Learning Rotation-Invariant Convolutional Neural Networks for Object Detection in VHR Optical Remote Sensing Images

Gong Cheng, Peicheng Zhou, and Junwei Han

***Abstract*—Object detection in very high resolution optical remote sensing images is a fundamental problem faced for remote sensing image analysis. Due to the advances of powerful feature representations, machine-learning-based object detection is receiving increasing attention. Although numerous feature representations exist, most of them are handcrafted or shallow-learning-based features. As the object detection task becomes more challenging, their description capability becomes limited or even impoverished. More recently, deep learning algorithms, especially convolutional neural networks (CNNs), have shown their much stronger feature representation power in computer vision. Despite the progress made in nature scene images, it is problematic to directly use the CNN feature for object detection in optical remote sensing images because it is difficult to effectively deal with the problem of object rotation variations. To address this problem, this paper proposes a novel and effective approach to learn a rotation-invariant CNN (RICNN) model for advancing the performance of object detection, which is achieved by introducing and learning a new rotation-invariant layer on the basis of the existing CNN architectures. However, different from the training of traditional CNN models that only optimizes the multinomial logistic regression objective, our RICNN model is trained by optimizing a new objective function via imposing a regularization constraint, which explicitly enforces the feature representations of the training samples before and after rotating to be mapped close to each other, hence achieving rotation invariance. To facilitate training, we first train the rotation-invariant layer and then domain-specifically fine-tune the whole RICNN network to further boost the performance. Comprehensive evaluations on a publicly available ten-class object detection data set demonstrate the effectiveness of the proposed method.**

***Index Terms*—Convolutional neural networks (CNNs), machine learning, object detection, remote sensing images, rotation-invariant CNN (RICNN).**

I. Introduction

OBJECT detection in very high resolution (VHR) optical remote sensing images is a fundamental problem faced for aerial and satellite image analysis. In recent years, due to the advance of the machine learning technique, particularly the powerful feature representations and classifiers, many approaches regard object detection as a classification problem and have shown impressive success for some specific object detection tasks [1]–[24]. In these approaches, object detection can be performed by learning a classifier, such as support vector machine (SVM) [1], [7], [8], [12], [13], [20]–[24], AdaBoost [2]–[5], k-nearest neighbors [15], [17], conditional random field [6], [19], and sparse-coding-based classifier [9]–[11], [14], [16], which captures the variation in object appearances and views from a set of training data in a supervised [2]–[7], [9]–[14], [16]–[21], [23], [24] or semisupervised [15], [22] or weakly supervised framework [1], [8], [25], [51]. The input of the classifier is a set of image regions with their corresponding feature representations, and the output is their predicted labels, i.e., object or not. A recent review on object detection in optical remote sensing images can be found in [26].

As object detection is usually carried out in feature space, effective feature representation is very important to construct high-performance object detection systems. During the last decades, considerable efforts have been made to develop various feature representations for the detection of different types of objects in satellite and aerial images. Among various features developed for visual object detection, the bag-of-words (BoW) model [27] is maybe one of the most popular methods. The BoW model treats each image region as a collection of unordered local descriptors [28]–[30], quantizes them into a set of visual words, and then computes a histogram representation. The main advantages of the BoW model are its simplicity, efficiency, and invariance under viewpoint changes and background clutter. Consequently, the BoW model and its variants have been widely adopted by the community and have shown good performance for geospatial object detection [8], [12], [13], [17], [31].

The histogram of oriented gradients (HOG) feature [32] is another kind of features that has been successfully applied to remote sensing image analysis. The HOG feature was first proposed by Dalal and Triggs [32] to represent objects by the distribution of gradient intensities and orientations in spatially distributed regions. It has been widely acknowledged as one of the best features to capture the edge or local shape information of the objects. Since its introduction, it has shown great success in many object detection tasks [3], [4], [22], [33]. Moreover, some part model-based methods and the sparselets work [34]–[36] that built on the HOG feature have also shown impressive performance.

With the advent of compressed sensing, sparse-coding-based feature representations have been recently employed in remote

Manuscript received April 27, 2016; revised July 24, 2016; accepted August 11, 2016. Date of publication September 5, 2016; date of current version September 30, 2016. This work was supported in part by the National Science Foundation of China under Grant 61401357 and Grant 61473231, by the Fundamental Research Funds for the Central Universities under Grant 3102016ZY023, by the Innovation Foundation for Doctor Dissertation of NPU under Grant CX201622, and by the Aerospace Science Foundation of China under Grant 20140153003. (*Corresponding author: Junwei Han.*)

The authors are with the School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: junweihan2010@gmail.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.2016.2601622

sensing image analysis [9]–[11], [14]–[16], [37], [38] and have shown excellent performances. The core idea of sparse coding is to sparsely encode high-dimensional original signals by a few structural primitives in a low-dimensional manifold. The procedure of seeking the sparsest representation for the test sample in terms of an overcomplete dictionary (including both target and background samples) endows itself with a discriminative nature to perform classification. The sparse-coding-based feature can be calculated via resolving a least-squares-based optimization problem with sparse-norm regularization constraints. Furthermore, some texture features including, but are not limited to, Gabor feature, local binary pattern feature, and shape-based invariant texture feature [39] that aim to describe the local density variability and patterns inside the surface of an object are also developed for identifying textural objects such as airport [5], urban area [6], and vehicles [4], [18].

However, while the aforementioned feature representations have shown impressive success for some specific object detection tasks, they all are handcrafted features or shallow-learning-based features. The involvement of human ingenuity in feature design or shallow structure significantly influences the representational power as well as the effectiveness for object detection. Particularly as the visual recognition task becomes more challenging, the description capability of those features may become limited or even impoverished.

In 2006, a breakthrough in deep learning was made by Hinton and Salakhutdinov [40]. Since then, deep learning algorithms particularly convolutional neural networks (CNNs) have shown their stronger feature representation power in a wide range of computer vision applications [41]–[46]. On the one hand, compared with traditional human-engineering-based features that involve abundant human ingenuity for features design, the CNN feature relies on neural networks of deep architecture to directly generate feature representations from raw image pixels. Thus, the burden for feature design has been transferred to the network construction. On the other hand, compared with shallow learning techniques for feature representations (e.g., sparse coding), the deep architecture of the CNN model can extract much more powerful feature representation. Moreover, the CNN feature extracted from higher layers of the deep neural network displays more semantic abstracting properties, which can lead to significant performance improvement on object detection.

However, although the CNN feature has achieved great success in nature scene images, it is problematic to directly use it for object detection in optical remote sensing images because it is difficult to effectively handle the problem of object rotation variations. Essentially, this problem is not critical for nature scene images because the objects are typically in an upright orientation due to the Earth's gravity, and so, orientation variations across images are generally small. On the contrary, objects in remote sensing images such as airports, buildings, and vehicles usually have many different orientations since remote sensing images are taken from the upper airspace.

To tackle this problem, in this paper, we propose a novel and effective approach to learn a rotation-invariant CNN (RICNN) model for geospatial object detection, which is achieved by introducing and learning a new rotation-invariant layer on the basis of the existing high-capacity CNN architectures (e.g., AlexNet CNN [41]). However, different from the training of traditional CNN models that only optimizes the multinomial logistic regression objective, the proposed RICNN model is trained by optimizing a new objective function via imposing a regularization constraint, which enforces the training samples before and after rotating to share the similar features to achieve rotation invariance. To facilitate training, we first train the newly added rotation-invariant layer, with the remaining model parameters being fixed and then domain-specifically fine-tune the whole RICNN network with a smaller learning rate to further boost the performance. Comprehensive evaluations on a publicly available ten-class VHR object detection data set and comparisons with state-of-the-art methods, including traditional CNN features without considering rotation-invariant information, demonstrate the effectiveness of the proposed method.

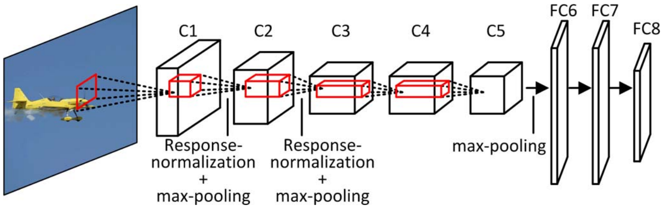

C1
C2
C3
C4
C5
FC6
FC7
FC8
Response-normalization + max-pooling
Response-normalization + max-pooling
max-pooling

Fig. 1. Architecture of AlexNet CNN. This network is composed of five successive convolutional layers (C1, C2, C3, C4, and C5) followed by three fully connected layers (FC6, FC7, and FC8).

The rest of this paper is organized as follows. Section II briefly introduces the architectures of AlexNet CNN [41], which is the building block of our RICNN. Section III describes how to learn RICNN by optimizing a new objective function in detail. Section IV presents the framework for object detection with RICNN. Section V reports comparative experimental results on a publicly available ten-class VHR geospatial object detection data set. Finally, conclusions are drawn in Section VI.

II. Architectures of AlexNet CNN

Recently, CNN-based approaches have been substantially improving upon the state-of-the-art methods in a wide range of computer vision applications [41]–[44]. Their success is largely due to the advent of AlexNet CNN [41], which is a deep convolutional network model trained on the ImageNet data set [47]. Recent works have discovered that the AlexNet CNN model can be employed as a generic and universal CNN feature extractor, and applying this feature representation to many visual recognition tasks has led to astounding performance [41]–[43]. This has motivated us to move from the hand-engineered features, such as scale-invariant feature transform [28] and HOG [32], to the era of CNN features. Therefore, in this section, we first introduce the architecture of AlexNet CNN, which is the building block of our RICNN.

The overall architecture of AlexNet CNN [41] is illustrated in Fig. 1. The AlexNet CNN model consists of eight weighted layers, including five convolutional layers C1, C2, C3, C4, and C5 and three fully connected layers FC6, FC7, and FC8. Moreover, response normalization layers follow the first and second convolutional layers. Max-pooling layers follow both response normalization layers and the fifth convolutional layer. The rectified linear units (ReLUs) are applied to the output of every convolutional and fully connected layer. The first convolutional layer has 96 kernels of size $11 \times 11 \times 3$ with a stride of 4 pixels (this is the distance between the receptive field centers of neighboring neurons in a kernel map). The second convolutional layer takes the output of the first convolutional layer as input and filters it with 256 kernels of size $5 \times 5 \times 96$. The third, fourth, and fifth convolutional layers are connected to one another without any intervening pooling or normalization layers. The third convolutional layer has 384 kernels of size $3 \times 3 \times 256$ connected to the outputs of the second convolutional layer. The fourth convolutional layer has 384 kernels of size $3 \times 3 \times 384$, and the fifth convolutional layer has 256 kernels of size $3 \times 3 \times 384$. The FC6 and FC7 fully connected layers have 4096 neurons each. The output of the last fully connected layer is fed to a 1000-way softmax that produces a distribution over the 1000 class labels. Readers can refer to [41] for more details.

III. Proposed Method

Fig. 2 illustrates the overall framework of our proposed RICNN training. It consists of two steps: data augmentation and RICNN training. The first step mainly generates a set of positive and negative training examples by using a generic object proposal detection method [48] and a simple rotating operation. In the second step, we design an RICNN model by introducing a rotation-invariant layer on the basis of the successful AlexNet CNN model [41] and replacing the AlexNet CNN's 1000-way softmax classification layer with a $(C+1)$-way softmax classification layer (for our C geospatial object classes plus one background class). The proposed RICNN model is then trained by optimizing a new objective function via imposing a regularization constraint term, which enforces the training samples before and after rotating to share the similar features to achieve rotation invariance. To facilitate training, the parameters of the first seven layers (five convolutional layers C1, C2, C3, C4, and C5 and two fully connected layers FC6 and FC7) are transferred from AlexNet CNN [41] and then domain-specifically fine-tuned to adapt to our VHR optical remote sensing image data set with a smaller learning rate, which allows fine-tuning to make progress while not clobbering the initialization. The parameters of the last two fully connected layers (FCa and FCb) are trained with a bigger learning rate, which makes the model able to jump out of the local optimum and to converge fast. We experimentally show that the proposed RICNN model can significantly outperform the traditional CNN model for the geospatial object detection task in Section V. Then, we will describe the two steps in detail.

A. Data Augmentation

Due to the limited size of the training set, performing data augmentation to artificially increase the number of training examples is necessary to avoid overfitting. Rather than only using ground truth objects, we adopt a similar scheme as in [42] and [43] to generate positive and negative training samples. To be specific, given the training image set, we first extract a number of object proposals using a generic object proposal detection method [48], which has been proven effective for generating category-independent object proposals. Then, we map each object proposal to the ground truth bounding box with which it has a maximum intersection over union (IoU) overlap and label it as a positive for the matched ground truth object class if the area overlap ratio a_o between the object proposal and the ground truth bounding box exceeds 0.5 by the following formula:

$$a_o = \frac{\text{area}(B_{\text{op}} \cap B_{\text{gt}})}{\text{area}(B_{\text{op}} \cup B_{\text{gt}})} \tag{1}$$

where $\text{area}(B_{\text{op}} \cap B_{\text{gt}})$ denotes the intersection of the object proposal and the ground truth bounding box, and $\text{area}(B_{\text{op}} \cup B_{\text{gt}})$ denotes their union. Otherwise, the proposal is considered as a negative sample. By using the above scheme, many "jittered" samples (those proposals with an overlap ratio between 0.5 and 1, but not ground truth) are included, which significantly expands the number of positive samples by about six times. Finally, we define K rotation angles $\phi = \{\phi_1, \phi_2, \ldots, \phi_K\}$ and their rotation transformations $T_\phi = \{T_{\phi_1}, T_{\phi_2}, \ldots, T_{\phi_K}\}$, with T_{ϕ_k} denoting the rotation of a sample with the angle of ϕ_k. Applying rotation transformations T_ϕ to all training samples $X = \{x_1, x_2, \ldots, x_N\}$ can yield a new set of training samples $T_\phi X = \{T_\phi x_1, T_\phi x_2, \ldots, T_\phi x_N\}$, with $T_\phi x_i = \{T_{\phi_k} x_i | k = 1, 2, \ldots, K\}$. The total training samples before and after rotating, i.e., $\mathcal{X} = \{X, T_\phi X\}$, will be used jointly to train the RICNN model.

B. Training Rotation-Invariant CNN

With more than 60 million parameters involved for learning the deep neural networks, directly training an RICNN model on our own data set is problematic, since the data set only contains hundreds of images, which is too small to train so many parameters. Fortunately, due to the wonderful generalization abilities of pretrained deep models that have been demonstrated by recent works [41]–[43], we could transfer these pretrained models to our own applications directly. In our work, we build on the successful AlexNet CNN [41] to learn our RICNN model. As shown in Fig. 2, to achieve rotation invariance, apart from replacing the AlexNet CNN's 1000-way softmax classification layer with a $(C+1)$-way softmax classification layer FCb (for our C object classes plus one background class), we introduce a new rotation-invariant layer FCa that uses the output of layer FC7 as input. Different from AlexNet CNN training, which only optimizes the multinomial logistic regression objective [41], the newly added rotation-invariant layer, together with the whole CNN network, is now trained by optimizing a new objective function via imposing a regularization constraint term to enforce the training samples before and after rotating to share the similar features, hence achieving rotation invariance.

To facilitate training and, particularly, to avoid overfitting caused by the limited size of the training set, the parameters (weights and biases) of layers C1, C2, C3, C4, C5, FC6, and FC7, denoted by $\{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_7\}$ and $\{\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_7\}$, are transferred from AlexNet CNN [41] to our target task and then domain-specifically fine-tuned with a smaller learning rate, which allows fine-tuning to make progress while not clobbering the initialization. For a training sample $x_i \in \mathcal{X}$, let $\boldsymbol{O}_7(x_i)$ be the output of layer FC7, $\boldsymbol{O}_a(x_i)$ be the output of layer FCa, $\boldsymbol{O}_b(x_i)$ be the output of softmax classification layer FCb, and $(\mathbf{W}_a, \boldsymbol{B}_a)$ and $(\mathbf{W}_b, \boldsymbol{B}_b)$ be the newly added parameters

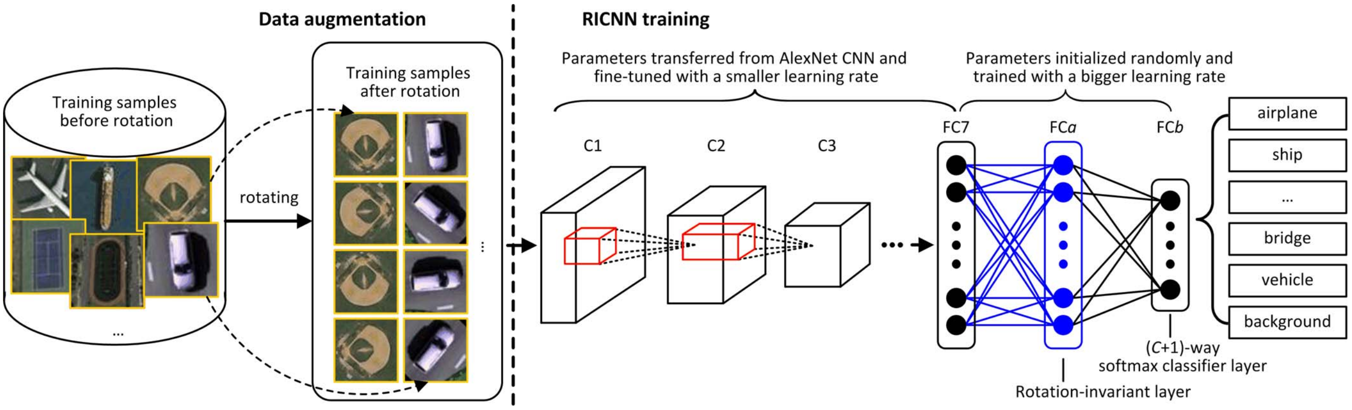

Data augmentation
RICNN training
Training samples before rotation
...
rotating
Training samples after rotation
Parameters transferred from AlexNet CNN and fine-tuned with a smaller learning rate
Parameters initialized randomly and trained with a bigger learning rate
C1
C2
C3
FC7
FCa
FCb
airplane
ship
...
bridge
vehicle
background
(C+1)-way softmax classifier layer
Rotation-invariant layer

Fig. 2. Framework of the proposed RICNN training.

of layers FCa and FCb. Thus, $\boldsymbol{O}_a(x_i)$ and $\boldsymbol{O}_b(x_i)$ can be computed by

$$\boldsymbol{O}_a(x_i) = \sigma\left(\mathbf{W}_a \boldsymbol{O}_7(x_i) + \boldsymbol{B}_a\right) \tag{2}$$

$$\boldsymbol{O}_b(x_i) = \varphi\left(\mathbf{W}_b \boldsymbol{O}_a(x_i) + \boldsymbol{B}_b\right) \tag{3}$$

where $\sigma(\boldsymbol{x}) = \max(\mathbf{0}, \boldsymbol{x})$ and $\varphi(\boldsymbol{x}) = \exp(\boldsymbol{x})/\|\exp(\boldsymbol{x})\|_1$ are the "ReLU" and "softmax" nonlinear activation functions. In all our experiments, FCa has a size of 4096, and FCb has a size equal to $(C+1)$ (C object classes plus one background class).

Given the training samples $\mathcal{X} = \{x_i \in X \cup T_\phi X\}$ and their corresponding labels $\mathcal{Y} = \{\boldsymbol{y}_{x_i} | x_i \in \mathcal{X}\}$, where $\boldsymbol{y}_{x_i}$ denotes the ground truth label vector of sample x_i with only one element being 1 and the others being 0, our objective is to train an RICNN model with the input–target pairs $(\mathcal{X}, \mathcal{Y})$. Apart from requiring that the RICNN model should minimize the misclassification error on the training data set, we also require that the RICNN model should have the rotation invariance capability for any set of training samples $\{x_i, T_\phi x_i\}$. To this end, we propose a new objective function to learn $\mathbf{W} = \{\mathbf{W}_1, \mathbf{W}_2, \ldots, \mathbf{W}_7, \mathbf{W}_a, \mathbf{W}_b\}$ and $\boldsymbol{B} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, \ldots, \boldsymbol{B}_7, \boldsymbol{B}_a, \boldsymbol{B}_b\}$ by the following formula:

$$J(\mathbf{W}, \boldsymbol{B}) = \min\left(M(\mathcal{X}, \mathcal{Y}) + \lambda_1 R(X, T_\phi X) + \frac{\lambda_2}{2}\|\mathbf{W}\|_2^2\right) \tag{4}$$

where λ_1 and λ_2 are two tradeoff parameters that control the relative importance of the three terms.

The first term $M(\mathcal{X}, \mathcal{Y})$ is the softmax classification loss function, which is defined by a $(C+1)$-class multinomial negative log-likelihood criterion with respect to input–target pairs $(\mathcal{X}, \mathcal{Y})$. It seeks to minimize the misclassification error for the given training samples and is given by

$$\begin{aligned} M(\mathcal{X}, \mathcal{Y}) &= -\frac{1}{N+NK} \sum_{x_i \in \mathcal{X}} \langle y_{x_i}, \log \boldsymbol{O}_b(x_i) \rangle \\ &= -\frac{1}{N+NK} \sum_{x_i \in \mathcal{X}} \sum_{k=1}^{C+1} y_{x_i}[k] \cdot \log\left(\boldsymbol{O}_b(x_i)[k]\right) \end{aligned} \tag{5}$$

where N is the total number of initial training samples in X, and K is the total number of rotation transformations for each $x_i \in X$.

The second term $R(X, T_\phi X)$ is a rotation invariance regularization constraint, which is imposed on the training samples before and after rotating, namely, X and $T_\phi X$, to enforce them to share the similar features. We define the regularization constraint term as

$$R(X, T_\phi X) = \frac{1}{2N} \sum_{x_i \in X} \left\| \boldsymbol{O}_a(x_i) - \overline{\boldsymbol{O}_a(T_\phi x_i)} \right\|_2^2 \tag{6}$$

where $\boldsymbol{O}_a(x_i)$ serves as the RICNN feature of the training sample x_i; $\overline{\boldsymbol{O}_a(T_\phi x_i)}$ denotes the average RICNN feature representation of rotated versions of the training sample x_i, and so, it is computed by

$$\overline{\boldsymbol{O}_a(T_\phi x_i)} = \frac{1}{K}\left(\boldsymbol{O}_a(T_{\phi_1} x_i) + \boldsymbol{O}_a(T_{\phi_2} x_i) + \cdots + \boldsymbol{O}_a(T_{\phi_K} x_i)\right). \tag{7}$$

As can be seen from (6), this term enforces the feature of each training sample to be close to the average feature representation of its rotated versions. If this term outputs a small value, the feature representation is sought to be approximately invariant to the rotation transformations.

The third term is a weight decay term that tends to decrease the magnitude of the weights of $\mathbf{W} = \{\mathbf{W}_1, \ldots, \mathbf{W}_7, \mathbf{W}_a, \mathbf{W}_b\}$ and helps prevent overfitting.

By incorporating (5) and (6) into (4), we have the following objective function:

$$J(\mathbf{W}, \boldsymbol{B}) = \min \left(\begin{array}{l} -\frac{1}{N+NK} \sum\limits_{x_i \in \mathcal{X}} \sum\limits_{k=1}^{C+1} y_{x_i}[k] \cdot \log\left(\boldsymbol{O}_b(x_i)[k]\right) \\ + \frac{\lambda_1}{2N} \sum\limits_{x_i \in X} \left\| \boldsymbol{O}_a(x_i) - \overline{\boldsymbol{O}_a(T_\phi x_i)} \right\|_2^2 + \frac{\lambda_2}{2}\left(\|\mathbf{W}\|_2^2\right) \end{array} \right). \tag{8}$$

We can easily see that the objective function defined by (8) not only minimizes the classification loss but also imposes a regularization constraint to achieve rotation invariance. In practice, we solve this optimization problem by using the stochastic gradient descent (SGD) method [49], which has been widely used in complicated stochastic optimization problems such as neural network training.

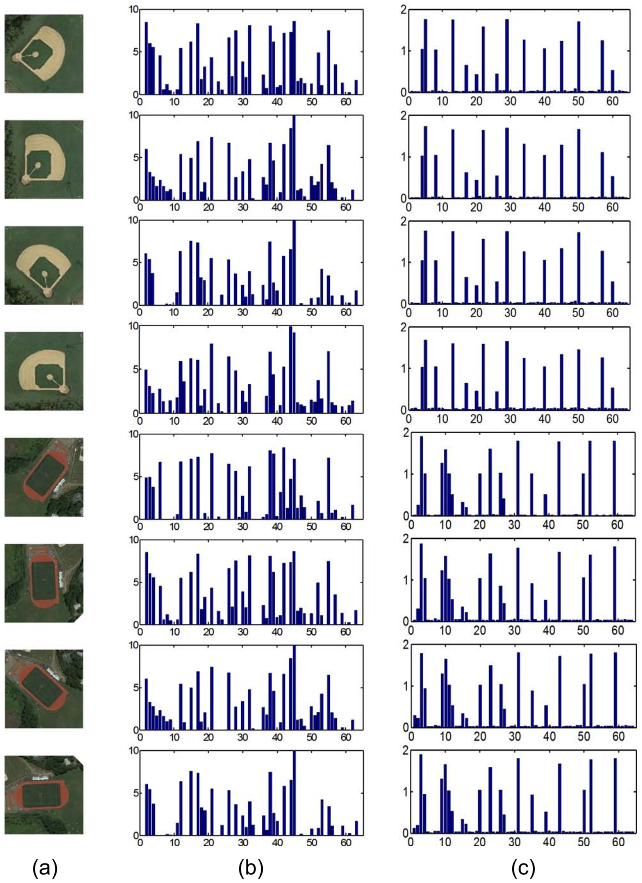


Fig. 3. (a) Eight test examples of two object classes (baseball diamond and ground track field) and their corresponding randomly sampled 64 feature values from the 4096-dimensional, (b) CNN features, and (c) RICNN features.

Fig. 3 shows eight test examples of two object classes (baseball diamond and ground track field) and their corresponding CNN features and RICNN features. To facilitate visualization, we randomly sampled 64 feature values from the 4096-diemsional CNN and RICNN features, respectively. The CNN features shown in Fig. 3(b) are extracted using the AlexNet CNN model, and the RICNN features shown in Fig. 3(c) are extracted using our trained RICNN model. As shown in Fig. 3, the CNN features of different test examples of the same object category are not rotation invariant, which makes the traditional CNN features difficult to handle the problem of object rotation variations. In contrast, the RICNN features of different test examples of the same object class are obviously rotation invariant, which demonstrates the effectiveness of the proposed RICNN training method.

IV. Object Detection With RICNN

Fig. 4 gives an overview of the proposed object detection system. It is mainly composed of two stages: object proposal detection and RICNN-based object detection. In the first stage, given an input image, to avoid exhaustive search at all positions, we adopt a class-independent and data-driven selective search strategy [48] to generate a small set of high-quality object proposals (candidate bounding boxes that may contain objects). In the second stage, we first extract RICNN features for each cropped object proposal using our trained RICNN model and then use all object category classifiers simultaneously to decide if each proposal contains the object of interest.

A. Object Proposal Detection

Most of the existing geospatial object detection methods are based on sliding-window search, in which each image is scanned at all positions of different scales, and then, a trained classifier is used to examine if each window contains a given target. However, the visual search space is huge, which makes an exhaustive search computationally expensive. Rather than sampling locations blindly using an exhaustive search, steering the sampling by a data-driven method to generate a small set of high-confidence object proposals to reduce search space is highly appealing.

As our main goal is to validate the proposed RICNN model rather than to develop an object proposal detection method, here, we adopt a state-of-the-art method named selective search [48] to generate object hypotheses. It is data driven and category independent; thus, it can find the possible locations of all object classes. Moreover, compared with other proposal detection methods, selective search is also reported in [48] to have a higher maximum average best overlap and recall but only with a comparable number of proposals. The selective search [48] procedure works briefly as follows. It first uses a graph-based image segmentation algorithm to create initial segments. Then, a greedy algorithm is used to iteratively merge segments together: First, the similarities between all neighboring segments are calculated. The two most similar segments are merged together, and new similarities are calculated between the resulting segment and its neighbors. This process repeats until there is only one segment left, which is the whole image. Each step generates a new segment and, thus, a new object proposal. Readers can refer to [48] for more details.

On our data set, we extract averagely about 500 object proposals per image. Fig. 5 shows some object proposals extracted on the test data set with the area overlap ratio a_o exceeding 0.5 calculated by (1). It is observed that although objects vary a lot in size, illumination, and occlusion in cluttered backgrounds, selective search can always extract reliable object proposals. To further demonstrate the applicability of the selective search method [48] on our object detection task, Fig. 6 reports the quantitative evaluation results measured by "Recall" for all ten object categories (the data set will be described in Section V-A). This metric of Recall is derived from the Pascal overlap criterion to measure the quality of the proposals. It measures the fraction of ground truth objects that are correctly found, where an object is considered to be found when the area overlap ratio in (1) is larger than 0.5. As shown in Fig. 6, we obtain a Recall bigger than 90% for all object classes. In Section V-C, we will investigate how the number of object proposals affects the Recall rate of each object category and the final object detection performance.

B. RICNN-Based Object Detection

1) Feature Extraction: Using the trained RICNN model, we extract a 4096-dimensional RICNN feature vector from each object proposal. Features are computed by forward propagating a mean-subtracted 227 × 227 image patch through five

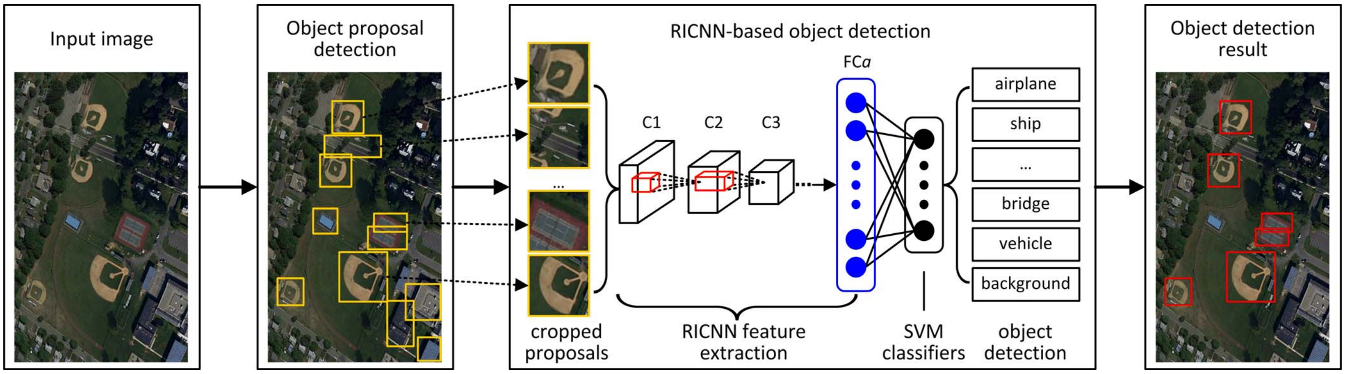


Fig. 4. Overview of the proposed object detection system. Given an input image, the system first generates a set of category-independent object proposals, then extracts features for each proposal using our trained RICNN model, and finally classifies each proposal using class-specific linear SVMs.

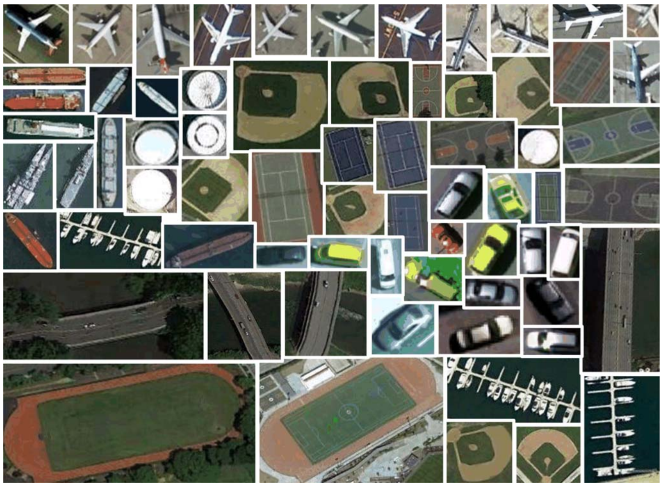


Fig. 5. Some object proposals extracted on the test data set.

convolutional layers (C1, C2, C3, C4, and C5) and three fully connected layers (FC6, FC7, and FCa), as illustrated in Fig. 2. To compute features for each object proposal, we need first to convert each proposal into a form that is compatible with the RICNN (its architecture requires inputs of a fixed 227×227 pixel size). In our work, we adopt a simple and effective transformation to do this. Briefly, regardless of the size or aspect ratio of the proposal, we crop all pixels within a tightest square bounding box that encloses it and then scale the cropped image region contained in that square to the required RICNN input size. Prior to cropping, we also consider including additional image context around the original object proposal, which is achieved by dilating the tight bounding box uniformly with a border size of 16 pixels as the work of Girshick *et al.* in [42] and [43].

2) Object Category Classifier Training: To achieve simultaneous multiclass object detection, given C geospatial object classes, we train C object category classifiers separately, where each object category classifier is a class-specific linear SVM. To train SVM classifiers for each object class, the positive samples are defined to be the ground truth bounding boxes of that class, and the negative samples are obtained by using a standard hard-negative mining technique [50] from the region proposals with $\leq$ 0.2 IoU overlap with all ground truth bounding boxes of that class. The overlap threshold, i.e., 0.2, is determined by a grid search over {0, 0.1, 0.2, 0.3, 0.4, 0.5} on our validation set. Moreover, there is only one free parameter in linear SVM used to tune the tradeoff between the amount of accepted errors and the maximization of the margin. In our work, this free parameter was optimized on our validation set, and we set it to 1 according to the experimental results. In Section V-D, we will discuss why the positives and negatives are defined differently in RICNN model training versus SVM training. We will also discuss the reasons involved in training class-specific SVMs rather than simply using the outputs of the final softmax classifier layer of the trained RICNN.

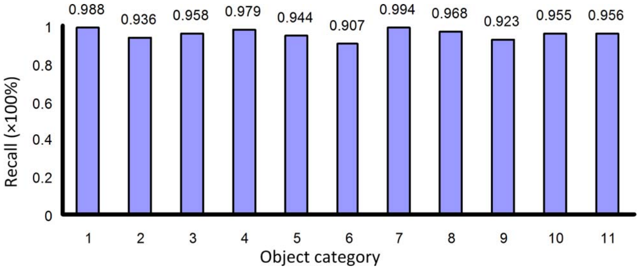


Fig. 6. Quantitative evaluation results measured by Recall for all ten object categories (1—airplane, 2—ship, 3—storage tank, 4—baseball diamond, 5—tennis court, 6—basketball court, 7—ground track field, 8—harbor, 9—bridge, 10—vehicle, and 11—average). The numbers on the bars denote the Recall values for each object category.

3) Object Detection: After generating the cropped object proposals and extracting their corresponding RICNN features, we run all trained class-specific SVM classifiers simultaneously to score each object proposal and predict its label. Then, object detection is performed by thresholding the scores with a user-defined threshold, and each detection is defined by a score, an object category label, and a bounding box derived from the object proposal. Furthermore, in practice, a number of object proposals near each object of interest are likely to be detected as the same target, resulting in multiple repeated detections for a single object. We therefore apply a nonmaximum suppression strategy as in [1], [7], [16], [22], [42], and [50] to eliminate repeated detections. Given all scored detections in an image, we greedily select the highest scoring ones while rejecting those that are at least 50% covered by a previously selected detection.

V. Experiments

A. Data Set Description

We evaluate the performance of the proposed RICNN-based object detection approach on a publicly available data set:

NWPU VHR-10 data set [22], [26].[1] This is a challenging ten-class geospatial object detection data set used for multiclass object detection. These ten classes of objects are airplane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, and vehicle. This data set contains a total of 800 VHR optical remote sensing images, where 715 color images were acquired from Google Earth with the spatial resolution ranging from 0.5 to 2 m, and 85 pansharpened color infrared images were acquired from Vaihingen data with a spatial resolution of 0.08 m.

There are two image sets in this data set: a positive image set including 650 images with each image containing at least one target to be detected and a negative image set including 150 images that do not contain any targets of the given object classes. From the positive image set, 757 airplanes, 302 ships, 655 storage tanks, 390 baseball diamonds, 524 tennis courts, 159 basketball courts, 163 ground track fields, 224 harbors, 124 bridges, and 477 vehicles were manually annotated with bounding boxes used for ground truth. The negative image set is used for semi-supervised learning-based object detection [22] and weakly supervised learning-based object detection [1], [8], and so, it is not used in this paper. In our work, the positive image set was divided into 20% for training, 20% for validation, and 60% for test, resulting in three independent subsets: a training set containing 120 images, a validation set containing 120 images, and a test set containing 410 images.

B. Evaluation Metrics

We adopt the precision–recall curve (PRC) and average precision (AP) to quantitatively evaluate the performance of an object detection system. They are two standard and widely used measures in many object detection works such as those in [1], [7], [8], [16], and [22].

1) Precision–Recall Curve: 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. Let TP, FP, and FN denote the number of true positives, the number of false positives, and the number of false negatives. The Precision and Recall metrics can be formulated as

$$\text{Precision} = \frac{TP}{(TP + FP)} \tag{9}$$

$$\text{Recall} = \frac{TP}{(TP + FN)}. \tag{10}$$

A detection is considered to be true positive if the area overlap ratio a_o between the predicted bounding box and the ground truth bounding box exceeds 0.5 by using (1). Otherwise the detection is considered as a false positive. In addition, if several detections overlap with the same ground truth bounding box, only one is considered as true positive, and others are considered as false positives.

2) Average Precision: The AP computes the average value of Precision over the interval from Recall = 0 to Recall = 1, i.e., the area under the PRC; hence, the higher the AP value, the better the performance, and *vice versa*.

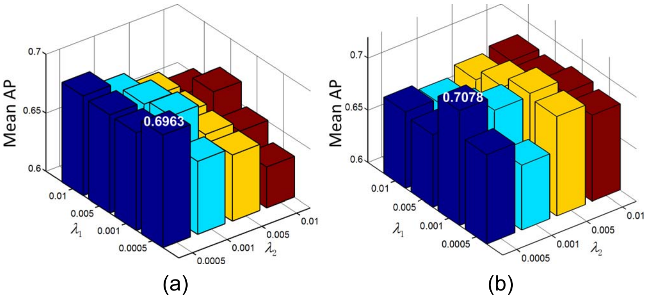


Fig. 7. Object detection results, measured in terms of mean AP over all ten object classes, under different parameter settings. (a) Learning rate $\eta = 0.005$. (b) Learning rate $\eta = 0.01$. The highest mean AP values of (a) and (b) are indicated with bold numbers on their corresponding bars.

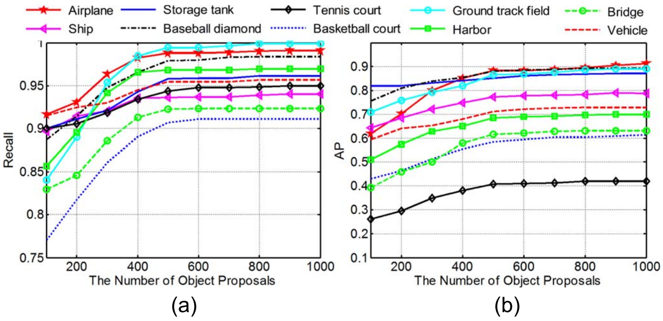


Fig. 8. (a) Tradeoff between the Recall rate and the average number of object proposals per image. (b) Tradeoff between AP and the average number of object proposals per image.

C. Implementation Details and Parameter Optimization

Based on the successful AlexNet [41] that was pretrained on the ImageNet data set [47], we train our RICNN model. To augment training data, we set $K = 35$ and $\phi = \{10°, 20°, \ldots, 350°\}$ to obtain 36× training samples. For RICNN model training, the learning rate is set to 0.01 (optimized in Fig. 7) for the last two layers and 0.0001 for the whole network fine-tuning and decreases by 0.5 every 10 000 iterations. In each SGD iteration of RICNN training, we randomly sample two positive examples over all object classes and two negative examples together with their corresponding $4 \times 35 = 140$ rotated examples to construct a minibatch of size 144.

In the training of RICNN model, the learning rate η for the last two fully connected layers and the tradeoff parameters λ_1 and λ_2 are three important parameters that can affect the performance of object detection; hence, we first investigate how object detection is affected by them by designing parameter optimization experiments on our validation set. In our work, we set $\eta = \{0.01, 0.005\}$, $\lambda_1 = \{0.01, 0.005, 0.001, 0.0005\}$, and $\lambda_2 = \{0.01, 0.005, 0.001, 0.0005\}$, respectively. Fig. 7 reports the object detection results, measured in terms of mean AP over all ten object classes, under different parameter settings. As shown in Fig. 7, these three parameters affect the object detection results moderately, and the best result is obtained with $\eta = 0.01$, $\lambda_1 = 0.001$, and $\lambda_2 = 0.0005$. Consequently, we empirically set $\eta = 0.01$, $\lambda_1 = 0.001$, and $\lambda_2 = 0.0005$ in our subsequent evaluations.

In the selective search algorithm [48], the number of object proposals is also a very important parameter, which directly affects the Recall rate of each object category and, hence,

[1] http://pan.baidu.com/s/1hqwzXeG

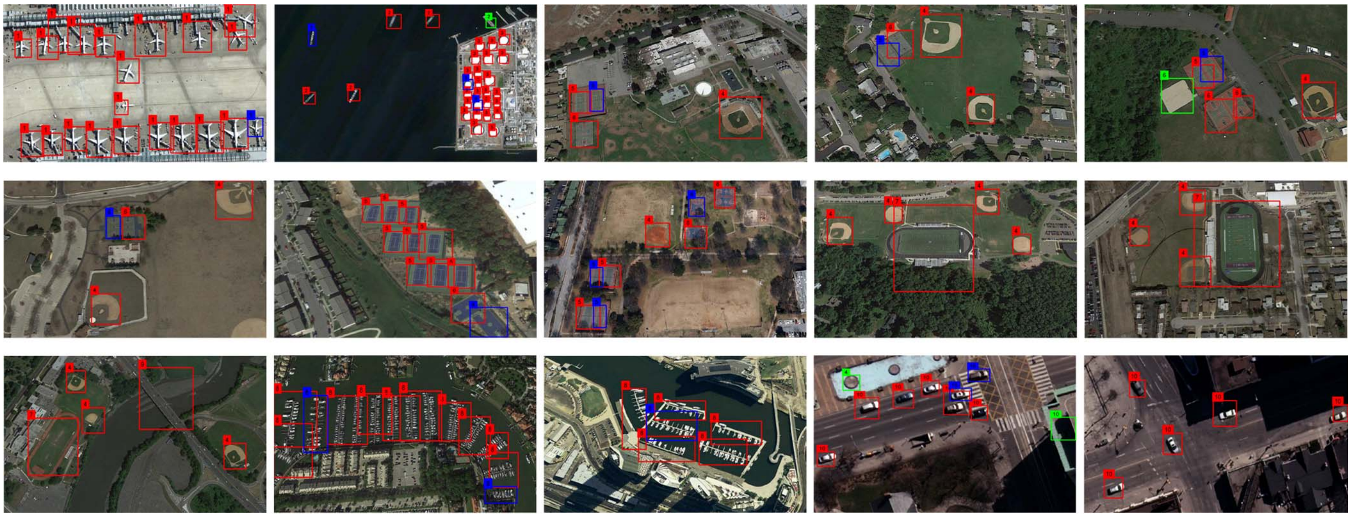


Fig. 9. Number of object detection results with the proposed approach. The true positives, false positives, and false negatives are indicated by red, green, and blue rectangles, respectively.

TABLE I
PERFORMANCE COMPARISONS OF EIGHT DIFFERENT METHODS IN TERMS OF AP VALUES. THE BOLD NUMBERS DENOTE THE HIGHEST VALUES IN EACH ROW

	BoW [12]	SSCBoW [13]	FDDL [16]	COPD [22]	Transferred CNN [41]	Newly trained CNN	RICNN without fine-tuning	RICNN with fine-tuning
Airplane	0.2496	0.5061	0.2915	0.6225	0.6614	0.7014	0.8603	**0.8835**
Ship	0.5849	0.5084	0.3764	0.6887	0.5693	0.6370	0.7602	**0.7734**
Storage tank	0.6318	0.3337	0.7700	0.6371	0.8432	0.8433	0.8504	**0.8527**
Baseball diamond	0.0903	0.4349	0.2576	0.8327	0.8163	0.8361	0.8731	**0.8812**
Tennis court	0.0472	0.0033	0.0275	0.3208	0.3499	0.3546	0.3958	**0.4083**
Basketball court	0.0322	0.1496	0.0358	0.3625	0.4592	0.4680	0.5791	**0.5845**
Ground track field	0.0777	0.1007	0.2010	0.8531	0.7998	0.8120	0.8549	**0.8673**
Harbor	0.5298	0.5833	0.2539	0.5527	0.6201	0.6228	0.6651	**0.6860**
Bridge	0.1216	0.1249	0.2154	0.1479	0.4229	0.4538	0.5848	**0.6151**
Vehicle	0.0914	0.3361	0.0447	0.4403	0.4287	0.4480	0.6798	**0.7110**
Mean AP	0.2457	0.3081	0.2474	0.5458	0.5971	0.6177	0.7103	**0.7263**

TABLE II
COMPUTATION TIME COMPARISONS OF EIGHT DIFFERENT METHODS

Methods	Average running time per image (second)
BoW [12]	5.32
SSCBoW [13]	40.32
FDDL [16]	7.17
COPD [22]	1.07
Transferred CNN [41]	5.24
Newly trained CNN	8.77
RICNN without fine-tuning	8.77
RICNN with fine-tuning	8.77

indirectly decides on the final object detection performance. Fig. 8 explores the tradeoff between the quality (measured with the Recall rate and AP for all ten object categories) and the quantity of the object proposals (measured with the average number of object proposals per image) on our test data set. Hence, we derive the following: 1) The Recall rate and AP improve rapidly and then tend to be stable with the increase of the number of object proposals. 2) In terms of Recall and AP, we have a reasonably consistent quality/quantity tradeoff. In particular, by considering both the computation cost and detection accuracy, when we extract about 500 object proposals per image, we obtain a good quantity/quality tradeoff (Recall is bigger than 90% for all object classes), which shows that the selective search method [48] is effective for finding a high-quality set of object proposals using a limited number of boxes. (3) There is obviously a gap between the Recall rate of object proposals and the Recall rate of final detections. This gap is mainly caused by misclassification, which indicates that there is still huge space for further boosting the object detection performance by designing more powerful feature representations and object detector.

D. *SVMs Versus Softmax Classifier*

To train our RICNN model, we treat all region proposals with $\geq$ 0.5 IoU overlap with a ground truth box as positives for that box's class and the rest as negatives. In contrast, to train

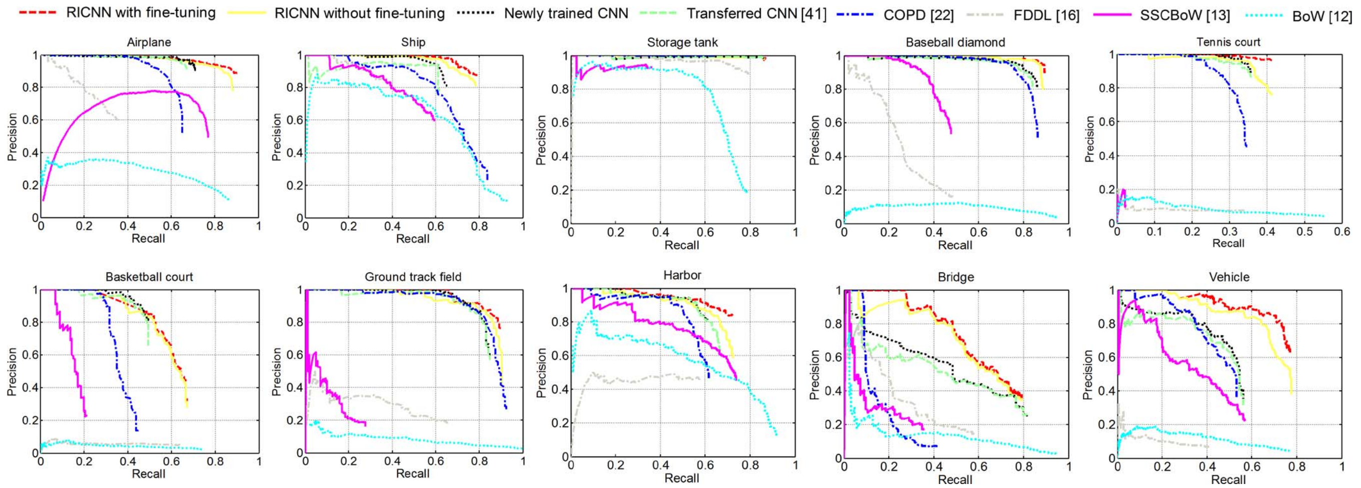


Fig. 10. PRCs of the proposed RICNN-based method and other state-of-the-art approaches for airplane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, and vehicle classes, respectively.

SVM classifiers for each object class, the positive samples are defined to be the ground truth bounding boxes of that class, and the negative samples are obtained by using a standard hard-negative mining technique [50] from the region proposals with ≤ 0.2 IoU overlap with all ground truth bounding boxes of that class. Proposals that have more than 0.2 IoU overlap but are not ground truths are ignored. This difference in how positives and negatives are defined mainly arises from the fact that the training data for the RICNN model are very limited. By using our data augmentation scheme (excluding rotation transformation), many "jittered" samples that have more than 0.5 IoU overlap but are not ground truths are included, which significantly expands the number of positive samples by approximately six times. This augmented training set is needed when training the entire RICNN network to avoid overfitting.

In the implementation of object detection, it would be more intuitive to directly adopt the last layer of the trained RICNN model, which is a softmax regression classifier, as the object detector. However, we observed that using these jittered samples that have more than 0.5 IoU overlap for object detector (classifier) training is suboptimal because the network is not trained for precise localization. This leads to the issue why we need to train class-specific SVMs rather than directly use the outputs of the final softmax classifier layer of the trained RICNN. We tried this and found that object detection performance measured in mean AP degenerated from 72.63% to 69.68% by using SVM classifiers and softmax classifier, respectively. This performance degeneration most likely arises from the fact that the definition of positive samples used for RICNN training does not emphasize precise localization, and the softmax classifier was trained on randomly sampled negative samples rather than on the set of hard negatives used for SVM training.

E. Experimental Results and Comparisons

Using the trained RICNN model and class-specific object category classifiers, we performed ten-class object detection on our test data set that contains 410 VHR images. Fig. 9 shows a number of object detection results with the proposed approach, in which the true positives, false positives, and false negatives are indicated by red, green, and blue rectangles, respectively. The ten numbers of one to ten on the rectangles denote the object categories of airplane, ship, storage tank, baseball diamond, tennis court, basketball court, ground track field, harbor, bridge, and vehicle, respectively. As shown in Fig. 9, despite the large variations in the orientations and sizes of objects, the proposed approach has successfully detected and located most of the objects.

To quantitatively evaluate the proposed RICNN model, we compared it with four traditional state-of-the-art methods, in which the BoW feature [12], the spatial sparse coding BoW (SSCBoW) feature [13], the sparse-coding-based feature [16], and the collection of part detectors (COPD) [22] are employed, respectively. Specifically, the method in [12] is based on the BoW model, in which each image region is represented as a histogram of visual words generated by the k-means algorithm. The method in [13] is based on the SSCBoW model. Similar to the BoW model, the SSCBoW model also represents each image region as a histogram of visual words, but it uses sparse coding to replace the k-means algorithm for visual word encoding. The method in [16] is based on Fisher discrimination dictionary learning (FDDL), in which a sparse-representation-based classification strategy is adopted to perform multiclass object detection, where each image window is described by a few representative atoms of a learned dictionary in a low-dimensional manifold. The method in [22] is based on the COPD. For our ten-class object detection task, the COPD is composed of 45 seed-based part detectors trained in HOG feature space, where each part detector is a linear SVM classifier corresponding to a particular viewpoint of an object class; hence, the collection of them provides an approximate solution for rotation-invariant object detection.

In addition, to further validate the effectiveness and superiority of the proposed RICNN model, we compared our RICNN model with 1) a transferred CNN model from AlexNet [41], which is employed as a universal CNN feature extractor and has shown great success for PASCAL Visual Object Classes object detection [42], [43]; 2) a newly trained CNN model (with the same architecture as our RICNN model but without considering rotation-invariant information) in which the newly added fully connected layer was first trained by using the traditional CNN objective function and then the whole network was domain-specifically fine-tuned; and 3) a newly trained RICNN model with only the rotation-invariant layer and the softmax layer being trained (without fine-tuning the other layers). For a fair comparison, 1) we adopted the same training data set and test data set for the proposed method and other comparison methods. In particular, the augmented data were used for all methods including CNN model training and RICNN model training. 2) We uniformly employed the same selective search method [48] for all comparison methods to generate object proposals. 3) Our RICNN model (without fine-tuning and with fine-tuning), the transferred CNN model, and the newly trained CNN model are all based on the AlexNet [41] model and were trained with exactly the same parameters (if applicable).

Tables I and II and Fig. 10 show the quantitative comparison results of eight different methods, measured by AP values, average running time per image, and PRCs, respectively. All of these methods were evaluated on a PC with two 2.8-GHz 6-core CPUs and 32-GB memory. Moreover, the transferred CNN, the newly trained CNN, and our RICNN were accelerated by a GTX Titan X GPU. As can be seen from them, 1) the proposed RICNN-model-based methods (without fine-tuning and with fine-tuning) outperform all other comparison approaches for all ten object classes in terms of AP. Specifically, our RICNN with fine-tuning obtained 48.06%, 41.82%, 47.89%, 18.05%, 12.92%, and 10.86% performance gains, in terms of mean AP over all ten object categories, compared with the BoW-based method [12], the SSCBoW-based method [13], the FDDL-based method [16], the COPD-based method [22], the transferred-CNN-model-based method [41], and the newly-trained-CNN-model-based method, respectively. This demonstrates the high superiority of the proposed method compared with the existing state-of-the-art methods. 2) By fine-tuning the RICNN network, the performance measured in mean AP was further boosted by 1.6% (72.63% versus 71.03%). 3) With the same computation cost, our RICNN-model-based methods (without fine-tuning and with fine-tuning) improve the third-best newly-trained-CNN-model-based method significantly for all object classes except for storage tank category due to its rotation-invariant circular shape. This adequately shows the effectiveness of the proposed RICNN model learning method. However, although our method has achieved the best performance, the detection accuracy for the object categories of tennis court and basketball court is still low. This is mainly because when we perform pooling operation to extract CNN features (as illustrated in Fig. 1), some critical and discriminative properties of these object categories (e.g., the straight lines and arc in tennis courts and basketball courts) are reduced and even removed. This has significantly degenerated the detection accuracy for those object classes characterized by line properties. In our future work, we will consider adopting heterogeneous visual feature integration to further improve the object detection performance.

VI. Conclusion

In this paper, to adapt the traditional CNN model to object detection in optical remote sensing images, we have proposed a novel and effective approach to learn an RICNN model by optimizing a new objective function, which enforces the training samples before and after rotating to share the similar features to achieve rotation invariance. The quantitative comparison results on a publicly available ten-class VHR object detection data set have demonstrated huge performance gain of the proposed method compared with state-of-the-art approaches. However, as we know, our newly added rotation-invariant layer will introduce additional computational cost compared with the original CNN models; hence, in our future work, we will focus on learning RICNN model by embedding the rotation-invariant regularizer into the objective function of the CNN model without introducing additional layers to improve the computational efficiency.

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] J. Leitloff, S. Hinz, and U. Stilla, “Vehicle detection in very high resolution satellite images of city areas,” *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2795–2806, Jul. 2010.

[3] S. Tuermer, F. Kurz, P. Reinartz, and U. Stilla, “Airborne vehicle detection in dense urban areas using HoG features and disparity maps,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 6, no. 6, pp. 2327–2337, Dec. 2013.

[4] H. Grabner, T. T. Nguyen, B. Gruber, and H. Bischof, “On-line boosting-based car detection from aerial images,” *ISPRS J. Photogramm. Remote Sens.*, vol. 63, no. 3, pp. 382–396, May 2008.

[5] Ö. Aytekin, U. Zöngür, and U. Halici, “Texture-based airport runway detection,” *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 3, pp. 471–475, May 2013.

[6] P. Zhong and R. Wang, “A multiple conditional random fields ensemble model for urban area detection in remote sensing optical images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 3978–3988, Dec. 2007.

[7] G. Cheng *et al.*, “Object detection in remote sensing imagery using a discriminatively trained mixture model,” *ISPRS J. Photogramm. Remote Sens.*, vol. 85, pp. 32–43, Nov. 2013.

[8] D. Zhang, J. Han, G. Cheng, Z. Liu, S. Bu, and L. Guo, “Weakly supervised learning for target detection in remote sensing images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 4, pp. 701–705, Apr. 2015.

[9] Y. Zhang, L. Zhang, B. Du, and S. Wang, “A nonlinear sparse representation-based binary hypothesis model for hyperspectral target detection,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2513–2522, Jun. 2015.

[10] Y. Zhang, B. Du, and L. Zhang, “A sparse representation-based binary hypothesis model for target detection in hyperspectral images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1346–1354, Mar. 2015.

[11] N. Yokoya and A. Iwasaki, “Object detection based on sparse representation and Hough voting for optical remote sensing imagery,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 2053–2062, May 2015.

[12] S. Xu, T. Fang, D. Li, and S. Wang, “Object classification of aerial images with bag-of-visual words,” *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 2, pp. 366–370, Apr. 2010.

[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] Y. Chen, N. M. Nasrabadi, and T. D. Tran, “Sparse representation for target detection in hyperspectral imagery,” *IEEE J. Sel. Top. Signal Process.*, vol. 5, no. 3, pp. 629–640, Jun. 2011.

[15] L. Zhang, L. Zhang, D. Tao, and X. Huang, “Sparse transfer manifold embedding for hyperspectral target detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 2, pp. 1030–1043, Feb. 2014.

[16] 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. Photogramm. Remote Sens.*, vol. 89, pp. 37–48, 2014.

[17] 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.

[18] L. Eikvil, L. Aurdal, and H. Koren, "Classification-based vehicle detection in high-resolution satellite images," *ISPRS J. Photogramm. Remote Sens.*, vol. 64, no. 1, pp. 65–72, 2009.

[19] X. Yao, J. Han, L. Guo, S. Bu, and Z. Liu, "A coarse-to-fine model for airport detection from remote sensing images using target-oriented visual saliency and CRF," *Neurocomputing*, vol. 164, pp. 162–172, 2015.

[20] X. Huang and L. Zhang, "Road centreline extraction from high-resolution imagery based on multiscale structural features and support vector machines," *Int. J. Remote Sens.*, vol. 30, no. 8, pp. 1977–1987, 2009.

[21] S. Das, T. Mirnalinee, and K. Varghese, "Use of salient features for the design of a multistage framework to extract roads from high-resolution multispectral satellite images," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3906–3931, Oct. 2011.

[22] 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, 2014.

[23] F. Bi, B. Zhu, L. Gao, and M. Bian, "A visual search inspired computational model for ship detection in optical satellite images," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 4, pp. 749–753, Jul. 2012.

[24] C. Zhu, H. Zhou, R. Wang, and J. Guo, "A novel hierarchical method of ship detection from spaceborne optical image based on shape and texture features," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 9, pp. 3446–3456, Sep. 2010.

[25] P. Zhou, G. Cheng, Z. Liu, S. Bu, and X. Hu, "Weakly supervised target detection in remote sensing images based on transferred deep features and negative bootstrapping," *Multidimension. Syst. Signal Process.*, vol. 27, no. 4, pp. 925–944, 2016.

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

[27] F. F. Li and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 524–531.

[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[29] G.-S. Xia, J. Delon, and Y. Gousseau, "Accurate junction detection and characterization in natural images," *Int. J. Comput. Vis.*, vol. 106, no. 1, pp. 31–56, 2014.

[30] F. Hu, G.-S. Xia, Z. Wang, X. Huang, and L. Zhang, "Unsupervised feature learning via spectral clustering of patches for remotely sensed scene classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 5, pp. 2015–2030, May 2015.

[31] G.-S. Xia, Z. Wang, C. Xiong, and L. Zhang, "Accurate annotation of remote sensing images via active spectral clustering with little expert knowledge," *Remote Sens.*, vol. 7, no. 11, pp. 15 014–15 045, 2015.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, 2005, pp. 886–893.

[33] C. Yao and G. Cheng, "Approximative Bayes optimality linear discriminant analysis for Chinese handwriting character recognition," *Neurocomputing*, vol. 207, pp. 346–353, 2016.

[34] 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.

[35] G. Cheng, J. Han, L. Guo, and T. Liu, "Learning coarse-to-fine sparselets for efficient object detection and scene classification," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 1173–1181.

[36] G. Cheng, P. Zhou, J. Han, L. Guo, and J. Han, "Auto-encoder-based shared mid-level visual dictionary learning for scene classification using very high resolution remote sensing images," *IET Comput. Vis.*, vol. 9, pp. 639–647, 2015.

[37] Y. Zhong, R. Feng, and L. Zhang, "Non-local sparse unmixing for hyperspectral remote sensing imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 1889–1909, Jun. 2014.

[38] B. Song, P. Li, J. Li, and A. Plaza, "One-class classification of remote sensing images using kernel sparse representation," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 4, pp. 1613–1623, Apr. 2016.

[39] G.-S. Xia, J. Delon, and Y. Gousseau, "Shape-based invariant texture indexing," *Int. J. Comput. Vis.*, vol. 88, no. 3, pp. 382–403, 2010.

[40] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Conf. Adv. Neural Inform. Process. Syst.*, 2012, pp. 1–9.

[42] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1–37.

[43] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[44] 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. 14 680–14 707, 2015.

[45] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, "Background prior-based salient object detection via deep reconstruction residual," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 25, no. 8, pp. 1309–1321, Aug. 2015.

[46] J. Han, D. Zhang, S. Wen, L. Guo, T. Liu, and X. Li, "Two-stage learning to predict human eye fixations via SDAEs," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 487–498, Feb. 2016.

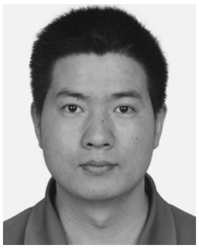
[47] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Int. Conf. Comput. Vision Pattern Recognit.*, 2009, pp. 248–255.

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

[49] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.

[50] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.

[51] X. Yao, J. Han, G. Cheng, X. Qian, and L. Guo, "Semantic annotation of high-resolution satellite images via weakly supervised learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 6, pp. 3660–3671, Jun. 2016.



Gong Cheng received the B.S. degree from Xidian University, Xi'an, China, in 2007 and the M.S. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, in 2010 and 2013, respectively.

He is currently an Associate Professor with Northwestern Polytechnical University. His main research interests include computer vision and remote sensing image analysis.

Peicheng Zhou received the B.S. degree from Xi'an University of Technology, Xi'an, China, in 2011 and the M.S. degree from Northwestern Polytechnical University, Xi'an, in 2014, where he is currently working toward the Ph.D. degree.

His research interests include computer vision and pattern recognition.

Junwei Han received the B.S. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an, China, in 1999 and 2003, respectively.

He is currently a Professor with Northwestern Polytechnical University. His research interests include computer vision and multimedia processing.