Multimedia Tools and Applications (2020) 79:34973–34992
https://doi.org/10.1007/s11042-020-08876-9

Fusion based feature reinforcement component for remote sensing image object detection

Dongjun Zhu[1,2] · Shixiong Xia[1,2] · Jiaqi Zhao[1,2] · Yong Zhou[1,2] · Qiang Niu[1,2] · Rui Yao[1,2] · Ying Chen[1,2]

Received: 6 March 2019 / Revised: 6 March 2020 / Accepted: 19 March 2020 /
Published online: 30 May 2020

Abstract

In recent years, convolutional neural networks (CNN) have been extensively used for generic object detection due to their powerful feature extraction capabilities. This has hence motivated researchers to adopt this technology in the field of remote sensing. However, remote sensing images can contain large amounts of noise, have complex backgrounds, include small dense objects as well as being susceptible to weather and light intensity variations. Moreover, from different shooting angles, objects can either have different shapes or be obscured by structures such as buildings and trees. Due to these, effective features extraction for proper representation is still very challenging from remote sensing images. This paper therefore proposes a novel remote sensing image object detection approach applying a fusion-based feature reinforcement component (FB-FRC) to improve the discrimination between object feature. Specifically, two fusion strategies are proposed: (i) a hard fusion strategy through artificially-set rules, and (ii) a soft fusion strategy by learning the fusion parameters. Experiments carried out on four widely used remote sensing datasets (NWPU VHR-10, VisDrone2018, DOTA and RSOD) have shown promising results where the proposed approach manages to outperform several state-of-the-art methods.

Keywords Remote sensing · Object detection · Reinforcement component · Fusion strategy

1 Introduction

With the development of remote sensing technologies, remote sensing image analysis is becoming more and more important. It can facilitate applications such as disaster control, environmental studies [6] and traffic planning [27]. As one fundamental task in computer vision, object detection is the basis of remote sensing image analysis. However, remote sensing images have vast backgrounds with many cluttered areas [26] and different size objects, which declining the performance of object detection on remote sensing images.

✉ Shixiong Xia
shixiongxia.cumt@outlook.com

Extended author information available on the last page of the article.

Object detection includes three main tasks: feature extraction, proposals classification and bounding box regression [1, 17]. Traditional feature extraction usually uses the hand-crafted features, such as Scale Invariant Feature Transform (SIFT) [20], Histogram of Oriented Gradients (HOG) [5] and texture features [11]. However, with the broad application of deep convolutional neural network (DCNN) [13] in image feature extraction, hand-crafted features are gradually replaced by automatically learned feature representation. The classification task is used to judge the category of objects. There are two main kinds of object detection models. One divides the object detection process into two steps. The first step is to generate the region proposals by doing a binary classification (object or background) on the extracted feature maps [8], the second step is to judge the category of each region proposal. The other just uses one step, doing a multi classification directly (including object categories and background) on the extracted feature maps. The regression task is used to revise the bounding box position and output the coordinate offset.

In early studies on remote sensing image object detection, Cheng et al. [2] applied multi-scale HOG features to build a discriminatively trained mixture method for object detection to detect different size objects in remote sensing images. To effectively identify the objects in remote sensing images, Senaras et al. [25] analyzed various object features (e.g., color, texture and shape), and applied different base-layer classifiers in the fuzzy stacked generalization architecture for detecting buildings. Han et al. [10] used a deep Boltzmann machine to find the spatial and structural information of features encoded in low-level and middle-level. Despite the great success of methods above, they are all based on hand-crafted features, which are time-consuming and require the domain expertise.

With the development of deep learning, DCNN has enjoyed a massive success in computer vision. As an essential task in computer vision, the object detection based on deep learning, such as R-CNN [9], Faster R-CNN [23] and YOLO [22], has a significant improvement on the detection performance compared with traditional object detection methods. Long et al. [19] proposed an unsupervised score-based bounding box regression for the accurate object localization in remote sensing images. Dai et al. [4] proposed the position-sensitive score maps to get accurate and fast object detection. However, those methods using the single-scale feature cannot adapt to the cluttered background and multi-scale objects in remote sensing images. Liu et al. [18] proposed a single shot multibox detector, using multi-scale feature maps to detect various size objects to improve the detection speed. However, the weak semantic of high-resolution feature maps limited the detection accuracy. Li et al. [14] used a coarse-to-fine merged manner to get discriminative candidate regions, nevertheless, the simple and single merged manner limited the feature representation due to the difference between the high-resolution features and low-resolution features after undergoing several convolution layers.

In recent years, some detection models take advantage of the pyramid structure of backbone networks, using nearest neighbor upsampling and element-wise sum to fuse different resolution feature maps to obtain strong feature representation, improving the performance on generic object detection, such as FPN [15]. However, there are a lot of complex background (e.g., cities, forests and grasslands), noise and dense tiny objects in remote sensing images. Simply using nearest neighbor upsampling and element-wise sum to fuse the high-resolution feature maps and low-resolution feature maps lacks enough feature information, which is not suitable for remote sensing object detection. Due to the large difference between high-resolution feature map and low-resolution feature map, the fused feature maps can not achieve a good balance between details and semantics.

In this paper, we propose a novel remote sensing image object detection method with the fusion based feature reinforcement component (FB-FRC). Firstly, we apply the feature reinforcement component (FRC) to filter out some redundant details and strengthen the semantics of high-resolution feature maps. FRC can generate a new feature layer and provide more feature information for fusion, making up the high-resolution feature maps lack of semantics and low-resolution feature maps lack of details. Then, two feature fusion strategies (hard fusion strategy and soft fusion strategy) are designed to get the strong feature representation. Finally, experiments carried out on four remote sensing images datasets (NWPU VHR-10 [3], VisDrone2018 [30], DOTA [28] and RSOD [29]) verify the effectiveness of the proposed method.

In summary, the main contributions of the proposed method are listed in the following.

1) The FRC is applied to filter out some redundant details and strengthen the semantics of high-resolution feature maps, providing more feature information for fusion.

2) The hard fusion and soft fusion strategies are proposed to fuse the feature maps of different scales to get strong feature representation.

The rest of this paper is structured as follows. The details of proposed method are described in Section 2. Section 3 presents experimental results. Finally, Section 4 lists the conclusions of this paper.

2 Proposed method

In this section, we introduce the FB-FRC for remote sensing image object detection in detail. The framework of proposed method is illustrated in Fig. 1. This is a two-step detection method. In step one, the FRC and fusion strategies are used to generate the pyramid feature maps with high object discrimination for obtaining the region proposals. In step two, the region proposals are fed into the classifier and regressor to get final detection results. The details are described in the following.

2.1 Enhancing feature extraction

The dense objects and cluttered background in remote sensing images are prone to reduce the performance of object detection. Therefore, we add the FRC and apply two fusion strategies to enhance the object feature representation.

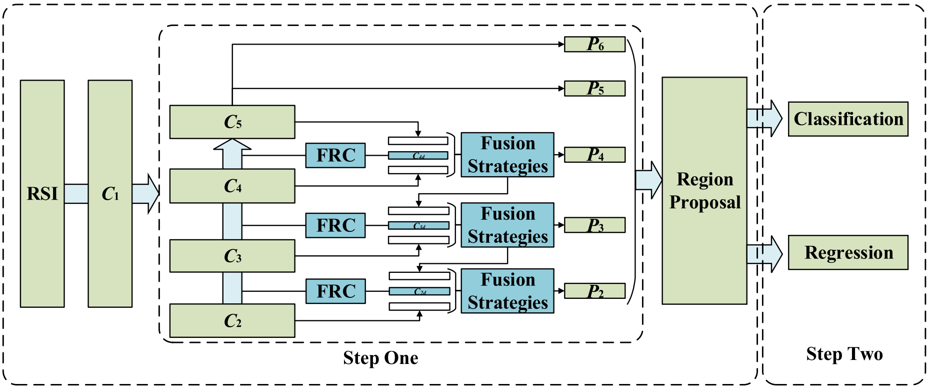


Fig. 1 The framework of the proposed method

The Fig. 2 shows the structure of FRC. The high-resolution feature map is downsampled by a residual block firstly, then undergoing a deconvolution (kernel size = 2, stride = 2) and 1×1 convolution to get the reinforced feature map, where a residual block is used to downsample the high-resolution feature map and further extract the semantic features. In the subsequent deconvolution operation, we use the small size kernel for upsampling. Different from ordinary images, the remote sensing images include the tiny and dense objects (e.g., the crowded pedestrians and congested vehicles), therefore, the two pixels next to each other in a feature map can represent two different objects. If the kernel size is too big, the corresponding size of receptive field would be large, which containing many pixels with different objects. After undergoing deconvolution, one pixel in the feature map contains the mixed feature from multiple objects, which disturb or even loss the tiny object feature. Therefore, we use a small kernel size in deconvolution. In FRC, the 1×1 convolution following deconvolution is applied to unify the dimensions for feature fusion. In this paper, the output of FRC is unified to 256 dimensions.

Referring to the description of ResNet in [12], we defined the residual block in FRC as:

$$y = F(x, W) + x \tag{1}$$

where x and y are the input and output of FRC respectively. The function F represents a residual mapping. The operation $F + x$ is performed by a shortcut connection and element-wise addition. The structure of the residual block is shown in Fig. 3, where F as a residual mapping consist of 3 convolution layers and 2 ReLU functions, the output of residual block is activated by a ReLU function. In residual block, the 1×1 convolution are responsible

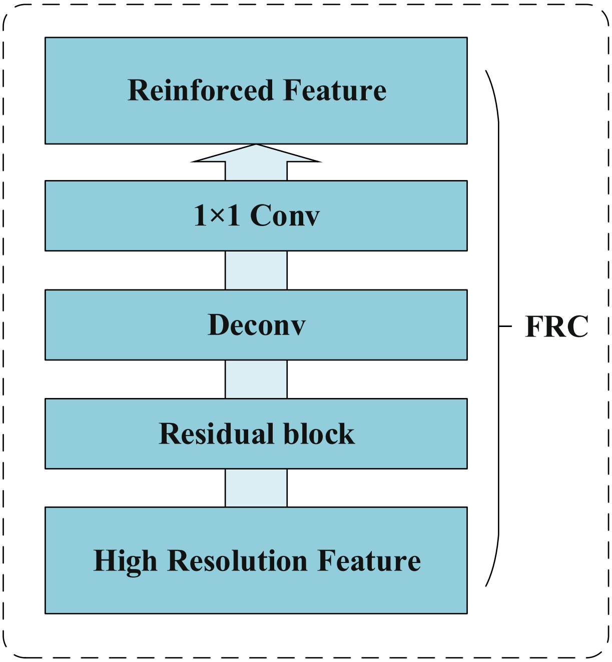


Fig. 2 The structure of FRC

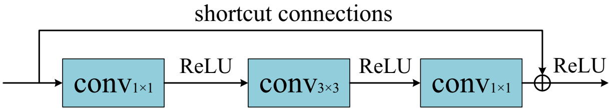


Fig. 3 The structure of residual block

for reducing and then increasing (restoring) dimensions, leaving the 3×3 convolution with smaller input/output dimensions [12], which decreases the computation and training time compared with using two 3×3 convolutions.

Different from blurring operations, FRC is applied to enhance the semantic while reduce the redundant details of the high-resolution feature map, and parameters in FRC are updated constantly during the training. In many cases, blurring operations are mainly used to filter image noise, where the parameters in the blurring operations are generally fixed or manually adjusted, such as mean blur and gaussian blur.

For the deeper networks, there is a degradation problem during training: with the depth increasing, accuracy gets saturated and then degrades rapidly. To solve this problem, the ResNet [12] was proposed, using residual learning to catch the subtle changes of networks to make the network training more effective. In this paper, the proposed method uses ResNet101 as the backbone network. As shown in Fig. 1, according to the times of downsampling, the backbone architecture can be divided into five stages, denoted as $\{C_1, C_2, C_3, C_4, C_5\}$, where the feature map resolution decreases continuously from C_1 to C_5, and the feature maps in one stage have the same resolution. The outputs of the last residual block in $\{C_2, C_3, C_4, C_5\}$ are denoted as $\{C_{2l}, C_{3l}, C_{4l}, C_{5l}\}$, in which $\{C_{2l}, C_{3l}, C_{4l}\}$ undergoes a FRC to generate the reinforced feature maps as $\{C_{2d}, C_{3d}, C_{4d}\}$, and FRC shares the first residual block with $\{C_3, C_4, C_5\}$. Compared with $\{C_{2l}, C_{3l}, C_{4l}\}$, $\{C_{2d}, C_{3d}, C_{4d}\}$ has stronger object semantic features and weaker background details. C_{5l} is the last feature output of C_5 and has very strong semantic, we just append a 3×3 convolution (stride=1) on C_{5l} to generate P_5 and apply another 3×3 convolution (stride=2) downsampling C_{5l} to generate P_6.

Figure 4 shows the examples of feature maps after undergoing FRC. The original image contains six oil tanks. Figure 4a and c show the feature maps of C_{2l} and C_{3l}. Figure 4b and d are C_{2d} and C_{3d} undergoing FRC. As shown in Fig. 4b and d, the background features in C_{2d} and C_{3d} are less cluttered than before, while the features of six oil tanks are more semantic, compared with the features in C_{2l} and C_{3l}.

As shown in Fig. 4, FRC can filter out some redundant details and strengthen the semantic of the feature maps. This makes the feature maps appear coarse-grained. Instead of simply using pooling and nearest neighbor upsampling, both residual block and deconvolution are applied in FRC, which is a lightweight component that can perform parameter learning during downsampling and upsampling. Therefore, the feature maps after FRC also retains some main details while enhancing the semantics. As shown in 1, we fuse the high-resolution feature map, the feature map undergoing FRC and the low-resolution feature map to get better object feature representation. As a new added layer, the feature map after undergoing FRC can make up for the high-resolution feature map's lack of semantics and low-resolution feature map's lack of details. Due to more feature information being considered, the fused feature map has better balance between details and semantics, making the remotely sensed objects easier to identify.

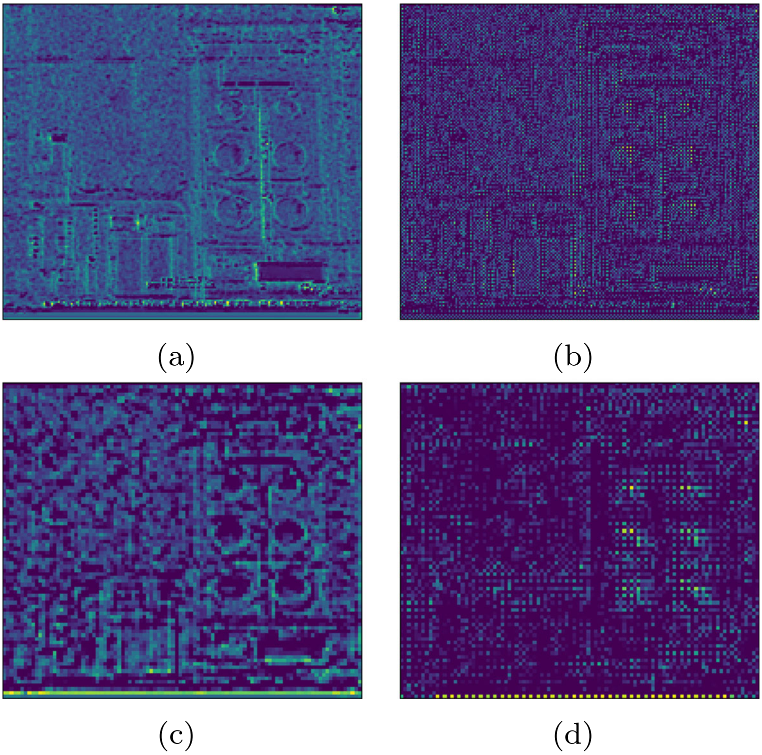


Fig. 4 Visualized feature map from a remote sensing image containing six oil tanks. **a** C_{2l} feature map. **b** C_{2d} feature map. **c** C_{3l} feature map. **d** C_{3d} feature map

2.2 Feature fusion strategies

As shown in Fig. 5a and b, two strategies are used to fuse the different feature maps, respectively. One is called hard fusion strategy by the element-wise sum of feature maps. The other is called soft fusion strategy by learning the fusion parameters.

For hard fusion strategy, the feature maps need to be unified to the same size and channel dimensions before fusion. Therefore, as shown in Figs. 1 and 5a, the feature map in (i+1) level appends a nearest neighbor upsampling and a 1×1 convolutional layer to generate F_1^i, where nearest neighbor upsampling can be defined as:

$$f(a+u, b+v) = \begin{cases} f(a,b) & u <= 0.5 \ and \ v < 0.5 \\ f(a,b+1) & u <= 0.5 \ and \ v > 0.5 \\ f(a+1,b) & u > 0.5 \ and \ v <= 0.5 \\ f(a+1,b+1) & u > 0.5 \ and \ v >= 0.5 \end{cases} \tag{2}$$

where (a, b) is pixel coordinates in the feature map before upsampling. $f(a, b)$ represents the value of pixel (a, b). $(a+u, b+v)$ is the mapped coordinates from the upsampled feature map into the original feature map, which $u \in$ (0,1) and $v \in$ (0,1).

The reinforced feature C_{id} is as F_2^i. C_{il} undergoes a 1×1 convolutional layer to output F_3^i. After that, $\{F_1^i, F_2^i, F_3^i\}$ is unified to 256 channels. Then, element-wise sum is applied to merge F_1^i, F_2^i and F_3^i. Finally, We use a 3×3 convolution to learn feature representation

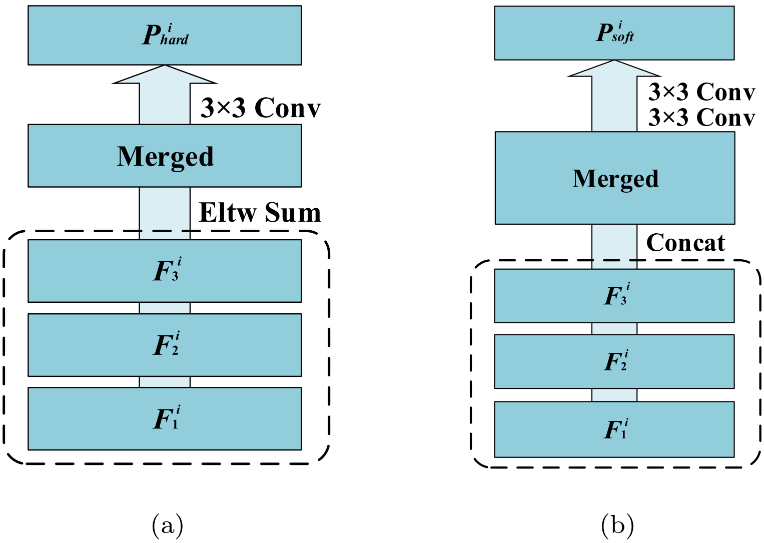


Fig. 5 The illustration of the two fusion strategies. **a** The hard fusion strategy. **b** The soft fusion strategy

from the merged feature map and unify the feature dimensions to generate the final fusion feature map P^i_{hard}. This can be expressed as (3).

$$P^i_{hard} = f_{sum}(F^i_1, F^i_2, F^i_3) \otimes conv_{3\times3} \tag{3}$$

Where ⊗ represents a convolution operation. To start the iteration, we attach a nearest neighbor upsampling and a 1×1 convolutional layer on C_{5l} to produce F^4_1 for fusion.

For soft fusion strategy, as shown in Figs. 1 and 5b, we make a nearest neighbor upsampling on the previous final fusion feature map $P_{(i+1)}$ to generate $F^{i'}_1$. And the reinforced feature C_{id} is as F^i_2. Then, C_{il} as $F^{i'}_3$ concat with $F^{i'}_1$ and $F^{i'}_2$. After the concat operation, the first 3×3 convolution is used to fuse three feature maps, the second 3×3 convolution extract feature representation from the fused feature map and unify feature dimensions to generate the final fusion feature map P^i_{soft}. The process which can be defined as (4).

$$P^i_{soft} = f_{concat}(F^i_1, F^i_2, F^i_3) \otimes conv_{3\times3} \otimes conv_{3\times3} \tag{4}$$

Similar to hard fusion strategy, we attach a nearest neighbor upsampling on C_{5l} to start the iteration.

The final fusion feature maps are called as $\{P_2, P_3, P_4\}$. $\{P_2, P_3, P_4, P_5, P_6\}$, as a feature pyramid, shares a 3×3 convolutional layer and two 1×1 convolutional layers to generate region proposals, where nonmaximum suppression (NMS) [21] is used to filter out the similar region proposals. After NMS, the region proposals are unified to the same dimensions by ROI pooling [23], then undergoing the two fully connected layers to produce the final predicted results.

2.3 Loss function of the proposed method

The classifier and regressor are shared between fusion feature map of each level to generate the region proposals. The classifier is used to predict the class probability (object or background) of each anchor in the fusion feature maps. The regressor estimates the coordinate offset of object bounding boxes, corresponding to the anchors' position. We define the anchor areas $\{512^2, 256^2, 128^2, 64^2, 32^2\}$ on $\{P_2, P_3, P_4, P_5, P_6\}$ respectively, and set the aspect ratios of anchors {0.5, 0.75, 1, 1.5, 2} on each level to fit different object shapes.

During training stage, we set the anchors whose values of Intersection-over-Union (IoU) [23] with any ground-truth is greater than 0.7 as positive labels and set the anchors' IoU with all ground-truth lower than 0.3 as negative labels, where IoU is used to measure the percentage of intersection between two bounding boxes, defined as:

$$\mathrm{IoU} = \frac{area(r_i) \cap area(g_i)}{area(r_i) \cup area(g_i)} \tag{5}$$

where r_i is a detection bounding box, and g_j represents a ground-truth. The $area(r_i)$ is the area enclosed by detection bounding box r_i.

The unmarked anchors are dropped out during training. The loss function for the region proposals is defined as (6).

$$\begin{aligned} Loss = \sum_{i \in Levels} \Big(\frac{1}{N_{cls}^{i}} \sum_{k \in A_i} L_{cls}(p_k, p_k^{*}) \\ + \lambda \frac{1}{N_{reg}^{i}} \sum_{k \in A_i} p_k^{*} L_{reg}(c_k, c_k^{*}) \Big) \end{aligned} \tag{6}$$

Where i is the index of level which the fusion feature maps belong to and A_i is the anchors set defined in the i-th level. p_k represents the probability that anchor i contains an object, and p_k^* is the label of ground-truth (1 for the positive labels, 0 for the negative labels). c_k represents the coordinates offsets of the predicted bounding box and c_k^* denotes the true coordinate offsets to ground-truth. $Loss_{cls}$ is softmax classfication loss and $Loss_{reg}$ is soomth L_1 loss which is used to learn four coordinate transformation [23] of the predicted bounding box and minimize the error between the predicted coordinates and ground truth coordinates. N_{cls}^i is the number of anchors in classification. Similar N_{reg}^i is the number of anchors in position regression. The λ is used to balance $Loss_{cls}$ and $Loss_{reg}$.

During training, the parameters is updated in a batch training, where the parameters is the weights of each layer in the network, used for network mapping. Taking a batch training as an example, the training process can be divided into two part. In the first part, a batch of images with the corresponding labels are fed into the proposed model, as shown in Fig. 1, after undergoing the step one and step two, the network outputs the predicted results, then the classification loss and regression loss can be calculated according to the predict results and labels. In the second part, based on the loss function, the gradients of each parameter in network are calculated by chain rule. Then the parameters in each network layer are updated according to the gradients and learning rate. This batch training process will be iterated during the training stage until the network converged.

3 Experimental study

To evaluate the effectiveness of the proposed method, experiments are conducted on four widely used remote sensing datasets, i.e. NWPU VHR-10 [3], VisDrone2018 [30], DOTA [28] and RSOD [29]. We compare the proposed method with several state-of-the-art methods, i.e. Faster R-CNN [23], R-FCN [4] and FPN [15]. The mean average precision (mAP) [16] and visualized results are adopted to evaluate the performance of these methods.

Table 1 Details of four remote sensing datasets

Datasets	Train images	Test images	Categories	Resolution
NWPU VHR-10	500	150	10	800∼1000
VisDrone2018	6471	548	10	900∼2000
DOTA	8813	2993	15	800∼4000
RSOD	736	194	4	900∼1000

3.1 Datasets and evaluation metric

Table 1 shows four commonly used remote sensing data sets, all of which are made by well-known scientific research teams in recent years. The main contents of these datasets are as follows.

In NWPU VHR-10, there are 650 positive label remote sensing images. We choose 500 images for training and the remaining 150 for testing. DOTA is a large-scale dataset including 2806 aerial images with very high-resolution. We cut the images larger than 3000 pixels to 1280×1280 pixels and get 8813 images for training and 2993 for testing. In VisDrone2018 and RSOD, we use the default training set and testing set.

We use VOC2007 11 point metric [7] to evaluate the proposed method performance, where mAP@[0.5:0.95] is the mean of mAPs which IoU thresholds from 0.5 to 0.95, step 0.05, and mAP@0.5 and mAP@0.75 are for the detail evaluation. The instructions of symbol in evaluation metric is shown in Table 2. The evaluation process is as follows:

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

The average precision (AP) of each category can be demonstrated according to precision and recall, which can be defined as:

$$AP = \frac{1}{11} \sum_{x \in MP} (x) \tag{9}$$

Table 2 Instructions of symbol in the evaluation metric

Symbol	Instruction
TP	The number of bounding boxes which are accurately detected
FP	The number of error detection
FN	The number of the ground-truth which have not been detected
MP	The maximum precision value from 0 to 1 step by 0.1 in the recall area
AP	The mean of *MP* of the 11 points
C	Collection of categories
mAP	The mean of all categories' AP

mAP is the mean of all categories' AP , as shown in (10).

$$mAP = \frac{\sum\limits_{c \in C} AP_c}{|C|} \tag{10}$$

The proposed method is implemented on MXNet and trained on a graphics workstation (2 CPUs E5-2609v4@1.70GHz, 32-GB memory and 2 NVIDIA GTX 1080TI GPUs). During training, we fit a remote sensing image each batch and train ten epochs until convergence, where an epoch completed an iteration of all training images. The learning rate is set as 0.005 in the first 2/3 epochs and 0.0005 in the last 1/3. For enough training, we augment datasets by flipping each image. The backbone networks of all methods are pre-trained on the ImageNet dataset [24] before training.

3.2 Experimental results and analysis

We compare the performance of proposed method with three state-of-the-art methods: Faster R-CNN, R-FCN and FPN. Faster R-CNN applies a region proposal network to obtain the region proposals, then these region proposals are unified to the same size by pooling operation for the later classification and regression. R-FCN uses the position-sensitive score maps to get better results of classification and position regression. FPN builds the feature pyramid of images to detect the different size objects. For a fair comparison, these state-of-the-art methods also use ResNet101 as the backbone, and the hyperparameters with the highest performance in the original papers are used during training. All the mAPs shown in the tables are converted to percentage (%).

Tables 3, 4, 5 and 6 show the performance of proposed method compared with other state-of-the-art methods on four widely used remote sensing images datasets, respectively, where the best results are shown in bold. On the whole, the proposed method performs better than other three methods at mAP@[0.5:0.95], mAP@0.5 and mAP@0.75. Compared with detecting on a single-scale feature map (Faster R-CNN, R-FCN), object detection on multi-scale feature maps (FPN and the proposed method) can assign detection tasks in detail. For the large-scale (high resolution) feature map, due to containing rich detail features, it is convenient to detect small objects. The small-scale (low resolution) feature map has strong semantics, benefiting the large objects detecting. Using the multi-scale feature maps for object detection can obtain more accurate results than the methods detecting on a single scale feature map. The results show that the methods using multi-scale feature are more than 10% higher on mAP than those using single-scale feature.

Table 3 The result of NWPU VHR-10 dataset with the same input size (800×800 pixels)

Methods	mAP$_{[0.5:0.95]}$	mAP$_{0.5}$	mAP$_{0.75}$	1	2	3	4	5	6	7	8	9	10
Faster rcnn	36.18	79.92	25.93	44.99	28.65	28.81	49.19	37.09	30.99	69.82	27.65	20.05	24.57
R-FCN	44.83	86.36	41.29	51.59	38.64	40.83	61.21	43.94	35.70	66.23	40.44	31.97	37.73
FPN$_{hard}$	55.01	87.37	62.59	64.83	51.61	60.30	67.04	61.19	43.70	75.98	45.35	29.52	50.62
FPN$_{soft}$	54.13	87.92	59.81	65.00	**53.12**	62.48	68.21	57.63	40.48	74.77	44.59	26.60	48.41
ours$_{hard}$	**57.13**	89.89	**66.11**	**66.20**	51.41	**62.62**	**68.58**	60.74	45.16	**77.47**	**51.92**	**36.83**	50.37
ours$_{soft}$	56.23	**90.08**	63.74	64.23	51.36	61.85	66.13	**62.19**	**46.73**	71.93	49.67	34.95	**53.26**

From 1 to 10 corresponding represent airplane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge and vehicle

Table 4 The result of VisDrone 2018 dataset with the same input size (1280×800 pixels)

Methods	mAP$_{[0.5:0.95]}$	mAP$_{0.5}$	mAP$_{0.75}$	1	2	3	4	5	6	7	8	9	10
Faster rcnn	15.97	27.38	16.13	10.16	1.41	7.30	35.74	24.18	24.90	10.52	11.46	27.42	6.62
R-FCN	19.10	30.02	21.61	14.79	2.12	5.63	43.29	31.12	28.51	11.08	9.58	34.89	9.99
FPN$_{hard}$	24.13	38.15	25.82	26.57	**8.91**	7.44	55.66	31.53	29.89	14.33	13.13	38.06	15.76
FPN$_{soft}$	23.36	37.31	26.08	25.18	8.87	7.49	55.02	31.82	28.89	12.00	14.44	34.77	15.08
ours$_{hard}$	**26.36**	**40.81**	**28.92**	**28.31**	8.42	**10.35**	**57.17**	**36.34**	**30.74**	**14.79**	**15.38**	**43.92**	**18.16**
ours$_{soft}$	25.01	39.15	27.90	27.41	8.45	9.97	55.61	33.57	30.34	13.63	14.07	39.99	17.08

For better training, the difficult samples (occluded objects) are dropped. From 1 to 10 corresponding represent pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus and motor

Table 5 The result of DOTA dataset with the same input size (1280×1280 pixels)

Methods	mAP$_{[0.5:0.95]}$	mAP$_{0.5}$	mAP$_{0.75}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Faster rcnn	23.94	44.31	23.36	42.19	7.88	15.91	27.66	53.48	25.92	29.39	25.74	11.64	20.69	11.45	23.93	26.90	23.53	12.77
R-FCN	26.88	48.01	26.67	48.86	10.56	22.43	28.31	61.59	29.49	25.69	30.47	16.68	25.69	17.23	17.26	25.71	27.61	15.63
FPN$_{hard}$	36.19	60.23	37.82	56.55	32.53	39.08	34.98	69.79	35.73	35.21	34.02	22.10	40.62	23.38	27.25	34.64	32.44	24.53
FPN$_{soft}$	37.58	60.66	39.91	57.25	31.50	40.24	39.13	71.54	38.55	34.77	36.13	22.83	42.35	23.02	31.99	37.33	31.23	**25.85**
ours$_{hard}$	**38.82**	**62.17**	41.91	**58.17**	32.32	41.23	**40.30**	**72.27**	39.26	**38.71**	37.56	**24.97**	42.11	22.36	**35.66**	**39.29**	34.43	23.69
ours$_{soft}$	38.70	61.46	**42.07**	57.60	**32.82**	**43.07**	38.76	71.43	**41.52**	36.80	**37.63**	24.14	**42.69**	**24.41**	28.41	39.06	**36.44**	25.71

From 1 to 15 corresponding represent plane, ship, storage-tank, baseball-diamond, tennis-court, basketball-court, ground-track-field, harbor,bridge, large-vehicle, small-vehicle, helicopter, soccer-ball-field, round-about, swimming-pool

Table 6 These methods trained on ROSD dataset with the same input size (800×800 pixels)

Methods	mAP$_{[0.5:0.95]}$	mAP$_{0.5}$	mAP$_{0.75}$	1	2	3	4
Faster R-CNN	44.40	84.94	41.84	41.13	58.42	24.18	53.86
R-FCN	48.24	83.99	49.85	51.15	53.98	24.10	63.73
FPN$_{hard}$	56.08	86.18	62.06	64.47	55.65	**33.43**	**70.79**
FPN$_{soft}$	56.08	88.03	61.79	64.64	58.83	31.02	69.81
ours$_{hard}$	56.81	88.07	**63.53**	64.06	**63.08**	29.39	70.70
ours$_{soft}$	**57.63**	**92.27**	63.19	**65.06**	63.03	33.08	69.34

From 1 to 4 corresponding represent aircraft, playground, overpass and oil tank

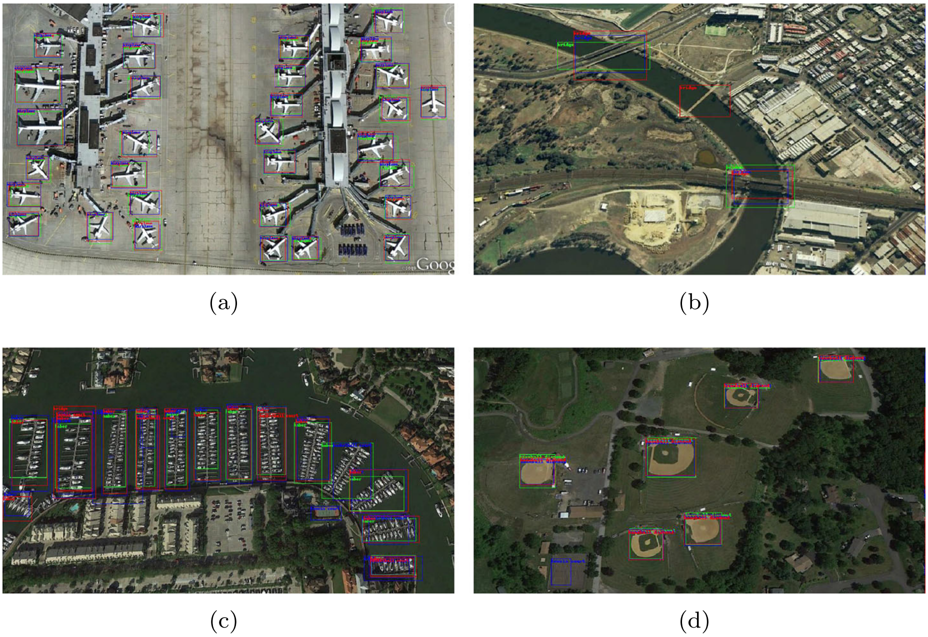
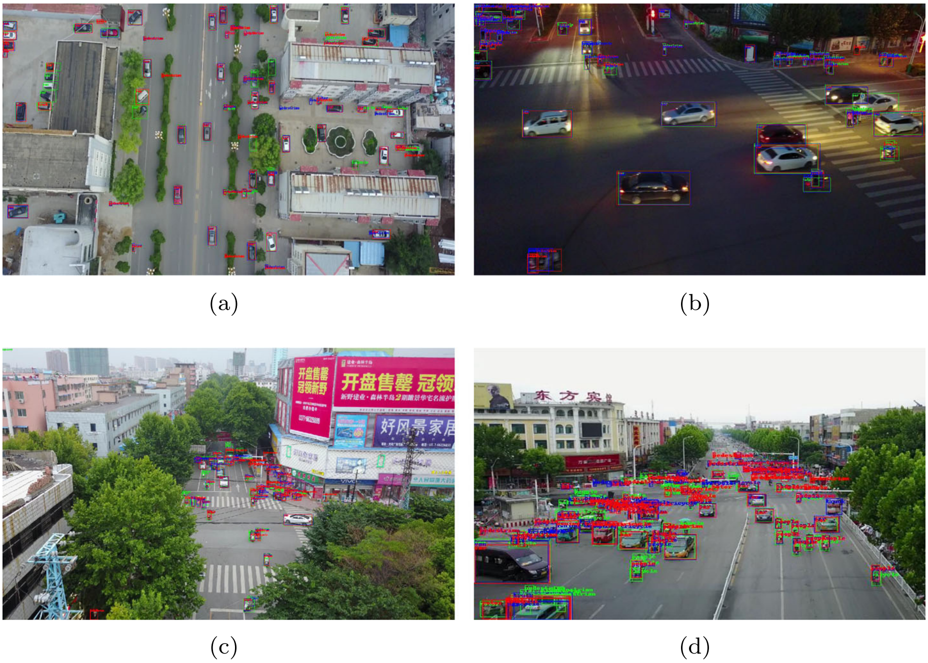

(a) (b) (c) (d)

Fig. 6 Visualized results of the NWPU VHR-10 detection results. Green rectangles: ground-truth. Red rectangles: detection results of hard fusion. Blue rectangles: detection results of soft fusion



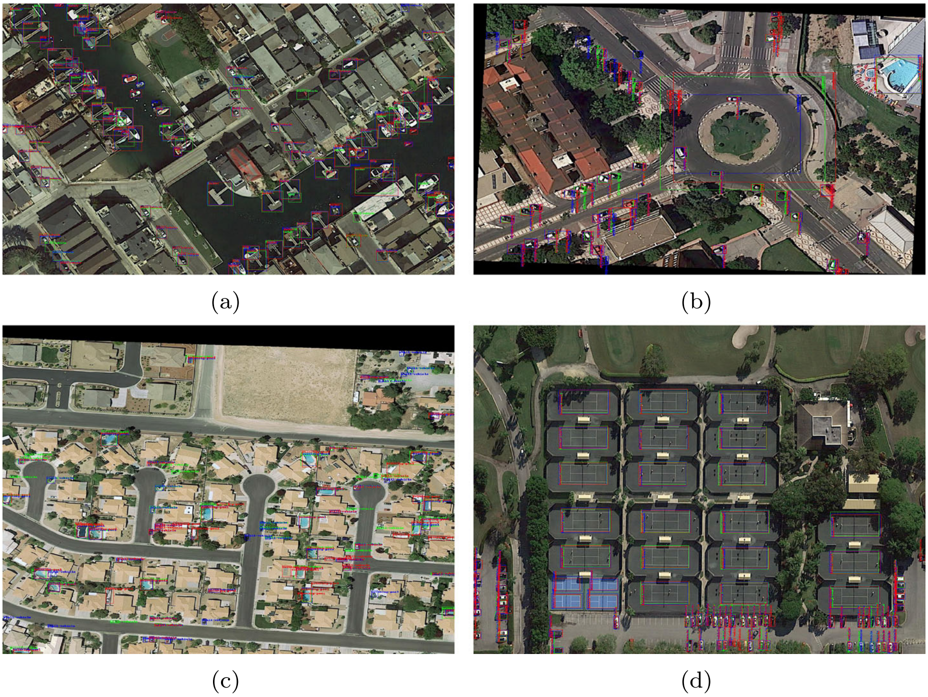
(a) (b) (c) (d)

Fig. 7 Visualized results of the VisDrone2018 detection results. Green rectangles: ground-truth. Red rectangles: detection results of hard fusion. Blue rectangles: detection results of soft fusion

Compared with FPN, the proposed method applies the FRC to provide more feature formation for the fusion, reinforcing the feature representation of objects. To prove the effectiveness of FRC, we compared the proposed method with FPN under the hard fusion strategy and soft fusion strategy, respectively. As shown in the results, the performance of proposed method in two fusion strategies has the higher AP across most categories and gets better mAP@[0.5:0.95], mAP@0.5 and mAP@0.75 than FPN. This indicates that adding the FRC step in the fusion process can effectively improve the accuracy of object detection.

For the different datasets, the performance of two fusion strategies is similar at NWPU VHR-10, DOTA and ROSD datasets. The difference of detection results on mAP@[0.5:0.95] between two fusion strategies is less than 1%. While dealing with VisDrone2018, the hard fusion strategy performs better than the soft fusion strategy on mAP and most categories' AP. And we found some potential reasons from the visible results.

Figures 6, 7, 8, and 9 exhibit visible results of the proposed method from four remote sensing images datasets respectively, where the green bounding boxes represent the ground-truth, red bounding boxes denote the detection results of hard fusion strategy, and the detection results of soft fusion strategy are drawn with blue lines. Figure 7 shows the detection results on VisDrone2018 dataset, it contains many occluded objects. In Fig. 7, the hard fusion strategy has better detection effect than the soft fusion strategy for some objects which are difficult to identify, such as the occluded cars and pedestrians in Fig. 7a, c and d. But this also makes the hard fusion strategy easy to produce some wrong results, such as the false recognition of some pedestrians in Fig. 7a. And in Figs. 6 and 8, the soft fusion

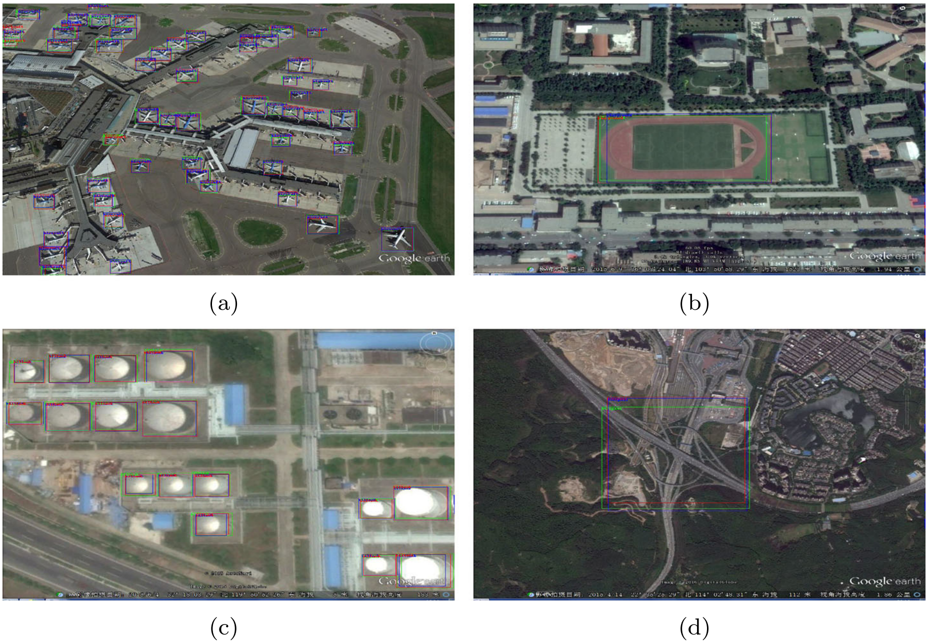

(a)

(b)

(c)

(d)

Fig. 8 Visualized results of the DOTA detection results. Green rectangles: ground-truth. Red rectangles: detection results of hard fusion. Blue rectangles: detection results of soft fusion



(a) (b) (c) (d)

Fig. 9 Visualized results of the RSOD detection results. Green rectangles: ground-truth. Red rectangles: detection results of hard fusion. Blue rectangles: detection results of soft fusion

strategy has a better detection effect on some small and crowded objects without occlusion compared with the hard fusion strategy, such as tennis courts in Fig. 6c and d, vehicles and ships in Fig. 8a and c. Unfortunately, some of these objects are not labeled, therefore, this may not make the mAP increase. The main reason for the results may be that the hard fusion strategy merges feature maps by element-wise sum directly, which preserves the features of occluded objects in feature maps. However, the soft fusion strategy uses the concat operation to merge feature maps, this may filter out some occluded object features after undergoing two 3×3 convolutions and enhance the object features without occlusion.

4 Conclusion

In this paper, the FB-FRC is proposed for remote sensing image object detection. We use the FRC to strengthen the semantics and filter out the redundant details of the high-resolution feature maps, providing more feature information for the fusion. Then two fusion strategies are designed to enhance the feature representation, further improving the detection performance. The experiments on four datasets show that the proposed method has better performance than the three state-of-the-art methods after adding the FRC. For two fusion strategies, from the experimental results, the hard fusion strategy has better performance in detecting occluded objects, while the soft fusion has better detection effect for some small and crowded objects without occlusion. In practical application, according to the situation, we can train and test two fusion strategies firstly and select a better fusion strategy.

In the future, we will focus on increasing the accuracy of position regression by adding some direction parameters and try to use the generative adversarial networks to reconstruct the occluded objects to improve the detection accuracy further.

Acknowledgements This work was supported by the National Key Research and Development Plan (No. 2016YFC0600908), the National Natural Science Foundation of China (No. U1610124, 61806206, 61572505, 61772530), the Six Talent Peaks Project in Jiangsu Province (No. 2015-DZXX-010), the Natural Science Foundation of Jiangsu Province (No. BK20180639, BK20171192) and the China Postdoctoral Science Foundation (No. 2018M642359).

References

1. Cheng G, Han J (2016) A survey on object detection in optical remote sensing images. ISPRS J Photogramm Remote Sens 117:11–28
2. Cheng G, Han J, Guo L, Qian X, Zhou P, Yao X, Xintao Hu (2013) Object detection in remote sensing imagery using a discriminatively trained mixture model. ISPRS J Photogramm Remote Sens 85:32–43
3. Cheng G, Han J, Zhou P, Guo L (2014) Multi-class geospatial object detection and geographic image classification based on collection of part detectors. ISPRS J Photogramm Remote Sens 98:119–132
4. Dai J, Yi Li, He K, Sun J (2016) R-fcn: object detection via region-based fully convolutional networks. Advances in neural information processing systems, pp 379–387
5. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE computer society conference on computer vision and pattern recognition, 2005. CVPR 2005, vol 1. IEEE, pp 886–893
6. Esmael AA, Santos JAD, Torres RDS (2018) On the ensemble of multiscale object-based classifiers for aerial images: a comparative study. Multimed Tools Appl 77(11):1–28
7. Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. Int J Comput Vis 88(2):303–338
8. Girshick R (2015) Fast r-cnn. In: Proceedings of the IEEE international conference on computer vision, pp 1440–1448
9. Girshick R, Donahue J, Darrell T, Malik J (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 580–587
10. Han J, Zhang D, Cheng G, Guo L, Ren J (2015) Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. IEEE Trans Geosci Remote Sens 53(6):3325–3337
11. Haralick RM, Shanmugam K et al (1973) Textural features for image classification. IEEE Transactions on Systems, Man, and Cybernetics (6): 610–621
12. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778
13. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems, pp 1097–1105
14. Li X, Wang S (2017) Object detection using convolutional neural networks in a coarse-to-fine manner. IEEE Geosci Remote Sens Lett 14(11):2037–2041
15. Lin T-Y, Dollár P, Girshick RB, He K, Hariharan B, Belongie SJ (2017) Feature pyramid networks for object detection. In: CVPR, vol 1, p 4
16. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: common objects in context. In: European conference on computer vision. Springer, Berlin, pp 740–755
17. Liu L, Ouyang W, Wang X, Fieguth P, Chen J, Liu X, Pietikäinen M (2019) Deep learning for generic object detection: a survey. International Journal of Computer Vision
18. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C-Y, Berg AC (2016) Ssd: single shot multibox detector. In: European conference on computer vision. Springer, Berlin, pp 21–37
19. Long Y, Gong Y, Xiao Z, Liu Q (2017) Accurate object localization in remote sensing images based on convolutional neural networks. IEEE Trans Geosci Remote Sens 55(5):2486–2498
20. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110

21. Neubeck A, Van Gool L (2006) Efficient non-maximum suppression. In: International conference on pattern recognition
22. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
23. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: towards real-time object detection with region proposal networks. pp 91–99
24. Russakovsky O, Deng J, Hao Su, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252
25. Senaras C, Ozay M, Vural FTY (2013) Building detection with decision fusion. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 6(3):1295–1304
26. Tao Q, Zhang Q, Sun S (2017) Vehicle detection from high-resolution aerial images using spatial pyramid pooling-based deep convolutional neural networks. Multimed Tools Appl 76(20):21651–21663
27. Wang C, Bai X, Wang S, Zhou J, Ren P (2018) Multiscale visual attention networks for object detection in vhr remote sensing images. IEEE Geoscience and Remote Sensing Letters
28. Xia G-S, Bai X, Ding J, Zhu Z, Belongie S, Luo J, Datcu M, Pelillo M, Zhang L (2018) Dota: a large-scale dataset for object detection in aerial images. In: Proc. CVPR
29. Xiao Z, Liu Q, Tang G, Zhai X (2015) Elliptic fourier transformation-based histograms of oriented gradients for rotationally invariant object detection in remote-sensing images. Int J Remote Sens 36(2):618–644
30. Zhu P, Wen L, Bian X, Ling H, Hu Q (2018) Vision meets drones: a challenge. arXiv:1804.07437

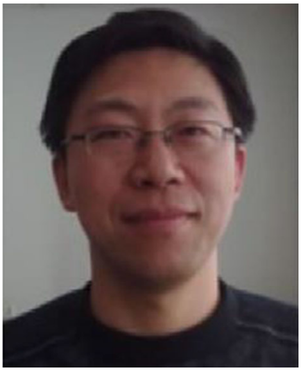
Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Dongjun Zhu received the B.E., M.S. degree in Computer Science and Engineering from Anhui University of Science and Technology in 2014 and 2017, respectively. He is currently working as a Ph.D. candidate in the School of Computer Science and Technology, China University of Mining Technology. His main research interests include deep learning, image processing and generative adversarial networks.

Shixiong Xia received the Ph.D. degree from the Department of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China. He is currently a professor and the deal with the Department of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China. He is also the director of the Mine Information Engineering Research Center in Chinese Coal Industry. His main research interests are intelligent information processing, artificial intelligence and pattern recognition, machine learning and data mining and other aspects of research.

Jiaqi Zhao received the B.Eng. degrees in intelligence science and technology in 2010, the Ph.D. degree in circuits and systems in 2017 from Xidian University, Xi'an, China. Between 2013-2014, he was an exchange Ph.D. student with the Leiden Institute for Advanced Computer Science (LIACS), University of Leiden, the Netherlands. He is currently with the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China. His current research interests include multiobjective optimization, deep learning and image processing.

Yong Zhou received the Ph.D. degree from the Department of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China. He is now a professor in China University of Mining and Technology. His research mainly focuses on data mining, machine learning and artificial intelligence.



Qiang Niu received the Ph.D. degree from the Department of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China. He is currently a professor and the associate dean with the Department of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China. He is also the deputy director of the Mine Information Engineering Research Center in Chinese Coal Industry. His main research interests are intelligent information processing, artificial intelligence and pattern recognition, machine learning and data mining and other aspects of research.

Rui Yao received the Ph.D. degree in computer science from Northwestern Polytechnical University, Xian, China, in 2013. From September 2011 to September 2012, he was a Visiting Student with the University of Adelaide, Adelaide, SA, Australia. He is currently associate professor with the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, China. His current research interests include computer vision and machine learning.

Ying Chen received the B.S. degree in software engineering from Shandong University, Weihai, China, in 2012. She is currently working toward the Ph.D. degree at the School of Computer Science and Technology, China University of Mining and Technology, Xuzhou. Her research interests include computer vision, Generative Adversarial Networksperson re-identification and visual quality assessment.

Affiliations

Dongjun Zhu[1,2] · **Shixiong Xia**[1,2] · **Jiaqi Zhao**[1,2] · **Yong Zhou**[1,2] · **Qiang Niu**[1,2] · **Rui Yao**[1,2] · **Ying Chen**[1,2]

Dongjun Zhu
djzhu@cumt.edu.cn

Jiaqi Zhao
jiaqizhao@cumt.edu.cn

Yong Zhou
yzhou@cumt.edu.cn

Qiang Niu
niuq@cumt.edu.cn

Rui Yao
ruiyao@cumt.edu.cn

Ying Chen
cheny@cumt.edu.cn

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou, 221116, China

[2] Engineering Research Center of Mine Digitization, Ministry of Education of the Peoples Republic of China, Xuzhou 221116, China