for environmental perception which resolves around “what objects are located at where”. With the adoption of deep-learning-based underwater object detection techniques on board, the perception capability of UMRs is expected to be enhanced greatly.

However, due to the constraints of existing technology, UMRs can only be equipped with embedded computing platforms, such as the Raspberry Pi, as shown in **Figure 2-(a)**, which has extremely limited computing power. A more high-end computing platform is the NVIDIA Jetson, provided by NVIDIA Corporation, and is shown in **Figure 2-(b)**. However, it also has limited computing power.

In order to circumvent the scarcity of limited computing resources, programs executed on such platforms must be significantly lightweight and efficient. However, existing deep learning models are usually computationally expensive. According to [17],

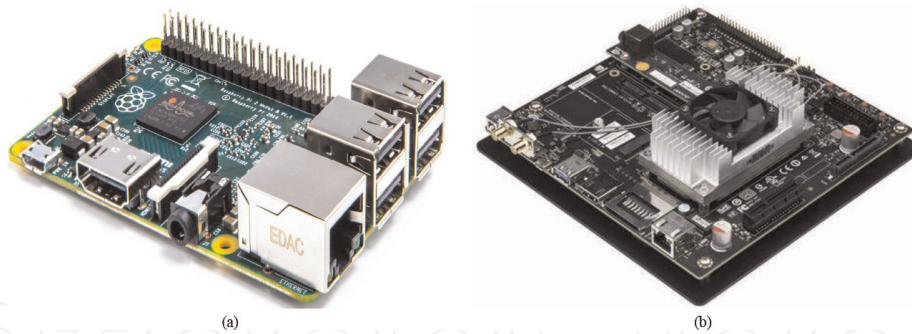


Figure 2. Embedded computing platforms for UMRs: (a) raspberry pi, (b) NVIDIA Jetson. (images from the internet).

a standard ResNeXt-50 has about 25.0×10^6 parameters and 4.2×10^9 FLOPS on 8 GPUs of NVIDIA M40. This demonstrates that deep learning models are not suitable for deployment on embedded platforms, and they pose a critical research challenge for underwater object detection. In order to circumvent this limitation, deep-learning-based underwater object detection algorithms should be efficient so that they are implementable. As such, viable real-time detection techniques of underwater objects are highly desired.

Real-time detection of underwater objects, as one of the key challenges in Marine Robotics 4.0, i.e., Industry 4.0 for Marine Robots, is critically reviewed in this chapter. To facilitate a full understanding of the subject matter, we have comprehensively and systematically reviewed and analyzed related techniques for real-time detection of underwater objects. Futuristic trends in real-time detection of underwater objects are also discussed.

2. Preliminaries

Underwater object detection not only needs to recognize all objects of interest, but also locate their positions in underwater images. As shown in **Figure 3**, position

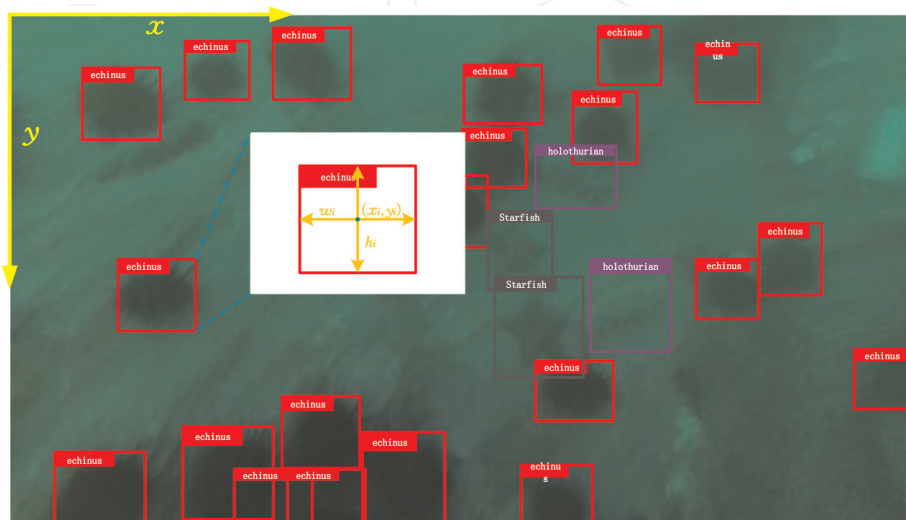


Figure 3. Underwater object detection. The detection result is presented by a bounding box with a label on it, where (x_i, y_i) denotes the coordinates of i -th object, and (w_i, h_i) denotes the width and height of box. (x, y) is the frame of axes for detection results, with origin at the top left corner of the image (image from the DUO dataset [18]).

information is generally represented by a rectangular bounding box defined by (x_i, y_i, w_i, h_i) , where (x_i, y_i) denotes center-point coordinates of i -th object, and (w_i, h_i) is the width and height of the box. The frame of axes (x, y) for the detection result is presented in yellow with the origin (0 – indexed) at the top left corner of the image. In addition, category label of the object is attached to the bounding box.

The underwater object detection problem can be formulated as follows:

$$X \xrightarrow{f(\theta)} \{ (p_i, c_i, x_i, y_i, w_i, h_i) \mid i \in (1, \dots, N) \} \quad (1)$$

where $f(\theta)$ indicates an object detector that is based on any neural networks parameterized by θ . The function $f(\theta)$ takes an image X as its input, and outputs N predictions for objects in that image. The term N denotes the number of objects detected in that image. Each prediction contains a confidence indicator p_i , the category label c_i that the object belongs to, and the position information encoded in the bounding box (x_i, y_i, w_i, h_i) . It is well-known that underwater object detection can provide valuable information for semantic understanding of the underwater environment, and it is a fundamental research topic in the community of marine science and engineering.

3. State of the arts using deep learning

Deep-learning-based object detection methods are typically associated with large model sizes, are usually sophisticated, and cannot match real-time requirements when applied on UMR platforms. However, as far as actual use of underwater object detection in shallow water for mission execution is concerned, real-time detection is the most important prerequisite. As such, deep-learning-based detectors for UMR platforms must be as efficient as possible. The key idea that underpins the lightweight model is to create an elegant and practical lightweight network architecture while achieving excellent performance. In the field of object detection, this is a never-ending quest for research excellence.

The development of underwater object detection techniques suitable for real-time performance has a long history. In this context, we will review representative literatures on real-time detection techniques, which can be categorized into three categories, namely two-stage detectors, one-stage detectors, and anchor-free detectors.

3.1 Two-stage detectors

The R-CNN (Regions with CNN features) for object detection [19] is the first successful two-stage deep learning object detector developed in the object detection community, but it is not suitable for real-time detection. As illustrated in **Figure 4**, it begins with a selective search [20] to extract a collection of object candidates (region proposals). Next, to extract features, each proposal is re-scaled to a fixed-size picture and input to a Convolutional Neural Network (CNN) which is pre-trained on ImageNet [21]. Finally, linear SVM classifiers are utilized to predict the existence of an object and to distinguish object types inside each region based on the features extracted by CNN.

However, the R-CNN applies CNN to each potential region for extracting features. There are a lot of overlaps, resulting in many redundant computations and resulting in very sluggish detection speed. In order to alleviate this problem, the Fast R-CNN [22]

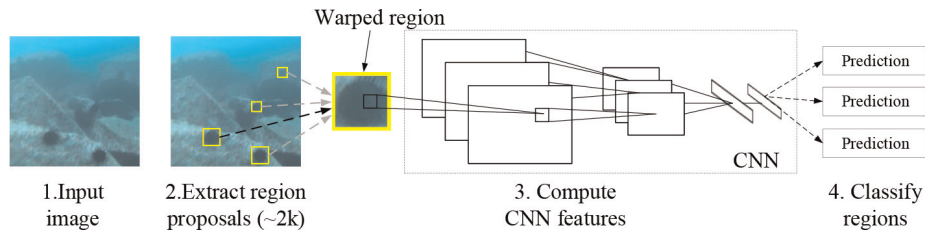


Figure 4. Network architecture of R-CNN, where CNN features extraction is applied on each candidate region (image from [19]).

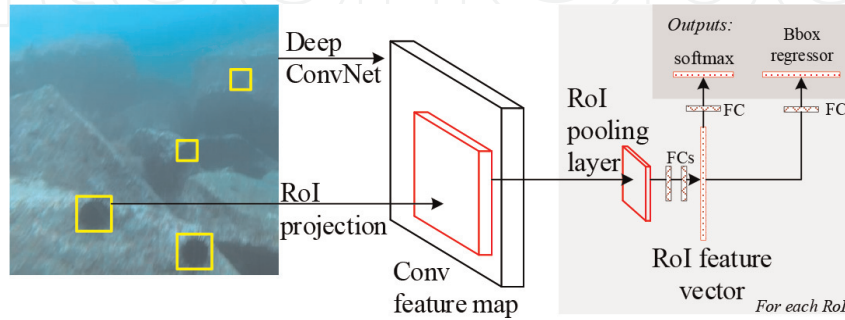


Figure 5. Network architecture of fast R-CNN, where features extraction is applied to the entire image only once (figure from [22]).

employ CNN to extract features from the entire picture only once and obtains features for each candidate region via a Region of Interest (ROI) pooling operation, as illustrated in **Figure 5**. In comparison with the R-CNN, it achieves superior accuracy on various benchmark datasets but improve image processing speed by 146 times under the same conditions and reduces the training time by 9 times.

In [23], Fast R-CNN is trained to detect underwater objects in sonar images. By using Bayesian optimization, which follows the Automated Machine Learning (AutoML) principle, the hyperparameter configuration of Fast R-CNN was set to be optimum. In [24], encouraged by the powerful detection performance obtained by CNNs on generic datasets, Fast R-CNN is applied to a domain-specific underwater environment for accurate identification and recognition of fish. At the time, Fast R-CNN was widely used in underwater object detection.

However, Fast R-CNN continues to employ complicated selective search approach for the generation of candidate region proposals, which turns out to be time-consuming. Ren *et al.* [25] propose a Region Proposal Network (RPN) that predicts candidates directly from the shared feature maps, as illustrated in **Figure 6**. This new

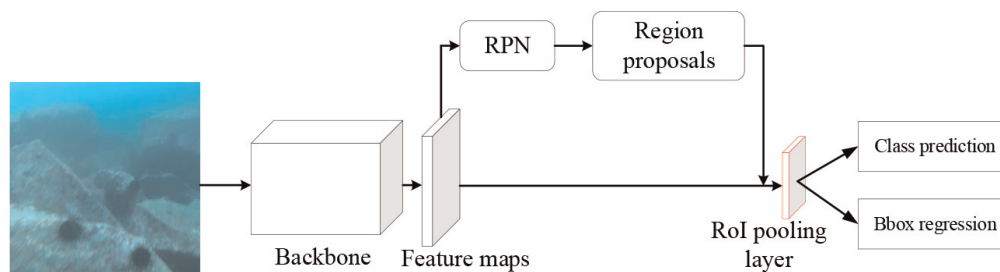


Figure 6. Network architecture of faster R-CNN, where region proposal network (RPN) is proposed for extraction of region candidates based on the shared feature maps (figure from [25]).

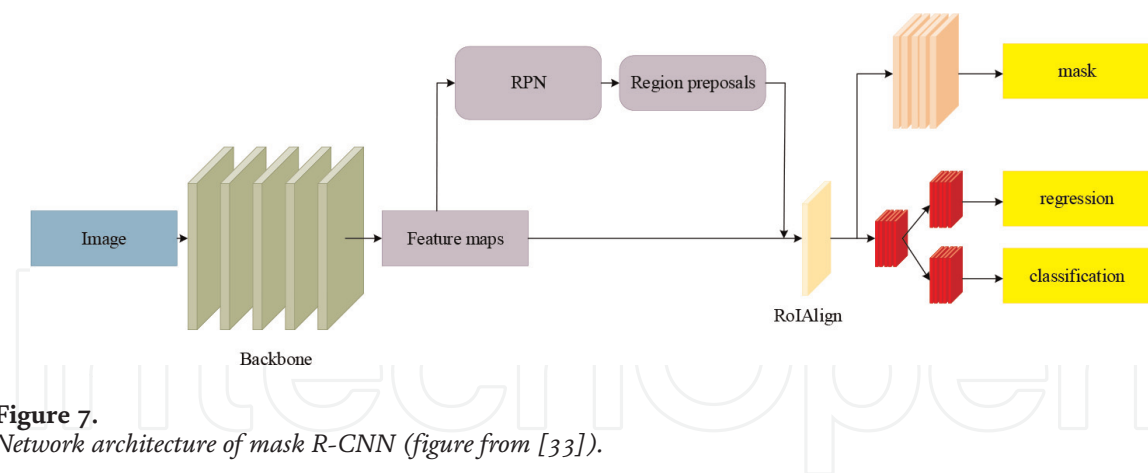


Figure 7. Network architecture of mask R-CNN (figure from [33]).

architecture is dubbed Faster R-CNN. Typical processing time of each picture in selective search is around 1 – 2s, but the RPN requires only approximately 10ms, resulting in tremendous increase in detection speed.

In [26], for faster detection and recognition of fishes by sharing CNNs with objectness learning, the backbone of Faster R-CNN is substituted with a pre-trained ZFNet [27]. In [28], the Faster R-CNN is enhanced to detect underwater organisms, which is exposed to many challenges, such as low-quality images, varying sizes or forms, and overlapping or occlusion objects. The backbone is replaced with ResNet [29]. For multi-scale feature fusion, the BiFPN architecture proposed in [30] is adopted. Finally, to minimize the amount of redundant bounding boxes in the training data, the EIoU (Effective IoU) [31] is utilized to replace IoU. On the URPC2018 dataset [32], the accuracy of the modified Faster R-CNN is 8.26% higher than the original version of Faster R-CNN. Faster R-CNN has dominated underwater object detection for a long time.

After that, the Faster R-CNN is extended by Mask R-CNN, which adds a branch for predicting an object mask in parallel with the current branch for bounding box identification [33], as illustrated in **Figure 7**. It can recognize objects in a picture quickly while also creating a high-quality segmentation mask for each instance. Thanks to the benefits of multi-task learning, Mask R-CNN outperforms all existing single-model entries on a wide range of computer vision tasks by adding only a minor overhead to Faster R-CNN.

In [34], to identify and separate underwater objects from forward-looking sonar pictures, a modified Mask R-CNN is proposed by replacing the Resnet backbone. The modified Mask R-CNN reduces the number of network parameters significantly while maintaining the detection performance. It is suitable for real-time detection. The Mask R-CNN is also utilized to identify common fishery species (yellowfin bream, *Acanthopagrus australis*) for animal movement studies to assess ecosystem health, comprehend ecological dynamics, and address management and conservation problems [35].

In this section, we have reviewed several representative two-stage detectors. By discarding the complicated module with high computational complexity, the detection speed improves significantly.

3.2 One-stage detectors

The aforementioned detectors are members of the R-CNN family of two-stage algorithms, which frame the detection as a “coarse-to-fine” process [36]. They are well-known for their excellent detection precision but low detection speed [37].

Another family of detectors, the YOLOs (You Only Look Once) [38–40] foregoes extraction of candidate region proposals and predicts detection outcomes directly from shared feature maps of CNN. These approaches are also known as one-stage detectors. The inference time is reduced to 50 ms by using a one-stage approach while maintaining relatively high accuracy, whereas other competitive models need more than 200 ms. This is a bigger leap forward in terms of real-time detection.

In 2015, R. Joseph *et al.* proposed the YOLO detector [38]. The key idea of the YOLO detector is to split the picture into grids and predict bounding boxes and probabilities for each cell by using a CNN directly. As illustrated in **Figure 8**, it splits the picture into a $S \times S$ grid, and predicts B bounding boxes with 1 confidence per box, and C class probabilities for each grid cell. The final predictions are encoded in a $S \times S \times (B * 5 + C)$ output tensor directly by the convolutional network.

In [41], a YOLO detector is trained on generating realistic sonar pictures by GANs [42] for underwater object recognition, which is required to automate activities like shipwreck investigation, mine clearance, and landmark-based navigation. Later, R. Joseph produced a series of enhancements to YOLO and offered its v2 and v3 editions [39, 40], which improved detection accuracy while maintaining fast detection speed.

YOLO v2 is an enhanced version of YOLO, with batch normalization [43], removal of fully connected layers, and the use of excellent anchor boxes acquired using k-means and multiscale training, in addition to the custom GoogLeNet network [44] being replaced by the simpler DarkNet19 network. In [45], YOLO v2 is presented as a coarse pre-detection module in the pipeline of rotational object detection using forward-looking sonar in underwater applications, where detection results of YOLO v2 are clipped from the sonar picture and fed to a more fine-grained detector.

The most extensively utilized approach in the industry is YOLO v3, where the Darknet-53 backbone harvests features, and three detection heads fuse different scale feature maps for object detection with different sizes. In [46], experiments to detect and classify sea cucumber, scallop, and sea urchin from underwater photos were carried out, and the results demonstrate that the YOLO v3 algorithm has a *mAP* value 6.4% higher and a recall rate 13.9% higher than Faster R-CNN. Furthermore, YOLO v3 has a detection speed of 20 frames per second, which is 12 frames per second faster

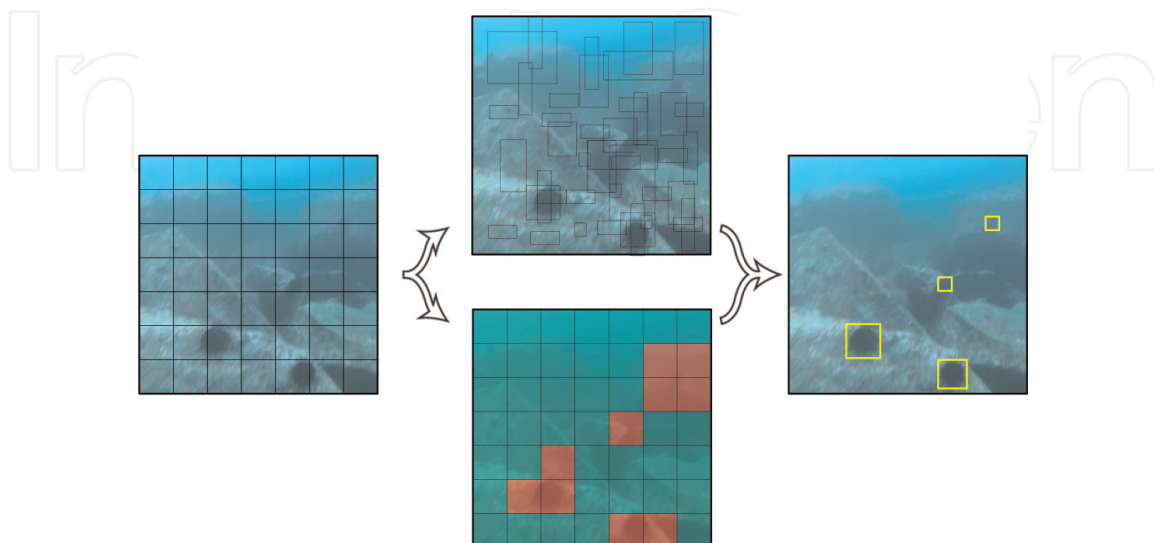


Figure 8. The YOLO detector is depicted as a regression issue in this picture. It splits the picture into a $S \times S$ grid, predicting B bounding boxes with 1 confidence per box, and C class probabilities for each grid cell. The tensor $S \times S \times (B * 5 + C)$ encodes these predictions.

than Faster R-CNN. In [47], YOLOv3 is integrated into an underwater manipulator (BlueROV2) to identify objects for grabbing.

YOLO v4 [48] has put to the test a large variety of strategies that are supposed to enhance accuracy of a CNN. Finally, it combines techniques such as Weighted-Residual-Connections [30], Cross-Stage-Partial-Connections [49], Cross mini-Batch Normalization [50], Self-adversarial-training [51], Mish activation [52], Mosaic data augmentation, DropBlock regularization [53], and CIoU loss [54] to achieve optimal object detection speed and accuracy. In [55], to construct a lightweight underwater object detector, YOLO v4 is combined with a multi-scale attentional feature fusion module. For real-time performance, it also replaces the CSPDarknet53 backbone [49] with MobileNet [56].

From two-stage detectors to one-stage detectors, the YOLO series has gained a qualitative leap in real-time underwater object detection. Leveraging meticulous design in the network architecture, one-stage detectors will improve performance and detection speed significantly.

3.3 Anchor-free detectors

Another significant paradigm shift in real-time object detection is from anchor-based to anchor-free techniques. The majority of the aforementioned approaches are anchor-based, whereby anchors of various sizes and aspect ratios are established on the picture, allowing object detection to predict related offsets. The usage of anchor boxes has long been thought to be a secret to successful detection [57].

Thousands of pre-defined anchor boxes are placed on the picture in anchor-based techniques, and the model predicts which anchor box will respond to the ground-truth. However, the generation of anchors via region proposal network [25] or k-means clustering [40] is a time-consuming process. Undoubtedly, anchor-based approaches will also result in duplicate predictions, necessitating the use of a non-maximum suppression algorithm [58] to eliminate undesirable outcomes. Unfortunately, non-maximum suppression is also an expensive operation, which slows down the speed of object detection significantly.

Anchor-free detectors aim to eliminate expensive operations that are related to anchor mechanism. Without the necessity for non-maximum suppression, anchor-free techniques remove the computation load raised by anchors and regress the category and position of the object directly by convolutional networks [57, 59]. They remove anchor-related computations like anchor clustering, allowing for even more real-time efficiency.

One of the most canonical anchor-free detectors, CenterNet [59], represents an object as a single point – the center-point of its bounding box. As illustrated in **Figure 9**, the neural network predicts the center-point heatmaps \hat{Y} , offsets \hat{O} and sizes \hat{S} of bounding boxes. By using key point estimation, CenterNet determines the center point of objects and regresses all other object parameters, such as size. The bounding box at position (x_i, y_i) may be generated from predictions at inference as follows:

$$\left(\hat{x}_i + \delta\hat{x}_i - \hat{w}_i\hat{y}_i + \delta\hat{y}_i - \hat{h}_i, \hat{x}_i + \delta\hat{x}_i + \hat{w}_i\hat{y}_i + \delta\hat{y}_i + \hat{h}_i, \right) \quad (2)$$

where $(\delta\hat{x}_i, \delta\hat{y}_i) = \hat{O}_{\hat{x}_i, \hat{y}_i}$ is the offset prediction and $(\hat{w}_i, \hat{h}_i) = \hat{S}_{\hat{x}_i, \hat{y}_i}$ is the size prediction. Without the use of IoU-based non-maxima suppression or other post-processing operations, all outputs are generated directly from key point estimations.

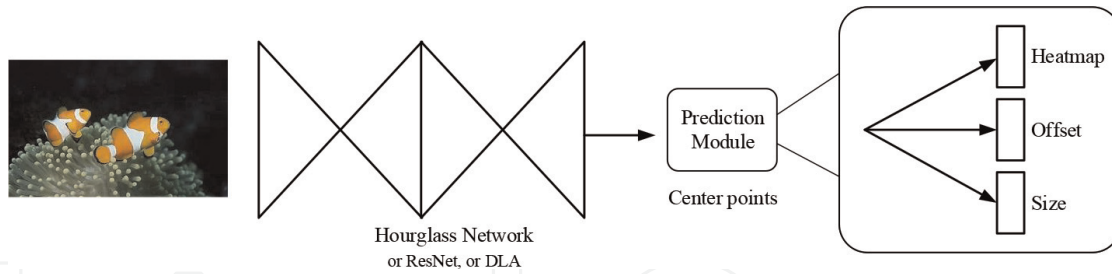


Figure 9.
Architecture of CenterNet (figure from [59]).

Contrary to complicated computation experienced in anchor mechanism, the detection speed of anchor-free models is improved over one-stage detectors significantly while maintaining superior detection accuracy. Anchor-free models have become the de facto solution for real-time detection [57]. For example, the AquaNet [5] and MRF-Net [60] are improved based on the anchor-free model termed CenterNet for underwater detection, and the efficiency and effectiveness are both verified by comprehensive experiments.

4. Futuristic trends

The limited computing resources of UMRs is the main factor that prevents the deployment of deep-learning-based models for real-time detection in underwater environment. Meanwhile, difficulties of communication in underwater environment prevent the possibility of exploring other cloud computing solutions. As a consequence, reducing model size seems to be the only feasible method moving forward.

In the literatures, the two strategies to achieve real-time underwater object detection, namely lightweight network design and model compression, have been proposed. Lightweight network design aims at developing some effective low-complexity network architecture, while model compression attempts to remove redundant parameters of a pre-trained model.

4.1 Lightweight network design

In the development of deep learning algorithms, by discarding or replacing the most complicated module in a model, both accuracy and inference speed in deep-learning-based object detection have been improved [44, 56, 61]. Re-designing fundamental components in the neural network architecture is another option for achieving light-weighting model.

GoogLeNet [44] presented an Inception block made up of 4 convolution paths in various configurations. Convolution with 1×1 kernel is extensively utilized in the Inception block to minimize the computational complexity. The network becomes more efficient by approximating the predicted ideal sparse structure using conveniently accessible dense construction pieces. In SqueezeNet [62], 1×1 convolutions are also utilized to replace 3×3 convolutions. It reduces the number of input channels to 3×3 convolutions and postpones the down-sampling operations in the network architecture. Finally, with the same detection accuracy, SqueezeNet is $50\times$ smaller than AlexNet [63] in size, resulting in higher detection speed.

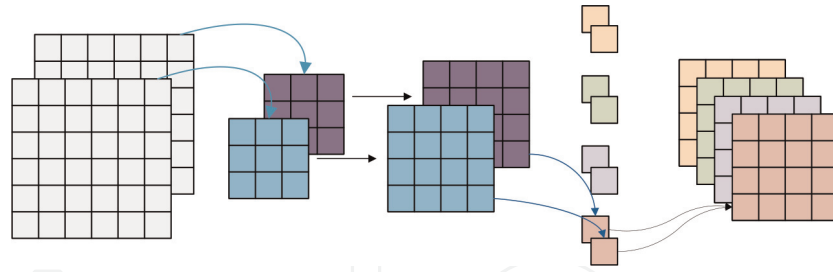


Figure 10.
Illustration of depth-wise separable convolution.

In contrast with conventional convolution, MobileNet [56] proposed depth-wise separable convolutions, which are a type of factorized convolution that factorizes a standard convolution into a depth-wise convolution and a 1×1 point-wise convolution, as shown in **Figure 10**, saving a significant amount of multi-adds operations and parameters while reducing accuracy by only 1%. ShuffleNet [64], on the other hand, makes use of two novel operations, point-wise group convolution and channel shuffle, to drastically reduce computational costs while preserving moderate detection accuracy. Xception [65], ResNeXt [17], and ChannelNet [66] are also wonderful works that adopt depth-wise separable convolution.

Depth-wise separable convolution, 1×1 convolution, and Max-pooling procedure are all employed extensively in the deep neural network presented in [61] to reduce computational complexity and model size. They also constructed an efficient receptive module inspired by Inception v3 architecture [67] to compensate for the inadequately retrieved features, as illustrated in **Figure 11**. Taking advantage of lightweight design, the proposed method outperforms or is comparable to state-of-the-art methods in terms of the *mAP* metric, and it significantly outperforms existing methods in terms of detection speed metrics, such as *GFLOPs*, processing time per image, and *FPS*. Experimental results demonstrate that the proposed algorithm can be executed on *RaspberryPi*, achieving real-time underwater object detection.

The underlying theory of lightweight network design is low-rank approximation. When information is encoded in data matrix X , a full-rank matrix \hat{X} , which is constructed by the linearly independent columns (or rows) of matrix X , can be obtained. It is quite conceivable (and rather frequent) for the rank of a matrix to be smaller than the total number of column vectors in the matrix. This means there are some redundant columns that can be generated by scaling and concatenating multiple

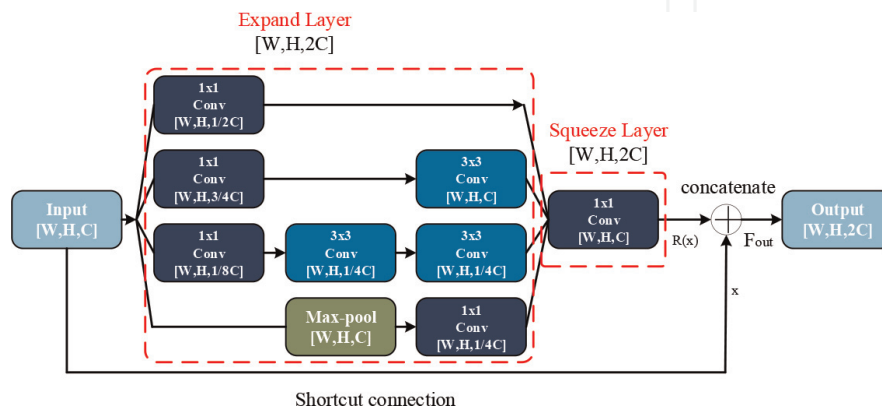


Figure 11.
Receptive module inspired by inception block (figure from [61]).

columns from the full-rank matrix \hat{X} . In other words, when a matrix contains redundant information, it can be represented by using fewer bits with little-to-no loss in information. Based on the theory of low-rank approximation, different effective and efficient techniques can be employed to design lightweight network architectures. Undoubtedly, it will play a pivotal role in realizing real-time detection of underwater objects.

4.2 Model compression

Another key method to achieve real-time detection is model compression [68], which aims to remove redundant parameters (or neurons) in the pre-trained models. Existing research has shown that deep networks exhibit parameter redundancy, which is useless for final prediction [69]. This serves as a theoretical foundation for compressing of deep learning models. Various compression methods have been proposed over the years, each of which has its pros and cons. Network pruning [70], knowledge distillation [71], and parameter quantization [72] are some of the most prominent strategies used to reduce network complexity.

Neural networks are typically over-parameterized, i.e., there are significant redundant parameters or neurons [73]. Based on this observation, we can reduce redundancy without compromising substantial performance degradation. In network pruning, the importance of neurons (or parameters) is first evaluated based on some metrics, such as the number of times it was not zero on a given dataset, the absolute values or the lifetime of the neurons, etc. Next, neurons that are of less importance will be removed.

With pruning, the model's performance is expected to drop. In general, performance degradation can be recovered by fine-tuning using the training dataset [74]. Network pruning can be applied at multiple granularities by different implementations, such as weight pruning, neuron pruning, kernel pruning, channel pruning, etc. By removing redundancy in the network, model complexity can be reduced, and generalization can be improved. Based on network pruning, even over 90% of the model size can be removed with little-to-no performance loss, and the computation speed of the model is improved significantly. In fact, network pruning has become a prerequisite for the deployment of deep learning on edge devices.

Knowledge distillation is another important technique for model compression. In general, training multiple distinct models on the same dataset and then averaging their predictions is a fairly easy technique to enhance the performance of almost any machine learning algorithm [75]. It is also widely believed that a large neural network usually outperforms a small one before over-fitting. Knowledge distillation compresses the knowledge in an ensemble (or a large model, known as a "Teacher Model") into a single small model (known as a "Student Model"), which is much easier to deploy on edge devices that are limited in computing resources [71]. It is achieved by minimizing the distance of predictive distribution between the Teacher Model and Student Model. The predictive distribution output by Teacher Model usually contains some implicit knowledge from the training dataset, which is helpful to guide model learning, easing out the optimization process. Through knowledge distillation, we can maintain superior performance of the larger model while reducing model size and consumption of computing resources.

Parameter quantization is concerned with re-organization of network parameters. The main objective of parameter quantization is to represent the neural network with

fewer bits [76]. For example, by compressing the 16-bit float parameters into 8-bit integers, one can halve the memory cost with little loss in performance [77]. However, the most commonly used quantization technique is parameter clustering [78, 79], where the parameters in a network are first clustered by clustering methods (e.g., k-means), and then every parameter is represented by the centroid of the corresponding cluster. Based on parameter clustering, the entire neural network can be represented by a cluster index table and the centroids. Each index is denoted by 2-bit unsigned integers. Hence, the deep learning model can be compressed significantly. In the extreme case, we can convert a network to a binary connect model, where all parameters are +1 or -1 [80]. Last but not least, some information encoding methods, such as Huffman encoding, that represents frequent clusters with fewer bits and rare clusters with more bits [81], can also be used as quantization techniques, since they are efficient encoding strategies.

In this section, we have reviewed two key techniques that help to reduce the model size but maintain moderate performance with only slight degradation. Through model reduction, the memory cost and computational complexity are reduced significantly, which makes real-time detection on resource-constrained devices more feasible. Indeed, lightweight network design and model compression are complementary and should be applied iteratively to obtain a more elegant model.

5. Conclusions

UMRs play a significant and pivotal role in ocean exploration in the era of Industry 4.0. Real-time object detection will equip UMRs with superior perception capabilities. In this chapter, we have identified real-time object detection as a key challenge of ocean exploration while using UMRs. Towards this end, crucial techniques pertaining to real-time detection of underwater objects have been critically reviewed and systematically analyzed based on the evolution in deep learning techniques. Three categories of detectors, namely two-stage detectors, one-stage detectors, and anchor-free detectors, have been reviewed and analyzed. Furthermore, futuristic trends of real-time detection, including lightweight network design and model compression, have been proposed and intensively discussed. It is hoped that readers will find this survey informative and useful in helping them to understand recent advancements in real-time detection of underwater objects, and will guide them in research in this exciting area, which will have a long-lasting impact to the mankind.

Acknowledgements

The authors would like to acknowledge the support of Fundamental Research Funds for the Central Universities under Grant 3132019344 and Leading Scholar Grant, Dalian Maritime University under Grant 00253007.

IntechOpen


IntechOpen

Author details

Meng Joo Er*, Jie Chen and Yani Zhang
Institute of Artificial Intelligence and Marine Robotics, College of Marine Electrical
Engineering, Dalian Maritime University, Dalian, China

*Address all correspondence to: mjer@dlnu.edu.cn

IntechOpen

© 2022 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Gordan M, Razak HA, Ismail Z, Ghaedi K. Recent developments in damage identification of structures using data mining. *Latin American Journal of Solids and Structures*. 2017;**14**: 2373-2401
- [2] Ghaedi K, Gordan M, Ismail Z, Hashim H, Talebkhah M. A literature review on the development of remote sensing in damage detection of civil structures. *Journal of Engineering Research and Reports*. 2021;**20**(10): 39-56
- [3] Gordan M, Sabbagh-Yazdi S-R, Ismail Z, Ghaedi K, Carroll P, McCrum D, et al. State-of-the-art review on advancements of data mining in structural health monitoring. *Measurement*. 2022;**193**:110939
- [4] Li J, Xu C, Jiang L, Xiao Y, Deng L, Han Z. Detection and analysis of behavior trajectory for sea cucumbers based on deep learning. *IEEE Access*. 2019;**8**:18832-18840
- [5] Liu C, Wang Z, Wang S, Tang T, Tao Y, Yang C, et al. A new dataset, poisson Gan and aquanet for underwater object grabbing. *IEEE Transactions on Circuits and Systems for Video Technology*. 2022;**32**:2831-2844
- [6] Song C, Cui W. Review of underwater ship hull cleaning technologies. *Journal of Marine Science and Application*. 2020;**19**(3): 415-429
- [7] Hoadley DS and Lucas NJ. *Artificial intelligence and national security*. Congressional Research Service. 2018
- [8] Gordan M, Chao OZ, Sabbagh-Yazdi S-R, Wee LK, Ghaedi K, Ismail Z. From cognitive bias toward advanced computational intelligence for smart infrastructure monitoring. *Frontiers in Psychology*. 2022;**13**:846610-846610
- [9] Talebkhah M, Sali A, Marjani M, Gordan M, Hashim SJ, Rokhani FZ. Iot and big data applications in smart cities: Recent advances, challenges, and critical issues. *IEEE Access*. 2021;**9**:55465-55484
- [10] Gordan M, Razak HA, Ismail Z, Ghaedi K, Tan ZX, Ghayeb HH. A hybrid ann-based imperial competitive algorithm methodology for structural damage identification of slab-on-girder bridge using data mining. *Applied Soft Computing*. 2020;**88**:106013
- [11] Gordan M, Ismail Z, Razak HA, Ghaedi K, Ibrahim Z, Tan ZX, et al. Data mining-based damage identification of a slab-on-girder bridge using inverse analysis. *Measurement*. 2020;**151**:107175
- [12] Gordan M, Ismail ZB, Razak HA, Ghaedi K, Ghayeb HH. Optimization-based evolutionary data mining techniques for structural health monitoring. *Journal of Civil Engineering and Construction*. 2020;**9**(1):14-23
- [13] Tan ZX, Thambiratnam DP, Chan TH, Gordan M, Abdul Razak H. Damage detection in steel-concrete composite bridge using vibration characteristics and artificial neural network. *Structure and Infrastructure Engineering*. 2020;**16**(9):1247-1261
- [14] Gordan M, Sabbagh-Yazdi S-R, Ghaedi K, Thambiratnam DP, Ismail Z. Introduction to monitoring of bridge infrastructure using soft computing techniques. In: *Applied Methods in Design and Construction of Bridges, Highways and Roads - Theory and Practice*. London, UK: IntechOpen; 2022 ch. 4

- [15] Prakash A, Behl A, Ohn-Bar E, Chitta K, Geiger A. Exploring data aggregation in policy learning for vision-based urban autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020. pp. 11763–11773. DOI: 10.1109/CVPR42600.2020.01178
- [16] Kendall A, Hawke J, Janz D, Mazur P, Reda D, Allen JM, et al. Learning to drive in a day. In: Proceedings of the International Conference on Robotics and Automation (ICRA). 2019. pp. 8248–8254. DOI: 10.1109/ICRA.2019.8793742
- [17] Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017. pp. 1492–1500. DOI: 10.48550/arXiv.1611.05431
- [18] Liu C, Li H, Wang S, Zhu M, Wang D, Fan X, et al. A dataset and benchmark of underwater object detection for robot picking. In: Proceedings of the IEEE International Conference on Multimedia & Expo Workshops. 2021. pp. 1–6. DOI: 10.1109/ICMEW53276.2021.9455997
- [19] Girshick R, Donahue J, Darrell T, Malik J. 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. DOI: 10.1109/CVPR.2014.81
- [20] Uijlings JR, Van De Sande KE, Gevers T, Smeulders AW. Selective search for object recognition. *International Journal of Computer Vision*. 2013;**104**(2):154-171
- [21] Jia D, Wei D, Richard S, Li JL, Kai L, and Li FF. Imagenet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2009. pp. 248–255. DOI: 10.1109/CVPR.2009.5206848
- [22] Girshick R. Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision. 2015. pp. 1440–1448. DOI: 10.1109/ICCV.2015.169
- [23] Karimanzira D, Renkewitz H, Shea D, Albiez J. Object detection in sonar images. *Electronics*. 2020;**9**(7):1180
- [24] Li X, Shang M, Qin H, Chen L. Fast accurate fish detection and recognition of underwater images with fast r-cnn. In: Proceedings of the IEEE Conference on OCEANS. 2015. pp. 1–5. DOI: 10.23919/OCEANS.2015.7404464
- [25] Ren S, He K, Girshick R, Sun J. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2016;**39**(6): 1137-1149
- [26] Li X, Shang M, Hao J, Yang Z. Accelerating fish detection and recognition by sharing cnns with objectness learning. In: Proceedings of the IEEE Conference on OCEANS. 2016. pp. 1–5. DOI: 10.1109/OCEANSAP.2016.7485476
- [27] Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. In: Proceedings of the European Conference on Computer Vision. 2014. pp. 818–833. DOI: 10.1007/978-3-319-10590-1_53
- [28] Shi P, Xu X, Ni J, Xin Y, Huang W, Han S. Underwater biological detection

algorithm based on improved faster-rcnn. *Water*. 2021;**13**(17):2420

[29] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016. pp. 770–778. DOI: 10.1109/CVPR.2016.90

[30] Tan M, Pang R, Le QV. Efficientdet: Scalable and efficient object detection. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020. pp. 10781–10790. DOI: 10.48550/arXiv.1911.09070

[31] Zhang YF, Ren W, Zhang Z, Jia Z, Wang L, Tan T. Focal and efficient iou loss for accurate bounding box regression. *Neurocomputing*. 2022;**506**: 146–157

[32] D. U. o. T. National Nature Science Foundation of China. China Underwater Robot Preessional Contest [Online]. Available: <http://www.urpc.org.cn/>

[33] He K, Gkioxari G, Dollár P, Girshick R. Mask r-cnn. In: *Proceedings of the IEEE international Conference on Computer Vision*. 2017. pp. 2961–2969. DOI: 10.1109/ICCV.2017.322

[34] Fan Z, Xia W, Liu X, Li H. Detection and segmentation of underwater objects from forward-looking sonar based on a modified mask rcnn. *Signal, Image and Video Processing*. 2021;**15**:1135-1143

[35] Lopez Marcano S, Jinks EL, Buelow CA, Brown CJ, Wang D, Kusy B, et al. Automatic detection of fish and tracking of movement for ecology. *Ecology and Evolution*. 2021;**11**(12): 8254-8263

[36] Zou Z, Shi Z, Guo Y, Ye J. Object detection in 20 years: A survey. *arXiv preprint arXiv:1905.05055*. 2019

[37] Yan D, Li G, Li X, Zhang H, Lei H, Lu K, et al. An improved faster r-cnn method to detect tailings ponds from high-resolution remote sensing images. *Remote Sensing*. 2021;**13**(11):2052

[38] Redmon J, Divvala S, Girshick R, Farhadi A. 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. DOI: 10.1109/CVPR.2016.91

[39] Redmon J, Farhadi A. Yolo9000: Better, faster, stronger. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017. pp. 7263–7271. DOI: 10.1109/CVPR.2017.690

[40] Redmon J, Farhadi A. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018

[41] Sung M, Kim J, Lee M, Kim B, Kim T, Kim J, et al. Realistic sonar image simulation using deep learning for underwater object detection. *International Journal of Control, Automation and Systems*. 2020;**18**(3): 523-534

[42] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde Farley D, Ozair S, et al. Generative adversarial networks. *Communications of the ACM*. 2020;**63**(11):139-144

[43] Ioffe S and Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the International Conference on Machine Learning*. 2015. pp. 448–456. DOI: 10.5555/3045118.3045167

[44] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In:

- Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015. pp. 1–9. DOI: 10.1109/CVPR.2015.7298594
- [45] Neves G, Ruiz M, Fontinele J, Oliveira L. Rotated object detection with forward-looking sonar in underwater applications. *Expert Systems with Applications*. 2020;**140**:112870
- [46] Yang H, Liu P, Hu Y, Fu J. Research on underwater object recognition based on yolov3. *Microsystem Technologies*. 2021;**27**(4):1837-1844
- [47] Haugaløkken BOA, Skaldebø MB, Schjølberg I. Monocular vision-based gripping of objects. *Robotics and Autonomous Systems*. 2020;**131**:103589
- [48] Bochkovskiy A, Wang C-Y, Liao H-YM. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv: 2004.10934. 2020
- [49] Wang CY, Liao HYM, Wu YH, Chen PY, Hsieh JW, Yeh IH. 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. DOI: 10.1109/CVPRW50498.2020.00203
- [50] Yao Z, Cao Y, Zheng S, Huang G, Lin S. Cross-iteration batch normalization. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021. pp. 12331–12340
- [51] Chen K, Chen Y, Zhou H, Mao X, Li Y, He Y, et al. Self-supervised adversarial training. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*. 2020. pp. 2218–2222. DOI: 10.1109/ICASSP40776.2020.9054475
- [52] Misra D. Mish: A self regularized non-monotonic neural activation function. arXiv preprint arXiv: 1908.08681. 2019;**4**:2
- [53] Ghiasi G, Lin TY, Le QV. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*. 2018;**31**
- [54] Zheng Z, Wang P, Liu W, Li J, Ye R, Ren D. Distance-iou loss: Faster and better learning for bounding box regression. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2020. pp. 12993–13000. DOI: 10.1609/aaai.v34i07.6999
- [55] Zhang M, Xu S, Song W, He Q, Wei Q. Lightweight underwater object detection based on yolo v4 and multi-scale attentional feature fusion. *Remote Sensing*. 2021;**13**(22):4706
- [56] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv: 1704.04861. 2017
- [57] Tian Z, Shen C, Chen H, He T. Fcos: Fully convolutional one-stage object detection. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019. pp. 9627–9636. DOI: 10.1109/ICCV.2019.00972
- [58] Hosang J, Benenson R, and Schiele B. Learning non-maximum suppression. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017. pp. 4507–4515. DOI: 10.1109/CVPR.2017.685
- [59] Zhou X, Wang D, Krähenbühl P. Objects as points. arXiv preprint arXiv: 1904.07850. 2019

- [60] Qin R, Zhao X, Zhu W, Yang Q, He B, Li G, et al. Multiple receptive field network (mrf-net) for autonomous underwater vehicle fishing net detection using forward-looking sonar images. *Sensors*. 2021;**21**(6):1933
- [61] Yeh CH, Lin CH, Kang LW, Huang CH, Lin MH, Chang CY, et al. Lightweight deep neural network for joint learning of underwater object detection and color conversion. *IEEE Transactions on Neural Networks and Learning Systems*. 2021:1-15. DOI: 10.1109/TNNLS.2021.3072414
- [62] Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and < 0.5 mb model size. arXiv preprint arXiv: 1602.07360. 2016
- [63] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*. 2012;**25**:1097-1105
- [64] Zhang X, Zhou X, Lin M, Sun J. 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. DOI: 10.1109/CVPR.2018.00716
- [65] Chollet F. Xception: Deep learning with depthwise separable convolutions. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017. pp. 1251–1258. DOI: 10.1109/CVPR.2017.195
- [66] Gao H, Wang Z, Cai L, Ji S. Channelnets: Compact and efficient convolutional neural networks via channel-wise convolutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2021;**43**(08): 2570-2581
- [67] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016. pp. 2818–2826. DOI: 10.1109/CVPR.2016.308
- [68] Cheng Y, Wang D, Zhou P, Zhang T. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. *IEEE Signal Processing Magazine*. 2018;**35**(1):126-136
- [69] Cheng Y, Yu FX, Feris RS, Kumar S, Choudhary A, Chang S. An exploration of parameter redundancy in deep networks with circulant projections. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2015. pp. 2857–2865. DOI: 10.1109/ICCV.2015.327
- [70] Liu Z, Sun M, Zhou T, Huang G, Darrell T. Rethinking the value of network pruning. arXiv preprint arXiv: 1810.05270. 2018
- [71] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv preprint arXiv: 1503.02531. 2015
- [72] Han S, Mao H, Dally WJ. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. arXiv preprint arXiv:1510.00149. 2015
- [73] LeCun Y, Denker J, Solla S. Optimal brain damage. *Advances in Neural Information Processing Systems*. 1989;**2**: 598-605
- [74] Liang T, Glossner J, Wang L, Shi S, Zhang X. Pruning and quantization for

deep neural network acceleration: A survey. *Neurocomputing*. 2021;**461**: 370-403

[75] Dietterich TG. Ensemble methods in machine learning. In: *Proceedings of the International Workshop on Multiple Classifier Systems*. 2000;**1578**:1-15

[76] Cheng Y, Wang D, Zhou P, Zhang T. A survey of model compression and acceleration for deep neural networks. arXiv preprint arXiv:1710.09282. 2017

[77] Vanhoucke V, Senior A, Mao MZ. Improving the speed of neural networks on cpus. In: *Proceedings of the Deep Learning and Unsupervised Feature Learning Workshop*. Granada Spain: NIPS; 2011

[78] Gong Y, Liu L, Yang M, Bourdev L. Compressing deep convolutional networks using vector quantization. arXiv preprint arXiv:1412.6115. 2014

[79] Wu J, Leng C, Wang Y, Hu Q, Cheng J. Quantized convolutional neural networks for mobile devices. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016. pp. 4820-4828. DOI: 10.48550/arXiv.1512.06473

[80] Courbariaux M, Bengio Y, David J-P. Binaryconnect: Training deep neural networks with binary weights during propagations. *Advances in Neural Information Processing Systems*. 2015; **28**:3123-3131

[81] Erdal E, Ergüzen A. An efficient encoding algorithm using local path on huffman encoding algorithm for compression. *Applied Sciences*. 2019; **9**(4):782