HDSNet: Hierarchical Detection Structure for YOLO Series Models

Zhong Qu (quzhong@cqupt.edu.cn)
Chongqing University of Posts and Telecommunications
Leyuan Gao
Chongqing University of Posts and Telecommunications
Shengye Wang
Chongqing University of Posts and Telecommunications

Research Article

Keywords: Anchor-base Detector, Feature Pyramid Network, Hierarchical Detection Structure, Multi-scale Feature Fusion, YOLO

Posted Date: April 15th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1544802/v1

HDSNet: Hierarchical Detection Structure for YOLO Series Models

Zhong Qu · Leyaun Gao · Shengye Wang

Received: ***** / Revised: **** / Accepted: **** / Published online: ***

Abstract The feature pyramid network is an effective feature enhancement network. It is proposed and proved that its success is due to a divide-and-conquer solution to the optimization problem in object detection rather than multi-scale feature fusion. Divide-and-conquer refers to using feature layers of different scales as prediction feature layers instead of using one single object detection head. In this paper, we propose the hierarchical detection structures on YOLOv3-5 baseline models, which have good improvements. Firstly, we add a 7×7 grid as a prediction feature layer on YOLOv3, achieving 67.8% *mAP*@.5 and 46.3% *mAP*@.5:.95 on VOC and COCO datasets, which are 1.9% and 0.8% higher than the original. Then, we add 10×10 and 5×5 grids simultaneously on YOLOv4-5 as prediction feature layers. On VOC and COCO datasets, YOLOv4 achieves 61.8% *mAP*@.5 and 36.4% *mAP*@.5:.95, which are 1.6% and 1.1% higher than the original. YOLOv5 has four different models of *S*, *M*, *L*, and *X*. Finally, we do comparative experiments on all four models. For VOC dataset, 61.3%, 66.6%, 69.9% and 70.9% *mAP*@.5:.95 are achieved in YOLOv5_S-X, which are 4.4%, 1.4%, 1.6% and 1.1% higher than the original, respectively. The results of comparative experiments on each model show that the hierarchical detection structure based on anchors is effective for YOLO series models.

Keywords: Anchor-base Detector · Feature Pyramid Network · Hierarchical Detection Structure · Multi-scale Feature Fusion · YOLO

1 Introduction

OBJECT detection architectures are divided into one-stage detectors and two-stage detectors. The two-stage object detectors are represented by Region Convolutional Neural Network (R-CNN) [1] series, including Fast R-CNN [2] and Faster R-CNN [3]. The one-stage object detectors are represented by Single Shot multi-box Detector (SSD) [4] and You Only Look Once (YOLO) [5] series, including YOLOv1-5 [5-9], You Only Look One-level Feature (YOLOF) [10], and YOLOX [11].

From another perspective, object detection methods can be divided into two types, namely, the anchor-base methods and the anchor-free methods. The two methods have different advantages and shortcomings. In the object detection head, the anchor-base methods comprehensively generate candidate boxes, filter candidate boxes through non-maximum suppression, and reserve only useful anchor boxes. The anchor-free methods don't generate candidate boxes in the detection head but restrict the search space of the object position based on key points. The anchor-base methods have higher detection accuracy, while the anchor-free methods have faster detection speed.

At the beginning of neural network application in object detection, Faster R-CNN [3] proposed the Region Proposal Network (RPN), which can generate nine candidate boxes of different proportions at each position of the image. These candidate boxes are called anchors. R-CNN network and YOLO network are typical anchor-base detectors, and YOLOX is the typical anchor-free detector.

Our work is based on the one-stage anchor-base methods, and the architecture of YOLO series models combined with multi-scale anchors are discussed, studied, and verified. The improvements to YOLOv3-5 models are proposed, and the effectiveness of our methods is proved by experiments. We conclude that the multi-scale anchor detectors are effective in the detection of YOLO series models.

In this paper, the multi-scale anchor boxes detection methods are implemented on YOLOv3-5 respectively, and we verify the effectiveness of our methods on the PASCAL Visual Object Classes (PASCAL VOC) and Microsoft Common Objects in Context (MS COCO) datasets. For these two datasets, our methods are all improved in YOLOv3, YOLOv4, and YOLOv5_S-X. Our main contributions are as follows:

(1) Our improvement on YOLOv3: We change the architecture of YOLOv3, taking the pure YOLOv3 network as the baseline, and changing the three detection heads into four detection heads. As for the 416×416 images, the scales of the four detection heads are 52×52, 26× 26,13 ×13, and 7×7.

(2) Our improvement on YOLOv4: We change the architecture of YOLOv4, taking the pure YOLOv4 network as the baseline, and changing the three detection heads into five detection heads. As for the 608×608 images, the scales of the five detection heads are 76×76, 38×38, 19×19, 10×10, and 5×5.

(3) Our improvement on YOLOv5_S-X: We change the architecture of YOLOv5_S-X, taking the pure YOLOv5_S-X network as the baseline, and change the three detection heads into five detection heads, too. As for the 640×640 images, the scales of the five detection heads are 80×80, 40×40, 20×20, 10×10, and 5×5.

(4) We name the different detection heads of different scales as hierarchical detection structure. Then, we verify the effectiveness of the proposed methods through experiments, the experiments show that the hierarchical detection structure based on multi-scale anchors is effective for YOLO series models.

The rest of this paper is arranged as follows. In Sec. 2, the YOLO series models and feature pyramid network are introduced. Sec. 3 introduces our proposed the hierarchical detection structures on YOLOv3, YOLOv4, and YOLOv5_S-X. Sec. 4 shows the experiments and the analysis of results on these models. In Sec. 5, the discussion and conclusions are drawn respectively.

2 Related Work

In this section, we briefly introduce the development process of YOLO series models and feature pyramid networks, which are the basis of our subsequent research.

2.1 YOLO Series Models

YOLOv1 [5] algorithm was proposed by Redmon *et al.* It directly defined object detection as a single regression problem and firstly proposed to divide an image into S×S grids and deal with each grid separately. It broke the previous traditional R-CNN methods of directly predicting the overall processing of images and did not use anchors for bounding box coordinates and class probability when predicting the result of S×S grids. However, YOLOv2-5 [6-9] algorithms had continued to use anchors for regression prediction, mapping anchors into the grids, which verifies the importance of anchor-base for object detection.

YOLOv2 [6] added many tricks conducive to model training based on YOLOv1, including batch normalization [12], fine-grained features [27], multi-scale training [28], and convolutional with anchor boxes [29]. The most important improvement was to use object boundary box prediction based on anchors.

YOLOv3 [7] used a new backbone network, Darknet-53. This new network is made up of a series of 1×1 and 3×3 convolutional layers stacked, which is named Darknet-53 because there are altogether fifty-three convolutional layers. YOLOv3 is the first to propose predicting three different layers of features, and each feature has three different anchor sizes.

The prediction process of YOLOv4 [8] is the same as YOLOv3, which also has three different feature layers for prediction, but the scales of these three feature layers are different according to the input images of different sizes. YOLOv4 used CSPDarknet53 as the backbone network, Meanwhile, the Spatial Pyramid Pooling Network (SPP) [13] and Path Aggregation Network (PANet) [14] were added in the feature processing part. Wang *et al.* proposed Cross Stage Partial Networks (CSPNet) [22] as a new backbone network to extract features. It has a good detection effect in different datasets [30].

YOLOv5 [9] used the BottleneckCSP network as the backbone for feature extraction. It proposed to use mosaic enhancement in the image preprocessing stage, a focus module for slicing the input image, and four scale models for different projects. These modules can not only make the inference of the network faster but also improve detection accuracy. In addition, many other training tricks were used, including training warmup [15], cosine annealing learning rate [16], autoauto anchor, label smoothing [17], and so on.

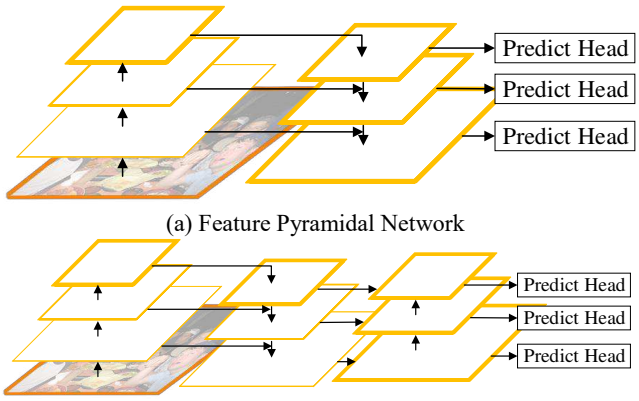
2.2 Feature Pyramid Network

He *et al.* [18] proposed the Feature Pyramid Network (FPN), which broke the traditional prediction method using only top-level features. As shown in Fig. 1(a), the prediction of FPN is implemented independently at different feature layers. FPN was firstly proposed based on the residual network [19]. By extracting shallow features and fusing them with deep features, then the fused features are predicted at the object detection head. Path Aggregation Network (PANet) [14] is the optimization of FPN, which is proposed by Liu *et al.* As shown in Fig. 1(b), the difference of PANet lies in the addition of a bottom-up path, which obtains more effective information through feature extraction and feature fusion again, and predicts the feature layers on this path.

2.3 You Only Look at One-level Feature

Chen *et al.* [10] proposed YOLOF, which had proposed and proved the success of FPN due to the divide-and-conquer solution to the optimization problem in object detection rather than multi-scale feature fusion. YOLOF proposed Multiple-in-Multiple-out (MiMo), Single-in-Multiple-out (SiMo), Multiple-in-Single-out (MiSo), and Single-in-Single-out (SiSo) structures based on RetinaNet [20], which are implemented 35.9%, 35.0%, 23.9%, and 23.7% *mAP@*.5 on MS COCO dataset, respectively. The MiMo and SiMo structures are shown in Fig. 2. The MiMo structure is only 0.9% higher than the SiMo structure, but the MiMo structure is 12% higher than the MiSo structure. The multi-scale feature prediction plays a much more important role than the multi-scale feature fusion in influencing the success of FPN.

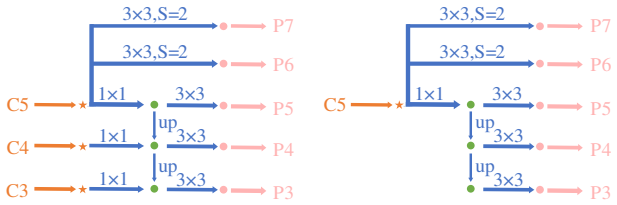
Whether such a conclusion applies to all models is a question we have been thinking about, and it is also the direction we are going to solve. Therefore, we conduct the verification based on three different YOLO models.



(a) Feature Pyramidal Network

(b) Path Aggregation Network

Fig. 1 The structures of FPN and PANet. (a) The left is the bottom-up path, the right is a top-down path, and the middle represents feature fusion; (b) Compared with the FPN structure, there is another bottom-up path on the right side, which can be predicted on this path.



(a) Multiple-in-Multiple-out structure (b) Single-in-Multiple-out structure

Fig. 2 The structures of Multiple-in-Multiple-out (MiMo) and Single-in-Multiple-out (SiMo). SiMo has less C3 and C4 input paths in the feature layers in the RetinaNet backbone than MiMo.

3 Proposed Methods

In this section, we introduce the improvements in YOLOv3, YOLOv4, and YOLOv5_S-X architectures in order. We explain each module and its functions in detail.

3.1 Hierarchical Detection Structure on YOLOv3

Feature Extraction: We improve the backbone network used by the YOLOv3 baseline model, and form a new backbone network. The backbone network plays an important role in feature extraction in each architecture and is the most important module for each network to accomplish learning tasks. The backbone network used by the YOLOv3 model is Darknet-53, as shown in Fig. 3, which is the overall architecture of our improved YOLOv3 model, with each different module marked in a different color. The size of the input image is 416×416. We add a residual block, Res-block, to the DarkNet-53 backbone, so we can extract the 7×7 predictive feature

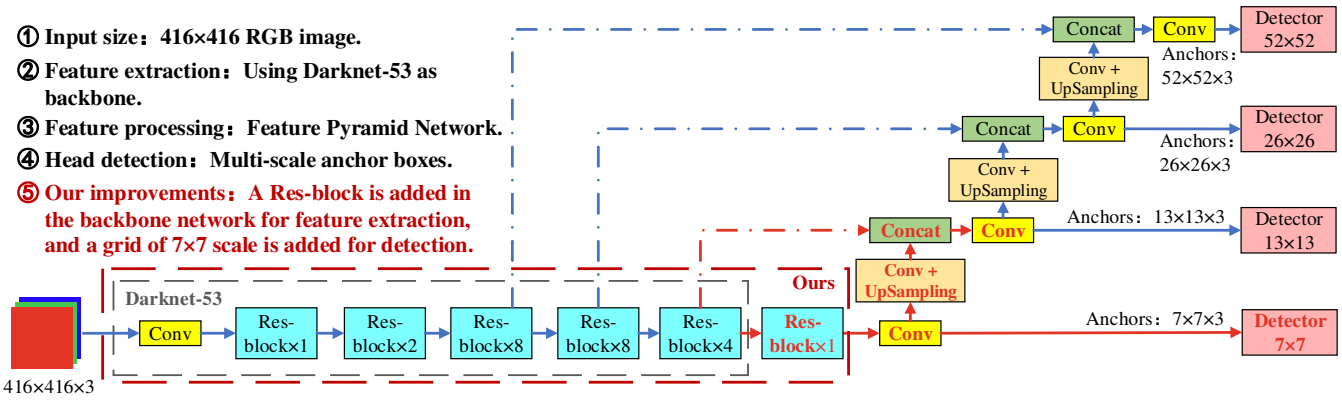


Fig.3 The improved network architecture of the YOLOv3 baseline. We simply show the 2D layout and the important modules of the network structure. The channel dimension is not shown. The part marked in red are the key to the new network structure.

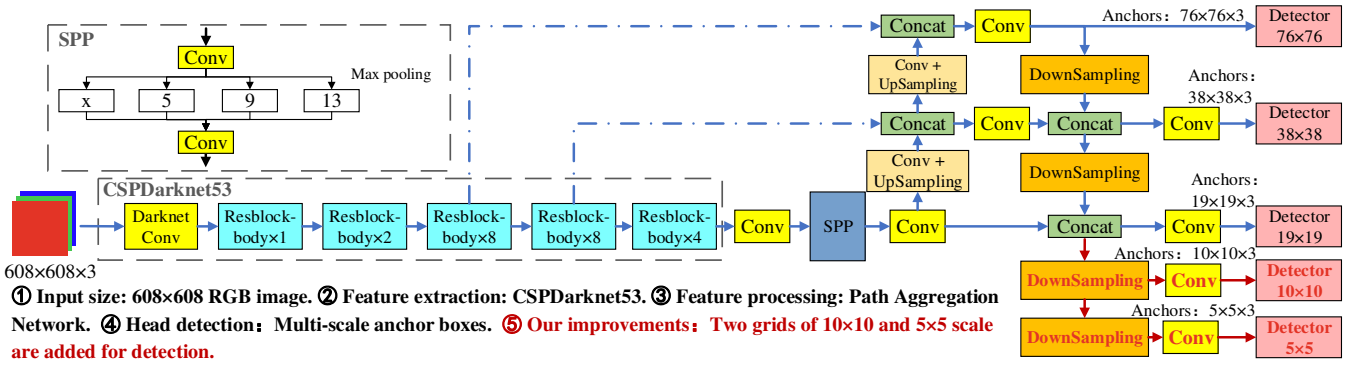


Fig.4 The improved network architecture of YOLOv4. We simply show the 2D layout and the important modules of the network structure. The channel dimension is not shown. The part marked in red are the key to the new network structure.

Table 1 Anchors Size for Our Improved YOLOv3 Model.

Feature map	Anchors size	Anchors Number
52×52×256	[10,13, 16,30, 33,23]	52×52×3=8112
26×26×512	[30,61, 62,45, 59,119]	26×26×3=2028
13×13×1024	[116,90, 156,198, 373,326]	13×13×3=507
7×7×2048	[153,103, 186,213, 458,426]	7×7×3=147

layers through this block. The original Darknet-53 has five Res-blocks, but the improved backbone network has six Res-blocks. The new backbone network can also play the role of feature extraction, which is the basis for enhanced feature processing.

Feature Processing: With our improvement of the backbone network, the FPN is strengthened and more useful feature information can be obtained. With the deepening of neural network depth, the deep network contains higher semantic information, while the shallow network contains more object location information. The semantic information helps to identify the category of objects and the object location information helps to identify the coordinates of this object. The FPN proposes to combine the respective advantages of shallow and deep feature layers, and it is proven that it can achieve better results. Our improved network also maintains the FPN structure, with feature fusion at 1024, 512, and 256 scales instead of only 512 and 256 scales. We analyze that the feature layer of the 1024 scale contains the feature information which the 512 and 256 scales do not have and cannot be ignored. Therefore, we also add feature fusion for the feature layer with the 1024 scale. Feature layers of the same scale are extracted from the backbone network and fused with feature layers of the same scale on the FPN structure. In the processing of continuous extraction and fusion, a lot of useful information that has been ignored can be used again, and the features of this information can help accurate prediction.

Feature Prediction: Four feature layers of different scales can be obtained through FPN, and these four feature layers are predicted respectively. These four different scale feature layers of 52×52, 26×26, 13×13, and 7×7 are obtained through the FPN structure, and their channel dimensions are 256, 512, 1024, and 2048, orderly. Feature layers of different scales are used for objects of different sizes. From the largest scale to the smallest scale, it targets small, medium, large, and x-large objects in the image, respectively. The size and number of the most suitable anchors for each scale obtained through experiments and tuning parameters are shown in Table 1. From the YOLOF [10], we know that the number of anchor boxes affects the detection results. The improved new network model generates a total of 10794 anchors, which are 147 more anchors than the original YOLOv3. Finally, the needless anchors are filtered through the non-maximum suppression mechanism [21], and only the anchors which can accurately mark the objects are retained.

3.2 Hierarchical Detection Structure on YOLOv4

Image Preprocessing: Compared with the YOLOv3, the YOLOv4 model uses Mosaic data augmentation in the process of image preprocessing.

Feature Extraction: The backbone network used by the YOLOv4 model is CSPDarknet53. As shown in Fig. 4, it is the overall architecture of our improved YOLOv4 model. Each differ-

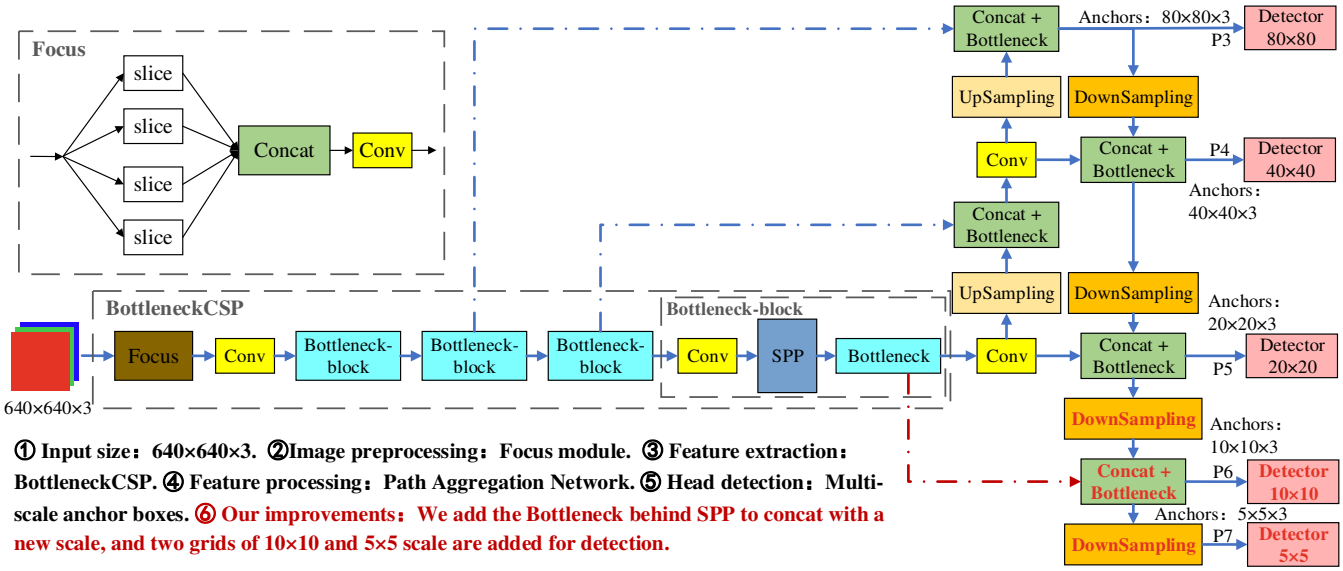


Fig.5 The improved network architecture of YOLOv5. We simply show the 2D layout and the important modules of the network structure. The channel dimension is not shown. The part marked in red are the key to the new network structure.

ent module is marked with a different color. The size of the input image is 608×608. Compared with YOLOv3, the SPP structure is added to the backbone network. Compared with Darknet-53, the CSPDarknet53 network adds a CSP structure [22] to the Darknet structure and uses five residual blocks, namely Resblock-body. Starting from the last feature layer of the backbone network, as the input of PANet, the features in the last three layers are used for feature fusion. The last three feature layers of the backbone network are three different scales, 76×76×256, 38×38×512, and 19×19×1024. The feature layers of these three scales are respectively fused with the feature layer which is the same scale in PANet, and then the prediction feature layers of three different scales can be obtained.

Feature Processing: YOLOv4 proposed the PANet structure as an enhanced feature processing network. PANet is the optimization of FPN, which also has the advantages of FPN, and obtains more feature information by adding one path. Our improved network also retains the PANet structure, adds two other feature maps of different scales through the last path, and uses the feature layers of these two scales as the prediction output.

Feature Prediction: The improved PANet can obtain five feature layers of different scales, and make predictions on these five feature layers respectively. The scales of these five feature layers are 76×76, 38×38, 19×19, 10×10, and 5×5. Like the improvement of YOLOv3, the feature layers of different scales target objects of different sizes. From the largest scale to the smallest scale, it targets small, medium, large, and x-large objects in the image, respectively. We obtain the most suitable anchors sizes of 10×10 and 5×5 scales through experiments as [156,123, 214,261, 478,432] and [182,146, 231,184, 496,456]. The other three scales are the setting of YOLOv4, [12,16, 19,36, 40,28], [36,75, 76,55, 72,146], and [142,110, 192,243, 459,401]. The improved new network model generates a total of 23118 anchors, which are 375 more anchors than the original YOLOv4 model. Finally, the needless anchors are filtered through the non-maximum suppression mechanism [21], and only the anchors which can accurately mark the objects are retained.

3.3 Hierarchical Detection Structure on YOLOv5

Image Preprocessing: Comparing the YOLOv5 model with the YOLOv4, in addition to using Mosaic data enhancement in the process of image preprocessing, a new module, Focus, is also proposed, as shown in Fig. 5.

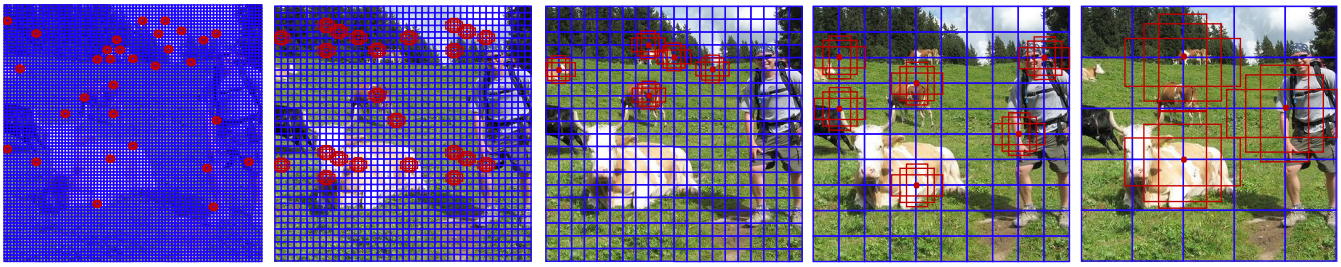
Feature Extraction: The backbone network BottleneckCSP [9] used by the YOLOv5 model is also a residual network structure. As shown in Fig. 5, it is the overall architecture of our improved YOLOv5 model. Each different module is marked with a different color. The size of the input image is 640×640. The backbone network uses four Bottleneck-block modules. In Fig. 5, we only draw the key modules of the architecture, but the details of each module are not shown. Starting from the last feature layer of the backbone network, as the input of PANet, the features in the last three layers are used for feature fusion. The last three feature layers of the backbone network are three different scales, 80×80×128, 40×40×256, and 20×20×512. The feature layers of these three scales are respectively fused with the feature layer of which is the same scale in PANet, and then the prediction feature layers of three different scales can be obtained.

Feature Processing: YOLOv5 uses the PANet structure as the reinforcement feature processing network. Our improved network also continues to use the PANet structure, and through the last path, two other feature maps of different scales, 10×10 and 5×5, can be added. As shown in Fig. 5, we add the Bottleneck behind SPP to concat with a new feature, and two grids of 10×10 and 5×5 scale are added for detection.

Feature Prediction: The improved PANet can obtain five feature layers of different scales, and make predictions on these five feature layers respectively. The scales of these five feature layers are 80×80, 40×40, 20×20, 10×10, and 5×5. As shown in Fig. 5, we name these feature layers P3, P4, P5, P6, and P7. As shown in Table 2, we obtain the most suitable anchors sizes for these two scales through experiments as [153,103, 186,213, 458,426] and [182,130, 200,234, 496,458]. The other three anchor size of different scale feature layers are [10,13, 16,30, 33,23], [30,61, 62,45, 59,119] and [116,90, 156,198, 373,326]. Like the improvement of YOLOv3 and YOLOv4, the feature layers of different scales target objects of different sizes. From the largest scale to the smallest scale, it targets small, medium, large, and x-large objects in the image, respectively,

Table 2 The five different anchor sizes and numbers of different modules for the improved YOLOv5.

Module	Feature map	Anchors size	Grid Scale	Anchors Number
P3	80×80	[10,13, 16,30, 33,23]	80×80	80×80×3=19200
P4	40×40	[30,61, 62,45, 59,119]	40×40	40×40×3=4800
P5	20×20	[116,90, 156,198, 373,326]	20×20	20×20×3=1200
P6	10×10	[153,103, 186,213, 458,426]	10×10	10×10×3=300
P7	5×5	[182,130, 200,234, 496,458]	5×5	5×5×3=75



(a) 80×80 (b) 40×40 (c) 20×20 (d) 10×10 (e) 5×5

Fig.6 The five different feature scales and corresponding anchor sizes on YOLOv5.

as shown in Fig. 6. The improved new network model generates a total of 25575 anchors, which are 375 more anchors than the original YOLOv5 model. Finally, the needless anchors are filtered through the non-maximum suppression mechanism [21], and only the anchors which can accurately mark the objects are retained.

4 Experiments Results and Analysis

In this section, we first describe experiment implementation details. Then we introduce the evaluation criterion and datasets for evaluation. Subsequently, we conduct ablation studies on the proposed improvements of the YOLOv4 and YOLOv5 models. We conduct experimental verification on YOLOv3, YOLOv4, and YOLOv5_S-X baseline models in turn.

4.1 Implementation Details

As for the latest YOLOX [11] algorithm of the YOLO series, because YOLOX is the anchor-free detector, the methods we proposed must be based on anchor-base. There is no comparability between the two, so we conduct a sufficiently experimental comparison and analysis of YOLOv3-5. As for the YOLOv1 model [5], the anchor-base method is not used, so it also cannot be compared. As for the YOLOv2 model [6], although the anchor-base method is added based on the YOLOv1, the single detection head method is used. And the structure of the YOLOv2 model is very complex. If an improved multi-scale detection head method based on the anchor-base is proposed on YOLOv2, the complexity and calculation of the network model will be doubled. We think that such experimental research is also meaningless, so no experimental verification is done on the YOLOv2 model.

We implement our improved network structure on the deep learning framework PyTorch [23]. Our network is built upon the famous open-source networks YOLOv3, YOLOv4, and YOLOv5_S-X. All the experiments are implemented on the Linux=3.10.0-1127.el7.x86_64 and GPU=GTX 2080Ti.

We do our all experiments on PASCAL VOC (Pattern Analysis, Statistical Modeling, and Computational Learning, Visual Object Classes) [24] and MS COCO (Microsoft COCO) [25] datasets. The training sets for all experiments are PASCAL VOC2007+2012 and MS COCO 2017, and the validation sets are PASCAL VOC (VOC07_test) and MS COC (test-dev 2017). Considering the time cost of network learning, we uniformly the learning epoch of each network to 100 epochs on both the VOC and the COCO datasets. We explain the experimental results of the improved three model structures on these two datasets respectively.

Table 3 Confusion Matrix. T/F represents True/False, and P/N represents Positive and Negative.

prediction(row)		Positive	Negative
actual (column)	True	TP	TN
	False	FP	FN

PASCAL VOC 2007+2012 dataset consists of about 16k training images and 5k test images over 20 object classes. MS COCO 2017 dataset involves 80 object classes. We experiment with the 80k images on the training set, 40k images on the validation set, and 20k images on the test-dev set.

4.2 Evaluation Criteria

There are many parameters for evaluating model performance for object detection algorithms, including precision, recall, average precision (*AP*), mean average precision (*mAP*), and so on. In this paper, we mainly use *Precision*, *Recall*, *AP*, and *mAP* to judge the effectiveness of the improvement models. We explain how the performance evaluation metrics are calculated through a confusion matrix [26]. A confusion matrix is made up of four components, namely, True Positive (*TP*), True Negative (*TN*), False Positive (*FP*), and False Negative (*FN*). For each object, if the prediction is correct, it is marked as True, if the prediction is wrong, it is marked as False, if the anchor contains the object, it is marked as Positive, and if there is no object in the anchors, it is marked as Negative. Therefore, for example, if the prediction of the object is the same as the actual, it is represented by *TP*; on the contrary, it is represented by *FN*, as shown in Table 3. The calculation formulas of precision and recall can be obtained as follows:

$$precision = \frac{TP}{TP + FP}, \tag{1}$$

$$recall = \frac{TP}{TP + FN}, \tag{2}$$

where p represents $precision$, r represents $recall$, so the AP can be calculated as:

$$AP = \int_0^1 p(r)dr \ . \tag{3}$$

The *mAP* represents the mean of *AP* for each class. Different datasets contain different numbers of classes. For the VOC dataset with 20 classes, *mAP* represents the average *AP* of these 20 classes. For the COCO dataset with 80 classes, *mAP* represents the average of *AP* of these 80 classes.

4.3 Ablation Studies

Our improvements on the YOLOv4 and YOLOv5 models both add two different scales, 10×10 and 5×5, to this new network structure. The effect of each new feature layer added as a prediction feature is different. Therefore, we conduct ablation studies on the effects of the model after adding different feature scales. We conduct ablation experiments on the VOC and COCO datasets.

(1) Ablation Studies on YOLOv4

As shown in Table 4, the original YOLOv4 model with only three feature scales achieves 84.3% *mAP*@.5 and 60.2% *mAP*@.5:.95 on the VOC dataset. Firstly, we add only a 10×10 grid as the prediction feature layer on the YOLOv4 baseline and achieve 85.0% *mAP*@.5 and 61.2% *mAP*@.5:.95, which are 0.7% and 1.0% higher than the original results. Then, we add only a 5 × 5 grid as the prediction feature layer and achieve 84.8% *mAP*@.5 and 60.9% *mAP*@.5:.95, which are 0.5% and 0.7% higher than the original results. Finally, we propose to simultaneously add 10×10 and 5×5 grids as prediction feature layers, and achieve 85.6% *mAP*@.5 and 61.8% *mAP*@.5:.95, which are 1.3% and 1.6% higher than the original results. From the experimental results, it can be seen if we add two grids of different scales 10×10 and 5×5 to the YOLOv4 baseline as the prediction feature layers, we can get the highest result in the VOC dataset.

As shown in Table 5, the original YOLOv4 model with only three feature scales achieves 35.3% *mAP*@.5:.95 and 44.2% AP_L on the COCO dataset. Firstly, we add only a 10×10 grid as the prediction feature layer on the YOLOv4 baseline and achieve 35.9% *mAP*@.5:.95 and 45.0% AP_L, which are 0.6% and 0.8% higher than the original results. Then, we add only a 5 × 5 grid as the prediction feature layer and aachieve35.7% *mAP*@.5:.95 and 44.8% AP_L, which are 0.4% and 0.6% higher than the original results. Finally, we propose to simultaneously add 10×10 and 5×5 grids as prediction feature layers, and finally achieve 36.4% *mAP*@.5:.95 and 45.6% AP_L, which are 1.1% and 1.4% higher than the original results. From the experimental results, it can be seen if we add two grids of different scales 10×10 and 5×5 to the YOLOv4 baseline as the prediction feature layers, we can get the highest result in the COCO dataset.

Based on those results on YOLOv4, to save time cost, when we conduct experimental verification on VOC and COCO dataset, we only compare the situation where two scales of grids are added at the same time.

Table 4 Ablation Experiments about the Different Scales of 10×10 and 5×5 on YOLOv4 and YOLOv5. Training Set is VOC07+12, and Test Set is VOC07_test.

Model	Grid Scale	*mAP*@.5	*mAP*@.5:.95
YOLOv4	original	84.3%	60.2%
	only add 10×10	85.0% (+0.7)	61.2% (+1.0)
	only add 5×5	84.8% (+0.5)	60.9% (+0.7)
	add 10×10 and 5×5	**85.6% (+1.3)**	**61.8% (+1.6)**
YOLOv5_S	original	81.9%	56.9%
	add 10×10	83.1% (+1.2)	59.5% (+2.6)
	only add 5×5	82.5% (+0.6)	57.7% (+0.8)
	add 10×10 and 5×5	**84.3% (+2.4)**	**61.3% (+4.4)**

Table 5 Ablation Experiments about the Different Scales of 10×10 and 5×5 on YOLOv4 and YOLOv5, on the COCO dataset. Train Data is Train_2017, Test Data is Test-dev 2017.

Model	Grid Scale	*mAP*@.5:.95	*mAP*@.5:.95 AP_L
YOLOv4	original	35.3%	44.2%
	only add 10×10	35.9% (+0.6)	45.0% (+0.8)
	only add 5×5	35.7% (+0.4)	44.8% (+0.6)
	add 10×10 and 5×5	**36.4% (+1.1)**	**45.6% (+1.4)**
YOLOv5_S	original	36.9%	47.0%
	add 10×10	37.8% (+0.9)	49.8% (+2.8)
	only add 5×5	37.4% (+0.5)	48.5% (+1.5)
	add 10×10 and 5×5	**38.4% (+1.5)**	**52.4% (+5.4)**

(2) Ablation Studies on YOLOv5

We conduct ablation experiments on the YOLOv5_S model. As shown in Table 4, the original YOLOv5_S model with only three feature scales achieves 81.9% *mAP*@.5 and 56.9% *mAP*@.5:.95 on the VOC dataset. Firstly, we propose to add only a 10×10 grid as the prediction feature layer on the YOLOv5 baseline and achieve 83.1% *mAP*@.5 and 59.5% *mAP*@.5:.95, which are 1.2% and 2.6% higher than the original results. Then, we propose to add only a 5 × 5 grid as the prediction feature layer and achieve 82.5% *mAP*@.5 and 57.7% *mAP*@.5:.95, which are 0.6% and 0.8% higher than the original results. Finally, we propose to simultaneously add 10×10 and 5×5 grids as prediction feature layers, and finally aachieve84.3% *mAP*@.5 and 57.7% *mAP*@.5:.95, which are 2.4% and 4.4% higher than the original results. From the experimental results, it can be seen if we add two grids of different scales 10×10 and 5×5 to the YOLOv5_S baseline as the prediction feature layers, we can get the highest result in the VOC dataset.

As shown in Table 5, the original YOLOv5_S model with only three feature scales achieves 36.9% *mAP*@.5:.95 and 47.0% AP_L on the COCO dataset. Firstly, we add only a 10×10 grid as the prediction feature layer on the YOLOv5 baseline and achieve 37.8% *mAP*@.5:.95 and 49.8% AP_L, which are 0.9% and 2.8% higher than the original results. Then, we propose to add only a 5 × 5 grid as the prediction feature layer and achieve 37.4% *mAP*@.5:.95 and 48.5% AP_L, which are 0.5% and 1.5% higher than the original results. Finally, we propose to simultaneously add 10×10 and 5×5 grids as prediction feature layers, and finally aachieve38.4% *mAP*@.5:.95 and 52.4% AP_L, which are 1.5% and 5.4% higher than the original results. From the experimental results, it can be seen if we add two

Table 6 Experiments on the PASCAL VOC dataset of YOLOv3.

Methods	Backbone	Image Size	Train Data	Test Data	*mAP*@.5	*mAP*@.5:.95
YOLOv3_1	Darknet-53	416×416	VOC2007+2012	VOC07_test	76.3%	55.7%
YOLOv3	Darknet-53	416×416	VOC2007+2012	VOC07_test	86.4%	65.9%
Ours	Ours	416×416	VOC2007+2012	VOC07_test	**88.0% (+1.6)**	**67.8% (+1.9)**

Table 7 Experiments on MS COCO dataset of YOLOv3. Train Data is Train_2017, Test Data is test-dev 2017. The YOLOv3 backbone is Darknet-53, our method backbone is a new backbone which we explain in section 3.1. The input image size is 416×416.

Methods	*mAP*@.5	*mAP*@.75	*mAP*@.5:.95	*mAP*@.5:.95 AP_S	*mAP*@.5:.95 AP_M	*mAP*@.5:.95 AP_L
YOLOv3	64.5%	49.5%	45.5%	29.6%	50.3%	57.8%
Ours	**65.4% (+0.9)**	50.2% (+0.7)	**46.3% (+0.8)**	28.8%	50.6% (+0.3)	**61.8% (+4.0)**

Table 8 Experiments on the PASCAL VOC dataset of YOLOv4.

Methods	Backbone	Image Size	Train Data	Test Data	*mAP*@.5	*mAP*@.5:.95
YOLOv4	CSPDarknet53	608×608	VOC2007+2012	VOC07_test	84.3%	60.2%
Ours	CSPDarknet53	608×608	VOC2007+2012	VOC07_test	**85.6% (+1.3)**	**61.8% (+1.6)**

Table 9 Experiments on MS COCO dataset of YOLOv4. Train Data is Train_2017, Test Data is test-dev 2017. The YOLOv4 and our method backbone are CSPDarknet53. The input image size is 608×608.

Methods	*mAP*@.5	*mAP*@.75	*mAP*@.5:.95	*mAP*@.5:.95 AP_S	*mAP*@.5:.95 AP_M	*mAP*@.5:.95 AP_L
YOLOv4	57.0%	37.2%	35.3%	19.1%	43.4%	44.2%
Ours	**57.8% (+0.8)**	37.4% (+0.2)	**36.4% (+1.1)**	19.2% (+0.1)	43.3%	**45.6% (+1.4)**

grids of different scales 10×10 and 5×5 to the YOLOv5_S baseline as the prediction feature layers, we can get the highest result in the COCO dataset.

Based on those results on YOLOv5_S, to save time cost, when we conduct experimental verification on VOC and COCO dataset, we only compare the situation where two scales of grids are added at the same time.

4.4 Experiments on YOLOv3

There are many versions of the YOLOv3 model, and the latest version contains many optimization tricks added by the author, including mosaic enhancement [9], label smoothing [17], cosine annealing learning rate [16], and so on. As shown in Table 6, we use YOLOv3_1 to represent the first model of YOLOv3, and the detection results of the latest version of YOLOv3 are much better than the results of the first version. And the detection results of the latest version of the YOLOv3 model are better than the detection results of the YOLOv4 first model. In this paper, we choose the latest version of YOLOv3 as our baseline model.

We propose to use the YOLOv3 model as the baseline, improve the backbone network, and change the original five residual blocks to six residual blocks. We design to fuse feature on the newly added residual block to retain the effective feature information. A new 7×7 grid is added to the original 52×52, 26×26, and 13×13, which are all used as the prediction feature layers to form a new detection network.

(1) Experiments on the PASCAL VOC dataset

As shown in Table 6, we compare the improved method with the original method. The original YOLOv3 method achieves 86.4% *mAP*@.5 and 65.9% *mAP*@.5:.95 on the VOC dataset. Our proposed method achieves 88.0% *mAP*@.5 and 67.8% *mAP*@.5:.95 on the VOC dataset, which is 1.6% and 1.9% higher than the original.

(2) Experiments on the MS COCO dataset

As shown in Table 7, the original YOLOv3 method achieves 64.5% *mAP*@.5 and 45.5% *mAP*@.5:.95 on the COCO dataset. Our proposed method achieves 65.4% *mAP*@.5 and 46.3% *mAP*@.5:.95 on the COCO dataset, which is 0.9% and 0.8% higher than the original. It is worth noting that the YOLOv3 method achieves 57.8% AP_L and our method achieves 61.8% AP_L, which is 4.0% higher than the original. It shows that our method can not only improve the detection effect of the model but also has the best detection effect on large objects.

4.5 Experiments on YOLOv4

Through the ablation experiments in Sec. 4.3, we obtain that the effect of increasing the grids of both 10×10 and 5×5 scales is the best. Therefore, the methods for the comparative experiments in this section are all models with two grids of 10×10 and 5×5 added at the same time.

We propose to use the YOLOv4 model as a baseline to improve the multi-scale anchor boxes detection method. Based on the original three different scales 76×76, 38×38, 19×19 grids, 10×10, and 5×5 grids are added, which are used as prediction feature layers to form a new detection network.

(1) Experiments on the PASCAL VOC dataset

As shown in Table 8, we compare the improved method with the original method. The original YOLOv4 method achieves 84.3% *mAP*@.5 and 60.2% *mAP*@.5:.95 on the VOC dataset. Our proposed method achieves 85.6% *mAP*@.5 and 61.8% *mAP*@.5:.95 on the VOC dataset, which is 1.3% and 1.6% higher than the original.

Table 10 Experiments on PASCAL VOC dataset of YOLOv5_S-X.

Methods	Backbone	Image Size	Train Data	Test Data	*mAP*@.5	*mAP*@.5:.95
YOLOv5_S	BottleneckCSP	640×640	VOC2007+2012	VOC07_test	81.9%	56.9%
Ours_S	BottleneckCSP	640×640	VOC2007+2012	VOC07_test	**84.3% (+2.4)**	**61.3% (+4.4)**
YOLOv5_M	BottleneckCSP	640×640	VOC2007+2012	VOC07_test	86.0%	65.2%
Ours_M	BottleneckCSP	640×640	VOC2007+2012	VOC07_test	**87.2% (+1.2)**	**66.6% (+1.4)**
YOLOv5_L	BottleneckCSP	640×640	VOC2007+2012	VOC07_test	88.2%	68.3%
Ours_L	BottleneckCSP	640×640	VOC2007+2012	VOC07_test	**89.0% (+0.8)**	**69.9% (+1.6)**
YOLOv5_X	BottleneckCSP	640×640	VOC2007+2012	VOC07_test	88.6%	69.8%
Ours_X	BottleneckCSP	640×640	VOC2007+2012	VOC07_test	**89.1% (+0.5)**	**70.9% (+1.1)**

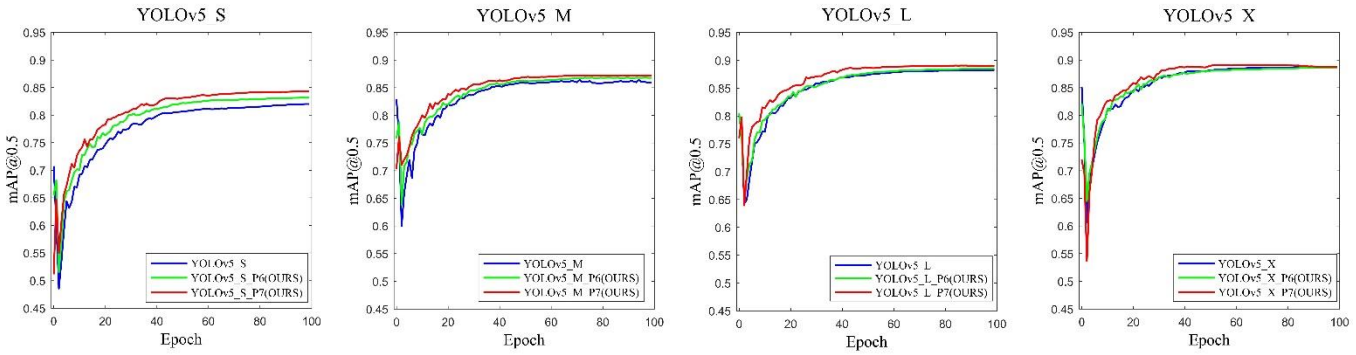


Fig.7 *mAP*@0.5-Epoch curves. From left to right are the *mAP*@0.5-Epoch curve on YOLOv5 S-X on the PASCAL VOC dataset.

(2) Experiments on the MS COCO dataset

As shown in Table 9, the original YOLOv4 method achieves 57.0% *mAP*@.5 and 35.3% *mAP*@.5:.95 on the COCO dataset. Our proposed method achieves 57.8% *mAP*@.5 and 36.4% *mAP*@.5:.95 on the COCO dataset, which is 0.8% and 1.1% higher than the original. It is worth noting that the YOLOv4 method achieves 44.2% AP_L and our method achieves 45.6% AP_L, which is 1.4% higher than the original. It shows that our method can not only improve the detection effect of the model but also has the best detection effect on large objects.

4.6 Experiments on YOLOv5

Through the ablation experiments in Sec. 4.3, we obtain that the effect of increasing the grids of both 10×10 and 5×5 scales is the best. Therefore, the methods for the comparative experiments in this section are all models with two grids of 10×10 and 5×5 added at the same time.

We propose to use the YOLOv5 model as a baseline to improve the multi-scale anchor box detection method. We have done comparative experiments and results in the analysis of the four models of *S, M, L*, and *X* of YOLOv5. Based on the original three different scales 80×80, 40×40, 20×20 grids, 10×10, and 5×5 grids are added by us, which are used as prediction feature layers to form a new detection network.

(1) Experiments on the PASCAL VOC dataset

As shown in Table 10, we compare the improved method with the original method. The original YOLOv5_S-X methods achieve 81.9%, 86.0%, 88.2%, 88.6% *mAP*@.5 and 56.9%, 65.2%, 68.3%, 69.8% *mAP*@.5:.95 on the VOC dataset. Our proposed methods achieve 84.3%, 87.2%, 89.0%, and 89.1% *mAP*@.5 and 61.3%, 66.6%, 69.9%, 70.9% *mAP*@.5:.95 on the VOC dataset. The *mAP*@.5 are higher 2.4%, 1.2%, 0.8%, 0.5%, and the *mAP*@.5:.95 are higher 4.4%, 1.4%, 1.6%, 1.1%. As shown in Fig. 7, it is the graph of the *mAP*@.5 of our methods and the original methods of YOLOv5_S-X trained for 100 epochs. We name the structure that only adds a 10×10 grid as P6 and the structure that adds both 10×10 and 5×5 grids as P7. We can see that in both the P6 structure and the P7 structure, the experimental results are better than the original.

(2) Experiments on the MS COCO dataset

As shown in Table 11, the original YOLOv5_S-X methods achieve 36.9%, 43.8%, 46.1%, 49.5% *mAP*@.5:.95 on the COCO dataset. Our proposed methods achieve 38.4%, 45.0%, 46.8%, 49.8% *mAP*@.5:.95 on the COCO dataset, which are 1.5%, 1.2%, 0.7%, and 0.3% higher than the original. It is worth noting that the YOLOv5_S-X methods achieve 47.0%, 56.1%, 60.2%, 63.2% AP_L and our method achieve 52.4%, 57.4%, 61.5%, 64.8% AP_L, which are 5.4%, 1.3%, 1.3%, 1.6% higher than the original. It shows that our method can not only improve the detection effect of model, but also has the best detection effect on large objects.

There are four groups of detection images that are detected by the Ours_S method on the COCO dataset. As shown in Fig. 8, we divide the images into four groups, including single object images, large images, small images, and dense images. Our method has great detection results.

4.7 Analysis of Experiments Results

From the results of the ablation experiments, we can get that the method of adding grids of different scales as the prediction feature layer is useful to improve the detection effect of the YOLO models. If the two grids of different scales can be added, the best results can be achieved by adding both at the same time.

During the experiment, we also have tried to add a smaller grid as the prediction feature layer, such as adding a 100×100 grid on the YOLOv5_S model, but the detection accuracy achieved by the

Table 11 Experiments on MS COCO dataset of YOLOv5_S-X. Train Data is Train_2017, Test Data is test-dev 2017. The YOLOv5_S-X and our method backbone are BottleneckCSP. The input image size is 640×640.

Methods	*mAP*@.5	*mAP*@.75	*mAP*@.5:.95	*mAP*@.5:.95 AP_S	*mAP*@.5:.95 AP_M	*mAP*@.5:.95 AP_L
YOLOv5_S	55.5%	40.6%	36.9%	21.6%	42.0%	47.0%
Ours_S	56.2% (+0.7)	42.1% (+0.5)	**38.4% (+1.5)**	21.6%	43.0%	**52.4% (+5.4)**
YOLOv5_M	61.1%	48.1%	43.8%	27.5%	49.1%	56.1%
Ours_M	62.4% (+1.3)	48.9% (+0.7)	**45.0% (+1.2)**	27.4%	49.6%	**57.4% (+1.3)**
YOLOv5_L	65.3%	51.7%	46.1%	29.8%	52.3%	60.2%
Ours_L	65.9% (+0.6)	51.9% (+0.2)	**46.8% (+0.7)**	30.1%	52.9%	**61.5% (+1.3)**
YOLOv5_X	67.7%	54.0%	49.5%	32.4%	64.7%	63.2%
Ours_X	67.8% (+0.1)	54.1% (+0.1)	**49.8% (+0.3)**	32.4%	54.8%	**64.8% (+1.6)**

Fig.8 The detection results of the Ours_S method on the COCO dataset.

model is reduced. On the YOLOv3 and YOLOv4 models, we also have tried to add smaller grids as prediction feature layers, such as 100×100, but the achieved results are also lower than the original. We analyze that our method of adding smaller grids as prediction feature layers only greatly increases the complexity of the model, but cannot improve the detection effect of the model. As for smaller grids, we need to extract information from shallower layers to satisfy smaller grids' sizes. However, the semantic information contained in the shallow layer is too limited, and adding a smaller grid will only cause the model to overfit, and cause a decrease of learning speed or detection result.

As shown in Fig. 9, we compare the method of the YOLOv5_S model on the VOC dataset and our method by adding a 100×100 grid as the prediction feature layer. During the training, the method of 100×100 grid is lower than YOLOv5_S. The method of adding 100×100 achieves 75.3% *mAP*@.5, which is 6.6% lower than the original 81.9% *mAP*@.5. Therefore, it can be obtained that the deep layer contains sufficiently semantic information, and the prediction of the deep layer information allocated to grids of different sizes achieves better results than the shallow layer.

From the YOLOv5_S-X experiment results, it can be concluded that both our methods and the originals have good detection results on both datasets. Because the four YOLOv5 models improve the detection effect by increasing the network depth, with the deepening of network depth, some defects of the small network model can be made up, so the detection accuracy is improved. This thought is consistent with our idea of improving the multi-scale anchor box detector. With the deepening of the network structure, our improved methods also can make up for some missing information, and improve the detection results by capturing this missing information. Therefore, with the deepening of the network model, both the original YOLOv5_S-X models and the improved

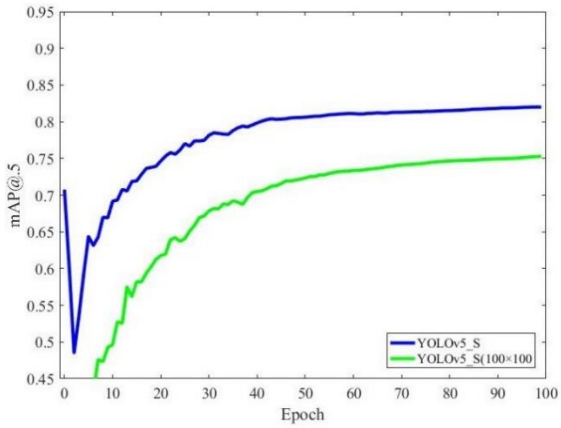


Fig.9 *mAP*@0.5-Epoch curve. The experimental results of adding a 100×100 grid on YOLOv5_S are compared with the original experimental results.

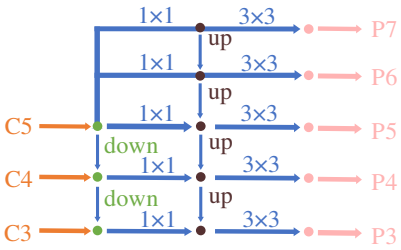


Fig.10 Simplified diagram of the PANet structure on the YOLOv4 and YOLOv5 models.

models proposed by us have the problem of decreasing accuracy growth.

From the network of YOLOv4 and YOLOv5, we can get that the structure of Multiple-in-Multiple-out is also crucial to the PANet structure. Our proposed method on PANet increased the prediction feature layers can be simplified as shown in Fig. 10. Compared with the FPN structure, the PANet structure has one more path for feature fusion. The feature layers of the same scale are obtained by continuous upsampling and downsampling, and the feature layers of the same scale are fused to form the prediction feature layers we need. Increasing the number of predicted feature layers can improve detection results.

We can get from the comparative experimental results on each model, adding a larger scale grid as prediction feature layers, which has a certain improvement on each model. Therefore, we can propose that a multi-scale anchor box detector for the feature is effective for YOLO series models.

5 Conclusions

We introduce the related work, the improved network structure modules, the experimental results, and the analysis detail. The comparative experiments are conducted on each model and the results are analyzed detailly. We propose to add grids of different scales as prediction feature layers on the YOLOv3-5 architectures to improve the detection effect. The semantic information contained in the deep layer is higher than that in the shallow layer, and the effect of assigning the deep layer information to the grid to realize detection is much better than that of assigning the shallow layer information to the grid to realize the detection effect. We propose to add a 7×7 grid as a prediction feature layer on the YOLOv3 baseline, and simultaneously add 10×10 and 5×5 grids as prediction feature layers on the YOLOv4 and YOLOv5 baseline. The method of hierarchical detection structure by adding different scale grids as the prediction feature layer is useful to improve the detection effect on YOLO models.

In the future, we will continue to study the factors that affect the accuracy of object detectors. The influence of the width and depth of the model will be mainly discussed, and a reasonable algorithm is designed to extract the information contained in each layer of the network to achieve the best detection accuracy.

Acknowledgement The authors wish to thank the associate editors and anonymous reviewers for their valuable comments and suggestions on this paper. This work was supported by the National Natural Science Foundation of China (No. 62176034).

References

1. R. Girshick, J. Donahue, and J. Malik, Region-based convolutional networks for accurate object detection and segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(1), 142-158 (2016)
2. R. Girshick, Fast r-cnn, in Proceedings of the IEEE International Conference on Computer Vision, Santiago, pages 1440-1448 (2015).
3. K. He, R. Girshick, S. Ren, and J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, 39(6), 1137-1149 (2017)
4. W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and A.C. Berg, SSD: Single shot multibox detector, in Proceedings of 14th European Conference on Computer Vision, Amsterdam, 9905, 21–37 (2016)
5. J. Redmon, S. K. Divvala, R. Girshick, and A. Farhadi, You only look once: Unified, real-time object detection, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, pages 779–788 (2015)
6. J. Redmon, and A. Farhadi, Yolo9000: Better, faster, stronger, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, pages 6517–6525 (2017)
7. J. Redmon, and A. Farhadi, Yolo v3: An incremental improvement, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, arXiv:1804.02767, (2018)
8. A. Bochkovskiy, C. Wang, and H. Liao, Yolov4: Optimal speed and accuracy of object detection, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, arXiv:2004.10934, (2020)
9. G. Jocher, AyushExel, Borda, alexstoken, and et al. yolov5. https://github.com /ultralytics/ yolov5, (2021)
10. Q. Chen, Y. Wang, T. Yang, X. Zhang, J. Cheng, and J. Sun, YOLOF: You only look one-level feature, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, arXiv: 2103.10934, (2021)
11. G. Zheng, S. Liu, F. Wang, Z. Liu, and J. Sun, YOLOX: Exceeding YOLO series in 2021, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, arXiv: 2017.08430, (2021)
12. S. Ioffe, and C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in Proceedings of the International Conference on Machine Learning, Lille, 37, 448–456 (2015)
13. K. He, X. Zhang, S. Ren, and J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, 37(9), 1904–1916 (2015)
14. S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, Path aggregation network for instance segmentation, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, pages 8759–8768 (2018)
15. C. Wang, A. Bochkovskiy, and H. Liao, Scaled-YOLOv4: Scaling Cross Stage Partial Network, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, arXiv: 2011.08036, (2021)
16. I. Loshchilov, and F. Hutter, SGDR: Stochastic Gradient Descent with Restarts, in Proceedings of 5th International Conference on Learning Representations, Toulon, arXiv: 1608.03983, (2017)
17. Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, Bag of Freebies for Training Object Detection Neural Networks, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, arXiv: 1902.04103, (2019)

18. T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, Feature pyramid networks for object detection, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, pages 936–944, (2017)
19. K. He, X. Zhang, S. Ren, and J. Sun, Deep residual learning for image recognition, in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, pages 770–778, (2016)
20. T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, Focal loss for dense object detection, in Proceedings of the IEEE International Conference on Computer Vision Systems, China, pages 2980–2988, (2017)
21. A. Neubeck, and L.V. Gool, Efficient Non-Maximum Suppression, in Proceedings of the 18th International Conference on Pattern Recognition, Hong Kong, 3, 850-855, (2006)
22. C. Wang, H. Liao, Y. Wu, P. Chen, and J. Hsieh, I. Yeh, CSPNet: A New Backbone that can Enhance Learning Capability of CNN, in Proceedings of 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, pages 1571-1580, (2020)
23. A. Paszke, S Gross, S. Chintala, G. Chanan, E. Yang, Z. Devito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, Automatic differentiation in PyTorch, in Proceedings of 31st Conference on Neural Information Processing Systems, Long Beach, pages 1-4, (2017)
24. M. Everingham and J. Winn, The pascal visual object classes challenge results, IEEE International Journal of Computer Vision, 88(2), 303-338, (2007)
25. T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. Zitnick, Microsoft COCO: Common Objects in Context, in Proceedings of 13th European Conference on Computer Vision, Zurich, 8693, 740-755 (2014)
26. R. Hamid, T. Nathan, G. Young, S. Amir, R. Ian and S. Silvio, Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression, in Proceedings of Conference on Computer Vision and Pattern Recognition, Long Beach, pages 658-666, (2019)
27. T. Wang, L. Yuan, X. Zhang, and J. Feng, Distilling Object Detectors with Fine-Grained Feature Imitation, in Proceedings of 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, pages 4928-4937, (2019)
28. B. Singh, M. Najibi, and L. Davis, SNIPER: Efficient Multi-Scale Training, in Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, pages 9333-9343, (2018)
29. W. Kim, S. Moon, J. Lee, D. Nam, and C. Jung, Multiple player tracking in soccer videos: an adaptive multiscale sampling approach, Multimedia Systems, 24, 611-623 (2018)
30. N. Abid, T. Ouni, A. C. Ammari, and M. Abid, Efficient and high-performance pedestrian detection implementation for intelligent vehicles, Multimedia Systems, 28, 69-84 (2022)

Author biographies

Zhong Qu received the Ph. D degree in computer application technology from Chongqing University in 2009, the M.S degree in computer architecture from Chongqing University in 2003. He is currently a professor at Chongqing University of Posts and Telecommunications. His research interests are in the areas of digital image processing, digital media technology, and cloud computing.

Le-yuan Gao is an M.S degree candidate at Chongqing University of Posts and Telecommunications in the school of software engineering. Her research interests include computer vision, pattern recognition, and deep learning.

Sheng-ye Wang is a Ph. D. degree candidate in the College of Computer Science and Technology of Chongqing University of Posts and Telecommunications. Her main research interests are neural networks and deep learning.