Sensors and Materials, Vol. 32, No. 12 (2020) 4537–4558
MYU Tokyo

S & M 2427

YOLOv3 Object Detection Algorithm with Feature Pyramid Attention for Remote Sensing Images

Zhe Cheng,[1] Jingguo Lv,[1*] Anqi Wu,[1] and Ningning Qu[2]

[1]Beijing University of Civil Engineering and Architecture, Beijing 102616, China
[2]Beijing Field Information Technology Co., Ltd., Beijing 100071, China

(Received September 29, 2020; accepted December 7, 2020)

Keywords: remote sensing image, object detection, spatial pyramid, attention mechanism

In object detection in remote sensing images, owing to the complex background environment, there are problems of poor robustness to interference and low detection accuracy for small objects. The algorithm proposed in this paper combines the attention mechanism with the spatial pyramid structure to improve the You-only-look-once algorithm version 3 (YOLOv3), and it also includes the pyramid attention module to improve the performance of the detection model. The feature pyramid attention module is introduced into deep features, and the feature pyramid attention structure is combined with global context information to better learn object features. The global attention upsampling module is introduced into low-level features, and the global information provided by global pooling is used as a guide to select low-level features. The object detection model can more fully acquire the features of important information and selectively suppress irrelevant features, thereby improving the detection accuracy of the algorithm. To verify the performance of the proposed algorithm, it is used to detect airplanes, storage tanks, ships, baseball diamonds, and running tracks in remote sensing images, and its performance is compared with that of other algorithms. Experiments prove that the proposed algorithm has better detection performance and can improve the detection accuracy of each object in remote sensing images.

1. Introduction

Object detection is an important research problem in the fields of computer vision and image processing, and it has been a research hotspot in theory and application in recent years. It has important application value in both military and civilian fields. Remote sensing technology is widely used in crop monitoring,(1) environmental change and disaster monitoring, resource exploration, and military reconnaissance. Therefore, the application of object detection to the field of remote sensing has important research value. However, remote sensing images are more complex and changeable than natural image scenes and the object scales are different, which generate many challenges in the detection of objects of different scales in remote sensing images. Therefore, the detection of remote sensing objects in complex scenes has high significance in research.(2)

*Corresponding author: e-mail: lvjingguo@bucea.edu.cn
https://doi.org/10.18494/SAM.2020.3130

ISSN 0914-4935

In recent years, with the rapid development of deep learning and the excellent results of the Alex-Net model in the ILAVRC challenge, an increasing number of deep convolutional neural networks are being used in the field of computer vision, introducing new research directions to object detection. The current algorithms are mainly based on candidate-window- and regression-based object detection algorithms. With the development of convolutional neural networks, two-stage algorithms for object detection based on candidate windows have begun to appear. The regions with convolutional features method (RCNN) proposed by Girshick[3] applied the convolutional neural network to object detection for the first time, using a selective search[4] to extract candidate frames, but this method is time-consuming and computationally complicated. In response to these problems, He *et al*. proposed the spatial pyramid pooling networks algorithm (SPPNet),[5] using a convolutional neural network to extract features of the candidate frame, but the training time was increased. The fast region-based convolutional network (Fast-RCNN)(6) was an improvement on these methods that used the visual geometry group network VGG-16[7] as the backbone network to integrate feature extraction, object classification, and position regression into a model, thereby improving the detection speed and accuracy. However, the selective search method adopted in Fast-RCNN cannot guarantee real-time detection. In 2017, Ren *et al*. proposed Faster-RCNN towards real-time object detection with region proposal networks,[8] which used a region proposal network (RPN) to generate candidate regions to truly realize end-to-end training of the object detection network, but it has the problem of inaccurate positioning. In response to this problem, Dai *et al*. proposed a region-based fully convolutional network (R-FCN),[9] which uses a residual network (ResNet) as a feature extraction network to improve the effectiveness of feature extraction and classification. Aiming to solve the problem of inaccurate positioning of the prediction box in Faster-RCNN, Cai and Vasconcelos proposed a cascade structure detector called a cascade region-based convolutional network (Cascade RCNN),[10] which set different intersection over union (IOU) thresholds for training to improve the accuracy of the network prediction box, but the detection speed cannot be guaranteed. Owing to the low speed of object detection methods based on candidate windows, regression-based object detection algorithms began to be proposed. The YOLOv1 algorithm (YOLO = you only look once)[11] treats object detection as a regression problem. To improve the detection accuracy, Redmon and Farhadi proposed YOLOv2[12] based on YOLOv1 using Darknet-19 as the backbone network of YOLOv2, with anchor boxes used to predict bounding boxes to improve detection accuracy. In 2016, Liu *et al*. proposed a multiscale feature fusion method called a single shot multibox detector (SSD),[13] which adds an anchor mechanism to the RPN network, and proposed a similar a priori box method to generate a bounding box for objects. It uses feature maps of different layers for detection, which is more accurate than YOLO but has poor detection accuracy for small objects. Attempts have been made to solve the problems of SSD[14–17] by improving its feature extraction and detection accuracy. In 2018, Redmon and Farhadi proposed YOLOv3,[18] with Darknet-53 as the backbone network, using the idea of feature pyramid networks (FPN)[19] and feature maps of different scales for detection, thus improving the detection of small objects. In 2019, Choi *et al*. proposed an improved network of YOLOv3, Gaussian YOLOv3,[20] which further improved the detection accuracy. In 2020, Bochkovskiy *et al*. proposed YOLOv4.[21] This model uses CSPDarknet-53

with a larger receptive field and more parameters as the backbone network, adds an SPP module to increase the receptive field, and uses a path aggregation network (PANet)[22] for multichannel feature fusion, adding a series of tuning techniques to achieve higher accuracy and speed in real-time target detection algorithms.

Compared with natural images, remote sensing images have more complex backgrounds and more interference, which require higher detection performance of algorithms. Migrating a detection network based on natural images to detect objects in optical remote sensing images is not ideal. Object detection in remote sensing images has two features. One is that the detection performance is poor when the object and the background are similar. The second is that there are many small objects in remote sensing images. Since small objects contain less information, missed detections and false detections are more serious. It is thus necessary to improve the detection of small objects. In response to the above two problems, many researchers have proposed a series of research methods involving the design of a fusion module, adding an attention mechanism, optimizing the algorithm, and improving the performance of the model. Although the above methods introduced feature fusion methods to improve the detection accuracy of small objects, the information between feature layers was not fully utilized in the fusion process, and the amount of calculation was increased. To improve the performance of the detector, the attention mechanism was introduced to solve the problems associated with a complex background, but the improvement in performance was not obvious. In response to the above problems, by combining the advantages of the YOLOv3 algorithm, we add a pyramid attention module to improve the feature extraction capabilities of the network, merge the pyramid capabilities of different scales, enhance the extraction of features of the object, and further improve the detection accuracy of the algorithm for small targets and its robustness to background interference. We use this algorithm to detect airplanes, oil tanks, ships, baseball diamonds, and running tracks in remote sensing images, and compare and analyze its performance with that of other algorithms.

2. Principle of Algorithms

2.1 Principle of YOLO algorithms

The proposed YOLOv1 algorithm converts the object detection problem into a regression problem. Inputting a picture into the detection network can directly return the position coordinates and object category of the object bounding box, so as to achieve end-to-end detection and avoid lengthy processing procedures. The YOLOv1 algorithm first unifies the picture into a size of 448 × 448, and then divides it into $S \times S$ cells, with the center of the object at the center of the grid; this grid is used for predicting the confidence, category, and location of the object. The YOLO algorithm uses GoogleNet as the backbone network, including 24 convolutional layers and two fully connected layers. The convolutional layer extracts features from the image and outputs object category probabilities and coordinates in the fully connected layer.

The YOLOv2 algorithm uses a new network structure, Darknet-19, and is based on YOLOv1. It introduces the anchor mechanism of Faster-RCNN and uses higher resolution images while adding fine-grained features and optimization strategies such as batch standardization and dimensional clustering to improve the speed and accuracy of detection by the algorithm.

The YOLOv3 algorithm, which is based on YOLOv2, has further improved performance. It adopts the Darknet-53 structure with a deeper network layer, and adds a residual module to the network to better extract object features. Owing to the overlap of some categories (such as women and persons), multilabel classification is used instead of Softmax with a logistic classifier. To improve the detection accuracy for small objects, we use an upsampling approach and fusion method on the fusion feature maps of multiple scales. The following is a detailed introduction to the network structure and multiscale detection.

1) Darknet-53

Darknet-53 adopts the idea of the ResNet[23] network and adds residual modules to the network, where 1, 2, 8, 8, and 4 are the numbers of repeated residual modules, and each residual module consists of two convolution layers and a residual layer. The entire network structure has no pooling layer, and the downsampling operation of the network is completed by setting the convolution step size to 2. After this convolution layer, the size of the image is reduced by half. The specific network structure is shown in Table 1.

Table 1
Darknet-53 network structure.

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	416 × 416
	Convolutional	64	3 × 3/2	208 × 208
	Convolutional	32	1 × 1	
1×	Convolutional	64	3 × 3	
	Residual			208 × 208
	Convolutional	128	3 × 3/2	104 × 104
	Convolutional	64	1 × 1	
2×	Convolutional	128	3 × 3	
	Residual			104 × 104
	Convolutional	256	3 × 3/2	52 × 52
	Convolutional	128	1 × 1	
8×	Convolutional	256	3 × 3	
	Residual			52 × 52
	Convolutional	512	3 × 3/2	26 × 26
	Convolutional	256	1 × 1	
8×	Convolutional	512	3 × 3	
	Residual			26 × 26
	Convolutional	1024	3 × 3/2	13 × 13
	Convolutional	512	1 × 1	
4×	Convolutional	1024	3 × 3	
	Residual			13 × 13

2) Multiscale detection

The features learned at the bottom of the network are simple and intuitive, and the geometric contour and position information is rich, which is beneficial for object positioning and small-object detection. The higher the level is, the lesser the geometric detail and position information are, the more abstract and global the learned features are, and the richer the semantic information is, which is suitable for large-object detection and complex-object classification. Therefore, YOLOv3 uses multiscale detection to detect multiple levels of feature maps, as shown in Fig. 1.

As shown in the figure, after the 79th layer, a 32-fold downsampling prediction result is obtained after the convolution operation. The scale size is 13 × 13, the downsampling multiple is high, and the receptive field of the feature map is relatively large, which is suitable for detecting larger objects. The result of the 79th layer is combined with the result of the 61st layer through upsampling, and then the prediction result of 16-fold downsampling is obtained through the convolution operation. The scale size is 26 × 26, with a medium-scale receptive field, which is suitable for detecting medium-scale objects. The result of the 91st layer is upsampled and combined with the result of the 36th layer. After the convolution operation, an 8-fold downsampling result is obtained. The scale size is 52 × 52, and the receptive field is the smallest, which is suitable for detecting small objects.

2.2 Attention mechanism

In essence, the attention mechanism is similar to the human selective visual attention mechanism and is a model that simulates the attention mechanism of the human brain. It can be seen as a combination function, by which the probability distribution of attention is calculated to highlight the impact of a key input on the output. The core goal of the attention mechanism is to select more critical information for the current task goal from a large amount of information and give it a higher weight.

Specifically, as shown in Fig. 2, the attention mechanism model maps an input $X = (x_1, x_2, ..., x_n)$ to an output $Y = (y_1, y_2, ..., y_m)$. In the mechanism model, the encoder transforms an input sequence X into an intermediate semantic $C = f(x_1, x_2, ..., x_n)$ through

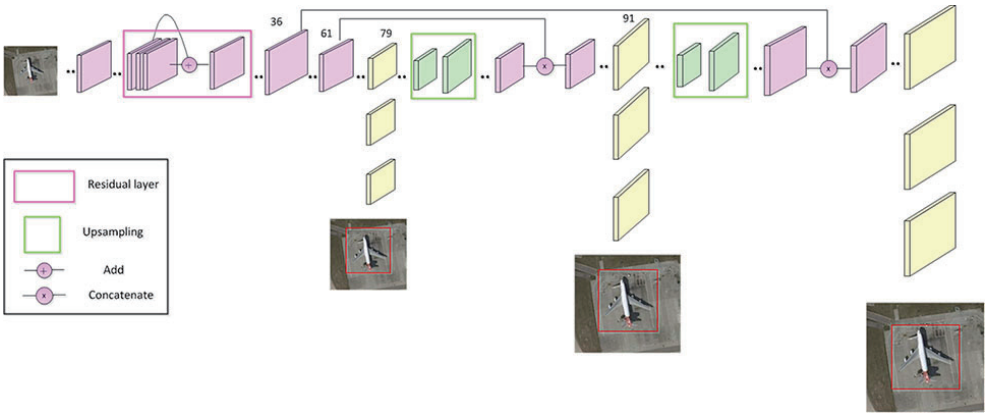


Fig. 1. (Color online) YOLOv3 multiscale detection map.

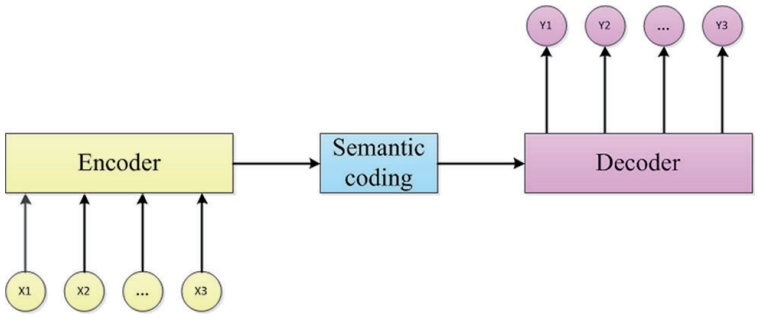


Fig. 2. (Color online) Attention mechanism model.

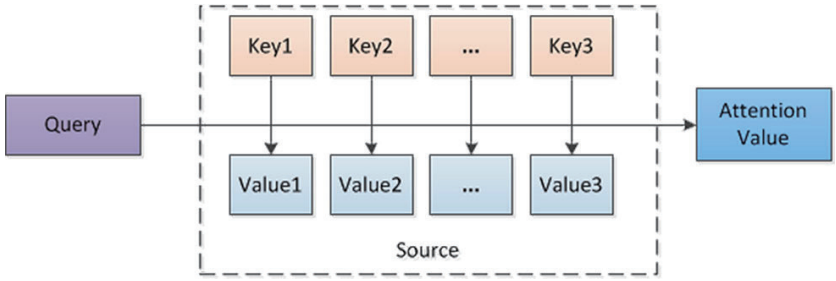


Fig. 3. (Color online) Implementation of attention mechanism.

a nonlinear transformation. The task of the decoder is to predict and generate the output $y_i = g(y_1, y_2, ..., y_{i-1}, C)$ at time i from the intermediate semantic representation C of the input sequence X. $f()$ and $g()$ are both nonlinear transformation functions.

The implementation process is shown in Fig. 3. Imagine the constituent elements in the source as a series of <*Key*, *Value*> data pairs. At this time, given a certain element *Query* in the target, by calculating the similarity or correlation between *Query* and each *Key*, one can obtain the weight coefficient of each *Key* corresponding to each *Value*, and then each *Value* is weighted and summed to obtain the final attention value. In essence, the attention mechanism performs a weighted summation of the *Value* values of the elements in the source, and *Query* and *Key* are used to calculate the weight coefficients of the corresponding *Values*.

The specific realization of the attention mechanism can be expressed by

$$Attention(Query,\ Source) = \sum_{i=1}^{L_x} Similarity(Query, Key_i) \times Value\ , \tag{1}$$

where $L_x = ||Source||$ represents the length of the source, and the meaning of the formula is as described above. Conceptually, attention is still understood as selectively extracting a small amount of important information from a large amount of information and focusing on this important information, ignoring the least important information. The focusing process is reflected in the calculation of the weight coefficient. The larger the weight, the more focus there is on the corresponding *Value*, that is, the weight represents the importance of the information, and *Value* is the corresponding information.

3. Object Detection Algorithm Combined with Attention Mechanism

The YOLOv4 algorithm uses a variety of tuning strategies. Although it performs well in terms of accuracy and speed, the improved network structure is more complex, while the YOLOv3 algorithm structure is relatively simple and flexible and is suitable for remote sensing images with a large amount of data. However, it is more effective for natural image detection, and the background environment of remote sensing images is more complicated, so the network hierarchy of the YOLOv3 algorithm is not applicable. The fusion method adopted by YOLOv3 only indirectly integrates the low-level and high-level semantic information, and misses much of semantic information. In addition, the background of remote sensing images is more complicated and has a greater interference effect. When YOLOv3 detects remote sensing images, if the object is similar to the background, the detection performance is poor. To resolve the above problems, we combine the attention mechanism with the spatial pyramid structure based on YOLOv3 to improve the model's robustness to background interference. The network structure has five main parts: input, backbone network, pyramid attention module, prediction, and output. The network structure is shown in Fig. 4, and each module will be introduced in detail next.

3.1 Pyramid attention module

1) Feature pyramid attention module

This module combines the attention mechanism with the pyramid convolution. The attention mechanism increases the weight of the part with the object information and obtains the output with attention. At the same time, the pyramid convolution structure uses convolution kernels

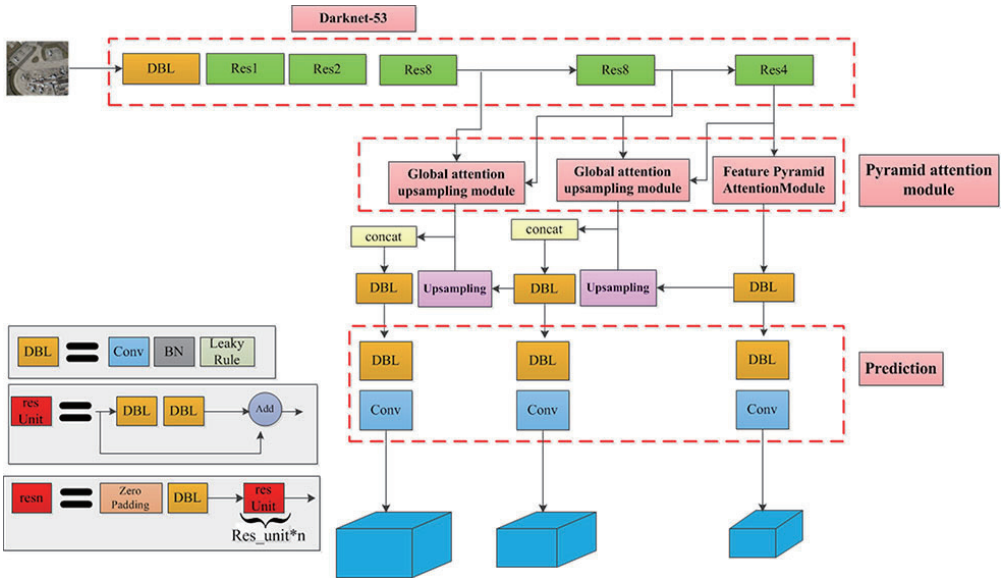


Fig. 4. (Color online) Proposed algorithm network structure.

with different sizes (3 × 3, 5 × 5, and 7 × 7) to represent different receptive fields, which can solve the problem of different objects and different scales. Compared with channel attention, this module has richer pixel-level information. The combination with the pyramid structure produces better pixel-level attention applied to deep-level features and improves the detector's robustness to background interference, thereby improving detection accuracy.

As shown in Fig. 5, after the high-level features are extracted, the pooling operation is no longer performed. Instead, the higher-level semantics are realized through three continuous convolutions. The higher-level semantics will be closer to the real coordinate situation and pay more attention to the object. Therefore, the higher-level semantics is used as a kind of attention guide.

To obtain the output result, the original feature map is subjected to a 1 × 1 convolution operation and linearly superimposed with the operation result of the pyramid feature fusion module. This method strengthens the characteristics of the desired target through the attention mechanism and improves the target's robustness to interference. At the same time, the pyramid convolution structure adopts convolution kernels of different sizes, which represent different receptive fields, realizes multiscale detection, and improves the detection accuracy of small target objects. The high-level feature resolution is small, and the use of a large convolution kernel will not significantly increase the computational burden.

2) Global attention upsampling module

This module can not only more effectively adapt to feature mapping at different scales, but also provide guidance information for low-level feature mapping in a simple way, so as to select more accurate resolution information. In addition, this module uses the extraction of global context information of high-level features to guide the weighting of the information of low-level features. This process also does not significantly add to the computational burden.

As shown in Fig. 6, we use high-level features as a guide and set the corresponding weights so that the weights of the bottom and high levels are consistent, and the high-level features use global pooling to obtain the weights. After multiplying, we add up the bottom layer. In this way, a new high-level integration is carried out while reducing the complexity of the calculation. Specifically, a 3×3 convolution is used for channel processing of low-level features, and then the global pooled information is used for weighting to obtain the weighted low-level features, which are upsampled and then added to the deep-level information.

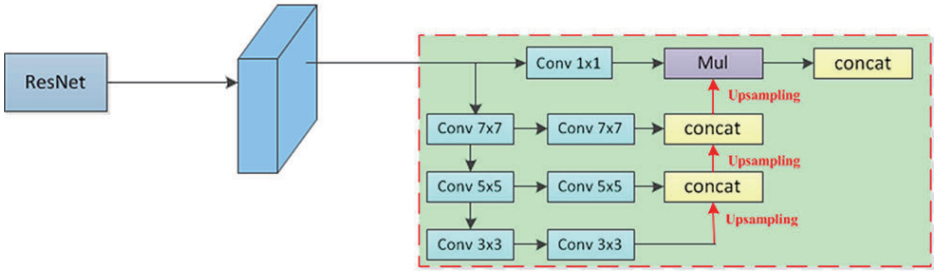


Fig. 5. (Color online) Feature pyramid attention module.

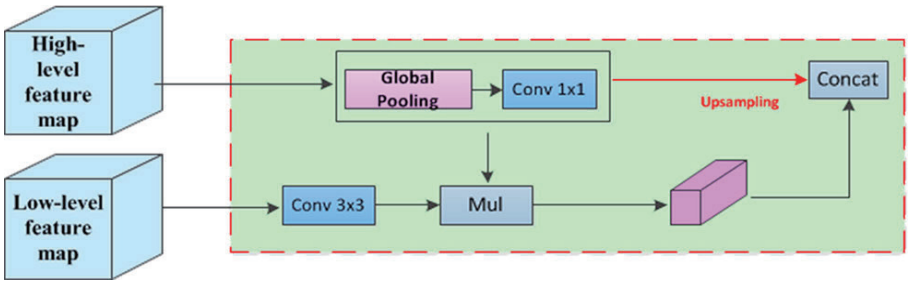


Fig. 6. (Color online) Global attention upsampling module.

To improve the feature extraction capability of the network, the feature pyramid attention module extracts different levels of nonlinear information through the proposed attention mechanism, and the pyramid structure extracts feature information of different sizes and increases the pixel-level receptive field. The global attention upsampling module guides the underlying features and selects more accurate resolution information. The two modules fuse the information extracted from the high- and low-level features to improve the robustness to interference and the detection capability for small objects.

3.2 Training process

1) Design and matching of anchor boxes

For the design of the anchor boxes, K-means clustering is used to obtain the sizes of the anchor boxes. Three types of anchor boxes are set for each scale, and nine sizes of anchor boxes are obtained by clustering. Larger anchor box of 116 × 90, 156 × 198, and 373 × 326 are matched on the smallest 13 × 13 feature map, with which larger objects are detected. Medium-size anchor boxes of 30 × 1, 62 × 45, and 59 × 119 are matched on the medium-size feature map, with which medium-size objects are detected. On the larger 52 × 52 feature map, smaller anchor boxes of 10 × 13, 16 × 30, and 33 × 23 are matched, with which smaller objects are detected. Each cell corresponds to three anchor boxes. The anchor box corresponding to the ground truth box with the largest IOU and its corresponding bounding box are used to predict the object.

2) Prediction mechanism

The direct prediction method is adopted to predict the relative offset value of the center point of the bounding box relative to the upper left corner of the corresponding cell. After learning the offset, the anchor box coordinates originally given by the network can be fine-tuned by linear regression to gradually approach the ground truth and obtain the coordinates of the prediction box. The coordinates can be expressed by

$$\begin{cases} b_x = \sigma(t_x) + c_x \\ b_y = \sigma(t_y) + c_y \\ b_w = p_w e^{t_w} \\ b_h = p_h e^{t_h} \end{cases}, \tag{2}$$

where t_x, t_y, t_w, and t_h are four offsets: t_x and t_y, are the predicted coordinate offsets and t_w and t_h are the scale scaling offsets; c_x and c_y are the coordinates of the upper left corner of the corresponding cell; and b_x, b_y, b_w, and b_h are the coordinates of the predicted value.

3) Loss function

The loss function is used to measure the quality of a set of parameters. The measurement method is used to compare the difference between the network output and the real output. The loss function is mainly used to increase the accuracy of the positioning and the object.

The loss function includes three parts: the bounding box positioning error, confidence error, and classification error. Among them, the bounding box positioning error adopts the complete intersection over union (CIOU) loss, which considers not only the overlap area, but also the center point distance and the aspect ratio. The confidence error and classification error adopt the cross-entropy loss function, whose formula is

$$Loss = L_{box} + L_{cls} + L_{obj}. \tag{3}$$

Here, L_{box} represents the positioning error of the bounding box, which is the difference between the coordinates obtained by the anchor box when predicting the bounding box and the real coordinates. L_{cls} represents the confidence error, which is calculated using the cross-entropy loss, which represents the probability that the target frame contains the target. L_{obj} represents the classification error. When the bounding box determines that there is a target in the current box, the bounding box will calculate the classification loss. The positioning error of the bounding box is

$$L_{box} = 1 - IOU + \frac{d^2}{c^2} + \alpha v, \tag{4}$$

where d and c represent the center points of the prediction box and the ground truth box, respectively. d represents the Euclidean distance between the two center points, and c represents the diagonal distance between the prediction box and the smallest bounding rectangle of the ground truth box. v is a parameter used to measure the consistency of the aspect ratio and α is a parameter used to make trade-offs, which are calculated as follows:

$$v = \frac{4}{\pi}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2, \tag{5}$$

$$\alpha = \frac{v}{(1 - IOU) + v}. \tag{6}$$

Here, w^{gt} and h^{gt} represent the width and height of the ground truth box, w and h represent the width and height of the prediction box, and

$$IOU = \frac{|A \cap B|}{|A \cup B|} = \frac{|I|}{|U|}, \tag{7}$$

where A and B represent the areas of the prediction box and the ground truth box, and I and U represent the intersection area and the union area, respectively. The cross-entropy loss is calculated as

$$H(p, q) = -\sum p(x) \log q(x), \tag{8}$$

where p represents the true value and q represents the predicted value.

Cross-entropy loss is used to evaluate the difference between the current training probability distribution and the true distribution. Reducing the cross-entropy loss improves the prediction accuracy of the model. From the formula of the cross-entropy loss function, the confidence error is

$$\begin{aligned} L_{cls} = & \sum_{i=0}^{s^2} \sum_{j=0}^{B} I_{ij}^{obj} \left[-\hat{c}_i^{\,j} \log(c_i^{\,j}) - (1 - \hat{c}_i^{\,j}) \log(1 - c_i^{\,j}) \right] \\ & + \sum_{i=0}^{s^2} \sum_{j=0}^{B} I_{ij}^{noobj} \left[-\hat{c}_i^{\,j} \log(c_i^{\,j}) - (1 - \hat{c}_i^{\,j}) \log(1 - c_i^{\,j}) \right], \end{aligned} \tag{9}$$

where S^2 represents the number of grids, B represents the number of anchor boxes generated by each grid, I_{ij}^{obj} represents whether the jth anchor box of the ith grid is responsible for predicting the target: if it is responsible, then $I_{ij}^{obj} = 1$, otherwise, it is 0. c_i^j is the probability that the target object is contained in the prediction box. $\hat{c}_i^{\,j}$ represents the ground truth value, and its value is determined on the basis of whether the jth anchor box of the ith grid is responsible for predicting an object: if it is responsible, then $\hat{c}_i^{\,j} = 1$, otherwise, $\hat{c}_i^{\,j} = 0$.

From the formula of the cross-entropy loss function, the classification error obtained is

$$L_{obj} = \sum_{j=0}^{B} I_{ij}^{obj} \left[-\widehat{P_{i,c}}^{\,j} \log(P_{i,c}^{\,j}) - (1 - \widehat{P_{i,c}}^{\,j}) \log(1 - P_{i,c}^{\,j}) \right]. \tag{10}$$

$\widehat{P_{i,c}}^{\,j}$ represents the ground truth value: if it belongs to the cth category, then $\widehat{P_{i,c}}^{\,j} = 1$, otherwise, $\widehat{P_{i,c}}^{\,j} = 0$. $P_{i,c}{}^j$ is the predicted value, which represents the probability that the (i, j)th prediction box belongs to category c.

4. Experiments and Analysis

We validated the proposed algorithm through experiments. For the experiments, under the Ubuntu16.04 operating system, a computer with an NVIDIA GeForce GTX1070Ti GPU graphics card was configured, with CUDA10.0 and CUDNN7.1 installed to accelerate the GPU.

TensorFlow deep learning was configured on the basis of the Anaconda 3.6 frame. We used Darknet-53 as the network framework, selected remote sensing data sets for the experiments, and compared the proposed algorithm with other object detection algorithms. To further verify the detection performance of this algorithm, a test set with more small targets was selected, and the algorithm was compared with Faster-RCNN and the YOLOv3 algorithm. We can obtain better detection results from the detected images. Compared with other algorithms, the detection accuracy of the algorithm in this paper is higher, especially for small target objects. The detection accuracy is above 90% and the highest accuracy is 99%.

4.1 Data sets

The data sets used were NWPUVHR-10, RSOD-Dataset, and DOTA, and we selected a total of 1860 images containing airplanes, ships, storage tanks, baseball diamonds, and running tracks. We labeled the image data, and at the same time converted the data into the format of the VOC data set. Finally, we randomly divided the samples into the training set, validation set, and test set at the ratio of 6:2:2. Targets with an area less than 32 × 32 pixels were considered small targets, those with an area between 32 × 32 and 96 × 96 pixels were considered medium-size targets, and those with an area greater than 96 × 96 pixels were considered large targets. The specific data set distribution is shown in Table 2. We adopted data enhancement, rotation, cropping, and other operations to increase the amount of data.

4.2 Experimental results

We used the proposed algorithm to train and test the data set, and some of the detection results obtained are shown in Figs. 7–11. Figure 7 shows the detection of airplanes and storage

Table 2
Contents of data sets.

Data set	Category	Number of images	Number of targets	Number of targets		
				Small targets	Medium-size targets	Large targets
Training set	Airplanes	280	3021	2271	750	0
	Ships	260	2695	2041	654	0
	Storage tanks	270	3420	2889	531	0
	Baseball diamonds	140	239	2	237	0
	Running tracks	146	157	0	2	155
Validation set	Airplanes	97	223	128	95	0
	Ships	88	246	160	86	0
	Storage tanks	92	312	221	91	0
	Baseball diamonds	49	87	1	86	0
	Running tracks	46	51	0	0	51
Test set	Airplanes	95	241	144	97	0
	Ships	87	239	147	92	0
	Storage tanks	93	298	215	83	0
	Baseball diamonds	50	67	1	66	0
	Running tracks	47	51	0	0	51

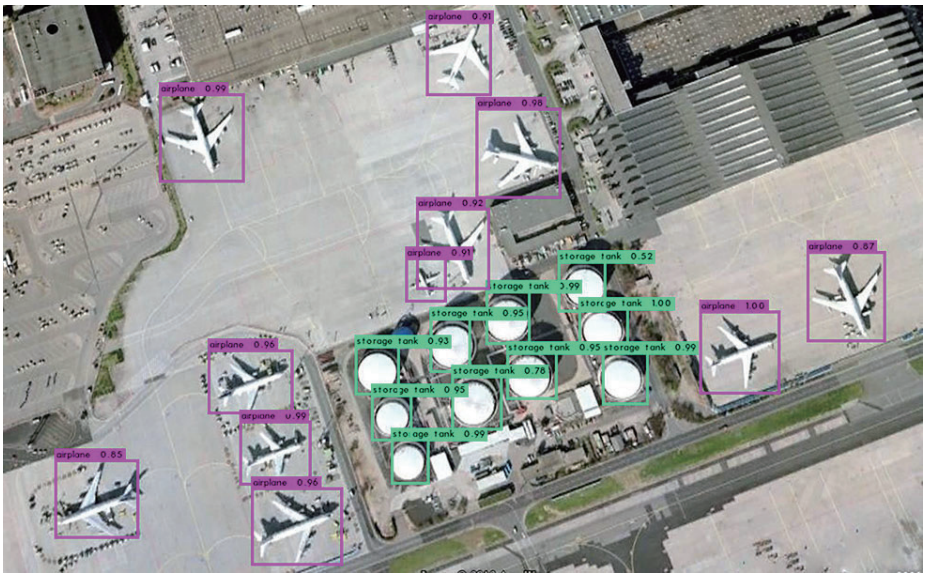


Fig. 7. (Color online) Detection of airplanes and storage tanks.

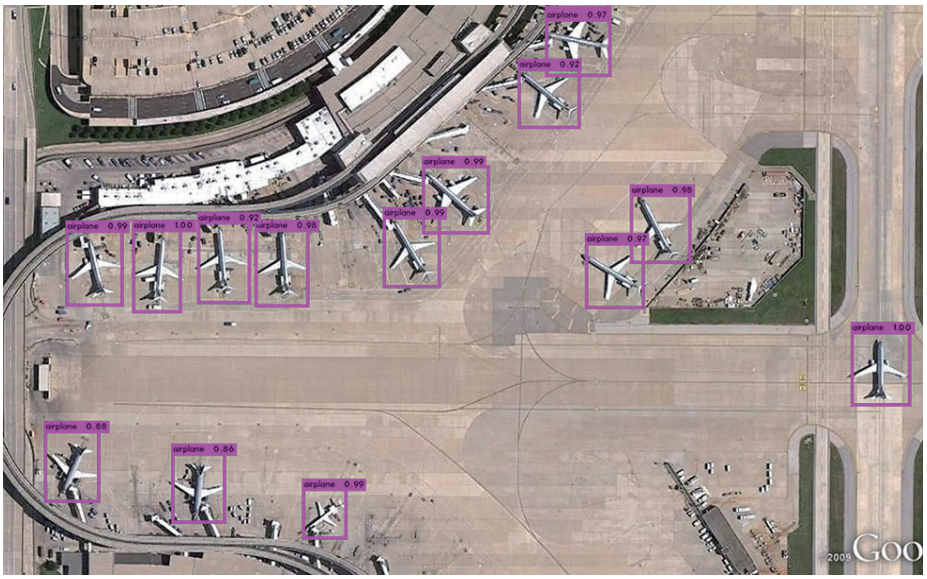


Fig. 8. (Color online) Detection of airplanes.

tanks. The sizes of the airplanes in the picture are different, and the storage tanks are densely arranged. Figure 8 shows the detection of airplanes. In the picture, the distribution of the airplanes is scattered. Figure 9 shows the detection of ships and oil tanks. The storage tanks are arranged very densely and their scale is small. Figure 10 shows the detection of ships, which are small and relatively long and narrow, with some ships having a similar color to the background. Figure 11 shows the detection of running tracks and baseball diamonds, which are large and clear targets.

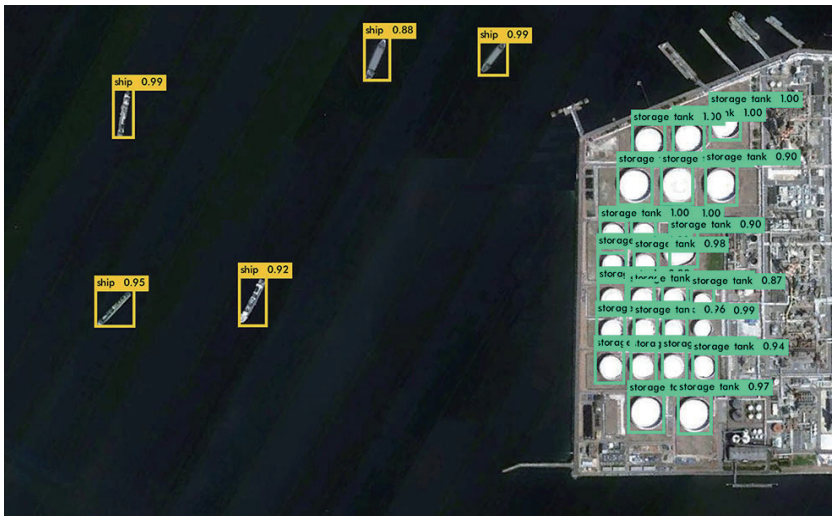


Fig. 9. (Color online) Detection of ships and storage tanks.

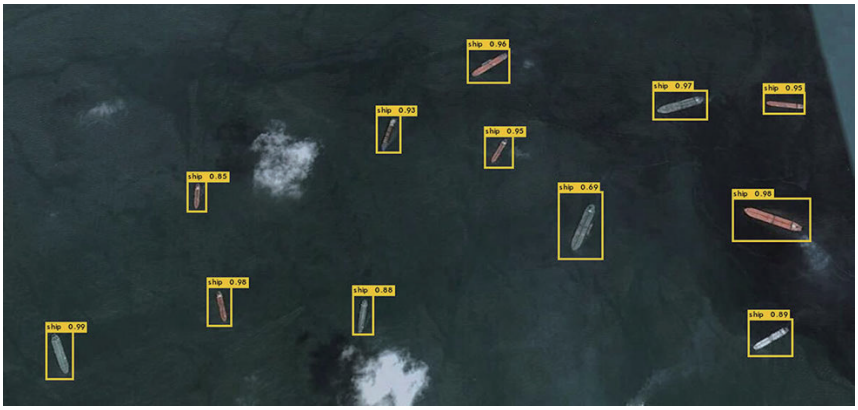


Fig. 10. (Color online) Detection of ships.

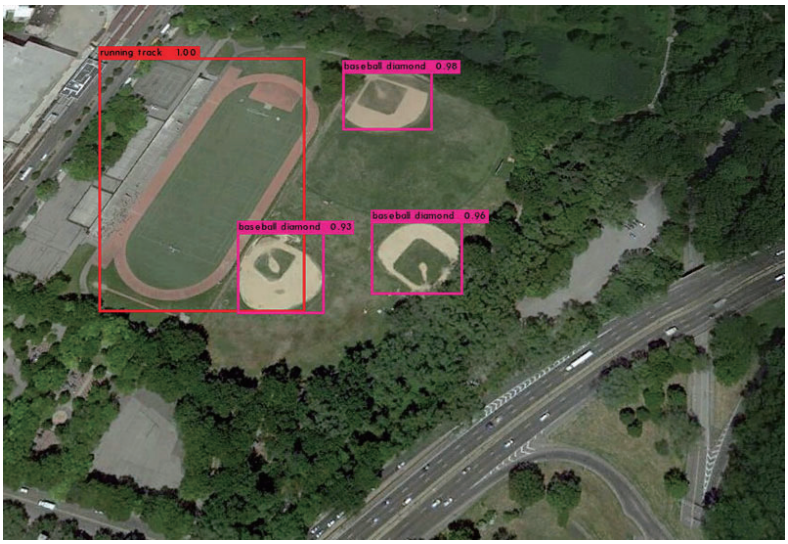


Fig. 11. (Color online) Detection of running tracks and baseball diamonds.

4.3 Accuracy evaluation

The evaluation indicators used are average precision (AP) and mean average precision (mAP). mAP is used to measure the average detection accuracy of multiple types of target. The higher the mAP, the higher the comprehensive performance of the model in all categories. AP and mAP are given by

$$AP = \int_0^1 p(r)dr \,, \tag{11}$$

$$mAP = \frac{1}{N_{cls}} \sum_i AP_i \,. \tag{12}$$

The precision-rate–recall rate (P–R) curves of each category and the mAP are respectively shown in Figs. 12 and 13.

The average accuracy is used to measure the accuracy of the detection algorithm from the two perspectives of recall and accuracy. It is an intuitive standard for evaluating the accuracy of the detection model and can be used to analyze the effectiveness of detecting a single category. The calculation formulas for recall and accuracy are respectively

$$Recall = \frac{TP}{TP + FP}, \tag{13}$$

$$Precision = \frac{TP}{TP + FP} \,. \tag{14}$$

Here, *TP* (true positives) denotes positive samples correctly identified as positive samples, *TN* (true negatives) denotes negative samples correctly identified as negative samples, *FP* (false positives)

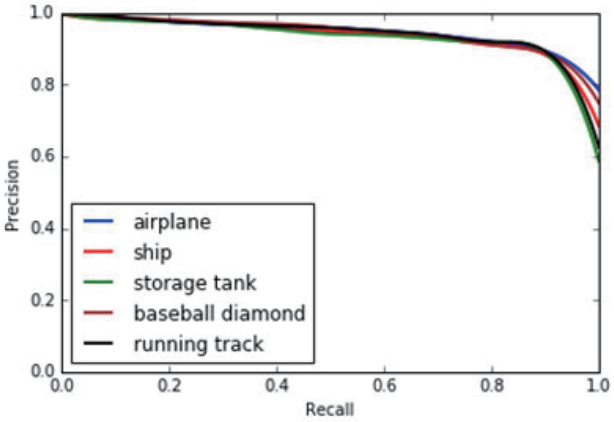


Fig. 12. (Color online) P–R curves of each category.

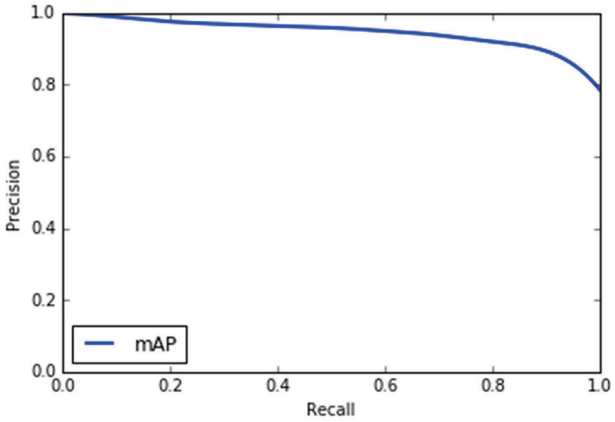


Fig. 13. (Color online) P–R curve of mAP.

denotes negative samples incorrectly identified as positive samples, and *FN* (false negatives) denotes positive samples incorrectly identified as negative samples.

Frames per second (FPS) is used to measure the detection speed of the target detector, which is calculated as

$$FPS = \frac{N_{test}}{T_{time}}, \tag{15}$$

where N_{test} is the number of samples in the test set and T_{time} is the time taken when testing the test set.

The running speed, AP, and mAP are calculated for each category. As shown in Table 3, the proposed algorithm achieves an average accuracy of 94.06% and the detection accuracy for each type of target is above 90%. The lowest detection accuracy is 92.21% and the highest detection accuracy is 96.83%. The lowest detection speed is 28 FPS and the highest speed is 33 FPS, showing the high detection performance of the algorithm.

4.4 Comparative experiment

To demonstrate the superior detection performance of the proposed algorithm, it is compared with other algorithms. The results obtained are shown in Table 4. From the data in the table, it can be seen that the proposed algorithm has the highest detection performance, with 8.76 percentage points higher accuracy than the Faster-RCNN algorithm and a 7 FPS higher speed. Compared with the YOLOv3 algorithm, the accuracy is improved by 1.56% and the speed is increased by 2 FPS.

To further verify that the proposed algorithm has high detection accuracy for small targets and is robust to interference, images with dense small targets and similar backgrounds are

Table 3
Accuracy of test results in various categories.

Category	Speed (FPS)	AP (%)	mAP (%)
Airplanes	29	96.83	94.06
Ships	32	93.36	
Storage tanks	33	92.21	
Baseball diamonds	31	94.35	
Running tracks	28	93.57	

Table 4
Performances of different algorithms.

Algorithm	Framework	Speed (FPS)	mAP (%)
Faster-RCNN	VGG-16	23	85.3
SSD	VGG-16	29	88.6
DSSD	ResNet101	32	90.2
FSSD	VGG-16	26	87.5
YOLOv3	Darknet-53	28	92.5
Proposed	Darknet-53	30	94.06

selected for verification and the performance of the proposed algorithm is compared with those of Faster-RCNN and YOLOv3 for 230 test sets containing 2471 targets. The results are shown in Fig. 14.

It can be seen that the proposed algorithm has the highest rate of correct detection and the lowest rates of false detection and missed detection. To illustrate the stable performance of the algorithm, we give the following examples.

The images in Fig. 15 show the detection results of airplanes. The pink boxes indicate detected airplanes and the blue boxes indicate missed detections. It is found that Faster-RCNN

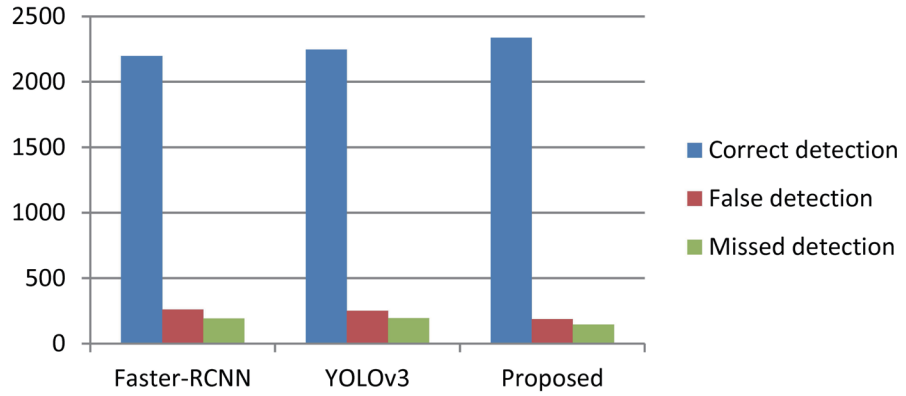


Fig. 14. (Color online) Detection performance results of different algorithms.

(a) (b) (c)

Fig. 15. (Color online) Results of different algorithms. (a) Faster-RCNN, (b) YOLOv3, (c) Proposed.

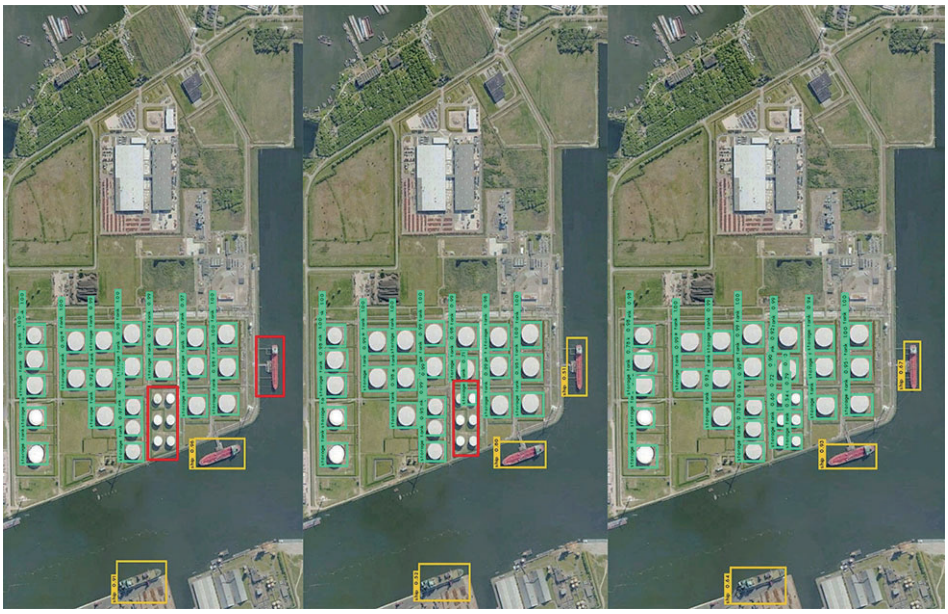
and YOLOv3 missed objects. When detecting a dense arrangement of airplanes, as in Fig. 15, Faster-RCNN and YOLOv3 failed to detect smaller airplanes. The proposed algorithm had no missed detections and showed higher accuracy.

Figure 16 shows the detection results of storage tanks and ships, where the yellow boxes represent the detection results of ships, the green boxes represent the detection results of storage tanks, and the red boxes represent missed detections. For a dense arrangement of storage tanks, it was found that larger storage tanks were correctly detected by all three algorithms. However, Faster-RCNN and YOLOv3 failed to detect some of the smaller tanks. The proposed algorithm had no missed detections and showed higher accuracy. This experiment shows that the detection performance of the proposed algorithm has higher accuracy for small targets.

5. Discussion

5.1 Selection of different levels of feature pyramid attention modules

The feature pyramid attention module integrates a variety of features of different scales through a U-shaped structure, and the pyramid convolution structure uses convolution kernels of different sizes (3 × 3, 5 × 5, 7 × 7, and 9 × 9). In the selection process, after many analyses and experiments, a three-layer convolution operation can more accurately merge the adjacent scale features between the upper and lower layer features, and improve the feature extraction capability of the network. In the experiment, convolutional structures with different layers were constructed, which were named build-1 (3 × 3), build-2 (3 × 3, 5 × 5), build-3 (3 × 3, 5



(a) (b) (c)

Fig. 16. (Color online) Results of different algorithms. (a) Faster-RCNN, (b) YOLOv3, (c) Proposed.

Table 5
Experimental results.

Base model	Improved model	Accuracy (%)	Recall rate (%)	mAP (%)	FPS
YOLOv3	YOLOv3	96.5	82.34	88.57	29
	build-1	97.9	85.72	89.48	28
	build-2	98.1	83.45	90.25	28
	build-3	98.9	87.46	92.13	27
	build-4	97.5	84.51	90.31	26

× 5, 7 × 7), and build-4 (3 × 3, 5 × 5, 7 × 7, 9 × 9), and we conducted experiments on these convolutional structures to find the most suitable model for remote sensing image detection. The experimental results are shown in Table 5.

It can be seen from the experimental data that the best experimental results were obtained when build-3 was added. At the same time, it was found that more convolution kernel layers are not necessarily better. When saturation is reached, the effect of feature extraction is not further improved.

5.2 Choice of loss function

When evaluating the performance of object detection, IOU is used to evaluate the overlap rate of the prediction box and the ground truth box, which reflects the effectiveness of detection. However, IOU only considers the change in the overlapping area, not the change in the non-overlapping area or the change in size. A higher overlap ratio of the obtained cross-to-bin ratio does not mean higher accuracy of the obtained prediction box. This evaluation method reduces the positioning accuracy of the prediction box. Therefore, there will be a large number of overlapping prediction boxes during the detection process, similar to the mutual occlusion of objects in natural images. When encountering densely distributed objects, the overlap phenomenon is more serious, which causes objects to be missed and reduces the detection recall rate. To improve the detection accuracy, the bounding box positioning error in the loss function is changed to the CIOU loss, which considers not only the overlap area, but also the center point distance and aspect ratio. The problem of the large overlap of prediction boxes is thus avoided, and the detection accuracy is improved.

5.3 Limitations

The algorithm in this paper includes the feature pyramid attention module, so that the object detection model can more fully obtain the features of important information and selectively suppress irrelevant features. This improves the detection performance: it not only improves the accuracy of small-object detection, but also alleviates the problem of background interference. However, it is still necessary to improve the real-time performance of the algorithm and further improve the efficiency of processing remote sensing data.

6. Conclusion

Through the analysis of existing object detection algorithms, this paper aims at the problems of high computational complexity and algorithm efficiency of traditional pyramid models. Through information screening, we integrate the attention mechanism with the pyramid model and improve the feature extraction ability of the network on the basis of almost no increase in the amount of calculation, thereby improving the detection accuracy of the algorithm. Specifically, this algorithm combines the attention mechanism with the feature pyramid based on the YOLOv3 algorithm. We add the pyramid attention module, which mainly includes the feature pyramid attention module and the global attention upsampling module. We also introduce the feature pyramid attention module into deep-level features combined with global context information to better learn object features. The global attention upsampling module is introduced into low-level features, and the global information provided by global pooling is used as a guide to select low-level features. Finally, the filtered low-level features and high-level features are combined to improve the detection accuracy of the algorithm model for small objects and the robustness to background interference. To verify the effectiveness of the algorithm, we compared it with other algorithms and demonstrated its superior performance. We also verified its detection accuracy for small objects through the analysis of false detections, missed detections, and the accuracy rate. The proposed algorithm improves the detection accuracy of each object in the remote sensing image, thus improving the detection performance. At the same time, we found that combining the RPN network based on a one-stage algorithm can also play an important role in the research of object detection. We will experiment and analyze it in the follow-up work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (grant no. 41871367), the Ministry of Science and Technology of the People's Republic of China (grant no. 2018YFE0206100), and the BUCEA Postgraduate Innovation Project (PG2020086).

References

1 B. F. Wu, F. Zhang, C. L. Liu, L. Zhang, and Z. M. Luo: Int. J. Remote Sens. **6** (2004) 498. https://doi.org/10.3321/j.issn:1007-4619.2004.06.002
2 J. E. Ball, D. T. Anderson, and C. S. Chan: Remote Sens. **11** (2017) 1. https://doi.org/10.1117/1.jrs.11.042609
3 R. Girshick, J. Donahue, T. Darrell, and J. Malik: Proc. 2014 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE, 2014) 580–587. https://doi.org/ 10.1109/CVPR.2014.81
4 C. Gu, J. J. Lim, P. Arbeláez, and J. Malik: Proc. 2009 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE, 2009) 1030–1037. https://doi.org/10.1109/CVPR.2009.5206727
5 K. He, X. Zhang, S. Ren, and J. Sun: Lect. Notes Comput. **8691** (2014) 346. https://doi.org/10.1007/978-3-319-10578-923
6 R. Girshick: Proc. IEEE 2015 Int. Conf. Computer Vision (IEEE, 2015) 1440–1448. https://doi.org/10.1109/ICCV.2015.169
7 K. Simonyan and A. Zisserman: Proc. 3rd Int. Conf. Learning Representations (ICLR, 2015) 1–14.
8 S. Ren, K. He, R. Girshick, and J. Sun: IEEE Trans. Pattern Anal. Mach. Intell. **39** (2017) 1137. https://doi.org/10.1109/TPAMI.2016.2577031

9 J. Dai, Y. Li, K. He, and J. Sun: Neural Inf. Process. Syst. (2016) 379. https://doi.org/10.2175/106143009X1248095237035
10 Z. Cai and N. Vasconcelos: Proc. 2018 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE, 2018) 6154–6162. https://doi.org/10.1109/TPAMI.2016.257703110.1109/CVPR.2018.00644
11 J. Redmon, S. Divvala, R. Girshick, and A. Farhadi: Proc. 2016 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE, 2016) 779–788. https://doi.org/10.1109/CVPR.2016.91
12 J. Redmon and A. Farhadi: Proc. 2017 IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE, 2017) 6517–6525. https://doi.org/10.1109/CVPR.2017.690
13 W. Liu, A. Dragomir, E. Dumitru, S Christian, R Scott, Y. F. Cheng, and C. B. Alexander: Lect. Notes Comput. **9905** (2016) 21. https://doi.org/10.1007/978-3-319-46448-0_2
14 J. Dai, Y. Li, K. He, and J. Sun: Neural Inf. Process. Syst. (2016) 379.
15 C. Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg: DSSD: Deconvolutional Single Shot Detector (2017). http://arxiv.org/abs/1701.06659
16 Z. Shen, Z. Liu, J. Li, Y. G. Jiang, Y. Chen, and X. Xue: Proc. 2017 IEEE Int. Conf. Computer Vision (IEEE, 2017) 1937–1945. https://doi.org/ 10.1109/ICCV.2017.212
17 Z. Li and F. Zhou: FSSD: Feature Fusion Single Shot Multibox Detector (2017). http://arxiv.org/abs/1712.00960
18 J. Redmon and A. Farhadi: YOLOv3: An Incremental Improvement (2018). http://arxiv.org/abs/1804.02767
19 X. Li, T. Lai, S. Wang, Q. Chen, C. Yang, and R. Chen: Proc. 2019 IEEE Int. Conf. Parallel Distributed Processing with Applications (2019) 1500–1504. https://doi.org/10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00217
20 J. Choi, D. Chun, H. Kim, and H. J. Lee: Proc. 2019 IEEE Int. Conf. Computer Vision (IEEE, 2019) 502–511. https://doi.org/ 10.1109/ICCV.2019.00059
21 A. Bochkovskiy, C.-Y. Wang, and H. Y. M. Liao: YOLOv4: Optimal Speed and Accuracy of Object Detection (2020). http://arxiv.org/abs/2004.10934
22 S. Ioffe and C. Szegedy: Proc. 32nd Int. Conf. Machine Learning (IEEE, 2015) 448–456.
23 K. He, X. Zhang, S. Ren, and J. Sun: Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (IEEE, 2019) 770–778. https://doi.org/10.1109/CVPR.2016.90

About the Authors

Zhe Cheng received her B.E. degree in surveying and mapping engineering from Shandong Agricultural and Engineering University in 2018. She is currently pursuing her master's degree at Beijing University of Civil Engineering and Architecture. Her current research interests include photogrammetry, remote sensing, target detection, and target tracking. (84638021@qq.com)

Jingguo Lv received his M.E. degree in photography and remote sensing from Wuhan University in 2003 and his Ph.D. degree in charting and geographic information systems from Beijing Normal University in 2009. Since 2009, he has been teaching at Beijing University of Civil Engineering and Architecture, where he is as an associate professor. His research interests include remote sensing information extraction, digital image processing, and visual tracking. (lvjingguo@bucea.edu.cn)

Anqi Wu started her bachelor's degree at Beijing University of Civil Engineering and Architecture in 2017. Her major is photogrammetry and remote sensing. (wuanqi@stu.bucea.edu.cn)

Ningning Qu is a senior engineer who has long been engaged in the space information industry and has rich experience in remote sensing technology applications and software development. He has worked as a technical manager, product manager, and other positions in domestic first-line remote sensing industry companies such as Aerospace Star Atlas and Aerospace Hongtu. (quningning@fieldinfo.cn)