ISPRS Journal of Photogrammetry and Remote Sensing 146 (2018) 182–196

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

Deep networks under scene-level supervision for multi-class geospatial object detection from remote sensing images

Yansheng Li[a], Yongjun Zhang[a,*], Xin Huang[a], Alan L. Yuille[b]

[a] School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China
[b] Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218, USA

ARTICLE INFO

Keywords:
Multi-class geospatial object detection
Deep networks
Scene-level supervision
Discriminative convolutional weights
Class-specific activation weights

ABSTRACT

Due to its many applications, multi-class geospatial object detection has attracted increasing research interest in recent years. In the literature, existing methods highly depend on costly bounding box annotations. Based on the observation that scene-level tags provide important cues for the presence of objects, this paper proposes a weakly supervised deep learning (WSDL) method for multi-class geospatial object detection using scene-level tags only. Compared to existing WSDL methods which take scenes as isolated ones and ignore the mutual cues between scene pairs when optimizing deep networks, this paper exploits both the separate scene category information and mutual cues between scene pairs to sufficiently train deep networks for pursuing the superior object detection performance. In the first stage of our training method, we leverage pair-wise scene-level similarity to learn discriminative convolutional weights by exploiting the mutual information between scene pairs. The second stage utilizes point-wise scene-level tags to learn class-specific activation weights. While considering that the testing remote sensing image generally covers a large region and may contain a large number of objects from multiple categories with large size variations, a multi-scale scene-sliding-voting strategy is developed to calculate the class-specific activation maps (CAM) based on the aforementioned weights. Finally, objects can be detected by segmenting the CAM. The deep networks are trained on a seemingly unrelated remote sensing image scene classification dataset. Additionally, the testing phase is conducted on a publicly open multi-class geospatial object detection dataset. The experimental results demonstrate that the proposed deep networks dramatically outperform the state-of-the-art methods.

1. Introduction

Multi-class geospatial object detection from remote sensing images (Cheng et al., 2014) consists of localizing objects of interest (e.g., airplanes, bridges) on the earth's surface and predicting their categories. Compared with object detection from natural images (Everingham et al., 2010; Russakovsky et al., 2015), geospatial object detection suffers from additional challenges, including large size variations, dense distributions, and arbitrary orientations (Marcos et al., 2018) of the object instances on the earth's surface. Hence, multi-class geospatial object detection requires more specific exploration.

Motivated by the great success of deep learning (Krizhevsky et al., 2012; LeCun et al., 2015), many researchers in the remote sensing community (Cheng et al., 2016; Deng et al., 2018; Ding et al., 2018; Long et al., 2017; Zhong et al., 2018; Zou and Shi, 2018) have transferred deep networks pre-trained on large-scale natural image datasets such as ImageNet (Russakovsky et al., 2015) and MSCOCO (Lin et al., 2014), to geospatial object detection. However, these geospatial object detection methods (Cheng et al., 2016; Deng et al., 2018; Ding et al., 2018; Long et al., 2017; Zhong et al., 2018; Zou and Shi, 2018) highly depend on bounding box annotations to train or fine-tune deep networks. It is well known that bounding box annotations are time-consuming and become almost impossible when the object volume is very large. As scene-level tags are much easier to collect than bounding box annotations, the past decade has witnessed major advances in constructing remote sensing image scene datasets (Cheng et al., 2017; Li et al., 2018b; Xia et al., 2017; Yang and Newsam, 2010; Zhou et al., 2018b), but the progress has been relatively slow in building geospatial object detection datasets with accurate bounding box annotations. To alleviate the labor of bounding box annotations, this paper tries to leverage the already existing remote sensing scene datasets to provide weak supervision to train deep networks for multi-class geospatial object detection.

In the existing remote sensing scene datasets (Cheng et al., 2017; Li et al., 2018b; Xia et al., 2017; Yang and Newsam, 2010; Zhou et al.,

* Corresponding author.
E-mail addresses: yansheng.li@whu.edu.cn (Y. Li), zhangyj@whu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.isprsjprs.2018.09.014
Received 19 May 2018; Received in revised form 11 September 2018; Accepted 14 September 2018
Available online 21 September 2018

2018b), each scene contains one kind of dominant object and has varied backgrounds. Scene tags only record the category type of the dominant object in each scene, and do not contain any knowledge about the number, location, size, or orientation of the objects or backgrounds. In addition, scenes with the same tag often contain different numbers of objects with varied locations, sizes and orientations. There is no doubt that learning geospatial object detectors using the already existing remote sensing scene datasets is very cost-effective, but the learning process is very challenging because the majority of the object information is not provided.

With the aid of global pooling operations, such as global maximum pooling (GMP) and global average pooling (GAP), researchers (Zhou et al., 2014, 2016, 2018a; Oquab et al., 2015) in the computer vision community have shown that deep networks trained with only image-level/scene-level tags are informative of object locations. Unfortunately, these methods ignore the mutual information in image/scene pairs when optimizing the deep networks. In the literature, the mutual information has been widely regarded as a vital cue in the co-saliency task (Zhang et al., 2016), which also aims at collaboratively detecting common objects in multiple images. Intuitively, exploiting the mutual information in the optimization of deep networks is highly likely to improve the performance.

In this paper, we exploit the mutual information between scene pairs to train deep networks to overcome the aforementioned drawback in the existing methods (Zhou et al., 2014, 2016, 2018a; Oquab et al., 2015). With the consideration that the remote sensing image generally covers a large region and may contain many objects from multiple categories with a large size variation, we propose a multi-scale scene-sliding-voting strategy to calculate the class-specific activation maps (CAM). Furthermore, we study a set of CAM-oriented segmentation methods including a straightforward segmentation method, a diffusion-based segmentation method, and a modification-based segmentation method. As the activation maps are class-specific, it is possible to assign a suitable segmentation method for each activation map by object category, which can further improve the overall performance.

Overall, this paper trains deep networks on one large-scale remote sensing image scene classification dataset, but the learned deep networks are tested on a different multi-class geospatial object detection dataset. As can be seen, the learning supervision is extremely weak as only scene-level tags are taken as supervision and the training and testing data comes from different tasks and datasets. Even under this extreme setting, our proposed method still yields promising results, and outperforms the baselines (Oquab et al., 2015; Zhou et al., 2016). The main contributions of this paper can be summarized as follows:

- This paper proposes a new framework to train deep networks under scene-level supervision for multi-class geospatial object detection. To the best of our knowledge, this is the first method that considers the mutual information between scene pairs to train deep networks for the weak supervision scenario.
- Taking the characteristics of remote sensing images into account, we present a multi-scale scene-sliding-voting strategy to calculate the CAM of remote sensing images.
- This paper gives a set of CAM-oriented segmentation methods and analyzes their application cases, which makes selecting the best segmentation method for each activation map by object category possible.
- Last but not least, this paper reveals the use of knowledge transfer between different tasks and datasets using deep networks.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 specifically introduces how to train deep networks under scene-level supervision. Section 4 shows the multi-class geospatial object detection method using the learned deep networks under scene-level supervision. Section 5 reports the experimental results. Finally, Section 6 gives the conclusion of this paper.

2. Related work

In this section, we briefly review the most relevant works in the literature that include weakly supervised deep networks and multi-class geospatial object detection.

To alleviate the labor of bounding box annotations, pioneers in computer vision exploit scene-level or image-level tags as weak supervision for localizing objects in images or scenes. More specifically, Pinheiro and Collobert (2015) and Cinbis et al. (2017) combined multi-instance learning with deep convolutional features to localize objects. Oquab et al. (2014) proposed a method to localize objects by evaluating the output of deep networks on multiple overlapping patches. Although promising results have been reported, these methods still cannot be trained in an end-to-end way. In the most recent years, region proposals-based methods using weak supervision (Bilen and Vedaldi, 2016; Tang et al., 2017) have been proposed to address object detection. With the aid of global pooling operations, Oquab et al. (2015) and Zhou et al. (2016) trained deep networks in an end-to-end manner under weak supervision for class-specific object detection. In the most recent years, this idea has been widely explored in semantic segmentation (Chen et al., 2018; Kolesnikow and Lampert, 2016) and saliency detection (Wang et al., 2017). As these methods were originally designed for natural images, they cannot be directly used for remote sensing image analysis as they have insufficient capability to handle the challenges in remote sensing images, which contain complex backgrounds and densely distributed objects with arbitrary orientations.

In the early days, many variants of hand-crafted features have been explored to detect multi-class geospatial objects (Cheng et al., 2013, 2014; Xiao et al., 2015) under the supervision of bounding box annotations. Afterwards, many researchers (Cheng et al., 2016; Long et al., 2017; Zou and Shi, 2018) transferred deep networks pre-trained on large-scale natural image datasets to the geospatial object detection task. Although these methods achieved improved performance, they (Cheng et al., 2016; Long et al., 2017; Zou and Shi, 2018) still require bounding box annotations of geospatial objects to fine-tune the transferred deep networks. To alleviate the dependence on bounding box annotations, Han et al. (2015) proposed a probabilistic framework to jointly integrate saliency, interclass compactness, and interclass separability to initialize training instances from remote sensing images with binary labels indicating whether one image contains the objects of interest or not. In addition, the training instances were further utilized to iteratively learn object detectors. As a first effort, this approach achieved promising results on single-class geospatial object detection but could not be readily extended to the multi-class case. As reviewed in Cheng and Han (2016), how to leverage weak supervision to address multi-class geospatial object detection needs further exploration.

3. Learning deep networks under scene-level supervision

In this section, we first analyze the vulnerability of the existing weak supervision based deep networks (Oquab et al., 2015; Zhou et al., 2014, 2016, 2018a) from an architectural perspective. With the aid of a global pooling operation (e.g., GMP or GAP), existing weak supervision based deep networks adopt the architecture depicted in the second stage in Fig. 1 to learn convolutional weights and class-specific activation weights in an end-to-end manner. Due to the usage of the global pooling operation, there is only a very weak connectivity between the scene tag and convolutional layers. To facilitate understanding, we give a toy example to explain why the global pooling operation yields the weak connectivity and show the drawback of this weak connectivity in Fig. 2. In the case of the forward propagation shown in Fig. 2(a), the spatial units of each channel in the last convolutional layer are aggregated into one single unit in the aggregation feature vector. Accordingly, in the case of the backward gradient propagation shown in Fig. 2(b), the gradient value of each unit in the aggregation feature vector is equally divided into the spatial units of each corresponding

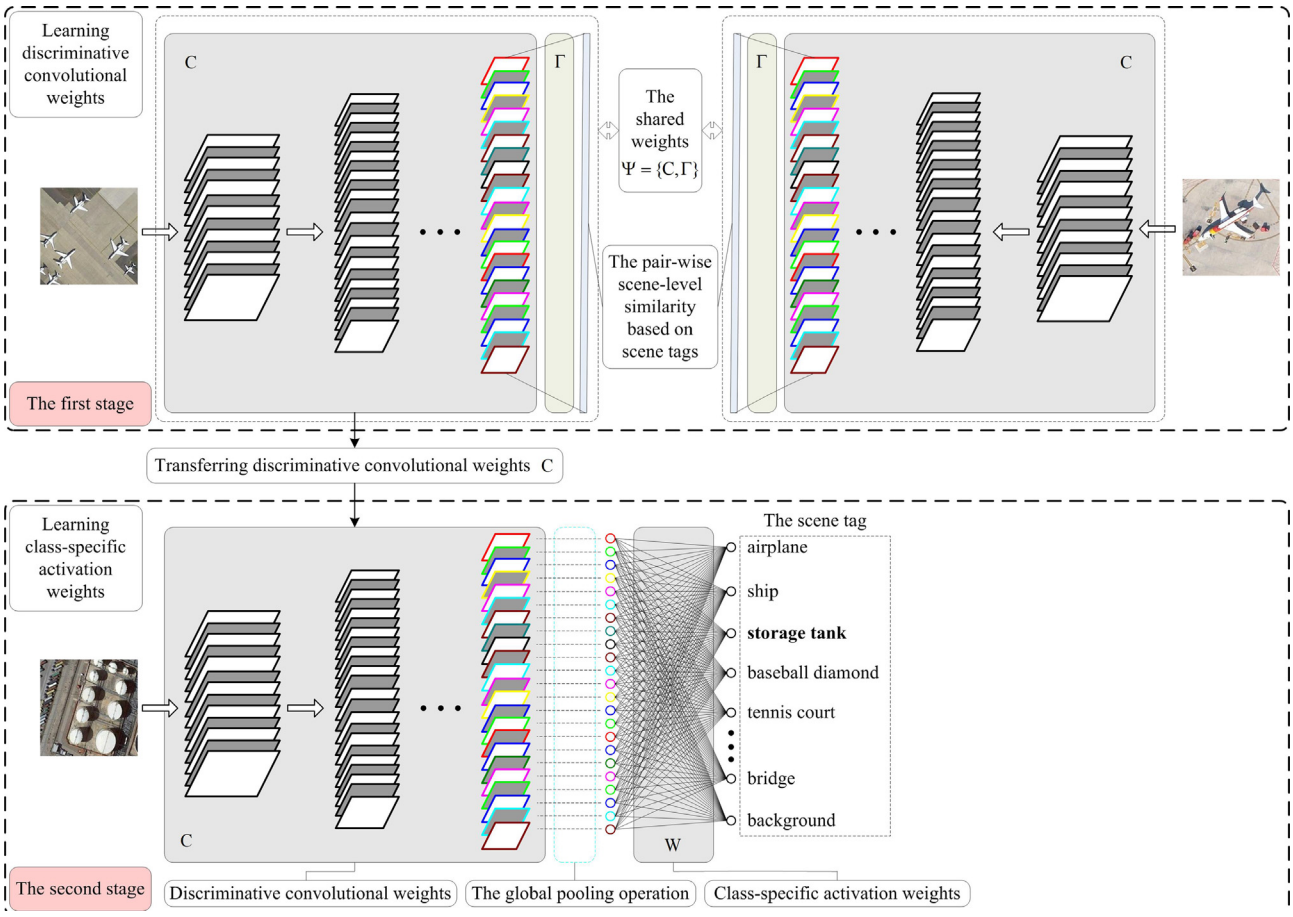
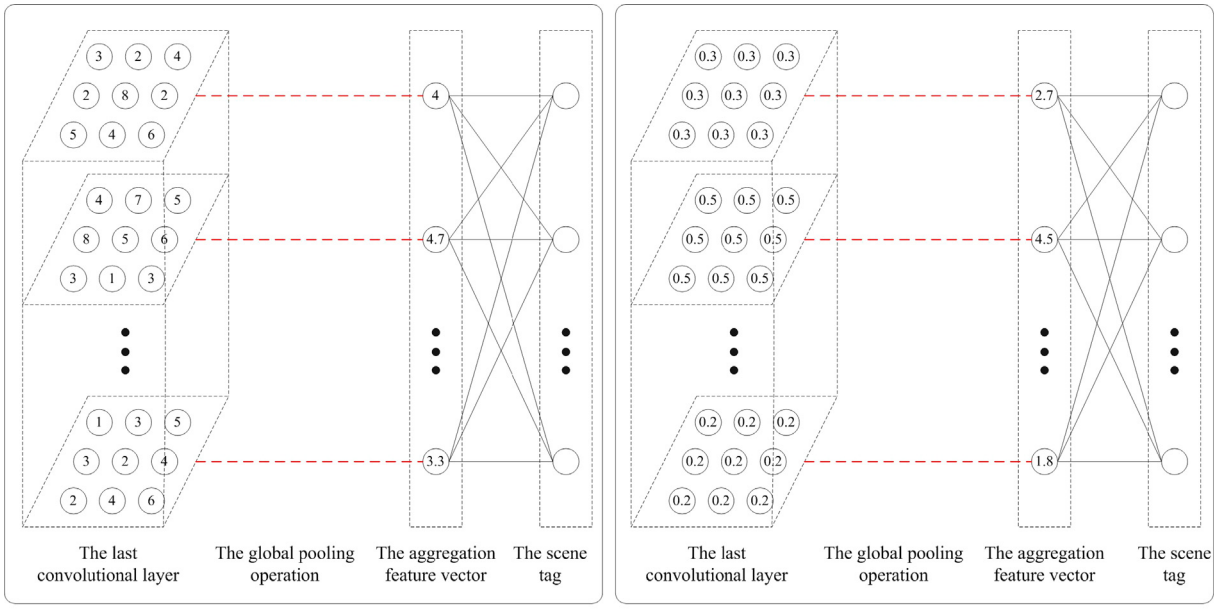


Fig. 1. The workflow for training deep networks under scene-level supervision. The workflow includes two stages: the first stage learns discriminative convolutional weights by mining the mutual information between scene pairs; the second stage learns class-specific activation weights based on the point-wise scene tags by optimizing a simple convex objective function.



(a) The forward propagation example (b) The backward propagation example

Fig. 2. A toy example to show the weakness of the end-to-end architecture with the global pooling operation. In this example, the global average pooling is adopted.

channel in the last convolutional layer, which impairs perceiving the spatial variance of each channel. As a consequence, this architecture would partly decay the update of the convolutional layers and would fail to learn powerful convolutional weights, which further hurts the overall performance.

To overcome the aforementioned limitation in the existing methods (Oquab et al., 2015; Zhou et al., 2014, 2016, 2018a), this paper proposes an optimization method based on two stages, as illustrated in Fig. 1, to successively train convolutional weights and class-specific activation weights. As depicted in Fig. 1, the two stages have different architectures, but both take the scene tags as the supervision. The first stage aims at learning discriminative convolutional weights through exploiting the mutual information between scene pairs. After fixing the discriminative convolutional weights, the second stage learns the class-specific activation weights to gain category information of the objects. In the following, we will describe the two stages in detail.

3.1. Learning discriminative convolutional weights

As depicted in the upper part of Fig. 1, the first stage of the proposed framework adopts Siamese-like networks, which consist of twin networks accepting distinct inputs, and computes the similarity between the highest-level feature representations. Different from the weak connectivity of the global pooling operation, the highest-level feature representation is fully connected with the last convolutional layer by dense weights in this stage. Siamese networks were first proposed to solve the signature verification problem (Bromley et al., 1993). Afterwards, they were utilized in the one-shot object recognition task (Koch et al., 2015). In contrast to the existing methods (Bromley et al., 1993; Koch et al., 2015), which mainly focused on object-level analysis, this paper extends this concept to scene understanding and aims to learn discriminative convolutional weights by mining the mutual information between scene pairs. As illustrated in the upper part of Fig. 1, the left and right scenes come from the same scene category (i.e., the airplane category). The left scene contains multiple small airplanes, but the right scene contains only one large airplane. If the Siamese networks can successfully perceive that the left and right scenes come from the same category, this means that the networks possess the discriminative ability to perceive common objects even with different sizes and orientations in the two scenes. Intuitively, the pair-wise scene-level similarity pursuit benefits by outputting discriminative networks, which possess the scale-invariant and rotation-invariant abilities for perceiving objects. We introduce the implementation details as follows.

Given a remote sensing image scene dataset $\{(S_i, y_i)|i = 1, 2, \cdots, N\}$ where S_i denotes the scene and y_i stands for its scene tag, the similarity matrix $\Theta^1 \in R^{N\times N}$ of the scene dataset is specifically defined by $\Theta^1_{i,j} = 1$, *if* $y_i = y_j$ and $\Theta^1_{i,j} = 0$, *if* $y_i \neq y_j$. Let $\mathbf{\Psi} = \{\mathbf{C}, \mathbf{\Gamma}\}$ denote all of the weights of the Siamese networks, where $\mathbf{C}$ stands for the weights of the hierarchical convolutional layers, and $\mathbf{\Gamma}$ denotes the weights of the fully connected layer. We note that, here, the fully connected layer has dense connectivity with the preceding convolutional layer by dense weights. Different from the weak connectivity of the global pooling operation, the dense connectivity benefits perceiving the spatial variance of each channel in the preceding convolutional layer by backward gradient propagation and further achieving the sufficient update of all the convolutional layers. Based on the similarity matrix of the training dataset, the Siamese networks $\mathbf{\Psi} = \{\mathbf{C}, \mathbf{\Gamma}\}$ can be learned by optimizing the objective function as follows:

$$\begin{aligned}\min_{\mathbf{\Psi}} \boldsymbol{J} &= \sum_{\Theta_{i,j}\in\Theta} \sum_{k=1}^{2} (-\Theta^k_{i,j} \log p(\Theta^k_{i,j}=1|\mathbf{F})) + \lambda \cdot \sum_{i=1}^{N} \|\,|\mathbf{f}_i| - \mathbf{1}\|_2^2 \\ &= \sum_{\Theta_{i,j}\in\Theta} (-\Theta^1_{i,j}\Upsilon_{i,j} + \log(1 + e^{\Upsilon_{i,j}})) + \lambda \cdot \sum_{i=1}^{N} \|\,|\mathbf{f}_i| - \mathbf{1}\|_2^2 \end{aligned} \tag{1}$$

where $\Theta^2_{i,j} = 1 - \Theta^1_{i,j}$, $\mathbf{f}_i = \varphi(S_i; \mathbf{\Psi})$ denotes the highest-level feature vector of one scene S_i through the mapping of the hyper-parameters $\mathbf{\Psi}$. $p(\Theta^1_{i,j} = 1|\mathbf{F})$ denotes the pair-wise likelihood function and is calculated by the sigmoid function $\sigma(\Upsilon_{i,j}) = 1/(1 + e^{-\Upsilon_{i,j}})$. $\Upsilon_{i,j} = (\mathbf{f}_i^T \mathbf{f}_j)/(\rho \cdot l)$, denotes the dimensional of the feature vector $\mathbf{f}$, and ρ is a similarity factor. $p(\Theta^2_{i,j} = 1|\mathbf{F}) = 1 - p(\Theta^1_{i,j} = 1|\mathbf{F})$. λ denotes the regularization coefficient. As in (Li et al., 2018a; Liu et al., 2016), the regularization term mainly works for feature normalization, which benefits by improving the stability of the pair-wise likelihood calculation.

With respect to the feature vector $\mathbf{f}_i$, the first part of the objective function in Eq. (1) is differentiable, but the second part is non-differentiable due to the presence of the absolute operator. As suggested in (Liu et al., 2016), we calculate the derivatives $\partial \boldsymbol{J}/\partial \mathbf{f}_i$ on two intervals using Eq. (2). By utilizing the widely adopted back-propagation algorithm, the derivatives $\partial J/\partial \mathbf{f}_i$ are further utilized to update the whole weights $\mathbf{\Psi}$.

$$\frac{\partial J}{\partial \mathbf{f}_i^m} = \begin{cases} \sum\limits_{j:\Theta_{i,j}\in\Theta} (\sigma(\Upsilon_{i,j}) - \Theta^1_{i,j})\mathbf{f}_j^m + 2\lambda(\mathbf{f}_i^m - 1), & \mathbf{f}_i^m \geqslant 0 \\ \sum\limits_{j:\Theta_{i,j}\in\Theta} (\sigma(\Upsilon_{i,j}) - \Theta^1_{i,j})\mathbf{f}_j^m + 2\lambda(\mathbf{f}_i^m + 1), & \mathbf{f}_i^m < 0 \end{cases} \tag{2}$$

where m is the element index of the feature vector $\mathbf{f}$.

Removing the fully connected layer $\mathbf{\Gamma}$ in $\mathbf{\Psi}$, the discriminative convolutional weights $\mathbf{C}$ in $\mathbf{\Psi}$ are used to train the class-specific activation weights, as follows.

3.2. Learning class-specific activation weights

By transferring the discriminative convolutional weights $\mathbf{C}$ learned in the first stage and fixing them, the second stage then learns the class-specific activation weights; a visual illustration of the second stage is shown in the bottom part in Fig. 1. For a given image scene S_i, $\mathbf{T}_i^k(x, y) = \varphi(S_i; \mathbf{C})$ denotes the feature of the last convolutional layer, where k stands for the depth channel, and (x, y) denotes the spatial location.

Despite its apparent simplicity, the global pooling operation (Oquab et al., 2015; Zhou et al., 2016) has been successfully utilized to learn class-specific activation weights and is also adopted in this paper. By global pooling $\mathbf{T}_i^k(x, y)$ per channel, $\mathbf{T}_i^k$ denotes the scalar activation value of $\mathbf{T}_i^k(x, y)$ at the k-th channel where GMP (Oquab et al., 2015) and GAP (Zhou et al., 2016) are two candidates for conducting the global pooling function. In this experimental setup, each training image scene only contains one kind of object and thus has one unique object category label. Hence, the softmax-based cross-entropy loss function is taken to model the connectivity between the global pooling result and the scene tag, and is specified by Eq. (3).

$$\min_{W} E = \sum_{i=1}^{N} \sum_{c} \left(p(y_i = c) \cdot \log\left(\frac{\exp(w_k^c \cdot \mathbf{T}_i^k + w_0^c)}{\sum_c \exp(w_k^c \cdot \mathbf{T}_i^k + w_0^c)} \right) \right) \tag{3}$$

where c denotes the scene category. As also noted in (Oquab et al., 2015; Zhou et al., 2016), w_k^c indicates the contribution of $\mathbf{T}^k$ (i.e., the k-th channel) for category c.

Using the discriminative convolutional weights $\mathbf{C}$ that were learned in the first stage, the second stage learns the class-specific activation weights $\mathbf{W} = \{w_k^c\}$ by optimizing the simple convex function in Eq. (3).

As an alternative baseline, the derivatives of the convex function in Eq. (3) can be utilized to update the class-specific activation weights $\mathbf{W}$ and the convolutional weights $\mathbf{C}$ in an end-to-end manner. However, this approach, which has been adopted in (Oquab et al., 2015; Zhou et al., 2014, 2016, 2018a), may fail to learn powerful convolutional weights $\mathbf{C}$ as discussed in first paragraph of Section 3. This is why we learn the convolutional weights $\mathbf{C}$ and the class-specific activation weights $\mathbf{W}$ in two separate stages instead of using the popular end-to-end way to learn all of the weights of the deep networks in one effort.

Next, we will introduce how to conduct multi-class geospatial object detection using the learned deep networks, whose parameters are

composed of the discriminative convolutional weights **C** and the class-specific activation weights **W**.

4. Multi-class geospatial object detection with the learned deep networks under scene-level supervision

Section 4.1 introduces how to automatically generate the CAM of large remote sensing images using the learned deep networks in Section 3. In Section 4.2, we study a set of CAM-oriented segmentation methods to localize objects from the CAM. In addition, we give a brief summary of our proposed multi-class geospatial object detection approach in Section 4.3.

4.1. Computing the class-specific activation maps via multi-scale scene-sliding-voting

Given one scene S, let $\mathbf{T}^k(x, y) = \varphi(S; \mathbf{C})$ denotes the feature of the last convolutional layer based on the discriminative convolutional weights **C** learned in Section 3.1. We define the probability $p(y = c|S)$ that the scene S contains objects of category c by Eq. (4) and the activation map $\mathbf{M}_c^S(x, y)$ of the scene S for object category c by Eq. (5).

$$p(y = c|S) = \frac{\exp(w_k^c \cdot \mathbf{T}^k + w_0^c)}{\sum_c \exp(w_k^c \cdot \mathbf{T}^k + w_0^c)} \tag{4}$$

$$\mathbf{M}_c^S(x, y) = \sum_k w_k^c \cdot \mathbf{T}^k(x, y) + w_0^c \tag{5}$$

where w_0^c is the bias coefficient of object category c.

Generally speaking, remote sensing images cover a large region, and their spatial resolution also changes significantly, which makes the size of objects in the image vary greatly. To tackle these challenges, we present a multi-scale scene-sliding-voting method to generate the CAM of large remote sensing images.

We first introduce the single-scale scene-sliding-voting strategy, as illustrated in Fig. 3. Given a large remote sensing image I, we obtain a set of overlapped scenes $\{S_1, S_2, \cdots, S_n\}$ by sliding windows from left to right and top to bottom. We calculate the probability and the CAM of each scene from $\{S_1, S_2, \cdots, S_n\}$ using Eqs. (4) and (5). The probability $p(y = c|I)$ that the image I contains objects of category c can be calculated by Eq. (6). Through mosaicking the scene-level CAMs where the overlapped regions are fused by the maximum voting, the activation map $\mathbf{M}_c^I(x, y)$ of the image I for object category c can be calculated by Eq. (7).

$$p(y = c|I) = \max(p(y = c|S_1), p(y = c|S_2), \cdots, p(y = c|S_n)) \tag{6}$$

$$\mathbf{M}_c^I(x, y) = \text{Mosaic}(\mathbf{M}_c^{S_1}(x, y), \mathbf{M}_c^{S_2}(x, y), \cdots, \mathbf{M}_c^{S_n}(x, y)) \tag{7}$$

where the visual mosaicking process can refer to Fig. 3.

To address the multi-scale nature of objects, we construct an image pyramid and fuse the results on the image pyramid. Given a testing image I, we construct the pyramid $\{I^1, I^2, \cdots, I^m\}$ by downsampling and upsampling I. Based on Eqs. (6) and (7), we get the pyramid probabilities $\{p(y = c|I^1), p(y = c|I^2), \ldots, p(y = c|I^m)\}$ and the pyramid CAMs $\{\mathbf{M}_c^{I^1}(x,y), \mathbf{M}_c^{I^2}(x,y), \cdots, \mathbf{M}_c^{I^m}(x,y)\}$. Furthermore, we get the multi-scale probability $p(y = c|I^{ms})$ by a maximum fusion (i.e., take the maximum probability across all scales) of the pyramid probabilities. By resizing the activation map at each scale to the size of the original image I, we calculate the multi-scale CAM $\mathbf{M}_c^{I^{ms}}(x, y)$ by a maximum fusion (i.e., take the maximum value at each spatial location across all scales) of the activation maps at different scales. We note that maximum fusion is adopted here to avoid missing any true positives.

4.2. Detecting objects by segmenting the class-specific activation maps

As mentioned earlier, the CAM are sufficient for indicating the location of objects; thus, we can detect objects by segmenting the CAM. Given one testing image I, if $p(y = c|I^{ms}) > \mathrm{O}$, where O is an empirical probability constant, we segment the activation map $\mathbf{M}_c^{I^{ms}}(x, y)$ to detect objects of category c; otherwise, we skip the segmentation module because I probably does not contain objects of category c.

Based on the dramatically varied characteristics of geospatial objects, we consider a set of CAM-oriented segmentation methods. In the experimental section, we give quantitative evaluations of these methods, which may benefit developing practical geospatial object

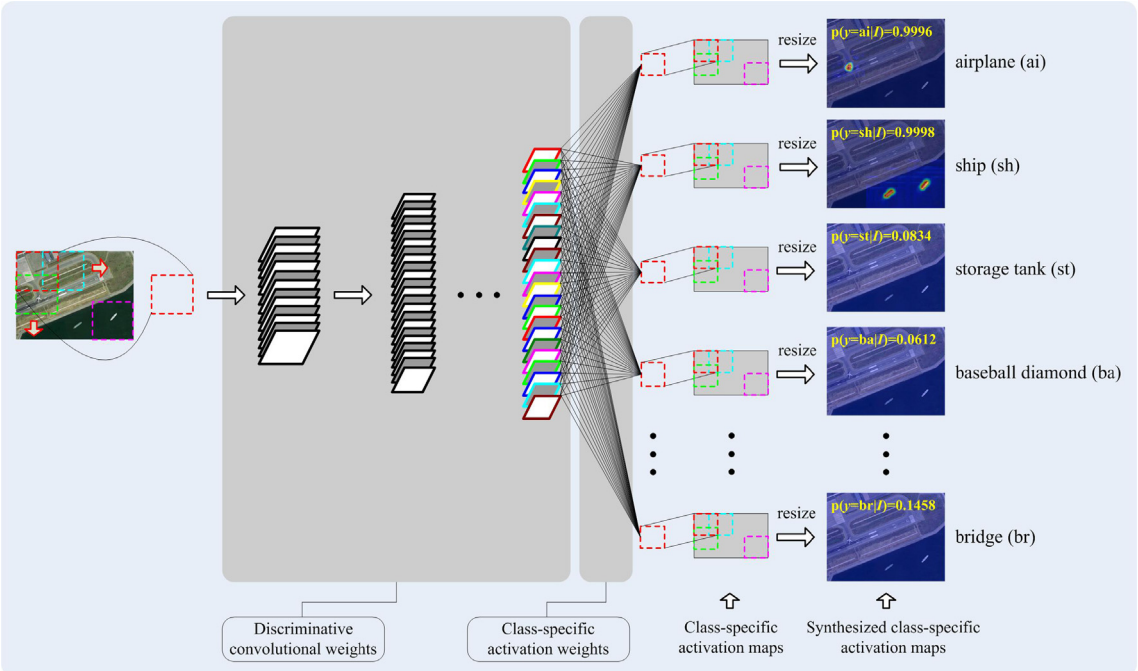


Fig. 3. A visual example of calculating the CAM of one large remote sensing image by the single-scale scene-sliding-voting strategy. Each synthesized activation map is the combination of the original image and the activation map for a particular object category. Throughout this paper, the activation maps are visually shown as heat maps. In the synthesized activation maps, the deeper the red color, the larger the probability of object presence.

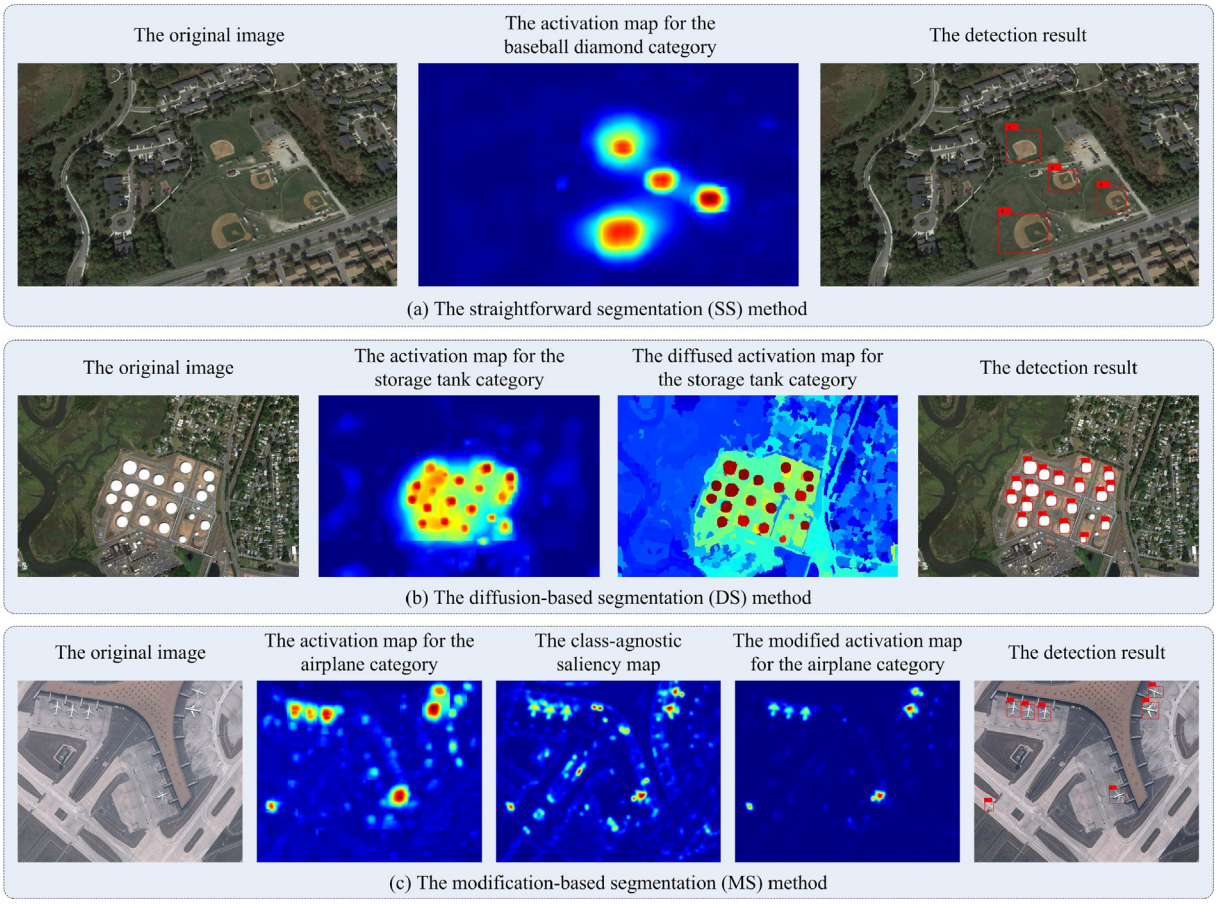


Fig. 4. The CAM-oriented segmentation methods for localizing objects from the CAM. (a) shows the intermediate results in SS, (b) gives the intermediate results in DS, and (c) shows the intermediate results in MS.

detection applications by configuring the segmentation strategy by object category.

4.2.1. The straightforward segmentation method

Some types of geospatial objects (e.g., the baseball diamond, the bridge) are generally far away from each other, we can easily segment the CAM to localize these objects. As in Zhou et al. (2016), we use a straightforward segmentation method that sets thFactor·maxVal to be the threshold to segment objects from the activation map, where maxVal denotes the max value of the activation map, and thFactor is a constant. Due to its simplicity, we call this technique the straightforward segmentation (SS) method. Fig. 4(a) gives a visual example of the input and output of SS.

4.2.2. The diffusion-based segmentation method

Unlike objects in natural images, many types of geospatial objects (e.g., the storage tank, the tennis court) are densely distributed. Therefore, a direct segmentation (i.e., SS) of the activation map can only coarsely locate the regions that contain objects but fails to estimate accurately the size or number of objects. To address this problem, we smooth the initial activation map by superpixel-based diffusion to suppress the background and then detect objects from the smoothed result. More specifically, we use a recent diffusion method (Dou et al., 2017) to diffuse the initial activation map and then use SS to detect objects from the diffused activation map. Fig. 4(b) gives a visual comparison between the initial activation map and the diffused result. In our implementation, the superpixel generation method and the critical parameters were designed based on (Dou et al., 2017). We call this technique the diffusion-based segmentation (DS) method.

4.2.3. The modification-based segmentation method

For densely distributed geospatial objects (e.g., airplane) with rich structures, DS does not work well, as rich structures result in low-quality superpixels, which hurts the diffusion performance of DS. To segment of these types of geospatial objects, we use a class-agonistic saliency map to modify the initial activation map and then apply SS to segment objects from this map. As is well known, saliency methods aim at enhancing all salient regions (e.g., structural regions, complex textures, and high-contrast regions). The classical Fourier transform-based saliency method (Guo et al., 2008), which has a low computation complexity, was adopted to generate the class-agonistic saliency map. More specifically, the modification was implemented by pixel-wise multiplication of two maps. An intuitive validation for this idea is shown in Fig. 4(c). This is called the modification-based segmentation (MS) method.

4.3. Overview of the proposed multi-class geospatial object detection approach

As aforementioned, our proposed multi-class geospatial object detection approach involves many contents and seems to be very complicated. To make this easier to follow, we briefly summarize the training and testing phases of our proposed approach in Algorithm 1. As depicted in Algorithm 1, the weights of deep networks are optimized in the training phase and fixed in the testing phase. In addition, the first step in the testing phase depends on the weights of deep networks, but the second step in the testing phase does not use the deep networks.

Algorithm 1. The Proposed Multi-Class Geospatial Object Detection Approach

The training phase

Input: Remote sensing image scene dataset $\{(S_i, y_i)|i = 1, 2, \cdots, N\}$

Output: Discriminative convolutional weights **C**; class-specific activation weights **W**

1: Learn discriminative convolutional weights **C** by optimizing Eq. (1) in Section 3.1

2: Learn class-specific activation weights **W** by optimizing Eq. (3) in Section 3.2

The testing phase

Input: The testing remote sensing image I; discriminative convolutional weights **C**; class-specific activation weights W

Output: Bounding boxes of multi-class objects that the testing image I contains

1: Calculate the class-specific activation maps of I based on **C** and **W** using the multi-scale scene-sliding-voting strategy in Section 4.1

2: Extract multi-class objects from the class-specific activation maps using one of the recommended segmentation methods {SS, DS, MS} in Section 4.2

5. Experimental results and discussions

Section 5.1 first introduces the experimental setup of this paper. From the prediction perspective, Section 5.2 uses pixel-level metrics to evaluate the class-specific object prediction performance of the CAM. Using the widely adopted bounding-box-level metrics for multi-class geospatial object detection, Section 5.3 reports quantitative detection results of our method as well as some baselines. Finally, Section 5.4 shows the limitations of this work in this paper and gives suggestions on how to further improve this work.

5.1. Experimental setup

In this section, we specifically introduce the evaluation datasets in Section 5.1.1, and we give the implementation details of our proposed method in Section 5.1.2.

5.1.1. Evaluation datasets

Compared with other remote sensing image scene datasets such as UC-Merced (Yang and Newsam, 2010) with 21 scene categories and AID (Xia et al., 2017) with 30 scene categories, NWPU-RESISC45 (Cheng et al., 2017) is much larger in terms of the number of scene categories and the number of scene samples. Therefore, NWPU-RESISC45 was adopted as the training scene dataset of this work. In the literature, existing geospatial object detection datasets include the TAS aerial car detection dataset (Heitz and Koller, 2008) with one kind of object, the LEVIR dataset (Zou and Shi, 2018) with three kinds of objects, the OAOP dataset (Long et al., 2017) with four kinds of objects, and the NWPU VHR-10 dataset (Cheng et al., 2014) with 10 kinds of objects. In this work, the NWPU VHR-10 dataset minus the vehicle class, which is called NWPU VHR-9, is taken as the testing geospatial object detection dataset because there does not exist a vehicle scene category in the training scene dataset. This work takes NWPU VHR-9 as the testing dataset, not only because it contains more object types but also because all of its object types have a semantic category correspondence with the training scene dataset.

In Fig. 5, the first 9 scene categories in NWPU-RESISC45 semantically correspond to the geospatial object types in NWPU VHR-9, and the other scene categories in NWPU-RESISC45 are used as background supervision. To augment the training dataset, we rotate each scene by 90°, 180°, and 270°. This augmentation enlarged the training dataset 4 times. Finally, the augmented training dataset contains 45 scene categories and each category contains 2800 samples. The testing dataset (i.e., NWPU VHR-9) has 565 large testing images. In addition, the testing images in NWPU VHR-9 have different sizes, and one image may contain objects from multiple categories. More specifically, Fig. 6 illustrates the testing dataset.

5.1.2. Implementation details

In our implementation, we follow the architecture of the VGG-F net (Chatfield et al., 2014). The module for learning discriminative convolutional weights includes the general parameters and the special parameters. As far as the general parameters, we follow the typical setting in deep learning. More specifically, we set the learning rate to 0.01, and the weight decay is set to 0.0005. For the special parameters in our proposed deep learning framework, we set them based the experience in our previous work (Li et al., 2018a). Specifically, the similarity factor ρ and the regularization coefficient λ are empirically set to 0.5 and 10, respectively. Without doubt, the performance of our proposed method can be further improved if we further tune these parameters. We don't do that here because training deep networks is very time-consuming. In the future, we may evaluate these parameters when we have sufficient computational resources.

In the module for generating the CAM, we consider multiple image scaling coefficients {0.25, 0.50, 1.0, 1.5} to construct the image pyramid. In each scale scene-sliding-voting process, the sliding scene size was set to 256 by 256, and the sliding step was set to 128. The parameter settings in this module mainly aim at pursuing the goal that the sliding scenes in the testing image have the similar scale with the training scenes. Hence, the readers can configure the parameters of this module for their specific applications based on this hint.

As far as the parameters in the module for segmenting the CAM, we specifically analyze their sensitivity and give the recommendation settings in Section 5.3.3.

All approaches including our proposed approach and other baselines are implemented by MATLAB and conducted on a Dell station with 8 Intel Core i7-6700 processors, 32 GB of RAM, and the NVIDIA GeForce GTX 745.

5.2. Performance evaluation of the class-specific activation maps

To directly verify the class-specific object prediction performance of the CAM, this section adopts the pixel-level metrics in the saliency evaluation task (Wang et al., 2017), which calculates the similarity between the estimated confidence map and the ground truth map. Section 5.2.1 introduces the evaluation metrics, and Section 5.2.2 gives the quantitative comparison result with some baselines.

5.2.1. Evaluation measures

Based on the bounding box annotations of NWPU VHR-9, we generate the class-specific binary ground truth map (CGTM). By segmenting the CAM at different thresholds, we calculate the pixel-level Precision and Recall values by comparing the segmented CAM with CGTM per object category. Furthermore, the Precision-Recall curve (Wang et al., 2017) is taken to evaluate the class-specific object prediction performance of the CAM.

5.2.2. Comparison with some baselines

Because this is the first time that the idea has been applied to the remote sensing domain, there do not exist any methods that are specially designed for this idea. To verify the superiority of our proposed method, we design some baselines by employing the other competitive candidates of the major modules of our proposed method where the considered major modules include the training method, the global pooling operation and the voting strategy as shown in Table 1.

Fig. 5. The remote sensing image scene training dataset (i.e., the NWPU-RESISC45 dataset). This dataset has 45 scene categories and there are 700 image scenes per category. Two random scenes are shown for each category.

We specifically introduce the candidates of the training method, the global pooling operation and the voting strategy as follows. As mentioned in Section 3, this paper learns discriminative convolutional weights **C** based on the pair-wise scene-level similarity constraint (PSS) and learns class-specific activation weights **W** based on the point-wise scene category prediction constraint (PCP) in two separate stages. To show the superiority of this training method with two stages, we consider another two candidates. The first candidate uses the derivatives of the convex function in Eq. (3) to train the convolutional weights **C** and the class-specific activation weights **W** in an end-to-end manner based on PCP, similar to (Oquab et al., 2015; Zhou et al., 2016). The second candidate transfers the already-trained convolutional weights **C** on the ImageNet, and trains the class-specific activation weights **W** on the remote sensing image scene dataset based on PCP. As far as the global pooling operation, we consider two popular ones, GMP (Oquab et al., 2015) and GAP (Zhou et al., 2016), in our implementation. In addition to our proposed scene-sliding-voting (SSV) strategy shown in Section 4.1, we take the image-level-voting (ILV) method as a candidate to generate the CAM to verify the superiority of SSV, where ILV means directly applying the learned deep networks on the whole image without any scene partition. To summarize, four baselines and four variants of our proposed method are shown in Table 1.

In Fig. 7, we report the Precision-Recall curves of our proposed method and some baselines for detecting 9 object categories. As shown in Fig. 7, our proposed CAM with PSS + GAP + SSV simultaneously achieves the best object prediction performance in all 9 object categories. Our proposed CAM outperforms the baselines by a large margin, which shows the superiority of PSS compared to PCP. Furthermore,

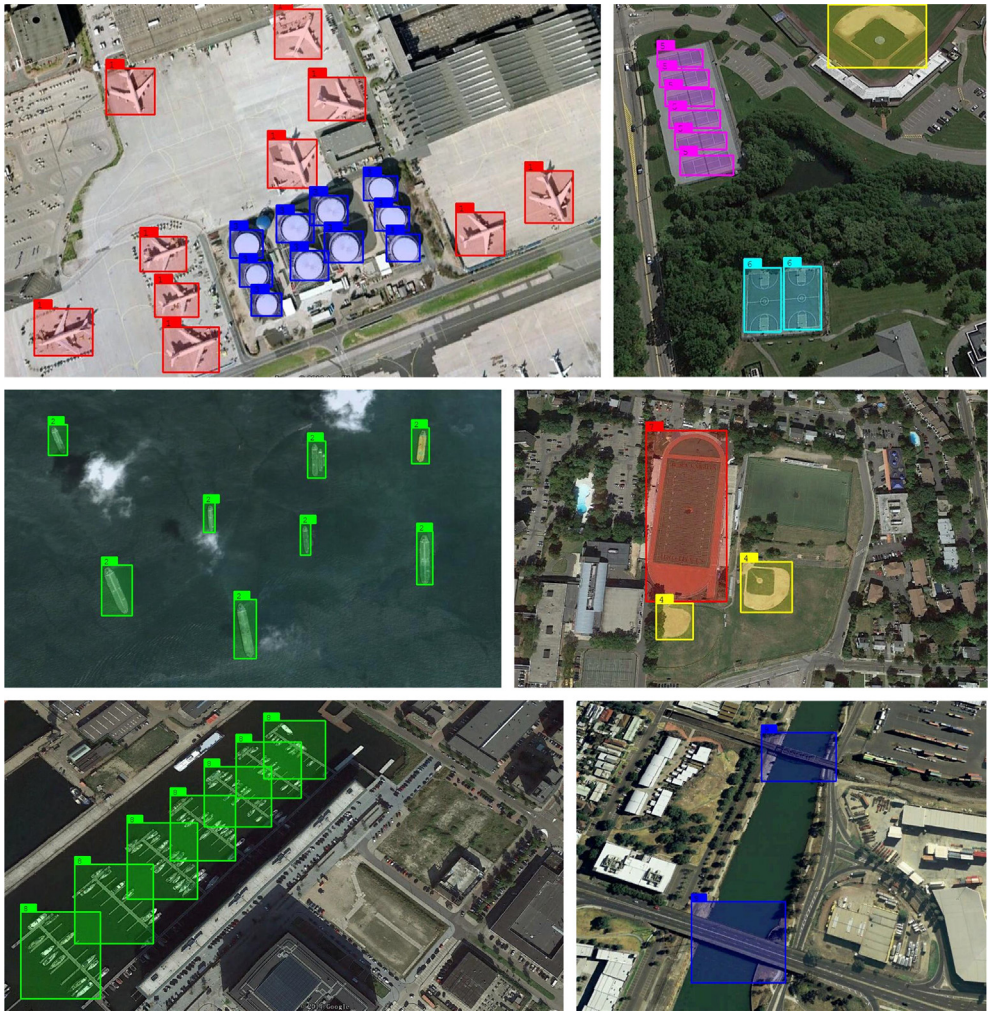


Fig. 6. The testing geospatial object detection dataset (i.e., NWPU VHR-9). Six remote sensing images are randomly selected from this dataset and illustrated with the bounding box annotations. Specifically, '1–9' on the bounding boxes stand for the airplane, ship, oil tank, baseball diamond, tennis court, basketball court, ground track field, harbor, and bridge, respectively.

GAP performs better than GMP, and SSV also improves the object prediction performance when compared with ILV.

In addition, we show our proposed CAM with PSS + GAP + SSV in a synthesized manner in Fig. 8. As depicted in Fig. 8, our proposed CAM accurately indicates the presence of objects.

5.3. Performance evaluation of multi-class geospatial object detection

This section evaluates the multi-class geospatial object detection performance. In the following, Section 5.3.1 gives the evaluation measures for multi-class geospatial object detection. Section 5.3.2 reports the quantitative comparison result with baselines. Finally, Section 5.3.3 analyzes the sensitivity of critical parameters.

Table 1
The method abbreviations based on different configurations. 'CAM' is the abbreviation of class-specific activation maps; 'GMP' is the abbreviation of global maximum pooling; 'GAP' is the abbreviation of global average pooling; 'PSS' denotes the pair-wise scene-level similarity constraint; 'PCP' stands for the point-wise scene category prediction constraint; 'ILV' is the abbreviation of image-level-voting; 'SSV' is the abbreviation of scene-sliding-voting.

The method abbreviation	The training method	The global pooling operation	The voting strategy
Baseline1 (Oquab et al., 2015): the CAM with PCP1 + GMP + ILV	Both of **C** and **W** trained by PCP	GMP	ILV
Baseline2 (Zhou et al., 2016): the CAM with PCP1 + GAP + ILV	Both of **C** and **W** trained by PCP	GAP	ILV
Baseline3 (Oquab et al., 2015): the CAM with PCP2 + GMP + ILV	**C** transferred from the already-trained networks on ImageNet; **W** trained by PCP	GMP	ILV
Baseline4 (Zhou et al., 2016): the CAM with PCP2 + GAP + ILV	**C** transferred from the already-trained networks on ImageNet; **W** trained by PCP	GAP	ILV
Our proposed CAM with PSS + GMP + ILV	**C** trained by PSS; **W** trained by PCP	GMP	ILV
Our proposed CAM with PSS + GAP + ILV	**C** trained by PSS; **W** trained by PCP	GAP	ILV
Our proposed CAM with PSS + GMP + SSV	**C** trained by PSS; **W** trained by PCP	GMP	SSV
Our proposed CAM with PSS + GAP + SSV	**C** trained by PSS; **W** trained by PCP	GAP	SSV

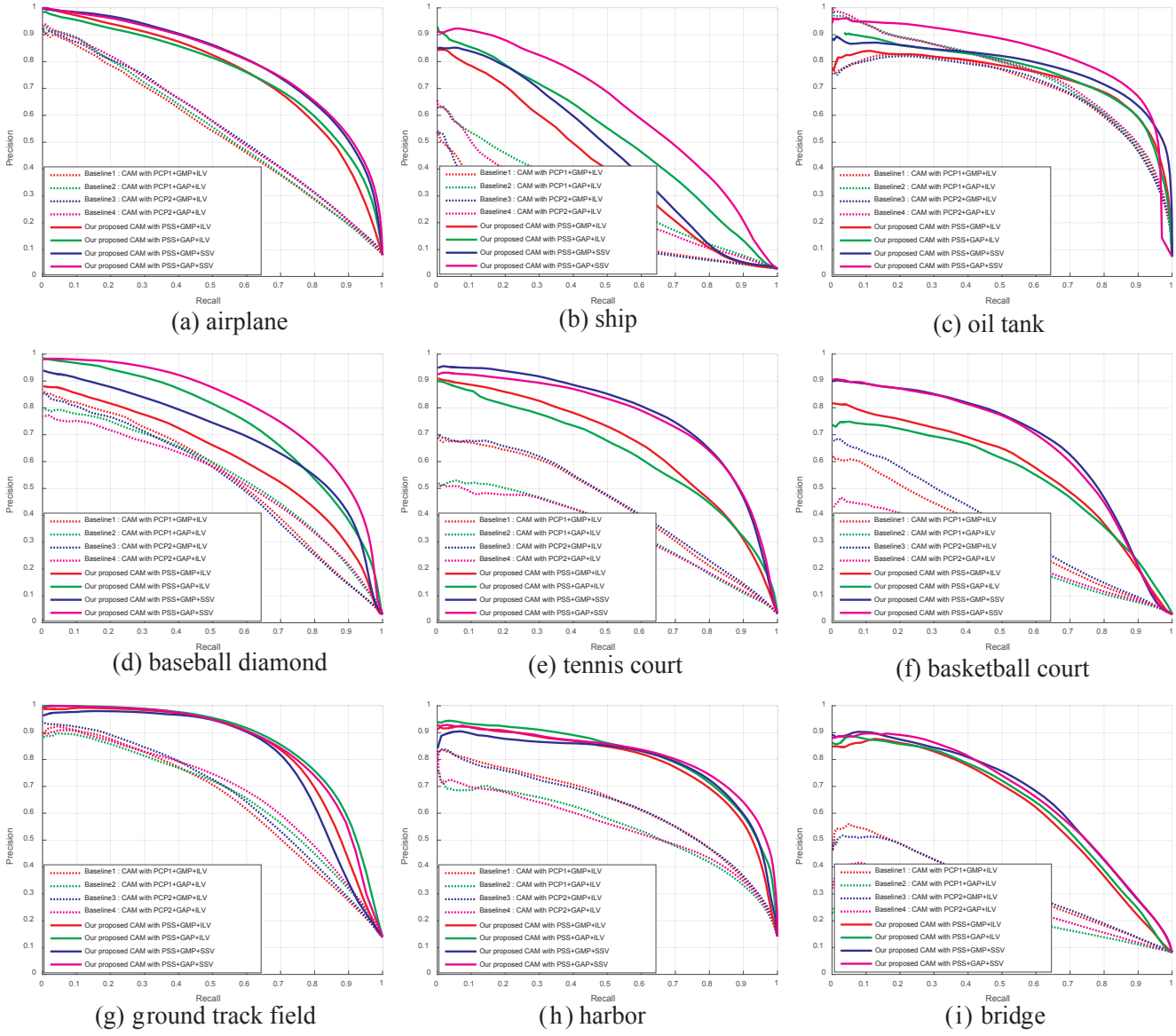


Fig. 7. The Precision-Recall curves of the proposed methods and some baselines on 9 object categories.

5.3.1. Evaluation measures

Unlike the pixel-level Precision and Recall measures in Section 5.2, this section uses the bounding-box-level Precision and Recall metrics for evaluating object detection performance. As also adopted in Cheng et al. (2014), the Precision metric measures the fraction of detections that are true positives, and the Recall metric measures the fraction of positives that are correctly identified. As suggested in Cheng et al. (2014), a detection is considered to be a true positive if the area overlap ratio between the predicted bounding box and the ground truth bounding box exceeds 0.5; otherwise, a detections is considered to be a false positive. Furthermore, we also consider comprehensive metrics including average precision (AP) and F-measure, where AP computes the average value of Precision over the interval from Recall = 0 to Recall = 1, and the F-measure is calculated by:

$$\text{F - measure} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{8}$$

5.3.2. Comparison to some baselines

In this section, we further evaluate the object detection performance of the CAMs including our proposed CAM with PSS + GAP + SSV, baseline1 (Oquab et al., 2015): the CAM with PCP1 + GMP + ILV, baseline2 (Zhou et al., 2016): the CAM with PCP1 + GAP + ILV, baseline3 (Oquab et al., 2015): the CAM with PCP2 + GMP + ILV, and baseline4 (Zhou et al., 2016): the CAM with PCP2 + GAP + ILV. As depicted in Section 4.2, the considered segmentation methods include SS, DS, and MS.

Each combination of a CAM and a segmentation method constitutes a potential object detection method. We report object detection performance of different combinations in Table 2. More specifically, by varying the segmentation factor constant thFactor in Section 4.2, we calculate the overall evaluation metric (i.e., AP) to indicate the performance of a particular object detection method. As shown in Table 2, our proposed CAM can achieve the best object detection performance on all geospatial object types by tuning the segmentation methods.

In accordance with Algorithm 1, we report the running time of our proposed object detection method in two phases (i.e., the training and testing phases). More specifically, Tables 3 and 4 report the running time of the training and testing phases, respectively. As shown in Table 3, the most majority of the computational load in the training phase focuses on learning discriminative convolutional weights, and the whole weights of deep networks can be trained in 26 h due to the usage of GPU. In addition, Table 4 summarizes the running time of the testing

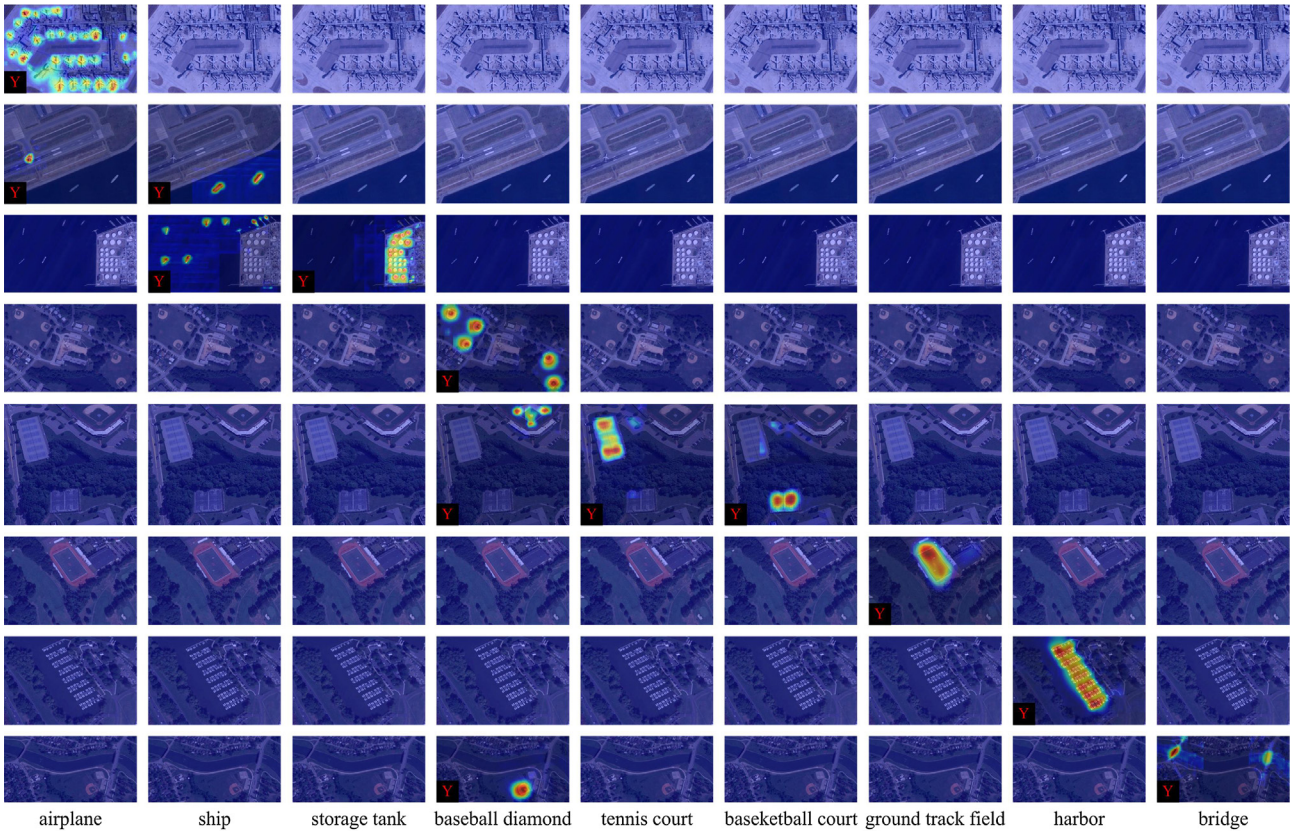


Fig. 8. The synthesized activation maps of testing remote sensing images. In the synthesized activation maps, the original image is overlaid with the activation maps for different object categories. Each column shows the activation results of different input images on one particular object class, and each row shows the activation results of one input image over different object classes. Based on the ground truth, we mark the activation result with 'Y' if the input image contains objects of the corresponding class shown in the bottom, which benefits intuitively showing the quality of the activation maps.

Table 2
Performance comparisons of our proposed CAM and four other baselines based on three segmentation methods in terms of the evaluation metric of AP.

CAM methods	Segmentation methods	Airplane	Ship	Oil tank	Baseball diamond	Tennis court	Basketball court	Ground track filed	Harbor	Bridge
Baseline1	SS	6.10%	0.01%	2.88%	9.61%	0.05%	0.03%	3.97%	0.01%	0.06%
(Oquab et al., 2015)	DS	0.03%	2.80%	26.4%	6.54%	0.03%	0.01%	13.2%	0.03%	0.01%
	MS	14.4%	2.90%	0.03%	3.24%	0.01%	0.02%	0.04%	10.5%	0.04%
Baseline2	SS	4.94%	0.04%	3.33%	10.2%	0.01%	0.02%	2.90%	0.02%	0.03%
(Zhou et al., 2016)	DS	0.01%	3.43%	30.9%	7.96%	2.30%	0.04%	12.4%	0.01%	0.01%
	MS	11.9%	2.85%	0.01%	4.06%	0.01%	0.01%	0.01%	10.1%	0.02%
Baseline3	SS	7.26%	0.02%	0.00%	14.1%	0.03%	0.00%	2.90%	0.01%	0.07%
(Oquab et al., 2015)	DS	0.01%	5.68%	41.4%	7.94%	2.53%	1.65%	9.04%	0.00%	0.02%
	MS	17.3%	5.15%	0.02%	5.07%	0.03%	0.01%	0.02%	16.6%	0.03%
Baseline4	SS	6.73%	0.03%	0.02%	15.6%	0.04%	0.29%	2.74%	0.01%	0.05%
(Zhou et al., 2016)	DS	0.02%	4.42%	37.1%	9.67%	0.02%	1.04%	7.83%	0.02%	0.01%
	MS	16.8%	4.12%	0.00%	7.42%	0.01%	0.35%	0.02%	16.6%	0.03%
Our proposed CAM	SS	18.6%	3.11%	6.93%	27.8%	3.71%	6.51%	1.63%	4.54%	5.15%
	DS	0.03%	4.80%	51.2%	12.1%	8.24%	9.44%	13.5%	0.03%	0.02%
	MS	26.9%	5.80%	0.01%	19.8%	0.02%	5.12%	0.01%	38.4%	3.03%

Table 3
Computation time of different modules in the training phase.

The training stage	Running time on the whole scene dataset
Learn discriminative convolutional weights	25.8 (hours)
Learn class-specific activation weights	0.35 (hours)

Table 4
Computation time of different modules in the testing phase.

The testing stage		Average running time per image
Calculate the class-specific activation maps		13.8 (seconds)
Extract multi-class objects from the class-specific activation maps	SS	0.53 (seconds)
	DS	11.5 (seconds)
	MS	0.62 (seconds)

phase in two sub-steps. More specifically, we report the average running time of different segmentation methods in the second sub-step. It is noted that one can easily accelerate the testing phase by the parallelization modification as needed.

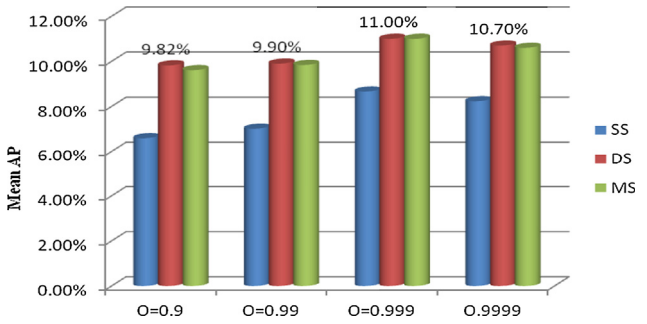


Fig. 9. The overall object detection performance of our proposed CAM combined with three segmentation methods, measured in terms of the evaluation metric of Mean AP over all nine object classes, under different probability constants O.

5.3.3. *Sensitivity analysis of critical parameters*

In the following, we specifically analyze the sensitivity of the critical parameters including the probability constant O, the segmentation method, and the factor constant thFactor in Section 4.2.

In Fig. 9, we report the overall object detection performance of our proposed CAM combined with three segmentation methods (i.e., SS, DS, and MS) under different probability constants where the performance is measured by Mean AP over all nine object classes. As shown in Fig. 9, $O \geqslant 0.999$ can make our proposed method perform better than a small probability constant, but the performance of our proposed method starts to decrease when $O = 0.9999$. Hence, the probability constant O is empirically set to 0.999 in our implementation.

With the probability constant O fixed to 0.999, we further analyze the sensitivity of the segmentation methods and the factor constant thFactor. Using the F-measure metric, Fig. 10 reports the object detection performance of our proposed CAM based on different segmentation methods and factor constants.

As shown in Fig. 10, SS can make our proposed CAM achieve the best detection performance on the baseball diamond and the bridge categories because objects in these categories are generally scattered which makes straightforward segmentation possible. DS achieves the best detection performance on the ship, the oil tank, the tennis court, the basketball court, and the ground track field categories. As verified by this quantitative result, DS not only enhances densely distributed small objects (e.g., the oil tank), but also helps to enhance large objects (e.g., the ground track field) to detect the whole object and achieve better performance than SS, because SS often detects parts of the object in this case. In addition, MS clearly improves the detection performance for the airplane and the harbor categories, when compared to SS and DS.

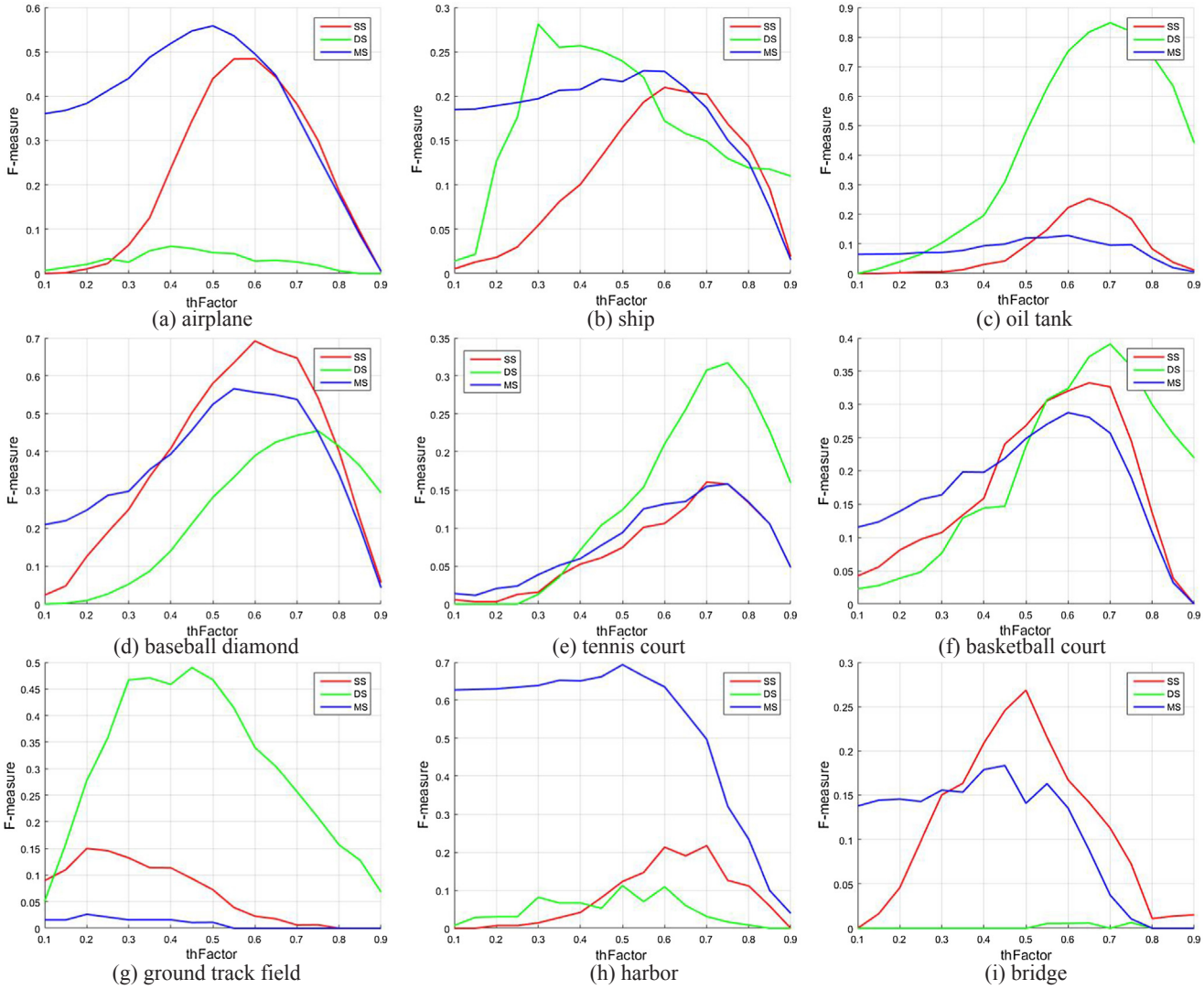


Fig. 10. Performance evaluation of three segmentation methods under different factor constants thFactor where the performance is measured by F-measure.

Fig. 11. Some visual detection results of our proposed CAM based on the class-configured segmentation method. The true positives, false positives, and false negatives are indicated by red, blue, and yellow rectangles, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

As only extremely weak supervision is adopted in this paper, it is worth mentioning that the final segmentation module is sensitive to the segmentation methods and the segmentation factors. In practical applications, we can use a small validation dataset with bounding box annotations to select the optimal segmentation method and its corresponding factor per object category to obtain robust object detection performance. Based on the analysis in Figs. 10, 11 visually shows the geospatial object detection results of our proposed CAM under the best configuration of the segmentation method and factor per object category (i.e., the class-configured segmentation strategy).

5.4. Limitations and future perspectives

Although the proposed geospatial object detection approach does not depend on bounding box annotations, which saves annotation cost, the proposed approach still requires a semantic category correspondence between objects and scenes. To tackle this limitation, we can exploit zero-shot learning techniques (Han et al., 2018; Li et al., 2017; Zhang and Saligrama, 2016) to combine the detectors of existing object types to address unseen object types in scene datasets.

Due to the dense distribution of objects and the complex structures of backgrounds, the final geospatial object detection performance is

sensitive to the segmentation methods. While this problem can be addressed by selecting a suitable segmentation strategy per object category using a validation dataset, a uniform segmentation solution is still preferred. To pursue uniform segmentation, we can utilize our proposed CAM to output high-confidence class-specific object proposals with the aid of class-agonistic object proposal techniques (Uijlings et al., 2013) and further refine the class-specific object proposals by using advanced learning methods that are robust to label noise (Patrini et al., 2017).

In the future, we may extend the proposed geospatial object detection approach to more challenging object detection tasks such as infrared object detection (Li and Zhang, 2018) and SAR object detection (Tan et al., 2015).

6. Conclusion

This paper proposes a new learning framework that can transfer knowledge from the remote sensing image scene classification task to the multi-class geospatial object detection task. To make full use of the supervision from scene tags, we exploit pair-wise scene-level similarity and point-wise category prediction constraints to learn discriminative convolutional weights and class-specific activation weights. Based on these learned weights, we propose a multi-scale scene-sliding-voting strategy to compute the CAM. In addition, we present a set of CAM-oriented segmentation methods for detecting objects from the CAM. We train deep networks on a publicly open remote sensing image scene dataset, and we conduct multi-class geospatial object detection on another remote sensing geospatial object detection dataset. Even under this extremely weak supervision, the proposed approach achieves promising geospatial object detection results and outperforms the baselines. In our future work, we will exploit zero-shot learning to address the detection of unseen object types in the scene dataset and to unify the segmentation process with the aid of class-agonistic object proposal techniques and noise-tolerated learning.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under grant 2018YFB0505003; the National Natural Science Foundation of China under grants 41601352 and 41322010; the China Postdoctoral Science Foundation under grants 2016M590716 and 2017T100581; the Hubei Provincial Natural Science Foundation of China under grant 2018CFB501.

References

Bilen, H., Vedaldi, A., 2016. Weakly supervised deep detection networks. In: Proceedings of the 2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2016). IEEE, Las Vegas, pp. 2846–2854.

Bromley, J., Bentz, J., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Sackinger, E., Shah, R., 1993. Signature verification using a siamese time delay neural network. Int. J. Pattern Recogn. Artif. Intell. 7, 669–688.

Cheng, G., Han, J., Guo, L., Qian, X., Zhou, P., Yao, X., Hu, X., 2013. Object detection in remote sensing imagery using a discriminatively trained mixture model. ISPRS J. Photogramm. Remote Sens. 85, 32–43.

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.

Cheng, G., Han, J., 2016. A survey on object detection in optical remote sensing images. ISPRS J. Photogramm. Remote Sens. 117, 11–28.

Cheng, G., Zhou, P., Han, J., 2016. Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images. IEEE Trans. Geosci. Remote Sens. 54, 7405–7415.

Cheng, G., Han, J., Lu, X., 2017. Remote sensing image scene classification: benchmark and state of the art. Proc. IEEE 105, 1865–1883.

Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A., 2014. The devil is in the details: An evaluation of recent feature encoding methods. Proceedings of the British Machine Vision Conference (BMVC, 2014), vol. 2(4).

Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A., 2018. DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE Trans. Pattern Anal. Mach. Intell. 40, 834–848.

Cinbis, R., Verbeek, J., Schmid, C., 2017. Weakly supervised object localization with multi-fold multiple instance learning. IEEE Trans. Pattern Anal. Mach. Intell. 39, 189–203.

Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., Zou, H., 2018. Multi-scale object detection in remote sensing imagery with convolutional neural networks. ISPRS J. Photogramm. Remote Sens (in press).

Ding, P., Zhang, Y., Deng, W., Jia, P., Kuijper, A., 2018. A light and faster regional convolutional neural network for object detection in optical remote sensing images. ISPRS J. Photogramm. Remote Sens. 141, 208–218.

Dou, H., Ming, D., Yang, Z., Pan, Z., Li, Y., Tian, J., 2017. Object-based visual saliency via laplacian regularized kernel regression. IEEE Trans. Multimedia 19, 1718–1729.

Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A., 2010. The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. 88, 303–338.

Guo, C., Ma, Q., Zhang, L., 2008. Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform. In: Proceedings of the 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2008). IEEE, Anchorage, AK, pp. 1–8.

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, 3325–3337.

Heitz, G., Koller, D., 2008. Learning spatial context: Using stuff to find things. In: Proceedings of the 10th European Conference on Computer Vision (ECCV 2008). Springer, Marseille, France, pp. 30–43.

Han, J., Zhang, D., Cheng, G., Liu, N., Xu, D., 2018. Advanced deep-learning techniques for salient and category-specific object detection: a survey. IEEE Signal Process Mag. 35, 84–100.

Kolesnikow, A., Lampert, C., 2016. Seed, expand and constrain: three principles for weakly-supervised image segmentation. In: Proceedings of the 14th European Conference on Computer Vision (ECCV 2016). Springer, Amsterdam, The Netherlands, pp. 695–711.

Koch, G., Zemel, R., Salakhutdinov, R., 2015. Siamese neural networks for one-shot image recognition. In: Proceedings of the 32nd International Conference on Machine Learning (ICML 2015), Lille, France, vol. 37.

Krizhevsky, A., Sutskever, I., Hinton, G., 2012. Imagenet classification with deep convolutional neural networks. In: Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS 2012), Lake Tahoe, pp. 1097–1105.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

Li, Y., Zhang, Y., Huang, X., Zhu, H., Ma, J., 2018. Large-scale remote sensing image retrieval by deep hashing neural networks. IEEE Trans. Geosci. Remote Sens. 56, 950–965.

Li, Y., Zhang, Y., Huang, X., Ma, J., 2018b. Learning source-invariant deep hashing convolutional neural networks for cross-source remote sensing image retrieval. IEEE Trans. Geosci. Remote Sens (in press).

Li, Y., Zhang, Y., 2018. Robust infrared small target detection using local steering kernel reconstruction. Pattern Recogn. 77, 113–125.

Li, A., Lu, Z., Wang, L., Xiang, T., Wen, J., 2017. Zero-shot scene classification for high spatial resolution remote sensing images. IEEE Trans. Geosci. Remote Sens. 55, 4157–4167.

Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollar, P., Zitnick, C., 2014. Microsoft COCO: common objects in context. In: Proceedings of the 13th European Conference on Computer Vision (ECCV 2014). Springer, Zurich, Switzerland, pp. 740–755.

Liu, H., Wang, R., Shan, S., Chen, X., 2016. Deep supervised hashing for fast image retrieval. In: Proceedings of the 2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2016). IEEE, Las Vegas, NV, pp. 2064–2072.

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, 2486–2498.

Marcos, D., Volpi, M., Kellenberger, B., Tuia, D., 2018. Land cover mapping at very high resolution with rotation equivariant CNNs: Towards small yet accurate models. ISPRS J. Photogramm. Remote Sens (in press).

Oquab, M., Bottou, L., Laptev, I., Sivic, J., 2014. Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the 2014 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2014). IEEE, Columbus, OH, pp. 1717–1724.

Oquab, M., Bottou, L., Laptev, I., Sivic, J., 2015. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: Proceedings of the 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2015). IEEE, Boston, MA, pp. 685–694.

Patrini, G., Rozza, A., Mennon, A., Nock, R., Qu, L., 2017. Making deep neural networks robust to label noise: a loss correction approach. In: Proceedings of the 2017 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2017). IEEE, Honolulu, HI, pp. 1944–1952.

Pinheiro, P., Collobert, R., 2015. From image-level to pixel-level labeling with convolutional networks. In: Proceedings of the 2015 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2015). IEEE, Boston, MA, pp. 1713–1721..

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L., 2015. Imagenet large scale visual recognition challenge. Int. J. Comput. Vision 115, 211–252.

Tan, Y., Li, Q., Li, Y., Tian, J., 2015. Aircraft detection in high-resolution SAR images based on a gradient textural saliency map. Sensors 15, 23071–23094.

Tang, P., Wang, X., Huang, Z., Bai, X., Liu, W., 2017. Deep patch learning for weakly supervised object classification and discovery. Pattern Recogn. 71, 446–459.

Uijlings, J., van de Sande, K., Gevers, T., Smeulders, A., 2013. Selective search for object recognition. Int. J. Comput. Vis. 104, 154–171.

Wang, L., Lu, H., Wang, Y., Feng, M., 2017. Learning to detect salient objects with image-level supervision. In: Proceedings of the 2017 IEEE Computer Society Conference on

Computer Vision and Pattern Recognition (CVPR 2017). IEEE, Honolulu, HI, pp. 136–145.
Xia, G., Hu, J., Hu, F., Shi, B., Bai, X., Zhong, Y., Zhang, L., Lu, X., 2017. AID: a benchmark data set for performance evaluation of aerial scene classification. IEEE Trans. Geosci. Remote Sens. 55, 3965–3981.
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, 618–644.
Yang, Y., Newsam, S., 2010. Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, San Jose, California, pp. 270–279.
Zhang, Z., Saligrama, V., 2016. Zero-shot learning via joint latent similarity embedding. In: Proceedings of the 2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2016). IEEE, Las Vegas, NV, pp. 6034–6042.
Zhang, D., Han, J., Han, J., Shao, L., 2016. Cosaliency detection based on intrasaliency prior transfer and deep intersaliency mining. IEEE Trans. Neural Networks Learn. Syst. 27, 1163–1176.
Zhong, Y., Han, X., Zhang, L., 2018. Multi-class geospatial object detection based on a position-sensitive balancing framework for high spatial resolution remote sensing imagery. ISPRS J. Photogramm. Remote Sens. 138, 281–294.
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2014. Object detectors emerge in deep scene cnns. arXiv arXiv: 1412.6856.
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2016. Learning deep features for discriminative localization. In: Proceedings of the 2016 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2016). IEEE, Las Vegas, NV, pp. 2921–2929.
Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A., 2018a. Places: a 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. 40, 1452–1464.
Zhou, W., Newsam, S., Li, C., Shao, Z., 2018b. PatternNet: A benchmark dataset for performance evaluation of remote sensing image retrieval. ISPRS J. Photogramm. Remote Sens (in press).
Zou, Z., Shi, Z., 2018. Random access memories: a new paradigm for target detection in high resolution aerial remote sensing images. IEEE Trans. Image Process. 27, 1100–1111.