2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Single Shot Feature Aggregation Network for Underwater Object Detection

Lu Zhang[1,6], Xu Yang[1], Zhiyong Liu[1,4,6,*], Lu Qi[2], Hao Zhou[3] and Charles Chiu[5]

[1] State Key Laboratory of Management and Control for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2] The Chinese University of Hong Kong, Hong Kong, China
[3] Harbin Engineering University Harbin, Harbin, China
[4] Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Shanghai, China
[5] School for Higher and Professional Education, Chai Wan, Hong Kong, China
[6] University of Chinese Academy of Sciences, Beijing, China *zhiyong.liu@ia.ac.cn

***Abstract*—The rapidly developing ocean exploration and observation make the demand for underwater object detection become increasingly urgent. Recently, deep convolutional neural networks (CNN) have shown strong ability in feature representation and CNN-based detectors also achieve remarkable performance, but still facing the big challenge when detecting multi-scale objects in a complex underwater environment. To address this challenge, we propose a novel underwater object detector, introducing multi-scale features and complementary context information for better classification and location ability. In the auto-grabbing contest of 2017 Underwater Robot Picking Contest sponsored by National Natural Science Foundation of China (NSFC), we won the 1-st place by using proposed method for real coastal underwater object detection.**

I. Introduction

Although the ocean has extremely rich biological resources, it is not fully developed because human underwater work is dangerous and costly. Thus, people resort to automation devices like underwater robots, submarine, etc. Such devices need to analyze underwater images without human interference, which makes underwater object detection a fundamental yet huge demand.

For general object, many endeavors have been made to develop an efficient detector. In traditional fashion, underwater object detectors typically base on hand-craft features such as SIFT [1] and HOG [2], then follow a separate classifier like SVM. However, such methods suffer from weak representation ability and expensive multi-stage pipeline. Fortunately, in recent years, with the help of deep CNN [3], [4] and large computer vision dataset like ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [5]–[7], detection quality has achieved impressive success. Two-stage CNN-based methods (*e.g.*, Faster R-CNN [8]–[10], R-FCN [11]) firstly generate region proposals for potential objectness, then a classify network refines those proposals to get the final prediction. This kind of methods maintains state-of-the-art performance on PASCAL VOC [12] and MS COCO [13] but cannot meet real-time requirements. On the other hand, one-stage CNN-based methods like SSD [14] and YOLO [15] classify and localize objects simultaneously. Though slightly sacrificing some accuracy, one-stage methods make inference efficient and applicable to tasks which need to be real-time, like face detection [16].

However, so far, comparing with detecting objects on and over the ground in ImageNet [6] and PASCAL VOC [12], less progress was made in the unconstrained natural underwater scene. There are some tough challenges in underwater object detection, the challenges mainly come from three aspects:

1) Multi-scale and small objects. Since underwater images are usually collected by automatic devices, it is hard to constrain the distance between the imaging equipment and objects in a definite scope. Thus, scales of objects are divergent. Meanwhile, due to the huge exploration space in the ocean, scales tend to be small.

2) Unstable or lost features. Marine organisms live in complex and changeable underwater environment. With the water and its constituents, images may degrade due to blurring and scattering of light [17], resulting in unstable or lost features.

3) Efficiency requirements. When used in real-time applications where other parts of the system are computational as well, the detector should be efficient.

Considering above challenges and specific difficulties in underwater object detection tasks, we propose a robust and efficient single shot feature aggregation network which only contains a single fully convolutional network and can be trained end-to-end. Specifically, by introducing the multi-box feature pyramid module, we enhance multi-level features to handle various scales of objects, which is especially beneficial for detecting small objects. Moreover, to remedy unstable or lost features and build more robust feature representation, we propose the feature integration module which allows each level of feature pyramid to utilize useful information from both main and auxiliary features. Besides, we carefully design the anchor boxes and matching strategy to better suit the underwater detection task. Our method is evaluated on the challenging underwater object detection dataset[1] provided by NSFC, which

[1]Publicly available in http://www.cnurpc.org/index.html

978-1-5386-3788-3/18/$31.00 ©2018 IEEE

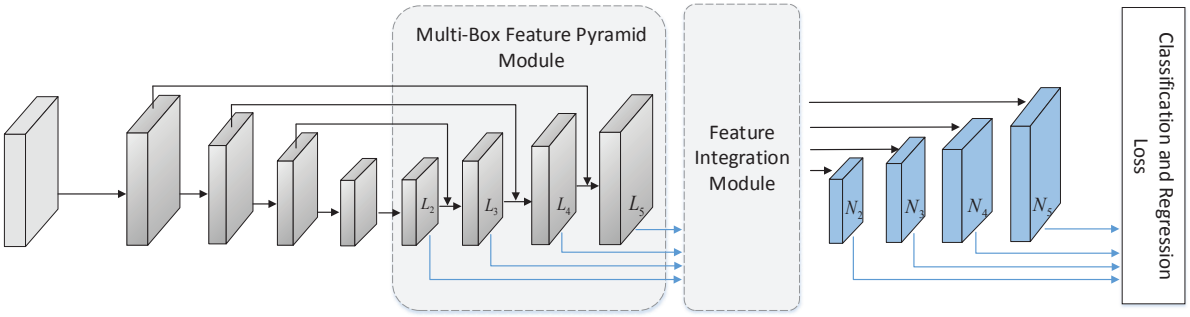

Fig. 1. The network structure of proposed approach. Given an input image, the feature is extracted using multi-box feature pyramid module and then go through a feature integration module to get the aggregated feature maps, then we make the classification and regression prediction to calculate the final loss respectively. Multi-box feature pyramid module is built on the VGG16 [18] backbone network with batch normalization [19], and details about the feature integration module will be discussed in section III.C.

has three typical marine organisms including sea cucumber, sea urchin, and scallop. Our approach beat both one-stage and two-stage state-of-the-art models on the public NSFC-dataset, and by using a single lightweight fully convolutional network, our model achieves 24 FPS on a portable device with NVIDIA GTX 1070.

II. Related Work

So far, limited pattern recognition methods are applied in underwater object detection tasks. In this section, we briefly review both underwater and general objection detection approaches.

Classic object detection and visual recognition methods are mainly based on the sliding-window paradigm with hand-craft features like SIFT [1], HOG [2], and Haar [20]. Besides, Felzenszwalb *et al.* proposed the deformable part model (DPM) [21], which prevailed on PASCAL VOC [12] for many years. For underwater object detection, brightness and color feature are used in [22], Mehdi *et al.* [23] adopt Haar and shape feature in automated fish detection. In [24], scale-invariant feature transform (SIFT) [1] and template of the objects of interest are used for discrimination and localization.

With the revival of convolutional neural networks, CNN-based detectors achieve significant improvement in detecting accuracy. Modern CNN-based object detection approaches can be roughly divided into two categories: two-stage method and one-stage method. We briefly review these two kinds of methods. **1) Two-stage method:** This kind of method consists of two stages, the first is proposal generation (*e.g.*, EdgeBoxes [25], DeepMask [26], [27], Selective Search [28], RPN [10]) and the second determines position and class label of objects using convolutional neural networks. The two-stage method achieves state-of-the-art performance, maintaining top results on PASCAL VOC and MS COCO. **2) One-stage method:** One-stage method unifies the proposal and prediction processes, making detector faster comparing to a two-stage one. Redmon *et al.* propose a real-time detector YOLO [15], using an end-to-end convolutional neural network to directly predict object's classes and locations, but there is still a large accuracy gap between YOLO and other two-stage methods. After that, SSD [14] adopts the concept of anchor boxes in [10] and tiles anchor boxes of different scales respectively on a certain layer to improve detection performance. Later, DSSD [29] is proposed to introduce context information for better feature representation. Inspired by above CNN-based general object detection approaches, Li *et al.* [30] adopt Fast R-CNN [9] framework for underwater object detection. In [31], the backbone network is modified to achieve efficient fish detection.

Meanwhile, some underwater object detection methods are implemented by the help of specific sensors like underwater ultrasonic signal [32], high-resolution sonar [33], to name a few, which is out of our scope.

III. Our Approach

In this section, we first present the architecture of the proposed model. Then we introduce the multi-box feature pyramid module. After that, we discuss different strategies of aggregating contextual information in different levels of features to obtain robust underwater object detector. Finally, we introduce our training methodology. The details are described as follows.

A. Network Structure

Our framework is illustrated in Figure 1. We use VGG16 [18] as the backbone network and initialize it by the ImageNet pre-trained weights. For our underwater detection task, we convert $fc6$ and $fc7$ in VGG16 [18] to convolutional layers and subsample their parameters as in SSD [14], but keep $pool_5$ unchanged. Next, the multi-box feature pyramid module is performed to introduce extra connections for propagating high-level features to augment the semantic information of lower layers in a high-to-low fashion. Then a complementary feature integration module is conducted to incorporate different levels of features for robust representations. Finally, we perform multi-class classification and class-agnostic bounding box regression to get the final prediction of the object.

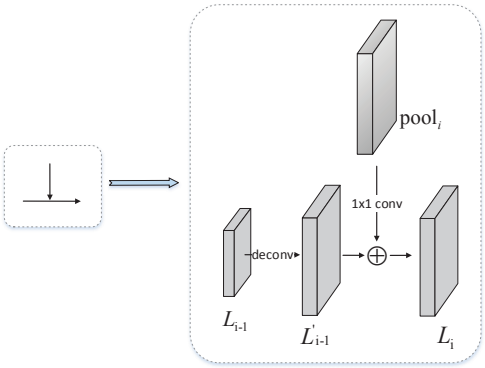

Fig. 2. Connection scheme of the multi-box feature pyramid module. Convolution and deconvolution layers are used to fix the channels of feature map to 256 for consistency before fusion.

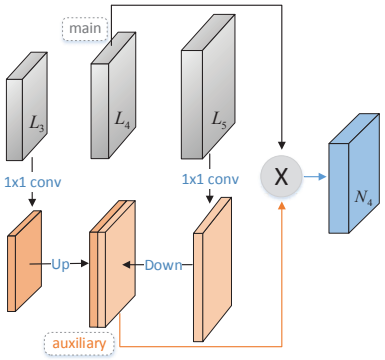

Fig. 3. Feature integration module, "**X**" indicates the normalization and integration process. We use 3×3 convolution and deconvolution with stride 2 for downsampling and upsampling respectively.

B. Multi-Box Feature Pyramid

In the unconstrained underwater environment, scales of objects are various and tend to be relatively small. Unlike using a single level of feature map to extract information as in Faster R-CNN [10] and R-FCN [11], we choose to tile multi-scale anchor boxes on multiple layers and utilize fine-grained information to improve performance.

Since we use straight feedforward network (e.g., VGG16 [18]) as backbone, we argue that higher layers have stronger high-level representation and lower layers remain more detailed features as in [34]. Thus, simply tiling anchor boxes on fine-grained layers to detect small objects is an intuitive but not suitable choice because such layers lack of semantic information for prediction. In SSD [14], extra layers are added following backbone network to capture high-level information, but with the network become deeper and thinner, the receptive field of per pixel on feature maps increases, small objects may not even have information in these layers. To take advantage of fine-grained information and alleviate the harmfulness of the lack of semantic information, we forgo extra layers in SSD [14] and introduce the multi-box feature pyramid module inspired by [35] to enhance low-level feature maps with semantic information and reasonable classification and localization abilities. Specifically, We adopt $pool2$, $pool_3$, $pool_4$, $pool_5$ to generate our pyramid in an hourglass manner with shortcut connection, which are more fine-grained than features used by the existing method. A concrete connection scheme is illustrated in Figure 2.

C. Feature Integration Module

With above feature pyramid fashion, scales of anchor boxes are *discretized* for different layers, hence anchor boxes in scales like 0.39 and 0.41 could be distributed to two different feature maps, but these two anchor boxes are fairly similar. Moreover, in a specific underwater scene, image degradation can destroy some levels of features. For instance, according to organisms like sea urchin and scallop, texture information is easy to be damaged and color information sometimes can be more stable, while for sea cucumber, texture information would be more discriminative. In experiments which use a single level of feature map, we find that using conv5 will improve the accuracy of sea cucumber but decrease which of sea urchin and scallop comparing with conv4. Therefore, simply extracting features from one single level of feature map may suffer from nonoptimal anchor assignment and missing important features, then lead to inaccurate classification and localization.

Further, context is a very important factor for visual recognition tasks and details are helpful to discriminate object of a small scale. Thus, in order to utilize both kinds of information and alleviate above problems, we introduce **contiguous feature** as the "auxiliary" feature for further feature aggregation. Contiguous feature has several advantages: (a) Multi-level semantic features. Deep, medial and shallow features are complementary as shown in many computer vision task, integration of them is beneficial for underwater object detection task. (b) Feature consistency. Too deep or shallow levels of features have more resolution differences and need large-scale upsampling or downsampling, which can introduce noise and errors. Thus it is reasonable to fuse features which have similar scale and spatial distribution in a contiguous way. (c) Computation efficiency. Making use of all other levels of features is also a choice but demands heavy computation, yet the average computation of contiguous feature doesn't increase as we use more levels of features.

As shown in Figure 3, we take the generation process of N_4 for an example. For L_4, we utilize its neighboring feature maps L_3 and L_5 to get the auxiliary feature map. First, L_3 and L_5 are reduced to half of its channels by a 1×1 convolution layer, and we upsample and downsample respectively to generate feature maps of the same width and height with L_4. Then, we concatenate the two half-channels feature maps to get the final *auxiliary* feature for the *main* L_4. Considering L_2 and L_5 only have a single neighbor, for N_2 and N_5, we skip the channel reduction and concatenation step, directly implement the downsampling or upsampling process.

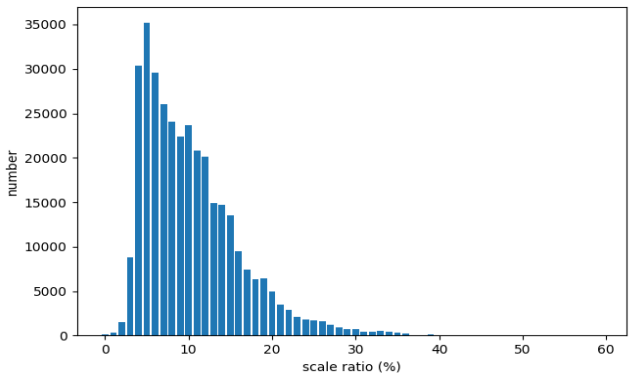

Fig. 4. Statistics of ground truth boxes of training set in the NSFC-dataset. Scale ratio (%) is object's ratio with respect to the original image.

Next, to match the order of magnitude of the main and auxiliary feature maps before integration, we first apply a normalization operation. We explore two different methods: L2 Normalization and 1×1 convolutional layer to model normalization. Then we study how to integrate main and auxiliary features. We perform four integration strategies: element-wise sum, element-wise max, element-wise product and channel-level concatenation. Experiments show that 1×1 convolutional normalization and element-wise sum achieves the best performance. Specific performance of each normalization and integration strategies are discussed in section IV.

D. Anchor Box Design and Matching Strategy

During training, anchor boxes need to be matched with ground truth boxes for subsequent loss calculation and back-propagation process. Thus a proper anchor box design and matching strategy could speed up training process and is crucial for model's performance.

There are two prior conditions: 1) as shown in Figure 4, underwater object's scale is various and relatively small; 2) we tile anchor boxes on semantically enhanced high-resolution feature maps. To better match ground truth bounding box and utilize the discrimination of high-resolution feature maps, we use eight scales on four pyramid layers evenly distributed from 0.02 to 0.6 with aspect ratios 1:1, 1:2 and 2:1.

For anchor boxes matching, we follow [14] to match each ground truth box with anchor boxes with best Jaccard overlap firstly. Then we decrease the threshold from 0.5 to 0.4 for matching ground truth boxes and anchor boxes with a Jaccard overlap higher than this threshold to improve the recall rate for small objects.

E. Training

Our model is trained on a challenging dataset provided by NSFC including nearly 20,000 underwater 720×405 images of three typical creatures: sea cucumber, sea urchin, and scallop. We use VGG16 [18] as our backbone network, which is pretrained on ImageNet [7] dataset, then we fine-tune the model on NSFC-dataset. This subsection introduces our training strategies and other implementation details.

1) Data Augmentation: We use a few of data augmentation strategies to enhance data diversity and generalization of our model. Each training image is processed by random crop, random padding, color jittering and horizontal flipping in sequence.

2) Loss Function: Our loss function is similar to which is defined in [14]. Specifically, we adopt softmax loss for classification and smooth L1 loss for bounding box regression:

$$L(\{p_i\},\{c_i\},\{l_i\},\{g_i\}) = \frac{1}{N}\Big(\sum_i L_{cls}(p_i, c_i^*) + \alpha \sum_i [c_i^* \geq 1] L_{loc}(l_i, g_i^*)\Big) \tag{1}$$

In Equ.(2), p_i indicates the predicted confidence and c_i^* is the ground truth label of i-th anchor boxes. For our underwater object detection task, we set the weight term $\alpha = 4$ by cross-validation.

3) Hard Negative Mining: After matching, most anchor boxes are negative. This introduces the imbalance between negatives and positives, which is very harmful to the training process. In order to overcome this problem, we sort anchor boxes by loss values and resample from top ones to keep the negative and positive anchor boxes up to 3:1.

4) Other implementation details: Our experiments are all based on VGG16 [18] with a minor change, please refer to [14] for more details. We use batch size 8 for training. The maximum number of training epoch is 40. Meanwhile, we set the initial learning rate to 10^{-3} for the first 20 epochs, and 10^{-4} for the last 20 epochs. All new layers are randomly initialized with the "Xavier" [36] method.

IV. Experiment

We train the proposed model with 320×320 input size for fast training and inference on the NSFC-dataset. We implement our approach in MXNet [37].

A. Evaluation on benchmark

Since the detailed frameworks or code of the four typical state-of-the-art methods [10], [11], [14], [38] are publicly available, we can adapt them to our task for comparisons. All models are similarly initialized with pretrained model and fine-tune for the NSFC-dataset with same hyperparameters. We apply the same anchor assignment strategy for those handcrafted anchor-based methods [10], [11], [14]. The detection accuracy is measured by mean Average Precision (mAP). Results are shown in TABLE I.

From TABLE I, we can see a counterintuitive phenomenon: two-stage methods are not performing better than one-stage SSD. We argue that this phenomenon results from the ROI pooling operation in [10], [11]. As is noted in [39], pooling operation could merge nearby response, lead to crowding and decrease recognition accuracy in a messy setting. Though modern detector's performance is generally verified on datasets like PASCAL VOC and MS COCO, there is still a considerable domain gap between those datasets and underwater

TABLE I
MEAN AVERAGE PRECISION ON BENCHMARK

Method	seacucumber	seaurchin	scallop	mAP
F-RCNN [10]	47.0	56.8	66.4	56.7
R-FCN [11]	46.9	56.1	61.0	54.7
SSD [14]	51.8	62.7	69.1	61.2
YOLOv2 [38]	43.2	46.6	56.2	48.7
Ours	**55.4**	**65.6**	**70.8**	**63.9**

image datasets, which incorporate more clutters and flankers in similar appearance with target objects. Since YOLOv2's performance still trails two-stage methods, we argue that the multi-layer features for multi-scale anchor tiling of SSD contribute a lot as well.

Based on one-stage paradigm, our model improves the performance by promoting both mAP and average precision (AP) for each individual category, which shows the effectiveness of our approach.

B. Ablation Study

For clarity and consistency, all the ablation experiments use same hyperparameters and single-model training and testing.

1) Normalization and Integration Strategy: In this part, we explore different strategies for normalizing and integrating main and auxiliary information.

TABLE II
PERFORMANCE OF DIFFERENT NORMALIZATION AND INTEGRATION STRATEGIES.

Strategy	SUM	MAX	PROD	CONCAT
None	63.0	63.1	63.3	62.6
L2 Norm	62.5	61.3	-	61.6
1x1 conv	**63.9**	63.1	62.8	62.2

Metric: mAP(%) on PASCAL VOC.

Normalization. Since features from different layers show different scales of activations, it is crucial to normalize the feature properly before integration. We investigated two different normalization strategies: L2 Normalization and 1×1 convolution. Experiments show that 1×1 conv achieves best results.

Integration strategy. From TABLE II we can see that element-wise sum generally performs best. This operation is effective and widely used in computer vision tasks [40], [41]. And we argue that element-wise max can be seen as an ensemble process and the element-wise product could weaken activations at the early stage of training. For concatenation, training could become harder because the network needs to learn how to effectively integrate the concatenated features without prior relation information of channels of the features. Moreover, other integration strategies are worth to be researched like learnable integration or other non-linearity structure.

2) Module Ablation: To better understand our approach , we ablate each proposed module to examine how they contribute to final performance. TABLE III shows our results.

TABLE III
ABLATION STUDY OF VARIOUS DESINGS

Module			
Multi-Box Feature Pyramid			×
Feature Integration		×	×
mAP	**63.9**	62.6	60.8

Multi-Box Feature Pyramid. We construct the network on VGG16 without any extra connection in multi-box feature pyramid module to demonstrate the effect of it. By comparing the second and third columns in TABLE III (62.6% vs 60.8%), we find that multi-box feature pyramid improves mAP by 1.8%. The main reason is that semantic information is enhanced on fine-grained feature maps by this module, which can significantly help to promote the accuracy.

Feature Integration. To validate the effectiveness of feature integration module, we remove the feature normalization and integration process of our model. We find that the mAP drops by 1.3% (from 63.9% to 62.6%), which indicates that the feature integration module can yield a considerable improvement of the performance.

C. Runtime Performance

Our model falls into the efficient one-stage category, it is a single shot network to predict classification and localization results simultaneously. We forgo extra layers in [14] and introduce multi-box feature pyramid and feature aggregation modules with few additional computation overhead. The speed is evaluated with batch size 1 on a portable machine with NVIDIA GTX 1070 and CUDA 8.0. Our model achieves 24 FPS in inference, which is considerable in the real-time application.

V. CONCLUSION

In this paper, we propose a novel framework for underwater object detection which consists of two interconnected modules. Firstly, the multi-box feature pyramid module introduces semantic cues for accurate underwater object detection. Then we propose feature integration strategies for further feature aggregation to enhance the feature robustness and complementarity. With the combination of these two modules and appropriate anchor design and matching, we beat four typical state-of-the-art detectors on the challenging NFSC-dataset, which validates the effectiveness of our approach. The proposed detector is fast, achieving 24 FPS on a portable device.

ACKNOWLEDGMENT

This work was supported in part by the National Key Research and Development Plan of China under Grants 2017YFB1300202 and 2016YFC0300801, in part by the NSFC under Grants U1613213, 61627808, 61503383,

61210009, 91648205, 61702516, and 61473236, in part by the MOST under Grant 2015BAK35B00 and Grant 2015BAK35B01, in part by the Guangdong Science and Technology Department under Grant 2016B090910001.

REFERENCES

[1] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 1. IEEE, 2005, pp. 886–893.

[3] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[6] J. Deng, A. Berg, S. Satheesh, H. Su, A. Khosla, and L. Fei-Fei, "Imagenet large scale visual recognition competition 2012 (ilsvrc2012)," 2012.

[7] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

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

[9] R. Girshick, "Fast R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[10] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[11] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387.

[12] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD:single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.

[16] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "S3FD: Single shot scale-invariant face detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 192–201.

[17] R. Schettini and S. Corchs, "Underwater image processing: state of the art of restoration and image enhancement methods," *EURASIP Journal on Advances in Signal Processing*, vol. 2010, no. 1, p. 746052, 2010.

[18] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.

[20] P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[21] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.

[22] C. Spampinato, Y. H. Chen Burger, G. Nadarajan, and R. B. Fisher, "Detecting, tracking and counting fish in low quality unconstrained underwater videos." *VISAPP (2)*, vol. 2008, pp. 514–519, 2008.

[23] M. Ravanbakhsh, M. R. Shortis, F. Shafait, A. Mian, E. S. Harvey, and J. W. Seager, "Automated fish detection in underwater images using shape-based level sets," *The Photogrammetric Record*, vol. 30, no. 149, pp. 46–62, 2015.

[24] D. Lee, G. Kim, D. Kim, H. Myung, and H.-T. Choi, "Vision-based object detection and tracking for autonomous navigation of underwater robots," *Ocean Engineering*, vol. 48, pp. 59–68, 2012.

[25] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *European Conference on Computer Vision*. Springer, 2014, pp. 391–405.

[26] P. O. Pinheiro, R. Collobert, and P. Dollár, "Learning to segment object candidates," in *Advances in Neural Information Processing Systems*, 2015, pp. 1990–1998.

[27] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár, "Learning to refine object segments," in *European Conference on Computer Vision*. Springer, 2016, pp. 75–91.

[28] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[29] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017.

[30] X. Li, M. Shang, H. Qin, and L. Chen, "Fast accurate fish detection and recognition of underwater images with fast r-cnn," in *OCEANS'15 MTS/IEEE Washington*. IEEE, 2015, pp. 1–5.

[31] X. Li, Y. Tang, and T. Gao, "Deep but lightweight neural networks for fish detection," in *OCEANS 2017-Aberdeen*. IEEE, 2017, pp. 1–5.

[32] H. Cho, J. Gu, H. Joe, A. Asada, and S.-C. Yu, "Acoustic beam profile-based rapid underwater object detection for an imaging sonar," *Journal of Marine Science and Technology*, vol. 20, no. 1, pp. 180–197, 2015.

[33] L. Henriksen, "Real-time underwater object detection based on an electrically scanned high-resolution sonar," in *Autonomous Underwater Vehicle Technology, 1994. AUV'94., Proceedings of the 1994 Symposium on*. IEEE, 1994, pp. 99–104.

[34] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[35] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *arXiv preprint arXiv:1612.03144*, 2016.

[36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[37] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *arXiv preprint arXiv:1512.01274*, 2015.

[38] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *arXiv preprint arXiv:1612.08242*, 2016.

[39] A. Volokitin, G. Roig, and T. A. Poggio, "Do deep neural networks suffer from crowding?" in *Advances in Neural Information Processing Systems*, 2017, pp. 5631–5641.

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

[41] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, and H. Lu, "CoupleNet: Coupling global structure with local parts for object detection," in *Proc. of Intl Conf. on Computer Vision (ICCV)*, 2017.