IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 1, 2020, accepted September 9, 2020, date of publication September 21, 2020,
date of current version October 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3025617

Integrate MSRCR and Mask R-CNN to Recognize Underwater Creatures on Small Sample Datasets

SHAOJIAN SONG, (Member, IEEE), JINGXU ZHU, XIUHUA LI, AND QINGBAO HUANG
School of Electrical Engineering, Guangxi University, Nanning 530004, China

Corresponding author: Shaojian Song (sjsong03@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61863003, and in part by the Natural Science Foundation of Guangxi Province under Grant 2016GXNSF AA380327.

ABSTRACT The poor quality of optical imaging caused by the complex and varying underwater environment is a significant challenge to underwater target recognition. Moreover, the insufficiency of relevant datasets may lead to the overfitting problem in target recognition models based on deep learning. Taking the instance segmentation of three underwater creatures (echinus, holothurian, and starfish) as an example, we propose a new method for recognition of underwater creatures. It combines the MSRCR (multi-scale Retinex with color restoration) image enhancement algorithm and the Mask R-CNN (region-based convolutional neural work) framework, and achieves a mAP (mean average accuracy) value higher than 90% on a small sample dataset. This method consists of three major steps. First, the dataset with 84 images is augmented (flip, adding noise, and GAN (generative adversarial networks)) to 430 images, and all images are enhanced with MSRCR to improve their qualities; Second, the model is pre-trained on the COCO (Microsoft common objects in context) dataset to shorten the training time and overcome overfitting; Finally, the pre-trained model is transferred to the underwater dataset, and the whole training process is completed. We achieve 97.46% precision and 94.52% recall, and the mAP (intersection over union (IOU) = 50) is 94.84%. The effectiveness of the proposed method is verified by comparing it with several popular target recognition models, including SSD (Single Shot Detector), YOLOv3 (You only look once), original Mask R-CNN, and a SIFT-based (Scale-invariant feature transform) model.

INDEX TERMS Object recognition, mask R-CNN, image enhancement, underwater creature.

I. INTRODUCTION

Seventy-one percent of the Earth's surface is occupied by oceans, which contain rich resources [1]. Due to human physiological limits, people usually need the assistance of underwater vehicles with different functions to complete long-term underwater works [2], [3]. During the working process of underwater vehicles, correct detection and identification of underwater targets are essential for its safety and efficiency. Hence, an underwater vehicle is usually equipped with an optical vision system or imaging sonar system to capture underwater environmental information. The optical vision system can acquire more interpretable information compared with the imaging sonar system, and it is more conducive to enhancing the recognition capability and automation of underwater vehicles [4], [5]. Underwater optical imaging is more demanding on photography equipment than conventional optical imaging, requiring dedicated lens, flash, image sensors, and so on. Common underwater photography equipment ranges from the amateur-grade devices such as GoPro to professional-grade devices such as HY-CR109. However, even with dedicated equipment, the quality of underwater imaging is still inferior to conventional imaging.

The poor quality of underwater imaging results from the selective absorption of light (water hindering the propagation of red and yellow light the most, and the propagation of blue and green light the least) and the scattering of light (resulting from the impurities and the flow of water). A similar issue is imaging on foggy days [6]. Actually, underwater imaging, together with extreme weather imaging, including rain, fog, and snow, can be classified into non-uniform media imaging. Besides, infrared imaging [7] is similar to underwater imaging in the sense of selective absorption of light (selecting the infrared light artificially).

The associate editor coordinating the review of this manuscript and approving it for publication was Seifedine Kadry.

Compared with the other non-uniform media imaging, underwater imaging tends to be persistent and typical. With the aforementioned factors resulting in color attenuation, blue or green color tone, noise, and bright spots in underwater images, great challenges present for underwater target detection and instance segmentation.

To detect moving underwater targets, Jie *et al.* [8] referred to frog eyes and proposed a hierarchical background model from the perspective of bionics. Considering limited underwater computing resources, Yiru *et al.* [9] proposed a fast method to segment underwater images using the improved Markov random field model combined with the hard clustering means. To further solve the problem of low-visibility conditions underwater, Dark Channel Prior, wavelet transform kernel, and hierarchical multi-scale decomposition algorithms were integrated to segment images in [10]. Srividhya [11] initialized the number of clusters of a Gaussian mixture model to recognize fish, and used inner distance shape matching to improve recognition accuracy.

Although the above methods without involving deep learning have made some progress in certain specific situations, but due to the series of problems underwater images tend to have compared with general optical images, it is still difficult to achieve satisfactory recognition rate for underwater creatures. With the rise of deep machine learning, Alex *et al.* [12] proposed a remarkable deep CNN model whose accuracy took the first place in ILSVRC2012. Since then, deep learning has become a new methodology for underwater biological target recognition, especially multi-target multi-class underwater target detection. In [13] and [14], the multi-domain collection of datasets was applied to train deep learning models for detecting fish. This method can expand the dataset, but can easily cause the problem of data imbalance. Hongwei *et al.* [15] proposed a deep architecture to recognize the live fish in the water by combing CNNs, principal component analysis, block-wise histograms, spatial pyramid pooling, and linear SVM (support vector machine) together. In their method, masks of fish instead of the original images were fed to the architecture. In [16], the outputs of the Gaussian mixture model and optical flow algorithm, together with greyscale fish image, were fed to CNNs and RPN (region proposal networks) instead of RGB images. The model worked well on multi-target detection, but only for binary classification. Wenwei and Shari [17] proposed a deep learning architecture, and YOLO was applied for training to recognize the fish in underwater videos using three very different datasets, which were recorded on real-world water power sites. Nevertheless, they only achieved a mAP of 0.5392. Hai *et al.* [18] applied the Faster R-CNN [19] to autonomous underwater vehicle to detect marine fishes. Its adaptability to the changes of marine environment was significant, but the good results were achieved with fixed point observation and relatively good water quality. Tayyab *et al.* [20] used a 32-layer CNN to classify the fish. Their method was effective, but it was only used for high-quality fish image classification, and could not detect fish in images.

To improve the recognition accuracy of underwater targets, some special problems of the underwater images, such as poor image purity, loss of detail, and blue or green color tone need to be dealt with. Several image enhancement techniques, including Dark Channel Prior, wavelet transform kernel algorithms were integrated to segment underwater images on low-visibility conditions in [10]. And in [21], the MSR (multi-scale Retinex) was adopted to enhance underwater images for improving detection, and it is the predecessor of MSRCR [22].

It is well known that models based on deep learning are prone to small sample overfitting problems. Different from the conventional images acquisition on land, the acquisition of underwater images requires professional equipment and personnel, including underwater photographers and lifeguards. Furthermore, when faced with an underwater scene, the complex and varying environments often lead to unsatisfactory imaging. Hence, available datasets for target recognition of underwater creatures are rather rare which results in scarce underwater deep learning methods [15]. Additionally, even though a few datasets are available, most of them are only used for fish target recognition [13]–[20]. Only a few works of literature are focusing on other underwater objects except fish. In [21], holothurians have been detected with the pruned SSD algorithm [23]. Mahmood *et al.* [24] combined hand-crafted features with VGG (visual geometry group network) [25] representations to classify coral reefs, and achieved a state-of-art classification accuracy on the MLC (Moorea Labelled Coral) dataset. Shuo *et al.* [26] used MobileNetV2 [27] as the backbone of SSD to detect crabs fast, and they also replaced the standard convolution with depthwise separable convolution. The speed of their method reached over 70 frames per second. Vitjan *et al.* [28] applied a deep encoder-decoder network to detect jellyfish polyp on a small sample dataset, but their images are clear, high resolution with 4288×2844 pixels. The methods in [26] and [28] are both only for binary classification.

To address the problem of insufficient sample, artificial images were applied to expand datasets in [29], and a deep model based on SegNet [30] was trained with the annotated artificial images. Hubert and Ganesh [31] proved that using GAN to generate images for training can improve the robustness of deep models. Benjamin *et al.* [32] demonstrated that using GAN to enlarge dataset can improve the recognition of handwritten digits.

In 2019, Jian *et al.* introduced a dataset called Marine Underwater Environment Database [33], and this dataset contains hundreds of object categories, benefitting the development of underwater vision technology. But, it is a pity that this dataset is in particular for saliency detection [34], which only pays attention to salient objects but not all objects and it does not concern the classification of objects.

In conclusion, insufficient datasets, together with poor quality of images, result in difficulties in underwater target recognition based on deep learning. Existing studies rarely

achieve high recognition accuracy, especially for multi-target multi-class recognition. Therefore, underwater images must be enhanced, reconstructed, and augmented [35], so as to narrow the gap with conventional images.

To consider the aforementioned factors and realize multi-target multi-class recognition based on a small sample underwater dataset, image augmentation and enhancement and deep learning framework are integrated to develop a method with Mask R-CNN [36] as the main body in this article. The overall structure is shown in Fig. 1.

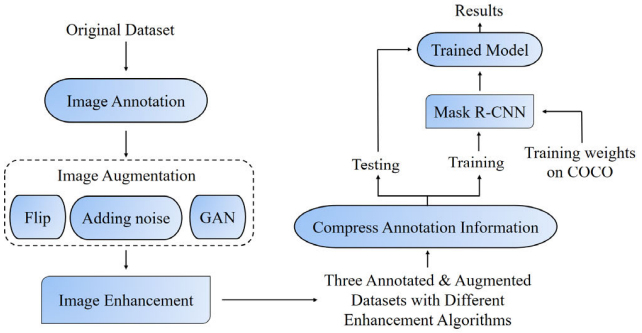


FIGURE 1. The structure of the proposed method.

The main contributions of this article are as follows:

1) MSRCR is integrated with the Mask R-CNN framework to enhance images before training so as to improve recognition accuracy. It is demonstrated that an appropriate enhancement algorithm can benefit the recognition accuracy in underwater scene.

2) The problem of overfitting caused by a small sample dataset is addressed by data augmentation and transfer training. We also apply GAN to generate images for augmentation.

3) The mask branch is used to determine the attribution of each pixel, and instance segmentation for an underwater environment is accomplished.

The rest of this article is organized as follows. Section 2 briefly illustrates the principles of the algorithms used in this article, including MSRCR and Mask R-CNN; Section 3 presents experimental details of the proposed method; Section 4 shows the comparative experimental results; and Section 5 draws the conclusions.

II. ALGORITHM PRINCILPE OF MSRCR AND MASK R-CNN

Underwater images usually have a variety of defects, such as uneven illumination, low contrast, poor purity, loss of detail, and blue (green) color tone. Through enhancement processing, the perception gap between underwater images and conventional images can be reduced, thus it can improve the accuracy of Mask R-CNN. The principles of MSRCR and Mask R-CNN are described as follows.

A. MSRCR ALGORITHM

MSRCR was developed from Retinex by Land [37], and we replace the color restoration function of MSRCR for better performance. The essence of Retinex is that the image is represented by the product of the illuminating component and the reflected component, as shown in (1):

$$I(x, y) = R(x, y) \cdot L(x, y) \tag{1}$$

where $I(x, y)$ represents a pixel value of the image acquired by cameras (reflection component), and (x, y) are the coordinates of a pixel; $R(x, y)$ corresponds to the high-frequency component of the pixel, which is independent of illumination, indicating the original appearance of an object; $L(x, y)$ corresponds to the low-frequency component of the pixel, indicating the illumination component.

MSRCR extends MSR by adding color restoration which is crucial for dealing with blue (green) color tone. The image obtained by MSR processing is shown in (2):

$$R_{msr}(x, y) = \sum_{n=1}^{N} W_n \{lgI(x, y) - lg[F(x, y) * I(x, y)]\} \tag{2}$$

where the difference between the logarithms of the two sides of (1) is employed. W_n is the weight of the n-th scale, N is the number of scales [22], $R_{msr}(x, y)$ is a pixel value of the image after multi-scale processing, and $F(x, y)$ is a Gaussian function. The illumination component is obtained by convolving the Gaussian function with the input image.

After enhancement, a color image tends to be color distorted. We can form an improved algorithm, MSRCR, by adding a color restoration processing, as shown in (3):

$$R_{msrcr}(x, y) = C(x, y) \cdot R_{msr}(x, y) \tag{3}$$

where $C(x, y)$ is the color restoration function. For faster speed and better color recovery, the restoration function used in this article is redefined, as shown in (4):

$$C(x, y) = g \cdot \left[lg(\alpha \cdot I(x, y) + 1) - I'(x, y)\right] \tag{4}$$

where the gain constant g and controlled nonlinear α are hyperparameters to be determined empirically. Through experimentation, their values are set to 1 and 128 respectively to cope with an underwater scene. I' adds two channels to I'', and the values of newly added channels are all zero. The expression of I'' is shown in (5).

$$I''(x, y) = lg\left[\sum_{i=1}^{3} I_i(x, y)\right] + 3 \tag{5}$$

The color-restored image also needs to be quantized to the interval [0, 255]. This article uses the linear quantization method, as shown in (6).

$$image = Clip\left(\frac{R_{msrcr} - min}{max - min} \times 255\right) \tag{6}$$

In (6), *Clip* represents a shear function that clips values outside [0, 255] to the boundary of the range. The values of *min* and *max* are calculated as (7):

$$\begin{cases} min = mean - dynamic \cdot std \\ max = mean + dynamic \cdot std \end{cases} \tag{7}$$

where *mean* is the mean of all pixels in R_{msrcr}, *std* is the standard deviation of pixels, and *dynamic* is a hyperparameter. A smaller *dynamic* value can produce higher contrast. Normally, its value is 3.0, and it was set to 2.5 for better visual perception in this article.

B. MASK R-CNN FRAMEWORK

Mask R-CNN is a highly versatile instance segmentation framework derived from Faster R-CNN [19]. The training process of Mask R-CNN is briefly depicted in Algorithm 1, and the more detailed descriptions are in the following.

Algorithm 1 Training Process of Mask R-CNN

For each image in training set:

1: Send a new image and use CNNs to extract different scales of feature maps;

2: Combine the extracted feature maps to form a feature pyramid;

3: Traverse the feature pyramid and propose regions of interest;

4: Perform bounding regression and foreground & background classification on the regions of interest;

5: Map the regions of interest with higher scores to the feature map extracted by CNNs, and do normalization for unified processing;

6: Do bounding regression and classification on the mapped feature maps;

7: Assign the pixels in the regressed regions to decide whether it belongs to an object or not;

8: Calculate the loss based on ground truth and update the weight by stochastic gradient descent;

End For

Repeat the For loop until the end of the last epoch.

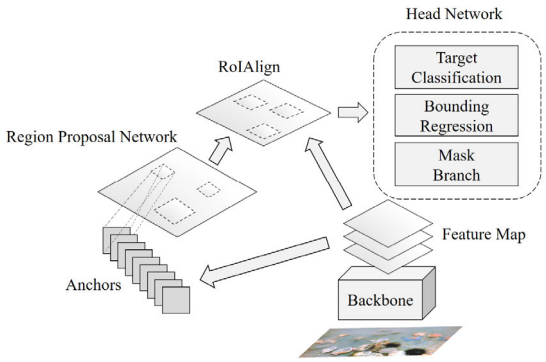


FIGURE 2. Architecture of mask R-CNN.

The structure of Mask R-CNN is illustrated in Fig. 2. It consists of the backbone network, the feature fusion network, the RPN, the RoIAlign (Region of Interest Align), and the head network. The backbone network, also known as the feature extraction network, is used to extract features of different scales from images. The extracted features are called a feature map. For the extraction of feature maps of different scales with rich semantic information, the backbone network usually adopts a deep convolutional network, such as VGG [25], Inception [38], or ResNet (deep residual networks) [39]. The selected deep convolutional network is divided into five stages, and five different scales of feature maps are output from each stage, which are recorded as C1–C5, as shown in Fig. 3. These feature maps of different scales are sent to the feature fusion network after being processed by a 1×1 256-channel convolution kernel. The feature fusion network used in Mask R-CNN is an FPN (feature pyramid network) [40], which superimposes features of different scales from small to large in the form of a pyramid, taking both feature accuracy and rich feature information into account. Finally, FPN outputs the merged features P2–P6. The specific process is depicted in Fig. 3.

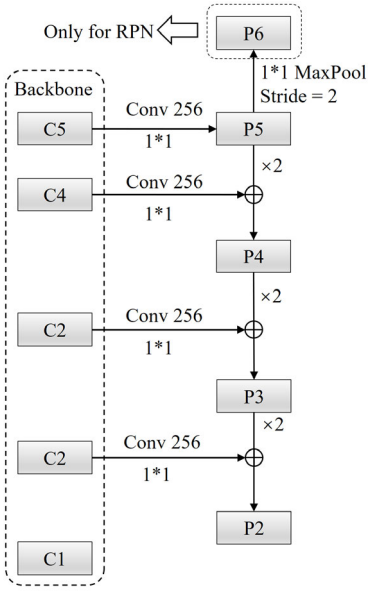


FIGURE 3. Process of feature fusions. Conv represents the convolutional operation. ⊕ denotes the addition of feature maps of the same scale at the pixel level. ×2 represents up-sampling (deconvolution) with magnification being 2. P2–P6 is the feature pyramid. P6, the output of maximum pooled down-sampling, is used only in RPN.

RPN proposes a large number (about 20000) of RoIs (regions of interest) from the feature map, which is the first stage of the two-stage detection. The RoIs proposed at this stage are derived from the anchor method [19]. The specific steps include foreground and background classification, border regression, and a non-maximum suppression algorithm [41] to remove repeated anchor boxes. First, for each point on the five feature maps P2–P6, several anchor boxes are generated according to different aspect ratios and scales. The aspect ratios are usually 1:1, 1:2, and 2:1. Then, the anchor boxes are fed into two fully connected layers to get foreground and background scores, the offsets of the center point, and the zoom ratios of width and height. The foreground and background classification of anchor boxes are obtained by Softmax. Finally, a part of anchor boxes with the highest foreground score is selected for bounding regression,

as shown in (8):

$$\begin{cases} x' = x + \Delta x \times w \\ y' = y + \Delta y \times h \\ h' = h \times e^{\Delta h} \\ w' = w \times e^{\Delta w} \end{cases} \tag{8}$$

where x, y, h, w denote the center point coordinates, height, and width of the anchor box before adjustment, respectively.

$$\Delta x, \Delta y, \Delta h, \Delta w$$

denote the offsets of coordinates and zoom ratios of the bounding box, respectively. x', y', h', w' denote the coordinates, height, and width of the anchor box after adjustment, respectively. The box borders that are beyond the boundary of the image are clipped and the repeated anchor boxes are removed by non-maximum suppression. After the above three steps, RoIs are screened out.

The sizes of RoIs proposed by RPN are not consistent, so the RoIs need to be normalized for uniform processing. In Faster R-CNN, RoIPool [19] is used for normalization. This method includes two quantization processes: mapping the anchor boxes in the picture to feature maps, and normalizing feature maps to a uniform size. However, because of the down-sampling, quantization is bound to cause pixel deviation. In Mask R-CNN, RoIAlign replaces RoIPool to ensure pixel-to-pixel alignment between network inputs and outputs. First, the level k of RoIs is determined according to its width and height, as shown in (9):

$$k = k_0 + log_2(\sqrt{wh}/s_0) \tag{9}$$

where, k_0 and s_0 represent the reference level and reference area, respectively. They are set to 4 and 224 in [36]. For pixel-point matching, bilinear interpolation is used to convert floating-point coordinates to image values, as shown in Fig. 4. Then, the corresponding Pk ($k = 2$ to 5) is selected from the feature maps P2–P5 for feature extraction.

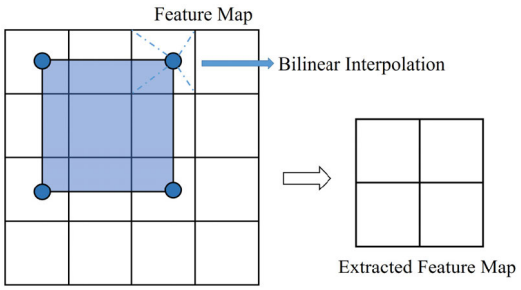


FIGURE 4. Bilinear interpolation resolves boundary mismatching. Values of the blue points are obtained based on the values of surrounding points.

The feature maps processed by RoIAlign are fed into the final head network, including the classification branch, the bounding-box regression branch, and the mask branch. The classification branch and bounding-box regression branch are consistent with those in the RPN network, thus

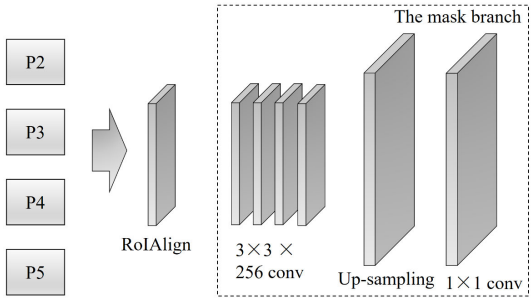


FIGURE 5. Mask branches to decide the assignment of pixels.

are not repeated here. The input of the mask branch is the bounding-box of the second regression, and the output is a mask. Specifically, first, feature maps P2–P5 are processed by RoIAlign; second, the processed feature maps are fed into four convolutional layers and one up-sampling layer; finally, a fully connected layer is used to output the mask. The specific process is shown in Fig. 5.

After these steps, for a specific input picture, an instance segmentation result with accurate bounding boxes, object types, and object masks can be output.

III. IMPLEMENTATION DETAILS

To deal with the challenges of underwater scenes and a small sample dataset, some measures are taken in image processing and model training to manage overfitting, improve accuracy, and save computing resources. The following paragraphs outline the specific implementation details of the proposed method.

A. OVERFITTING MANAGEMENT

To manage overfitting, augmentation is adopted in the preprocessing of the dataset, and transfer learning, freeze training are adopted during training.

The underwater images used in this article are selected from the underwater robot competition UPRC2018 [42]. The whole dataset contains three types of underwater creatures: echinus, starfish, and holothurian, and it presents complex scenes, such as fickle shades, blue or green color tone, and non-uniform sizes, as shown in Fig. 6.

There are only 84 images in the initial dataset (echinus: 183; starfish: 172; holothurian: 149). Thus, the data augmentation is applied to extend the dataset to reduce overfitting. First, we adopt SinGAN [43] to generate hundreds of images from the initial dataset, and the most part of generated images differ from the real world a lot because of the poor quality of the initial images. We pick up 29 images which are close to the real world to extend the dataset, some of which are shown in Fig. 7. Then, the images are flipped upside down and left to right with a 50% probability. Last, Gaussian noise is added to each image with its mean value being 0, and its variance being 255×0.02. The specific effect is shown in Fig. 8. After augmentation, a final dataset which consisted of 430 images is acquired. Specifically, the dataset contains

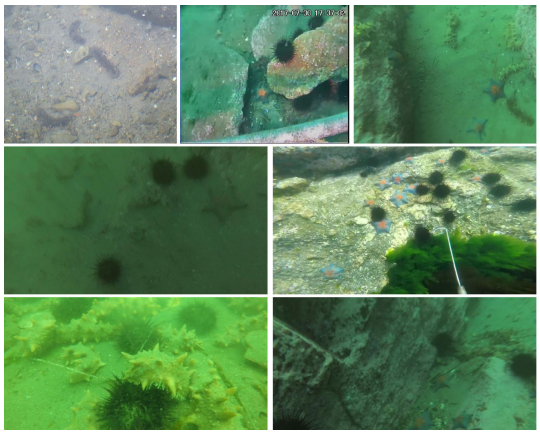


FIGURE 6. Some representative images in UPRC2018. These images are also used in our datasets.

FIGURE 7. Images generated by GAN.

782 echini, 720 starfishes, and 760 holothurians, totaling 2262 creatures. The VIA (VGG Image Annotator) [44] is used for annotation, and annotated information is saved as a JSON (JavaScript object notation) file.

Deep learning is prone to overfitting in the case of a small sample dataset. Except the data augmentation, this issue can also be managed to some extent by using the training weight of COCO dataset for transfer learning and freezing the training weights of C1–C4 (as shown in Fig. 3) during the training process.

B. IMAGE ENHANCEMENT

With the influence of water flow, impurities, and uneven light, underwater images tend to have many problems compared with typical images, resulting in difficulties in detection and recognition. To address this issue, we compare several image enhancement algorithms, including CLAHE (contrast limited adaptive histogram equalization) [45], Dark Channel Prior [46], and MSRCR. The effects of the three algorithms are shown in Fig. 9.

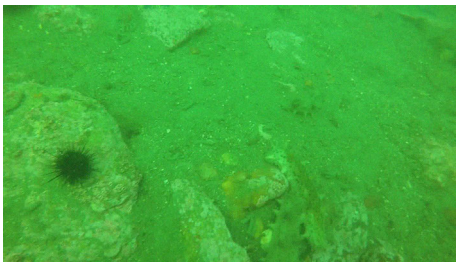


FIGURE 8. An image with Gaussian noises.

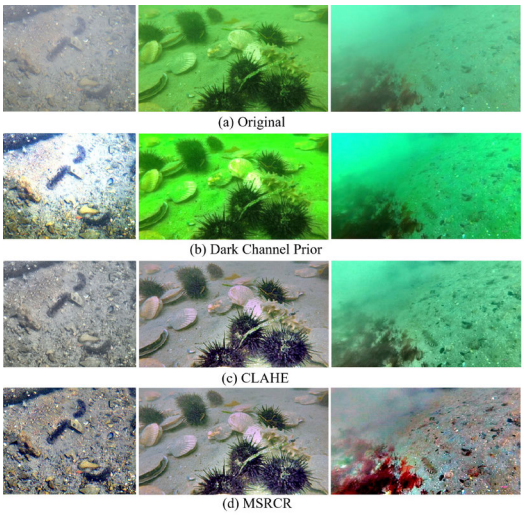


FIGURE 9. Original images and enhanced images.

As can be seen from Fig. 9, MSRCR effectively eliminates the problem of blue or green tone, and has the best visual perception. To quantify their effects, the comentropy, contrast, and sharpness are calculated based on grayscale.

The calculation of comentropy is shown in (10):

$$Comentropy = \frac{1}{Num} \sum \sum_{i=0}^{255} P(i)\, log_2 P(i) \tag{10}$$

where, Num represents the number of images, and $P(i)$ represents probability of pixel value i.

The calculation of contrast is shown in (11):

$$Contrast = \frac{1}{Num} \sum \sum_{\delta} \delta(i,j)^2 \cdot P(\delta) \tag{11}$$

where $\delta(i, j)$ represents the difference between two adjacent pixels i and j, and $P(\delta)$ represents the probability of the difference δ.

The calculation of sharpness is shown in (12).

$$\begin{aligned} Sharpness &= \frac{1}{Num} \sum \left(\sum_Y \sum_X Q(x,y) \right) \\ Q(x,y) &= |f(x,y) - f(x+1,y)| \cdot |f(x,y) - f(x,y+1)| \end{aligned} \tag{12}$$

where Y and X represent the height and width of an image, and $f(x, y)$ represents the value of pixel (x, y).

TABLE 1. Comentropy, contrast and sharpness. And their values are standardized processed with original dataset as baseline.

Dataset	Comentropy	Contrast	Sharpness	Mean
Original	1	1	1	1
Dark Channel	1.068	1.047	0.783	0.966
CLAHE	1.089	1.078	1.186	1.118
MSRCR	**1.121**	**1.109**	**1.689**	**1.306**

The results of the aforementioned quality measures are shown in Table 1, where MSRCR presents the highest score.

Additionally, the RGB histograms are presented to demonstrate the balanced distribution of MSRCR in Fig. 10.

As can be seen in Fig. 9 and Fig. 10, the images with MSRCR show a balanced color distribution effect, and the green channel of Dark Channel Prior rise sharply to near 70000 after the color value of 250, which results in greenish images. Since MSRCR can make the color distribution more balanced while having the best visual perception and the best scores of quality measures, MSRCR is chosen for enhancing the underwater images.

The scale N of MSRCR is set to 3 to balance the running speed and enhancement effects. The *dynamic* is set to 2.5, and the three dimensions of Gaussian function are set to 2, 52, and 152, respectively. The final enhancement effects are shown in Fig. 11. Besides, in this article, the MSRCR is embedded into Mask R-CNN to ensure continuity from image enhancement to model training.

C. IMPLEMENTATION OF MASK R-CNN

We extend the Matterport's Mask R-CNN framework [47] by adding several aforementioned image processing algorithms, such as MSRCR, CLAHE, Dark Channel Prior, flip, and adding noise. The whole framework runs in a Tensorflow, Keras, and OpenCV environment. The batch size is 2, image resize shape is 768 × 768, mini-mask shape is 56 × 56, the number of training RoIs per image is 200, and the training epoch is 60. The configuration of the hardware is as follows. CPU: Intel i3-7100; GPU: Nvidia GeForce 1060 6GB; Memory: dual-channel 16GB DDR4; Operation system: Windows 10. The training time is approximately 19 hours.

Because computing resources are usually limited in underwater vehicles, both real-time performance and accuracy are important. We choose ResNet50 as the backbone, and use the mini-mask method to compress annotation information for saving memory. The mini-mask is a lossy compression that adjusts the mask information to a smaller size and restores it when needed. Using a 100 × 100 mini mask instead of a 1024 × 1024 mask can save more than 99% of memory at the cost of losing pixel segmentation accuracy. The specific performance loss will be discussed in the following section. The principle of the mini-mask is illustrated in Fig. 12.

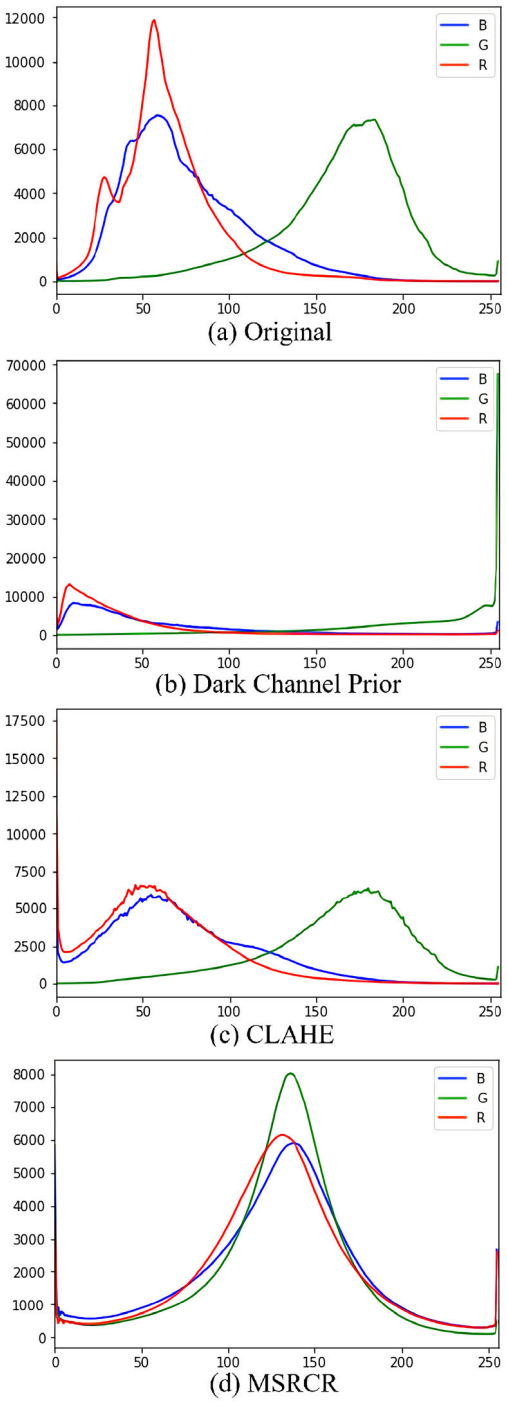


FIGURE 10. RGB histograms. The horizontal axis represents the RGB level, ranging from 0 to 255, and the vertical axis represents the number of pixels.

IV. RESULTS AND DISCUSSIONS

We verify the effectiveness of the proposed method by comparing the test results of some popular target detection models with the results of the proposed method on our dataset. The comparative models include a target detection model based on SIFT and the deep learning methods: SSD [23], YOLOv3 [48], and Mask R-CNN (MRCNN). Additionally,

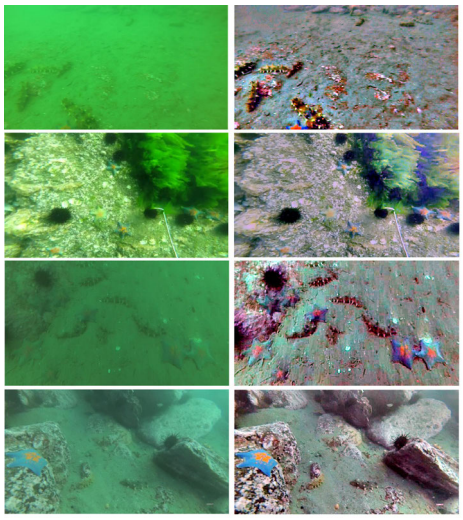


FIGURE 11. Enhancement effects of MSRCR.

TABLE 2. Recognition accuracy of the proposed method and the other methods, calculated based on region. FPS is the number of pictures detected per second.

Model	Recall/%	Precision/%	mAP(IOU50)/%	FPS
SIFT	3.97	100	2.87	0.56
SSD	69.10	79.41	70.62	3.66
YOLOv3	76.56	89.62	79.01	10.67
MRCNN	83.95	94.66	84.63	0.74
Proposed	**94.52**	**97.46**	**94.84**	**0.69**

we apply five-fold cross-validations on the augmented dataset for each model. The specific results are shown in Table 2.

It can be seen from Table 2 that the proposed method has the highest mAP and Recall. Compared with Mask R-CNN, the mAP of the proposed method increases by 12.06% at the expense of a 6.8% reduction in speed. The SIFT-based model achieves a 100% precision, but it is far worse than deep learning models in mAP and Recall, and its speed is the slowest. The main reason for its low speed is that it runs on a CPU, whereas the other deep models run on a GPU. Therefore, the speed of SIFT-based model is only used for references.

The mask precision of the proposed method is 43.31, which is similar to the result in [36]. However, as a result of using the mini-mask, along with the nature of underwater creatures, especially the echinus, the contour edges are not clear, including many protrusions and depressions that cause inaccuracies in the annotation. Therefore, the mask precision is only used for reference as well. The final effect of instance segmentation is shown in Fig. 13, whose corresponding precision-recall is shown in Fig. 14.

As can be seen from Fig. 13, the performance of detection and segmentation are satisfactory. The pixel segmentation of the echinus contour edge is not as satisfactory owing to the appearance characteristics of echinus and the annotation

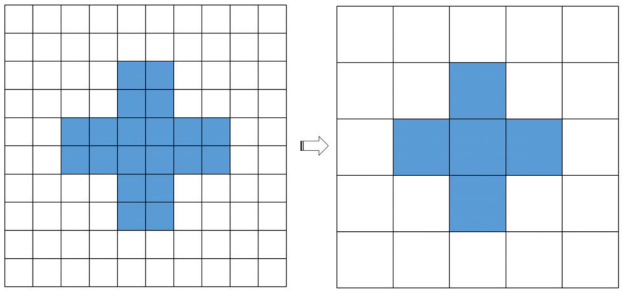


FIGURE 12. Mini-mask converts a 10 × 10 mask to a 5 × 5 mask.

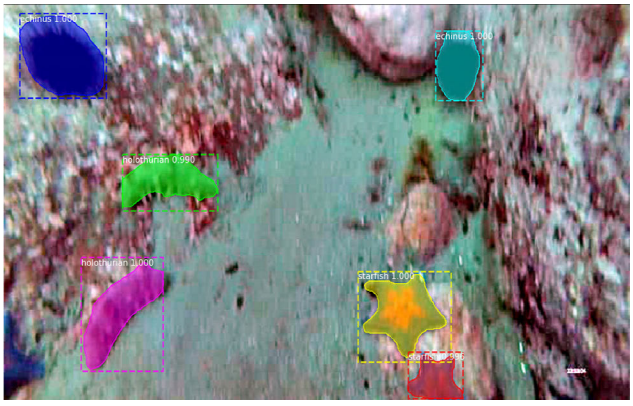


FIGURE 13. Results of classification and instance segmentation finished by the proposed method.

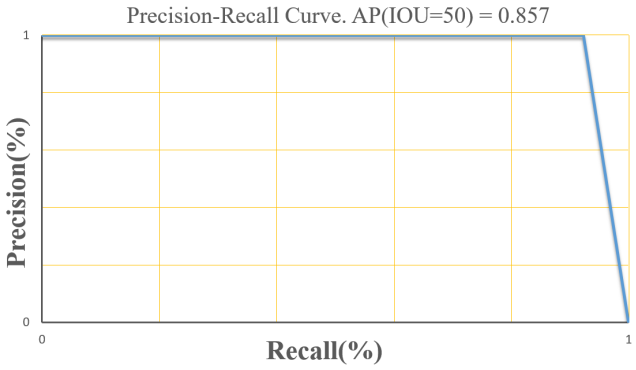


FIGURE 14. Precision–recall curve (corresponding to Fig. 13).

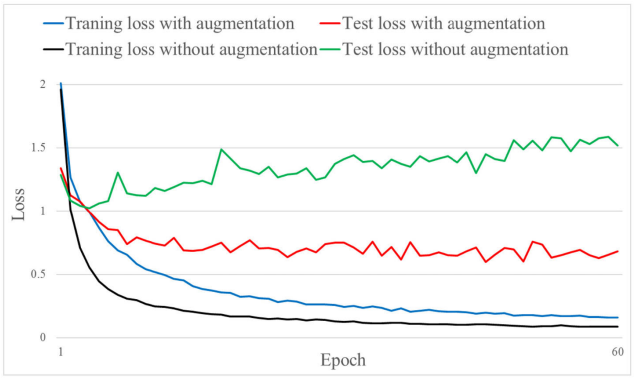
deviation mentioned above. The only missed one is a starfish, whose body is shown in the lower-left corner incompletely.

The effectiveness of the augmentation, which is used to reduce overfitting, is also verified by experiments, as shown in Table 3. Obviously, image augmentation has a positive impact on recognition accuracy. The effect of GAN, which consumes a lot of computing resources, is not distinct. This may be caused by the small number of added images.

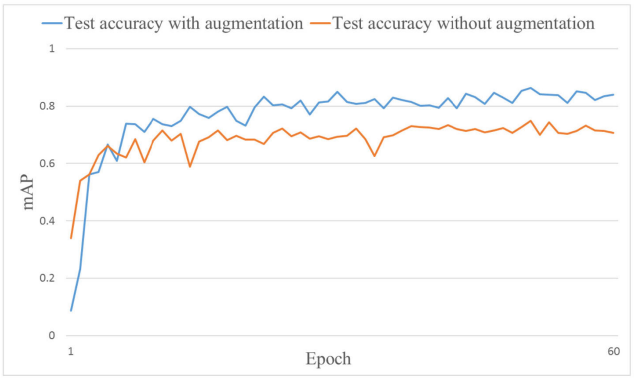
As for the loss, without augmentation, the training loss decreases faster and stabilizes at a smaller value, but its test loss keeps vibrating, as shown in Fig. 15 (a). It indicates that the model without augmentation has a higher degree and faster speed of fitting, but its result becomes worse during

TABLE 3. Recognition accuracy with and without augmentation.

Augmentation	Recall/%	Precision/%	mAP/%
None	70.84	88.50	70.65
Only GAN	71.43	88.69	71.72
All	83.95	94.66	84.63



(a) Loss with respect to epoch



(b) Test mAP with respect to epoch

FIGURE 15. Loss and mAP with respect to epoch. On the augmented dataset and non-augmented dataset.

testing. As shown in Fig. 15 (b), the accuracy without augmentation peaks faster (at the 9th epoch) than the augmented one, and then remains in a lower position. This case demonstrates that, to some extent, an augmented dataset is less likely to be overfitted during training, or the degree of overfitting is relatively small.

In addition, through comparative experiments of different enhancement algorithms, it is demonstrated that all three enhancement algorithms can benefit the feature extraction in an underwater environment. The results are shown in Table 4.

It can be seen that the result of the Dark Channel Prior is the worst among these three algorithms, which is in line with the expectation. The poor performance may be caused by the fact that its primary function is defog inland. Even so, all three algorithms achieve better results than experiments without enhancement, which verifies that an appropriate enhancement algorithm can bring about better training result of CNNs.

TABLE 4. Recognition accuracy with different enhancement algorithms.

Enhancement	Recall/%	Precision/%	mAP/%
None	83.95	94.66	84.63
Dark Channel	87.03	96.59	87.45
CLAHE	89.92	97.69	91.62
MSRCR	**94.52**	**97.46**	**94.84**

TABLE 5. Influences of mini-mask.

Mini-mask	Recall/%	Precision/%	mAP/%	Training time
No	95.64	97.13	96.45	24.9 hours
Yes	**94.52**	**97.46**	**94.84**	**19.0** hours

Finally, through the comparative experiment, it can be found that the result with the mini-mask is not significantly different from the result without the mini-mask. Meanwhile, a large amount of memory is saved and the training speed is improved by 31.05%. Specific results are shown in Table 5.

V. CONCLUSION

This article proposes an object detection and instance segmentation method that incorporates the MSRCR enhancement algorithm into the Mask R-CNN framework to detect and segment underwater creatures on a small sample dataset. Through comparative experiments, it is shown that the accuracy of the proposed method is improved compared with a conventional method (SIFT) or popular deep learning methods (SSD, YOLOv3, Mask R-CNN). Additionally, by testing different enhancement algorithms, this article demonstrates that appropriate image enhancement algorithms can improve the accuracy of deep learning models in an underwater scenario with small sample datasets. This improvement is proportional to the objective assessments of the images. Besides, the effectiveness on reducing overfitting of the augmentation methods (flip, adding noise, GAN) was validated too.

This article provides a viable solution to the development of an underwater optical vision system. However, considering the scarcity of computing resources in the underwater condition, the practical application of the proposed method is still challenging because of its low speed. Besides, like most underwater optical vision systems, this method is not suitable for long-distance underwater object recognition. In future work, we will strive to improve the computational efficiency of the model and continue to expand our dataset.

REFERENCES

[1] A. Hans, "Laser spectroscopy for monitoring and research in the ocean," *Phys. Scripta*, vol. 1998, no. T78, pp. 68–72, Nov. 2006.
[2] M. The Vu, H.-S. Choi, J. Kang, D.-H. Ji, and S.-K. Jeong, "A study on hovering motion of the underwater vehicle with umbilical cable," *Ocean Eng.*, vol. 135, pp. 137–157, May 2017.
[3] P. Gjanci, C. Petrioli, S. Basagni, C. A. Phillips, L. Boloni, and D. Turgut, "Path finding for maximum value of information in multi-modal underwater wireless sensor networks," *IEEE Trans. Mobile Comput.*, vol. 17, no. 2, pp. 404–418, Feb. 2018.

[4] K. Chua and A. M. Rizal, "Robotics vision-based heuristic reasoning for underwater target tracking and navigation," *Int. J. Adv. Robot. Syst.*, vol. 2, no. 3, pp. 245–250, Feb. 2005.

[5] C. Xavier, G. Rafael, and R. Pere, "An approach to vision-based station keeping for an unmanned underwater vehicle," in *Proc. IEEE Int. Conf. Intell. Robot. Syst.*, Lausanne, Switzerland, 2002, pp. 799–804.

[6] J.-P. Tarel, N. Hautiere, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer, "Vision enhancement in homogeneous and heterogeneous fog," *IEEE Intell. Transp. Syst. Mag.*, vol. 4, no. 2, pp. 6–20, Apr. 2012.

[7] G. Huilin, Z. Zhiyu, K. Lou, W. Wei, L. Runbang, D. Robertas, and W. Marcin, "Classification of infrared objects in manifold space using Kullback-Leibler divergence of Gaussian distributions of image points," *Symmetry*, vol. 12, no. 3, pp. 434–447, Mar. 2020.

[8] J. Shen, T. Fan, M. Tang, Q. Zhang, Z. Sun, and F. Huang, "A biological hierarchical model based underwater moving object detection," *Comput. Math. Methods Med.*, vol. 2014, no. 3, pp. 1-8, Jul. 2014.

[9] Y. Wang, L. Fu, K. Liu, R. Nian, T. Yan, and A. Lendasse, "Stable underwater image segmentation in high quality via MRF model," in *Proc. OCEANS MTS/IEEE Washington*, Washington, DC, USA, Oct. 2015, pp. 1–4.

[10] H. Zheng, X. Sun, B. Zheng, R. Nian, and Y. Wang, "Underwater image segmentation via dark channel prior and multiscale hierarchical decomposition," in *Proc. OCEANS Genova Discov. Sustain. Ocean Energy New World*, Genova, Italy, May 2015, pp. 1–4.

[11] K. Srividhya, "Intelligent object recognition in underwater images using evolutionary-based Gaussian mixture model and shape matching," *Signal Image Video Process.*, vol. 14, no. 5, pp. 877–885, Feb. 2020.

[12] K. Alex, S. Ilya, and H. Geoffrey, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Proces. Syst.*, Lake Tahoe, NV, USA, 2012, pp. 1097–1105.

[13] D. A. Konovalov, A. Saleh, M. Bradley, M. Sankupellay, S. Marini, and M. Sheaves, "Underwater fish detection with weak multi-domain supervision," 2019, *arXiv:1905.10708*. [Online]. Available: http://arxiv.org/abs/1905.10708

[14] A. Salman, A. Jalal, F. Shafait, A. Mian, M. Shortis, J. Seager, and E. Harvey, "Fish species classification in unconstrained underwater environments based on deep learning," *Limnol. Oceanogr. Methods*, vol. 14, no. 9, pp. 570–585, Sep. 2016.

[15] H. Qin, X. Li, J. Liang, Y. Peng, and C. Zhang, "DeepFish: Accurate underwater live fish recognition with a deep architecture," *Neurocomputing*, vol. 187, pp. 49–58, Apr. 2016.

[16] A. Salman, S. A. Siddiqui, F. Shafait, A. Mian, M. R. Shortis, K. Khurshid, A. Ulges, and U. Schwanecke, "Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system," *ICES J. Mar. Sci.*, vol. 77, no. 4, pp. 1295–1307, Feb. 2019.

[17] W. Xu and S. Matzner, "Underwater fish detection using deep learning for water power applications," 2018, *arXiv:1811.01494*. [Online]. Available: http://arxiv.org/abs/1811.01494

[18] H. Huang, H. Zhou, X. Yang, L. Zhang, L. Qi, and A.-Y. Zang, "Faster R-CNN for marine organisms detection and recognition using data augmentation," *Neurocomputing*, vol. 337, pp. 372–384, Apr. 2019.

[19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[20] H. T. Rauf, M. I. U. Lali, S. Zahoor, S. Z. H. Shah, A. U. Rehman, and S. A. C. Bukhari, "Visual features based automated identification of fish species using deep convolutional neural networks," *Comput. Electron. Agricult.*, vol. 167, Dec. 2019, Art. no. 105075.

[21] Z. Qiu, Y. Yao, and M. Zhong, "Underwater sea cucumbers detection based on pruned SSD," in *Proc. IEEE 3rd Adv. Inf. Manage., Communicates, Electron. Autom. Control Conf. (IMCEC)*, Chongqing, China, Oct. 2019, pp. 738–742.

[22] Z.-U. Rahman, D. J. Jobson, and G. A. Woodell, "Investigating the relationship between image enhancement and image compression in the context of the multi-scale retinex," *J. Vis. Commun. Image Represent.*, vol. 22, no. 3, pp. 237–250, Apr. 2011.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," 2015, *arXiv:1512.02325*. [Online]. Available: http://arxiv.org/abs/1512.02325

[24] A. Mahmood, M. Bennamoun, S. An, F. Sohel, F. Boussaid, R. Hovey, G. Kendrick, and R. B. Fisher, "Coral classification with hybrid feature representations," in *Proc. Int. Conf. Image Process. (ICIP)*, Phoenix, AZ, USA, 2016, pp. 519–523.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[26] S. Cao, D. Zhao, X. Liu, and Y. Sun, "Real-time robust detector for underwater live crabs based on deep learning," *Comput. Electron. Agricult.*, vol. 172, May 2020, Art. no. 105339.

[27] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," 2018, *arXiv:1801.04381*. [Online]. Available: http://arxiv.org/abs/1801.04381

[28] V. Zavrtanik, M. Vodopivec, and M. Kristan, "A segmentation-based approach for polyp counting in the wild," *Eng. Appl. Artif. Intell.*, vol. 88, Feb. 2020, Art. no. 103399.

[29] M. O'Byrne, V. Pakrashi, F. Schoefs, and B. Ghosh, "Semantic segmentation of underwater imagery using deep networks trained on synthetic imagery," *J. Mar. Sci. Eng.*, vol. 6, no. 3, pp. 93–102, Aug. 2018.

[30] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[31] C. Hubert and J. Ganesh, "Training dataset extension through multiclass generative adversarial networks and K-nearest neighbor classifier," in *Proc. Commun. Comput. Info. Sci.*, Solapur, India, 2019, pp. 596–610.

[32] B. Jahic, N. Guelfi, and B. Ries, "Software engineering for dataset augmentation using generative adversarial networks," in *Proc. IEEE 10th Int. Conf. Softw. Eng. Service Sci. (ICSESS)*, Beijing, China, Oct. 2019, pp. 59–66.

[33] M. Jian, Q. Qi, H. Yu, J. Dong, C. Cui, X. Nie, H. Zhang, Y. Yin, and K.-M. Lam, "The extended marine underwater environment database and baseline evaluations," *Appl. Soft Comput.*, vol. 80, pp. 425–437, Jul. 2019.

[34] M. Jian, W. Zhang, H. Yu, C. Cui, X. Nie, H. Zhang, and Y. Yin, "Saliency detection based on directional patches extraction and principal local color contrast," *J. Vis. Commun. Image Represent.*, vol. 57, pp. 1–11, Nov. 2018.

[35] M. Li, J. Liu, W. Yang, X. Sun, and Z. Guo, "Structure-revealing low-light image enhancement via robust retinex model," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2828–2841, Jun. 2018.

[36] H. Kaiming, G. Georgia, D. Piotr, and G. Ross, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[37] E. H. Land, "The retinex theory of color vision," *Sci. Amer.*, vol. 237, no. 6, pp. 108–128, Dec. 1977.

[38] S. Christian, L. Wei, J. Yangqing, S. Pierre, R. Scott, A. Dragomir, E. Dumitru, V. Vincent, and R. Andrew, "Going deeper with convolutions," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, 2015, pp. 1–9.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[40] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 936–944.

[41] A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proc. Int. Conf. Pattern Recognit.*, Hong Kong, 2006, pp. 850–855.

[42] *UPRC2018*. [Online]. Available: http://2018.cnurpc.org

[43] T. R. Shaham, T. Dekel, and T. Michaeli, "SinGAN: Learning a generative model from a single natural image," 2019, *arXiv:1905.01164*. [Online]. Available: http://arxiv.org/abs/1905.01164

[44] D. Abhishek and Z. Andrew, "The VIA annotation software for images, audio and video," in *Proc. ACM Int. Conf. Multimedia (MM)*, Nice, France, 2019, pp. 2276–2279.

[45] J. Duan, M. Bressan, C. Dance, and G. Qiu, "Tone-mapping high dynamic range images by novel histogram adjustment," *Pattern Recognit.*, vol. 43, no. 5, pp. 1847–1862, May 2010.

[46] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 12, pp. 2341–2353, Dec. 2011.

[47] A. Waleed. *Mask R-CNN for Object Detection and Instance Segmentation on Keras and Tensorflow*. Accessed: Mar. 20, 2018. [Online]. Available: https://github.com/matterport/Mask_RCNN

[48] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767

SHAOJIAN SONG (Member, IEEE) received the B.S. degree in industrial electrical automation and the M.S. degree in control science and engineering from Guangxi University, Nanning, China, in 1994 and 2001, respectively. Since 1994, he has been with the School of Electrical Engineering, Guangxi University, where he became a Professor, in 2010. He was with the New York State Center for Future Energy Systems, Rensselaer Polytechnic Institute, USA, from 2014 to 2015. His current research interests include optimization control, machine learning, power electronics and energy conversion, active distribution networks, and state estimation. He is also a Reviewer of journals such as the IEEE TRANSACTIONS ON POWER ELECTRONICS, IEEE ACCESS, the IEEE JOURNAL OF EMERGING AND SELECTED TOPICS IN POWER ELECTRONICS, and the *Asian Journal of Control*.

JINGXU ZHU was born in Jiangsu, China, in 1995. He is currently pursuing the master's degree with the School of Electrical Engineering, Guangxi University, Nanning, Guangxi, China. His research interests include pattern recognition and image processing.

XIUHUA LI received the B.S. degree in detection technology and automation equipment and the Ph.D. degree in agricultural electrification and automation from China Agricultural University, Beijing, China, in 2008 and 2012, respectively. Since 2012, she has been with the School of Electrical Engineering, Guangxi University, where she became an Associate Professor, in 2015. She was with Florida University, USA, from 2010 to 2011. Her current research interests include spectral detection technology, remote sensing image analysis, the Internet of Things, and big data. She is also a Reviewer of journals such as *Computers and Electronics in Agriculture* and the *International Journal of Agricultural and Biological Engineering*.

QINGBAO HUANG is currently pursuing the Ph.D. degree in software engineering with the South China University of Technology, Guangzhou, China. He is currently an Associate Professor with the School of Electrical Engineering, Guangxi University, China. His research interests include pattern recognition and image processing, natural language processing, knowledge graph, and multi-modal intelligence.