DIOD: Fast and Efficient Weakly Semi-Supervised Deep Complex ISAR Object Detection

Bin Xue, *Student Member, IEEE*, and Ningning Tong

***Abstract*—Inverse synthetic aperture radar (ISAR) object detection is one of the most important and challenging problems in computer vision tasks. To provide a convenient and high-quality ISAR object detection method, a fast and efficient weakly semi-supervised method, called deep ISAR object detection (DIOD), is proposed, based on advanced region proposal networks (ARPNs) and weakly semi-supervised deep joint sparse learning: 1) to generate high-level region proposals and localize potential ISAR objects robustly and accurately in minimal time, ARPN is proposed based on a multiscale fully convolutional region proposal network and a region proposal classification and ranking strategy. ARPN shares common convolutional layers with the Inception-ResNet-based system and offers almost cost-free proposal computation with excellent performance; 2) to solve the difficult problem of the lack of sufficient annotated training data, especially in the ISAR field, a convenient and efficient weakly semi-supervised training method is proposed with the weakly annotated and unannotated ISAR images. Particularly, a pairwise-ranking loss handles the weakly annotated images, while a triplet-ranking loss is employed to harness the unannotated images; and 3) to further improve the accuracy and speed of the whole system, a novel sharable-individual mechanism and a relational-regularized joint sparse learning strategy are introduced to achieve more discriminative and comprehensive representations while learning the shared- and individual-features and their correlations. Extensive experiments are performed on two real-world ISAR datasets, showing that DIOD outperforms existing state-of-the-art methods and achieves higher accuracy with shorter execution time.**

***Index Terms*—Inverse synthetic aperture radar (ISAR), object detection, region proposal network (RPN), weakly semi-supervised deep joint sparse learning (WSSDJSL).**

I. Introduction

INVERSE synthetic aperture radar (ISAR) is used to explore the response of the sensing element to the object motion to produce 2-D detailed images of moving objects that are not cooperative with the sensor [1]. ISAR imagery plays an important role, particularly in sensing and military applications, such as object detection, localization, and recognition. In particular, accurate ISAR object detection [2] is a most important and challenging problem in computer vision tasks, especially under the complex conditions that are more difficult than detection within natural and infrared images. However, we found that the progress on solving this problem had been slow since the special recognition tasks began in 2009 [3]–[5].

There are some serious obstacles to achieve an excellent ISAR object recognition system.

1) The multimodal problem, such as many different viewpoints, scales, deformations, and frequencies within ISAR images.
2) ISAR images are randomly covered with a variety of noises, and the structure and scattering characteristics of ISAR objects are seriously weakened.
3) ISAR image objects are generally smaller than natural image objects.
4) The large similarities existing in the true and mendacious objects, and the significant intraclass diversity among different styles and properties of objects.

However, some commonly used techniques can detect objects well only in specific labeled images, and require assigning the classes and positions of the objects and background disturbers [6]; such assignment is a very time-consuming and laborious task when annotating objects manually. Certain methods model objects of different classes independently and must relearn the background repeatedly [7], [8]. Other methods focus on low-level features to differentiate objects [9], [10] or require all the objects share a common aspect ratio [11]. In general, all the conventional algorithms must construct the features of ISAR objects manually; this is a complicated, inconvenient, and time-consuming task. Moreover, the conventional algorithms attempt to mine some discriminative superficial local features, instead of the latent, superficial, and deep features.

Recently, with the successful application of deep convolutional neural networks (DCNNs), DCNN has become one of the most promising means to address some of the main problems described above [12], [13]. Although DCNN has achieved great successes in extracting features, some problems remain. The existing DCNN detection methods are often time consuming and it is difficult for them to solve some specific problems, such as learning Albert Einstein's Theory of Relativity directly. Simultaneously, there is a lack of available labeled data to train DCNN, especially in the ISAR field. The preceding achievements and challenges raise a number of issues: identification of high-quality ISAR features, achieving satisfactory results with few labeled data, and optimizing the whole system in a more reliable, efficient, and practical way.

Manuscript received January 26, 2018; revised June 21, 2018; accepted July 10, 2018. Date of publication July 27, 2018; date of current version July 19, 2019. This work was supported by the National NSFC under Grant 61571459, Grant 61631019, and Grant 61701526. This paper was recommended by Associate Editor J. Su. *(Corresponding author: Bin Xue.)*

The authors are with Air Force Engineering University, Xi'an 710051, China (e-mail: xxbbxl@sina.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/TCYB.2018.2856821

Until recently, the most successful methods to object detection used the well known "sliding window" paradigm, in which a computationally efficient classifier tests for object presence in each candidate image window. Sliding window classifiers scale linearly with the number of windows tested, the steady increase in complexity of the core classifiers has led to improved detection quality, but at the cost of significantly increased computation time per window. One approach for overcoming the tension between computational tractability and high detection quality is through the use of "object proposals." Under the assumption that all objects of interest share common visual properties that distinguish them from the background, one can design or train a method that, given an image, outputs a set of proposal regions that are likely to contain objects. If high object recall can be reached with considerably fewer windows than used by sliding window detectors, significant speed-ups can be achieved, enabling the use of more sophisticated classifiers. This may also improve detection quality by reducing spurious false positives. So, it is helpful and necessary to deploy ARPN in ISAR detection.

It is known that no public efficient ISAR object detection algorithms currently exist. In this paper, a weakly semi-supervised ISAR object detection method, called deep ISAR object detection (DIOD), is proposed, based on an advanced region proposal network (ARPN) and weakly semi-supervised deep joint sparse learning (WSSDJSL). Specifically, this paper provides three main contributions.

1) To generate high-level region proposals and localize potential ISAR objects robustly and accurately in less time, a novel and efficient ARPN method is proposed, based on a multiscale fully convolutional region proposal network (RPN) and a region proposal classification and ranking strategy (RPCRS). In this paper, "seed" boxes at multiple scales and aspect ratios are built as references that can avoid image enumerating, offering almost cost-free proposals computation. Moreover, an efficient RPCRS is proposed. The strategy achieves good performance while providing significant computational savings.
2) A public problem is the lack of sufficient available labeled training data in the field of ISAR object detection. The limited amount of available annotated training data becomes one of the bottlenecks in feasible DCNN training. To relieve the problem, a weakly semi-supervised training (WSST) method is proposed to train increased high-capacity and robust DCNNs on two real-world ISAR datasets. In particular, a pairwise-ranking loss handles images that are weakly annotated, and a triplet-ranking loss is employed to harness unannotated images. This proposed method is found to achieve better performance than that of the majority of the state-of-the-art object detection models that adopt unsupervised training.
3) To further improve the accuracy and speed of the whole detection system, a novel efficient sharable-individual mechanism (SIM) and a relational-regularized joint sparse learning (RRJSL) strategy are proposed. SIM is introduced to not only learn shared convolutional features to retain their common properties but also learn an individual metric for each to keep its specific property. RRJSL can simultaneously learn the latent shared feature, the individual features, and the relations among them. Learning both mid- and low-level representations jointly is helpful to make representation more discriminative, comprehensive, and compact. Moreover, SIM and RRJSL are used to weakly fine-tune between ARPN and Inception-ResNet with efficient inference and transfer learning. In this regard, the proposed method produces more compelling accuracy and speed than the state-of-the-art methods.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III introduces the proposed ISAR objects detection network. The experimental results, comparisons, and ablation studies are presented in Section IV. Section V concludes this paper.

II. Related Work

A. Deep Learning for Object Detection

It has been proven that CNN is good at detecting objects within natural images, however, it continues to have difficulty detecting geospatial objects effectively in high resolution optical remote sensing images because there are many variances in the remote sensing images. To effectively address the problem, Cheng *et al.* proposed a method in [14], to learn rotation-invariant factors based on CNN to improve the detection performance. Moreover, they used an optimized objective function with a regularization constraint to train DCNN, and then the samples' representations are explicitly enforced, hence achieving rotation invariance.

A deep information-driven method is proposed in [15], based on a deep dynamic neural network and combined with a hidden Markov model (HMM) to segment and recognize objects within depth and RGB images. A novel deep belief network considers skeletal dynamic information, and a 3-D CNN managed and fused batches of depth and RGB images. The gesture sequence is achieved through the modeling of the emission probabilities of the HMM.

B. Object Proposal Method

Object proposal methods are aimed to generate a small number of high-quality category-independent proposals, such that each object is well captured by at least one proposal. Object proposals have been used in many computer vision tasks, such as segmentation [16], object detection [17], and classification [18]. A substantial number of object proposal methods have been comprehensively surveyed and studied in [19] and [20], such as selective search (SS) [21], EdgeBoxes (EB) [22], multiscale combinatorial grouping [16], and RPN [23]. There is no learned parameter in SS [21], which is based on the bag-of-words model to merge superpixels to generate proposals greedily. However, SS achieves a lower bound on the recall and requires design features and control of the number of proposals to be performed manually; SS also requires extraction of 10^3–10^5 bounding boxes per image, requiring a massive amount of computation time.

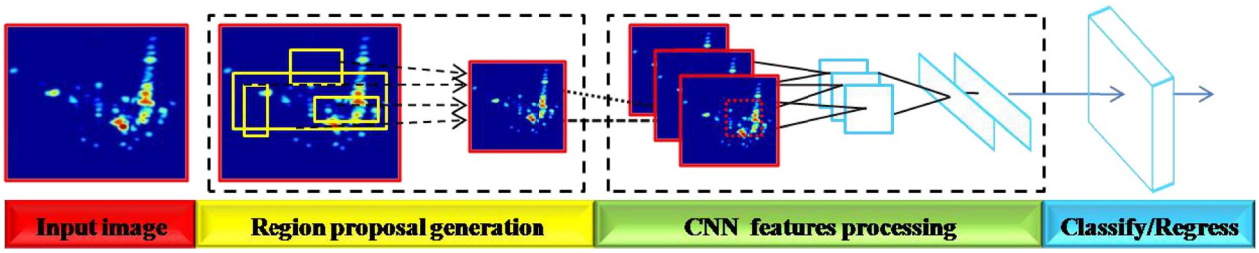


Fig. 1. Overview framework of DIOD.

EB [22] starts from a sliding window pattern, but builds on the measure or estimation of the number of edges by a simple box objectness score. A simplified but informative representation of an image is provided by edges, and a group of bounding box proposals is generated to reduce the group of positions. However, when edges are crossing complicatedly, or under the shadow of an object, its performance is not good in general. Moreover, the choice of the number and position of the seed is another problem, counting only the number of edge pixels does not result in an informative box and provides poor performance.

C. Scared Labeled Data Solution

In general, supervised learning is used in training DCNN, because DCNN can achieve satisfactory performance with huge labeled datasets. However, there is the public problem of the lack of sufficient available labeled training data in many fields. It is difficult and expensive to collect and process such data. Dosovitskiy *et al.* [24] used unlabeled data to train DCNN on a set of surrogate classes, which are generated automatically from unlabeled images, with an unsupervised learning algorithm and achieved excellent performance on several datasets.

To solve the problem, in [25], a novel deep autoencoder with rich complementary features was proposed, the autoencoder's decoding layers are replaced by classification layers, leading the features to become more discriminative. Moreover, the proposed autoencoder combines the discrete cosine transform features into the bottleneck layer to make the bottleneck features complementary.

It is of great significance for cognitive robotic systems to use excellent and suitable learning algorithms to exhibit highly intelligent and adaptive behaviors. Cui *et al.* [26] presented a sparse constrained restricted Boltzmann machine method, which limits the expectation of the hidden unit's values on RBM to achieve more effective and sparse feature representations in image-based robotic perception and actions.

III. DIOD: ISAR Object Detection Network

Fig. 1 shows an overview framework of the proposed ISAR object detection method, namely, DIOD, which includes three main sections. In the first section, the class-independent high-level region proposals are produced using ARPN. Next, in the feature processing section, for each proposal generated from the former section, the corresponding features are processed with WSSDJSL. Finally, linear SVMs and bounding boxes are used to classify and regress, respectively.

In this paper, ARPNs share convolutional layers with Inception-ResNet to produce high-quality region proposals, which offers nearly cost-free proposals computation. WSST is used to address the problem of the lack of sufficient available labeled images. SIM and RRJSL are used to further improve the accuracy and speed of the whole system.

A. Advanced Region Proposal Networks

An ARPN, a type of fully convolutional network [27], is built by adding some convolutional layers. ARPN takes an image of arbitrary size as its input, and outputs rectangular region proposal grids with efficient inference and transfer learning, each of which has an objectness score to record the probability of being an object; at each position, the region bounds and scores can be simultaneously regressed. A common group of convolutional layers are shared between ARPN and Inception-ResNet. Fig. 2 shows the architecture of an ARPN at one single position.

The elementary parts of ARPNs, i.e., convolution, pooling, and activation, depend only on the relative spatial coordinates. For a particular layer, x_{ij} and y_{ij} represent the location (i, j)'s data vector of the particular and following layer, respectively; thus, y_{ij} [27] is given by

$$y_{ij} = f_{ks}\left(\left\{x_{si+\delta i, sj+\delta j}\right\}, 0 \le \delta i, \delta j \le k\right) \tag{1}$$

where k and s denote kernel size and stride factor, respectively, f_{ks} represents the type of layer: convolution, max pooling, and activation function, corresponding to matrix multiplication, spatial max, and nonlinearity, respectively. The final shared convolutional layer outputs convolutional feature maps, and a mini network, which is above the maps, is slid to produce region proposals. An $n \times n$ spatial window of the input feature map is taken as the input, and each sliding window is mapped into a lower-dimensional feature.

1) Seeds: Seed boxes are built in ARPNs to effectively predict proposals with a large range of scale and aspect ratios. Several proposals at each location of the sliding window can be simultaneously predicted. Suppose there are a maximum of k possible proposals, corresponding to k reference boxes at each position; in the reg layer, k boxes have $4k$ outputs that represent the coordinates, and the cls layer has k scores, which are produced by logistic regression to evaluate each proposal's object probability. A seed, which is situated at the

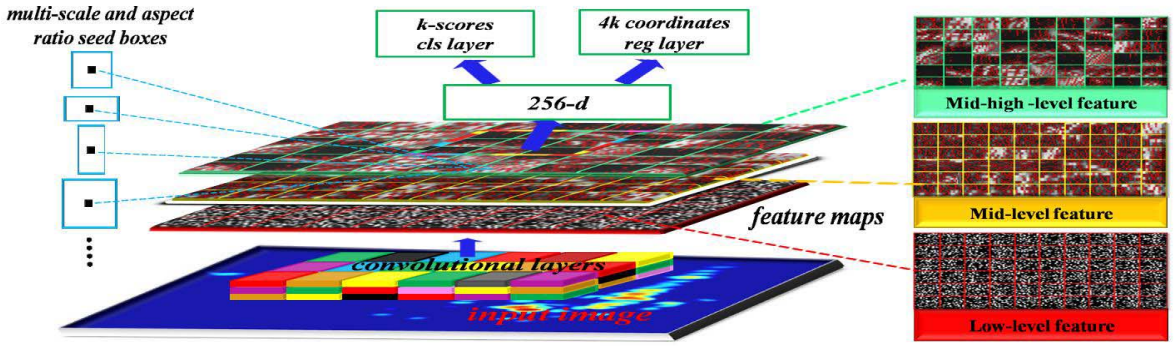


Fig. 2. Architecture of an ARPN.

sliding window's center, corresponds to an aspect ratio and a scale style (Fig. 2). Three aspect ratios and three scales are used to produce $k = 9$ proposals at each position.

2) Region Proposal Classification and Ranking: Many redundant proposals may be produced. Thus, a ranker is proposed to order a group of top-ranked proposals, which may belong to the same objects, and ensure that each object has a set of top-ranked regions that simultaneously suppresses the undesirable redundant sample proposals.

Our ranker incrementally adds regions based on the combination of an object's appearance score and a penalty for overlapping with previously added related proposals. By considering the overlap with higher ranked proposals, our ranker ensures that redundant proposals are suppressed.

By writing a scoring function $S(x, r, w)$ over the set of proposals x and their ranking r, we cast the ranking problem as a joint inference problem [28]. The goal is to find the parameters w such that $S(x, r, w)$ gives higher scores to rankings that place related proposals for all objects in high ranks

$$S(x, r, w) = \sum_i \alpha(r_i) \cdot \left(w_\alpha^T \Psi(x_i) - w_p^T \Phi(r_i) \right). \tag{2}$$

$S(x, r, w)$ denotes the ranking score over a set of proposals x, $\alpha(r_i)$ denotes the ith balance weight between Ψ (x) and Φ (r). The score is a combination of appearance features Ψ (x) and overlapping penalty terms Φ (r), where r denotes the rank of a set of proposals, ranging from 1 to the number of regions M. This allows us to jointly learn the appearance model and the tradeoff for overlapping regions. Φ (r) is the concatenation of two vectors $\Phi_1(r)$ and $\Phi_2(r)$; $\Phi_1(r)$ is a penalization function which penalizes regions owing high overlap with previously ranked proposals and $\Phi_2(r)$ is the suppression function which further suppresses the proposals that overlap with multiple higher ranked proposals. The second penalty is necessary to continue to enforce diversity after many proposals have at least one overlapping related region. Since the strength of the penalty should depend on the number of overlaps, we want to determine the overlap specific weights. To do so, we quantize the overlaps into bins of 10% and map the values to a 10-D vector $q(ov)$, with 1 for the bin it falls into and 0 for all other bins

$$\Phi_1(r_i) = q\left(\max_{\{j | r_j < r_i\}} ov(i, j) \right) \tag{3}$$

$$\Phi_2(r_i) = \sum_{\{j | r_j < r_i\}} q(ov(i, j)). \tag{4}$$

The overlap score $ov(i, j)$ between two proposals i and j is computed as the area of their intersection divided by their union, A_i and A_j are the sets of pixels belonging to region i and j, as follows:

$$ov(i, j) = \left(A_i \bigcap A_j \right) / \left(A_i \bigcup A_j \right). \tag{5}$$

Each proposal's score is weighted by $\alpha(r)$, a monotonically decreasing function. Because top-ranked proposals are given more weight, they are encouraged to have higher scores. We found the specific choice of $\alpha(r)$ is not particularly important, as long as it decreases to zero for moderate rank values. We use $\alpha(r) = \exp([(r-1)^2]/\sigma^2)$ with $\sigma = 100$.

3) Multiscale and Multiaspect Ratio Seeds: To address the problem of scale variation, a novel strategy for considering multiple scales and aspect ratios [29] is presented by the design of seeds. There are two popular methods for multiscale prediction. One is revising the images at multiple scales in a pyramid style, and then computing feature maps at each scale, e.g., [30] and [31]; this method is typically useful but is too time consuming. The other utilizes different sizes of pyramid filters [32], with several scales and aspect ratios run on the feature maps; this method is more cost-efficient.

We adopt the two methods jointly based on pyramid seeds; bounding boxes are classified and regressed concerning seed boxes with multiple aspect ratios and scales. The convolutional features computed on a single-scale image and a single-size sliding window can be simply used according to the multiscale seed design (Fig. 3), which is a key element for sharing features without excess cost for addressing scales.

B. Weakly Semi-Supervised Training

The number of parameters to learn is enormous; thus, DCNN requires a large amount of training data. However, there is just a small amount of available annotated training data; this lack of data is one of the bottlenecks in feasible DCNN training. To relieve the problem, a WSST method is proposed. In particular, a weakly supervised pairwise-ranking loss handles images that are weakly annotated, and a semi-supervised triplet-ranking loss is employed to harness unannotated images. In this paper, an image with a few tags

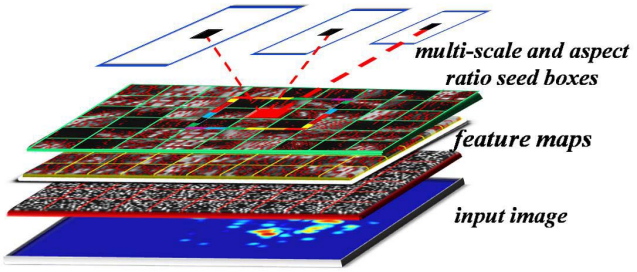


Fig. 3. Novel scheme for addressing multiple scales and sizes.

describing only part of the image content is used as a weakly annotated image, and an image with no tags at all is used as an unannotated image.

Assume we have a set of training images $I = \{x_i\}$. For the ith image x_i, we have a corresponding labeling vector $y_i \in \{0, 1\}^m$, where $y_i^j = 1$ indicates that the jth label of image x_i is "present" (positive), whereas $y_i^j = 0$ indicates that the label is "missing" (false negative). In other words, in this paper, x_i is not assumed to be fully annotated (therefore weakly annotated), where there may be labels that should be present but instead are unfortunately missing. In the setting of weak labeling, $y_i^j = 0$ denotes that image x_i does not have the jth concept at all. There may also be images in I that do not have any label information, i.e., $\sum y_i^j = 0$. In this case, we call x_i unannotated. We denote the training set I as two disjoint sets, weakly annotated images I_w and unannotated images I_u, i.e., $I = I_w \bigcup I_u$.

After obtaining the weakly annotated and unannotated images in the training set, we learn the prediction function $g(\cdot)$ that outputs the label score vector $a(x)$ of image x according to the learned features $f(x)$ via the convolutional neural network CNN$(\cdot)$. The learned features of CNN$(\cdot)$ and the score vector of $g(\cdot)$ are denoted as follows:

$$\text{CNN learned feature} : f(x) = \text{CNN}(x) \tag{6}$$

$$\text{Annotation score} : a(x) = g(f(x)) \tag{7}$$

where $f(x) \in R^p$ denotes the learned convolutional features, $g(\cdot)$ is the annotation score prediction function, $a(x) \in R^m$ is the label score vector, p is the dimension of the features, and m is the size of the label sets.

1) Weakly Supervised Training for Weakly Annotated Images: Assume that we are given one annotated image $x_i \in I_w$ and its labeling vector y_i. We tend to devise a ranking loss that assigns a higher score to positive labels than to negative labels, while considering the missing labels of x_i.

We denote the sets of indices of positive labels and negative labels of x_i as

$$C_{x_i}^+ = \left\{j | y_i^j = 1\right\} \tag{8}$$

$$C_{x_i}^- = \left\{j | y_i^j = 0\right\} \tag{9}$$

where $C_{x_i}^+$ and $C_{x_i}^-$ are the sets of indices of positive labels and negative labels of x_i, respectively.

Specifically, a weakly weighted pairwise-ranking loss is devised to optimize the top-k accuracy of image annotation for $x_i \in I_w$ as follows:

$$\min \sum_{x_i \in I_w} \sum_{s \in C_{x_i}^+} \sum_{t \in C_{x_i}^-} L_w(r_s) \max\left(0, m_s - a^s(x_i) + a^t(x_i)\right) \tag{10}$$

where $L_w(\cdot)$ is a weakly weighted pairwise-ranking loss function for different ranks of positive labels, r_s is the rank for the positive label s of x_i, a^s and a^t are the output scores for the positive label s and the negative label t, respectively, and m_s is the margin.

2) Semi-Supervised Training for Unannotated Images: In this section, we show how to exploit unannotated images in I_u for feature learning to enhance the performance of annotation. Traditionally, we calculate the semantic similarity $\text{sim}(x_i, x_j)$ of two images x_i and x_j according to their annotated tags y_i and y_j, respectively, with sim$(\cdot)$ defined as follows:

$$\text{sim}\left(x_i, x_j\right) = \sum_{s=1}^{m} \left(y_i^s \times y_j^s\right). \tag{11}$$

Therefore, we may directly constrain the features $f(x_i)$ and $f(x_j)$ to be similar with respect to their similarity as follows:

$$\min w\left(\text{sim}\left(x_i, x_j\right)\right) \left\| f(x_i) - f\left(x_j\right)\right\|_2^2. \tag{12}$$

As a result, we propose to utilize relative similarities between image triplets instead of directly relying on the pairwise similarity. Given one image triplet (x_i, x_j, x_k), where x_i and x_j are from I_w with overlapping positive labels [i.e., $\text{sim}(x_i, x_j) > 0$], and x_k is from I_w or I_u which is less similar to x_i than x_j, the relative similarity r sim$(\cdot)$ in terms of (x_i, x_j, x_k) is defined as follows:

$$r \text{ sim}\left(x_i, x_j, x_k\right) = \text{sim}\left(x_i, x_j\right) - \text{sim}(x_i, x_k). \tag{13}$$

Here, we expect the learned features of the images in a triplet to meet their relative semantic similarity defined by r sim. Therefore, we optimize the following objective:

$$\min \sum_{r \text{ sim}\left(x_i, x_j, x_k\right) > 0} \begin{aligned} &\max\Big(0, \left\| f(x_i) - f\left(x_j\right)\right\|_2^2 \\ &\quad - \left\| f(x_i) - f(x_k)\right\|_2^2 + m_f\Big) \end{aligned} \tag{14}$$

where m_f is the margin, $x_i, x_j \in I_w$ and $x_k \in I_u$. We call the objective the *triplet similarity* loss.

3) Weakly Semi-Supervised Learning: Thus, an objective function that achieves WSST can be expressed as follows:

$$\sum_{(x_i, x_j, x_k)} \begin{aligned} &\Big\{\textstyle\sum_{s \in C_{x_i}^+} \sum_{t \in C_{y_j}^-} L_w(r_s) \max\left(0, m_s - a^s(x_i) + a^t(x_i)\right) \\ &+ \alpha \max\Big(0, \left\| f(x_i) - f\left(x_j\right)\right\|_2^2 - \left\| f(x_i) - f(x_k)\right\|_2^2 + m_f\Big)\Big\} \end{aligned} \tag{15}$$

where r sim$(x_i, x_j, x_k) > 0$.

C. Sharable-Individual Mechanism and Relational-Regularized Joint Sparse Learning

1) Sharable-Individual Mechanism: Two domain-individual subnetworks $d_1(x)$ and $d_2(y)$ are applied to the samples of the two different modalities. Next, the outputs of $d_1(x)$ and $d_2(y)$ are concatenated into a shared subnetwork $s(\cdot)$. We make a superposition of $d_1(x)$ and $d_2(y)$ to feed

$s(\cdot)$. At the output of $s(\cdot)$, the features of the two samples are extracted separately as $s_1(x)$ and $s_2(y)$. In the following, we introduce the detailed setting of the subnetworks and the joint sparse learning strategy.

a) Domain-individual subnetwork: Two branches of neural networks are separated to handle the samples from different domains. Each branch includes one convolutional layer with three filters of size 5×5 and a stride step of two pixels. The rectified nonlinear activation is utilized. Next, a max-pooling operation is performed with size of 3×3 and stride step of three pixels.

b) Shared subnetwork: For this component, we stack one convolutional layer and two fully connected layers. The convolutional layer contains 32 filters of size 5×5 and the filter stride step is set as 1 pixel. The kernel size of the max-pooling operation is 3×3 and its stride step is 3 pixels. The output vectors of the two fully connected layers are of 400 dimensions. We further normalize the output of the second fully connected layer before it is fed to the next subnetwork.

2) Relational-Regularized Joint Sparse Learning: For the joint sparse learning strategy, a group of subspaces are learned to compare the inhomogeneous features with different dimensions, and then another group of subspaces are jointly mined with the same dimension to obtain the latent shared features from different channels, followed by their shared and individual components being encoded and learned. Each subspace corresponds to each type of inhomogeneous feature.

To achieve good translation invariance as well as improve the speed and accuracy of the whole system, RRJSL is proposed.

Define a local descriptor set $X = [x_1, x_2, \ldots, x_N] \in R^{d \times N}$ as an image; $B^{d \times K}$ is a dictionary that denotes the local descriptors, where$c_i$is the ith local descriptor column of X, and K is the dictionary's size. The sparse representations $\hat{Z}$ of a descriptor set can be expressed as follows:

$$\hat{Z} = \arg\min_{Z} \|X + D \odot C - BZ\|_F^2 + \gamma \|Z\|_{2,1}^2 + \varphi_1 R_1(Z) + \varphi_2 R_2(Z) \tag{16}$$

$$R_1(Z) = \sum_{u,v=1}^{F} \exp\left(-\left\|c^u - c^v\right\|_2^2\right) \left\|z^u - z^v\right\|_2^2 \tag{17}$$

$$R_2(Z) = \sum_{j,k=1}^{S} \exp\left(-\left\|r_j - r_k\right\|_2^2\right) \left\|r_j Z - r_k Z\right\|_2^2 \tag{18}$$

where c^u and c^v are the uth and vth column of input x, respectively, r_j and r_k are the jth and kth row of x, respectively, z^u and z^v are the uth and vth row of Z, respectively, and F and S denote the feature and subject, respectively. In particular, feature–subject associated context is integrated in a regularized discriminative least squares regression module with $l_{2,1}$-norm. $\hat{Z}$ is the sparse representations for $B^{d \times K}$ in X, $\|\cdot\|_F^2 \|\cdot\|_{2,1}^2$ are the matrix's F- and $l_{2,1}$-norm, respectively. D, γ are a nonnegative matrix and a regularization parameter, respectively, $\odot$ is a Hadamard product operator of matrices. R_1 and R_2 are a feature–feature and a subject–subject association-based regularization term, respectively. φ_1 and φ_2 are the regularization parameters of R_1 and R_2, respectively.

In each SGD iteration, the forward pass generates proposals that are treated as fixed, precomputed proposals. BP occurs as usual, where for the shared layers, the backward propagated signals from both the ARPN loss and the Inception-ResNet loss are combined. We first train ARPN, and use the proposals to train Inception-ResNet. The network tuned by Inception-ResNet is used to initialize ARPN, and then this process is iterated.

D. Loss Function

Use class labels to assign each seed's proposals as being an object or not. In particular, a positive label is assigned to a seed that has a top-ranked intersection-over-union (IoU) overlap ratio; generally, this approach is adequate to ensure positive samples. However, a situation should be considered in which another positive label is also assigned to the seed that has an IoU overlap ratio exceeding 0.75 using a ground-truth box in this paper. Both situations are considered to obtain any positive samples.

According to the loss in [33], an objective loss function $L(\{p_i\}, \{t_i\})$ is minimized using the concepts above and is described as

$$L(\{p_i\}, \{t_i\}) = (1/N_{\text{cls}}) \sum_i L_{\text{cls}}\left(P_i, P_i^*\right) + \lambda\left(1/N_{\text{reg}}\right) \sum_i P_i^* L_{\text{reg}}\left(t_i, t_i^*\right) \tag{19}$$

where i is a seed's index in a mini-batch, p_i denotes the forecasted probability of the ith seed being an object, and p_i^* denotes the ground-truth label of the ith seed. If the seed is positive, then $p_i^* = 1$; if it is negative, then $p_i^* = 0$. N_{cls} and N_{reg} are the size of the mini-batch (i.e., $N_{\text{cls}} = 256$) and the number of the seed's location (i.e., $N_{\text{reg}} = 2304$), respectively. N_{cls} and N_{reg} are used to normalize the cls and reg terms, respectively, in (20). λ is a balancing parameter to weight both terms. t_i and t_i^* denote the four coordinates of the predicting bounding and ground-truth box associated with a positive seed, respectively.

The classification loss L_{cls} is log loss over two classes being an object or not, and the regression loss $L_{\text{reg}}(t_i, t_i^*) = R(t_i - t_i^*)$, where R is a robust loss function

$$R\left(t_i - t_i^*\right) = \begin{cases} 0.5\left(t_i - t_i^*\right)^2, & \text{if } \left|t_i - t_i^*\right| < 1 \\ \left|t_i - t_i^*\right| - 0.5, & \text{if } \left|t_i - t_i^*\right| \geq 1 \end{cases} \tag{20}$$

where $p_i^* = 1$ indicates that the regression loss only for positive seeds is activated, and $p_i^* = 0$ indicates that the regression loss is disabled. The cls and reg layers' outputs include $\{p_i\}$ and $\{t_i\}$, respectively.

E. ARPN Training, Unsupervised Discriminative Learning, and Weakly Fine-Tuning

DFFC, BP, and stochastic gradient descent with momentum (MSGD) [33] are used to train ARPN end-to-end, as shown in Fig. 4. The dictionary is trained for local descriptors by minimizing the training error of the image-level features, which are extracted by max pooling over the sparse codes within a spatial pyramid [34]. The achieved dictionary is

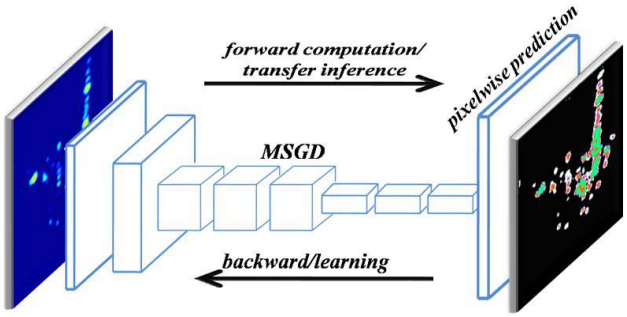


Fig. 4. DFFC, BP, and MSGD with ARPNs.

remarkably more effective than the unsupervised dictionary in terms of classification. Moreover, the max-pooling procedure over different spatial scales equips the proposed model with local translation-invariance [35]. All the new layers are randomly initialized by drawing weights from a zero-mean Gaussian distribution with a standard deviation of 0.01.

There are several steps in features computing and learning to detect objects. Additionally, these discriminative, invariant features are used to classify true and mendacious objects or different components of the same object. Region proposal patches are extracted from images randomly; this approach is required to transform the region's image data into a format that is compatible with DCNN and transform the unlabeled data into labeled. Next, a feature mapping is learned using an unsupervised learning algorithm. To achieve an excellent model, we train it by weakly fine-tuning the learned features.

F. Main Differences Between Faster R-CNN and DIOD

The framework of DIOD is similar to Faster R-CNN, but there are some critical differences between them.

1) Faster R-CNN [23] focuses on the problems of object localization and detection, while DIOD can simultaneously solve the problems of object localization, labeling, and detection in a unified framework conveniently and efficiently. DIOD not only works well for the fields owing a large number of authoritative data sets, such as PASCAL VOC [36] and ImageNet [37] but also works well for the fields of lacking available specific labeled images in practice. DIOD can be end-to-end both in training and testing stages more conveniently and efficiently with the help of WSST, SIM, and RRJSL.
2) There are four disjunct training steps and laborious fine-tuning processes to train and fine-tune Faster R-CNN in [23], while DIOD is trained end-to-end with WSST and weakly fine-tuning, which is more convenient and flexible than Faster R-CNN, and DIOD is easier to achieve overall optimal performance. Especially when there are few annotated and many unannotated images, DIOD is more generalized and robust than Faster R-CNN.
3) The sharing mechanism of Faster R-CNN only considers the shared convolutional layers in RPN and Fast R-CNN, while the SIM of DIOD not only learns shared convolutional features to retain their common properties but also learns an individual metric for each to keep its specific property.
4) In this paper, RRJSL is proposed for joint regression and classification via discriminative sparse learning and relational regularization with unsupervised transfer learning, which takes the complementary relationships among the homogenous and heterogeneous features, and improves the translation-invariance performance with complex patterns, automatically learns filter banks at different scales in a joint fashion with enforced scale-specificity. It not only improves the classification performance on object detection but also provides an unsupervised solution for transfer learning.
5) In [23], Faster R-CNN trains and tests both RPN and detection networks on images of a single scale. While in this paper, we randomly assign one of three scales for each image before it is fed into the network. The multiscale training scheme makes our model more robust toward different sizes, and improve the detection performance.
6) In this paper, we do not regress the box coordinates like OverFeat [38], MultiBox [39], and RPN [23], and instead decide the window which the pixel in the output corresponds to as a proposal. Combining the proposals classification and ranking strategy with the multiscale scheme obtains higher accuracy and faster speed than the box coordinates regression.

IV. Experiments

A. Real-World ISAR Datasets

Two real-world ISAR datasets are constructed for ISAR object detection, namely, ISAR-1 and ISAR-2, which consist of images containing more intraclass variations and multimodal conditions, which are more challenging than existing datasets.

Thin plate spline is used to interpolate sparse keypoints to generate ground truth. ISAR-1 consists of eight ISAR object classes with ten keypoint annotations for each image. ISAR-2 contains 15 classes with 12 keypoint annotations for each image. Given the images and regions, ground-truth data between all possible image pairs within each subclass are produced.

B. Implementations

The same training strategies are used in the whole system, i.e., pretrain and fine-tune. DIOD was evaluated in two ISAR datasets with Caffe [40], and there are 6000, 3000, and 2000 images in the training, validation, and testing sets, respectively. DIOD is trained end-to-end using FC, BP, and MSGD. The mean average precision (mAP) is selected as the evaluation metric. The whole model is trained on an 8-GPU implementation; the training process takes approximately 46.2 and 31.5 h on the ISAR-1 and ISAR-2 datasets, respectively. Extensive ablation studies are performed to validate the efficiency of our method. Fig. 5 shows the structure of Inception-ResNet. The ISAR object detection results of the

TABLE I
DETECTION RESULTS OF DIOD WITH DIFFERENT SETTINGS OF SEEDS (5 SCALES AND 5 ASPECT RATIOS)

Implementations	Aspect Ratios	Scales	mAP(%)	Runtime(s)
1 scale, 1 aspect ratio	1:1	64×64	45.5	0.437
		128×128	66.5	0.634
		256×256	67.4	1.351
1 scale, 3 aspect ratios	{1:2,1:1,2:1}	128×128	69.3	1.141
		256×256	68.5	1.342
1 scale, 5 aspect ratios	{1:3, 1:2, 1:1, 2:1, 3:1}	64×64	53.3	1.193
	{1:3, 1:2, 1:1, 2:1, 3:1}	128×128	59.8	1.313
	{1:3, 1:2, 1:1, 2:1, 3:1}	256×256	61.5	1.642
3 scales, 1 aspect ratio	1:1	{128×128, 256×256, 512×512}	71.2	0.618
5 scales, 1 aspect ratio	1:1	{32×32, 64×64, 128×128, 256×256, 512×512}	64.2	2.154
3 scales, 3 aspect ratios	{1:2,1:1,2:1}	{128×128, 256×256, 512×512}	71.4	0.192
		{64×64, 128×128, 256×256}	69.2	0.186
		{32×32, 64×64, 128×128}	61.9	0.164
		{64×64, 128×128, 512×512}	65.6	0.183
3 scale, 5 aspect ratios	{1:3, 1:2, 1:1, 2:1, 3:1}	{64×64, 128×128, 512×512}	72.7	2.612
5 scales, 3 aspect ratios	{1:2,1:1,2:1}	{32×32, 64×64, 128×128, 256×256, 512×512}	71.9	3.205
	{1:3,1:1,3:1}		68.1	3.056
	{1:3,1:2,1:1}		67.3	2.941
4 scale, 5 aspect ratios	{1:3, 1:2, 1:1, 2:1, 3:1}	{32×32, 64×64, 128×128, 256×256}	72.5	2.874
5 scale, 4 aspect ratios	{1:3, 1:2, 1:1, 2:1}	{32×32, 64×64, 128×128, 256×256, 512×512}	72.3	3.317
5 scales, 5 aspect ratios	{1:3, 1:2, 1:1, 2:1, 3:1}	{32×32, 64×64, 128×128, 256×256, 512×512}	72.8	3.615

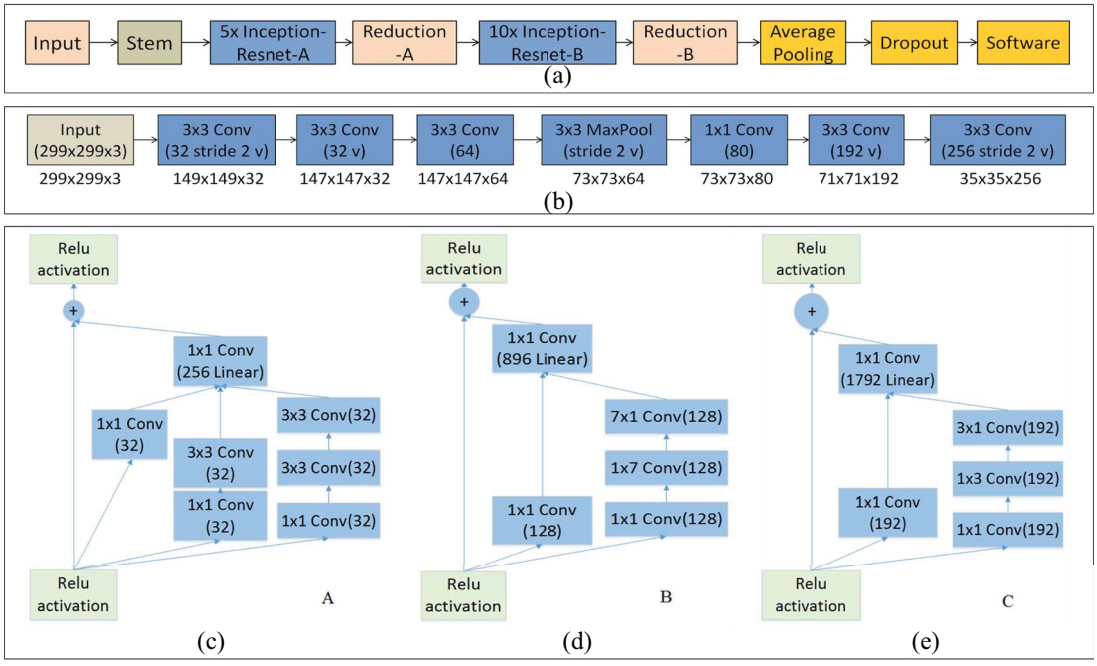


Fig. 5. Structure of Inception-Resnet. (a) Inception-Resnet. (b) Stem. (c) Inception-Resnet-A. (d) Inception-Resnet-B. (e) Inception-Resnet-C.

algorithm are shown in Fig. 6, along with the two final layers' feature maps of the ISAR objects.

C. Experiments on the Role of ARPN

1) Effects of Multiple Scales and Aspect Ratios: Table I investigates the effects of the strategy of multiple scales and aspect ratios. Table I shows that more scales and aspect ratios generally generates higher mAP, but not necessarily. And the more scales and aspect ratios used, the more time costs. The mAPs of 5 scales-5 aspect ratios (72.8%), and 5 scales-4 aspect ratios (72.3%), 4 scales-5 aspect ratios (72.5%), 5 scales-3 aspect ratios (71.9%), and 3 scales-5 aspect ratios (72.7%)

Fig. 6. ISAR object detection results of DIOD, and the two final layers' feature maps of the ISAR objects.

are higher than the mAP of 3 scales-3 aspect ratios (71.4%). But the speed of 3 scales-3 aspect ratios (with 0.192 s) runs much faster than all of them (with 3.615, 3.317, 2.874, 3.205, and 2.612 s, respectively). When using 3 scales with 1 aspect ratio or 1 scale with 3 aspect ratios, the mAP is higher than the mode of 1 scale with 1 aspect ratio, demonstrating that

TABLE II
DETECTION RESULTS ON ISAR TEST SET. REMOVING EITHER CLS OR RAK WITH DIFFERENT NUMBERS OF CANDIDATES, USING SS/EB CANDIDATE METHODS FOR TRAINING AND TESTING

Train-time region candidates method	Boxes	Test-time region proposals method	Proposals	mAP (%)
EdgeBoxes	2000	ARPN+Inception-ResNet	1000	58.9
EdgeBoxes	2000	ARPN+Inception-ResNet	400	59.5
EdgeBoxes	2000	ARPN+Inception-ResNet	100	57.2
Selective Search	2000	ARPN+Inception-ResNet	1000	58.3
Selective Search	2000	ARPN+Inception-ResNet	400	59.8
Selective Search	2000	ARPN+Inception-ResNet	100	58.1
EdgeBoxes	2000	ARPN+Inception-ResNet (without *cls*)	1000	56.6
EdgeBoxes	2000	ARPN+Inception-ResNet (without *cls*)	400	55.3
EdgeBoxes	2000	ARPN+Inception-ResNet (without *cls*)	100	43.8
Selective Search	2000	ARPN+Inception-ResNet (without *cls*)	1000	58.6
Selective Search	2000	ARPN+Inception-ResNet (without *cls*)	400	55.4
Selective Search	2000	ARPN+Inception-ResNet (without *cls*)	100	46.1
EdgeBoxes	2000	ARPN+Inception-ResNet (without *rak*)	1000	51.1
EdgeBoxes	2000	ARPN+Inception-ResNet (without *rak*)	400	50.6
EdgeBoxes	2000	ARPN+Inception-ResNet (without *rak*)	100	42.9
Selective Search	2000	ARPN+Inception-ResNet (without *rak*)	1000	53.8
Selective Search	2000	ARPN+Inception-ResNet (without *rak*)	400	54.1
Selective Search	2000	ARPN+Inception-ResNet (without *rak*)	100	41.3

TABLE III
DETECTION RESULTS ON ISAR TEST SET. ARPN WITH INCEPTION-RESNET AS THE DETECTOR BUT USING DIFFERENT PROPOSAL METHODS FOR TRAINING AND TESTING

Trained-time region proposals		Test-time region proposals					mAP (%)
Method	Boxes	Method	W	S	J	Proposals	
EdgeBoxes	2000	EdgeBoxes	No	No	No	2000	58.9
Selective Search	2000	Selective Search	No	No	No	2000	59.4
RPN (Faster)	2000	RPN (Faster)	No	No	No	400	59.6
RPN (Faster)	2000	RPN (Faster)	No	Yes	No	400	62.4
ARPN+Inception-ResNet	2000	ARPN+Inception-ResNe	No	No	No	400	59.9
ARPN+Inception-ResNet	2000	ARPN+Inception-ResNe	Yes	No	No	400	61.7
ARPN+Inception-ResNet	2000	ARPN+Inception-ResNe	No	Yes	No	400	63.6
ARPN+Inception-ResNet	2000	ARPN+Inception-ResNe	No	No	Yes	400	62.8
ARPN+Inception-ResNet	2000	ARPN+Inception-ResNe	Yes	Yes	No	400	66.5
ARPN+Inception-ResNet	2000	ARPN+Inception-ResNe	Yes	No	Yes	400	68.9
ARPN+Inception-ResNe	2000	ARPN+Inception-ResNe	No	Yes	Yes	400	67.8
ARPN+Inception-ResNe	2000	ARPN+Inception-ResNe	Yes	Yes	Yes	400	72.1

TABLE IV
PERFORMANCE COMPARISON OF SEVERAL STATE-OF-THE-ART METHODS

Method	Proposals	mAP (%)	mAP *vs.*	Runtime/s	Runtime *vs.*
OverFeat	2000	53.6	19.0 +	13.894 s	127.5×
OverFeat	400	45.3	27.3+	8.452 s	77.5×
RCNN	2000	59.1	13.5+	5.635 s	51.7×
SPP	-	58.5	14.1+	3.821 s	35.1×
Fast RCNN	1000	61.9	10.7+	3.278 s	30.1×
Faster R-CNN	400	65.8	6.8 +	1.693 s	15.5×
YOLO	-	63.1	9.5 +	1.914 s	17.6×
R-FCN	-	65.2	7.4 +	1.628 s	14.9×
Mask R-CNN	-	64.5	8.1+	1.269 s	11.6×
Ours	400	72.6	-	0.109 s	-

it is efficient to select multiple size seeds as regression reference boxes. Considering the tradeoff between accuracy and speed, 3 scales and 3 aspect ratios are used as default in this experiment.

2) Role of the Classification and Ranking Strategy: The effects of the classification and ranking strategy of ARPN are shown in Table II. In particular, cls and rak denote classification and ranking strategies, respectively. *N* proposals are achieved from the unscored regions when *cls* is removed; when we select SS in the train-time and *N* is 1000, 400, and 100, the mAP can be 58.6%, 55.4%, and 44.1%, respectively. When *rak* is removed, the mAPs descend to 53.8%, 54.1%, and 41.3% when *N* is 1000, 400, and 100, respectively.

In other words, an accurate detection system requires not only multiple aspect ratios and scales but also appropriate strategies to generate high-quality proposals. The mAPs of ARPN+Inception-ResNet with the classification and ranking strategies are found to be higher than the other modes.

3) Effects of Different Region Proposal Methods: Three state-of-the-art object proposal methods, including EB, SS, and RPN, are evaluated using the proposed ARPN. The results, which use Inception-ResNet, of trained and tested object detection with different region proposal approaches are shown in Table III. 2000, 2000, and 400 proposals are produced by SS, EB, and RPN, respectively. Under the fast mode, EB has an mAP of 58.9%, and SS has an mAP of 59.4%. Four hundred proposals are generated by the ARPN with Inception-ResNet, which achieves a competitive mAP of 72.1%.

D. Role of Weakly Semi-Supervised Training

For each image, we assign the k highest ranked labels to the image and compare these labels with the ground truth. When evaluating on ISAR-1, all methods are trained with random initialization with pretraining and use a linear mapping as the ranking score activation for DIOD. In Table III, "W," "S," and "J" denote WSST, sharing, and joint sparse learning, respectively. "Yes" and "No" denote using this item and not using this item, respectively. According to Table III, DIOD with WSST outperforms the other methods when considering both the weakly labeled and unlabeled images at the same time.

E. Role of Sharable-Individual Mechanism and Relational-Regularized Joint Sparse Learning

To examine the effects of the SIM and RRJSL strategy, after the second step, we pause and use several separate networks; this process achieves a smaller mAP of 67.8% (ARPN+Inception-ResNet, S&J, Table III). Because the quality of the region is high, the proposals are improved when the features are used to fine-tune the ARPN.

Next, the ARPN's influence on training the detection system is removed, and 2000 SS proposals and Inception-ResNet are used in the training. This detector is fixed, and the mAP of detection is evaluated by changing the proposal regions used. The ARPN does not share features with this detector in these experiments. When using 400 ARPN proposals, Inception-ResNet in test-time achieves an mAP of 61.7%, leading to some loss of mAP because of the inconsistency between the training/testing proposals.

F. Complexity Analysis

A serious problem to multiscale DCNN is the increase of trainable parameters and computational complexity, as more scales to be considered for the same input. There are four points making the detection system efficient.

1) A part of convolutional features of DIOD are shared by SIM and JSL. It is a good way for parameters and features sharing to boost the scale-invariant features learning ability, and simultaneously cut down the probability of over-fitting and the quantity of the trainable parameters.
2) There is lower complexity, such as lower-dimensional feature vectors in DIOD, while having comparable accuracy with other methods. For example, in OverFeat, the complexity of the convolutional feature computation is $O(n \cdot 227^2)$ with the window number n (around 2000), while the complexity of DIOD is $O(r \cdot s^2)$ with aspect ratio r, and scale s. Supposed the aspect ratios r is 2:1 (1:2), s is 512 (28) in the single-scale style, the complexity of DIOD is around 1/200 of OverFeat at best.
3) The image pyramid strategy, for example, for smaller sizes using down-sampling strategy, but for larger sizes not up-sampling filters. Furthermore, pooling operation among adjacent scales and positions is an effective method to achieve invariance and reduce model complexity.
4) With the help of WSST, SIM, and JSL, it takes fewer region proposal and feature computing time, and the detection system further gets more efficient performance.

G. Final Comparison With the State-of-the-Art Methods

Table IV shows the experimental results of the proposed method with several state-of-the-art methods on the ISAR-2 dataset, including OverFeat, RCNN [41], SPP [42], Fast RCNN [43], Faster RCNN [23], YOLO [44], R-FCN [45], and Mask R-CNN [46]. The proposed approach was pretrained with standard convolution and relational-regularization.

The detailed execution times of the presented method and the other eight state-of-the-art methods above are shown in Table IV. The overall execution time includes image resizing, network forwarding, and post-processing.

Considering the tradeoff between accuracy and speed, the proposed method achieves the best performance. Our method outperforms the Faster R-CNN, YOLO, R-FCN, and Mask R-CNN methods by approximately 6.8%, 9.5%, 7.4%, and 8.1% in mAP, respectively. Compared with these methods, the presented method executes much faster, except for Mask R-CNN and R-FCN (with a small cost increase), i.e., the proposed method can be used in additional ordinary equipment and wider application scenarios.

In the experiments, we find show that implementing some factors appropriately, which are uncomplicated but important and easy to be ignored, are possible so crucial for the success of feature learning methods in reality, and more significant than feature learning method and the depth of model's selection themselves, which have been studied a lot.

1) Seeds of inappropriate scales and aspect ratios are associated with few examples, so are noisy and harmful for detection accuracy. It is difficult to find an appropriate bounding box to cover each object or all the parts of the same object.
2) Because of subsampling and pooling operations in CNN, the resolutions of feature maps are insufficient, which is an important factor that influences the ability of

object proposal methods in finding small objects. The resolution of feature maps should be improved.

3) Some simple training schemes may produce different detection performance, including the robustness, accuracy, and speed.
4) The performance of localization, labeling, and detection can be improved mutually by incorporating the localization, labeling, and detection in a unified framework.
5) Appropriate preprocessing and post-processing also are important and helpful for deep learning framework.

V. Conclusion

We proposed a fast and efficient weakly semi-supervised ISAR object detection method based on ARPN and WSSDJSL. ARPN was proposed to generate high-level region proposals and localize potential ISAR objects robustly and accurately in less time; ARPN shares common convolutional features with Inception-ResNet and offers almost cost-free proposal computation with excellent performance. A WSST method was adopted to solve the problem of the lack of sufficient labeled training data, thereby achieving superior performance over the traditional unsupervised strategy. Moreover, a novel SIM and joint sparse learning method were proposed to improve the accuracy and speed of the whole system. Compared with the state-of-the-art methods, DIOD can achieve increased outstanding accuracy while executing significantly faster.

References

[1] S.-J. Lee, M.-J. Lee, J.-H. Bae, and K.-T. Kim, "Classification of ISAR images using variable cross-range resolutions," *IEEE Trans. Aerosp. Electron. Syst.*, to be published, doi: 10.1109/TAES.2018.2814211.

[2] F. Colone, D. Pastina, and V. Marongiu, "VHF cross-range profiling of aerial targets via passive ISAR: Signal processing schemes and experimental results," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 53, no. 1, pp. 218–235, Feb. 2017, doi: 10.1109/TAES.2017.2649999.

[3] M. Martorella, E. Giusti, A. Capria, F. Berizzi, and B. Bates, "Automatic target recognition by means of polarimetric ISAR images and neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3786–3794, Nov. 2009, doi: 10.1109/TGRS.2009.2025371.

[4] M. Martorella *et al.*, "Target recognition by means of polarimetric ISAR images," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 1, pp. 225–239, Jan. 2011, doi: 10.1109/TAES.2011.5705672.

[5] Z. Peng *et al.*, "A portable FMCW interferometry radar with programmable low-if architecture for localization, ISAR imaging, and vital sign tracking," *IEEE Trans. Microw. Theory Techn.*, vol. 65, no. 4, pp. 1334–1344, Apr. 2017, doi: 10.1109/TMTT.2016.2633352.

[6] M. Khare, R. K. Srivastava, and A. Khare, "Single change detection-based moving object segmentation by using Daubechies complex wavelet transform," *IET Image Process.*, vol. 8, no. 6, pp. 334–344, Jun. 2014, doi: 10.1049/iet-ipr.2012.0428.

[7] M. Casares and S. Velipasalar, "Adaptive methodologies for energy-efficient object detection and tracking with battery-powered embedded smart cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1438–1452, Oct. 2011, doi: 10.1109/TCSVT.2011.2162762.

[8] B. Kalantar, S. B. Mansor, A. A. Halin, H. Z. M. Shafri, and M. Zand, "Multiple moving object detection from UAV videos using trajectories of matched regional adjacency graphs," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 9, pp. 5198–5213, Sep. 2017, doi: 10.1109/TGRS.2017.2703621.

[9] X. Wang, H. Ma, X. Chen, and S. You, "Edge preserving and multi-scale contextual neural network for salient object detection," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 121–134, Jan. 2018, doi: 10.1109/TIP.2017.2756825.

[10] A. Omid-Zohoor, C. Young, D. Ta, and B. Murmann, "Towards always-on mobile object detection: Energy versus performance tradeoffs for embedded HOG feature extraction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 5, pp. 1102–1115, May 2018, doi: 10.1109/TCSVT.2017.2653187.

[11] G. Wang, X. Wang, B. Fan, and C. Pan, "Feature extraction by rotation-invariant matrix representation for object detection in aerial image," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 6, pp. 851–855, Jun. 2017, doi: 10.1109/LGRS.2017.2683495.

[12] H.-F. Yang, K. Lin, and C.-S. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 437–451, Feb. 2018, doi: 10.1109/TPAMI.2017.2666812.

[13] C. Yang *et al.*, "Neural networks enhanced adaptive admittance control of optimized robot-environment interaction," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2018.2828654.

[14] H. Yang, X. He, X. Jia, and I. Patras, "Robust face alignment under occlusion via regional predictive power estimation," *IEEE Trans. Image Process.*, vol. 24, no. 8, pp. 2393–2403, Aug. 2015, doi: 10.1109/TIP.2015.2421438.

[15] G. Cheng, P. Zhou, and J. Han, "Learning rotation-invariant convolutional neural networks for object detection in VHR optical remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016, doi: 10.1109/TGRS.2016.2601622.

[16] J. Pont-Tuset, P. Arbeláez, J. T. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping for image segmentation and object proposal generation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 128–140, Jan. 2017, doi: 10.1109/TPAMI.2016.2537320.

[17] D. Sarikaya, J. J. Corso, and K. A. Guru, "Detection and localization of robotic tools in robot-assisted surgery videos using deep neural networks for region proposal and detection," *IEEE Trans. Med. Imag.*, vol. 36, no. 7, pp. 1542–1549, Jul. 2017, doi: 10.1109/TMI.2017.2665671.

[18] Z. Zhang and P. H. S. Torr, "Object proposal generation using two-stage cascade SVMs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 102–115, Jan. 2016, doi: 10.1109/TPAMI.2015.2430348.

[19] S. Huang, W. Wang, S. He, and R. W. H. Lau, "Stereo object proposals," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 671–683, Feb. 2017, doi: 10.1109/TIP.2016.2627819.

[20] Z. Zhang *et al.*, "Sequential optimization for efficient high-quality object proposal generation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1209–1223, May 2018, doi: 10.1109/TPAMI.2017.2707492.

[21] J. R. R. Uijlings, K. E. A. V. D. Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Sep. 2013, doi: 10.1007/s11263-013-0620-5.

[22] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. ECCV*, Zürich, Switzerland, 2014, pp. 391–405.

[23] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 128–140, Jun. 2017, doi: 10.1109/TPAMI.2016.2577031.

[24] A. Dosovitskiy, P. Fischer, J. T. Springenberg, M. Riedmiller, and T. Brox, "Discriminative unsupervised feature learning with exemplar convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1734–1747, Sep. 2016, doi: 10.1109/TPAMI.2015.2496141.

[25] S. Petridis and M. Pantic, "Deep complementary bottleneck features for visual speech recognition," in *Proc. ICASSP*, Shanghai, China, 2016, pp. 2304–2308.

[26] Z. Cui, S. S. Ge, Z. Cao, J. Yang, and H. Ren, "Analysis of different sparsity methods in constrained RBM for sparse representation in cognitive robotic perception," *J. Intell. Robot. Syst.*, vol. 80, no. 1, pp. 121–132, Feb. 2015, doi: 10.1007/s10846-015-0213-3.

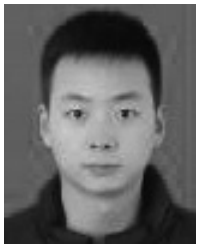
[27] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017, doi: 10.1109/TPAMI.2016.2572683.

[28] C. Li, L. Lin, W. Zuo, W. Wang, and J. Tang, "An approach to streaming video segmentation with sub-optimal low-rank decomposition," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 1947–1960, May 2016, doi: 10.1109/TIP.2016.2537211.

[29] J. Yan *et al.*, "Forecasting the high penetration of wind power on multiple scales using multi-to-multi mapping," *IEEE Trans. Power Syst.*, vol. 33, no. 3, pp. 3276–3284, May 2018, doi: 10.1109/TPWRS.2017.2787667.

[30] F. S. Khan, J. V. D. Weijer, R. M. Anwer, M. Felsberg, and C. Gatta, "Semantic pyramids for gender and action recognition," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3633–3645, Aug. 2014, doi: 10.1109/TIP.2014.2331759.

[31] L. Seidenari, G. Serra, A. D. Bagdanov, and A. D. Bimbo, "Local pyramidal descriptors for image recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 5, pp. 1033–1040, May 2014, doi: 10.1109/TPAMI.2013.232.
[32] S. Wang *et al.*, "Hydrogen bonding to carbonyl oxygen of nitrogen-pyramidalized amide—Detection of pyramidalization direction preference by vibrational circular dichroism spectroscopy," *Chem. Commun.*, vol. 52, no. 21, pp. 4018–4021, Feb. 2016, doi: 10.1039/C6CC00284F.
[33] K. Cohen, A. Nedić, and R. Srikant, "On projected stochastic gradient descent algorithm with weighted averaging for least squares regression," *IEEE Trans. Autom. Control*, vol. 62, no. 11, pp. 5974–5981, Nov. 2017, doi: 10.1109/TAC.2017.2705559.
[34] P. Hu, G. Wang, and Y.-P. Tan, "Recurrent spatial pyramid CNN for optical flow estimation," *IEEE Trans. Autom. Control*, to be published, doi: 10.1109/TMM.2018.2815784.
[35] F. Baumann, S. B. Dutta, and M. Henkel, "Kinetics of the long-range spherical model," *J. Phys. A Math. Theor.*, vol. 40, no. 27, pp. 7389–7409, Apr. 2007, doi: 10.1088/1751-8113/40/27/001.
[36] M. Everingham *et al.*, "The Pascal, visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jun. 2015, doi: 10.1007/s11263-014-0733-5.
[37] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, May 2014.
[38] P. Sermanet *et al.*, "OverFeat: Integrated recognition, localization and detection using convolutional networks," *eprint arxiv.*, pp. 1–16, 2013.
[39] W. Liu *et al.*, "SSD: Single shot multibox detector," in *Proc. ECCV*, Amsterdam, The Netherlands, 2015, pp. 21–37.
[40] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, Orlando, FL, USA, 2014, pp. 675–678.
[41] 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, doi: 10.1109/TPAMI.2015.2437384.
[42] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015, doi: 10.1109/TPAMI.2015.2389824.
[43] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, Santiago, Chile, 2015, pp. 1440–1448.
[44] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. CVPR*, Las Vegas, NV, USA, 2016, pp. 779–788.
[45] J. Dai, Y. Li, and K. He, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. NIPS*, Barcelona, Spain, 2016, pp. 1–11.
[46] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published, doi: 10.1109/TPAMI.2018.2844175.



Bin Xue (S'15) was born in Shangluo, China, in 1990. He received the B.S. degree in computer science and technology from the Zhongnan University of Economics and Law, Wuhan, China, in 2013 and the M.S. degree in information and communication engineering from Airforce Engineering University, Xi'an, China, in 2015, where he is currently pursuing the Ph.D. degree in electronic science and technology.

His current research interests include ISAR object recognition, deep learning, time series prediction, modern statistic analysis, and machine learning.

Ningning Tong received the B.S., M.S., and Ph.D. degrees from Air Force Engineering University, Xi'an, China, in 1984, 1988, and 2009, respectively.

She is currently a Professor with Air Force Engineering University. She has authored or co-authored over 60 research papers and four books. Her current research interests include wireless communications, radar signal processing, and electronic countermeasures.