Mathematical Biosciences and Engineering

MBE, 20(8): 15018–15043.
DOI: 10.3934/mbe.2023673
Received: 04 May 2023
Revised: 26 June 2023
Accepted: 27 June 2023
Published: 14 July 2023

http://www.aimspress.com/journal/MBE

Research article

Large kernel convolution YOLO for ship detection in surveillance video

Shuaiwen Sun and Zhijing Xu*

College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China

* **Correspondence:** Email: zjxu@shmtu.edu.cn.

Abstract: At present, ship detectors have many problems, such as too many hyperparameter, poor recognition accuracy and imprecise regression boundary. In this article, we designed a large kernel convolutional YOLO (Lk-YOLO) detection model based on Anchor free for one-stage ship detection. First, we discuss the introduction of large size convolution kernel in the residual module of the backbone network, so that the backbone network has a stronger feature extraction capability. Second, in order to solve the problem of conflict regression and classification fusion under the coupling of detection heads, we split the detection head into two branches, so that the detection head has better representation ability for different branches of the task and improves the accuracy of the model in regression tasks. Finally, in order to solve the problem of complex and computationally intensive anchor hyperparameter design of ship data sets, we use anchor free algorithm to predict ships. Moreover, the model adopts an improved sampling matching strategy for both positive and negative samples to expand the number of positive samples in GT (Ground Truth) while achieving high-quality sample data and reducing the imbalance between positive and negative samples caused by anchor. We used NVIDIA 1080Ti GPU as the experimental environment, and the results showed that the mAP@50 Reaching 97.7%, mAP@.5:.95 achieved 78.4%, achieving the best accuracy among all models. Therefore, the proposed method does not need to design the parameters of the anchor, and achieves better detection efficiency and robustness without hyperparameter input.

Keywords: Anchor-free; label assignment algorithm; positive and negative sample assignment; large size convolution kernel; multi-task feature conflict

1. Introduction

With the increase of international trade, maritime transportation has gradually developed into the

main transportation mode. Moreover, ships, as the only way of maritime transportation, have caused many problems. For example, port management, illegal ship smuggling, illegal fishing by ships, etc. In some narrow docks or waterways, in the process of dense navigation vessels, if the crew has improper operation, the vessels may block the waterway and block the sea traffic. Recently, there were ships blocking the Suez Canal, causing more than 400 ships to be blocked and stranded, resulting in hundreds of millions of dollars of economic losses. At the mouth of the coast, due to stretches, there are often vessels illegally bypassing supervision, such as fishing vessels fishing at sea during the closed season, vessel smuggling, illegal entry into the sea and so on. Therefore, the research of ship identification for ocean surveillance has important application value. The identification for coastline or marine surveillance can ensure an early warning effect on danger or behavior, and facilitate managers to deal with abnormal behavior and emergencies in a timely manner and complete decisions quickly.

Most of the previous studies on ship detection are divided into three categories: based on SAR images, remotely sensed images and visible light images. Additionally, most of the researchers have studied SAR images [1–4] and remote sensing images [5–7] taken by using satellites. However, the images do not satisfy our need for real-time feedback. Therefore, in this paper, we choose to use visible images [8,9] as the basis of research to achieve ship identification using knowledge related to object detection.

Nowadays, ship recognition has changed from traditional image algorithms, such as histogram of oriented gradient (HOG) detector [10], deformable parts model (DPM) [11] to methods in the field of object detection in computer vision [12], that is, convolutional neural network (CNN) is used to build a depth learning model to detect objects. Since 2012 [13], the convolutional neural network has replaced the invention of artificial feature construction. For example, VGG [14] builds the stack of multi-layer convolutions to achieve several times the performance of the artificial feature detector.

In recent years, the field of object detection is still developing rapidly, and a large number of object detectors with high performance have emerged. Object detection networks are also divided into two categories: one is the two-stage object detection network led by R-CNN [15] and the other is the one-stage object detection network represented by SSD [16] and YOLO [17]. In 2014, the birth of R-CNN officially made CNN the first choice for image classification and the first two-stage object detection algorithm. In 2015, Girshick raised the problem that Fast R-CNN [18] is cumbersome and complex to improve the network, and the training cost is high. Based on this, the Faster R-CNN [19], designed by Ren in 2016, proposed to first generate regional candidate frames and then identify and classify from different candidate frames, which became the mainstream detection method for two-stage detection networks.

The best known of the one-stage detectors is the YOLO (you only look once) proposed by J. Redmon et al. It also includes the SSD detection model proposed by W. Liu et al. and RetinaNet [20] designed by Lin et al. for solving the problem of positive and negative sample imbalance in single-stage networks. Lin et al. proposed the Feature Pyramid (FPN) [21] structure to extract feature information at different scales. Subsequently, researchers designed different pyramidal structures, including PaNet [22], NAS-FPN [23] and Bi-FPN [24] in order to better fuse feature information of different sizes.

In addition, researchers have also worked on backbone networks, proposing not only the residual structures ResNet [25] and ResNXet [26] that can correlate pre- and post-textual information, but also some Anchor-free single-stage Dense Box [27], CornerNet [28] CenterNet [29], etc. that do not use anchor boxes detectors. In recent years, Anchor-free detectors have received increasing attention.

Compared to Anchor-base, the Anchor-free network does not require hyperparameters about the anchor, and there are fewer parameters and computational costs. It is the basis for implementing end-to-end recognition algorithms. Today, in ship detection, people still use Faster R-CNN, SSD, YOLOv2 [30], YOLOv3 [31] and other networks that use anchor for regression, and there are no applications of Anchor-free networks yet. However, ship data generally present the problems of uneven data distribution, unreasonable Anchor size setting leading to poor overall recognition accuracy and poor network real-time performance, and the advantages of Anchor-free network can precisely solve these problems.

Therefore, this paper combines the research background and characteristics of ship detection, and proposes a deep network model for ship object recognition and classification improvement under video surveillance data, and uses data augmentation techniques to expand data samples and enhance the feature extraction of the network structure, in view of the problems of excessive hyperparameters, sparse data samples and insufficient performance of the detection network in ship object detection. The work as well as the innovation points of this paper are as follows:

1) Based on the design idea of YOLO and Transformer architecture, we propose LK-YOLO. On the basis of YOLOv5, large kernel convolution is introduced into the backbone residual block, backbone with stronger feature extraction ability and optimized neck network structure details are designed.

2) We replace the original coupled detection head, which is common in one-stage networks, with a modified uncoupled detection head, so that the detection head has a stronger feature representation capability and solves the problem of task conflict under shared weights of classification and regression.

3) To solve the negative optimization problem of anchor clustering for ship dataset, we use anchor-free detection instead of the traditional method of anchor clustering using k-means algorithm to achieve hyperparameter-free input to the detector. Using the idea of central prior and dynamic sample screening, the sampling strategy for positive and negative samples under the ship dataset is optimized to enable better training of the model and also to achieve better performance.

2. Related work

In recent years, the fire of attention mechanisms has not only given birth to attention mechanism modules, such as SENet [32] and CBAM [33], but also to the design of large-scale complex networks that use attention mechanisms entirely, such as Transformers [34]. In the beginning, Transformers were first applied to the field of natural language processing. This was followed by Vision Transformers (ViT) [35] uses entirely attention modules, without convolution, to achieve object recognition. Due to the superiority of the architecture, Vit was better able to draw on more complex models and large-scale datasets, and ViT got better detection results in the field of recognition.

In ship detection, CNN networks in deep learning have been used for ship detection. Zhenfeng Shao published SeaShips [36], the first ship dataset captured using surveillance, in 2018, which includes a total of 7000 images from the open-source part of passing ships on Hengqin Island, Zhuhai, China. However, there are not many areas where surveillance video is used for ship recognition, and implementing ship recognition under surveillance has very significant challenges due to camera performance and viewing angles, complex environments and changing weather. Shao used YOLOv2 as a framework [37] and combined saliency detection to design a ship recognition network, using the Canny operator to detect image edges and then perform the Hough transform to get all line segments before filtering out coastlines, segmenting sea and land to reduce the number of parameters for network operations. However, in fact, such an approach is not particularly enhanced for deep learning-based

object detection, and the graph will be resized into a fixed-size image once. Li [38] proposed an improved YOLOv3 Tiny, which redefined the anchor frame, fine-tuned the network structure and introduced the attention mechanism module CBAM. Han proposed ShipYolo [39] to embed CBAM into the YOLOv4 backbone residual structure and proposed DSPP structure to amplify the sensory field. Jun-Hwa Kim [40] proposed YOLOv5 and Online Copy& Paste and Mix-up methods to solve the ship category imbalance problem. Chen [41] addresses the problem of low ship detection accuracy for small targets and proposes a ship detection algorithm process using improved GAN networks to generate image samples combined with YOLOv2 to achieve improved detection performance of the detector for small and medium-sized targets of ships. Although the application of deep learning models in ship detection brings great performance gains, mainstream ship detection relies on Anchor-based detectors. The detectors need to first cluster the ship samples using clustering algorithms, such as K-means, to generate anchor frame size parameters, which are input to the model as hyperparameters. The optimization of anchor frame size parameters has been a common problem in ship detection, so we propose to design anchor-free detection algorithms to avoid the design of anchor size hyperparameters.

Anchor-free object detectors are also becoming mainstream in the research of general-purpose 2D object detection. Tian proposed FCOS [42], which implements an anchor-free detector composed entirely of convolutions. The model uses regression of centroids to predict targets, and achieves multi-scale object detection by dividing the size range of objects into different feature layers. The article also explores the differences between the regression approaches on Anchor-free and Anchor-basesd and the degradation of the detector performance due to the reduction of the number of positive samples after the elimination of anchor. Zhang [43] studied FCOS and ReinaNet and found that the essential difference between Anchor-base and Anchor-free lies in the different ways of label assign, and proposed the ATSS approach to balance positive and negative samples. Ge proposed Optimal transport assignment [44] for global assignment of positive samples for target recognition, allowing the detector to learn the most effective sample features more fully. In summary, for anchorless frame detection algorithms, we do not only design the detection process of the detector without anchor, but also need to design more efficient label assignment algorithms to compensate for the performance loss of sample reduction from not using anchor.

The ship dataset has a complex background, high target overlap rate and many duplicate data under the problem, we need to make improvements to the object detection model for the ship dataset. We needed to redesign the model to obtain a backbone network with better feature extraction capability. We also decouple the detection head for different tasks, allowing the regression and classification tasks to be separated. Finally, the use of anchor-free is proposed instead of input parameters to achieve no output hyperparameters. Also, the positive sample sampling algorithm is optimised to allow the model to be better trained for convergence.

3. Materials and methods

3.1. Backbone

In terms of network architecture, we have analysed the dominant one-stage object detection network YOLO, as well as ViT, which has recently excelled in the field of object detection. We have redesigned the network architecture and optimised the parameter details of each layer of the residual structure. The network structure of Large kernel convolution YOLO is shown in Figure 1.

Backbone uses a combination of CSPDarkNet-53 and a multi-layer residual structure for feature extraction of the input data. The FPN structure was chosen to have semantic feature-rich layers C3, C4 and C5, where the C5 feature layer was then subjected to pooling operations by SPPF. The FPN is top-down and retains the semantic features of the upper layers well, but the positioning information is blurred in the transfer process, so the bottom-up design of PANet is used to make the position information of the bottom layer fused to the upper layer, and the fusion of the two structures is well done to complement the information and make the positioning of the model more accurate. Finally, the prediction results are output from three dimensions.

In this paper, the design of Backbone is improved in two main directions: the design of the structural proportions of the residual blocks on the architecture and the detailed design of the residual blocks.

On the one hand, YOLO networks are designed with different proportions of residual structures on different feature layers. In YOLOv3 or YOLOv4, the ratio of residual blocks is [1,2,8,8,4], i.e., the ratio of feature layers is 1:4:4:2. The contribution of different depth network features in the detector to the model performance is different. In the FPN structure, the C3 feature layer is able to effectively extract object detectors under different sizes, so multiple residual modules are set up in C3 to ensure that the layer has excellent extraction capability. We finally chose a ratio of 1:3:3:1 to set the ratio of residual blocks in the backbone to ensure that the network has good feature extraction capability at C3 size.

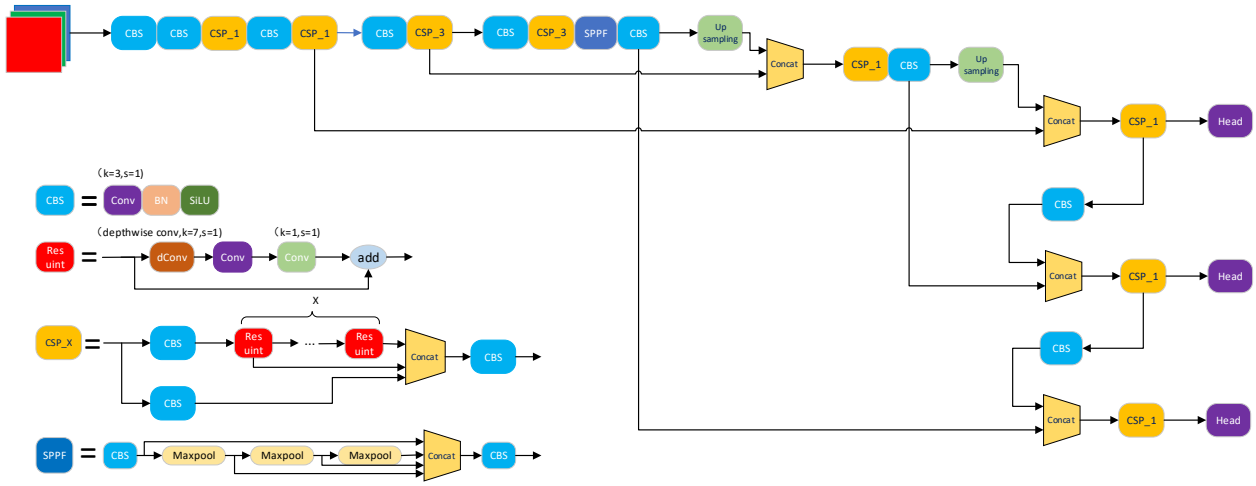


Figure 1. Large kernel convolution YOLO network.

On the other hand, from YOLOv3 to YOLOv5, the residual blocks of the backbone network went from DarkNet-53 to CSPDarkNet-53. The structure of each residual block was continuously deepened. Therefore, we modified the Res unit in the BottleneckCSP by introducing a large convolution module to replace the core Bottleneck module and implementing fine tuning on top of this module.

The network modifies the residual unit in BottleneckCSP by using the structure of a large 7 × 7 convolution kernel combined with Depth-wise convolution instead of the original 3 × 3 convolution, and fine-tunes the structure by changing the position of the convolution on top of the core of the module. The depth-wise convolution is used as a large kernel convolution to reduce the computational effort of the model since the large size of the convolution kernel inevitably leads to an increase in the number of convolutional computational parameters. The number of convolution kernels needs to be equal to the number of channels of the input, and the number of channels of the output is also constant and

equal to the number of channels of the input. This can effectively reduce the computation of multi-layer convolution by adjusting the size of the feature map without changing the number of channels when the structure is a large convolution kernel structure.

In Figure 2, we use a large kernel instead of the traditional 3 × 3 convolution kernel, and use an inverted triangular arrangement. In the past, the large kernel convolution structure was equated to multiple 3 × 3 convolution kernels due to its high computational effort and low performance. Depth-wise convolution is a good way to reduce the computation of convolution parameters, allowing the network to obtain a larger convolution depth with the same number of parameters, thus improving the network performance.

Due to the shortcomings of BatchNorm in min batch size, we use LayerNorm instead of BN to normalize the samples and reduce the use of regularization. LayerNorm stabilizes the data distribution by normalizing the dimension of Hidden size. Finally, the 1 × 1 convolution has the special property of dimensional adjustment, so we reduce the BN once after the 1 × 1 convolution to reduce the computational parameters.

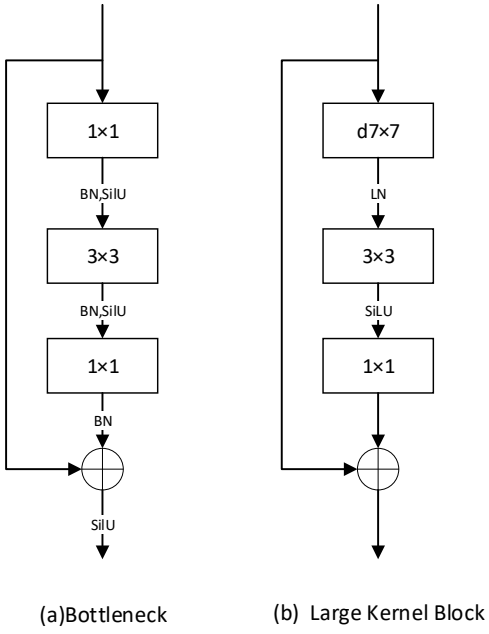


Figure 2. Residual unit structure.

3.2. Decoupled head

In object detection, the primary solution is object classification and regression. The classification task has to address the ability of the target to be correctly identified at different angles at different positions, and therefore needs to ensure translation and scale invariance. The regression task needs to ensure that the target position and shape are mapped onto the features, i.e., it requires translation and scale equivalence. Therefore, different tasks have different requirements for different features, with the classification task focusing on the main part of the target and the position regression task being more sensitive to the whole target and boundary issues. It is a challenge for the detection model to better integrate these two tasks and output the best results.

Two-stage networks solve the problem by reducing shared weights and differentiating predictions between classification and regression using different structures. However, there are few such solutions in one-stage networks. The detection head of a one-stage object detection network is usually designed as a simple convolution for the purpose of parameter reduction and simplicity. In YOLOv3 to YOLOv5, the feature maps output from multiple feature layers connect a structure with a combination of 1 × 1 and 3 × 3 to form the detection head of the network, generating a multidimensional tensor $H \times W \times anchor \times (Cls + Reg + Obj)$ containing regression information and classification, where anchor is the number of anchor boxes for each anchor pre-defined value in the detection network, Cls is the data the total number of sample categories, Reg is the 4 coordinate values of the point regression and Obj is the confidence level of each prediction frame. This approach has continued to be the dominant method for the first stage network.

The proposed decoupled detection head in this paper is shown in Figure 3. The detection head connects the three-layer feature output after the fusion of FPN and PANet, and each layer of features is first subjected to 3 × 3 ordinary convolution before accessing the parallel structure, undergoing 3 × 3 DConv convolution and finally adjusting the number of channels through one 1 × 1 convolution to output the result of branching. The detection head takes a similar structure to the YOLO head, with a transition from large convolution to small convolution. Here, the detection head needs another convolution on a different branch, so that the result of the extracted features better fits the task of that branch. We still choose Depth-wise Conv to simplify the parameters of the detection head and finish the output with a 1 × 1 convolution.

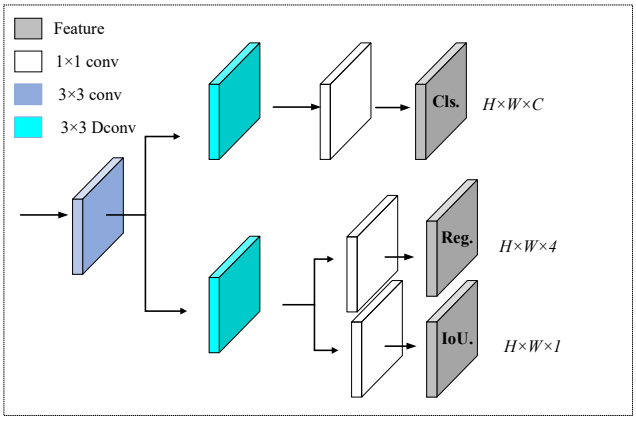


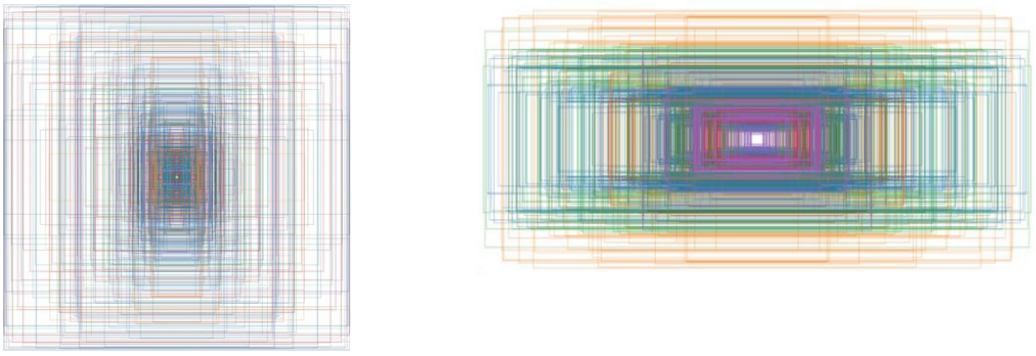
Figure 3. Decoupled head.

In addition, we include the Intersection of Union (IoU) branch to represent the regression confidence, which can be seen as the size of the IoU of the predicted output box versus the true box. In this paper, we choose to parallelize this branch with the regression branch, so that the two tasks share the weights to obtain the regression confidence Conf_IoU. Also, since there is a range of confidence, i.e., Conf_IoU $\in$ [0,1], the Sigmod activation function is plugged in after the output so that the output must be within the range. There are two reasons for this: first, the regression task and the IoU prediction values are somewhat correlated and there is feasibility and interpretability; second, the parameter calculation can be effectively reduced at the same level of branching, and the three-branch design is too redundant to avoid excessive computational cost wastage. At the same time, the parameter Conf_IoU has several practical meanings: first, it represents the judgment whether the target

box is foreground or background, i.e., the probability size of the box containing the target. Second, the size of the IoU of the predicted output box versus the real box when a target is present in the current box. This parameter can also be used as an evaluation parameter for the NMS to filter the boxes during network inference.

3.3. Anchor-free

Most of the algorithms in the field of ship detection have been migrated from anchor-base in Object detection. Anchor-base models are further trained and predicted based on the set of anchors, which are usually obtained by clustering the targets in the dataset using K-means or an improved K-means algorithm. However, there is a difference in the datasets to which the two apply. In the field of object detection, studies usually use the COCO dataset or the VOC dataset. As shown in Figure 4(a), the objects clustered in these datasets are filtered to obtain a uniform distribution of the number of anchors of three sizes, large, small and with a variety of aspect ratios. In the case of the ship data, most of the clustered anchors are shown in Figure 4(b). Anchor mainly appears as the shape feature of the ship, with long lengths and not high widths. Unlike the COCO dataset, there is also inconsistency in the sample data for the different scales of the ship dataset, with a smaller number of smaller ships. Anchor frames predicted on this basis can have mismatched sizes or similar sized objects being split into different scale layers for prediction. Therefore, for the ship dataset, resetting the parameters of the anchor would exist for the model to perform a negative optimisation calculation.



(a) COCO anchor (b) SeaShips anchor

Figure 4. Anchor boxes clustering results under different datasets.

At the same time, the model generates very dense prediction results on the images in order to achieve higher recall and accuracy. The huge number of anchor frames causes the network to compute very large amounts of anchor frames as well, because each prediction frame has to match the IoU between GT computation to select the positive and negative samples of the network. When the FPN structure is introduced into the network, the design of the feature layer allows the parameters to grow multifold. For example, under the COCO dataset, the target has 80 categories and assuming 3 anchor frames are preset at each centroid, with an input image of 416 × 416, for example, based on the number of each anchor frame, the center offsets x, y, w, h, confidence level and the number of categories, it is obtained from Eq (1) that 904,995 predictions need to be computed. Therefore, when YOLO becomes

Anchor-free algorithm, the number of anchor frames is assumed to be N. Then, the number of parameters becomes 1/N of the previous one, and only 301,665 results are needed.

The generic formula for the number of parameters predicted under the Anchor-base model with a minimum feature map of S×S is:

$$Parameters = N \times (S \times S + 2 \times S \times 2 \times S + 4 \times S \times 4 \times S) \times Nums_{class} \quad (1)$$

And the number of predicted resultant covariates under the Anchor-free is:

$$Parameters = (S * S + 2 * S * 2 * S + 4 * S * 4 * S) * Nums_{class} \quad (2)$$

When the anchor free network outputs the feature map from the detection head, it will traverse every pixel on the feature map and perform predictive decoding. According to the definition of feature maps, the pixels on the feature map after multi-layer convolution can be equivalent to a grid area divided on the size of the original image. The traversed pixels can be seen as generating a central anchor point on the grid, with each anchor point corresponding to a pixel on the feature map. Traverse each anchor point for decoding and prediction, that is, generate a prediction box at the central point in each region.

Due to the model abandoning the use of Anchor Boxes, each anchor point no longer generates multi-scale prediction boxes, but only generates a prediction target matching GT. And the feature points shift from the original prediction anchor box to the GT offset, transforming into generating prediction boxes directly calculating deviations from the four coordinate points of GT. The method of predicting regression changes from calculating the four offset values between the generated anchor box and the real box to directly predicting the biaxial offset of the center point and the width and height of the target, namely Top, Bottom, Left and Right. Finally, the result of the prediction box is decoded and mapped to the position of the original image size by multiplying the sampling step size of the current feature map.

Taking Figure 5 as an example, the red point in the GT center of the ship falls in the green area to predict the offset value of ltbr (left, top, bottom, right). By adding the offset values in four directions to the center coordinates (x, y) of the area, four coordinate points of the prediction box are obtained to generate a blue prediction box.

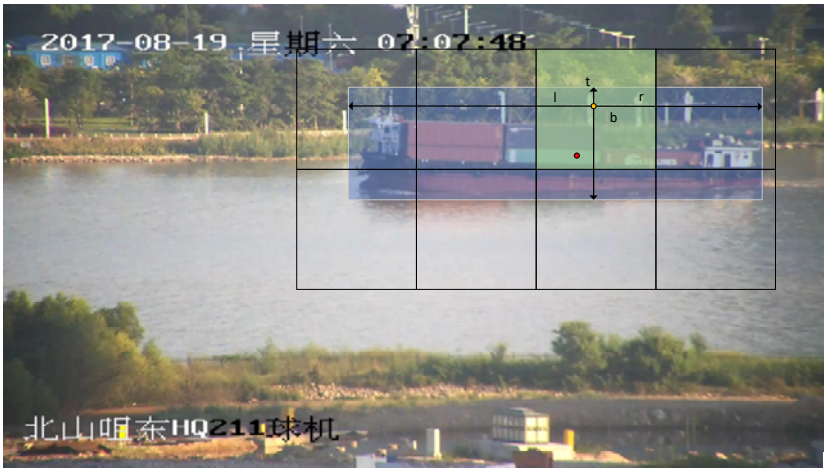


Figure 5. Network regression method.

Among the current mainstream object detection networks, the YOLO series and other object detection networks are based on the regression of anchors to achieve the object calibration. Take the YOLO as an example, the network will first divide the images into different grids (gird cells), and each gird cell is assigned multiple sizes of anchors to expand the positive samples, using the anchors to constrain the regression task, each regression its only responsible for one or more objects near the corresponding gird cell. The IoU of the anchor and the GT (Ground Truth) are derived one at a time using the anchor, and the anchor with the largest IoU is responsible for predicting that GT.

When the network model discards the anchor, the positive and negative sample screening strategy becomes a bottleneck in network performance. Each gird will only generate a single prediction sample when it loses its multi-size anchor. Therefore, we need to redesign the allocation of positive samples in a way that counteracts the impact of this problem. This can be broadly divided into three parts: positive and negative sample definition, positive and negative sample sampling and balancing loss. in the Anchor-free algorithm, we not only used the strategy of range filtering positive samples as above, but also designed a dynamic label selection algorithm.

The anchor-free strategy devised in this paper, again divides the image equally into different grids, generating a centroid at each grid. At the same time, the feature points at each location are mapped to the original image, and the number of targets predicted by each gird cell is changed from the number of anchor boxes to one object and directly predicted: the two-axis offset of the centroid, the width and height of the target and the IoU size with respect to the GT. Finally, as in Figure 6, the positive and negative samples are determined based on whether the centroid of the GT falls on the feature point to the centre of the GT.

We extend the range of positive samples so that GT and the two adjacent frames are positive sample areas. In other words, prediction frames whose centroids fall within the positive sample area are set as positive samples. Positive samples are also predefined in a scale range so that different size FPN layers are responsible for matching scale size samples.

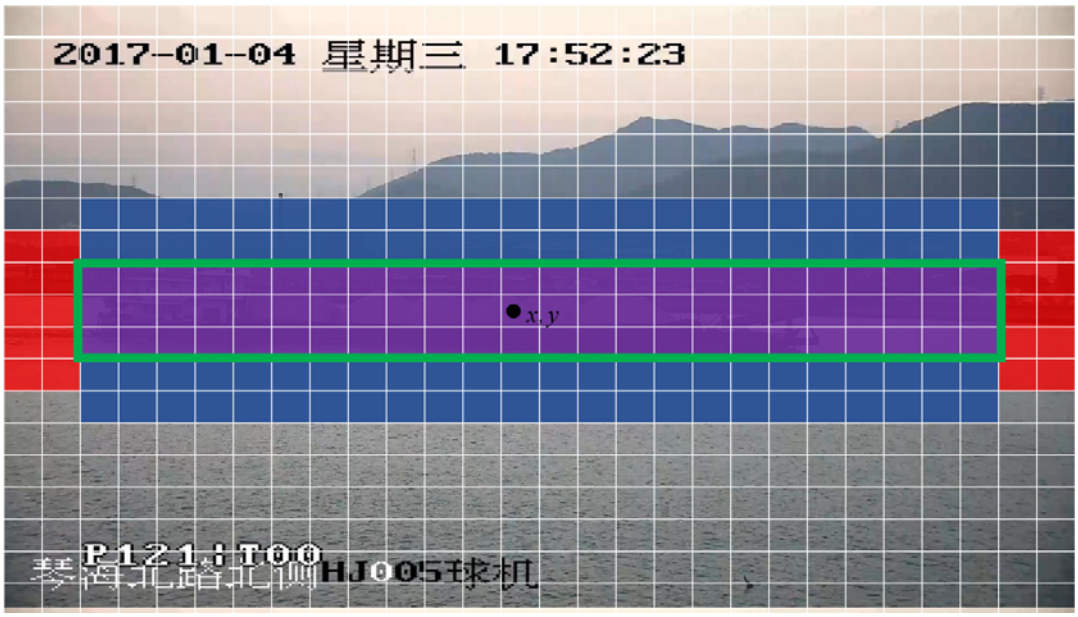


Figure 6. Positive sample screening region.

After that, we need to filter each positive sample again. The model no longer ranks anchor boxes by a single choice of calculating classification scores or positioning scores. The bounding boxes were sorted according to the classification confidence, and the box with the highest classification score was selected each time, causing a large proportion of the bounding boxes with accurate localization to be incorrectly suppressed. This is caused by the mismatch between classification confidence and localization accuracy. Therefore, we propose a new calculation Eq (3) that uses the weighted scores of classification and regression to measure the combined quality of the prediction result, and combine the two tasks to design a new evaluation index.

$$t = s^{\alpha} \times u^{\beta} \tag{3}$$

where s and u represent the classification score and the IoU confidence level, respectively, and α and β represent the weight coefficients of the two tasks to control the impressions of both tasks on the measures. When the sample has bias in one of the tasks, it is beneficial to control the task alignment and better assess the quality of the sample.

When a large number of prediction boxes appear for a target of the image, we filter the prediction boxes whose IoU between the prediction object and the GT is less than 0.3, consider the remaining boxes as higher quality predictions and sort all the remaining prediction boxes by IoU value from largest to smallest. The top 15 IoU are then summed and rounded, and the result is set to k. At the same time, the loss of each frame with GT is calculated, and the k prediction frames with the highest scores are selected using the weighted summed Score as the basis for selection. Since there may be the same prediction frame with both GTs having high IoU values. Faced with such ambiguous samples, we assign the edge frame to the highest scoring GT, while the other target is then re-screened from the corresponding candidate frames.

3.4. Loss function

The loss function is an important optimization criterion for model training and an important metric for model evaluation. The loss function of our proposed model consists of the three elements: regression loss, classification loss and IoU confidence loss. Regression loss predicts the centroid and the offset result of width and height, classification loss outputs the probability of each object category, and IoU branch outputs the ratio of the predicted frame to the real frame IoU.

In the regression, we use CIoU Loss as the regression loss. CIoU is to add the loss of detection frame scale to DIoU, adding the loss of length and width. solves the problem that DIoU loss cannot distinguish which region is more similar to GT when multiple centroids overlap. three items of CIoU correspond exactly to IOU, centroid distance, aspect ratio of are calculated so that the prediction frame will be more consistent with the GT.

$$CIoU = 1 - IoU + \frac{\rho^2(\mathrm{b},\mathrm{b}^{gt})}{c^2} + \alpha v \tag{4}$$

where b and b^{gt} denote the centroids of the prediction frame and GT respectively, $\rho^2(x, y)$ denotes the Euclidean distance between the two points and c denotes the diagonal distance of the smallest closed region that can contain both the prediction frame and the true frame.

$$\alpha = \frac{v}{(1-IoU)+v} \tag{5}$$

$$v = \frac{4}{\pi}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h}) \tag{6}$$

α and v are calculated as follows, with the former being the weighting factor (giving higher priority to the overlapping area factor in the non-overlapping case), while v measures the similarity of the aspect ratio.

In the above, we mentioned that the IoU confidence level ranges between 0 and 1. The same applies to the classification loss output. Therefore, the IoU loss function was chosen to be calculated by BCEWithLogitsLoss. In contrast to the commonly used Cross Entropy (CE), dichotomous classification is suitable for outputting data between 0 and 1, whereas CE Loss is often used for multi-classification to output probability values for n channels. BCEWithLogitsLoss also differs from BCELoss in that when performing the Loss calculation, the Sigmoid function is first used on the data to transform the values to between [0,1] before performing the BCELoss calculation.

$$BCE_{WithLogits} = \sigma\left(-[y_n \cdot log(x_n) + (1 - y_n) \cdot log(1 - x_n)]\right) \tag{7}$$

4. Experimental results and analysis

The experimental environment is under Ubuntu 18.04, with machine CPU I7-7700x, graphics GPU GTX 1080Ti, running memory 16 G, CUDA 10.1 and Cudnn 7.6.5 and Pytorch 1.2.0 was chosen as the deep learning framework. In the training process, we need to set the hyperparameters of the model, as detailed in Table 1. We set 300 training Epoch in order to ensure that the model learns sufficiently from a small sample of ships, and the first 100 cycles were freeze training to train only some of the network parameters to shorten the training time. Optimisation was performed using the Adma optimiser, with the weight decay value taken to be 5e-4 and the learning rate 1e-3. A fixed step decay (StepLR) with a gamma value of 0.95 was used as the learning rate decay strategy.

Table 1. Hyperparameters setting.

Hyperparameters	Value
Epoch	300
Optimizer	Adma
Learning rate	0.001
Learning rate decay strategy	StepLR
Weight decay	5e-4
Batch size	8

4.1. Dataset

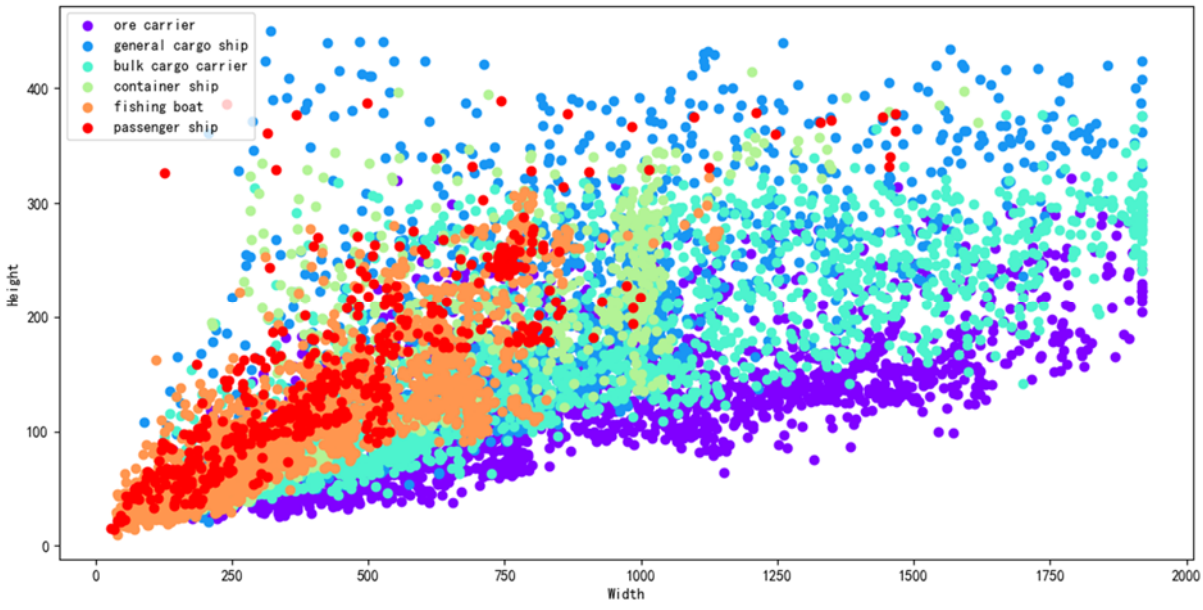
In this paper, the open source Seaships dataset from Wuhan University is used. The dataset consists of near-shore ship monitoring images taken from Hengqin Island, Zhuhai, with a total of 7000 images of 1920 × 1080. As shown in Figure 7, the dataset has six types of ships with different functions, including ore carrier, general cargo carrier, bulk cargo carrier, container ship, fishing ship and passenger ship. In the dataset, the images have several problems such as a large amount of background interference, overlapping ships, general cargo carrier and bulk cargo carrier are very similar, and small objects of fishing boats are difficult to recognize in distant images.

(a) ore carrier (b) general cargo carrier (c) bulk cargo carrier

(d) container ship (e) fishing ship (f) passenger ship

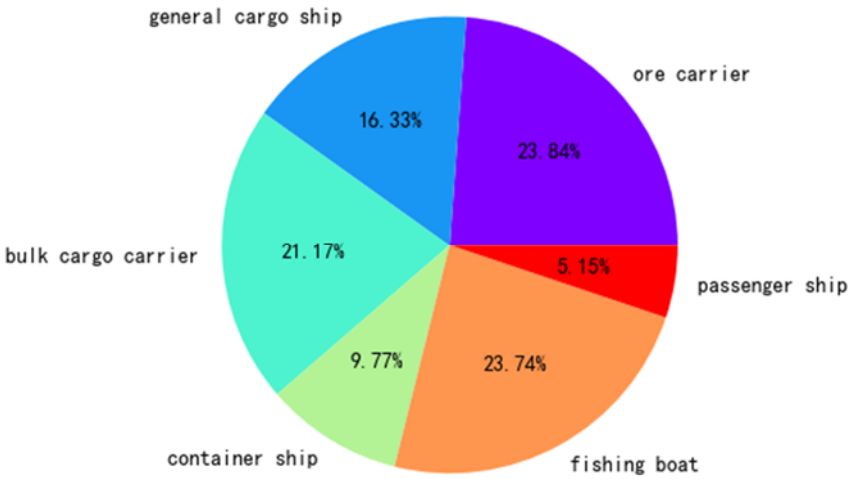
Figure 7. Different classes of ships in the dataset.

To visualise the nature of the data and understand the distribution, Figure 8(a) plots the distribution of the real frame dimensions of the dataset-ship sample using a scatter plot, with the horizontal and vertical coordinates representing the length and width of the ground truth of the ship. As well as Figure 8(b) visualises the percentage of the number of vessels in the sample using a pie chart.



(a)

Continued on next page



(b)

Figure 8. (a) Scatter diagram of ship size distribution, (b) Pie chart of different types of ships.

The experiment divides the dataset into a training set, a test set and a validation set in the ratio of 6.5:3:0.5, allowing us to verify the robustness and accuracy of the model. This paper also expands the dataset using data augmentation methods such as Mix Up and Mosaic to achieve better results for the model, as detailed in Section 4.2.

4.2. Train trick

4.2.1. Mix up

Mix up is an image blending augmentation scheme proposed in 2018, which can effectively improve the generalization ability of the model and allow the model to learn each class feature better. Mix up is to interpolate two images proportionally to generate mixed samples to expand the number of samples in the dataset. Mix up is to interpolate two images proportionally to mix the samples. We randomly choose two images within a Batch for mixing, let the samples be (x_i, y_i) and (x_j, y_j), after which we obtain the new image by weighted linear interpolation:

$$\hat{x} = \lambda x_i + (1-\lambda)x_j, \tag{8}$$

$$\hat{y} = \lambda y_j + (1-\lambda)y_j, \tag{9}$$

The formula has any value of $\lambda \in [0,1]$ and obeys the $\text{Beta}(\alpha, \alpha)$ distribution, where α is the hyperparameter and the default $\alpha = 1$. Currently the best training results are obtained when λ is taken as 0.5. As in Figure 9, the top left and bottom right corners are λ taken as 0 and 1 respectively, i.e., the two original plots. In the middle is the generated image when the value 0.5 is taken, when both ships are clearly shown in outline.

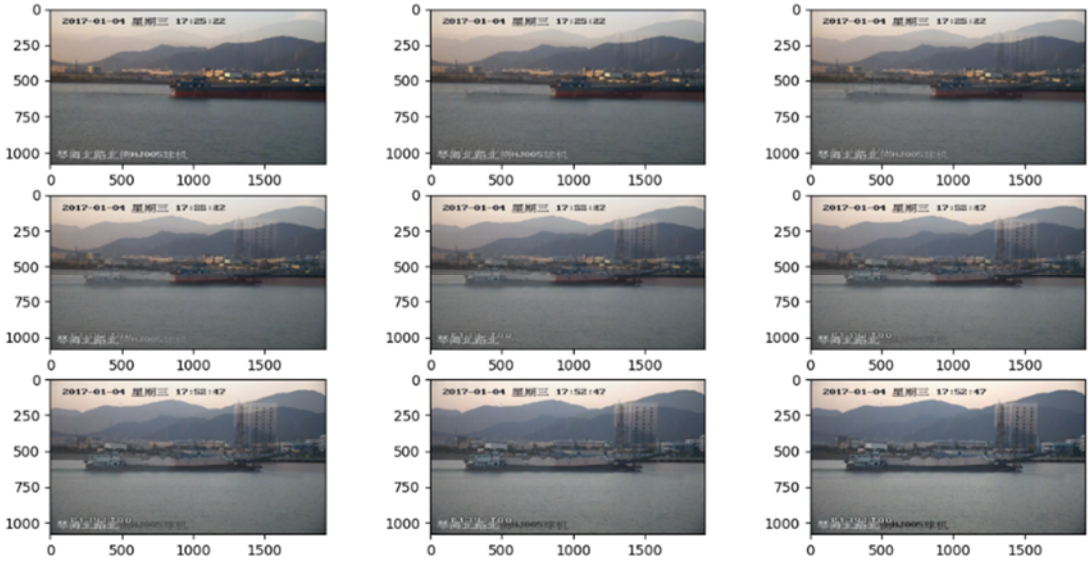


Figure 9. Generation diagram of lambda taking different values.

4.2.2. Mosaic

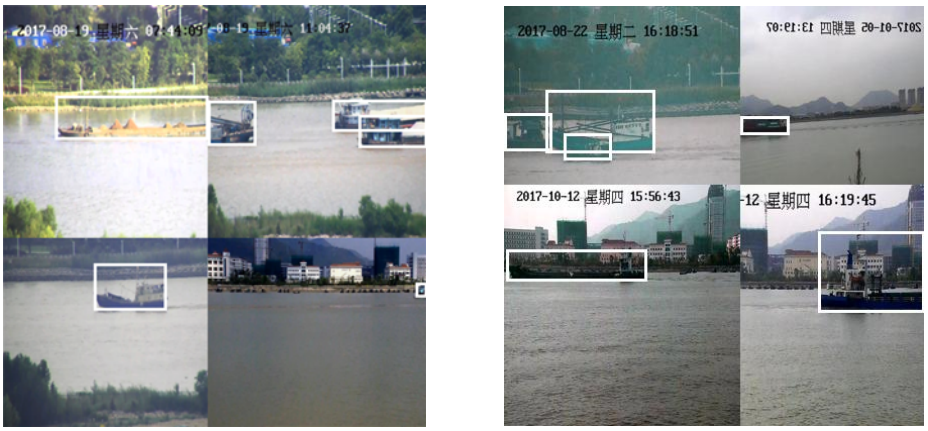


Figure 10. Mosaic composite picture.

The Mosaic approach to data enhancement was first introduced in YOLOv4. Its approach to data enhancement is based on CutMix, which crops the image in parts, fills in the pixels in other areas of the training set with the rest of the data, and distributes the classification results in a proportional way. Mosaic uses four images stitched together, so that the complex background noise is given to a great extent and the data from the four images is computed in the BatchNorm calculation. This can effectively increase the complexity of the image context, allowing the model to have more difficult samples to participate in the training. The algorithm is implemented as follows:

1) Randomly select four images from the training set.
2) Flip, scale and colour map each of the four images.
3) Combine the images and draw a frame based on their relative positions.

In the process of generating images, there is a problem that the real boxes of some classified objects are out of bounds, and we need to do some cropping of the generated boxes. The processing of the images is consistent with the training data.

As in Figure 10, the generated images will participate in the training of the model with the original images. The model has more samples to learn the category features. Also, the generated images are cropped and color shifted so that the model can filter the background information from more complex samples.

4.3. Evaluation index

In order to fully and accurately evaluate the recognition performance of model ships, this paper uses common metrics in the field of object detection to evaluate and compare different model algorithms. The average precision (AP) and the precision-recall (PR) curves for different classes of ships are plotted.

Here we briefly give the definitions of precision, recall, AP and PR curves. We set the total number of samples annotated to NP. When the overlap area between the IoU calculated by the bounding box and the real box exceeds a threshold, these samples are set as positive (TP) and the wrong ones are marked as negative (FP), while the samples with wrong detection by the real box are set as (FN). Recall indicates the proportion of the correct part of the overall detection result to the useful part of the entire dataset, and Precision indicates the proportion of the useful part of the overall detection result to the useful part of the entire detection result.

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \tag{10}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \tag{11}$$

Since the Precision and Recall metrics can conflict in special case situations, we can plot a Precision-Recall curve based on the Recall and Precision values for each category. For each different Recall value, the maximum Precision maximum at the time of these values is selected and the area under the PR curve is calculated as the AP value. Mean Average Precision (mAP) represents the average AP for all n categories. We use mAP as the model's accuracy evaluation indicator, where mAP@.5:.95 represents the average mAP at different IOU thresholds (from 0.05 to 0.95 and step size is 0.05).

$$AP = \int_0^1 P(R)dR \tag{12}$$

$$mAP = \frac{\sum_{i=1}^{n} AP_i}{n} \tag{13}$$

The experiments also relied on Frame Per Second (FPS), Parameters and Floating point operations (FLOPs) to evaluate the speed and complexity of the model operations, and the following formulas were used to calculate some of these metrics:

$$FPS = \frac{1}{t} \tag{14}$$

$$\text{Parameters} = (K_h * K_w * C_i n) * C_{out} + C_{out} \tag{15}$$

$$FLOPs = [(K_h * K_w * C_i n) * C_{out} + C_{out}] * (H * w) \tag{16}$$

where t is the average elapsed time for all images, Equations (15) and (16) are the Parameters and FLOPs of the current convolutional layer, and the total Parameters and FLOPs of the network sum to the superposition of all network layers.

4.4. Results and analysis

We have designed two parts of experiments, namely ablation experiments and multiple model comparison experiments, to investigate the impact of the improved modules on the overall algorithm and the overall performance metrics of the model. In the first part of the experiments, YOLOv5 is used as the benchmark model for multiple ablation experiments, where different modules are decoupled and added to the model to compare the performance improvement brought by different modules. In the second part of the experiments, in order to verify the overall performance of the model, multiple sets of classical object detection algorithms are designed for comparison, such as SSD, RetinaNet, Faster RCNN, YOLOv3, FCOS, etc.

4.4.1. Ablation experiment

Combining Lk-YOLO with the anchorless frame detection algorithm flow and dynamic label assignment algorithm, we can obtain an anchorless frame ship detection model. For different ship datasets, there is no need to perform K-means clustering to generate anchor frames and remove the NMS to screen the prediction frames to achieve an end-to-end ship detection process.

As in Table 2, the experimental results obtained by Lk-YOLO on the SeaShips dataset are as follows:

Table 2. Detection results of Lk-YOLO in the Seaships dataset.

	All	Ore carrier	General cargo ship	Bulk cargo carrier	Container ship	Fishing boat	Passenger ship
$mAP^{0.5}$	0.977	0.967	0.971	0.975	0.993	0.978	0.977
$mAP^{.5:95}$	0.774	0.751	0.795	0.804	0.824	0.713	0.759

Table 2 shows the mAP data under each ship type for both thresholds of the Lk-YOLO algorithm model. The model has 99.3% of AP values for container ships, mineral sands ships and fishing vessels, especially at mAP@.5:.95, show a significant decrease in AP values. This is due to the difficulty in detecting the network due to the small number of fishing vessel samples in the dataset, the small size of the vessels, and the occurrence of scenarios. Additionally, the mineral sands ship is only different from other ships in the way of usage, and the hull shape is similar to other types of ships, which makes it difficult for the model to extract the correct features, and therefore a situation of misclassification will occur, resulting in a smaller AP for the class under correct classification.

As shown in Figure 11, the model maintains a relatively smooth loss function curve for each type during the training process. However, at 100 cycles of learning, due to the shutdown of some training strategies, such as warm up and other learning strategies, the model shows a certain degree of fluctuation and the response appears in a large number of dense points in each graph. The accuracy

and recall of the model performed very well, also reflected in the curves of mAP@50 and mAP@.5:.95. The table demonstrates that the regression of the model's prediction frame can achieve excellent results with high IoU thresholds. Each prediction frame has a high overlap with the ground truth.

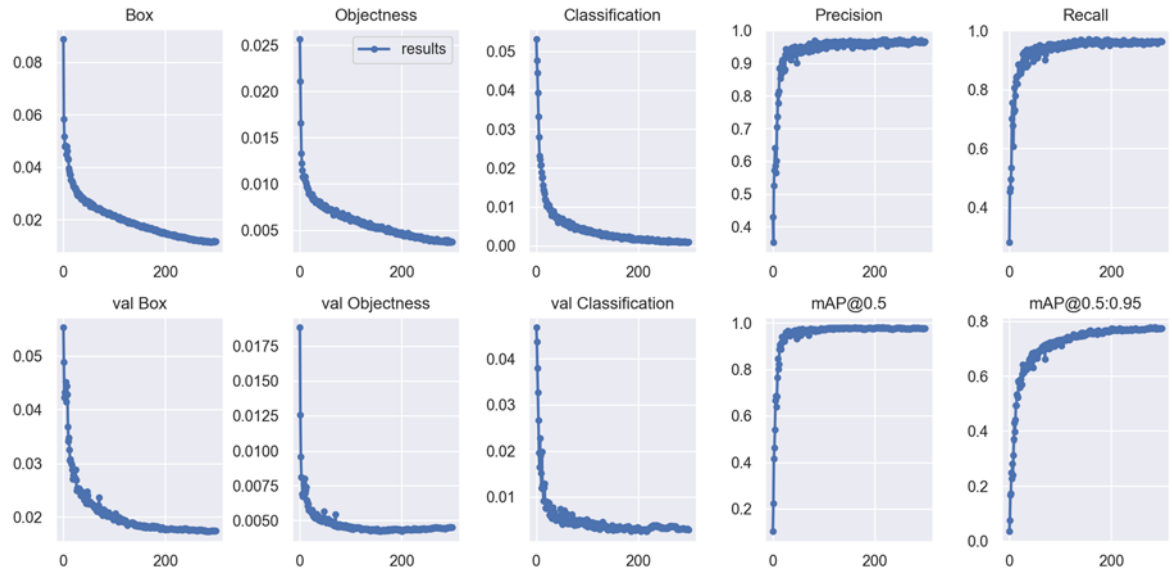


Figure 11. Lk-YOLO experimental data.

In this paper, ablation experiments were conducted using a modified Backbone as the baseline model to verify the performance improvement of the original model by the anchorless frame improvement algorithm and the changes to the original model, and to understand the impact of different modules on the performance under different evaluation metrics. The changes in model performance were recorded for each of the Bank's model changes added to the baseline model, and the experimental results were analysed to verify the findings. The results of the experiments are shown in Table 3.

Table 3. The influence of different modules on ship detectors.

Model	mAP	Parameters	GFLOPs
Baseline	0.953	15.4	25.1
+Data Augmentation	0.9	15.4	25.1
+Anchor free	0.945	14.8	20.5
+Label assign	0.969	14.8	20.5
+Decoupled Head	0.977	15.2	23.6

As shown in Table 2, the module designed in this paper increases the complexity and inference time of the model, but improves the overall performance of the model by 3% mAP. Afterwards, we choose the modified CSPDarkNet as the Baseline for our model, and we can see that the model, with a slight increase in the number of parameters, improves the mAP to 95.3% over YOLOv5s with a slight performance improvement and some increase in the number of parameters. Later, after training with some data augmentation strategies, the mAP increases again significantly, proving that the way the data is trained is important for deep learning. Afterwards, not using anchor, while reducing the

number of parameters, also makes it difficult to correctly regress the positive samples generated during training, leading to a decrease in model performance. When we use the more efficient Label assign for training, the sample training problem caused by anchor free is well solved, so the mAP is again improved and does not introduce any extra parameter computation. Finally, we replaced the detection head with a decoupled detection head, sacrificing speed for a small parameter boost, but the mAP was again improved.

4.4.2. Comparative experiments

In this section, the improved algorithm proposed in this paper is compared with the classical deep learning object detection algorithm, and the results are shown in Table 4. To ensure the fairness of the experiments, all models use the same training strategy, training parameters and the same way of data expansion. At the same time, the models chosen were all used from the original benchmark model.

Table 4. Comparison of different detection models.

Model	mAP^{50}	mAP^{75}	mAP	FPS	Parameters	GFLOPs
Faster RCNN	0.946	0.683	0.603	75	137.0 M	370.1
RetinaNet	0.727	0.52	0.465	84	37.9 M	170.1
SSD	0.906	0.751	0.663	86	26.2 M	62.7
YOLOv3	0.921	0.663	0.674	85	61.9 M	66.1
FCOS	0.864	0.594	0.567	88	32.0 M	60
YOLOv4	0.946	0.672	0.735	80	64.3 M	64.3
YOLOv5s	0.969	0.689	0.765	120	7.2 M	15.8
YOLOv7	0.980	0.786	0.779	90	37.2 M	105.2
Ours	0.977	0.784	0.774	95	15.2 M	23.6

In this experiment, we selected a one-stage object detection network, which also includes Anchor-free FCOS, and a two-stage object detection network, where the improved performance of the object detection algorithm is judged by an IoU of 0.5, which does not allow us to distinguish between the performance of the models. We then took an IoU of 0.5 as well as the mean values of the average accuracy of the specific threshold algorithm for all classes of targets between 0.5 and 0.95 (mAP@.5:.95) in steps of 0.05, respectively. In the table mAP is the metric mAP@.5:.95.

According to Table 4, we can see that the two-stage network is better than the early one-stage network in terms of accuracy, but the speed is far inferior to the one-stage network. The current development of the one-stage network has become more mature, and not only can it be the same as the two-stage network in terms of accuracy, but it is also far better than the two-stage network in terms of speed. The algorithm proposed in this paper still has performance improvement in comparison with the improved object detection algorithm. At an IoU threshold of 0.5, the detection network does not require much real frame screening, so each detection network is able to achieve good results. When mAP@.5:.95, mAP all showed some degree of degradation. It can be seen that the prediction confidence of the Lk-YOLO data are all high, and the prediction frames are highly overlapping with GT. Lk-YOLO still achieves a higher mAP and higher performance than YOLO5s with a higher IoU confidence criterion. At the same time, it has similar performance with lower number of parameters and FLOPs than YOLOv7.

To better demonstrate the performance differences between different models under the same dataset, we show in Figure 12 the detection performance of several models with typical features for different classes of ships Faster-RCNN, RetinaNet, YOLOv5, Lk-YOLO.

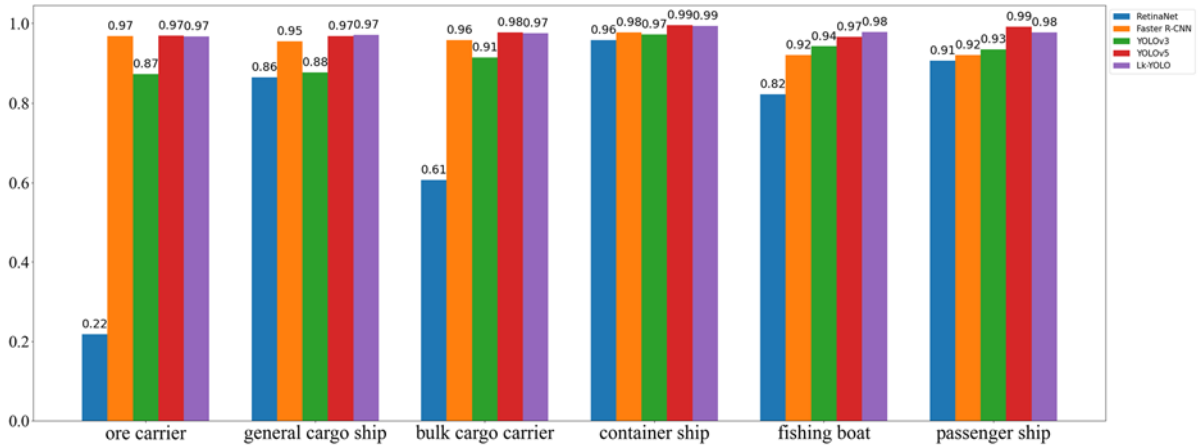


Figure 12. Experimental results of different models at mAP@50.

As can be seen in Figure 12, the models are all relatively high in mAP@50 on the SeaShips dataset. Only in the lower performance RetinNet in ore carrier achieves 21.8% AP value. None of the other classes of ships have excessive gaps. mAP@50, as a common metric, does not visualize the performance gap of the detectors with this dataset.

Therefore, we again plot the comparison of different models using mAP@.5:.95 as the evaluation metric. Since the performance gap between RetinaNet and the other models is too large, it is not put into the analysis under this metric.

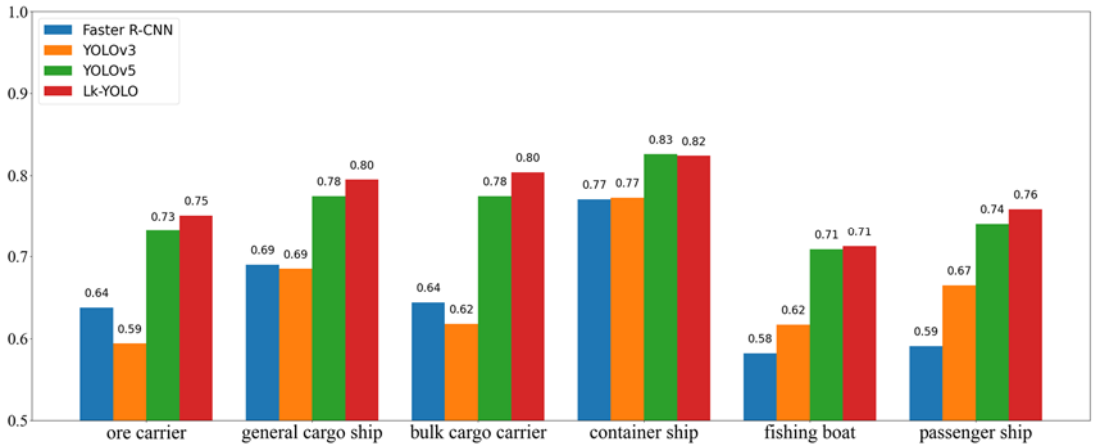


Figure 13. Experimental results of different models at mAP@.5:.95.

Figure 13 shows that Lk-YOLO has a clear performance advantage over other models. container ship is the best category in the dataset due to its huge size and the rich color information of the various containers on board. YOLOv5 achieves the best performance of 83% on container ship, which is slightly higher than Lk-YOLO.

On the small target fishing ship, Faster R-CNN only achieves 58% of the AP value, and Lk-YOLO is only a little ahead of YOLOv5 performance. For the remaining categories, Lk-YOLO achieves a clear advantage. ore carrier is not conducive to anchor frame regression due to its very flat and long characteristics and is very prone to overlap in the dataset. general cargo ship and bulk cargo carrier as the main vessels in the channel also have a large number of overlapping samples. It can be seen that Lk-YOLO is useful for blurring the boundaries in some scenarios.

The results of the experimental detection visualization are shown in Figure 11, and the performance of Lk-YOLO for some visualizations in the dataset:

Figure 14. Detection results.

Figure 15. Comparison of detection results between different models.

It can be seen that the model is able to identify multiple types of vessel results with high confidence scores in some multi-target scenarios. For the case of the combination of large and small targets in Figure 14(a),(b), there is a huge scale change in the prediction frame, and the model is still able to correctly classify and accurately locate the fishing vessel targets without any impact. In Figure14(c), both vessels are still able to identify the position and classification completely in the

complex background as well as truncated by the frame. When the vessels overlap to a certain extent as in Figure 14(f), the detector is able to return to the area shown by the object, and the bounding box of the ore carrier is also well identified across the bounding box of the bulk cargo carrier.

5. Conclusions

In order to solve the current problem of optimizing the anchor parameters of the ship detector in visible light, this paper proposes the Anchor-free detector Lk-YOLO. In this paper, we first redesign the residual units in Backbone by adding large kernels, and by combining the Depth-wise convolution of the 7 × 7 large kernel convolution and the Depth-wise convolution, without significantly increasing the model of FLOPs and the number of parameters to improve the detection accuracy of the model without significantly increasing the model. Second, the decoupled detection head is constructed, and the detection head is designed as a parallel structure to divide the classification and regression into different branching structures, so that the classification and regression tasks no longer share weights and solve the problem of branching module conflicts. Finally, the anchor-free strategy is used to replace a large number of a priori anchors with feature centroids, which solves the complex hyperparameter problem caused by anchors and avoids the calculation of anchor parameters by manual or K-means clustering, further simplifying the number of parameters of the model. Moreover, the dynamic Label assign method is chosen to solve the positive sample problem of anchorless training. During the training process, the classification score and regression score of each prediction frame are weighted and evaluated, allowing the detector to focus on the high-quality positive samples during training. The experiments are based on the Seaship datasets from Wuhan University, and the commonly used performance indicators of object detection, such as mAP, FPS, parameters, etc., are used to conduct ablation experiments on the same network to select different thresholds of IoU, different object detection algorithms under the same threshold and improved components to effectively and accurately evaluate the improvement of different modules on the network, and the design network on other networks superiority. Lk-YOLO reached 97.7% on mAP@50 and 77.4% on mAP@.5:.95. In the future, we will further improve the ship recognition detection network under visible light, while proposing better solutions to some still existing problems and optimizing and upgrading the network model algorithms.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by the National Natural Science Foundation of China (Grant no. 62271303) and Shanghai Pujiang Program (Grant no. 22PJD029).

Conflict of interest

The authors declare that there is no conflict of interest.

References

1. X. Xing, K. Ji, H. Zou, W. Chen, J. Sun, Ship classification in TerraSAR-X images with feature space based sparse representation, *IEEE Geosci. Remote Sens. Lett.*, **10** (2013), 1562–1566. https://doi.org/10.1109/LGRS.2013.2262073
2. J. Wei, P. Li, J. Yang, J. Zhang, F. Lang, A new automatic ship detection method using L-band polarimetric SAR imagery, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **7** (2017), 1383–1393. https://doi.org/10.1109/JSTARS.2013.2269996
3. M. Kang, K. Ji, X. Leng, Z. Lin, Contextual region-based convolutional neural network with multilayer fusion for SAR ship detection, *Remote Sens.*, **9** (2017), 860. https://doi.org/10.3390/rs9080860
4. T. Zhang, X. Zhang, A mask attention interaction and scale enhancement network for SAR ship instance segmentation, *IEEE Geosci. Remote Sens. Lett.*, **19** (2022), 1–5. https://doi.org/10.1109/LGRS.2022.3189961
5. Y. Feng, L. Wang, M. Zhang, A multi-scale target detection method for optical remote sensing images, *Multimedia Tools Appl.*, **78** (2019), 8751–8766. https://doi.org/10.1007/s11042-018-6325-6
6. Z. Li, D. Yang, Z. Chen, Multi-layer sparse coding based ship detection for remote sensing images, in *IEEE International Conference on Information Reuse & Integration*, (2015), 122–125. https://doi.org/ 10.1109/IRI.2015.28
7. X. Yang, H. Sun, K. Fu, J. Yang, X. Sun, M. Yan, et al., Automatic ship detection of remote sensing images from Google Earth in complex scenes based on multi-scale rotation dense feature pyramid networks, *Remote Sens.*, **10** (2018), 132. https://doi.org/10.3390/rs10010132
8. L. Ma, W. Xie, H. Huang, Convolutional neural network based obstacle detection for unmanned surface vehicle, *Math. Biosci. Eng.*, **17** (2019), 845–861. https://doi.org/10.3934/mbe.2020045
9. D. D. Bloisi, F. Previtali, A. Pennisi, D. Nardi, M. Fiorini, Enhancing automatic maritime surveillance systems with visual information, *IEEE Trans. Intell. Transp. Syst.*, **18** (2017), 824–833. https://doi.org/10.1109/TITS.2016.2591321
10. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, **1** (2005), 886–893. https://doi.org/10.1109/CVPR.2005.177
11. P. F. Felzenszwalb, R. B. Girshick, D. McAllester, Cascade object detection with deformable part models, in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, (2010), 2241–2248. https://doi.org/10.1109/CVPR.2010.5539906
12. Z. Q. Zhao, P. Zheng, S. T. Xu, X. D. Wu, Object detection with deep learning: a review, *IEEE Trans. Neural Networks Learn. Syst.*, **30** (2019) 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865
13. A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet classification with deep convolutional neural networks, in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, Lake Tahoe, NV, USA, **30** (2017), 1097–1105. https://doi.org/10.1145/3065386
14. K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, preprint, arXiv:1409.1556.
15. R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, (2014), 580–587. https://doi.org/10.1109/CVPR.2014.81

16. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, et al., SSD: single shot multibox detector, in *Proceedings of the European Conference on Computer Vision*, **9905** (2016), 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
17. J. Redmon, S. Divvala, R. Girshick, You only look once: unified, real-time object detection, in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 779–788. https://doi.org/10.1109/CVPR.2016.91
18. R. Girshick, Fast R-CNN, in *Proceedings of the International Conference on Computer Vision*, (2015), 1440–1448. https://doi.org/10.1109/ICCV.2015.169
19. S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, *IEEE Trans. Pattern Anal. Mach. Intell.*, **39** (2017), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
20. T. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in *Proceedings of the IEEE International Conference on Computer Vision*, **42** (2017), 2999–3007. https://doi.org/10.1109/TPAMI.2018.2858826
21. T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017), 2117–2125. https://doi.org/10.1109/CVPR.2017.106
22. S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 8759–8768. https://doi.org/10.1109/CVPR.2018.00913
23. G. Ghiasi, T. Lin, R. Pang, Q. Le, NAS-FPN: Learning scalable feature pyramid architecture for object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2019), 7029–7038. https://doi.org/10.1109/CVPR.2017.106
24. M. Tan, R. Pang, Q. V. Le, EfficientDet: Scalable and efficient object detection, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, (2020), 10778–10787. https://doi.org/10.1109/CVPR42600.2020.01079
25. K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2016), 770–778. https://doi.org/10.1109/CVPR.2016.90
26. S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, (2017), 1492–1500. https://doi.org/10.1109/CVPR.2017.634
27. L. Huang, Y. Yang, Y. Deng, Y. Yu, DenseBox: Unifying landmark localization with end to end object detection, preprint, arXiv:1509.04874.
28. H. Law, J. Deng, CornerNet: Detecting objects as paired keypoints, in *Proceedings of the European Conference on Computer Vision*, (2020), 734–750. https://doi.org/10.1007/s11263-019-01204-1
29. X. Zhou, D. Wang, P. Krähenbühl, Objects as points, preprint, arXiv:1904.07850.
30. J. Redmon, A. Farhadi, YOLO9000: Better, faster, stronger, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 6517–6525. https://doi.org/10.1109/CVPR.2017.690
31. J. Redmon, A. Farhadi, YOLOv3: An incremental improvement, preprint, arXiv:1804.02767.
32. J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in *IEEE Conference on Computer Vision and Pattern Recognition*, (2018), 7132–7141. https://doi.org/10.1109/CVPR.2018.00745

33. S. Woo, J. Park, J. Y. Lee, I. S. Kweon, CBAM: convolutional block attention module, preprint, arXiv:1807.06521.
34. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., Attention is all you need, preprint, arXiv:1706.03762.
35. A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, et al., An image is worth 16x16 words: Transformers for image recognition at scale, preprint, arXiv:2010.11929.
36. Z. Shao, W. Wu, Z. Wang, W. Du, C. Li, Seaships: a large-scale precisely annotated dataset for ship detection, *IEEE Trans. Multimedia*, **20** (2018), 2593–2604. https://doi.org/10.1109/TMM.2018.2865686
37. Z. Shao, L. Wang, Z. Wang, W. Du, W. Wu, Saliency-aware convolution neural network for ship detection in surveillance video, *IEEE Trans. Circuits Syst. Video Technol.*, **30** (2020), 781–794. https://doi.org/10.1109/TCSVT.2019.2897980
38. H. Li, L. Deng, C. Yang, J. Liu, Z. Gu, Enhanced YOLOv3 tiny network for real-time ship detection from visual image, *IEEE Access*, **9** (2021), 16692–16706. https://doi.org/10.1109/ACCESS.2021.3053956
39. X. Han, L. N. Zhao, Y. Ning, J. F. Hu, ShipYOLO: An enhanced model for ship detection, *J. Adv. Transp.*, **2021** (2021), 11. https://doi.org/10.1155/2021/1060182
40. J. H. Kim, N. Kim, Y. W. Park, C. S. Won, Object detection and classification based on YOLO-V5 with improved maritime dataset, *J. Mar. Sci. Eng.*, **10** (2022), 377. https://doi.org/10.3390/jmse10030377
41. Z. Chen, D. Chen, Y. Zhang, X. Cheng, M. Zhang, C. Wu, Deep learning for autonomous ship-oriented small ship detection, *Safety Sci.*, **130** (2020). https://doi.org/10.1016/j.ssci.2020.104812.
42. Z. Tian, C. Shen, H. Chen, T. He, FCOS: Fully convolutional one-stage object detection, in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, (2019), 9626–9635. https://doi.org/10.1109/ICCV.2019.00972
43. S. Zhang, C. Chi, Y. Yao, Z. Lei, S. Z. Li, Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection, in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2020), 9756–9765. https://doi.org/10.1109/CVPR42600.2020.00978
44. Z. Ge, S. Liu, Z. Li, O. Yoshie, J. Sun, OTA: Optimal transport assignment for object detection, in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR*), (2021), 303–312. https://doi.org/10.1109/CVPR46437.2021.00037

AIMS Press