S2F-YOLO: An Optimized Object Detection Technique for Improving Fish Classification

Feng Wang[1], Jing Zheng[2], Jiawei Zeng[2], Xincong Zhong[3], Zhao Li[2]*

[1] School of Electronics and Information Engineering, Guangdong Ocean University, China
[2] School of Mathematics and Computer, Guangdong Ocean University, China
[3] Southern Marine Science and Engineering Guangdong Laboratory (Zhanjiang), China
wangfeng116@163.com, zhengjing@stu.gdou.edu.cn, cengjiawei@stu.gdou.edu.cn,
xczhong1997@gmail.com, zhaoli@gdou.edu.cn

Abstract

The current emergence of deep learning has enabled state-of-the-art approaches to achieve a major breakthrough in various fields such as object detection. However, the popular object detection algorithms like YOLOv3, YOLOv4 and YOLOv5 are computationally inefficient and need to consume a lot of computing resources. The experimental results on our fish datasets show that YOLOv5x has a great performance at accuracy which the best mean average precision (mAP) can reach 90.07% and YOLOv5s is conspicuous in recognition speed compared to other models.

In this paper, a lighter object detection model based on YOLOv5(Referred to as S2F-YOLO) is proposed to overcome these deficiencies. Under the premise of ensuring a small loss of accuracy, the object recognition speed is greatly accelerated. The S2F-YOLO is applied to commercial fish species detection and the other popular algorithms comparison, we obtained incredible results when the mAP is 2.24% lower than that of YOLOv5x, the FPS reaches 216M, which is nearly half faster than YOLOv5s. When compared with other detectors, our algorithm also shows better overall performance, which is more suitable for actual applications.

Keywords: Improved YOLOv5, ShuffleNetV2, Focal loss, Fish detection

1 Introduction

In recent years, with the continuous iteration and update of deep learning, it has played a significant role in our lives, promoting the development of artificial intelligence and various fields. As one of the branches of machine learning, deep learning has performed better recently. Although it has problems such as a large number of computing parameters, high hardware requirements, and dependence on a large number of datasets, it is good at learning and the number of neural network layers is deeper so that it can learn more features about the datasets. Otherwise, the upper limit of deep learning in all aspects is higher than that of traditional machine learning by adjusting the training parameters. Moreover, many lightweight neural networks have been proposed and hardware upgrades such as the Graphics Processing Unit (GPU) provides a multi-core parallel computing structure, which can efficiently handle a large number of matrix operations and is more friendly to image processing, that is why deep learning is growing so fast.

The emergence of deep learning has greatly promoted the development of computer vision applications. And as one of the branches of computer vision, object detection is the most important and challenging. In fact, we can feel that object detection has been widely used in various fields such as autonomous driving, face recognition, medical image, and so on [1-2]. At present, in the field of object detection, it can be divided into one-stage detector and two-stage detector according to the type of detector. The most representative two-stage detector is Faster R-CNN and the one-stage detector has SSD and YOLO [3-4]. The main difference between the two detectors is that the two-stage detector has higher localization and object recognition accuracy, while the one-stage detector has higher inference speed.

Fish detection is a meaningful project. It can not only promote the research progress of artificial intelligence on the sea, but also facilitate the progress of the fishery economy and is widely used in fishery processing factories, marine detection, fishery breeding, marine resources research, and so on, which is expected to improve the efficiency of commercial, marine departments, and researchers.

Therefore, we will combine the YOLOv5 and the more lightweight neural network to achieve a fish detection system with high precision and a faster recognition rate in this paper. Whether it is to improve the economic output of the aquaculture industry or protect the Marine ecosystem, it is of great significance to the subsequent fish research and the contribution to the ocean.

2 Related Work

In 2006, Hinton proposed Deep Learning which is composed of multiple hidden and perception layers [5]. The deep learning algorithm in the early 21st century did not perform so outstanding, due to limitations of data volume and computing performance at this time. With the continuous development of big data and High-Performance Computing (HPC), some applications based on deep learning, such as

*Corresponding Author: Zhao Li; E-mail: zhaoli@gdou.edu.cn
DOI: 10.53106/160792642023112406004

autonomous driving and biometric recognition have achieved great commercial success.

In the field of marine life recognition, Xiu Li et al. collected 24277 images of 12 fish species from ImageCLEF and applied Fast R-CNN for detecting fish in a complex underwater environment [6]. Their experiment obtained 81.4% MAP and found that Fast R-CNN has extremely high model inference and training speed, which is 80 times and 16 times of R-CNN respectively. After that, they improved their model with region proposal networks to share convolutional features, getting 82.7% MAP at last [7]. Shoaib Ahmed Siddiqui et al. proposed 152 layers deep CNN-SVM model with a special cross-layer pooling approach that combines marine bio characteristics from various convolution layers, in order to enhance discriminative efficiency [8]. Kewei Cai et al. optimized darknet-53 in the YOLOv3 model with MobileNet v1. This new backbone convolution kernel consists of depthwise and pointwise convolution and its classification accuracy within the 16 species classes is 79.61% [9]. Kazim Raza proposed an improved yolov3 model by supplying the 4th detection scale in the convolution network and used K-mean++ clustering to their dataset with 9-12 anchor boxes. Their algorithm achieved an average detection rate of 91.30 for four species [10]. Yongcan Yu et al. integrated the transformer module and YOLOv5s to solve marine object recognition in Side-Scan Sonar (SSS) image with 85.6% mAP in the laboratory [11]. Moreover, the fish recognition scenario in aquaculture can be harsh and require detection equipment that is stable, accurate, and easy to deploy. Factories prefer to choose embedded system devices such as jetson nano and raspberry PI for model deployment. YOLO model is one stage detection framework and has a faster inference speed, so it is more suitable for industrial production processes [12].

3 Methodology

This section includes an overview of YOLOv5 architecture and the collection of our dataset.

3.1 Dataset

The quality and quantity of datasets are significant for deep learning, ImageNet and Kaggle competitions have proved that deep learning algorithms require massive high-quality source data, which means the more data amount and higher annotation accuracy we get, the more accurate training models we will achieve [13].

A fish dataset is the basis of our classification research, and the fish detection task has a great demand for the image variation factors such as capture time, illumination intensity, angle, posture, and so on. Take cynoglossus nigropinnatus as an example (Figure 1), it shows different colors at different angles, so variation factors need to be taken into account to improve the generalization and accuracy of the training model. An iPhone 12 camera was utilized for acquiring images and videos, and our team use a variety of methods to enrich the dataset: complex background, different shooting times and complex spatial positions.

Figure 1. Front and back images of cynoglossus nigropinnatus

In this paper, we obtain shooting materials on the fishing market and fishing boats, using an iPhone camera to acquire images and videos of fish. Each fish is photographed three times in 24 hours at 8 hours intervals. We finally obtain 1341 images of 7 species including nemipterus bathybius, siganus fuscescens, sillago sihama, cynoglossus nigropinnatus, caranx kalla, terapon jarbua and scolopsis vosmeri in four different environments: laboratory, grassland, cement road, and sand beach. Rich scenes and different fish characteristics in the dataset can make the algorithm adapt to the complex and changeable environment, which is conducive to generating models that are more in line with the actual situation. Some of the images in the dataset are shown in Figure 2.

Table 1. The number of 7 fish species for training and testing images

Fish Species	Training images	Testing images	Total
Caranx_kalla	154	28	182
Cynoglossus_nigropinnatus	172	32	204
Nemipterus_bathybius	173	32	205
Scolopsis_vosmeri	168	32	200
Siganus_fuscescens	136	25	161
Sillago_sihama	149	28	177
Terapon_jarbua	179	33	212
All	1131	210	1341

Additionally, we have also supplemented some of the fish images from the international fish dataset to further enrich ours. Some giant fishing companies or related government agencies often main high-volume fish databases, but access to this data could raise issues of data copyright, commercial confidentiality and even national security. Therefore, considering security and cost, our team decide to use an international open-source fish dataset.

Fishbase is an open-source fish database created and maintained by the Leibniz Institute of Oceanology, which provides researchers with comprehensive data on species, regional distribution, and population density. So far, it is one of the world's most important public fish databases and has collected around 61,000 pictures of the world's fish [14].

Fish4K was originally developed by Bob Fisher at the University of Edinburgh. It acquired a large number of marine life images through underwater cameras and

developed real-time query software so that international researchers can easily use it at any time [15].

LifeCLEF series of Marine biological dataset is jointly produced by Concetto Spampinato from The University of Catania and Bastian Boom from the University of Edinburgh. Most of its fish images are located in LifeCLEF2014 and LifeCLEF2015. The Underwater video dataset in version 2014 comes from Fish4K, then version 2015 builds on 2014 with new images and manual annotations. In LifeCLEF2016 and LifeCLEF2017, they created a project called SeaCLEF which involves salmon and coral reef areas fish data [16].

Figure 2. Seven species of fish in the fish database

3.2 YOLOv5 Model

3.2.1 YOLOv5 Framework

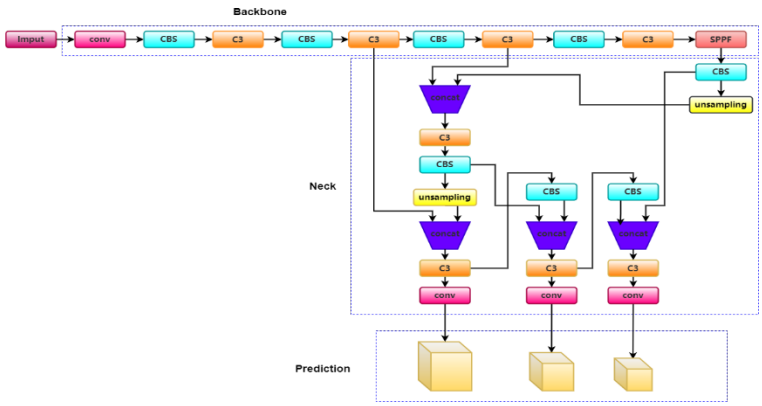

Figure 3. The framework of YOLOv5

Figure 3 is the overall structure of the model based on YOLOv5s, the functions of each main module are shown as follows:

A. Backbone consists of Conv structure and CSP structure. Conv structure can extract features, it is easier to export the model than the focus structure of YOLOv5. Cross Stage Partial (CSP) structure can effectively reduce the difficulty of calculation and memory cost, it is applied to YOLOv4 and have a great result [17]. But the C3 [18] block (as shown in Figure 4) is used in the backbone and neck sections to improve its architecture, such a structure maps the output from the shallow layer to the deep layer, which can

solve the problem of gradient disappearance very well and improve the efficiency of the calculation [19].

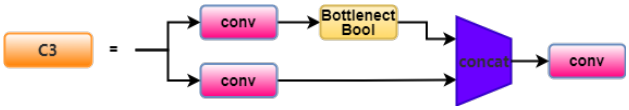

Figure 4. The C3 module of YOLOv5

B. In the neck module of YOLOv5, we solve the problem of inconsistent input image size by using Spatial Pyramid Pooling (SPP/SPPF) module with a block pooling layer [20], the composition of SPPF is shown in Figure 5.

The Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) are applied to the neck module for transferring the semantic and location feature respectively with deeper features aggregation [21-22].

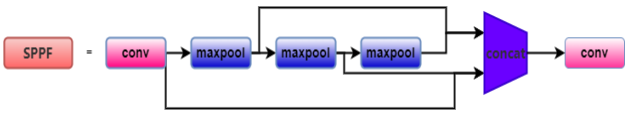

Figure 5. Composition of SPPF

C. In the head part, GIOU Loss [23] is used as the loss function of Bounding box regression, and DIOU is added to enhance the ability to process overlapping targets [24]. The GIOU loss is shown as Formula (1), (2), and (3), where A and B are the two detection boxes, and C is the smallest box capable of containing A and B.

$$IoU = \frac{|A \cap B|}{|A \cup B|}. \tag{1}$$

$$GIou = IoU - \frac{|C \setminus (A \cap B)|}{|C|}. \tag{2}$$

$$L_{GIoU} = 1 - GIoU. \tag{3}$$

3.2.2 Comparison of Four Versions of the Framework

The four versions of YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x have similar structures, the differences belong these four versions are the width and depth of each network, as shown in the following Table 2. All of the models adopt the size of 640*640 as the network input, and width and depth are two parameters that can control the parameters and GFLOPs of the network. With the increase of width and depth, the parameters and GFLOPs of YOLOv5's four versions are also growing, and their accuracy will be promoted, but their speed will decrease significantly. In contrast, S2F-YOLO, although its width and width are not pretty small, uses the ShuffleNet block structure in the network, which greatly reduces its floating point number calculation, making it significantly faster.

The above differences also affect the application scenarios of fish recognition, and appropriate models can be used according to different needs. For example, in the case of fish research, the pursuit of precision, or the production scene supported by high-speed configuration equipment, the model with deep network, high precision, and better fitting can be given priority, and the impact of speed is diluted. The application scenario for actual production purposes requires a delicate balance between accuracy and speed. Under the condition of ensuring accuracy, a faster model can be selected.

Table 2. Comparison of four YOLOv5 Versions

	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x	S2F-YOLO
Size	640	640	640	640	640
Width	0.5	0.75	1.0	1.25	1.0
Depth	0.33	0.67	1.0	1.33	0.66
Parameters	7311731	21570675	47506419	88574963	6307244
GFLOPs	17.0	51.6	116.3	220.4	9.9

3.3 Proposed Method

In this section, based on the comparison and analysis of the comprehensive performance of various versions of YOLOv5, we notice that among several popular versions of YOLOv5, YOLOv5s has a great advantage in terms of speed, but the accuracy is often inferior to that of YOLOv5x. However, the GFLOPs of YOLOv5x is pretty large, which greatly reduces the speed. In addition, in the actual training and detection tasks, due to the specificity and similarity of the categories, these models would show the phenomenon of sample imbalance, which inspired us to propose a new detector model named S2F-YOLO.

In the S2F-YOLO, the main network of CSPDarknet is replaced by the lightweight ShuffleNet V2 network, which can greatly reduce the calculation parameters, lessen the operation of floating point numbers [25], and better optimize the speed performance of Yolov5. Moreover, S2F-YOLO is combined with YOLO's C3 module and its SPPF (as seen in Figure 6), inheriting its advantages of better feature space extraction performance, improving the receptive field and reducing the accuracy loss caused by changing the network to a certain extent in the detection task. In addition, in order to improve the classification imbalance, the DIOU loss of YOLOv5 model was replaced by Focal loss. This loss

function could well improve the equilibrium and stability of sample identification, promote smooth training and improve the actual detection effect.

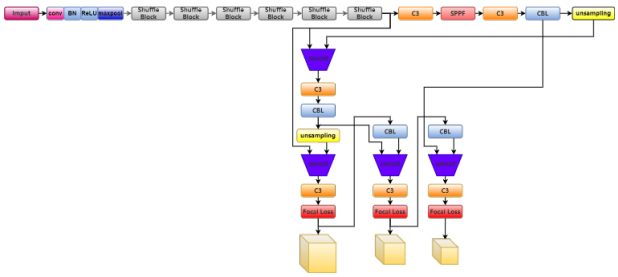

Figure 6. The improved network

3.3.1 ShuffleNet V2

The main purpose of using the Focus layer in the YOLOv5 structure is to reduce the number of parameters and computation, such as reducing FLOPs [26] (FLOPs, floating point operations), which represents the number of multiply-adds, while ensuring the under-sampling.

However, ShuffleNetV2 proposed that it is inadequate to use an indirect metric like FLOPs to calculate computation complexity. FLOPs is not an accurate enough estimation of actual runtime because the FLOPs metric only accounts for the convolution part. Although this part consumes the most time, the other operations including data I/O, data shuffle and element-wise operations (AddTensor, ReLU, etc.) also occupy a considerable amount of time. Therefore, shuffleNetV2 proposed four network design principles [25, 27] for a direct metric like Memory Access Cost (MAC) that can greatly increase the detection speed, and its network block is described in Figure 7. Based on these four principles, ShuffleNet V2 is a more efficient and lightweight architecture. What we do first is to change the Backbone part of YOLOv5, replace the original CSPDarknet53 module with ShuffleNet V2, and avoid image distortion with SPPF. After the convolution operation, the concat operation is added to realize feature fusion.

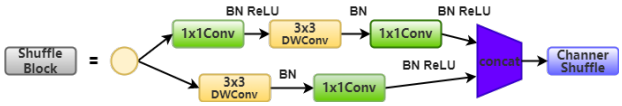

Figure 7. ShuffleNet Block

3.3.2 Focal Loss

In one-stage object detection, when the number of negative samples is large, its loss mistakenly classified as positive samples will account for the majority of the total loss, which is called class imbalance. To overcome the side effect of class imbalance, Tsung Yi Lin et al. come up with the Focal loss function to effectively improve the robustness and accuracy of the model without reducing the speed as much as possible.

Focal loss can make the model focus more on the samples that are difficult to classify by reducing the weight of the categorizable samples. In Formula (4), p∈[0,1] means the probability that the predicted label is the same as the true label, at the same time y = 1. Formula (5) is the cross entropy with a weighting factor for addressing class imbalance. The method is to use a small value α to reduce the weight of negative samples, but this method ignores the weights that are easy to classify and difficult to classify [28].

Focal loss solved this problem. It is defined as Formula (6), where p_t and FL(p_t) are the exported values for the probability of an event and focal loss, respectively. And γ∈[0,5] is a focusing parameter, and (1-p_t)^γ is a modulating factor. When γ gets an appropriate value, not only the loss of easy example is several times lower than the loss of cross-entropy, but also reduces the loss of hard example. This method effectively improves accuracy without losing speed.

$$p_t = \begin{cases} p, & if\ y = 1 \\ 1 - p, & otherwise \end{cases}. \tag{4}$$

$$CE(p_t) = -\alpha_t \log(p_t). \tag{5}$$

$$FL(p_t) = -\alpha(1 - p_t)^{\gamma} \log(p_t). \tag{6}$$

4 Experiment and Analysis

In the experiment, we use four versions of YOLOv5 and the improved model to compare their behaviors in our dataset. In addition, in order to draw a more scientific conclusion, we carried out a horizontal comparison of the model, using the classic target detection algorithms Faster RCNN and SSD as the control group to assist in testing the effectiveness of our model. We continue to use some useful tricks in YOLOv4 such as mosaic data augmentation, and cosine annealing scheduler to apply to YOLOv5. The four kinds of networks have different structures that might differ in the best parameters, so we train them separately and find the best hyperparameters in our dataset. Then we compare our models depending on the accuracy and speed, make it reach the balance point, and finally apply the optimal model to actual production and work. The steps of our experiment are as follows:

STEP1: Split the dataset. Use 80% of fish images for training and 20% for testing (Table 1).

STEP2: Set up the parameters. We need to modify some profiles and parameters to correspond to the four versions so that we can train successfully.

STEP3: Test. Save the best training model file and use it to predict the test set.

STEP4: Repeat STEP 2 and 3, and adjust the hyperparameters to find the most suitable parameters for the model.

STEP5: Record all indicators and compare the results.

The average precision (AP), mean average precision(mAP), AP50, AP75, and AP50:95 are used as evaluation criteria for model performance measurement. AP and mAP are defined as Formula (7), and (8):

$$AP = \frac{TP}{FP + TP}. \tag{7}$$

$$mAP = \frac{\Sigma_i^N AP_i}{N}. \quad \textbf{(8)}$$

There are four parameters for us to calculate the above criteria: false positive (FP), true positive (TP), i and N. FP is the incorrect detection with the positive sample and TP is the correct detection with the positive sample. In mAP calculation, N is the total number of fish classes being evaluated and AP_i is the value when AP is in the i-th class. Additionally, we used AP50, AP75 and AP50:5:95 to better measure model performance. Intersection Over Union (IOU) is the overlap of the different detection anchor regions divided by the total area of detection anchor regions. When IOU is equal to 0.5 and 0.75, we define the AP value as AP50 and AP75. AP@50:5:95 is the value when the IOU increases from 0.5 to 0.95 with a step size of 0.05.

4.1 Transfer Learning

Features extraction is one of the most crucial steps for classification. In deep learning project, most researchers prefer to apply transfer learning that use a pre-trained network weight on a large dataset as their initial weight to improve model training efficiency because there are potential connections among data lower-level features such as contour, grayscale and curves [29]. In this paper, we applied official pre-trained weights of YOLOv3, YOLOv4 and YOLOv5 to our training process and fine-tuned the model with freezing a part of layers, for the purpose of shortening the time to search for optimal hyperparameters, speed up training, and make the model converge faster.

4.2 Improved Mosaic

The mosaic [30] is based on the CutMix [31] data augment. It can randomly mix 4 training images while CutMix mixes only 2, which makes it possible to detect objects outside their normal context [26]. Compared with the original mosaic, we blurred the background of our images as Figure 8. This method can highlight the local features of a detected object, and let us easier to choose the training mini-batch size, but it may lead to model generalization descent.

Figure 8. Application of mosaic data augmentation in fish detection

4.3 Training Detail

The coding of our classification task is mainly performed by the open-source python library including PyTorch, OpenCV, etc. All the training and testing experiments were performed on a Windows 10 operating system with four Nvidia GeForce TESLA T4 and two Intel Xeon 5218R.

At the beginning of the experiment, we used open-source software (Label Me) [32] to label the dataset containing 7 kinds of fish and split them in a ratio of 8:2 for training and testing. In the initial stage of training, we use the same data set to train each model synchronously with the batch-size of 16 and an image-size of 640×640 [33].

In addition, we attempted to make combinations of different parameters to get a model which is closer to the characteristics of the fish species. For instance, we applied the label smoothing to suppress the overfitting to further generalize our model, and finally found the optimal smoothing value is 0.002. Moreover, we explored whether using mosaic data augmentation can contribute to the production of the best model. The results showed that using mosaic (mosaic=1.0) is a good choice when other parameters are unchanged. Besides, in this experiment, for the sake of enriching data sets and enhancing data with multi-scale and angles, we also used multiple data augmentation methods, such as HSV augmentation, transparency augmentation, and so on [34-35]. These approaches also prompt us to arrive at more desirable results.

When we try to modify the network, only using ShuffleNet V2 combined with YOLOv5, using a similar training method to YOLO, after adjusting the hyperparameters, it was found that the model speed increased significantly, but the accuracy decreased more seriously, and after analysis, focal loss, C3, and SPPF were combined with it, so that under the premise of small speed loss, the model was guaranteed to have considerable speed.

4.4 Test Result

Table 3 and Table 4 show the detailed results of the experiment, and Table 5 summarizes the outcome metrics of the experiment. From the experimental data, we can spot that in this dataset, the performance of YOLOv3 is inferior to YOLOv4 overall, while YOLOv4 is comparable to YOLOv5s in accuracy, but the speed (FPS) is lower than the latter. The accuracies of the four versions of YOLOv5 gradually improve as the network deepens and widens, but the speeds are decreasing sequentially. Under the conditions of the features of YOLOv5, the S2F-YOLO proposed in this paper ("S2F-YOLO" in each table) overall performance is better, which accuracy overall exceeds YOLOv5s, although compares to YOLOv5x mAP50 is 2.24% worse and mAP75 loss is 3.93% worse, the speed is about 47.95% swifter than the fastest YOLOv5s in the four editions of YOLOv5. The overall performance of the method exceeds YOLOv5s, and not much difference compared to the other versions. From the metrics of each table, this constructed model has the advantage of being more stable and robust. In addition, we list the results of simply combining ShuffleNet and YOLOv5 without applying C3 and SPPF layers and comparing the metrics with the others, it can be found that the accuracy of the improved model is better than that of YOLOv5+ShuffleNet V2. Undoubtedly, only combining YOLOv5 and ShuffleNet V2 can definitely reap the advantage in speed, but only such an approach will cause the trouble of accuracy loss. The

improved structure is more suitable for the characteristics of this dataset, as the accuracy loss is not pretty large and the detection speed is significantly improved. It is more suitable for the complex dataset of the scenarios used in this paper and the actual rapid detection of scene requirements.

At the same time, we compare the YOLO series with other algorithms, (as shown in Table 3 to Table 5), using the classic one-stage algorithm Faster RCNN and two-stage algorithm SSD (backbone for MobileNet V2 network). The results show that the accuracy of Faster RCNN is relatively optimistic, of which the corresponding indicator map50 reaches 90.92% and the speed index FPS obtained after testing under the same machine is 2M. The speed of the SSD algorithm inference is 46M, its map50 is 82.61% and map75 is 58.79%. Overall, the accuracy of Faster RCNN is good, but the speed is not superior to other detectors from the given models, and the speed and accuracy of the SSD algorithm reach a trade-off. In the table, it can be found that the model in this paper has a relatively comprehensive performance among the three, which can better act on practical application scenarios.

5 Conclusion

In this paper, we propose an improved YOLOv5 algorithm and construct a fish dataset, replace the backbone network with ShuffleNet V2 of YOLOv5, and modify its Loss to Focal Loss, which achieves better results in both accuracy and speed. The structure of ShuffleNet V2 combined with SPPF improves the speed. It's evident that Focal Loss is effectively solving the problem of sample imbalance, which enables training and prediction to proceed smoothly and improves the accuracy of the model. The data augmentation methods enrich the fish dataset such as Mosaic, Mix-up, and so on. They improve the effect and generalization of the model to a certain extent. Although the traditional YOLOv5x has a greater advantage in accuracy, the proposed model could achieve a fast inference speed with the mAP50 losing about 2%. Its FPS reaches 216M, nearly 10 times faster than the YOLOv5x.

Compared with other detectors such as Faster RCNN and SSD shows that S2F-YOLO has a better combination of performance, which is comprehensive in speed and accuracy metrics.

The results verified that the proposed method is more suitable for real fish identification scenarios, which require better performance in both speed and accuracy, rather than pursuing unilateral best performance. Therefore, we use ShuffleNet V2 as the backbone of the new model, replacing the original network CSPDarkNet53 during the experiment. By comparison, ShuffleNet V2 network is lighter, reduces a lot of calculation of the model and ensures smaller precision loss.

6 Future Work

In the future, we will continue to optimize the algorithm and follow up on the latest components of YOLO, exploring the following contents: 1. Based on the research of improving the detection of similar target features by attention mechanism, in our experiment, we noticed that the samples of fish are highly similar. Therefore, whether the attention mechanism components suitable for small target detection can be applied to the detection experiment of similar targets. 2. Use the horizontal comparison of lightweight networks, such as ShuffleNet, MobileNet, Darknet, etc. to research and implement algorithms with better speed and accuracy. 3. The transformer component, which is very popular nowadays in the object detection field, is expected to combine other artifacts to improve accuracy. It is worth noting that underwater image recognition is a very challenging project, which can take an important step toward the deep sea for artificial intelligence and attracts a lot of experts and scholars to study. The fish identification algorithm in this paper will be tried to be applied in this field in the future, listing the differences between land and sea image data sets, and based on these differences, optimizing the algorithm to improve the generalization of the model and make it more suitable for more complex environments. Through the above research, we will further improve the performance of YOLO in detection tasks. Meanwhile, we will further verify the classification effect in various fields, such as fish detection, bird detection, plant detection, etc., and strive to obtain an excellent model with more universal applicability.

Table 3. The mAP50 of all models training results indicators

	Sillago_ sihama	Siganus_ fuscescens	Cynoglossus_ nigropinnatus	Caranx_ kalla	Nemipterus_ bathybius	Terapon_ jarbua	Scolopsis_ vosmeri	mAP
YOLOv3	95.06	77.84	98.77	84.49	85.77	76.07	87.25	86.47
YOLOv4	94.48	81.86	99.56	79.68	90.38	73.27	90.58	87.11
YOLOv5s	97.23	74.39	99.5	85.84	86.31	76.32	94.25	87.69
YOLOv5m	94.32	81.17	97.67	90	85.94	85.94	92.62	88.33
YOLOv5l	96.15	80.65	99.32	86.1	85.05	80.71	93.98	88.85
YOLOv5x	98.73	87.31	99.5	94.04	84.01	73.94	92.98	90.07
YOLOv5+ ShuffleNet	94.53	70.22	93.72	82.74	80.21	68.78	87.87	82.58
S2F-YOLO	95.7	82.96	98.28	85.81	87.13	74.3	90.64	87.83
SSD	73.59	81.73	95.78	65.52	84.66	83.61	93.37	82.61
Faster RCNN	84.42	71.32	100	96.56	97.60	90.51	96.03	90.92

Table 4. The mAP75 of all models training results indicator

	Sillago_ sihama	Siganus_ fuscescens	Cynoglossus_ nigropinnatus	Caranx_ kalla	Nemipterus_ bathybius	Terapon_ jarbua	Scolopsis_ vosmeri	mAP
YOLOv3	80.05	63.29	59.95	53.65	70.14	68.09	64.2	65.62
YOLOv4	88.23	67.24	60.09	46.92	74.31	67.93	69.16	67.7
YOLOv5s	85.24	63.26	63.94	58.22	70.67	66.1	66.07	67.5
YOLOv5m	84.66	68.62	59.24	53.63	75.64	68.19	68.18	67.75
YOLOv5l	84.06	66.51	54.207	52.53	52.53	70.06	67.97	67.22
YOLOv5x	88.4	81.9	66.79	68.4	72.75	72.75	69.12	73.47
YOLOv5+ ShuffleNet	86.26	56.94	49.28	49.45	77.08	62.03	60.93	63.14
S2F-YOLO	82.7	74.38	53.07	65.13	74.48	67.39	69.68	69.55
SSD	40.71	42.91	47.01	51.80	75.57	79.45	74.05	58.79
Faster RCNN	80.42	64.41	95.85	72.13	91.91	89.29	92.36	83.75

Table 5. The parameters comparison of all models

Model	Size	AP50	AP75	AP@50:95	FPS
YOLOv3	640	86.47	65.62	58.60	38M
YOLOv4	640	87.11	67.70	58.80	55M
YOLOv5s	640	87.69	67.50	60.37	146M
YOLOv5m	640	88.33	67.75	61.59	76M
YOLOv5l	640	88.85	67.22	62.62	40M
YOLOv5x	640	90.07	73.48	63.04	21M
YOLOv5+ShuffleNet	640	82.58	63.14	55.27	239M
S2F-YOLO	640	87.83	69.55	61.40	216M
SSD	300	82.61	58.79	0.493	46M
Faster RCNN	-	90.92	83.75	65.41	2M

Acknowledgment

This work was partially supported by the program for scientific research start-up funds of Guangdong Ocean University (Grant nos. R20079), and the National Natural Science Foundation of China (Grant nos. 62066040).

References

[1] N. Le, V. S. Rathour, K. Yamazaki, K. Luu, M. Savvides, Deep reinforcement learning in computer vision: a comprehensive survey, *Artificial Intelligence Review*, Vol. 55, No. 4, pp. 2733-2819, April, 2022.

[2] P. Wawage, Y. Deshpande, Real-Time Prediction of Car Driver's Emotions using Facial Expression with a Convolutional Neural Network-based Intelligent System, *International Journal of Performability Engineering*, Vol. 18, No. 11, pp. 791-797, November, 2022.

[3] S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, V. Pattabiraman, Comparative analysis of deep learning image detection algorithms, *Journal of Big Data*, Vol. 8, No. 1, Article No. 66, May, 2021.

[4] N. Kumar, A. Goel, Detection, Localization and Classification of Fetal Brain Abnormalities using YOLO v4 Architecture, *International Journal of Performability Engineering*, Vol. 18, No. 10, pp. 720-729, October, 2022.

[5] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature*, Vol. 521, No. 7553, pp. 436-444, May, 2015.

[6] X. Li, M. Shang, H. Qin, L. Chen, Fast accurate fish detection and recognition of underwater images with Fast R-CNN, *OCEANS 2015 - MTS/IEEE*, Washington, DC, 2015, pp. 1-5.

[7] X. Li, M. Shang, J. Hao, Z. Yang, Accelerating fish detection and recognition by sharing CNNs with objectness learning, *OCEANS*, Shanghai, China, 2016, pp. 1-5.

[8] S. A. Siddiqui, A. Salman, M. I. Malik, F. Shafait, A. Mian, M. R. Shortis, E. S. Harvey, Automatic fish species classification in underwater videos: Exploiting pre-trained deep neural network models to compensate for limited labelled data, *ICES Journal of Marine Science*, Vol. 75, No. 1, pp. 374-389, January/February, 2018.

[9] K. Cai, X. Miao, W. Wang, H. Pang, Y. Liu, J. Song, A modified YOLOv3 model for fish detection based on MobileNetv1 as backbone, *Aquacultural Engineering*, Vol. 91, Article No. 102117, November, 2020.

[10] K. Raza, S. Hong, Fast and accurate fish detection design with improved YOLO-v3 model and transfer learning, *International Journal of Advanced Computer Science and Applications*, Vol. 11, No. 2, pp. 7-16, January, 2020.

[11] Y. Yu, J. Zhao, Q. Gong, C. Huang, G. Zheng, J. Ma, Real-Time Underwater Maritime Object Detection in Side-Scan Sonar Images Based on Transformer-YOLOv5, *Remote Sensing*, Vol. 13, No. 18, Article No. 3555, September, 2021.

[12] S. Zhao, S. Zhang, J. Liu, H. Wang, J. Zhu, D. Li, R. Zhao, Application of machine learning in intelligent fish aquaculture: A review, *Aquaculture*, Vol. 540, Article No. 736724, July, 2021.

[13] C. Sun, A. Shrivastava, S. Singh, A. Gupta, Revisiting Unreasonable Effectiveness of Data in Deep Learning Era, *IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 843-852.

[14] C. Boettiger, D. Lang, P. Wainwright, rfishbase: exploring, manipulating and visualizing FishBase data from R, *Journal of Fish Biology*, Vol. 81, No. 6, pp. 2030-2039, November, 2012.

[15] R. Fisher, Y. Chen-Burger, D. Giordano, L. Hardman, F. Lin, *Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data*, Springer, 2016.

[16] A. Joly, H. Müller, H. Goëau, H. Glotin, C. Spampinato, A. Rauber, P. Bonnet, W. Vellinga, B. Fisher, Lifeclef: Multimedia life species identification, *EMR: Environmental Multimedia Retrieval*, Orlando, FL, USA, 2014, pp. 7-13.

[17] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh, I. H. Yeh, CSPNet: A New Backbone that can Enhance Learning Capability of CNN, *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPR)*, Seattle, WA, USA, 2020, pp. 1571-1580.

[18] H. Park, Y. Yoo, G. Seo, D. Han, S. Yun, N. Kwak, *C3: Concentrated-Comprehensive Convolution and its application to semantic segmentation*, July, 2019, https://arxiv.org/abs/1812.04920v3.

[19] Z. Li, Y. Chen, Yi Song, K. Lu, J. Shen, Effective Covering Array Generation Using an Improved Particle Swarm Optimization, *IEEE Transactions on Reliability*, Vol. 71, No. 1, pp. 284-294, March, 2022.

[20] K. He, X. Zhang, S. Ren, J. Sun, Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 37, No. 9, pp. 1904-1916, September, 2015.

[21] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature Pyramid Networks for Object Detection, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, 2017, pp. 936-944.

[22] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path Aggregation Network for Instance Segmentation, *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 2018, pp. 8759-8768.

[23] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression, *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 658-666.

[24] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, D. Ren, Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression, *The Thirty-Fourth AAAI Conference on Artificial Intelligence*, New York, NY, USA, 2020, pp. 12993-13000.

[25] N. Ma, X. Zhang, H. Zheng, J. Sun, ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design, *European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 122-138.

[26] X. Zhang, X. Zhou, M. Lin, J. Sun, ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices, *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, USA, 2018, pp. 6848-6856.

[27] R. Ramadhana, K. Saddami, K. Munadi, F. Arnia, On Reducing ShuffleNets' Block for Mobile-based Breast Cancer Detection Using Thermogram: Performance Evaluation, *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)*, Vol. 10, No. 4, pp. 891-901, December, 2022.

[28] T. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal Loss for Dense Object Detection, *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017, pp. 2999-3007.

[29] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, Q. He, A Comprehensive Survey on Transfer Learning, *Proceedings of the IEEE*, Vol. 109, No. 1, pp. 43-76, January, 2021.

[30] A. Bochkovskiy, C. Wang, H. M. Liao, *YOLOv4: Optimal Speed and Accuracy of Object Detection*, April, 2020, https://arxiv.org/abs/2004.10934.

[31] S. Yun, D. Han, S. Chun, S. J. Oh, Y. Yoo, J. Choe, CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features, *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), 2019, pp. 6022-6031.

[32] A. Torralba, B. C. Russell, J. Yuen, LabelMe: Online Image Annotation and Applications, *Proceedings of the IEEE*, Vol. 98, No. 8, pp. 1467-1484, August, 2010.

[33] K. M. Kahloot, P. Ekler, Algorithmic Splitting: A Method for Dataset Preparation, *IEEE Access*, Vol. 9, pp. 125229-125237, September, 2021.

[34] C. Shorten, T. M. Khoshgoftaar, A survey on Image Data Augmentation for Deep Learning, *Journal of Big Data*, Vol. 6, No. 1, pp. 1-48, July, 2019.

[35] E. M. Schliep, J. A. Hoeting, Data augmentation and parameter expansion for independent or spatially correlated ordinal data, *Computational Statistics & Data Analysis*, Vol. 90, pp. 1-14, October, 2015.

Biographies

Feng Wang majored in communication and information system and received the MS degree from Wuhan University. He is a lecturer in the department of electronic engineering, Guangdong Ocean University. His interest is centered on application of IoT, embedded system, and big data processing.

Jing Zheng was born in Guangdong, China. majoring in computer science and technology. His research focuses on image recognition.

Jiawei Zeng was born in Guangdong, China. majoring in computer science and technology. His research focuses on image recognition.

Xincong Zhong received the MS degree in electronics and electrical engineering from the University of Sheffield. He is currently a research assistant at China Southern Marine Science and Engineering Guangdong Laboratory (Zhanjiang). His research interests include computer vision, deep learning and underwater wireless communication.

Zhao Li received the MS and PhD degrees in computer science and technology from Wuhan University. He is a professor in Computer Science and Technology, Guangdong Ocean University. His research focuses on image recognition, big data processing, and software defects prediction.