S & M 3718

# Improved YOLOv5 Algorithm for Oriented Object Detection of Aerial Image

Gang Yang,[1,2] Miao Wang,[1*] Quan Zhou,[1]
Jiangchuan Li,[1,2] Siyue Zhou,[1] and Yutong Lu[1]

[1]Beijing Institution of Surveying and Mapping, No. 60 Nanlishi Road, Beijing 100045, China
[2]Beijing Key Laboratory of Urban Spatial Information Engineering,
No. 60 Nanlishi Road, Beijing 100045, China

(Received May 7, 2024; accepted July 4, 2024)

***Keywords:*** object detection, remote sensing image, deep learning, YOLO

With the development of computer vision and remote sensor devices, object detection in aerial images has drawn considerable attention because of its ability to provide a wide field of view and a large amount of information. Despite this, object detection in aerial images is a challenging task owing to densely packed objects, oriented diversity, and complex background. In this study, we optimized three aspects of the YOLOv5 algorithm to detect arbitrary oriented objects in remote sensing images, including head structure, features from the backbone, and angle prediction. To improve the head structure, we decoupled it into four submodules, which are used for object localization, foreground, category, and oriented angle classification. To increase the accuracy of the features from the backbone, we designed a block dimensional attention module, which is developed by splitting the image into smaller patches based on a dimensional attention module. Compared with the original YOLOv5 algorithm, our approach has a better performance for oriented object detection—the mAP on DOTA-v1.5 is increased by 1.25%. It was tested to be effective on DOTA-v1.0, HRSC2016, and DIOR-R datasets as well.

## 1. Introduction

With the rapid expansion of artificial satellites and unmanned aerial vehicles, high-resolution remote sensing images are easily available and widely introduced to smart city systems,(1,2) environment monitoring,(3) intelligent traffic system,(4–7) and other technical fields. Generally speaking, remote sensing images contain a great deal of information and a large field of view, as they are captured from a high altitude and vertical perspective.(8) Therefore, how to automatically detect and identify the objects in these images has become a hot topic in current research.

Currently, many high-performance detectors have been proposed, which can generally be divided into two types: two-stage and one-stage object detectors. Two-stage detectors have achieved promising results on various benchmarks, whereas one-stage detectors maintain a faster detection speed.

[*]Corresponding author: e-mail: wangmiao@bism.cn

https://doi.org/10.18494/SAM5136

ISSN 0914-4935 © MYU K.K.
https://myukk.org/

General horizontal detectors have fundamental limitations in many practical applications of remote sensing object detection owing to their various scales, high density, and arbitrary orientation. Therefore, many oriented detectors are designed to address these issues.

As discussed above, we designed a one-stage method based on YOLOv5 to improve the performance of oriented object detection in remote sensing images. This method has both high detection accuracy and real-time performance. Extensively, we evaluated the proposed approach on two public datasets, namely, DOTA[9] and HRSC2016.[10] In summary, our contributions are as follows.

- We decoupled the detection head of YOLOv5 to alleviate the conflict between classification and regression.
- We designed a dimensional attention module (DAM) to make better use of the relationship between channel and spatial information.
- On the basis of the characteristics of remote sensing images, we developed a block dimensional attention module (BDAM) based on the DAM by block processing.
- We evaluated the state-of-the-art performance on two public datasets. Experiments showed that the strategies we proposed can be easily embedded into any detection framework with significant performance.

## 2. Related Work

In recent years, deep learning has subverted the field of traditional object detection by its own efforts.[11] In particular, the convolutional neural network (CNN) has made a great deal of achievement and has emerged as a powerful strategy for learning feature representations directly from data. The existing deep detection framework can be summarized into two categories: one- and two-stage frameworks. The one-stage framework directly predicts the object bounding box and label category including the You Only Look Once (YOLO) series,[12–15] single shot detector,[16] RetinaNet,[17] and SqueezeDet.[18] The two-stage framework consists of two steps. The first step is to roughly generate proposal bounding boxes using the region proposal network (RPN). The second step is to predict object category and fine-tune the bounding boxes. The two-stage framework is presented by using such methods as Mask R-CNN,[19] Fast R-CNN,[20] and Faster R-CNN.[21] Compared with the one-stage framework, most methods based on the two-stage framework have better detection accuracy, whereas the one-stage framework has better real-time performance because of its simple network structure.

Unlike nature images, objects in remote sensing images have characteristics of scale variance, high density, oriented diversity, and complex background, all of which make it of great difficulty to locate and identify objects in remote sensing images. Recently, numerous approaches have been proposed as solutions to these issues. Li *et al.*[22] proposed the object-wise semantic representation (OWSR), which combines the enhanced feature pyramid network (eFPN) and semantic segmentation module to deal with the issue of scale variance. Cheng *et al.*[23] proposed the rotation-invariant CNN (RICNN) model to address the oriented diversity of objects in remote sensing images. To alleviate the disturbances of complex background, Ding *et al.*[24] designed the region of interest (RoI) transformer to generate the rotated RoI (RRoI). As inspired by the RoI transformer, ReDet(25) was designed to improve the robustness of features for both oriented diversity and complex ground by obtaining the rotation-invariant features via the rotation-invariant network based on e2cnn(26) and the RiRoI align module. There are a large number of small objects in remote sensing images, which may pose challenges to deep detectors, because of their deficiency of information. To improve small object detection, Wu *et al.*(8) designed CDD-Net, whose local context feature network (LCFN) can cause the fusion of objects and their neighbor information, significantly increasing the amount of information about small objects. All the methods mentioned above are based on the two-stage framework, and they are always limited by their speed. Therefore, we tend to use one-stage framework methods when facing situations with high real-time demands. For example, Qu *et al.*(27) employed the convolutional block attention module (CBAM)(28) and adaptive feature fusion (AFFF) module(16) to YOLOv3 for the improvement of features in remote sensing images. R3Det(29) introduced the regression of oriented object angle to RetinaNet for oriented object detection. Inspired by CornerNet, Chen *et al.*(30) proposed an anchor-free method for oriented bounding box (OBB) detection by searching corners of objects in remote sensing images. Oriented R-CNN(31) is designed on the basis of proposal-based oriented object detection methods, designing a high-efficiency oriented RPN to break the computational bottleneck for generating oriented proposals.

At present, many algorithms based on the YOLO series are being proposed for object detection in remote sensing images, and they have outstanding accuracy and speed. Xu and Wu(32) employed the combination of YOLOv3(13) and DenseNet,(33) where DenseNet was adopted as the feature extractor and YOLOv3 was adopted as the main detection architecture. Considering the characteristics of remote sensing images, DenseNet has a better ability to extract the feature of tiny objects. However, its structure is too complex, so the speed is degraded. Cao *et al.*(34) added the pyramid pooling module (PPM)(35) based on YOLOv4(14) and replaced the Mish function with the original activation function, which improved the accuracy and recall rate of aircraft and dockyard in remote sensing images. Wan *et al.*(36) applied multiple layers of the feature pyramid, a multi-detection-head strategy, and a hybrid attention module based on YOLOv5 to improve the effect of the object detection network used in optical remote sensing images. Liu *et al.*(37) introduced the coordinate attention mechanism into YOLOv5 to enhance the feature and location information extraction ability of the shallow layer of the model and optimize the feature extraction ability of the model for different scale targets. The CSL(38) method simplifies the oriented angle prediction as a classification problem, handling the oriented angle in the training process with consideration of its periodicity and continuity.

In contrast with the methods above, we employed the channel and spatial contextual features in different regions of feature maps, thereby enabling our detector to obtain more discriminative feature representation. Extensively, the head structure is decoupled into four independent subnetworks for different prediction tasks.

## 3. Methodology

Our approach is based on YOLOv5 with PAFPN(39) and has a backbone of CSPDatknet.(14) Figure 1 shows the overview of the proposed improved YOLOv5, where the improved modules

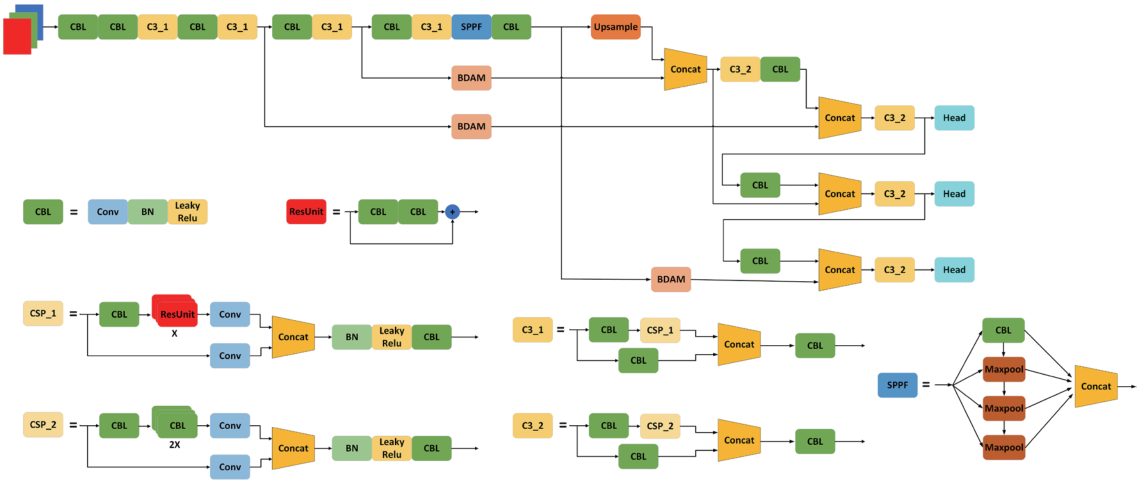


Fig. 1. (Color online) Overview of improved YOLOv5.

are marked with red. The structure of the improved YOLOv5 consists of three primary components: the backbone, neck, and heads. The backbone is used for extracting high-quality features using its CBL, C3, and SPPF modules. The neck, including both bottom-up and top-down connections, can fuse features from the backbone of different scales. The regression and classification tasks are finished in heads using fused features from the neck.

In view of the various directions of oriented detection in remote sensing images, OBB is a better choice for fitting object contours and reducing the overlap of bounding boxes in the dense area. In this research, OBB prediction is decoupled as horizontal bounding box (HBB) regression and oriented angle classification tasks. HBB regression consists of the regression tasks of the center position and the long and short sides of the bounding box. The oriented angle is defined as the angle between the long side and the horizontal line.

### 3.1 Decoupled head

In the original YOLOv5 algorithm, all regressions and classifications were completed in a head with a single convolutional layer. Therefore, conflicts between regressions and classifications may be generated in the learning process of our detector. Obviously, it is unreasonable to make use of identical heads for different tasks.

Following other proposed approaches,(15,17,40,41) the decoupled head is designed for reducing these conflicts. Figure 2 shows the structure of the decoupled head. The fully convolutional networks, containing 1 × 1 and 3 × 3 convolutional layers, are designed for object, category, and angle classification, and bounding box regression respectively.

### 3.2 Dimensional attention module

Each category of objects in remote sensing images usually has a specific foreground and background in terms of its special imaging pattern. For example, an airplane is always associated

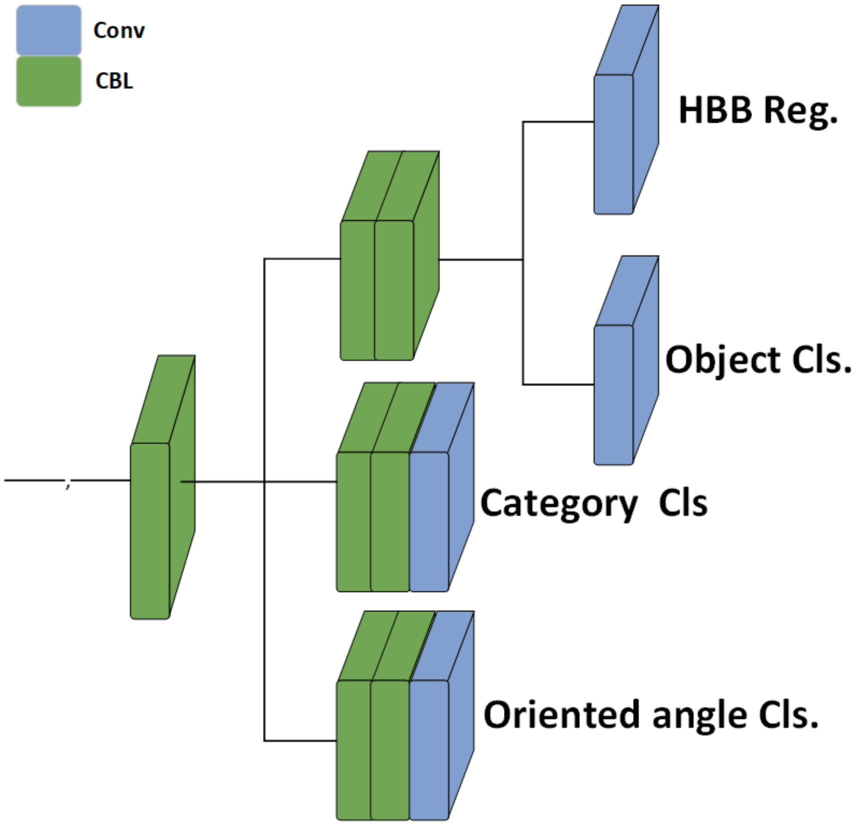


Fig. 2. (Color online) Structure of decoupled head.

with an airport, not a harbor or road. Swimming pools are usually blue rather than anything else. Both the spatial and channel contextual information play an important role in distinguishing between different object categories.

The DAM is discovered to obtain the spatial and channel contextual information, which can tell the detector where to focus to promote the representation of interest in remote sensing images. DAM consists of two cascaded submodules: a channel attention module (CAM) and a spatial attention module (SAM). The structure of DAM is shown in Fig. 3.

CAM focuses on meaningful content in each given image. Inspired by CBAM,(28) we use the average and max pooling layers to improve the efficiency of channel attention computation. The average pooling layer can extract the spatial feature, while the max pooling layer can obtain unique object information. In this regard, CAM can obtain more refined channel attention weights using combined features from average and max pooling layers. The overall process of CAM is summarized as follows:

$$f_{CAM}(\boldsymbol{F}) = \sigma\Big(f_c\Big(f_{cbl}\big(f_{max}(\boldsymbol{F})\big) + f_{cbl}\big(f_{avg}(\boldsymbol{F})\big)\Big)\Big), \tag{1}$$

where $\boldsymbol{F}$ denotes input feature maps. $f_{max}$, $f_{avg}$, $f_{cbl}$, and $f_c$ denote max pooling, average pooling, CBL, and convolutional processing, respectively. $\sigma$ is defined as sigmoid activation.

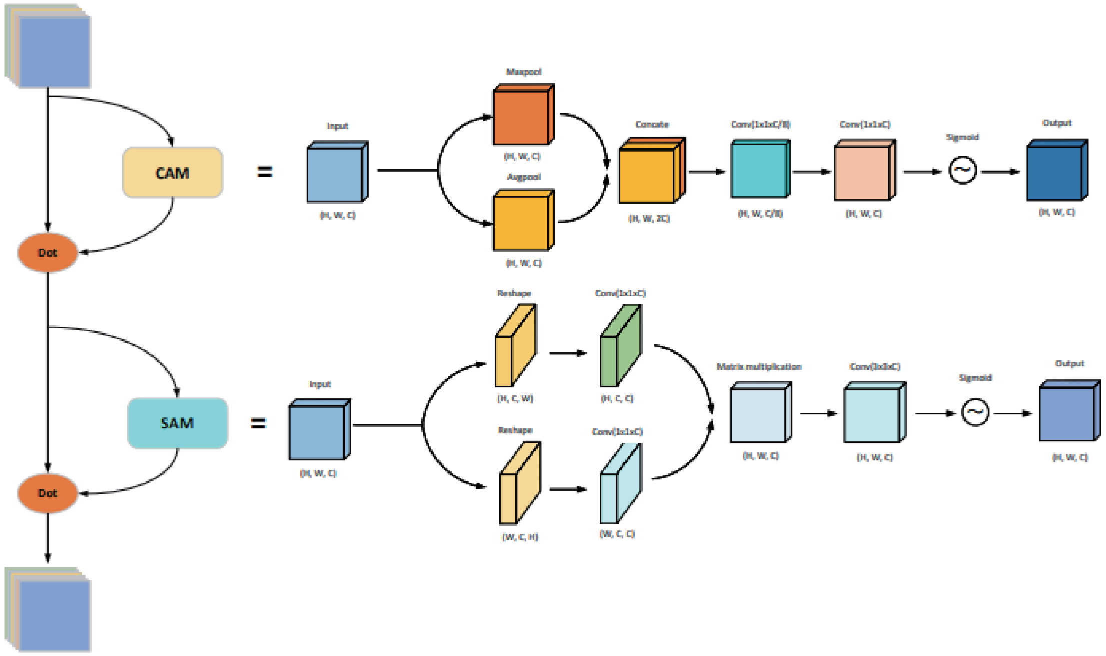


Fig. 3. (Color online) Structure of DAM.

Unlike CAM, SAM can tell the detector where the important area is in an input image. For the directional characteristics of objects in remote sensing images, *SAM* involves two submodules: a height attention module (*HAM*) and a width attention module (*WAM*), which can collect the global features of vertical and horizontal dimensions. The computation of *SAM* is defined as follows:

$$f_{HAM}\left(\boldsymbol{F}\right)=f_{cbl}\left(R_{h,c}\left(\boldsymbol{F}\right)\right),$$

$$f_{WAM}\left(\boldsymbol{F}\right)=f_{cbl}\left(R_{w,c}\left(\boldsymbol{F}\right)\right), \tag{2}$$

$$f_{SAM}\left(\boldsymbol{F}\right)=\sigma\left(f_{HAM}\left(\boldsymbol{F}\right)+f_{WAM}\left(\boldsymbol{F}\right)\right),$$

where $R_{h,c}$ and $R_{w,c}$ denote reshape operations between height, width, and channel, respectively. $f_{HAM}(\boldsymbol{F})$ and $f_{WAM}(\boldsymbol{F})$ denote the computing processes of *HAM* and *WAM*, respectively. $f_{SAM}$ is the overall function of *SAM*.

The output of DAM is treated as a weight to determine how much attention needs to be paid in different areas for a given image. Therefore, the whole process of DAM is defined as

$$f_{DAM}\left(\boldsymbol{F}\right)=\boldsymbol{F}\times f_{CAM}\left(\boldsymbol{F}\right)\times f_{SAM}\left(\boldsymbol{F}\right). \tag{3}$$

### 3.3 Blocked dimensional attention module

The difference between remote sensing and ordinary images is that remote sensing images usually have a larger field of view. As the distance between two pixels in a remote sensing image increases, their relationship will become faded in practice. The dependences of pixels are more important in nearby regions than in distant regions. In this regard, calculating the relationship between pixels that are far apart from each other is unnecessary and unreasonable. If we only compute the global attention of a whole remote sensing image, the relationship that should not exist between two pixels may be wrongly received by our detector, so that the detection accuracy is negatively affected.

To solve or alleviate the problem above, block processing is selected to establish the pixel dependency relationships in this study. As shown in Fig. 4, the input feature map is separated into patches of $N \times N$, where we set $N$ to be 8. All separated patches are processed by DAM and reorganized to give the subsequent network. By adding BDAM to a small local area of the feature maps, in this study, we focused on the calculation of the correlation between the pixels of neighboring areas. On the one hand, it is more in line with the characteristics of the remote sensing images; on the other hand, it can also improve the computing efficiency.

## 4. Experiment

### 4.1 Dataset

Three public datasets, DOTA,(9) DIOR-R,(42) and HRSC2016,(10) are used for the evaluation of our method. To prevent from overfitting, we employ various data augmentation strategies including Mosaic,(14) Mixup,(43) and rotated augmentation.

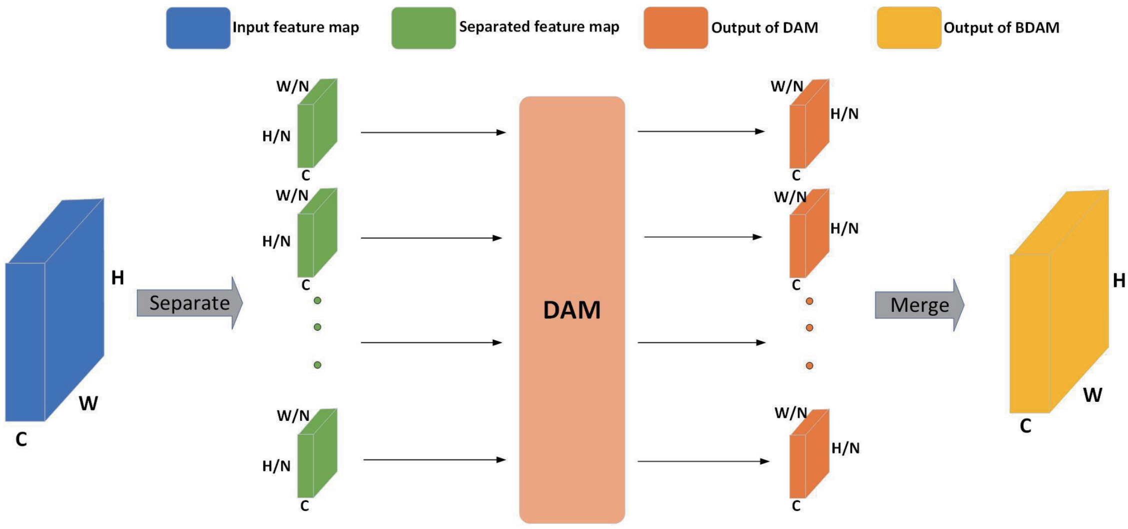


Fig. 4. (Color online) Structure of BDAM.

DOTA has two different versions, DOTA-v1.0 and DOTA-v1.5, both of which are based on the same images from Google Earth, GF-2, and JL-1 satellites. DOTA-v1.0 has 15 categories: plane (PL), baseball diamond (BD), bridge (BR), ground track field (GTF), small vehicle (SV), large vehicle (LV), ship (SH), tennis court (TC), basketball court (BC), storage tank (ST), soccer-ball field (SBF), roundabout (RA), harbor (HA), swimming pool (SP), and helicopter (HC). Improved from DOTA-v1.0, the container crane (CC) class joins the categories of DOTA-v1.5. Objects smaller than 10 pixels are annotated exhaustively by DOTA-v1.5. Thus, DOTA-v1.5 is a more challenging dataset for oriented detection in remote sensing images.

The DIOR-R dataset, extended from the DIOR dataset, is a large-scale publicly available oriented object detection dataset. It contains 23463 images and 192518 instances. Twenty object categories are annotated in the dataset: airplane (APL), airport (APO), baseball field (BF), basketball court (BC), bridge (BR), chimney (CH), dam (DAM), expressway service area (ESA), expressway toll station (ETS), golf field (GF), ground track field (GTF), harbor (HA), overpass (OP), ship (SH), stadium (STA), storage tank (STO), tennis court (TC), train station (TS), vehicle (VE), and windmill (WM). The size of all the images of DIOR-R is 800 × 800 pixels and the spatial resolution ranges from 0.5 to 30 m.

HRSC2016 is designed for ship detection in aerial images. All the images are collected from Google Earth. It contains 1061 images and more than 20 categories of ships, including Nimitz-class aircraft carriers, Perry-class frigates, and medical ships. The sizes of the images range from 300 × 300 to 1500 × 900. The amount of the HRSC2015 dataset is limited, but there are too many categories. Therefore, we merged all ship classes into a single class for detection performance.

### 4.2 Configuration details

**Dataset preprocessing**. All datasets containing DOTA-v1.0, DOTA-v1.5, and HRSC2016 are split into image patches of 1024 × 1024. The gap of neighbor image patches is set as 200 pixels. For fair and reasonable analysis, all experiments are performed under a single scale of original datasets.

**Parameter set**. We applied YOLOv5 as our baseline. The SGD optimizer with the momentum of 0.937, learning rate of 0.01, and weight decay of 0.0005 was initialized in the training process. We trained all the models in 150 epochs for DOTA-v1.0 and DOTA-v1.5, and 300 epochs for HRSC2016. We used a single GTX1080Ti for training with a batch size of 4 and prediction with a single batch size.

### 4.3 Ablation studies

To sufficiently verify that the decoupled head and BDAM are significant, we performed a series of ablation experiments over the DOTA-v1.5 dataset, as shown in Table 1.

To analyze the effect of BDAM given by different parameters N, Fig. 5 shows the object confidence predictions of YOLOv5m, YOLOv5m-BDAM4, YOLOv5m-BDAM8, and YOLOv5m-BDAM16, where image A is the input image and the images B, C, D, and E are feature maps of object confidence predictions from YOLOv5m, YOLOv5m-BDAM4,

Table 1
Roadmap of ablation studies in terms of AP on DOTA-v1.5 test set evaluation. Measurements on latency are performed without postprocessing.

| Method | mAP | Latency | Parameter size |
|---|---|---|---|
| YOLOv5-M | 71.46 | 0.048 | 44.9 |
| +BDAM | 72.16 | 0.059 | 45.4 |
| +Decoupled head | 72.71 | 0.102 | 90.7 |

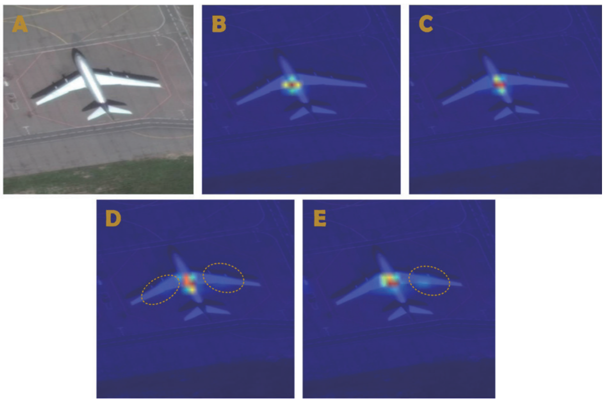


Fig. 5. (Color online) Feature map of object confidence prediction.

YOLOv5m-BDAM8, and YOLOv5m-BDAM16, respectively. As shown in Fig. 6, we can see that the activated area will be a better fit with the shape of objects when N of BDAM is set as 8 or 16. With the increment in N, the feature maps will be split into more tiles. Therefore, the detector can obtain more and smaller contextual information. In the case of small object detection, it is better to set N larger. However, if the parameter of N is set too large, some large objects will be inappropriately divided into too many parts. Through the above analysis and experiments, we set N of BDAM as 8 for a better balance between detections for small and large objects.

In this research, we take YOLOv5-M as the baseline with mAP of 71.46. BDAM can achieve 72.16 mAP with a slight improvement of prediction time and parameter size. In the case of using decoupled heads, mAP on DOTA-v1.5 can be promoted to 72.71, but the parameter size and prediction time are considerably increased.

### 4.4 Comparison with state-of-the-art methods

**Results on DOTA.** We report the full experimental results of single scale on DOTA datasets, including DOTA-v1.0 and DOTA-v1.5, as shown in Tables 2 and 3, where the results in red denote the best results and those in blue represent the second-best results in each column. With SCPDarknet as our backbone, we achieved 78.06 and 74.32 mAP on DOTA-v1.0 and DOTA-v1.5, respectively. The results of experiments verified the effectiveness of the proposed methods. In particular, the performance on the container crane category of DOTA-v1.5, which contains extremely few instances, is promoted significantly. The visualization of the oriented YOLOv5m-BDAM8 is shown as Fig. 6.

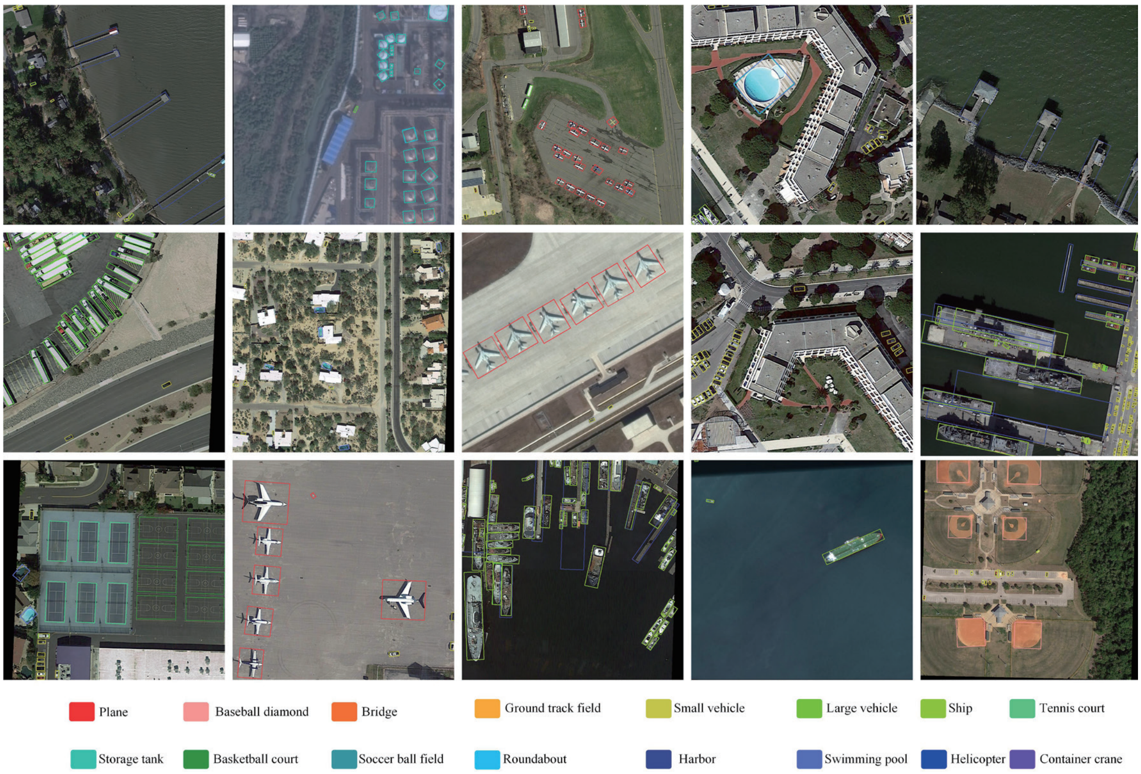


Fig. 6. (Color online) Visualization of oriented YOLOv5 on DOTAv1.5.

Table 2
(Color online) Performances on DOTA-v1.0 dataset.

| Method | PL | BD | BR | GTF | SV | LV | SH | TC | BC | ST | SBF | RA | HA | SP | HC | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FR-O[21] | 79.4 | 77.1 | 17.7 | 64.0 | 35.3 | 38.0 | 37.1 | 89.4 | 69.6 | 59.2 | 50.3 | 52.9 | 47.8 | 47.4 | 46.3 | 54.1 |
| RRPN[44] | 80.9 | 65.7 | 35.3 | 67.4 | 59.9 | 50.9 | 55.8 | 90.6 | 66.9 | 72.3 | 55.0 | 52.2 | 55.1 | 53.4 | 48.2 | 61.0 |
| Yang et al.[45] | 81.2 | 71.4 | 36.5 | 67.4 | 61.1 | 50.9 | 56.6 | 90.6 | 68.0 | 72.3 | 55.0 | 55.6 | 62.4 | 53.4 | 51.5 | 62.2 |
| RADet[46] | 79.4 | 76.9 | 48.0 | 65.8 | 65.4 | 74.4 | 68.8 | 89.7 | 78.1 | 74.9 | 49.9 | 64.6 | 66.1 | 71.6 | 62.2 | 69.0 |
| Cascade-FF[47] | 89.9 | 80.4 | 51.7 | 77.4 | 68.2 | 75.2 | 75.6 | 90.8 | 78.8 | 84.4 | 62.3 | 64.6 | 57.7 | 69.4 | 50.1 | 71.8 |
| DRN[48] | 88.9 | 80.2 | 43.5 | 63.3 | 73.4 | 70.6 | 84.9 | 90.1 | 83.8 | 84.1 | 50.1 | 58.4 | 67.6 | 68.6 | 52.5 | 70.7 |
| CenterMap[49] | 88.8 | 81.2 | 53.1 | 60.6 | 78.6 | 66.5 | 78.1 | 88.8 | 77.8 | 83.6 | 49.3 | 66.1 | 72.1 | 72.3 | 58.7 | 71.7 |
| R3Det[29] | 89.4 | 81.1 | 50.5 | 66.1 | 70.9 | 78.6 | 78.2 | 90.8 | 85.2 | 84.2 | 61.8 | 63.7 | 68.1 | 69.8 | 67.2 | 73.7 |
| RepPoints-O[50] | 89.1 | 82.3 | 56.7 | 74.9 | 80.7 | 83.7 | 87.6 | 90.8 | 87.1 | 85.8 | 63.6 | 68.6 | 75.9 | 73.5 | 63.7 | 77.6 |
| Oriented R-CNN[31] | 88.8 | 83.4 | 55.2 | 76.9 | 74.2 | 82.1 | 87.5 | 90.9 | 85.5 | 85.3 | 65.5 | 66.8 | 74.3 | 70.1 | 57.2 | 76.2 |
| DODet[51] | 89.6 | 83.1 | 51.4 | 71.0 | 79.1 | 81.9 | 87.7 | 90.8 | 86.5 | 84.5 | 62.2 | 65.3 | 71.9 | 70.7 | 62.9 | 75.8 |
| Oriented YOLOv5s | 89.0 | 84.5 | 49.6 | 63.7 | 81.2 | 84.6 | 88.3 | 90.8 | 86.2 | 87.6 | 59.5 | 67.6 | 75.1 | 81.4 | 64.8 | 76.9 |
| Oriented YOLOv5m | 88.1 | 85.1 | 53.3 | 63.5 | 81.3 | 84.9 | 88.3 | 90.6 | 87.3 | 88.2 | 60.9 | 61.9 | 76.7 | 81.7 | 63.7 | 77.1 |
| Oriented YOLOv5l | 89.1 | 85.5 | 50.6 | 63.3 | 81.2 | 84.6 | 88.5 | 90.8 | 86.2 | 87.8 | 59.5 | 67.6 | 76.2 | 81.4 | 69.8 | 77.3 |
| Oriented YOLOv5x | 88.5 | 86.0 | 56.2 | 63.7 | 81.2 | 85.5 | 88.6 | 90.8 | 87.5 | 87.9 | 60.8 | 64.2 | 77.3 | 80.9 | 71.3 | 78.1 |

**Results on DIOR-R.** The DIOR-R dataset contains 20 categories in remote sensing images. The size of all the images is set to 800 × 800. As shown in Table 4, our methods achieve the state-of-the-art performance with mAP of 70.09 under voc2007 metrics. The accuracy of our method is better than those of other methods in such categories as APL, APO, BF, BC, HA, OP, SH, STO, TC, and VE. The mAP on APL, SH, and TC can reach over 90, which exceeds others by at least 10. In particular, the mAP on ALP (29.82) is better than that on DODet.[51]

**Results on HRSC2016.** The HRSC2016 dataset contains many thin and long ship instances. As shown in Table 5, our methods achieve the state-of-the-art performance with mAP of 93.27 under voc2007 metrics.

Table 3
(Color online) Performances on DOTA-v1.5 dataset.

| Method | PL | BD | BR | GTF | SV | LV | SH | TC | BC | ST | SBF | RA | HA | SP | HC | CC | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Retina-O[17] | 71.4 | 77.6 | 42.1 | 64.6 | 44.3 | 56.7 | 73.3 | 90.8 | 76.0 | 59.9 | 46.9 | 69.2 | 59.6 | 64.5 | 48.0 | 0.8 | 59.1 |
| FR-O[21] | 71.8 | 74.4 | 44.4 | 59.8 | 51.2 | 68.9 | 79.3 | 90.8 | 77.3 | 67.5 | 47.7 | 69.7 | 61.2 | 65.2 | 60.4 | 1.5 | 62.0 |
| Mask-RCNN[19] | 76.8 | 73.5 | 49.9 | 57.8 | 51.3 | 71.3 | 79.7 | 90.5 | 74.2 | 66.0 | 46.2 | 70.6 | 63.0 | 64.4 | 57.8 | 9.4 | 62.6 |
| HTC[52] | 77.8 | 73.6 | 51.4 | 63.9 | 51.5 | 73.3 | 80.3 | 90.5 | 75.1 | 67.3 | 48.5 | 70.6 | 64.8 | 64.4 | 55.8 | 5.1 | 63.4 |
| YOLOv5_CSL_F[53] | 80.7 | 77.2 | 41.9 | 55.9 | 59.6 | 76.2 | 90.6 | 77.9 | 78.1 | 45.7 | 64.9 | 67.5 | 69.3 | 45.2 | 45.2 | 20.3 | 65.2 |
| ReDet[25] | 79.2 | 82.8 | 51.9 | 71.4 | 52.3 | 75.7 | 80.9 | 90.8 | 75.8 | 68.6 | 49.2 | 72.0 | 73.3 | 70.5 | 63.3 | 11.5 | 66.8 |
| FCOSR[54] | 80.5 | 85.2 | 51.1 | 70.8 | 57.7 | 76.7 | 81.1 | 90.9 | 78.0 | 77.6 | 51.9 | 68.7 | 75.8 | 72.6 | 69.3 | 31.0 | 69.9 |
| Oriented YOLOv5s | 80.7 | 84.9 | 47.3 | 61.7 | 65.2 | 80.9 | 88.8 | 90.9 | 77.6 | 79.0 | 55.8 | 73.4 | 71.2 | 76.5 | 62.8 | 34.4 | 70.7 |
| Oriented YOLOv5m | 80.9 | 85.7 | 52.8 | 63.0 | 66.8 | 82.1 | 89.4 | 90.9 | 79.4 | 85.3 | 55.0 | 72.8 | 74.9 | 77.3 | 71.6 | 35.0 | 72.7 |
| Oriented YOLOv5l | 81.0 | 84.9 | 53.9 | 64.7 | 67.4 | 82.7 | 89.6 | 90.8 | 79.2 | 85.9 | 54.8 | 74.7 | 75.8 | 76.8 | 70.4 | 39.4 | 73.2 |
| Oriented YOLOv5x | 89.1 | 85.6 | 54.4 | 64.7 | 67.7 | 82.9 | 89.7 | 90.8 | 83.4 | 85.7 | 55.3 | 75.6 | 75.9 | 77.1 | 76.5 | 34.4 | 74.3 |

Table 4
(Color online) Performances on DIOR dataset.

| Method | APL | APO | BF | BC | BR | CH | DAM | ETS | ESA | GF | GTF | HA | OP | SH | STA | STO | TC | TS | VE | WM | mAP |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FR-O[21] | 62.8 | 26.8 | 71.7 | 80.9 | 34.2 | 72.5 | 18.9 | 66.4 | 65.7 | 66.6 | 79.2 | 34.9 | 48.7 | 81.1 | 64.3 | 71.2 | 81.4 | 47.3 | 50.4 | 65.2 | 59.5 |
| Retina-O[17] | 61.4 | 28.5 | 73.5 | 81.1 | 23.9 | 72.5 | 19.9 | 72.4 | 58.2 | 69.2 | 79.5 | 32.1 | 44.8 | 77.7 | 67.5 | 61.1 | 81.4 | 47.3 | 38.0 | 60.2 | 57.5 |
| GV[55] | 65.3 | 28.8 | 74.9 | 81.3 | 33.8 | 74.3 | 19.5 | 70.7 | 64.7 | 72.3 | 78.6 | 37.2 | 49.6 | 80.2 | 69.2 | 61.1 | 81.4 | 44.7 | 47.7 | 65.0 | 60.0 |
| RT[24] | 63.3 | 37.8 | 71.7 | 87.5 | 40.6 | 72.6 | 26.8 | 78.7 | 68.1 | 68.6 | 82.7 | 47.7 | 55.6 | 81.0 | 78.2 | 70.2 | 81.6 | 54.8 | 43.2 | 65.5 | 63.8 |
| AOPG[42] | 62.4 | 37.7 | 71.6 | 87.6 | 40.9 | 72.4 | 31.0 | 65.4 | 78.0 | 78.0 | 71.9 | 42.3 | 54.4 | 81.1 | 72.7 | 71.3 | 81.4 | 60.0 | 52.3 | 70.0 | 64.4 |
| DODet[51] | 63.4 | 43.3 | 72.1 | 81.3 | 43.1 | 72.9 | 33.3 | 78.8 | 70.8 | 70.8 | 75.5 | 48.0 | 59.3 | 85.4 | 74.0 | 71.5 | 81.5 | 55.5 | 51.8 | 66.4 | 65.1 |
| Oriented YOLOv5s | 89.7 | 54.1 | 80.7 | 93.1 | 50.1 | 72.4 | 38.0 | 67.8 | 69.7 | 50.9 | 69.4 | 52.8 | 61.5 | 92.5 | 60.8 | 68.8 | 92.4 | 40.1 | 66.0 | 48.2 | 66.0 |
| Oriented YOLOv5m | 91.6 | 60.7 | 83.2 | 94.3 | 58.1 | 82.9 | 43.3 | 69.0 | 77.2 | 57.4 | 76.5 | 58.8 | 66.2 | 93.5 | 61.7 | 71.7 | 93.0 | 49.3 | 68.4 | 53.5 | 70.5 |
| Oriented YOLOv5l | 91.1 | 61.8 | 81.9 | 93.7 | 54.8 | 79.8 | 44.0 | 70.5 | 75.1 | 61.5 | 74.5 | 56.2 | 66.2 | 93.1 | 63.2 | 73.9 | 92.5 | 48.7 | 68.1 | 61.0 | 70.6 |
| Oriented YOLOv5x | 93.2 | 67.5 | 81.8 | 94.4 | 32.2 | 44.4 | 45.8 | 70.7 | 75.4 | 61.1 | 75.2 | 58.4 | 66.6 | 93.3 | 66.8 | 70.9 | 93.6 | 46.6 | 67.9 | 52.3 | 70.1 |

Table 5
Performances on HRSC2016 dataset.

| Method | CP[56] | BL2[56] | RC1[56] | RC2[56] | RRPN[44] | CSL[38] |
|---|---|---|---|---|---|---|
| mAP | 55.7 | 69.6 | 75.7 | 75.7 | 79.6 | 89.62 |
| Method | RRD[57] | RoI Trans[24] | R3Det[29] | Gliding Vertex[55] | Redet[25] | Oriented YOLOv5-x |
| mAP | 84.3 | 86.2 | 89.26 | 88.2 | 90.46 | 93.27 |

## 5. Conclusions

In this research, we proposed an improved method based on YOLOv5 to deal with object detection tasks in remote sensing images. Considering the conflicts between different types of regression and classification, the detector head is decoupled into four heads for boundary box regression, object, category, and angle classification. DAM is designed to integrate channel and spatial background features. On the basis of the analysis of remote sensing image features and DAM, BDAM was developed using a block processing method and applied to our detection algorithm. The effectiveness of our target detection method in remote sensing images was verified through experiments on the DOTA, DIOR-R, and HRSC2016 datasets.

## References

1 K. Su, J. Li, and H. Fu: 2011 Int. Conf. Electron. Commun. Control. (IEEE, 2011) 1028–1031.https://doi.org/10.1109/ICECC.2011.6066743
2 M. M. Rathore, A. Ahmad, and A. Paul: 2016 IEEE/CVF Int. Conf. Autom. (IEEE, 2016) 1–8. https://doi.org/10.1109/ICA-ACCA.2016.7778510
3 B. Zhang, D. Wu, L. Zhang, Q. Jiao, and Q. Li: Environ. Earth Sci. **65** (2012) 649. https://doi.org/10.1007/s12665-011-1112-y
4 Y. Zhang, Y. Lu, D. Zhang, L. Shang, D. Wang, and Risksens: 2018 IEEE Int. Conf. Big Data (IEEE, 2018) 1544–1553. https://doi.org/10.1109/BigData.2018.8621996
5 L. Ruiqi, Z. Xian, Z. Luo, and L. Lin: Cluster Comput. **22** (2019) 12581. https://doi.org/10.1007/s10586-017-1684-8
6 D. Ngoduy: Commun. Nonlinear Sci. Numer Simul. **18** (2013) 2699. https://doi.org/10.1016/j.cnsns.2013.02.018
7 J. Zhang, L. Chen, C. Wang, L. Zhuo, Q. Tian, and X. Liang: IEEE Trans. Intell. Transp. Syst. **18** (2017) 2993. https://doi.org/10.1109/TITS.2017.2665658
8 Y. Wu, K. Zhang, J. Wang, Y. Wang, Q. Wang, and Q. Li: IEEE Geosci. Remote Sens. Lett. **19** (2020) 1. https://doi.org/10.1109/LGRS.2020.3042465
9 G. S. Xia, X. Bai, J. Ding, Z. Zhu, B. Serge, L. Jiebo, D. Mihai, P. Marcello, and L. Zhang: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (IEEE, 2018) 3974–3983. https://doi.org/10.1109/CVPR.2018.00418
10 Z. Liu, H. Wang, L. Weng, and Y. Yang: IEEE Geosci. Remote Sens. Lett. **13** (2016) 1074. https://doi.org/10.1109/LGRS.2016.2565705
11 Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu: IEEE Trans. Neural Netw. Learn Syst. **30** (2019) 3212. https://doi.org/10.1109/TNNLS.2018.2876865
12 J. Redmon and A. Farhadi: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (IEEE, 2017) 7263–7271. http://doi.org/10.1109/CVPR.2017.690
13 J. Redmon and A. Farhadi: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit (IEEE, 2018) 1804–2767. https://doi.org/10.48550/arXiv.1804.02767
14 A. Bochkovskiy, C. Y. Wang, and H. YM. Liao: arXiv preprint (2020). https://doi.org/10.48550/arXiv.2004.10934
15 Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun: arXiv preprint (2021). https://doi.org/10.48550/arXiv.2107.08430
16 W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Y. Fu, and B. Alexander: Proc. Eur. Conf. Comput. Vis (Springer, 2016) 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
17 T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár: Proc. IEEE/CVF Int. Conf. Comput. Vis. (IEEE, 2017) 2980–2988. https://doi.org/10.1109/iccv.2017.324
18 B. Wu, F. Iandola, PH. Jin, and K. Keutzer: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit (IEEE, 2017) 129–137. https://doi.org/10.1109/CVPRW.2017.60
19 K. He, G. Gkioxari, P. Dollár, and R. Girshick: Proc. IEEE/CVF Int. Conf. Comput. Vis. (IEEE, 2017) 2961–2969.
20 R. Girshick: Proc. IEEE/CVF Int. Conf. Comput. Vis. (IEEE, 2015) 1440–1448. https://doi.org/10.1109/ICCV.2015.169
21 S. Ren, K. He, R. Girshick, and J. Sun: Adv. Neural Inf. Process Syst. **28** (2015) 91. https://doi.org/10.48550/arXiv.1506.01497

22 C. Li, C. Xu, Z. Cui, D. Wang, Z. Jie, T. Zhang, and J. Yang: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit (IEEE, 2019) 20–27.
23 G. Cheng, P. Zhou, and J. Han: IEEE Trans Geosci. Remote Sens. **54** (2016) 7405. https://doi.org/10.1109/TGRS.2016.2601622
24 J. Ding J, N. Xue, Y. Long, G. S. Xia, and Q. Lu: Proc. IEEE/CVF Conf. Comput. Vis. Pattern. Recognit (IEEE, 2019) 2849–2858.
25 J. Han, J. Ding, N. Xue, and G. S. Xia: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit (IEEE, 2021) 2786–2795. https://doi.org/10.48550/arXiv.2103.07733
26 M. Weiler and G. Cesa: Adv. Neural Inf. Process Syst. **32** (2019) 14334 https://doi.org/10.48550/arXiv.1911.08251
27 Z. Qu, F. Zhu, and C. Qi: Remote Sensing. **13** (2021) 3908. https://doi.org/10.3390/rs13193908
28 S. Woo, J. Park, J. Lee, and I S. Kweon: Proc. Eur. Conf. Comput. Vis (Springer, 2018) 3–19 . https://doi.org/10.1007/978-3-030-01234-2_1
29 X. Yang, J. Yan, Z. Feng and T. He: Proc. AAAI Conf. Artif. Intell. (AAAI, 2021) 3163–3171. https://doi.org/10.1609/aaai.v35i4.16426
30 X. Chen, L. Ma, and Q. Du: IEEE Geosci. Remote Sens. Lett. **19** (2021) 1. https://doi.org/10.1109/LGRS.2021.3079314
31 X. Xie, G. Cheng, J. Wang, X. Yao, and J. Han: Proc. IEEE/CVF Int. Conf. Comput. Vis. (IEEE, 2021) 3520–3529. https://doi.org/10.48550/arXiv.2108.05699
32 D. Xu and Y. Wu: Sensors. **15** (2020) 4276. https://doi.org/10.3390/s20154276
33 G. Huang, Z. Liu, Van Der Maaten, and K. Q. Weinberger: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit (IEEE, 2017) 4700–4708. https://doi.org/10.1109/CVPR.2017.243
34 C. Cao, J. Wu, X. Zeng, Z. Feng, T. Wang, and X. Yan: Sensors. **20** (2020) 4696. https://doi.org/10.3390/s20174696
35 H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia: Proc. IEEE/CVF Conf. Comput Vis. Pattern Recognit (IEEE, 2017) 2881–2890 . https://doi.org/10.1109/CVPR.2017.660
36 D. Wan, R. Lu, S. Wang, S. Shen, T. Xu, and X. Lang: Remote Sens. **15** (2023) 614. https://doi.org/10.3390/rs15030614
37 Z. Liu, Y. Gao, Q. Du, M. Chen, and W. Lv: IEEE Access. **11** (2023) 1742. https://doi.org/10.1109/ACCESS.2023.3233964
38 X. Yang and J. Yan: Proc. Eur. Conf. Comput. Vis (Springer, 2020) 677. https://doi.org/10.1007/978-3-030-58598-3_40.
39 S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia: Proc. IEEE/CVF Conf. Comput Vis. Pattern Recognit (IEEE, 2018) 8759. https://doi.org/10.1109/CVPR.2018.00913
40 Z. Tian, C. Shen, H. Chen, and T. He: Proc. IEEE/CVF Int. Conf. Comput Vis. (IEEE, 2019) 9627–9636. https://doi.org/10.48550/arXiv.1904.01355
41 Y. Wu, Y. Chen, L. Yuan, Z. Liu, L. Wang, H. Li, and Y. Fu: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit (IEEE, 2020) 10186. https://doi.org/10.48550/arXiv.1904.06493
42 G. Cheng, J. Wang, K. Li, X. Xie, C. Lang, Y. Yao, and J. Han: IEEE Trans. Geosci. Remote Sens. **60** (2022) 1. https://doi.org/10.48550/arXiv.2110.01931
43 H. Zhang, M. Cisse, YN. Dauphin and D. Lopez-Paz: Int. Conf. Learning Representations (Springer, 2018) 1. https://doi.org/10.48550/arXiv.1710.09412
44 J. Ma, W. Shao, H. Ye, L. Wang, H. Wang, Y. Zheng, and X. Xue: IEEE Trans. Multimedia **20** (2018) 3111. https://doi.org/10.1109/ACCESS.2018.2869884
45 Yang X, Sun H, Sun X, Yan M, Guo Z, and Fu K: IEEE Access. **6** (2018) 50839. https://doi.org/10.3390/rs12030389
46 Y. Li, Q. Huang, X. Pei, L. Jiao, and R. Shang: Remote Sens. **12** (2020) 389. https://doi.org/10.3390/rs12030389
47 L. Hou, K. Lu, J. Xue, and L. Hao: IEEE Int. Conf. Multimedia and Expo. (IEEE, 2020) 1–6. https://doi.org/10.1109/ICME46284.2020.9102807
48 X. Pan, Y. Ren, K. Sheng, W. Dong, H. Yuan, X. Guo, C. Ma, and C. Xu: Proc. IEEE/CVF Conf. Comput. Vis. PatternRrecognit. (IEEE, 2020) 11207–11216. https://doi.org/10.1109/CVPR42600.2020.01122
49 J. Wang, W. Yang, H. Li, H. Zhang, and G. S. Xia: IEEE Tran. Geosci. Remote Sens. **59** (2020) 4307. https://doi.org/10.1109/TGRS.2020.3010051
50 W. Li and J. Zhu: arXiv preprint (2021). https://doi.org/10.48550/arXiv.2105.11111
51 G. Cheng, Y. Yao, S. Li, K. Li , X. Xie, J. Wang, X. Yao, and J. Han: IEEE Tran. Geosci. Remote Sens. **60** (2022) 1. https://doi.org/10.1109/TGRS.2022.3149780
52 K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, and S. Sun: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit (IEEE, 2019) 4974–4983. https://doi.org/10.1109/CVPR.2019.00511

53 J. Wang, T. Xiao, Q. Gu, and Q. Chen: 2021 Int. Conf. Wirel Commun. Smart Grid. (IEEE, 2021) 197–203. https://doi.org/10.1109/ICWCSG53609.2021.00045

54 Z. Li, B. Hou, Z. Wu, L. Jiao, B. Ren, and C. Yang: arXiv (2021). https://doi.org/10.48550/arXiv.2111.10780

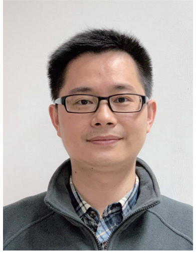
55 Y. Xu, M. Fu, Q. Wang, Y. Wang, K. Chen, G. S. Xia and X. Bai: IEEE Trans. Pattern Anal. Mach Intell. **43** (2020) 1452–1459. https://doi.org/10.1109/TPAMI.2020.2974745

56 Z. Liu, J. Hu, L. Weng, and Y. Yang: Proc. IEEE Int. Conf. Image Process (IEEE, 2017) 900–904. https://doi.org/10.1109/ICIP.2017.8296411

57 M. Liao, Z. Zhu, B. Shi, G. S. Xia, X. Bai: Proc. IEEE/CVF Conf. Comput Vis. Pattern Recognit. (IEEE, 2018) 5909–5918. https://doi.org/10.1109/CVPR.2018.00619

## About the Authors

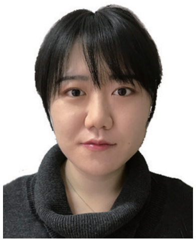
**Gang Yang** received his B.S and M.S degrees from Beijing Jiaotong University in 2014 and 2019, respectively. He is working for Beijing Institute of Survey and Mapping and Beijing Key Laboratory of Urban Spatial Information Engineering, Beijing, China. His research interests include machine learning and remote sensing image processing. (yanggang@bjtu.edu.cn)



**Miao Wang** received his B.S. and M.S. degrees from Wuhan University, China, in 2008 and 2010, respectively. Since 2010, he has been working for Beijing Institute of Surveying and Mapping as a senior engineer. His research interests are in urban spatial analysis and geographic information science. (wangmiao@bism.cn)

**Quan Zhou** received his B.S. degree from Capital Normal University, China, in 2015 and his M.S. degree from the University of Twente, Netherlands in 2018. Since 2018 he has been an engineer at Beijing Institute of Surveying and Mapping. His research interests are in smart city, remote sensing, and system architecture design. (zhouquan_brook@163.com)

**Jiangchuan Li** received his B.S. and M.S. degrees from Beijing University of Civil Engineering and Architecture, China, in 2019 and 2022, respectively. Since 2022, he has been an assistant engineer at Beijing Institute of Surveying and Mapping. His research interests are in GIS and remote sensing. (lijiangchuan@bism.cn)



**Siyue Zhou** received her B.E degree from Beijing Forestry University in 2017 and her M.A degree from the University of London in 2019. She is an engineer in Beijing Institute of Surveying and Mapping, Beijing, China. Her research interests include geoscience and urban space study. (siyuezhou@outlook.com)

**Yutong Lu** received her B.S degree in geographic information science from Beijing University of Civil Engineering and Architecture in 2017 and her M.S degree in civil engineering (environmental and water systems) from Northeastern University in 2020. She is an engineer at Beijing Institute of Surveying and Mapping, Beijing, China. Her research interests include geographic information system, natural resource ownership confirmation, and smart city. (luyutong@bism.cn)