This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the version available on IEEE Xplore.

RIFD-CNN: Rotation-Invariant and Fisher Discriminative Convolutional Neural Networks for Object Detection

Gong Cheng, Peicheng Zhou, Junwei Han[*]
School of Automation, Northwestern Polytechnical University, Xi'an, China
{gcheng,jhan}@nwpu.edu.cn, zpc19881119@gmail.com

Abstract

Thanks to the powerful feature representations obtained through deep convolutional neural network (CNN), the performance of object detection has recently been substantially boosted. Despite the remarkable success, the problems of object rotation, within-class variability, and between-class similarity remain several major challenges. To address these problems, this paper proposes a novel and effective method to learn a rotation-invariant and Fisher discriminative CNN (RIFD-CNN) model. This is achieved by introducing and learning a rotation-invariant layer and a Fisher discriminative layer, respectively, on the basis of the existing high-capacity CNN architectures. Specifically, the rotation-invariant layer is trained by imposing an explicit regularization constraint on the objective function that enforces invariance on the CNN features before and after rotating. The Fisher discriminative layer is trained by imposing the Fisher discrimination criterion on the CNN features so that they have small within-class scatter but large between-class separation. In the experiments, we comprehensively evaluate the proposed method for object detection task on a public available aerial image dataset and the PASCAL VOC 2007 dataset. State-of-the-art results are achieved compared with the existing baseline methods.

1. Introduction

Object detection is one of the most fundamental yet challenging problems in computer vision community. Since the groundbreaking success of deep convolutional neural networks (CNN) [1] in image classification task [2] on the ImageNet large scale visual recognition challenge (ILSVRC) [3, 4], CNN-based object detection methods have recently attracted a great deal of research interest and have achieved state-of-the-art performance [5-27].

Among various CNN-based methods for object detection, one of the most notable work is made by Girshick *et al.* [5] with the framework of region-CNN (R-CNN), which is actually a chain of conceptually simple steps: generating candidate object proposals, classifying them as foreground or background, and post-processing them to improve their

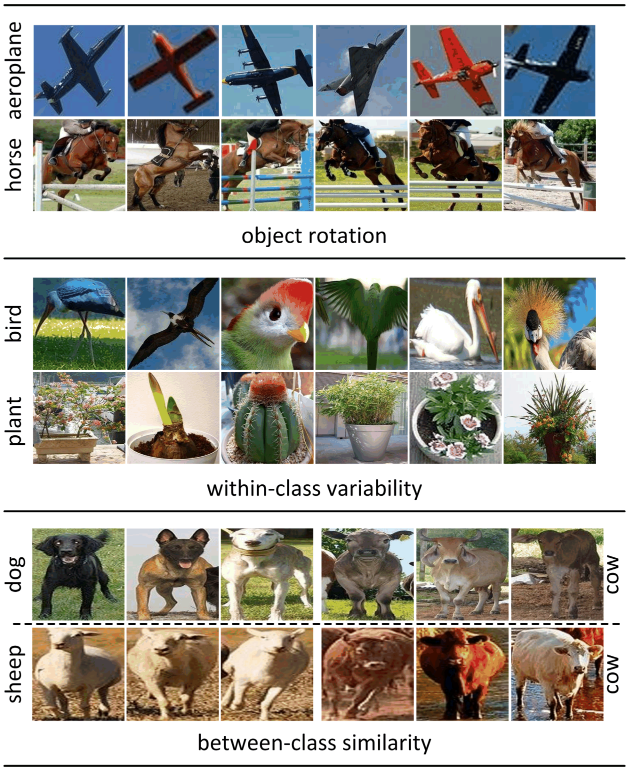

Figure 1. While the CNN features have shown impressive success for object detection, the problems of object rotation, within-class variability, and between-class similarity still remain several major challenges. Here we show some example patches for each challenge obtained from PASCAL VOC 2007 dataset [28]. All example patches are warped into a fixed 224×224 pixel size. In this situation, how to learn a more powerful feature representation that is rotation insensitive and meanwhile has small within-class scatter and big between-class separation is highly desirable. The proposed rotation-invariant and Fisher discriminative CNN model provides a possible solution to address these problems.

[*] Corresponding author.

fit to objects. Briefly, R-CNN framework proceeds as follows. First it extracts a few hundreds or thousands candidate object proposals which probably contain an object via the selective search algorithm [29] to reduce the computational cost. Then, R-CNN uses AlexNet model [2] to extract CNN features from object proposals and classifies them as objects or non-objects by using class-specific linear support vector machines (SVMs), where the AlexNet CNN model, with more than 60 million parameters, was first pre-trained on an auxiliary task of image classification in the ImageNet ILSVRC challenge [3] and then transferred and fine-tuned on a small set of images annotated for the detection task. Finally, the candidate object proposals are refitted to detected objects by using a bounding box regressor [30] to correct miss-localizations. This simple pipeline has achieved state-of-the-art detection performance on standard detection benchmarks (e.g., PASCAL VOC [28]) with a large margin over all the previously published methods, which are mostly based on deformable part model (DPM) [30].

The success of R-CNN method [5] is largely attributed to the ability of CNN model to extract more richer high-level object representation features as opposed to hand-engineered low-level features such as SIFT [31] and HOG [32]. However, while the CNN features have shown impressive success for object detection tasks, they are still difficult to effectively deal with the challenges (as illustrated in Figure 1) of object rotation, within-class variability, and between-class similarity, which are some important sources of detection error. In this situation, how to learn a more powerful feature representation that is rotation insensitive and meanwhile has small within-class scatter and big between-class separation is highly desirable. To address these problems and to further improve the state-of-the-arts, in this paper we propose a novel and effective method to learn a rotation-invariant and Fisher discriminative CNN (RIFD-CNN) model.

Our main contributions are summarized as follows: First, we build on the existing high-capacity CNN architectures [2, 9] to train a rotation-invariant CNN (RI-CNN) model by adding and learning a new rotation-invariant layer. This newly added rotation-invariant layer is trained through incorporating a regularization constraint term on the objective function of our RI-CNN model, which enforces the training samples before and after rotating to share the similar feature representations and hence achieving rotation-invariance. Evaluations on a public aerial image dataset [33] and comparisons with state-of-the-art methods demonstrate the effectiveness of the proposed RI-CNN model. Second, we propose a new method to train a Fisher discriminative CNN (FD-CNN) model by introducing and learning a Fisher discriminative layer. The Fisher discriminative layer is trained by imposing the Fisher discrimination criterion on the CNN features so that they have small within-class scatter but large between-class separation. Third, our RI-CNN model and FD-CNN model are complementary. By combining them together, we obtain a more powerful RIFD-CNN model. We have confirmed through comprehensive experiments that the proposed RIFD-CNN model can significantly improve the baseline methods on PASCAL VOC dataset [28].

2. Related Work

Object detection has been actively studied for the last few decades. The DPM [30] and its variants [20, 22, 23, 34-39] have been the leading methods for object detection tasks for years owing to the carefully crafted features like HOG [32]. In recent years, thanks to the availability of large scale training data, such as ImageNet [3], and the raise of high-performance computing systems, such as GPUs, various CNN-based methods [5-27] have been substantially improving upon the performance of object detection. Among them, the most related works and therefore also the baseline methods in our experiments are R-CNN method [5] and its improvement method [17].

The introduction of the R-CNN framework [5] opens the door to extract rich features through deep CNN models to improve object detection performance. In the work of [5], AlexNet CNN [2] was used to extract a set of deep features from category-independent region proposals provided by selective search [29] and then class-specific linear SVMs were adopted to classify them. By adopting the R-CNN framework [5] with a deeper 16-layers VGGNet CNN model [9], the performance was further boosted.

Some variants focusing on different aspects are also developed based on the successful R-CNN framework [5]. For instance, Zhang *et al.* [17] addressed the inaccurate localization problem by using a search algorithm based on Bayesian optimization that sequentially proposes candidate regions for an object bounding box and training the CNN with a structured loss that penalizes the localization inaccuracy explicitly. Zhu *et al.* [21] proposed an approach to improve the accuracy of object detection by exploiting a small number of accurate object segment proposals. They framed the problem as inference in a Markov random field, in which each detection hypothesis scores object appearance as well as contextual information using CNNs. Our method is also built upon the remarkable R-CNN framework [5] with the existing CNN architecture such as AlexNet [2] and VGGNet [9]. However, different from previous work, this paper mainly focuses on enriching the power of the CNN feature representations via imposing rotation invariance and fisher discriminative criterion on the objective function of our new RIFD-CNN model. Consequently, our work is also partly related with the ideas of learning invariant features such as [40-46] and the approaches incorporating discriminative terms into model training such as [47, 48]. In addition, other related works will be cited throughout the paper.

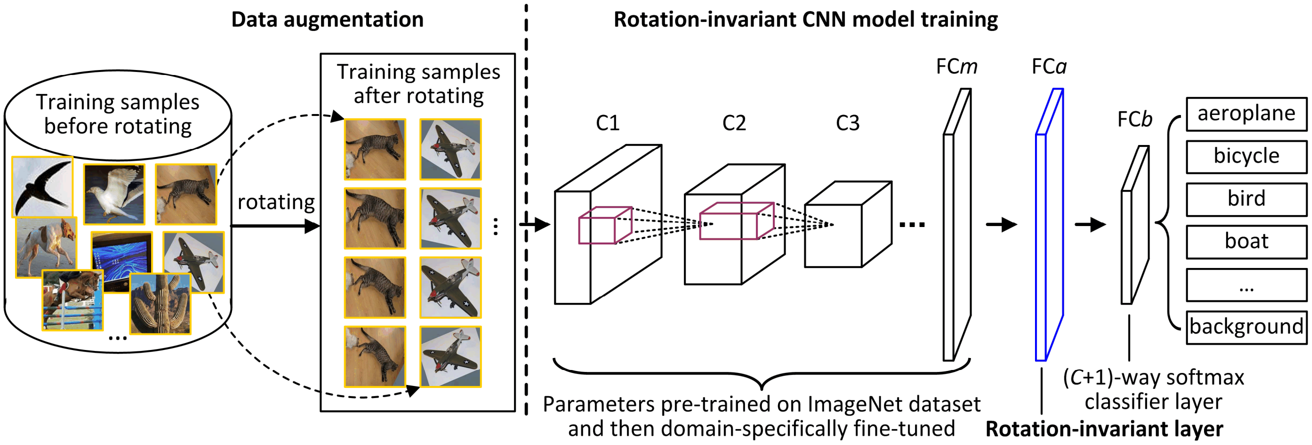
Data augmentation
Rotation-invariant CNN model training
Training samples before rotating
rotating
Training samples after rotating
C1
C2
C3
FC*m*
FC*a*
FC*b*
aeroplane
bicycle
bird
boat
...
background
Parameters pre-trained on ImageNet dataset and then domain-specifically fine-tuned
(*C*+1)-way softmax classifier layer
Rotation-invariant layer

Figure 2. The framework of the proposed RI-CNN training. It consists of two steps: data augmentation and model training. The first step mainly generates a set of augmented training samples by using a simple rotating operation. In the second step, we build on the existing high-capacity CNN architectures to train our rotation-invariant CNN model by adding and learning a new rotation-invariant layer.

3. Proposed method

The goal of our method is to learn a rotation-invariant and Fisher discriminative CNN model in order to advance the performance of object detection. This is achieved by introducing and learning a rotation-invariant layer (Figure 2) and a Fisher discriminative layer (Figure 3), respectively, on the basis of the existing high-capacity CNN architectures. To be specific, the rotation-invariant layer is trained by incorporating a regularization constraint term on the objective function of the RI-CNN model, which explicitly enforces the feature representations of the training samples before and after rotating to be mapped close to each other, and hence achieving rotation-invariance. The Fisher discriminative layer is trained by imposing the Fisher discrimination criterion on the CNN features so that they have small within-class scatter but large between-class separation. In the remainder of this section we first describe how to learn rotation-invariant CNN model and next detail the training of Fisher discriminative CNN model.

3.1. Learning rotation-invariant CNN model

The framework of the proposed rotation-invariant CNN (RI-CNN) model training is illustrated in Figure 2. It consists of two steps: data augmentation and model training. The first step mainly generates a set of positive and negative training samples by using a generic object proposal detection method [29] and a simple rotating operation. In the second step, we build on the existing popular CNN architectures, such as AlexNet [2] and VGGNet [9], to train our rotation-invariant CNN model by adding and learning a new rotation-invariant layer.

Data augmentation. Given a set of initial training samples $X=\{X^{+},X^{-}\}$, we generate a set of new training samples $\mathcal{X}_{\mathrm{RI}}=\{X,T_{\phi}X\}$ by rotating transformations, where X^{+} denotes the initial positive examples, X^{-} denotes the initial negative examples, and $T_{\phi}=\{T_{\phi_1},T_{\phi_2},\cdots,T_{\phi_K}\}$ is a family of K rotation transformations with T_{ϕ_k} denoting the rotation operation of a training sample with the angle of $\phi_k\in\phi$, $k=1,\cdots,K$. In our implementation, we treat all region proposals with ≥0.5 intersection over union (IoU) overlap with a ground-truth box as initial positives for that box's class and the rest as initial negatives.

Model training. As shown in Figure 2, in order to achieve rotation-invariance, we add a new rotation invariant fully-connected layer FC*a* that uses the output of layer FC*m* (*m* is the number of network layers except for classifier layer, e.g., in AlexNet [2] *m*=7 and in VGGNet [9] *m*=15) as input. Different from the training of traditional CNN models that only optimizes the multinomial logistic regression objective, our RI-CNN model 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. The pseudo-code of RI-CNN training is given in Algorithm 1.

To avoid over-fitting and to reduce the training cost, the parameters (weights and biases) of layers C1, C2, … , and FC*m*, denoted by $\{\mathbf{W}_1,\mathbf{W}_2,\cdots,\mathbf{W}_m\}$ and $\{\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_m\}$, are pre-trained on ImageNet dataset [3], domain specifically fine-tuned to adapt to the detection task (e.g., PASCAL VOC), and then transferred to our RI-CNN model. For a training sample $x_i\in\mathcal{X}_{\mathrm{RI}}$, let $\boldsymbol{O}_m(x_i)$ be the output of layer FC*m*, $\boldsymbol{O}_a(x_i)$ be the output of layer FC*a*, $\boldsymbol{O}_b(x_i)$ be the output of softmax classifier layer FC*b*, and $(\mathbf{W}_a,\boldsymbol{B}_a)$ and $(\mathbf{W}_b,\boldsymbol{B}_b)$ be the new parameters of layers FC*a* and FC*b*. Thus, $\boldsymbol{O}_a(x_i)$ and $\boldsymbol{O}_b(x_i)$ can be computed by

$$\boldsymbol{O}_a(x_i)=\kappa\left(\mathbf{W}_a\boldsymbol{O}_m(x_i)+\boldsymbol{B}_a\right) \tag{1}$$

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

where $\kappa(\boldsymbol{x})=\max(\mathbf{0},\boldsymbol{x})$ and $\varphi(\boldsymbol{x})=\exp(\boldsymbol{x})/\|\exp(\boldsymbol{x})\|_1$ are the ReLU and softmax non-linear activation functions. In all our experiments, FC*a* has a size of 4096, and FC*b* has a size equal to $(C+1)$ (C object classes plus background).

Given the training samples $\mathcal{X}_{\mathrm{RI}}=\{x_i \mid x_i\in X\cup T_\phi X\}$ and their corresponding labels $\mathcal{Y}_{\mathrm{RI}}=\{\boldsymbol{y}_{x_i}\mid x_i\in\mathcal{X}_{\mathrm{RI}}\}$, 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 a RI-CNN model with the input-target pairs $(\mathcal{X}_{\mathrm{RI}},\mathcal{Y}_{\mathrm{RI}})$. Apart from requiring that RI-CNN model should minimize the classification error on the training dataset, we also require that RI-CNN 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 the parameters $\mathbf{W}_{\mathrm{RI}}=\{\mathbf{W}_1,\mathbf{W}_2,\cdots,\mathbf{W}_m,\mathbf{W}_a,\mathbf{W}_b\}$ and $\boldsymbol{B}_{\mathrm{RI}}=\{\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_m,\boldsymbol{B}_a,\boldsymbol{B}_b\}$ by the following formula

$$J_{\mathrm{RI}}\left(\mathbf{W}_{\mathrm{RI}},\boldsymbol{B}_{\mathrm{RI}}\right)= \min\left(M\left(\mathcal{X}_{\mathrm{RI}},\mathcal{Y}_{\mathrm{RI}}\right)+\lambda_1 R\left(X,T_\phi X\right)+\frac{\lambda_2}{2}\left\|\mathbf{W}_{\mathrm{RI}}\right\|_2^2\right) \tag{3}$$

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

The first term $M(\mathcal{X}_{\mathrm{RI}},\mathcal{Y}_{\mathrm{RI}})$ in Eq. (3) is the softmax classification loss function, which is defined by a $(C+1)$-class multinomial negative log-likelihood criterion. It seeks to minimize the misclassification error for the given training samples and is computed by

$$M\left(\mathcal{X}_{\mathrm{RI}},\mathcal{Y}_{\mathrm{RI}}\right)=-\frac{1}{N\left(K+1\right)}\sum_{x_i\in\mathcal{X}_{\mathrm{RI}}}\left\langle\boldsymbol{y}_{x_i},\log\boldsymbol{O}_b\left(x_i\right)\right\rangle \tag{4}$$

where $\langle\boldsymbol{a},\boldsymbol{b}\rangle$ is the inner-product of $\boldsymbol{a}$ and $\boldsymbol{b}$, 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 $R(X,T_\phi X)$ in Eq. (3) 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

$$R\left(X,T_\phi X\right)=\frac{1}{2N}\sum_{x_i\in X}\left\|\boldsymbol{O}_a\left(x_i\right)-\overline{\boldsymbol{O}_a\left(T_\phi x_i\right)}\right\|_2^2 \tag{5}$$

where $\boldsymbol{O}_a(x_i)$ serves as the RI-CNN feature of the training sample x_i; $\overline{\boldsymbol{O}_a(T_\phi x_i)}$ denotes the average RI-CNN feature representation of rotated versions of the training sample x_i and so it is formulated as

$$\overline{\boldsymbol{O}_a\left(T_\phi x_i\right)}=\frac{1}{K}\sum\nolimits_{j=1}^{K}\boldsymbol{O}_a\left(T_{\phi_j}x_i\right) \tag{6}$$

Algorithm 1 Learning RI-CNN model

Input: a set of initial training samples $X=\{X^+,X^-\}$ and their corresponding ground-truth labels and a family of K rotation transformations $T_\phi=\{T_{\phi_1},T_{\phi_2},\cdots,T_{\phi_K}\}$ with T_{ϕ_k} denoting the rotation operation of a sample with the angle of ϕ_k

Output: the parameters of our RI-CNN model, denoted by $\mathbf{W}_{\mathrm{RI}}=\{\mathbf{W}_1,\mathbf{W}_2,\cdots,\mathbf{W}_m,\mathbf{W}_a,\mathbf{W}_b\}$ and $\boldsymbol{B}_{\mathrm{RI}}=\{\boldsymbol{B}_1,\boldsymbol{B}_2,\cdots,\boldsymbol{B}_m,\boldsymbol{B}_a,\boldsymbol{B}_b\}$

```
begin
Obtain the augmented input-target pairs $(\mathcal{X}_{\mathrm{RI}},\mathcal{Y}_{\mathrm{RI}})$
Initialize $(\mathbf{W}_a,\boldsymbol{B}_a)$ and $(\mathbf{W}_b,\boldsymbol{B}_b)$ randomly
while stopping criterion has not been met do
    compute classification error using Eq. (4)
    compute rotation-invariance constraint term using Eq. (5)
    compute objective function $J_{\mathrm{RI}}(\mathbf{W}_{\mathrm{RI}},\boldsymbol{B}_{\mathrm{RI}})$ using Eq. (7)
    update $\mathbf{W}_{\mathrm{RI}}$ and $\boldsymbol{B}_{\mathrm{RI}}$
end while
return $\mathbf{W}_{\mathrm{RI}}$ and $\boldsymbol{B}_{\mathrm{RI}}$
end begin
```

As can be seen from Eq. (5), 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 $\|\mathbf{W}_{\mathrm{RI}}\|_2^2$ in Eq. (3) is a weight decay term that tends to decrease the magnitude of the weights of $\mathbf{W}_{\mathrm{RI}}$, and helps preventing over-fitting.

By incorporating Eqs. (5) and (4) into Eq. (3), we have the following objective function

$$J_{\mathrm{RI}}\left(\mathbf{W}_{\mathrm{RI}},\boldsymbol{B}_{\mathrm{RI}}\right)= \min\left(\begin{array}{l}-\dfrac{1}{N\left(K+1\right)}\displaystyle\sum_{x_i\in\mathcal{X}_{\mathrm{RI}}}\left\langle\boldsymbol{y}_{x_i},\log\boldsymbol{O}_b\left(x_i\right)\right\rangle+\\ \dfrac{\lambda_1}{2N}\displaystyle\sum_{x_i\in X}\left\|\boldsymbol{O}_a\left(x_i\right)-\overline{\boldsymbol{O}_a\left(T_\phi x_i\right)}\right\|_2^2+\dfrac{\lambda_2}{2}\left\|\mathbf{W}_{\mathrm{RI}}\right\|_2^2\end{array}\right) \tag{7}$$

We can easily see that the objective function defined by (7) 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 stochastic gradient descent (SGD) method [49], which has been widely used in complicated optimization problems such as neural networks training.

3.2. Learning Fisher discriminative CNN model

As illustrated in Figure 3, our Fisher discriminative CNN (FD-CNN) model is designed by adding a new Fisher discriminative fully-connected layer FC*c* that uses the output of layer FC*m* or FC*a* (in this situation by combining RI-CNN and FD-CNN together we can obtain a more powerful RIFD-CNN model) as input. This newly added F-

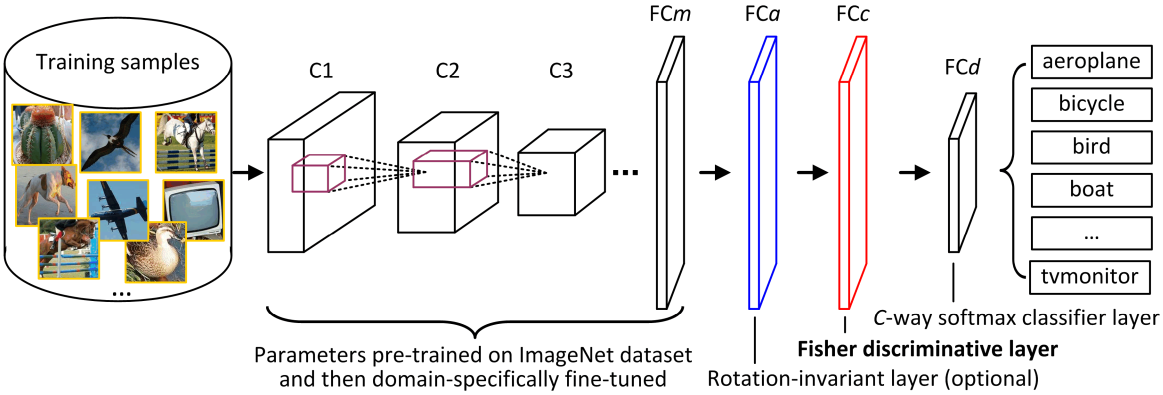
Training samples
...
C1
C2
C3
...
FCm
FCa
FCc
FCd
aeroplane
bicycle
bird
boat
...
tvmonitor
C-way softmax classifier layer
Fisher discriminative layer
Parameters pre-trained on ImageNet dataset
and then domain-specifically fine-tuned
Rotation-invariant layer (optional)

Figure 3. Architecture of the proposed FD-CNN model. It is achieved by adding a new Fisher discriminative layer on the existing CNNs.

isher discriminative layer is trained via imposing the Fisher discrimination criterion on the CNN features to enforce the learned FD-CNN features have small within-class scatter and large between-class separation. Algorithm 2 gives the pseudo-code of FD-CNN training.

Similar to RI-CNN training, to reduce the training cost, the parameters (weights and biases) of layers C1, C2, … , FC*m*, and FC*a*, denoted by $\{\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_m, \mathbf{W}_a\}$ and $\{\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_m, \boldsymbol{B}_a\}$, are pre-trained on ImageNet dataset [3], domain-specifically fine-tuned, and then transferred to our FD-CNN model. For a training sample $x_k \in \mathcal{X}_{\text{FD}}$, let $\boldsymbol{O}_a(x_k)$ be the output of layer FC*a*, $\boldsymbol{O}_c(x_k)$ be the output of layer FC*c*, $\boldsymbol{O}_d(x_k)$ be the output of softmax classifier layer FC*d*, and $(\mathbf{W}_c, \boldsymbol{B}_c)$ and $(\mathbf{W}_d, \boldsymbol{B}_d)$ be the new parameters of layers FC*c* and FC*d*. Thus, $\boldsymbol{O}_c(x_k)$ and $\boldsymbol{O}_d(x_k)$ can be computed by

$$\boldsymbol{O}_c(x_k) = \kappa\left(\mathbf{W}_c \boldsymbol{O}_a(x_k) + \boldsymbol{B}_c\right) \tag{8}$$

$$\boldsymbol{O}_d(x_k) = \varphi\left(\mathbf{W}_d \boldsymbol{O}_c(x_k) + \boldsymbol{B}_d\right) \tag{9}$$

In all our experiments, FC*c* has a size of 4096, and FC*d* has a size equal to the number of object classes. Here, different from the (*C*+1)-way softmax classifier layer FC*b* of RI-CNN model, FC*d* is now a *C*-way softmax classifier layer (without background class), so the training samples for FD-CNN learning now become all ground-truth bounding boxes for each object class denoted by $\mathcal{X}_{\text{FD}} = \{\mathcal{X}_{\text{FD}}^1, \mathcal{X}_{\text{FD}}^2, \cdots, \mathcal{X}_{\text{FD}}^C\}$, where $\mathcal{X}_{\text{FD}}^i$ is the ground-truth bounding boxes for the *i*-th object class. Given the training samples $\mathcal{X}_{\text{FD}} = \{x_k\}$ and their ground-truth label vectors $\mathcal{Y}_{\text{FD}} = \{\boldsymbol{y}_{x_k} \mid x_k \in \mathcal{X}_{\text{FD}}\}$, our objective is now to train a FD-CNN model with the input-target pairs $(\mathcal{X}_{\text{FD}}, \mathcal{Y}_{\text{FD}})$. Except for requiring that FD-CNN model should minimize the misclassification error on the training dataset, we also require that FD-CNN model should have powerful discriminative capability. For this purpose, we propose the following discriminative objective function to learn the parameters of $\mathbf{W}_{\text{FD}} = \{\mathbf{W}_1, \mathbf{W}_2, \cdots, \mathbf{W}_m, \mathbf{W}_a, \mathbf{W}_c, \mathbf{W}_d\}$ and $\boldsymbol{B}_{\text{FD}} = \{\boldsymbol{B}_1, \boldsymbol{B}_2, \cdots, \boldsymbol{B}_m, \boldsymbol{B}_a, \boldsymbol{B}_c, \boldsymbol{B}_d\}$

$$J_{\text{FD}}(\mathbf{W}_{\text{FD}}, \boldsymbol{B}_{\text{FD}}) = \min\left(M(\mathcal{X}_{\text{FD}}, \mathcal{Y}_{\text{FD}}) + \lambda_3 F(\mathcal{X}_{\text{FD}}) + \frac{\lambda_4}{2}\|\mathbf{W}_{\text{FD}}\|_2^2\right) \tag{10}$$

where λ_3 and λ_4 are two trade-off parameters that control the relative importance of the three terms.

The first term $M(\mathcal{X}_{\text{FD}}, \mathcal{Y}_{\text{FD}})$ in Eq. (10) is a classification error function that seeks to minimize the classification error for the given training samples and is computed by

$$M(\mathcal{X}_{\text{FD}}, \mathcal{Y}_{\text{FD}}) = -\frac{1}{|\mathcal{X}_{\text{FD}}|}\sum_{x_k \in \mathcal{X}_{\text{FD}}} \left\langle \boldsymbol{y}_{x_k}, \log \boldsymbol{O}_d(x_k)\right\rangle \tag{11}$$

where $|\mathcal{X}_{\text{FD}}|$ is the number of training samples in $\mathcal{X}_{\text{FD}}$.

The second term $F(\mathcal{X}_{\text{FD}})$ in Eq. (10) is a discrimination regularization constraint imposed on the CNN features. Based on the Fisher discrimination criterion [50], this can be achieved by minimizing the within-class scatter of $\mathcal{X}_{\text{FD}}$, denoted by $S_{\text{W}}(\mathcal{X}_{\text{FD}})$, and maximizing the between-class scatter of $\mathcal{X}_{\text{FD}}$, denoted by $S_{\text{B}}(\mathcal{X}_{\text{FD}})$. $S_{\text{W}}(\mathcal{X}_{\text{FD}})$ and $S_{\text{B}}(\mathcal{X}_{\text{FD}})$ are defined as

$$S_{\text{W}}(\mathcal{X}_{\text{FD}}) = \sum_{i=1}^{C}\sum_{x_k \in \mathcal{X}_{\text{FD}}^i}\left(\boldsymbol{O}_c(x_k) - \boldsymbol{m}_i\right)\left(\boldsymbol{O}_c(x_k) - \boldsymbol{m}_i\right)^T \tag{12}$$

$$S_{\text{B}}(\mathcal{X}_{\text{FD}}) = \sum_{i=1}^{C} n_i\left(\boldsymbol{m}_i - \boldsymbol{m}\right)\left(\boldsymbol{m}_i - \boldsymbol{m}\right)^T \tag{13}$$

where n_i is the number of samples in the *i*-th object class, $\boldsymbol{m}_i$ and $\boldsymbol{m}$ are the mean feature representations of $\mathcal{X}_{\text{FD}}^i$ and $\mathcal{X}_{\text{FD}}$, respectively, and are computed by

$$\boldsymbol{m}_i = \frac{1}{n_i}\sum_{x_k \in \mathcal{X}_{\text{FD}}^i} \boldsymbol{O}_c(x_k), \quad \boldsymbol{m} = \frac{1}{|\mathcal{X}_{\text{FD}}|}\sum_{x_k \in \mathcal{X}_{\text{FD}}} \boldsymbol{O}_c(x_k) \tag{14}$$

Algorithm 2 Learning FD-CNN model

```
Input: the input-target pairs $(\mathcal{X}_{\mathrm{FD}}, \mathcal{Y}_{\mathrm{FD}})$
Output: the parameters of our FD-CNN model, denoted by $\mathbf{W}_{\mathrm{FD}} = \{\mathbf{W}_1, \cdots, \mathbf{W}_m, \mathbf{W}_a, \mathbf{W}_c, \mathbf{W}_d\}$ and $\boldsymbol{B}_{\mathrm{FD}} = \{\boldsymbol{B}_1, \cdots, \boldsymbol{B}_m, \boldsymbol{B}_a, \boldsymbol{B}_c, \boldsymbol{B}_d\}$
begin
Initialize $(\mathbf{W}_c, \boldsymbol{B}_c)$ and $(\mathbf{W}_d, \boldsymbol{B}_d)$ randomly
while stopping criterion has not been met do
    compute classification error using Eq. (11)
    compute discriminative regularization term using Eq. (15)
    compute objective function $J_{\mathrm{FD}}(\mathbf{W}_{\mathrm{FD}}, \boldsymbol{B}_{\mathrm{FD}})$ using Eq. (16)
    update $\mathbf{W}_{\mathrm{FD}}$ and $\boldsymbol{B}_{\mathrm{FD}}$
end while
return $\mathbf{W}_{\mathrm{FD}}$ and $\boldsymbol{B}_{\mathrm{FD}}$
end begin
```

Intuitively, the discriminative regularization term $F(\mathcal{X}_{\mathrm{FD}})$ can be defined as

$$F(\mathcal{X}_{\mathrm{FD}}) = tr\left(S_{\mathrm{W}}(\mathcal{X}_{\mathrm{FD}})\right) - tr\left(S_{\mathrm{B}}(\mathcal{X}_{\mathrm{FD}})\right) \qquad (15)$$

Thus, by incorporating Eqs. (15) and (11) into Eq. (10), we can form the following discriminative objective function of FD-CNN model

$$J_{\mathrm{FD}}\left(\mathbf{W}_{\mathrm{FD}}, \boldsymbol{B}_{\mathrm{FD}}\right) = \min \left(\begin{array}{l} -\frac{1}{\left|\mathcal{X}_{\mathrm{FD}}\right|} \sum_{x_k \in \mathcal{X}_{\mathrm{FD}}} \left\langle \boldsymbol{y}_{x_k}, \log \boldsymbol{O}_d\left(x_k\right) \right\rangle + \\ \lambda_3 \left(tr\left(S_{\mathrm{W}}(\mathcal{X}_{\mathrm{FD}})\right) - tr\left(S_{\mathrm{B}}(\mathcal{X}_{\mathrm{FD}})\right) \right) + \frac{\lambda_4}{2} \left\| \mathbf{W}_{\mathrm{FD}} \right\|_2^2 \end{array} \right) \qquad (16)$$

As can be seen from Eq. (16), the new discriminative objective function not only minimizes the classification loss, but also imposes a regularization constraint to make the learned CNN features be more discriminative.

4. Experiments

In this section, we first demonstrate that the use of our RI-CNN features can outperform the state-of-the-arts [2, 46, 51, 52] for rotation-invariant object detection, specifically for finding aerial cars that appear at arbitrary orientations on a publicly available satellite image dataset [33]. We next focus on comprehensively evaluating the proposed RI-CNN model, FD-CNN model, and especially their combination (RIFD-CNN model) for standard object detection tasks on PASCAL VOC 2007 dataset [28]. The experimental results show that our method significantly improves the existing baseline methods such as [5, 17]. The performance of object detection is measured according to the PASCAL criterion [28], i.e., the average precision (AP) and the mean AP over all object classes. Without explicit statement, we adopt the standard IoU criteria of 0.5 for all experimental evaluation.

Model	AP (%)
RC-RBM IHOF [46] + linear SVM	72.7
Gradients IHOF [46] + linear SVM	74.7
RC-RBM [46] + Gradients IHOF [46] + linear SVM	77.6
Standard HOG [32] + slot kernel structured SVM [52]	75.7
Rotation-invariant HOG [51] + linear SVM	82.6
Rotation-invariant HOG [51] + Random Forest	84.2
Fine-tuned CNN with AlexNet [2] + linear SVM	90.2
Our RI-CNN with AlexNet [2] + linear SVM	94.6

Table 1 The detection result comparison for different methods on the aerial car detection dataset.

4.1. Aerial car detection

In this experiment, we use a public dataset introduced by [33], which has been widely used by some published work such as [33, 46, 51, 52]. This dataset consists of 30 aerial images with a total number of 1319 manually labeled cars that appear at arbitrary orientations. The task is challenging due to the low resolution and the varying illumination conditions caused by the shadows of buildings. Like the compared work [33, 46, 51, 52], we perform 5-fold cross validation and report average results across all folds.

Implementation. We adopt the most popular AlexNet CNN [2] pre-trained on ImageNet [3] as our building block. To adapt it to the new aerial car detection task, we first perform SGD fine-tuning of the whole CNN parameters with a 2-way softmax classification layer (one for car and the other for background) using augmented training samples obtained from the aerial dataset [33]. In this step, we sample 32 positives and 96 negatives for each SGD iteration to form a mini-batch of size 128. The SGD learning rate is set to 0.0005 to allow fine-tuning to make progress while not clobbering the initialization. We set the momentum to 0.9, and the weight decay to 0.0005 for all the layers. After that, we further train a RI-CNN model by adding and learning a new rotation-invariant layer as described in section 3.1. In this step, we randomly sample 2 positive examples and 2 negative examples together with their corresponding 4×35=140 rotated examples to form a mini-batch of size 144. The SGD learning rate is set to 0.01 for the last two layers training and 0.0001 for the whole network fine-tuning, and decreases by 0.5 every 10000 iterations. The parameters of Eq. (7) are set to λ_1=0.001 and λ_2=0.0005. To augment the training data, we treat all region proposals (provided by [33]) with ≥0.5 IoU overlap with a ground truth box as initial positives and the rest as initial negatives. Then, we use 35 rotation transformations $T_\phi = \{T_{\phi_1}, T_{\phi_2}, \cdots, T_{\phi_K}\}$ with $\phi = \{10^\circ, 20^\circ, \cdots, 350^\circ\}$ to perform the rotation operation on each initial training sample to obtain 36x training samples. The augmented data are used for both fine-tuning and RI-CNN model training. Finally, we train a simple and efficient linear SVM classifier to classify all region proposals as cars or background.

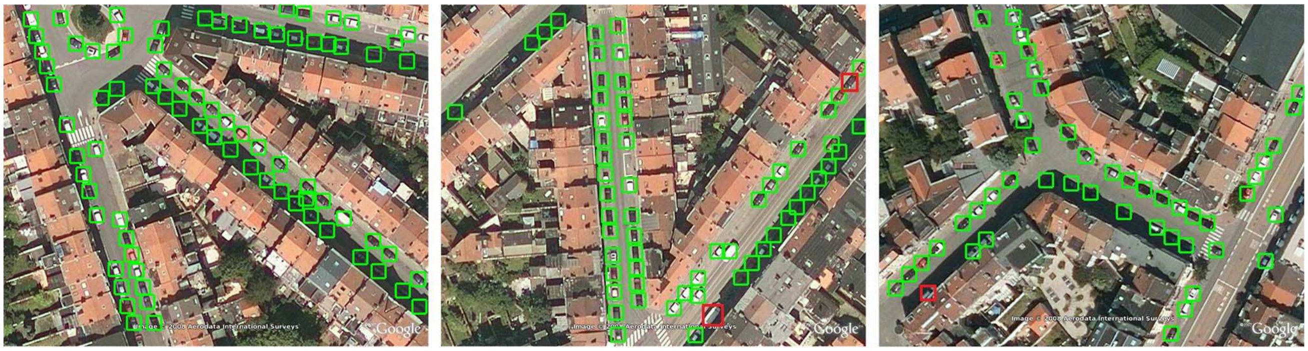

Figure 4. Some example detections (true positive in green, false negative in red) by using our RI-CNN features with a linear SVM.

Results and comparison with state-of-the-arts. Figure 4 shows some example detections (true positive in green, false negative in red) by using our RI-CNN features with a linear SVM. As can be seen from Figure 4, despite the large variations in the orientations, the proposed method has successfully detected and located most of the cars. Besides, we also compare our results with some state-of-the-arts in Table1. As can be seen from Table 1, using a simple linear SVM classifier, our RI-CNN model can 1) significantly improve the performance of the traditional CNN model with AlexNet architecture [2] fine-tuned on the aerial car dataset, which is also the baseline of our method, and 2) outperform all other recent publications [46, 51, 52] which address the rotation problem in different ways, where [52] uses slot kernel structured SVM and the standard HOG feature, [46] focuses on learning rotation-invariant feature and descriptor called RC-RBM and IHOF, respectively, and [51] presents rotation-invariant HOG descriptors using Fourier analysis in polar and spherical coordinates.

4.2. Object detection on PASCAL VOC 2007

In this experiment, we focus on the PASCAL VOC 2007 dataset [28], which is the most common benchmark to evaluate object detection algorithms. This dataset consists of 9963 complex scene images with 5011 training images and 4952 testing images, in which bounding boxes of 20 diverse object classes were manually labeled. The task is to predict bounding boxes of the objects of interest if they are present in the images.

Implementation. We build on the high-performance VGGNet CNN model [9], that was pre-trained on ImageNet [3] and then fine-tuned on PASCAL VOC 2007 dataset to train our RI-CNN model, FD-CNN model, and RIFD-CNN model, respectively. To augment the training data, we adopt a family of 6 rotation transformations $T_{\phi} = \{T_{\phi_1}, T_{\phi_2}, \cdots, T_{\phi_K}\}$ with $\phi = \{\pm 10^{\circ}, \pm 20^{\circ}, \pm 30^{\circ}\}$ to carry out the rotation operation on each training sample to obtain 7x training samples. As [5], we map each object proposal (obtained via selective search method [29]) to the ground-truth instance with which it has maximum IoU overlap (if any) and label it as a positive for the matched ground-truth class if the IoU is at least 0.5. All other proposals are labeled as negative examples for all classes. For both RI-CNN model and FD-CNN model training, the learning rate is set to 0.01 for the last two layers and 0.0001 for the whole network fine-tuning, and decreases by 0.5 every 10000 iterations. The parameters of Eqs. (7) and (16) are set to λ_1=0.001 , λ_2=0.0005 , λ_3=0.005 , and λ_4=0.0005 . In each SGD iteration of RI-CNN training, we randomly sample 10 positive examples over all classes and 10 negative examples together with their corresponding 20×6=120 rotated examples to construct a mini-batch of size 140. Following the R-CNN framework [5], we use our trained models to extract new CNN features from object proposals provided by selective search method [29], classify them with class-specific linear SVMs (trained using ground-truth positive samples and negative samples obtained via hard negative mining [30]), and then perform non-maximum suppression and bounding box regression [30].

Results and comparison with state-of-the-arts. Table 2 reports the detection performance of our improved CNN models including RI-CNN model, FD-CNN model, and their combination (RIFD-CNN model). These results are obtained based on the building block of R-CNN [5] with VGGNet [9] and bounding box regression (BB), which is therefore also our baseline method. Compared with the baseline method, from Table 2 we can observe 1) RI-CNN model only improves the performance slightly for mAP (+0.7%) averaged over 20 categories. This can be easily explained: different from aerial images (as illustrated in Figure 4) in which objects appear at arbitrary orientations, the objects in nature scene images are typically in an upright orientation due to the Earth's gravity and so the orientation variations across images are generally small. For the improved object classes, rotations are mainly caused by aeroplane/bird flying and animal jumping. 2) FD-CNN model improves the performance for all 20 object

Model	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN with VGGNet & BB	73.4	77.0	63.4	45.4	44.6	75.1	78.1	79.8	40.5	73.7	62.2	79.4	78.1	73.1	64.2	35.6	66.8	67.2	70.4	71.1	66.0
+ RI-CNN	75.6	77.1	65.6	45.1	45.2	74.9	78.1	81.2	40.7	73.5	62.8	81.2	79.6	74.5	66.1	35.1	67.0	66.9	70.1	73.3	66.7
+ FD-CNN	76.3	**80.8**	68.5	**50.1**	46.1	77.2	79.6	81.9	47.7	75.9	66.2	81.6	79.9	74.3	69.8	**41.1**	68.9	70.3	73.3	73.8	69.2
+ RIFD-CNN	**77.4**	**80.8**	**70.7**	49.9	**47.2**	**77.6**	**79.9**	**82.7**	**48.6**	**76.1**	**67.1**	**81.9**	**80.2**	**74.6**	**71.9**	40.9	**69.5**	**70.7**	**73.5**	**74.2**	**69.8**

Table 2. Detection performance on PASCAL VOC 2007 test set using our improved CNN models including RI-CNN model, FD-CNN model, RIFD-CNN model, and the baseline method of R-CNN [5] with VGGNet [9] and bounding box regression (BB). The entries with the best APs for each object category are bold-faced.

Model	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [30]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
DPM ST [34]	23.8	58.2	10.5	8.5	27.1	50.4	52.0	7.3	19.2	22.8	18.1	8.0	55.9	44.8	32.4	13.3	15.9	22.8	46.2	44.9	29.1
DPM HSC [38]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3
CNN-DPM-BB [22]	50.9	64.4	43.4	29.8	40.3	56.9	58.6	46.3	33.3	40.5	47.3	43.4	65.2	60.5	42.2	31.4	35.2	54.5	61.6	58.6	48.2
E2E-DPM [23]	49.3	69.5	31.9	28.7	40.4	61.5	61.5	41.5	25.5	44.5	47.8	32.0	67.5	61.8	46.7	25.9	40.5	46.0	57.1	58.2	46.9
DP-DPM [20]	44.6	65.3	32.7	24.7	35.1	54.3	56.5	40.4	26.3	49.4	43.2	41.0	61.0	55.7	53.7	25.5	47.0	39.8	47.9	59.2	45.2
Sliding-window CNN [12]	64.1	72.3	62.8	44.0	44.2	66.4	72.5	67.7	35.2	68.9	35.9	62.7	69.0	65.7	65.8	36.2	60.1	50.3	63.2	66.0	58.6
R-CNN with AlexNet	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN with VGGNet	71.6	73.5	58.1	42.2	39.4	70.7	76.0	74.5	38.7	71.0	56.9	74.5	67.9	69.6	59.3	35.7	62.1	64.0	66.5	71.2	62.2
R-CNN with AlexNet & BB	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
R-CNN with VGGNet & BB	73.4	77.0	63.4	45.4	44.6	75.1	78.1	79.8	40.5	73.7	62.2	79.4	78.1	73.1	64.2	35.6	66.8	67.2	70.4	71.1	66.0
+ FGS [17]	74.2	78.9	67.8	51.6	**52.3**	75.7	78.7	76.6	45.4	72.4	63.1	76.6	79.3	70.7	68.0	40.3	67.8	61.8	70.2	71.6	67.2
+ StructObj [17]	73.1	77.5	69.2	47.6	47.6	74.5	78.2	75.4	44.5	76.3	64.9	76.7	76.3	69.9	68.1	39.4	67.0	65.6	68.7	70.9	66.6
+ StructObj + FGS [17]	74.1	**83.2**	67.0	50.8	51.6	76.2	81.4	77.2	48.1	**78.9**	65.6	77.3	78.4	75.1	70.1	41.4	69.6	60.8	70.2	73.7	68.5
+ RIFD-CNN (ours)	77.4	80.8	**70.7**	49.9	47.2	77.6	79.9	82.7	48.6	76.1	67.1	81.9	80.2	74.6	71.9	40.9	69.5	**70.7**	73.5	**74.2**	69.8
+ RIFD-CNN + FGS (ours)	**78.9**	82.5	69.6	**54.2**	49.7	**78.3**	**82.0**	**83.4**	**51.1**	76.0	**69.0**	**82.2**	**80.7**	**77.2**	**73.1**	**42.6**	**70.3**	70.4	**74.2**	74.1	**71.0**

Table 3. Performance comparison of our method against other methods on PASCAL VOC 2007 test set. Rows 1-7 show sliding window detectors that employ different features. Rows 8-14 show results for R-CNN framework [5] with two different networks (AlexNet [2] and VGGNet [9]) and its improvement work [17] as a strong baseline. The last two rows report the results of our method and its combination with FGS. The entries with the best APs for each object category are bold-faced.

categories. Especially for those classes with big within-class variability or large between-class similarity such as bird, boat, chair, plant, etc., we obtain significant performance improvement. 3) Further improvement has been achieved by combing RI-CNN model and FD-CNN model together (RIFD-CNN model), boosting the baseline model [5] by 3.8% in mAP. The results demonstrate that our method is effective for addressing the problems of object rotation, within-class variability, and between-class similarity.

In Table 3, we compare our method with other published work [5, 12, 17, 20, 22, 23, 30, 34, 38] on PASCAL VOC 2007 test set. Rows 1-7 show sliding window detectors that employ different features, where the first [30] uses only HOG, the next two [34, 38] use different feature learning approaches to augment or replace HOG, and the last four [12, 20, 22, 23] employ CNN features. Rows 8-14 show the results for R-CNN method [5] with two different networks (AlexNet [2] and VGGNet [9]) and its improvement work [17] as a strong baseline. The last two rows report the results of our method and its combination with fine-grained search (FGS) method [17]. For all methods, the fine-tuning/ training of the networks (if applicable) as well as the training of the detection SVMs were performed on VOC 2007 train+val dataset. As shown in Table 3, we achieve state-of-the-art results for 17 out of 20 categories compared with the existing baseline methods [5, 17]. To be specific, our best result (RIFD-CNN + FGS) improves upon the best results of the baseline methods of [5] and [17] by 5.0% and 2.5%, respectively, in terms of mAP, which demonstrates the effectiveness and superiority of our method.

5. Conclusions

In this paper, we proposed an effective method to boost the performance of object detection in R-CNN framework [5] by learning a rotation-invariant and Fisher discriminative CNN model. The proposed method could effectively address the challenges of object rotation, within-class variability, and between-class similarity. In the experiments, we have comprehensively evaluated the proposed method for object detection tasks on a public available aerial image dataset [33] and the PASCAL VOC 2007 dataset [28]. On both two datasets, we have achieved state-of-the-art performance compared with the existing baseline methods.

Acknowledgements

This work was supported in part by the National Science Foundation of China under Grants 61401357, 61522207, and 61473231.

References

[1] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation,* 1(4): 541-551, 1989.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012.

[3] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and F. Li. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*: 1-42, 2015.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015.

[7] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*, 2014.

[8] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[10] C. Szegedy, A. Toshev, and D. Erhan. Deep neural networks for object detection. In *NIPS*, 2013.

[11] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, and C.-C. Loy. Deepid-net: Deformable deep convolutional neural networks for object detection. In *CVPR*, 2015.

[12] G. Papandreou, I. Kokkinos, and P.-A. Savalle. Modeling Local and Global Deformations in Deep Learning: Epitomic Convolution, Multiple Instance Learning, and Sliding Window Detection. In *CVPR*, 2015.

[13] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky. Sparse Convolutional Neural Networks. In *CVPR*, 2015.

[14] R. Girshick. Fast R-CNN. In *ICCV*, 2015.

[15] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

[16] D. Yoo, S. Park, J.-Y. Lee, A. Paek, and I. S. Kweon. AttentionNet: Aggregating Weak Directions for Accurate Object Detection. In *ICCV*, 2015.

[17] Y. Zhang, K. Sohn, R. Villegas, G. Pan, and H. Lee. Improving object detection with deep convolutional networks via bayesian optimization and structured prediction. In *CVPR*, 2015.

[18] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV*, 2014.

[19] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[20] R. Girshick, F. Iandola, T. Darrell, and J. Malik. Deformable part models are convolutional neural networks. In *CVPR*, 2015.

[21] Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler. segDeepM: Exploiting Segmentation and Context in Deep Neural Networks for Object Detection. In *CVPR*, 2015.

[22] P.-A. Savalle, S. Tsogkas, G. Papandreou, and I. Kokkinos. Deformable Part Models with CNN Features. In *Parts and Attributes Workshop, ECCV*, 2014.

[23] L. Wan, D. Eigen, and R. Fergus. End-to-End Integration of a Convolutional Network, Deformable Parts Model and Non-Maximum Suppression. In *CVPR*, 2015.

[24] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous detection and segmentation. In *ECCV*, 2014.

[25] N. Zhang, J. Donahue, R. Girshick, and T. Darrell. Part-based R-CNNs for fine-grained category detection. In *ECCV*, 2014.

[26] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko. LSDA: Large scale detection through adaptation. In *NIPS*, 2014.

[27] J. Dai, K. He, and J. Sun. Convolutional Feature Masking for Joint Object and Stuff Segmentation. In *CVPR*, 2015.

[28] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) Challenge. *IJCV,* 88(2): 303-338, 2010.

[29] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV,* 104(2): 154-171, 2013.

[30] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI,* 32(9): 1627-1645, 2010.

[31] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV,* 60(2): 91-110, 2004.

[32] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[33] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, 2008.

[34] J. J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *CVPR*, 2013.

[35] G. Cheng, J. Han, L. Guo, and T. Liu. Learning coarse-to-fine sparselets for efficient object detection and scene classification. In *CVPR*, 2015.

[36] S. Fidler, R. Mottaghi, and R. Urtasun. Bottom-up segmentation for top-down detection. In *CVPR*, 2013.

[37] 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.,* 98: 119-132, 2014.

[38] X. Ren and D. Ramanan. Histograms of sparse codes for object detection. In *CVPR*, 2013.

[39] 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.,* 53(8): 4238-4249, 2015.

[40] K. Sohn and H. Lee. Learning Invariant Representations with Local Transformations. In *ICML*, 2012.

[41] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial Transformer Networks. In *NIPS*, 2015.

[42] M. Paulin, J. Revaud, Z. Harchaoui, F. Perronnin, and C. Schmid. Transformation pursuit for image classification. In *CVPR*, 2014.

[43] K. Lenc and A. Vedaldi. Understanding image representations by measuring their equivariance and equivalence. In *CVPR*, 2015.

[44] L. Sifre and S. Mallat. Rotation, scaling and deformation invariant scattering for texture discrimination. In *CVPR*, 2013.

[45] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *NIPS*, 2014.

[46] U. Schmidt and S. Roth. Learning rotation-aware features: From invariant priors to equivariant descriptors. In *CVPR*, 2012.

[47] M. Yang, L. Zhang, X. Feng, and D. Zhang. Fisher discrimination dictionary learning for sparse representation. In *ICCV*, 2011.

[48] J. Xie, Y. Fang, F. Zhu, and E. Wong. DeepShape: Deep Learned Shape Descriptor for 3D Shape Matching and Retrieval. In *CVPR*, 2015.

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

[50] R. Duda, P. Hart, and D. Stork. *Pattern classification (2nd Ed.)*. Wiley-Interscience, 2000.

[51] K. Liu, H. Skibbe, T. Schmidt, T. Blein, K. Palme, T. Brox, and O. Ronneberger. Rotation-invariant HOG descriptors using fourier analysis in polar and spherical coordinates. *IJCV,* 106(3): 342-364, 2014.

[52] A. Vedaldi, M. Blaschko, and A. Zisserman. Learning equivariant structured output SVM regressors. In *ICCV*, 2011.