Small Sample Underwater Target Recognition Based on Mobilenet_ YOLOV4 Algorithm

Jun Zhang*, Xiaohong Peng, Zixiang Liang, Rongfa Chen, ZhaoLi*
School of mathematics and computer science, Guangdong Ocean University, Zhanjiang 524088
**Corresponding author.*

Abstract

Objectives: Underwater target recognition through simulation robot, or manual acquisition of seabed image data, the cost of sampling is high, the sample data obtained is limited, and the image quality is poor, and the data can be used for training is small. Methods: Aiming at this problem, this paper improves the algorithm based on yoov4, modifies its feature extraction backbone network, and proposes three kinds of YOLOV4 algorithms based on different Mobile net backbone networks to test the underwater target recognition in the case of small samples. In this paper, the real image of the seabed is used as the original data for training, and the data which is different from the training set is used for prediction. Result: Compared with the original YOLOV4 algorithm under the same conditions, the experimental results of MobilenetV1_YOLOV4 algorithm has the best MAP(86.04%) and FPS(52); and the histogram equalization method is used to enhance the image, which can be used as a further supplementary recognition of the missed target, and reduce the missed rate. Conclusions: The algorithm takes into account both lightweight and accuracy, and provides support for underwater target recognition in marine operation development and aquaculture.

Keywords*: YOLOV4, mobile net, underwater target recognition, small sample, image processing*

I. Background

It is an important strategy for China's agricultural modernization to vigorously exploit the marine resources and develop modern marine fishery. To rationally develop and utilize marine fishery resources, strengthen marine environmental protection and promote marine ecological restoration [1], we should take the road of modern marine fishery development with high efficiency, product safety and resource conservation. This requires us to effectively protect fishery resources and strengthen the management of fishery resources. At present, in the field of marine pasture construction, most of the fishing methods of sea cucumbers, sea urchins, starfish, scallops and other seafood are still costly and inefficient, and even cause serious damage to the seabed ecology [2]. With the development of AI (artificial intelligence), it is hoped to use underwater robots to catch sea treasures, and the key lies in the accurate identification of species.

Southeast China is close to the sea, with vast sea area and long coastline. The network of inland waters is complicated, with abundant fishery resources [3]. There are numerous fishery varieties, including more than 3,000 kinds of sea and freshwater fish known and more than 150 common species [4]. However, it is still a big problem for the identification of marine species. When underwater bionic robots shoot underwater, they will inevitably be affected by the marine environment, such as vivid plants. Besides, underwater pictures will also be affected by noise, light, low contrast, and fuzzy details, which will have a great impact on target identification.

In the fishery field, target recognition is increasingly perfect. French [5] and Chen [6] have adopted CNN network

to realize fish recognition and classification detection. However, with the rise of deep learning, target detection methods are more diffusely applied in the fishery field. For example, Du Weidong [7] proposed a fish recognition method based on SVM and multi-directional data decision fusion, with an accuracy rate of over 90%; Lin Mingwang [8] et al. used VGG16 model to classify and identify fish, but the stability of the data image is still poor due to the large background interference; Wang Wencheng [9] et al. realized the algorithm of fish recognition and detection based on deep learning, and classified and identified them with different algorithms by using ResNet50 network model; Li Chongchong [10] et al. improved the regression loss function of YOLOV3 prior frame and prediction frame, and adopted the recognition and detection of underwater fish targets based on YOLOV3 model; Yuan Chunhong [11] et al. used Faster R-CNN target detection method in underwater fish species recognition, with an accuracy rate as high as 98.12%. But the processing was slow.

At present, the target detection technology has some shortcomings in underwater target detection, such as low recognition accuracy, high equipment requirements, long time-consuming training model, etc. As a result, it is necessary to build a deep learning network with both lightweight and accuracy, so as to accurately detect and recognize seafood in real time. To solve this problem, this paper, based on YOLOV4 algorithm, improves the backbone of feature extraction network by using four kinds of marine organisms captured by public underwater cameras, and replaces the common convolution in PANet (feature fusion module) with depthwise separable convolution to further reduce the amount of parameters and lessen the computation, so as to train the model and identify marine organisms. Meanwhile, reasonable optimization and parameter adjustment are carried out to increase the recognition accuracy and speed as much as possible, and improve the robustness of the model.

II. Introduction of YOLOV4

2.1 Introduction of YOLOV4 algorithm

YOLO, a one-shot target detection technology, was introduced by Joseph Redmon and Ali Farhadi in 2016. At present, there are currently five versions of the technology. YOLOV4 [12], especially its optimizer, uses two optimization functions of bags: BoF used during training refers to the method of increasing training cost without increasing reasoning cost to improve detection accuracy, which generally is data enrichment. BoS used during reasoning is a method of improving target detection accuracy by increasing reasoning cost and changing network structure.

YOLO (You Only Look Once) regards target detection as a regression problem to solve. It is based on a single end-to-end network, which unifies the classification process of processing training process. The network undertakes the whole processing. When processing images by traditional methods, firstly, we need to manually design features, extract image feature points, and input the extracted features as models into the classifier. After using deep learning method, feature extraction is carried out by convolutional neural network to complete processes from the input of original image to the output of object position and category. In terms of network design, the main difference between YOLO and RCNN [13], Fast RCNN [14] and Faster RCNN [15] is that YOLO training and detection are carried out in a single network. Nevertheless, RCNN, Fast RCNN use separate modules to obtain candidate frames, and the training process is also divided into different modules; secondly, YOLO takes object detection as a regression problem to solve, and the positions of all objects in the image, their categories and their corresponding confidence probabilities can be obtained by reasoning once in the input image. However, RCNN, fast RCNN, and Faster RCNN divides the detection results into two parts: object classification and object position regression.

Secondly, YOLOV4 improves the network structure on the basis of YOLOV3 [16]. CSPDarknet53 [17] is adopted

in the main part, and tricks such as mish activation function and Mosaic data expansion are adopted to further improve the model accuracy. SPP module and PANet are also added, and the prediction part is generally consistent with YOLOV3.

2.2 YOLOV4 algorithm structure

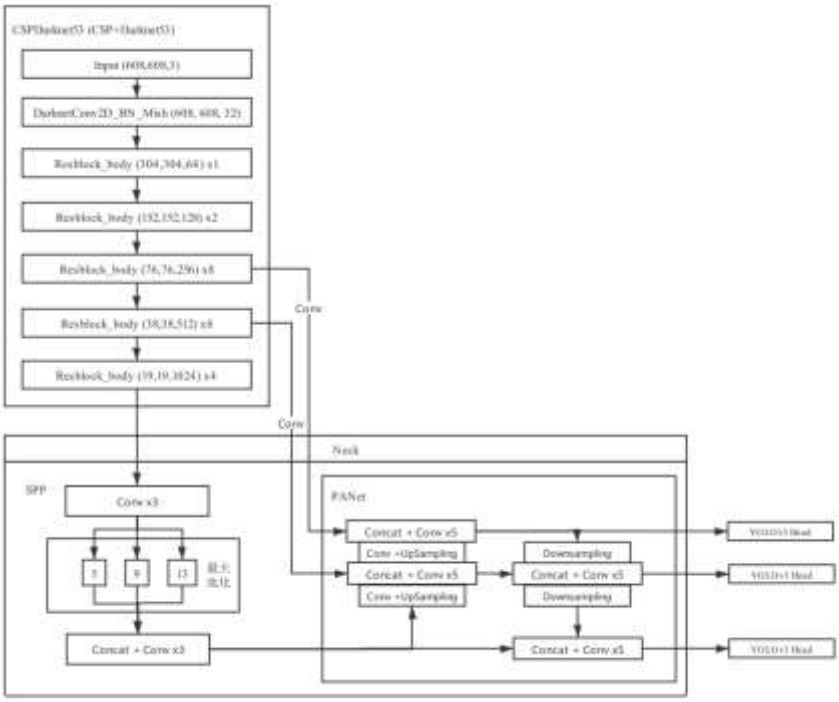


Fig 1: YOLOV4 network structure diagram

CSPDarknet and YOLOV4 use CSP connection, and Darknet-53 as the backbone of feature extraction. Compared with ResNet-based design, CSPDarknet53 model has higher target detection accuracy. However, Mish and other technologies can promote the classification accuracy of CSPDarknet53. In addition, the CSPDarknet structure can also reduce parameters, thus reducing the amount of calculation [18]. Although some parameters are reduced, the parameters required training are still as large as 64M.

It can be seen from the figure that the mish activation function is not completely truncated when it is negative, but allows a relatively small negative gradient inflow, thus avoiding the gradient saturation problem. For example, sigmoid and tanh activation functions usually have gradient saturation problems. In the limit case of both sides, the gradient approaches 1. Nevertheless, the mish activation function cleverly avoids this problem. In addition, the mish function also ensures the smoothness of every point, so that its gradient reduction effect is better than that of the relu function [19].

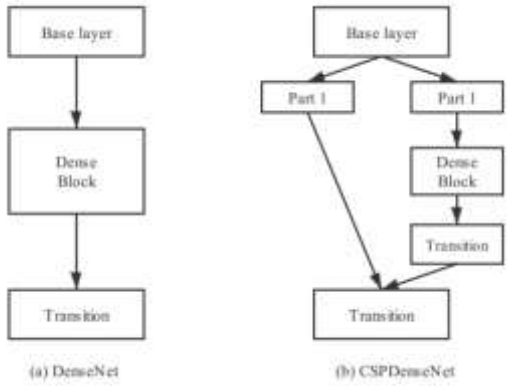


Fig 2: CSPDENSET structure

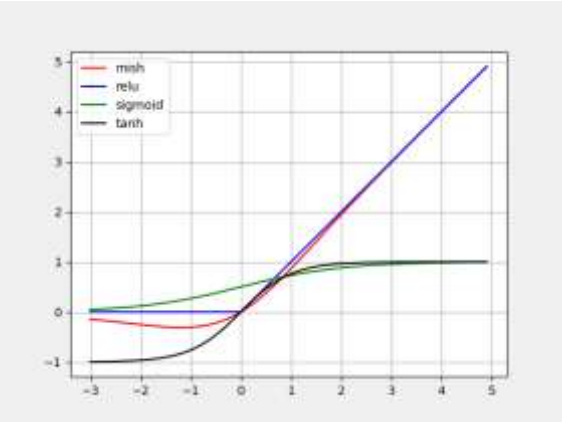


Fig 3: mish and common activation functions

Mosaic data enrichment is an extension of CutMix, which mixes two pictures. Mosaic data enrichment mixes four pictures with different semantic information. It can not only indirectly increase the training set in disguise, but also detect targets beyond the conventional context, thus reducing the training difficulty of the model and imporve the robustness of the model.

For SPP [20] spatial pyramid pooling, YOLOV4 uses three maximum pooling methods, namely 5×5, 9×9 and 13×13, for multi-scale feature fusion, which can increase the receiving range of backbone features more effectively.

CIoU. IoU is an important index in target detection $\square\square\square = \frac{|\square^{\square} \cap \square^{\square}|}{|\square^{\square} \cup \square^{\square}|}$. It is calculated by the ratio of intersection and union between prediction frame and real frame, and is often used to evaluate the advantages and disadvantages of prediction frame. However, L2 norm is generally used for fine adjustment of prediction frames. Some studies show that it is not a method to optimize IoU, and IoU loss appears as a result. Furthermore, based on DIoU, CIoU is proposed. CIoU's penalty term is to add an influence factor $\square, \square$ to DIoU's penalty term. This factor take into account of the length-width ratio of that prediction frame fitting the length-width ratio of the target frame. We define it as:

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(b, b^g)}{c^2} + \alpha v$$

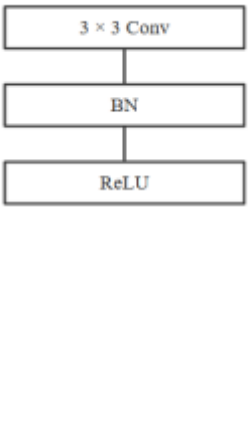
$$\alpha = \frac{v}{(1 - IoU) + v}$$

$$v = \frac{4}{\pi^2}(arctan\frac{w^g}{h^g} - arctan\frac{w}{h})^2$$

2.3 Introduction of Mobilenet

The MobileNet model is a lightweight deep neural network for embedded devices. The core idea of its usage is depthwise separable convolution. It can split a standard convolution into two structures, namely depthwise convolution and pointwise convolution, thus generating new features and effectively reducing network parameters and model size. The number of feature maps after depthwise convolution (dw) is the same as the number of

channels in the input layer, which cannot be expanded. Moreover, this operation independently convolutes each channel of the input layer, and does not effectively utilize the feature information of different channels in the same spatial position. Therefore, pointwise convolution is needed to combine these feature maps. The structure of depthwise separable convolution and Mobilenet is as follows:



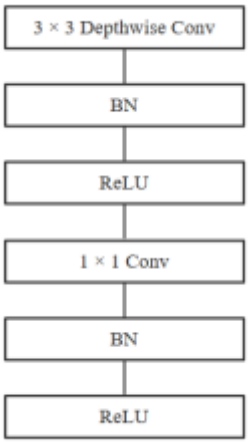


Fig 4: General convolution *Fig 5: Depthwise separable convolution*

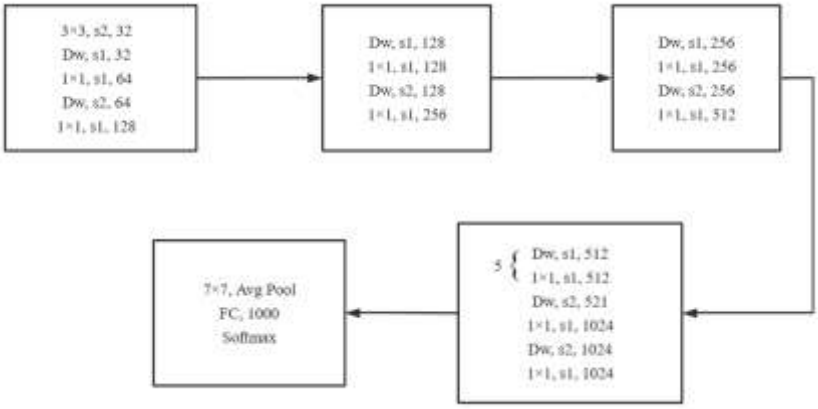


Fig 6: Mobilenet network structure

Assuming that the input and output of the standard convolution are the feature graph of $\square_{\square} \times \square_{\square} \times \square$and the feature graph of $\square_{\square} \times \square_{\square} \times \square$ respectively, and the convolution kernel size is $\square \times \square$, the calculation formula of the output feature graph is:

$$\boldsymbol{G}_{k,l,n} = \sum_{i,j,m} \boldsymbol{K}_{i,j,m} \cdot \boldsymbol{F}_{k+i-1,j+j-1,m}$$

Mobilenet optimizes the computational complexity through depthwise separable convolution, and transforms the standard convolution into depthwise convolution and pointwise convolution. BN layer and ReLU activation function are connected behind each layer. Each input dimension of depthwise convolution corresponds to a convolution kernel. For the same input, the calculation formula of output feature graph of depthwise convolution

is:

$$\widehat{\boldsymbol{G}}_{k,l,n} = \sum_{i,j,m} \boldsymbol{K}_{i,j,m} \cdot \widehat{\boldsymbol{F}}_{k+i-1,j+j-1,m}$$

In MobilenetV1 [21], to make the structure smaller and the computation less, two parameters are introduced. One is $\square$ called width factor. The function of the width factor α is to multiply the channels of each layer by a certain proportion in each layer of sparse network, thus reducing the number of channels of each layer. Common values are 1, 0.75, 0.5 and 0.25. Another parameter is $\square$ called resolution factor. Its function is to multiply the size of the feature map in each layer by a certain proportion. The interaction of two parameters can balance the speed and accuracy, and reduce the amount of calculation by reducing the number of parameters. The parameter quantity $\square$ of a standard convolution $\square$ is $\square = \square \times \square \times \square^2$. The parameter quantity $\square'$ of depthwise separable convolution $\square'$ is $\square' = \square^2 \times \square + 1 \times 1 \times \square \times \square$. Wherein, $\square^2$ represents the kernel size of convolution, and $\square, \square$ represent the number of input and output channels. The ratio of the two parameters is:

$$\frac{G'}{G} = \frac{K^2 \times M + 1 \times 1 \times M \times N}{M \times N \times K^2} = \frac{1}{K^2} + \frac{1}{N} \sim \frac{1}{K^2}$$

It can be estimated from the above formula that if the convolution kernel size is selected as 3x3, the calculation amount will be reduced by 8-9 times.

MobilenetV2 [22] introduces an inverted residual block structure, which contains shorcut as the residual network unit. The difference is that the structure has few input and output dimensions. The dimension first is expanded by linear convolution. Then, the feature is extracted by deep convolution. Finally, the dimension is reduced by mapping. The structure can well maintain the network performance and make the network lighter. Since the depthwise separable convolution in MobilrnetV1 cannot change the number of channels, the extracted features are limited by the number of input channels. As a result, number of channels is increased first, and then the same depthwise separable convolution as MobilenetV1 is performed. The structure diagram and parameters of an inverted residual block are as follows:

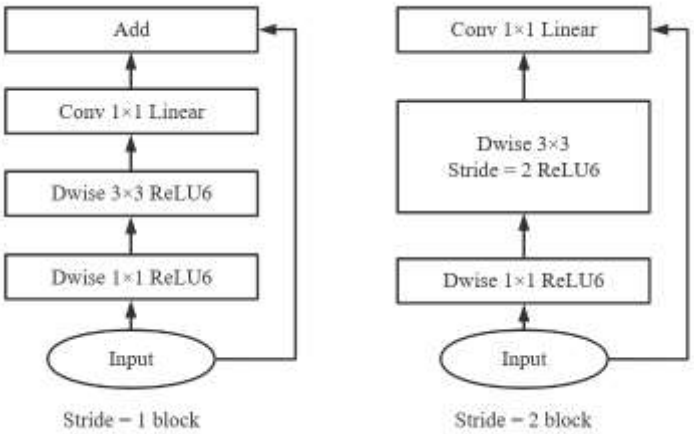


Fig 7: Inverted residual structure

$$Params = M \times tM + K^2 \times M + 1 \times 1 \times tM \times N$$

To avoid the loss of features by ReLU function, after 11 convolutions to decrease the number of channels, ReLU activation function is abandoned, and linear activation function is used to sum by element. This consideration is based on the fact that the number of channels has been reduced after 11 convolution, which has caused the loss of certain information. When the number of feature channels becomes smaller, most of the values tend to be less than 0. Finally, after ReLU activation function, some feature information will be lost again. Therefore, by expanding first and using linear activation, information loss can be avoided, thus improving the accuracy.

MobilenetV3 [23] is obtained by adding SE attention module on the basis of MobilenetV1 depthwise separable convolution and MobilenetV2 inverted residual block, and the number of channels of all SE modules is set as the number of expanded channels. This not only increases the accuracy, but also does not affect the network efficiency. Since computing sigmoid by embedded devices will consume considerable computing resources, the author proposed h-swish as an activation function instead of swish. And with the deepening of network, the cost of nonlinear activation function will also decrease. Therefore, only using h-swish in deeper layers can gain greater advantages. This nonlinearity has many advantages while maintaining the accuracy. ReLU6 can be implemented in many software and hardware frameworks. Secondly, h-swish avoids the loss of numerical accuracy when quantizing, reduces the computation and improves the performance of the model.

$$\begin{cases} swish(x) = x \cdot \sigma(x) \\ h - swish(x) = x \cdot \dfrac{ReLU6(x+3)}{6} \end{cases}$$

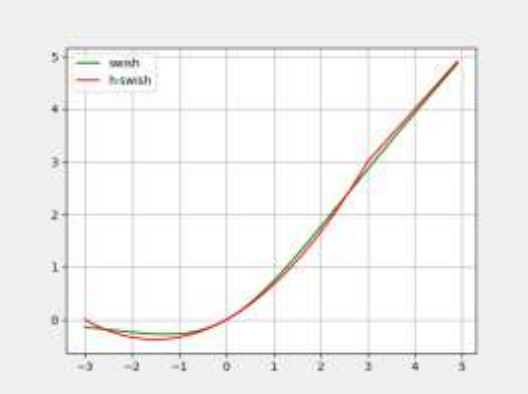


Fig 8: swish and h-swish activation functions

In this paper, three network structures are used to respectively replace CSPDarknet, the feature extraction backbone network in YOLOV4. The improved network structure and algorithm flow are as follows:

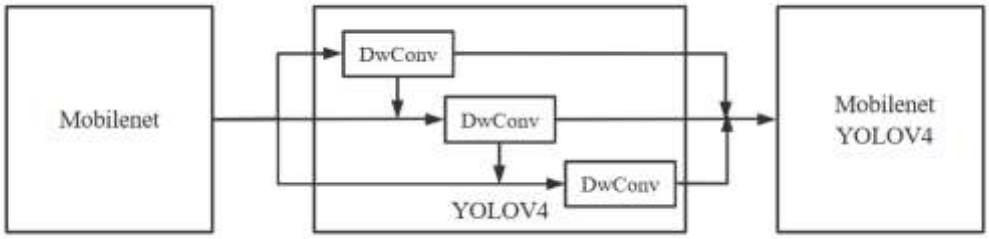


Fig 9: Structure diagram of Mobilenet_ YOLOV4

Algorithm flow

Step 1: Input the original 416×416×3 images into different Mobilenet series backbones (width factor α is selected as 1). Three different effective feature layers will be obtained, including:

Table 1 Size of three original feature layers

	MobilenetV1	MobilenetV2	MobilenetV3
Feature1	52*52*256	52*52*32	52*52*40
Feature2	26*26*512	26*26*96	26*26*112
Feature3	13*13*1024	13*13*320	13*13*160

Step 2: Convolve the last 13*13 feature layers three times (two darknet convolutions and one depthwise separable convolution), and then perform the maximum pooling operation three times. After that, carry out concat operation on them and conduct down-sampling. The calculation formula of cancat is as follows (m,n represents the amount of input and output channels respectively):

$$Z_{concat} = \sum_{i=1}^{m} D_I * K_i + \sum_{i=1}^{n} D_I * K_{i+n}$$

Step 3: perform one darknet convolution on the feature layers of 26*26 and 52*52 respectively, and then perform cancat operation on the feature layer obtained from the previous down-sampling. After that, perform five convolutions (three darknet convolutions and two depthwise separable convolutions);

Step 4: perform depthwise separable convolution, darknet convolution and down-sampling on the three feature layers obtained, and then carry out cancat operation on the previous feature layer. Three different feature layers will be obtained, including:

$$y_i = (batch\_size, 13 * i, 13 * i, 3{,}85), i = 1{,}2{,}3$$

Step 5: send these three feature layers to YOLO_head module for prediction, stack the results of each feature layer and compare them with the set threshold to get the prediction results and output them.

After using three effective feature layers obtained from three Mobilenet series backbone structures to replace the feature layers in the original YOLOV4, the following table can be obtained by comparing the network structure parameters with the original YOLOV4 algorithm:

Table 2 Parameters after modification of backbone network

Network structure	Parameter quantity (millions)
YOLOV4	64.4
MobilenetV1-YOLOV4	41.0
MobilenetV2-YOLOV4	39.1
MobilenetV3-YOLOV4	40.0

To further reduce the model parameters and make the whole model C lightweight [24], we replaced the common convolution with the depthwise separable convolution in the original SPP module and PANet module of YOLOV4, which further reduced the parameters. The following table shows the model parameters after replacing the convolution block in PANet module:

Table 3 Parameters after modification of backbone network and PANet

Network structure	Parameter quantity (millions)
MobilenetV1-YOLOV4	12.7
MobilenetV2-YOLOV4	10.8
MobilenetV3-YOLOV4	11.9

In this paper, based on the target recognition with small sample data, the recognition accuracy and speed of the basic YOLOV4 algorithm and the three Mobilenet-YOLOV4 algorithms with greatly reduced parameters are compared. The reduction of parameters can not only effectively reduce the model computation, but also improve the training speed.

III. Experiment and Analysis

3.1 Data training and parameter setting

When preprocessing the image, we labeled the image with Labelimg tool, and classified the objects to be detected into four categories: scallops, holothurian, echinus and starfish. Because of the imbalance of sample data, we included as many samples as possible into the training set. We selected 163 pictures as training data, including four kinds of marine life with shapes as many as possible.

At the same time, to prevent the possible over-fitting phenomenon of small samples during training [25], early stopping, a high-order skill of neural network training, is added in the training stage. For deep neural network, it is expected to reduce the optimized parameters and get better optimization results by deepening the network level. The use of early stopping method can prevent the over-fitting of training by intercepting and saving the parameter model with the best result in the whole process of model training. In this way, the training time will be greatly

reduced and the accuracy will be higher [26]. Here, we set the initial image input size to 416×416×3, turned on Mosaic data enrichment in YOLOV4, and set the learning rate at 10e-4. We turned off label smoothing and applied cosine annealing to update the learning rate.

3.2 Evaluation index

Using different performance indicators to evaluate the algorithm often has different results, that is to say, the quality of the model is relative. The quality of the method depends not only on the algorithm and data, but also on the requirements of the task. Therefore, it is very necessary to select a reasonable model evaluation index. In this paper, Precision, Recall, AP, mAP, F1-Score and FPS are mainly used for evaluation.

We set a threshold value of 0.5. That is, when $IoU > 0.5$, it is considered as effective detection. The number of effective detection frames is recorded as TP; otherwise, it is considered as invalid detection. The amount of Invalid detection frames or the number of repeated detection frames for the same real frame is recorded as FP,and FN is the number of undetected real frames. Then:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F_{\propto} = \frac{(\propto^2 + 1)P * R}{\propto^2 (P + R)}$$

$$when \propto = 1,\ F_1 - Score = \frac{2}{\frac{1}{Precision} + \frac{1}{Recall}} = \frac{2 \times Precision \times Recall}{Precision + Recall} \in [0,1]$$

Generally, we always use Precision and Recall to measure the quality of the model. But at the same time, we should weigh these two quantities. Therefore, we can use F1-Score to combine these two quantities, which is also the harmonic average of Precision and Recall. The bigger the three values are, the better they are. AP represents the accuracy of a single category, and mAP is to average all categories of AP, that is:

$$mAP = \frac{\sum_{q=1}^{Q} AP(q)}{Q}$$

Apart from detection accuracy, another significant evaluation index of target detection algorithm is identify speed. Only when speed is fast can real-time detection be realized. FPS is used to evaluate the speed of target detection, namely the amount of pictures which can be processed per second.

3.3 Comparison of different models

Table 4 Model AP Table

Model	scallops	holothurian	echinus	starfish	mAP
YOLOV4	91%	42%	66%	80%	69.87%
MobilenetV1-YOLOV4	93%	76%	76%	100%	86.04%
MobilenetV2-YOLOV4	69%	44%	61%	79%	63.12%

MobilenetV3-YOLOV4	92%	72%	69%	100%	83.48%

Table 5 FPS table of model

Model	FPS
YOLOV4	38
MobilenetV1-YOLOV4	52
MobilenetV2-YOLOV4	43
MobilenetV3-YOLOV4	46

Table 6 Evaluation indicators of different models

(a)YOLOV4

YOLOV4	Precision	Recall	F1-Score
echinus	84.00%	42.68%	0.57
holothurian	72.73%	24.24%	0.36
scallop	92.16%	61.84%	0.74
starfish	100.00%	60.00%	0.75

(b)MobilenetV1_YOLOV4

MobilenetV1-YOLOV4	Precision	Recall	F1-score
echinus	75.49%	77.00%	0.76
holothurian	72.28%	75.25%	0.74
scallop	88.12%	82.41%	0.85
starfish	100.00%	100.00%	1

(c) MobilenetV2_YOLOV4

MobilenetV2-YOLOV4	Precision	Recall	F1-score
echinus	67.09%	53.00%	0.59
holothurian	74.19%	23.71%	0.36
scallop	83.05%	45.37%	0.59
starfish	66.67%	57.41%	0.92

(d) MobilenetV3_YOLOV4

MobilenetV3-YOLOV4	Precision	Recall	F1-score
echinus	68.82%	64.00%	0.66
holothurian	77.03%	58.76%	0.67
scallop	93.33%	77.78%	0.85
starfish	100.00%	57.14%	0.73

3.4 Target identification and detection

We applied the model obtained from the above training for target recognition [27]. Based on an underwater picture in the same environment, it can be seen from the recognition result diagram that although the picture has low pixels,

low picture quality and occlusion between objects [19], most targets in the picture can still be recognized.

Nevertheless, we still can't recognize the targets in the areas with poor water quality [28]. At the same time, we may enhance the image to be recognized before the target recognition, and then use the trained model for recognition.

At present, the widely applied method for image processing and enrichment is histogram equalization [29], which is a simple and effective image enrichment technology. It changes the gray level of each pixel in the image by changing the histogram of the image, and is mainly used to enhance the contrast of images with small dynamic range. The basic principle of Histogram equalization is that the number of pixels in the image of grey value (i.e., a major role on the image grey value) for broadening, and a small number of pixels of grey value (i.e., does not play a major role on the picture of the grey value) to merge, thus increasing the contrast, make the image clear, reach the purpose of enhancing. However, histogram equalization is proposed for gray images. Considering that underwater images are mostly concentrated in green, we may split the three-channel images, equalize the histograms of each channel, and then fuse the channels to enrich a color image. The enhanced recognition results are as follows:

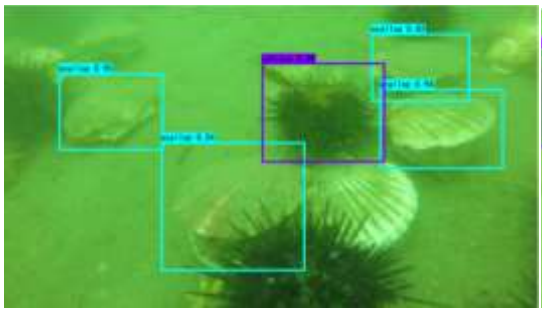


Fig 10: YOLOV4 recognition

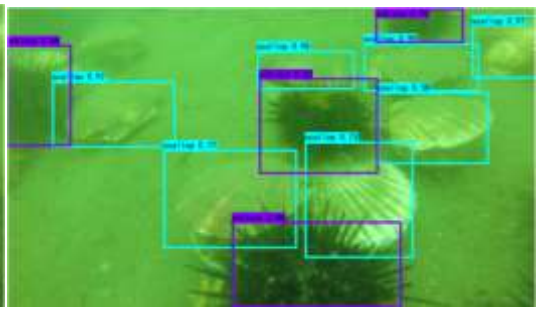


Fig 11: MobilenetV1_YOLOV4 recognition

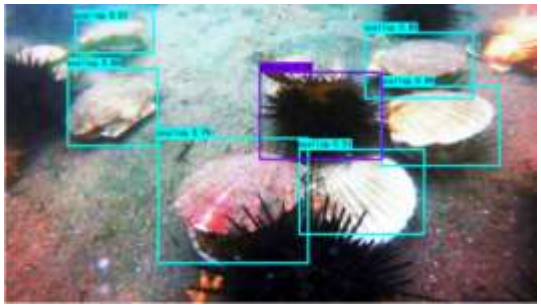


Fig 12: Enhanced YOLOV4 recognition

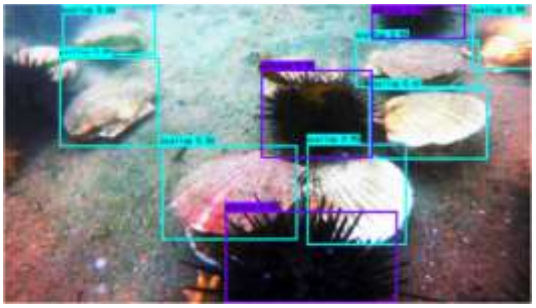


Fig 13: Enhanced recognition

Seen from the recognition images of YOLOV4 algorithm and MobilenetV1_YOLOV4 algorithm, echinus is dark in color, dense and occluded, which leads to incomplete recognition. After using histogram equalization to enhance the image, it can be seen that some unrecognized objects can be recognized when the recognition accuracy has little influence, but some occluded objects will be lost.

IV. Experimental Conclusion and Analysis

In this experiment, the computer is equipped with 16G memory and RTX2060 graphics card, and cuda10.1 for

learning acceleration. In windows10 system, YOLOV4 recognition algorithm and three improved YOLOV4 algorithms are used to train and recognize the deep-sea data set disclosed on the network. By adding the early stopping strategy, the training speed is significantly improved. According to the training on small data sets, MobilenetV1_YOLOV4 model is the best among the four kinds of underwater target recognition, and the mAP is 86.04%; when the water is too turbid, we can use the histogram equalization to enhance the images, which can supplement the recognition of some missed targets with little loss of accuracy. In the future, we may consider using a larger data set to further improve the generalization ability and robustness of the model, or using image enrichment related technology to carry out targeted noise reduction and smoothing before recognition to further improve the recognition rate of the model.

References

[1] H. Q. Ge. “Practice and countermeasures of marine ecosystem management in China and its surrounding waters,” Ecological Economy, vol. 360, no. 12, pp. 170-177, 2020.
[2] L. J. Zhang. “Improvement of China's marine management system from the perspective of marine biodiversity protection,” Journal of Guangdong Ocean University, vol. 30, no. 2, pp. 15-18, 2010.
[3] J. L. Liu, X. J. Chen. “Research progress and hot spot analysis of marine biodiversity,” Progress in Fishery Sciences, vol. 42, no. 01, pp. 204-216, 2021.
[4] W. L. Ji. “A summary of the research on the innovation and development capability of China's marine economy,” Journal of Guangdong Ocean University, vol. 37, no.005, pp. 30-33, 2017.
[5] G. French, M. Fisher, M. Mackiewicz, et al. “Convolutional neural networks for counting fish in fisheries surveillance video,” 2015.
[6] G. Chen, S. Peng, S. Yi. “Automatic fish classification system using deep learning,” IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI), 2017.
[7] W. D. Du, H. S. Li, Y. K. Wei, et al. “Fish identification method based on SVM decision fusion,” Journal of Harbin Engineering University, vol. 000, no. 005, pp. 623-627, 2015.
[8] M. W. Lin. “Application of deep learning in fish image recognition and classification,” Digital Technology & Application, no. 4, pp. 96-97, 2017.
[9] W. H. Wang, H. Jiang, Q. Qiao, et al. “Research on classification and recognition of ten fish images based on ResNet50 network,” Rural Economy and Science-Technology, vol. 471, no. 19, pp. 68-70, 2019.
[10] C. C. Li. “Detection and recognition of underwater fish targets based on YOLOv3,” Journal of Northwest A&F University, 2020
[11] H. C. Yuan, S. Zhang. “Underwater fish target detection method based on Faster R-CNN and image enhancement,” Journal of Dalian Ocean University.
[12] A. Bochkovskiy, C. Y. Wang, H. Liao. “YOLOv4:Optimal speed and accuracy of object detection,” 2020.
[13] R. Girshick, J. Donahue, T. Darrell, et al. “Rich feature hierarchies for accurate object detection and semantic segmentation,” IEEE Computer Society, 2013.
[14] J. N. Li, D. Xiao, et al. “Scale-Aware Fast R-CNN for Pedestrian Detection,” IEEE Transactions on Multimedia, 2017.
[15] S. Ren, K. He, R. Girshick, et al. “Faster R-CNN: towards real-time object detection with region proposal networks,” IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 2017.
[16] J. Redmon, A. Farhadi. “YOLOv3: an incremental improvement,” arXiv e-prints, 2018.
[17] V. Mandal, A. R. Mussah, Y. Adu-Gyamfi. “Deep learning frameworks for pavement distress

classification: a comparative analysis," 2020.
[18] G. Ghiasi, T. Y. Lin, Q. V. Le. "DropBlock: a regularization method for convolutional networks," 2018.
[19] K. He, X. Zhang, S. Ren, et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition," IEEE Transactions on Pattern Analysis & Machine Intelligence, vol. 37, no. 9, pp. 1904-16, 2014.
[20] S. Liu, L. Qi, H. Qin, et al. "Path aggregation network for instance segmentation," IEEE, 2018.
[21] C. Kcab, C. Xmab, W. Wei, et al. "A modified YOLOv3 model for fish detection based on MobileNetv1 as backbone - ScienceDirect," Aquacultural Engineering, 2020.
[22] M. Sandler, A. Howard, M. Zhu, et al. "MobileNetV2: Inverted Residuals and Linear Bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018.
[23] X. Chu , B. Zhang, R. Xu. "MoGA: searching beyond MobileNetV3," 2019.
[24] J. P. Wang, K. Gao, H. Z. Jiang, H. P. Zhou. "Pitaya detection method based on improved lightweight convolutional neural network," Transactions of the Chinese Society of Agricultural Engineering, vol. 396, no. 20, pp. 226-233, 2020.
[25] H. Liang, L. L. Jin, C. S. Yang. "Research on underwater target recognition based on deep learning in small samples," Journal of Wuhan University of Technology(Transportation Science & Engineering), vol. 43, no. 01, pp. 10-14, 2019.
[26] Y. Y. Liu, J. M. Zhang, K. P. Wang, X. H. Feng, X. H. Yang. "Fast underwater target recognition method under unbalanced data set," Computer engineering and application, vol. 960, no. 17, pp. 241-247, 2020.
[27] "Summary of target recognition algorithms," China Plant Engineering, vol. 412, no. 01, pp. 101-104, 2019.
[28] J. L. Guan, X. Zhi. "Mask wearing detection method based on YOLOv4 convolutional neural network".
[29] J. K. Paik, S. I. Su, C. H. Lee. "Histogram equalization method and device in contrast enhancement apparatus for image processing system: US," US6163621 A[P], 2000.