Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks

Yang Long, Yiping Gong, Zhifeng Xiao, and Qing Liu

***Abstract*— In this paper, we focus on tackling the problem of automatic accurate localization of detected objects in high-resolution remote sensing images. The two major problems for object localization in remote sensing images caused by the complex context information such images contain are achieving generalizability of the features used to describe objects and achieving accurate object locations. To address these challenges, we propose a new object localization framework, which can be divided into three processes: region proposal, classification, and accurate object localization process. First, a region proposal method is used to generate candidate regions with the aim of detecting all objects of interest within these images. Then, generic image features from a local image corresponding to each region proposal are extracted by a combination model of 2-D reduction convolutional neural networks (CNNs). Finally, to improve the location accuracy, we propose an unsupervised score-based bounding box regression (USB-BBR) algorithm, combined with a nonmaximum suppression algorithm to optimize the bounding boxes of regions that detected as objects. Experiments show that the dimension-reduction model performs better than the retrained and fine-tuned models and the detection precision of the combined CNN model is much higher than that of any single model. Also our proposed USB-BBR algorithm can more accurately locate objects within an image. Compared with traditional features extraction methods, such as elliptic Fourier transform-based histogram of oriented gradients and local binary pattern histogram Fourier, our proposed localization framework shows robustness when dealing with different complex backgrounds.**

***Index Terms*— Convolutional neural network (CNN), object localization, remote sensing images, unsupervised score-based bounding box regression (USB-BBR).**

I. Introduction

As SENSOR technology and aerospace remote sensing technology have improved, the quality and quantity of remote sensing images have also undergone great improvement. Researchers can now conveniently acquire remote sensing images with high spatial or spectral resolutions. These remote sensing images improve researchers' chances of understanding the image content, especially when analyzing their semantic meaning (the high-level features in remote sensing images). However, while semantic analysis is a difficult and important part of remote sensing analysis, object detection is the basic task required for semantic analysis. Consequently, the problem of object detection has attracted considerable research attention and has been extensively studied. The difference between object detection and object localization is subtle. Object detection focuses on detecting the presence of entire objects. But object localization has higher requirements than object detection does. Object localization requires that objects be located accurately. In this paper, we aim to propose an accurate object localization framework for remote sensing images.

Currently, there is small demand for accurate localization in the remote sensing field. The majority of studies focuses on detection rather than localization (the two processes have been confused by some people). Object detection in remote sensing images faces far more challenges because of more complex background information they contain than that of natural images. Remote sensing images offer information about the texture, shape, and structure of ground objects, and they can be used for precise object identification. However, in addition to providing ample information for object detection, they also present information redundancy problems. Moreover, because of noise interference, weather, illumination intensity, and other factors, object detection in remote sensing images is a troublesome issue.

In this paper, we focus on accurate localization of detected objects rather than simple object detection. Based on this aspect, we use object localization to summarize this paper. In this paper, we tackle the feature extraction problem for object detection in remote sensing images using convolutional neural network (CNN) models. CNN relies on the specific layer structure to learn the essential features of input images, thus avoiding the effort of designing a feature extraction strategy. In addition, CNN models have a wide range of application. CNNs with deeper layer structure tend to have better learning abilities. In this paper, the feature extraction strategy is based on CNN models with a deep layer that can describe objects in remote sensing images. Finally, we propose a new object localization framework for remote sensing images that can detect and locate objects accurately.

The rest of this paper is organized as follows. Section II reviews related works on object detection and applications of CNN in remote sensing images. Section III presents the details of the proposed object localization framework. Section IV discusses the object detection experiments for the proposed object localization framework, and Section V analyzes the

Manuscript received May 28, 2016; revised August 16, 2016 and November 21, 2016; accepted December 13, 2016. Date of publication January 19, 2017; date of current version March 17, 2017. This work was supported by the National Key Research and Development Program of China under Grant 2016YFB0502602. *(Corresponding author: Zhifeng Xiao.)*

Y. Long, Y. Gong, and Z. Xiao are with the State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China (e-mail: longyang@whu.edu.cn; gongyp15@163.com; xzf@whu.edu.cn).

Q. Liu is with Shenzhen Prafly Technology Co., Ltd., Shenzhen 518048, China (e-mail: liuqing19900618@126.com).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2016.2645610

detection performance of the proposed object localization framework. Finally, Section VI concludes this paper.

II. Related Work

Object detection in remote sensing images has been widely researched in recent years. Many researchers have used local features to extract characteristics, such as scale invariant feature transform (SIFT) [1], histogram of oriented gradients (HOG) [2], and Saliency [3], [4]. These local features have a certain invariance, but this invariant ability needs to be enhanced when dealing with the problems of various object orientations in remote sensing images. Cheng *et al.* [5] introduced a rotation-invariant layer on the basis of the existing CNN architectures to achieve rotation invariant and the proposed rotation-invariant CNN model achieves significantly performance. Xiao *et al.* [6] used the elliptic Fourier transform (EFT) to improve the invariance of HOG features. While local features are the low-level features of images, object detection is a part of the high-level semantic analysis, which is more closely aligned with actually understanding image content. Image understanding is the ultimate goal pursued by all image processing researchers constantly pursue. Cheng and Han [7] provided a review of the recent progress in object detection in remote sensing images and proposed two promising research directions, namely, deep learning-based feature representation and weakly supervised learning-based geospatial object detection.

As image processing theory has developed, many studies have focused on the mid-level features of remote sensing images. The most popular mid-level feature is the part-based model [8]–[12]. The main idea behind part-based models is that objects consist of several visually important parts; therefore, the object detection task can be decomposed into processes that detect these parts. To acquire the semantic information of images, some studies have applied semantic models to extract semantic information from remote sensing images [13]–[15].

Recently, deep learning models have received increased attention. The most popular deep learning methods are the CNN models. CNN does not need handcrafted features, and it requires fewer parameters than other networks, because it shares weights for the same filter. A CNN model can learn the essential features of input images based on its specific network structure. CNN has been widely used in object classification, object detection, speech recognition, and so on. Zhong [16] proposes a large patch CNN for the scene classification of high spatial resolution imagery, which contains a large patch sampling layer used to generate hundreds of possible scene patches for the feature learning.

Many theoretical studies concerning CNN have been made [17]–[29]. As computer technology has advanced, deeper and more efficient CNN models have been proposed. AlexNet, developed by Krizhevsky *et al.* [30], was a ground breaking CNN architecture and a winning model in the 2012 ImageNet Large-Scale Visual Recognition Challenge (ILSVRC-2012). GoogleNet [31] was a winner of ILSVRC-2014; the main hallmark of this architecture was its improved utilization of the computing resources inside the network. This improvement was achieved through a carefully crafted design that allowed the depth and width of the network to increase while keeping the computational demands constant. The VGG models proposed by Simonyan and Zisserman [32] were used to investigate the relationship between the depth of a convolutional network and its accuracy in a large-scale image recognition setting. SPP-net [33] could generate a fixed-length representation regardless of the size or scale of the image, thus eliminating the requirement for a fixed-size input image. Resnet [34] was a winner of ILSVRC-2015 and COCO-2015; the layers were reformulated as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions to ease the training of networks that are substantially deeper than those used previously.

In the field of remote sensing images, many object detection experiments use only trained CNN models on large data sets for pretraining [35]–[39]. Wu *et al.* [40] altered the typical CNN proposed by Lecun *et al.* [41] to create a model used for aircraft detection. He obtained the candidate object regions using the BING technique. Zhang *et al.* [38] used trained CNN models to extract surrounding features that were combined with local features (HOG) to describe oil tanks and then applied gradient orientation to select candidate regions from satellite images. Qiling Jiang *et al.* [42] used a graph-based superpixel segmentation to extract a set of image patches and then trained a CNN to classify these patches into vehicles and nonvehicles. Zhu *et al.* [43] used CNN features from combined layers to perform orientation-robust aerial object detection. Ding *et al.* [44] investigated the capabilities of a CNN model combined with data augmentation operations in SAR target recognition. Sevo and Avramovic [45] proposed a novel two-stage approach for CNN training and implemented a network-based method for automatic content-based object detection on high-resolution aerial images. Salberg [37] extracted features from a pretrained deep CNN and used it for automatic detection of seals in aerial remote sensing images. Zhang *et al.* [46] constructed an iterative weakly supervised learning framework to automatically mine and augment the training data set from the original image and combined the candidate region proposal network and a localization network to extract the proposals and locate aircraft in large-scale very high resolution (VHR) images.

In this paper, we use a suitable feature extractor based on the CNN model to extract the essential features of objects from remote sensing images. The method used to obtain the candidate object regions is crucial for object detection. The common sliding window method performs an exhaustive search, but it is time-consuming. Moreover, the window can have only one size at a time; thus, the location precision of the slide window technique is not high. Therefore, we propose a new object localization framework to address the location problem.

III. Proposed Framework

The proposed object localization framework follows a pipeline approach. First, when dealing with a test image, we use a selective search algorithm to generate category-independent possible regions. Then, all these candidate regions

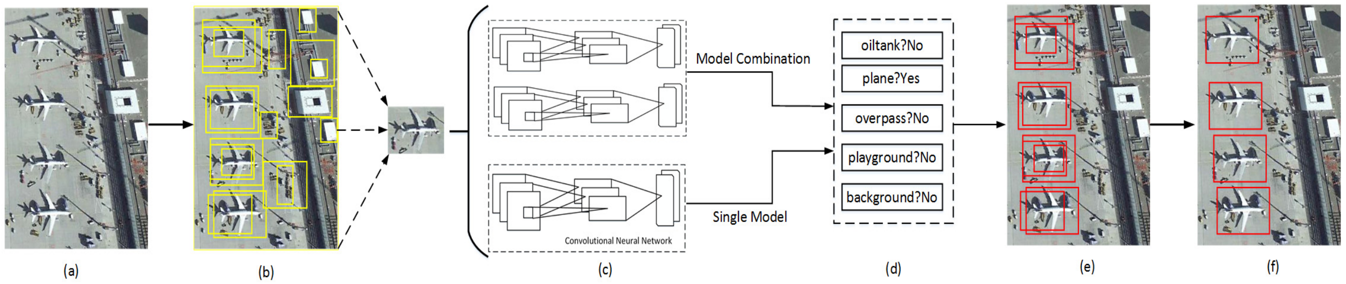

Fig. 1. Proposed object localization framework. (a) Test image. (b) Selective search method produces most of the candidate object regions from the test image and adds some candidate regions extracted from low region density areas, where the selective search method generates few regions. (c) CNN is applied to extract the features from these candidate regions. (d) Classification results of the regions. We used two approaches to obtain the classification results of candidate regions: one is a single model strategy and the other is a model combination strategy that averages the outputs of two CNN models. (e) Classification results of the candidate regions. (f) Final detection results after the accurate object localization process.

are sent to a combined model consists of 2-D reduction CNNs. Class labels and classification scores for each candidate region are an average output of two CNN models. Finally, we perform an accurate object localization process to address these classified regions. For accurate object localization, we propose using the unsupervised score-based bounding box regression (USB-BBR) method to improve box localization precision after using the nonmaximum suppression (NMS). In this section, we present our design for each procedure. The proposed object localization framework is shown in Fig. 1.

A. Region Proposal

Traditionally, a sliding window technique has been used for object detection; however, the sliding window technique is an exhaustive search method and is computationally expensive. Recently, Uijlings *et al.* [47] proposed a selective search algorithm that produces object regions by taking the underlying image structure into account. The selective search algorithm yields a completely class-independent set of locations. It also generates fewer locations, which simplifies the problem because the sample variability is lower. More importantly, it frees up computational power, which can then be utilized for more robust machine learning techniques and more powerful appearance models.

Objects can occur at any scale within an image because of the diverse means for acquiring images. Moreover, images at the same scale may be different sizes. Therefore, we collect images that share approximately the same image with the aim of acquiring all the similar objects at one scale. Then, we can apply the selective search algorithm to address the objects that have different sizes within an image. Furthermore, the boundaries of some objects are less clear than the boundaries of others. The selective search applies a diverse set of strategies to deal with different size conditions, lighting conditions, and other imaging cases. These strategies make the selective search method stable, robust, and independent of the object class.

The quality of region proposals directly influences the CNN detection result and the accuracy of object localization. Consequently, we analyzed the influence of the region proposals generated by selective search on our data set. Hosang *et al.* [48] provided a deep analysis concerning ten different object-proposal methods and found that the recall of the selective search method is approximately 0.83 for an intersection-of-union (IoU) of 0.5 on the ImageNet 2013 validation set when 1000 proposals exist per image. The experiments also indicate that the greater the number of candidates, the higher the recall will be. While the recall value may be different for remote sending images, in any image classification task, it is a key factor that influences both the CNN detection result and the accuracy of object localization.

B. Feature Extraction

A CNN model consists of convolution layers, pooling layers, and full-connection layers. A convolution layer has several filters and generates different feature maps using these filters on local receptive fields in the maps of the previous layer or input. The filter size can be $n \times n$ (where n is smaller than the input size). Weights are shared between convolution layers. The pooling layer uses filters to generalize the brief representation of the convolution layer to reduce the number of parameters. There are several pooling types; these include max pooling and average pooling. The pooling operation provides a form of translation invariance. The feature becomes more complex and global as the layers become deeper.

In this paper, the CNN models chosen to extract features are AlexNet and GoogleNet due to their superior performance. To retain more information for backpropagation, for both the AlexNet and GoogleNet networks, we add a 64-D inner product layer before the last inner-product layer. For AlexNet, we reduce the dimension of the second full-connection layer from 4096 to 64, and for GoogleNet, we add a 64-D layer after the last convolutional layer. Moreover, the two CNNs are combined to detect objects simultaneously and the result of this combined CNN models is the averaged outputs of the two CNN models.

To perform feature extraction, we first extract image patches from the candidate regions generated by the region generation process. Then, we normalize the image patches to 227×227 to fit the input dimensions of the CNN models and use them as input to the CNN to extract features. The last layers of the CNN models are softmax classification layers. For the combined CNN model, we need to perform forward propagations of the two CNN models. After forward propagation, the two CNN models produce classification results of these candidate regions. The regions will have class labels and class scores after forward propagation. For examples, assume that the candidate region is b and that the class label and classified scores for the entire set of classes computed by model A are

l_A and s_A, while the class label and classified scores for the entire set of classes computed by model B are l_B and s_B. The class score S of the model combination is the average of s_A and s_B, while the class label L of the model combination is the label according to the maximum of S.

C. Accurate Object Localization

To produce the optimum bounding box for locating the object, we propose a two-stage object accurate localization method: NMS and USB-BBR. The NMS method is widely used to tackle the problem of the redundancy of the bounding boxes, while the accurate localization issue has not been solved yet, because it lacks the ability to perform an integrated optimization of the remaining higher quality boxes. Our experiments show that NMS retains extra boxes that are not accurate for locating the object. In view of this situation, we propose the USB-BBR method, which can optimizes the bounding boxes.

Given all the scored regions in an image, we apply a greedy NMS (independently for each class) algorithm that rejects a region when it has a larger IoU with a higher-scoring selected region.

The NMS method is mainly used to eliminate region overlap; however, it may result in several regions for one object, and some regions may have little overlap with the ground truth. The detection precision and recall will be reduced in such cases. We want to use an optimal bounding box to replace these regions to enhance the location precision of object detection. We use the USB-BBR method to reduce the location errors. All the scored regions are allocated into different groups; each group belongs to one object that needs to be detected. For a set of scored candidate regions $B = \{b_1, b_2, \ldots, b_n\}$ where n is the total number of regions in the set, the regions in B are first sorted by their area in ascending order. The next step is to classify the regions of B into m groups, where a group $G = \{G_1, G_2, \ldots, G_m\}$. The grouping begins with the first sorted region, computing the overlap area ratio of the first sorted region to that from the rest of B. If the overlap area ratio is greater than or equal to a given threshold, the two regions are classified into the same group; otherwise, only the first sorted region is classified into the group. The group process completes when all the regions in B have been computed.

Though this grouping process, each object that needs to be detected may have a group of scored regions. Our goal is to regress the regions of each group G_k, namely, to produce the optimum bounding box for locating the object. We assume that $I_k = (x_k, y_k, w_k, h_k)^T$ is the regressed bounding box of G_k, where (x_k, y_k) is the center coordinate of I_k and (w_k, h_k) are the width and height of I_k, respectively. Thus, the goal is to obtain I_k. For $b_{ki} \in G_k$, $D_{ki} = (x_{ki},\ y_{ki},\ w_{ki},\ h_{ki})^T$ corresponds to the ith scored region in G_k. Given $c_i = I_k - D_{ki}$, we can obtain I_k by

$$L(I_k) = \operatorname{argmin} \sum_{i=1}^{n} u_i c_i^T c_i \tag{1}$$

where u_i is the region score b_{ki} in G_k.

Algorithm 1 USB-BBR

```
Input: The full set of regions $B = \{b_1, b_2, \ldots, b_n\}$, a threshold $\delta$, where $0 < \delta < 1$, a max iteration, $t$, and an iteration step $s$, where $0 < s < \delta$
Set: $G = \{b_1, b_2, \ldots, b_m\}$ $(m = n)$
Set: $R = \{b_1, b_2, \ldots, b_m\}$
$I = \emptyset$
while $t > 0$ and $\delta > 0$ do
  sort the elements of $G$ in ascending order by the corresponding region areas of $R$
  $i = 0$
  while $G \neq \emptyset$ do
    $i = i + 1$
    get the area $a$ of $r$ which is the first element of $R$
    get the first element $G^{'}$ of $G$
    $G_i^{'} = G^{'}$
    remove $G^{'}$ from $G$
    get $L$ (the length of $G$)
    for $j = 1, 2, \ldots, L$ do
      get the overlap area $overa$ between $r_j$ and $r$
      if $overa/a \geq \delta$ then
        $G_i^{'} = G_i^{'} \bigcup G_j$
        remove $G_j$ from $G$
      end if
    end for
  end while
  $m = i$
  $G = \{G_1, G_2, \ldots, G_m\}$ $(G_1 = G_1^{'}, G_2 = G_2^{'}, \ldots, G_m = G_m^{'})$
  update the elements of $R$ according to $G$
  $t = t - 1$
  $\delta = \delta - s$
end while
obtain the final region grouping set $G = \{G_1, G_2, \ldots, G_m\}$
for $k = 1, 2, \ldots, m$ do
  $I_k = argmin \sum_{i=1}^{l} u_i c_i^T c_i$ ($l$ is the length of $G_k$)
  append $I_k$ to I
end for
Output: $I$
```

The first iteration regression result of G is denoted as $R = \{r_1, r_2, \ldots, r_m\}$, computed by (1). The regions in R may belong to the same object, so we use an iterative process to update G. Each iteration uses a descending threshold and the same grouping method to update G by comparing the overlap area ratio of regions in R. Then, R is computed from the updated G. The grouping results occur in the next update iteration, and the iteration process completes when it reaches a specified maximum number of iterations or the threshold is less than or equal to 0. The I is computed by the final grouping set, G. The threshold for the overlap area ratio decreases as the iteration time increases. The USB-BBR algorithm is solved by the least-squares method. Algorithm 1 describes the process of USB-BBR.

Fig. 2 shows the USB-BBR process. After sorting the classified regions in B, the grouping process begins moving

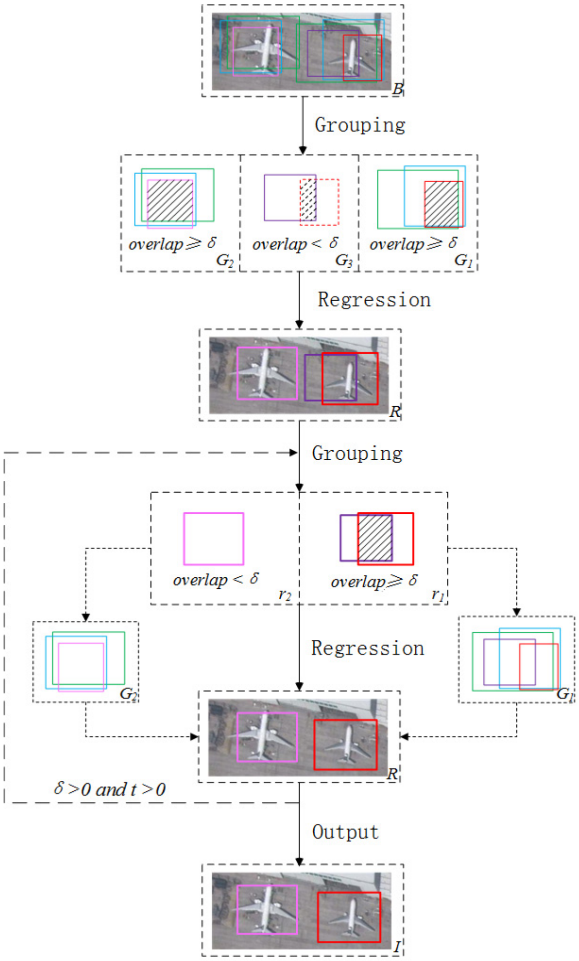

Fig. 2. Illustration of USB-BBR.

from the first sorted region to the last one. All regions are divided into three groups by computing their overlap area ratio. Then, the regression optimal region of each group is computed. The regression result of B is R. The regions in R may still belong to the same object, so we use an iterative process to update G and R. The final regression result is computed when the iterative update process completes. Here, I is the regression result computed by the final grouping set.

IV. Experiments and Results

A. Data Set

Because of the lack of public data sets intended for object detection in remote sensing images, we collected 2326 images downloaded from Google Earth and Tianditu [49]. We labeled the objects in these images with four categories: oil tank, aircraft, overpass, and playground. The image resolution for each class is listed in Table I. The sensors involved are panchromatic and multispectral due to the various sources of the image data sets. In this way, the diversity of the data set poses comprehensive performance challenges.

Using a CNN is different from using other machine learning methods; CNNs need ample training samples to obtain good learning abilities. In addition to the size of the training data set, the quality of the training data set is also critical. A poor-quality training data set—even one containing large amounts of data—also impacts the learning ability of CNNs. To address the object diversity in remote sensing images and the difficult in collecting data for some object classes, we augmented all the positive samples by translation, scale, and rotation transforms. The details of these transforms are listed in Table II. The rotation transform typically performs 90°, 180°, and 270° rotation; however, when there are few instances of a class, the angle of rotation deceases to increase the number of rotations, enlarging the quantity in the data set.

TABLE I
Image Resolution of Each Class

Class	Oil tank	Aircraft	Overpass	Playground
Resolution(m)	$0.3 \sim 1$	$0.5 \sim 2$	$1.25 \sim 3$	$0.4 \sim 1$

TABLE II
Details of the Transforms of Positive Training Samples

Type	argument
Translation Transform	$dw = 0.1w\ dh = 0.1h$
Scale Transform	$ds = [0.8, 0.9, 1.1, 1.2]$
Rotation Transform	$d\theta = 90^\circ, 180^\circ, 270^\circ$

The data set includes two parts: positive samples and negative samples. The collection of positive samples also has two parts. First, we randomly selected 40% of the ground-truth boxes of the collected images as the positive samples for each class. For oil tanks and aircrafts, the translation and scale transform were used to enlarge the number of samples. For overpasses and playgrounds, in addition to the first two transforms, the rotation transform was used to perform data augmentation.

As the second data source for positive samples, we applied the selective search method to produce candidate regions from each image and then computed the IoU between each region and the ground-truth box. When the IoU was greater than or equal to 0.5 compared with a ground-truth region chosen during the first positive sample collection process, the region became a positive sample; otherwise, when the IoU was less than 0.3 with the ground-truth region, the region became a negative sample. This second data source for positive samples was used to enhance the adaptability of the CNN models. Because we used the selective search to generate almost all the candidate regions, these regions do not necessarily completely surround the entire object. To maintain the consistency of the training process and detection process, we added positive samples produced by selective search to the data set. The number of regions from this second data source of positive samples is large; however, there were fewer overpasses than examples of the other classes, which is why the overpass samples required the rotation transform for data augmentation. The negative samples were also obtained through the selective search algorithm by computing the IoU for each class.

The entire data set was divided into a training data set and a validation data set at a ratio of 5:1 (training to validation, respectively). The number of positive samples of each class in

TABLE III
STATISTICS OF THE TRAINING AND VALIDATION DATA SETS

Class	Image Number	Original Object Number	Augment Object Number	Total
Oil tank	165	1586	140498	142084
Aircraft	446	4993	221696	226689
Overpass	176	180	13636	13816
Playground	189	191	27741	27932
Background	2326	668984	0	668984

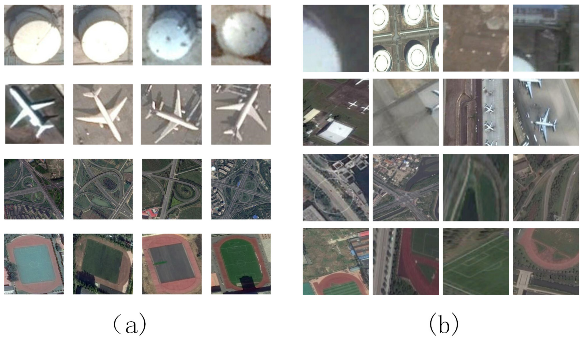

Fig. 3. Examples from the training data set. (a) Positive samples. (b) Negative samples.

TABLE IV
STATISTICS OF THE TEST DATA SET

Class	Oil tank	Aircraft	Overpass	Playground
Object Number	2442	4256	403	413
Image Number	247	668	401	396

the data set is 12 000, containing 2000 validation samples; the number of negative samples for each class is 48 000, containing 8000 validation samples. The ratio of positive samples selected from the first and second positive data set sources is 5:1, and the positive samples from the second part of the positive data set were used to enhance the adaptability of the CNN models. All the CNN data set samples were resized to 227 × 227. Table III shows the data set statistics.

Fig. 3 shows positive and negative training samples. From top to bottom, these samples show an oil tank, aircraft, overpass, and playground.

We evaluated the object detection performance on the test data set. The sizes of the test images ranged from 1044×915 to 1288×992. Objects in the test images have different sizes. The numbers of objects in the test data set are listed in Table IV.

B. Experiment Procedures

The experiment included two procedures, as shown in Fig. 4: the training process and the detection process. The training process used the GPU and the Compute Unified Device Architecture (CUDA) to improve the speed. The end products of the training process are the trained CNN models. The detection process was used to detect objects in test images. It has three main tasks: generate the candidate regions from each test image, extract the features, and obtain the classification results for the candidate regions using the trained CNN models. At the end of this process, the localization precision of these classified regions is enhanced by applying the accurate localization process that included both the non-maximum suppression (NMS) and USB-BBR as described in the preceding section.

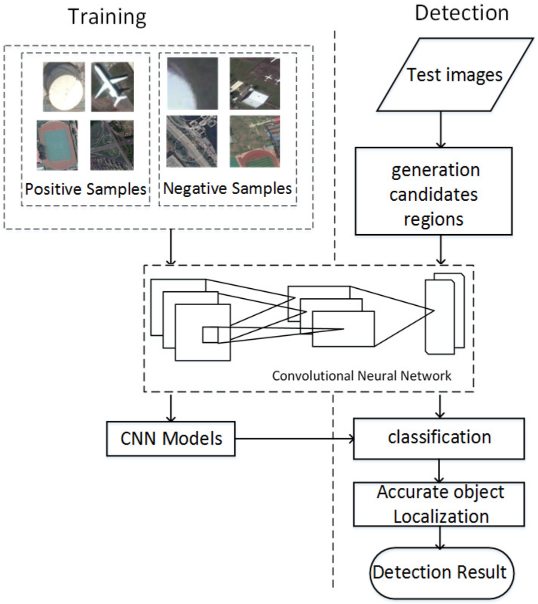

Fig. 4. Experimental process.

TABLE V
ARGUMENTS FOR TRAINING CNN

Argument	value
Learning rate	0.01 (0.001 for finetune)
Batch size	256(AlexNet) 32(GoogleNet)
Dropout	0.5
Momentum	0.9
Weight decay	0.0005
Max iter	50000

We tested three types of models in this paper: a retrained model (AlexNet, GoogleNet, and AlexNet + GoogleNet), a fine-tuned model (AlexNet-finetune, GoogleNet-finetune, and AlexNet-finetune + GoogleNet-finetune), and a dimension-reduction model (AlexNet-DR, GoogleNet-DR, and AlexNet-DR + GoogleNet-DR). There are two ways to initialize network weights in the training process: one is a random initialization using small amounts of data and the other is a fine-tuned initialization by using the trained CNN models on a large data set. The training process was performed using the open source Caffe framework [50].

We used small patches to train our CNN models through the backpropagation algorithm, employing the GPU and CUDA. We set the learning rate to 0.01 and set the batch size to 256 for AlexNet, and 32 for GoogleNet. Moreover, some tricks, such as local response normalization, momentum, overlapping pooling, and dropout, have been used in these networks to improve their properties. The arguments for CNN training are listed in Table V. The training process was run on a RedHat Linux server with the Nvidia GTX Titan X GPU with 12-GB RAM.

TABLE VI

ARGUMENTS OF THE USB-BBR FOR EACH CLASS

Class	γ	$\triangle\gamma$	max iter
Oil tank	0.6	0.025	12
aircraft	0.6	0.025	12
Overpass	0.6	0.09	10
playground	0.6	0.04	10

During the detection process, we ran the selective search method on the test images to extract approximately 1500 proposed regions. Next, we warped the candidate regions and input them to the trained CNN model to perform forward propagation, obtain object features and, finally, classification results. Given all the scored regions in an image, we applied the accurate object localization process to increase the detection performance of these regions. First, the greedy NMS was applied (for each class independently) to reject regions that have an IoU overlap greater than a given threshold with a higher-scoring selected region. This process can greatly decrease the number of overlapped boxes. Second, the USB-BBR method was applied to reduce localization errors and obtain the final detection results.

The arguments for the USB-BBR method are the initial overlap ratio threshold γ, the decreasing step size of γ, denoted as $\triangle\gamma$, and the maximum number of iterations $maxiter$. We used our proposed USB-BBR algorithm on the entire training data set for each type of detection object. We hoped that algorithm applied on the training data set could be easily used on the corresponding validation data set. Based on this thought, for each specific object, the entire training data set was used to obtain the augments for the BBR algorithm, but without cross validation. It is worth mentioning that the regression box founded by the USB-BBR algorithm is insensitive to the initial overlap setting when it is larger than 0.6. Table VI lists the arguments of the USB-BBR method for each class.

We used recall and precision to evaluate the performance of detection result. We obtained the ground-truth area by manual annotation. Recall represents the number of detected objects divided by the total number of actual objects (ground truth), while precision indicates the accuracy of the total detected objects. We use the widely used IoU criterion to evaluate the experiment result using our object localization framework. If the value of IoU is greater than or equal to 0.5, the region is a TruePositive; otherwise, it is a FalsePositive. The IoU for oil tanks, aircraft, and playgrounds is 0.5; for overpasses, it is 0.4 because of the uncertainty of the manually labeled ground truth.

C. Experiment Results

The baseline values for object detection based on our framework are listed in Table VII. These results are from the AlexNet-DR and GoogleNet-DR model combination. Compared with retrained model and fine-tuned model, the dimension-reduction model performs better. Moreover, the results also indicate that the fine-tuned weight initialization can improve both the detection recall and precision, and the model combination yields a better detection performance than does using a single model. In terms of computing cost, the running time of the combination model is approximately the two times of that of a single model. The size of test image we used is 1280 × 1280 pixels.

TABLE VII

BASELINE DETECTION RESULTS OF THE PROPOSED LOCALIZATION FRAMEWORK

class	Recall	Precision	Running Time/per image (s)
Oil tank	0.94472	0.97713	38.2825
Aircraft	0.94995	0.96885	37.2919
Overpass	0.88302	0.87640	5.3598
Playground	0.97183	0.93243	11.928

TABLE VIII

ARGUMENTS FOR EFT-HOG

Argument	$annulus\_num$	$cell\_num$	bin_num	$angle$	f
Value	4	18	6	180	0.75

The overall performance of the feature extraction method used in this paper was compared with two other methods, namely, the local binary pattern histogram Fourier feature (LBP-HF) [51] and the EFT-HOGs [6]. The LBP-HF combines a discrete Fourier Transform and the LBP to obtain the rotational invariance feature. The arguments for LBP-HF are the radius and the number of sampling points, which determine the circular region used. In our experiment, the radius was set to 3, and the number of sampling points was 24. EFT-HOG uses a circle HOG (C-HOG) feature rather than the typical rectangle HOG (R-HOG) feature, because the C-HOG feature has a better rotational invariance than does the R-HOG feature. To further strengthen the rotation invariance of the C-HOG descriptors, the author mapped the features to the Cartesian coordinate system and then performed an EFT.

There are several optional EFT-HOG arguments: the numbers of annuli, cells, bins, angle, and the number of adopted elliptic Fourier coefficients divided by the number of total coefficients. These are denoted as $annulus\_num$, $cell\_num$, bin_num, $angle$, and f, respectively. The images in the data set were resized to 227 × 227; therefore, the EFT-HOG arguments for the four tested classes are the same as those listed in Table VIII.

After extracting features from the training data set, the feature descriptors from LBP-HF and EFT-HOG were used to train a support vector machines (SVMs) classifier. The kernel function for SVM training is the linear function, and a fivefold cross-validation method is used to avoid overfitting. The trained SVM classifier can be used to classify the candidate object regions. The candidate regions are the same as those used for the proposed CNN-based method. The classified candidate regions also underwent the accurate object localization process to obtain the final detection result. The performance comparisons between LBP-HF, EFT-HOG, and the CNN-based method are listed in Table IX. We can see that the CNN-based method performs than the artificial features. Fig. 5 shows a portion of the comparison of the detection results of the CNN, LBP-HF, and EFT-HOG methods. From top to bottom are the results of detecting the oil tank, aircraft,

TABLE IX
PERFORMANCE COMPARISONS OF LBP-HF, EFT-HOG, AND THE CNN-BASED METHOD

class	CNN-based		LBP-HF		EFT-HOG	
	Recall	Precision	Recall	Precision	Recall	Precision
Oil tank	**0.94472**	**0.97713**	0.67639	0.56447	0.73301	0.65825
Aircraft	**0.94995**	**0.96885**	0.68580	0.58134	0.73120	0.64041
Overpass	**0.88302**	**0.87640**	0.60300	0.41495	0.71457	0.60256
Playground	**0.97183**	**0.93243**	0.60527	0.40116	0.70526	0.58989

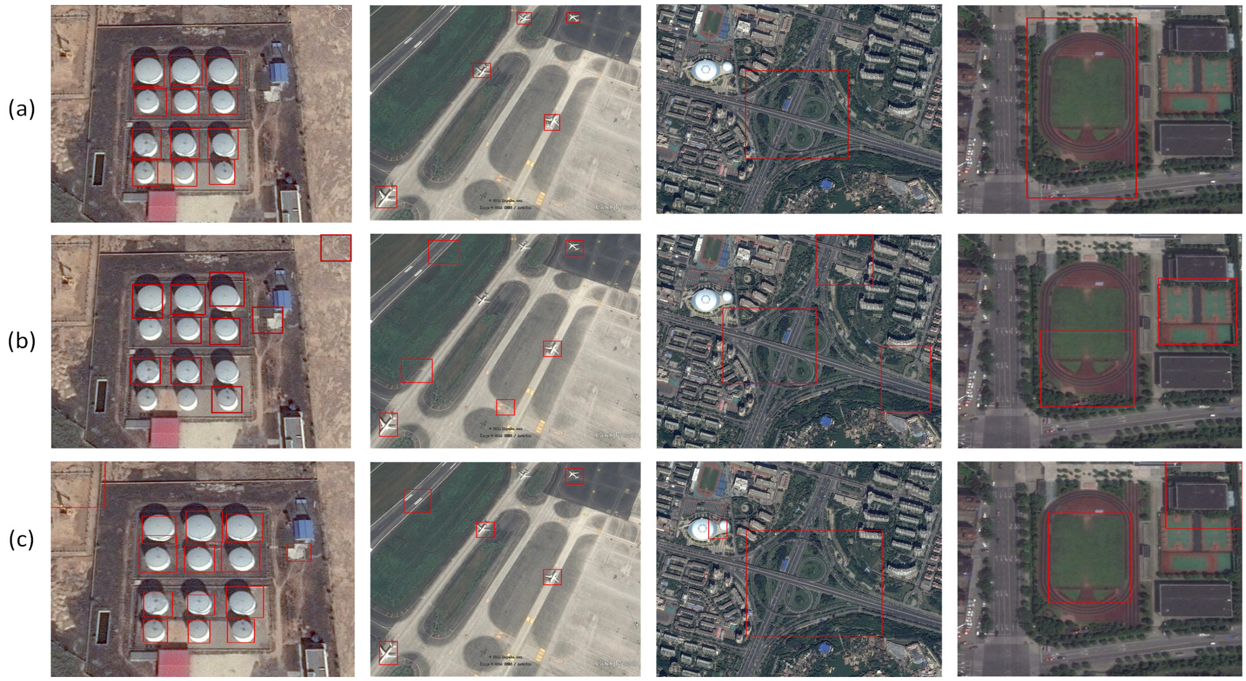

Fig. 5. Comparison of the detection results for the CNN, LBP-HF, and EFT-HOG methods. From left to right, the results are for oil tanks, aircraft, overpasses, and playgrounds. (a) Detection results of the CNN method. (b) Detection results of the LBP-HF method. (c) Detection results of the EFT-HOG method.

overpass, and playground classes; Fig. 5(a) is the detection result of the CNN method, Fig. 5(b) is the detection result of LBP-HF method, and Fig. 5(c) is the result of the EFT-HOG method. As listed in the results, the LBP-HF method suffers from many false detections and many missed detections. The EFT-HOG method achieves a better detection performance than LBP-HF, but it is worse than the CNN method.

The detection performance of the proposed localization framework based on the CNN method is significantly better than the performances of LBP-HF and EFT-HOG detection frameworks. The last two object feature extraction methods enhance invariance through human intervention, but still result in a worse performance than CNN. These results also indicate that CNN can learn better-quality features than those designed by human. CNN needs only to train on the data set to obtain the intrinsic features; it does not require human intervention. CNN can conveniently extract features from a data set when the data quality and quantity of the data set meet the needs of the employed CNN models. This drastically reduces the work and difficulty involved in feature extraction.

Fig. 6 shows the examples of detection failures of the proposed localization framework. The yellow boxes denote false negatives; the red boxes are detected positives by the proposed framework. FP denotes false positives. For the oil tanks, many false negatives have low gray values; these may be difficult to distinguish from the background. As for aircraft, objects of other classes that appear similar to aircraft may be mistakenly classified as aircraft. In analyzing the failure detection results for overpasses, we found that the regions of overpasses could not easily be determined to be correct, because the manually labeled ground truth was too subjective. If a classified positive region has a low IoU value, that region is evaluated as a false positive. The playground detection results show the same situation. Other objects that have similar color compositions or shapes may also be misclassified as playgrounds.

We found that the recall and precision for overpass detection were noticeably lower than the recall and precision for the other three classes—but this result is not caused by CNNs inability to learn overpass features. Instead, the lower detection result for overpasses is caused by the manually labeled ground truth. Overpass regions are subjective; thus, it is difficult to ensure objectively correct regions for overpasses. The detection evaluation index in our experiment is strict; consequently, it results in poor detection result for overpasses. When we changed the detection evaluation index slightly, the detection performance for overpasses increased.

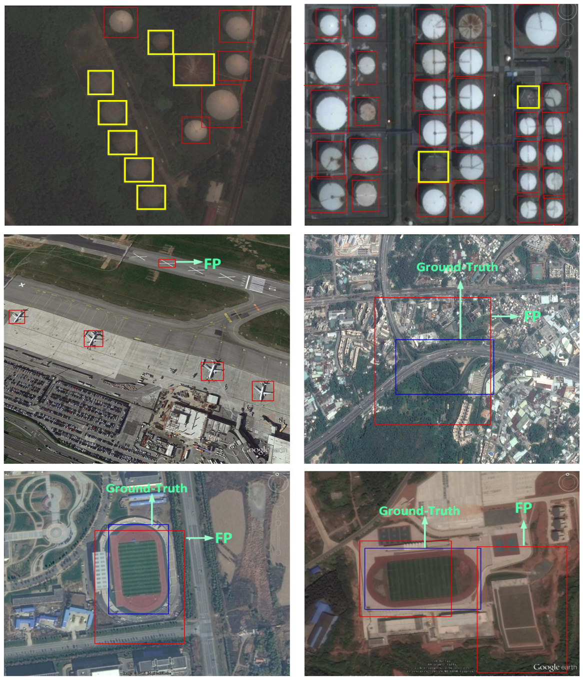

Fig. 6. Examples of detection result failures for the proposed localization framework. The yellow boxes denote FalseNegative and the red boxes are detected positives by the proposed framework. FP: FalsePositive.

TABLE X

DETECTION RESULTS FOR OVERPASSES USING DIFFERENT EVALUATION INDEXES

Evaluate Index	Recall	Precision
$IoU = 0.3$	0.96604	0.94485
$IoU = 0.4$	0.88302	0.87640
$IoU = 0.5$	0.68302	0.66544

Table X shows the detection result for overpasses using this altered evaluation index. Fig. 7 shows a portion of the detection results with different evaluation index values. Fig. 7(a) is the detection result when IoU = 0.5, Fig. 7(b) is the detection result when IoU = 0.4, and Fig. 7(c) is the detection result when IoU = 0.3. For each result, the blue box is the ground truth, while the green and red boxes are the FalsePositive and TruePositive, respectively, as judged by the evaluation index. We found that the regions evaluated as FalsePositive using IoU = 0.5 are instead judged as TruePositive by the last two evaluation indexes. This indicates that if we choose a less rigorous evaluation, index causes the recall and precision for overpass detection to increase. In our main experiment, we used the stricter evaluation index to measure the performance of the proposed localization framework. The experimental results demonstrate that the proposed localization framework has a better detection performance than the compared methods and has good location precision.

V. ANALYSIS

In this section, we analyze the effects of the proposed framework on detection performance. For the proposed object localization framework, we used a large data set to train the CNN models and applied model fine-tuning to initialize the weighs. We also used a model combination method when classifying candidate regions, and the accurate object localization process to enhance the location precision. We will systematically analyze the performance effects caused by these strategies.

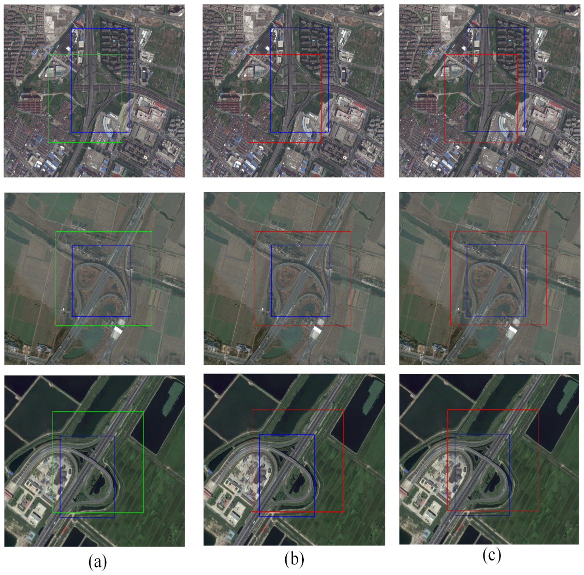

Fig. 7. Detection result for overpasses using different evaluation indexes. (a) Detection result when IoU = 0.5. (b) Detection result when IoU = 0.4. (c) Detection result when IoU = 0.3. For each result, the blue box shows the ground truth, the green box is the FalsePositive judged by the evaluation index, and the red box is the TruePositive judged by the evaluation index.

TABLE XI

PLAYGROUND DETECTION RESULTS OF CNN MODELS TRAINED ON DIFFERENT DATA SET SIZES

Size	GoogleNet		GoogleNet-finetune	
	Recall	Precision	Recall	Precision
P-500	0.75180	0.58543	0.85614	0.74618
P-1000	0.82971	0.75578	0.87018	0.78481
P-3000	0.86022	0.76190	0.89825	0.83660
P-5000	0.88087	0.81605	0.92982	0.87458
P-11000	**0.94035**	**0.89333**	**0.94035**	**0.89333**

TABLE XII

OIL TANK DETECTION RESULTS OF CNN MODELS TRAINED ON DIFFERENT DATA SET SIZES

Size	GoogleNet		GoogleNet-finetune	
	Recall	Precision	Recall	Precision
P-500	0.88038	0.80752	0.93295	0.91552
P-1000	0.91474	0.83365	0.94412	0.91204
P-3000	**0.93502**	0.89536	0.94578	0.93113
P-5000	0.93416	0.92007	0.94992	**0.95506**
P-11000	0.91769	**0.93923**	**0.96723**	0.92482

A. Size of Training Data Set

It is known that CNN models need ample training data sets to learn the essential features of tasks. However, creating a large number of data set examples with labels involves a great deal of effort and time. Moreover, some classes have few available samples for collection. In many cases, the size of the training data set is insufficient. Still, the important determining

TABLE XIII
COMPARISON OF THE DETECTION RESULTS OF ALEXNET, GOOGLENET, AND THEIR MODEL COMBINATION

Class	AlexNet		GoogleNet		AlexNet+GoogleNet	
	Recall	Precision	Recall	Precision	Recall	Precision
Oil tank	**0.93489**	0.93642	0.91769	0.93923	0.90287	**0.97564**
Aircraft	0.91518	0.92959	0.88628	0.83359	**0.92552**	**0.96426**
Overpass	**0.86364**	**0.76510**	0.76515	0.52604	0.83969	0.69401
Playground	0.84561	0.67697	0.94035	0.89333	**0.95789**	**0.93814**

TABLE XIV
COMPARISON OF THE DETECTION RESULTS OF ALEXNET-FINETUNE, GOOGLENET-FINETUNE, AND THEIR MODEL COMBINATION

Class	AlexNet-finetune		GoogleNet-finetune		AlexNet-finetune+GoogleNet-finetune	
	Recall	precision	Recall	Precision	Recall	Precision
Oil tank	0.93857	0.97532	**0.96724**	0.92482	0.94185	**0.98207**
Aircraft	0.91236	0.92895	0.93092	0.94694	**0.93351**	**0.97473**
Overpass	0.87170	0.82206	**0.89057**	0.84286	0.87547	**0.85294**
Playground	0.86316	0.79355	**0.94035**	0.89333	0.93684	**0.90508**

TABLE XV
COMPARISON OF THE DETECTION RESULTS OF ALEXNET-DR, GOOGLENET-DR, AND THEIR MODEL COMBINATION

Class	AlexNet-DR		GoogleNet-DR		AlexNet-DR+GoogleNet-DR	
	Recall	Precision	Recall	Precision	Recall	Precision
Oil tank	0.95250	0.95367	**0.95373**	0.95962	0.94472	**0.97713**
Aircraft	0.90249	0.86685	0.94784	0.94540	**0.94995**	**0.96885**
Overpass	0.87170	0.79655	**0.91698**	**0.87726**	0.88302	0.87640
Playground	0.87676	0.79048	0.95789	0.89508	**0.97183**	**0.93243**

factor for the performance of a CNN is the training data set. If either the quantity or quality of the training data set is poor, the CNN model will have poor learning ability even when a good network architecture is adopted. Therefore, we first analyze the effects of different training data set sizes on detection performance.

In our detection experiments, we used four object classes: oil tank, aircraft, overpass, and playground. We chose the oil tank and playground classes specifically to analyze the effects of different training data set sizes on detection performance. The numbers of samples of oil tanks and playgrounds in the different sized positive training data sets are [500, 1000, 3000, 5000, and 11000]. The number of negative samples in each training data set was four times the number of positive samples. The ratio of samples in the training data set to those in the validation data set was 5:1. We trained the GoogleNet CNN model on these different size data sets. The weight initialization included both the random values strategy and the fine-tuning strategy. Then, all these CNN models were applied to extract features and obtain the classification results. We compared the recall and precision of each CNN model's object detection result to analyze the effects of training data set size.

Tables XI and XII separately show oil tank detection results and playground detection results trained using different data set sizes. These detection results indicate that the CNN models require ample numbers of samples to achieve good learning ability. However, there is a peek point where the recall or precision achieves the highest value, and when the number of data set increased, the value of recall or precision decreased.

B. Feature Extraction and Classification

The common method for obtaining class labels of candidate regions is to perform forward propagation in the CNN model. In our experiment, in addition to this method, we also used a model combination to obtain the class labels of candidate regions. The model combination is used to reduce the false detection rate by using a feature combination from two different CNN models. The model combination is similar to performing forward propagations twice, and it obtains the classification result from the outputs of two CNN models.

In our experiments, we tested three types of models in this paper: a retrained model, a fine-tuned model, and a dimension-reduction model, as shown in Tables XIII–XV. These results show that weight initializations play a crucial role in the learning ability of CNN models, and the model combination strategy can improve the precision value to a certain extent. The detection results of the fine-tuned model are listed in Table XIV. From an analysis of the detection performance listed in table XIII, XIV, and XV, the fine-tuned methods result in better performance in detection results, causing an increase in recall and precision to a certain extent. Namely, the fine-tuned model can increase the learning performance of CNN models. And the detection precision of a combination model is higher than that of a single model.

Beyond the fine-tuning strategy and model combination strategy, we also explored three dimension-reduction models: AlexNet-DR, GoogleNet-DR, and AlexNet-DR + GoogleNet-DR, all of which added a 64-D inner-product layer before the last inner-product layer. The parameters are initialized by the pretrained models that trained on ImageNet data. When applying the fine-tuning method, the learning rate also decreases by 0.1%. The results of the dimension-reduction models are listed in Table XV.

C. Region Proposal and Unsupervised Score-Based Bounding Box Regression

In the first stage, we use the selective search method to obtain a lower quantity of high quality object regions. The performance of the region proposal is profoundly affected

TABLE XVI
COMPARISON DETECTION RESULTS OF REGION PROPOSAL AND USB-BBR USING GOOGLENET-FINETUNE MODEL

Class	region proposal		CNN		CNN with NMS		CNN with NMS and USB-BBR	
	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision
Oil tank	**0.99959**	0.22273	0.98607	0.85488	0.98567	0.83120	0.96724	**0.92482**
Aircraft	**0.99436**	0.13526	0.99272	0.89058	0.98919	0.89489	0.93092	**0.94694**
Overpass	**0.98868**	0.06164	0.98868	0.38624	0.92830	0.24707	0.89057	**0.84286**
playground	**0.98947**	0.02494	0.98947	0.37726	0.97163	0.58386	0.94035	**0.89333**

by the number of object locations and the criterion IoU criterion. To evaluate the performance of the region proposals on our test data set, we took all object locations generated by selective search into account and used an IoU of 0.4 for overpasses and 0.5 for the other three types to calculate the effect of this criterion on recall. The results are listed in Table XVI. As shown in Table XVI, the recall criterion of region proposal gradually decreases while the precision increases, indicating that the CNN detection and our localization process can improve the accuracy of the object location. This decrease in recall is expected, because after CNN detection and NMS, fewer boxes are predicted as objects of interest than before. But when using the USB-BBR, the precision improves, because the locations of regions have been optimized as well as the number of bounding boxes, particularly the bounding boxes of false positive regions, which decreased over the entire object detection framework.

In addition to the region proposals, we also tested the recall criterion of boxes detected by the GoogleNet-finetune model and the GoogleNet-finetune with NMS model. We obtained the classification results of candidate regions by the forward propagation of CNN. There are many overlapping regions in the classification results. The NMS method was used to eliminate smaller regions that have an IoU greater than a threshold value with a higher-scoring region. The number of overlapped regions decreases considerably. However, even after NMS the results still did not meet our need, because each object always has many regions.

We hoped to obtain an optimal bounding box to locate the objects more precisely. To deal with the problem of several regions corresponding to one object, we used the USB-BBR algorithm after NMS. The detection results of using the GoogleNet-finetune model with the USB-BBR method are also listed in Table XVI. Compared with the result from using a CNN with NMS, using a CNN with USB-BBR can greatly increase the localization precision, because the process optimizes several regions into one region; consequently, the location accuracy is much higher, especially for overpasses and playgrounds. As Table XVI shows, the recalls of the detection results with USB-BBR are lower than these of the GoogleNet-finetune model without USB-BBR; however, this is expected because there is an interconstraint relationship between recall and precision. The region numbers of detected regions with and without the USB-BBR for the GoogleNet-finetune model are listed in Table XVII. The average number of regions for each class is the total region number divided by the total number of test objects. The average number of overpasses and playgrounds regions after USB-BBR is greatly decreased. This result leads to the reduced detection recall values for these two classes after USB-BBR.

TABLE XVII
NUMBER OF REGIONS DETECTED BY THE GOOGLENET-FINETUNE MODEL BOTH WITH AND WITHOUT THE USB-BBR METHOD

Class	GoogleNet-finetune		GoogleNet-finetune with BBR	
	Total	Average	Total	Average
Oil tank	2897	1.2	2543	**1.0**
Aircraft	4719	1.1	4136	**1.0**
Overpass	1449	5.5	266	**1.0**
Playground	471	1.7	303	**1.1**

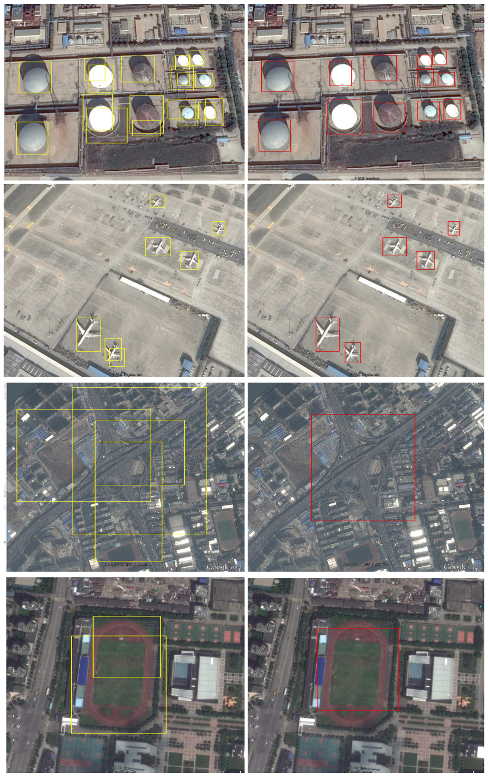

Fig. 8. Comparative detection results both with and without USB-BBR. The first column shows the detection results without USB-BBR. The second column shows the detection results with USB-BBR.

Fig. 8 shows a portion of the comparison detection results both with and without the USB-BBR method. The first column is the detection result without using the bounding box regression method; the second column is the detection result using the USB-BBR method. As Fig. 8 shows, the USB-BBR has better localization precision. Therefore, the USB-BBR method can increase the detection localization precision.

VI. CONCLUSION

In this paper, we proposed an object localization framework based on CNN in remote sensing images. The framework uses the CNN models to extract object features and obtain classification results. In the first stage, we used a selective search method to generate the major part of the candidate object regions. In the second stage, we designed a dimension-reduction model using trained models to initialize the network weights and then use it to extract features and classify the objects to different categories. We also tested a retrained model and a fine-tuned model. In the third stage, we proposed a new USB-BBR algorithm, as part of the accurate object localization process, to obtain better detection localization precision, and we used NMS to decrease the number of overlapped regions. The addition of the USB-BBR method can help to obtain an optimal bounding box for each group of classified regions. In addition, we investigated the influences of different sizes of training data sets, different weight initialization methods, and different model combinations on detection performance. These results can help guide other researchers to obtain good results. The results of the experiments indicate that the proposed localization framework is both simple and robust. In further work, we will continue to enhance this framework and improve its detection and localization performance.

References

[1] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015.

[2] W. Shao, W. Yang, G. Liu, and J. Liu, "Car detection from high-resolution aerial imagery using multiple features," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2012, pp. 4379–4382.

[3] F. Zhang, B. Du, and L. Zhang, "Saliency-guided unsupervised feature learning for scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 4, pp. 2175–2184, Apr. 2015.

[4] J. Han *et al.*, "Efficient, simultaneous detection of multi-class geospatial targets based on visual saliency modeling and discriminative learning of sparse coding," *ISPRS J. Photogram. Remote Sens.*, vol. 89, pp. 37–48, Mar. 2014.

[5] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.

[6] Z. Xiao, Q. Liu, G. Tang, and X. Zhai, "Elliptic Fourier transformation-based histograms of oriented gradients for rotationally invariant object detection in remote-sensing images," *Int. J. Remote Sens.*, vol. 36, no. 2, pp. 618–644, 2014.

[7] G. Cheng and J. Han, "A survey on object detection in optical remote sensing images," *ISPRS J. Photogram. Remote Sens.*, vol. 117, pp. 11–28, Mar. 2016.

[8] W. Zhang, X. Sun, K. Fu, C. Wang, and H. Wang, "Object detection in high-resolution remote sensing images using rotation invariant parts based model," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 74–78, Jan. 2014.

[9] Y. Huang, L. Zhang, P. Li, and Y. Zhong, "High-resolution hyper-spectral image classification with parts-based feature and morphology profile in urban area," *Geo-Spatial Inf. Sci.*, vol. 13, no. 2, pp. 111–122, 2010.

[10] G. Cheng, J. Han, P. Zhou, and L. Guo, "Multi-class geospatial object detection and geographic image classification based on collection of part detectors," *ISPRS J. Photogramm. Remote Sens.*, vol. 98, pp. 119–132, Dec. 2014.

[11] G. Cheng, J. Han, L. Guo, Z. Liu, S. Bu, and J. Ren, "Effective and efficient midlevel visual elements-oriented land-use classification using VHR remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 8, pp. 4238–4249, Aug. 2015.

[12] Y. Li, X. Sun, H. Wang, H. Sun, and X. Li, "Automatic target detection in high-resolution remote sensing images using a contour-based spatial model," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 5, pp. 886–890, Sep. 2012.

[13] H. Sun, X. Sun, H. Wang, Y. Li, and X. Li, "Automatic target detection in high-resolution remote sensing images using spatial sparse coding bag-of-words model," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 1, pp. 109–113, Jan. 2012.

[14] G. Cheng, L. Guo, T. Zhao, J. Han, H. Li, and J. Fang, "Automatic landslide detection from remote-sensing imagery using a scene classification method based on BoVW and pLSA," *Int. J. Remote Sens.*, vol. 34, no. 1, pp. 45–59, 2013.

[15] X. Yao, J. Han, G. Cheng, L. Guo, and X. Qian, "Semantic annotation of high-resolution satellite images via weakly supervised learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 1–12, Jun. 2016.

[16] Y. Zhong, F. Fei, and L. Zhang, "Large patch convolutional neural networks for the scene classification of high spatial resolution imagery," *J. Appl. Remote Sens.*, vol. 10, no. 2, p. 025006, 2016.

[17] P. Agrawal, R. Girshick, and J. Malik, "Analyzing the performance of multilayer neural networks for object recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 329–344.

[18] H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson, "Factors of transferability for a generic convnet representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1790–1802, Sep. 2015.

[19] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," in *Proc. CVPR*, Jun. 2014, pp. 806–813.

[20] L. Bottou, "Stochastic gradient descent tricks," *Neural Netw., Tricks Trade*, vol. 1, no. 1, pp. 421–436, 2012.

[21] Y.-L. Boureau, J. Ponce, and Y. LeCun, "A theoretical analysis of feature pooling in visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 111–118.

[22] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2013, pp. 8609–8613.

[23] A. Grubb and J. A. Bagnell, "Stacked training for overfitting avoidance in deep networks," in *Proc. Int. Conf. Mach. Learn.*, Oct. 2013, p. 1.

[24] A. G. Howard. "Some improvements on deep convolutional neural network based image classification." Unpublished paper, 2013. [Online]. Available: https://arxiv.org/abs/1312.5402

[25] A. Mahendran and A. Vedaldi. "Understanding deep image representations by inverting them." Unpublished paper, 2014. [Online]. Available: https://arxiv.org/abs/1412.0035

[26] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proc. Int. Conf. Mach. Learn.*, May 2013, pp. 1139–1147.

[27] J. Dean *et al.*, "Large scale distributed deep networks," in *Proc. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1–11.

[28] A. Krizhevsky. (2014). "One weird trick for parallelizing convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1404.5997

[29] T. Paine, H. Jin, J. Yang, Z. Lin, and T. Huang. "GPU asynchronous stochastic gradient descent to speed up neural network training." Unpublished paper, 2013. [Online]. Available: https://arxiv.org/abs/1312.6186

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2012, pp. 1097–1105.

[31] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Sep. 2015, pp. 1–14.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[34] K. He, X. Zhang, S. Ren, and J. Sun. "Deep residual learning for image recognition." Unpublished paper, 2015. [Online]. Available: https://arxiv.org/abs/1512.03385

[35] P. Zhou, D. Zhang, G. Cheng, and J. Han, "Negative bootstrapping for weakly supervised target detection in remote sensing images," in *Proc. IEEE Int. Conf. Multimedia Big Data*, Apr. 2015, pp. 318–323.

[36] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, 2015.

[37] A.-B. Salberg, "Detection of seals in remote sensing images using features extracted from deep convolutional neural networks," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 1893–1896.

[38] L. Zhang, Z. Shi, and J. Wu, "A hierarchical oil tank detector with deep surrounding features for high-resolution optical satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 10, pp. 1–15, Oct. 2015.

[39] T. Ishii, R. Nakamura, H. Nakada, Y. Mochizuki, and H. Ishikawa, "Surface object recognition with CNN and SVM in Landsat 8 images," in *Proc. IAPR Int. Conf. Mach. Vis. Appl.*, May 2015, pp. 3–6.

[40] H. Wu, H. Zhang, J. Zhang, and F. Xu, "Typical target detection in satellite images based on convolutional neural networks," in *Proc. IEEE Int. Conf. Syst., Man*, Oct. 2015, pp. 2956–2961.

[41] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[42] Q. Jiang, L. Cao, M. Cheng, C. Wang, and J. Li, "Deep neural networks-based vehicle detection in satellite images," in *Proc. Int. Symp. Bioelectron. Bioinf.*, Oct. 2015, pp. 184–187.

[43] H. Zhu, X. Chen, W. Dai, K. Fu, Q. Ye, and J. Jiao, "Orientation robust object detection in aerial images using deep convolutional neural network," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 3735–3739.

[44] J. Ding, B. Chen, H. Liu, and M. Huang, "Convolutional neural network with data augmentation for SAR target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 3, pp. 364–368, Mar. 2016.

[45] I. Ševo and A. Avramović, "Convolutional neural network based automatic object detection on aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 5, pp. 740–744, May 2016.

[46] F. Zhang, B. Du, L. Zhang, and M. Xu, "Weakly supervised learning based on coupled convolutional neural networks for aircraft detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5553–5563, Sep. 2016.

[47] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

[48] J. Hosang, R. Benenson, and B. Schiele. "How good are detection proposals, really?" Unpublished paper, 2014. [Online]. Available: https://arxiv.org/abs/1406.6962

[49] *Tianditu*, accessed on Feb. 1, 2016. [Online]. Available: http://map.tianditu.com/map/index.html

[50] Y. Jia *et al.* "Caffe: Convolutional architecture for fast feature embedding." Unpublished paper, 2014. [Online]. Available: https://arxiv.org/abs/1408.5093

[51] G. Zhao, T. Ahonen, J. Matas, and M. Pietikäinen, "Rotation-invariant image and video description with local binary pattern features," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1465–1477, Apr. 2012.

Yang Long received the B.S. degree in resources environment and the management of urban and rural planning from Hubei Normal University, Huangshi, China, in 2014. He is currently pursuing the M.S. degree with the State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan, China.

His research interests include deep learning and image retrieval.

Yiping Gong received the B.S. degree in geographic information system from Lanzhou University, Lanzhou, China, in 2015. She is currently pursuing the M.S. degree with the State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan, China.

Her research interests include deep learning and object detection.

Zhifeng Xiao received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2008.

From 2014 to 2015, he was a Visiting Scholar with the Computational Biomedicine Imaging and Modeling Center, Rutgers University, New Brunswick, NJ, USA. He is currently an Associate Professor with the State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University. His research interests include remote sensing image processing, computer vision, and machine learning. His work consists of object detection in remote sensing images, large-scale content-based remote sensing image retrieval, and scene analysis on remote sensing images.

Qing Liu received the M.S. degree from the State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan, China, in 2016.

She is currently with an artificial intelligence company, Shenzhen Prafly Technology Co., Ltd., Shenzhen, China. Her research interests include deep learning, object detection, and machine learning.