

Article

SAR Target Recognition via Supervised Discriminative Dictionary Learning and Sparse Representation of the SAR-HOG Feature

Shengli Song ^{1,2,*}, Bin Xu ¹ and Jian Yang ¹

¹ Department of Electronic Engineering, Tsinghua University, Beijing 100084, China; xubin07161@gmail.com (B.X.); yangjian_ee@mail.tsinghua.edu.cn (J.Y.)

² Luoyang Electronic Equipment Test Center, Luoyang 471000, China

* Correspondence: ssl_1980@126.com; Tel.: +86-10-6279-4726

Academic Editors: Xiaofeng Li and Prasad S. Thenkabail

Received: 25 May 2016; Accepted: 17 August 2016; Published: 20 August 2016

Abstract: Automatic target recognition (ATR) in synthetic aperture radar (SAR) images plays an important role in both national defense and civil applications. Although many methods have been proposed, SAR ATR is still very challenging due to the complex application environment. Feature extraction and classification are key points in SAR ATR. In this paper, we first design a novel feature, which is a histogram of oriented gradients (HOG)-like feature for SAR ATR (called SAR-HOG). Then, we propose a supervised discriminative dictionary learning (SDDL) method to learn a discriminative dictionary for SAR ATR and propose a strategy to simplify the optimization problem. Finally, we propose a SAR ATR classifier based on SDDL and sparse representation (called SDDL SR), in which both the reconstruction error and the classification error are considered. Extensive experiments are performed on the MSTAR database under standard operating conditions and extended operating conditions. The experimental results show that SAR-HOG can reliably capture the structures of targets in SAR images, and SDDL can further capture subtle differences among the different classes. By virtue of the SAR-HOG feature and SDDL SR, the proposed method achieves the state-of-the-art performance on MSTAR database. Especially for the extended operating conditions (EOC) scenario “Training 17°—Testing 45°”, the proposed method improves remarkably with respect to the previous works.

Keywords: synthetic aperture radar; target recognition; dictionary learning; sparse representation; histogram of oriented gradients; MSTAR

1. Introduction

Automatic target recognition (ATR) is one of the important applications of synthetic aperture radar (SAR) in civilian and military fields. Although many works have been done in the past few decades [1–8], it is still a highly challenging problem. Generally, the process of SAR ATR includes four sequential stages: detection, discrimination, feature extraction and classification. In the first two stages, the potential regions of interest (ROIs) are located, and the false ROIs are removed. In the last two stages, distinctive features are extracted from ROIs, and then, the extracted features are classified. In this paper, feature extraction and classification are studied.

For the good performance of SAR ATR, feature extraction is a key factor, the aim of which is to capture the characteristic of targets. In high resolution SAR images, geometric structures are the important features of targets. Inspired by the simplicity and robustness of a local (statistical) feature, i.e., histogram of oriented gradients (HOG) [9], we adapt HOG to speckled SAR images by using the ratio-based gradient definition (called SAR-HOG). Experiments show that SAR-HOG can

indeed depict the geometric property of targets. In addition to obtaining the distinctive features of targets, classification is used to capture the differences among the different classes. Inspired by the effectiveness of the sparse representation classifier (SRC) [10] and dictionary learning [11], we further propose the supervised discriminative dictionary learning (SDDL) to learn a discriminative dictionary by combining the merits of supervised dictionary learning (SDL) [12] and discriminative dictionary learning (DDL) [13]. In addition, in order to amplify the differences while properly suppressing the common features of the different classes, we represent the learned dictionary as the concentration of the class-specific sub-dictionaries and a shared sub-dictionary. Meanwhile, we propose a new strategy to combine the dictionary and the parameter of the classifier to simplify the optimization procedure. Based on the proposed SDDL and SRC, we propose the SAR ATR classifier, i.e., supervised discriminative dictionary learning and the sparse representation classifier (SDDLSC). Finally, SAR-HOG and SDDLSC are applied to the public MSTAR database.

The main contributions of this paper can be summarized as follows: (1) We propose a novel local feature for SAR ATR named SAR-HOG, which can effectively capture the main structures of targets in speckled SAR images. (2) We propose SDDL to learn the discriminative dictionary for SAR ATR. This is the first work to combine the SDL and DDL. Meanwhile, we propose a new strategy to simplify the optimization problem. (3) We propose a SAR ATR classifier SDDLSC based on SDDL and SRC, in which both the reconstruction error and the classification error are considered. (4) We perform extensive experiments to demonstrate the performance of the proposed method on the MSTAR database under the standard operating conditions (SOC) and extended operating conditions (EOC) scenarios. In addition, the performance of the proposed method with respect to the reduced size of the training set is also evaluated. Experiments show that our method achieves the state-of-the-art SAR ATR performance.

The remainder of the paper is organized as follows. In Section 2, we briefly introduce the related work, including the feature extraction methods and classification methods for SAR ATR. The novel local feature SAR-HOG is detailed in Section 3. The proposed supervised discriminative dictionary learning and the whole SAR ATR algorithm are detailed in Section 4. Experimental results on the MSTAR database are given in Section 5. Section 6 finally concludes our work.

2. Related Work

2.1. Work Related to Feature Extraction for SAR ATR

The present feature extraction methods for SAR images can be roughly grouped into three categories. (1) Using raw images or transformed images: Zhao et al. [1] used the raw images as the feature of targets. Srinivas et al. [2] adopted the wavelet decomposition images. Dong et al. [3] utilized the monogenic signal to capture the characteristics of the SAR image. Generally, the features represented by raw images or transformed images have a high dimension; thus, linear or nonlinear dimensionality reduction techniques are usually used. Mishra [4] compared the linear methods, i.e., principle component analysis (PCA) and linear discriminant analysis (LDA), for SAR ATR. Huang et al. [5] proposed a nonlinear method using tensor global and local discriminant embedding for dimensionality reduction in SAR ATR. (2) Using scattering center features: Zhou et al. [6] adopted scattering center features at different target poses for classification. However, an offline global scattering model is needed to establish SAR image templates. (3) Using global or local statistical features: Clemente et al. [7] used the pseudo-Zernike moments as the global statistical features for SAR ATR. Local (statistical) features arise from the small parts of an image, e.g., HOG [9], scale-invariant feature transform (SIFT) [14], etc. The local features are usually invariant to image rotation, image scaling and minor changes in viewing direction [15,16]. For optical images, local feature extraction methods usually detect key points at first and then compute the local descriptors [15,16]. However, there are two problems when applying the existing local feature extraction methods to SAR ATR. On the one hand, for limited resolution SAR images, the pixels of targets are usually limited. Additionally,

key point detectors cannot always reliably capture the structures of targets [9]. On the other hand, the gradient computation by difference in local feature extraction is not suitable for the SAR image due to the speckle noise. By far, only a few local features are investigated for SAR images. Dai et al. [17] adopted the multilevel local pattern histogram for SAR terrain and land-use classification. Cui et al. [18] proposed a ratio-detector-based feature extraction method for very high resolution SAR image patch indexing. Dellinger et al. [19] proposed a SIFT-like algorithm called SAR-SIFT for the registration of SAR images. These investigations indicate that the local features have great potential in SAR applications. Some reviews of local feature extraction can be found in [15,16]. Apart from the above hand-designed feature extraction methods, convolutional neural networks (CNNs) have been used for automatic feature extraction for SAR ATR [8]. However, CNNs need huge training samples to gain the desired performance. In short, feature extraction for SAR images is still an open-ended question.

2.2. Work Related to Classification for SAR ATR

Classification in SAR ATR normally refers to supervised classification. Many classic classifiers have been used in SAR ATR, such as SVM [1], kNN [3], etc. The earlier mentioned dimensionality reduction methods PCA and LDA can also be taken as classifiers [4]. Recently, the sparse representation classifier proposed by Wright [10] has been successfully applied to SAR applications, such as polarimetric SAR image classification [20] and SAR ATR [3]. The success of SRC is largely guaranteed by the high redundancy and low coherency of the dictionary atoms. However, the dictionary in SRC constructed by stacking training samples may neither be optimal for solving reconstruction problems (i.e., denoising, inpainting) nor for solving discriminative problems (i.e., classification) [21]. Thus, dictionary learning [11] is used to learn a more representative and compact dictionary. Several dictionary learning methods have been proposed for reconstruction tasks, such as K-SVD [22] and online dictionary learning [23]. However, these methods are not necessarily suitable for discriminative tasks [24]. Therefore, discriminative dictionary learning methods are proposed for solving discriminative problems. Ramirez et al. [13] proposed an incoherent dictionary learning method by introducing an incoherence term to encourage the independency of sub-dictionaries corresponding to different classes; thus, the dictionary becomes more discriminative. Gao et al. [25] further developed this method. They explicitly represented the similar atoms existing in sub-dictionaries as a shared sub-dictionary. Meanwhile, they also imposed incoherence constraints among class-specific sub-dictionaries. Although DDL can efficiently improve the performance of the dictionary, it does not use the class labels in the training set. Mairal et al. [12] proposed supervised dictionary learning by using class labels to improve the classification performance. Zhang et al. [26] proposed a method called discriminative K-SVD (DK-SVD) to jointly learn the dictionary and classifier by minimizing the summation of the reconstruction and classification errors. Inspired by DK-SVD, Jiang et al. [27] further proposed a method called label consistent K-SVD (LCK-SVD). They enforced a label consistency constraint on the dictionary and combined the dictionary and the parameters of the classifier into a single parameter to make the optimization more simple. A survey of supervised dictionary learning and sparse representation can be found in [28].

3. SAR-HOG

Because of the much greater aspect sensitivity of the SAR image compared with the optical image, the SAR image is more sensitive to depression angle change and the pose variation of targets [29]. Therefore, we should pay more attention to the stable SAR image pixels, which are mainly dominated by strong backscatter returns from structures with aspect insensitivity. HOG is a simple and efficient local feature. It suggests that, for capturing stable structures of targets, one should use fine-scale derivatives, fine orientation binning, a dense grid and high-quality normalized descriptor blocks [9]. Inspired by this idea, we propose a HOG-like local feature for SAR images called SAR-HOG by using the ratio-based gradient definition.

SAR-HOG computation includes three steps: gradient computation, orientation binning, normalization and feature description [9]. In the following, we detail the computation.

3.1. Gradient Computation

Original HOG uses the simplest scheme to compute the gradients, i.e., 1-D $[-1, 0, 1]$ masks are employed to compute the gradients of the raw image without smoothing [9]. Because of the multiplicative speckle noise existing in SAR images, the gradient by difference is not a constant false alarm rate operator. It is more suitable to use the ratio instead of the difference to compute the gradients of SAR images [19]. Here, we use the simplest ratio of average (ROA) [30], i.e.,

$$R_i = \frac{M_1(i)}{M_2(i)} \quad (1)$$

where R_i denotes the ratio and $M_1(i)$ and $M_2(i)$ denote the local means on opposite sides of the current pixel along direction i . $i = 1$ means the horizontal direction, and $i = 3$ means the vertical direction. Figure 1 shows the scheme of the ROA. The average region for $M_1(i)$ calculation is $\max[\frac{win-1}{2}, 1] \times win$ (or $win \times \max[\frac{win-1}{2}, 1]$), where win denotes the odd size of the average region.

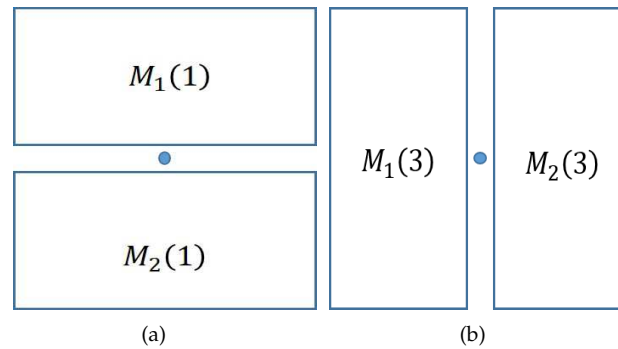


Figure 1. Scheme of the ratio of average (ROA). (a) The ratio of the local means for the horizontal direction. (b) The ratio of the local means for the vertical direction.

Inspired by SAR-SIFT [19], we define the horizontal gradient G_H and vertical gradient G_V as:

$$G_H = \log(R_1), \quad G_V = \log(R_3) \quad (2)$$

Then, the gradient magnitude G_m and orientation G_θ can be computed by:

$$G_m = \sqrt{G_H^2 + G_V^2}, \quad G_\theta = \text{atan}\left(\frac{G_V}{G_H}\right) \quad (3)$$

where the $\text{atan}(\cdot)$ denotes the inverse tangent function.

It should be noted that original HOG computes gradients without smoothing, and this is equal to computing $M_1(i)$ and $M_2(i)$ using only one pixel, respectively, i.e., $win = 1$. However, the effect of doing so in practice is not good for SAR images. Additionally, the main reason depends on speckle noise. We find that the moderately-sized region for averaging can improve the performance.

3.2. Orientation Binning

Given G_m and G_θ , we calculate the histogram of oriented gradients in local spatial regions (called cells) like that in [9]. Specifically, the SAR image is divided into small cells (see Figure 2). For all of the pixels within a cell, the orientations are quantized into a fixed number of angular bins, and the magnitudes are accumulated into orientation bins. In original HOG, 6–8 pixel-wide cells and nine

angular bins do best [9]. However, we find that smaller cells work better, and the angular bins should be a bit bigger.

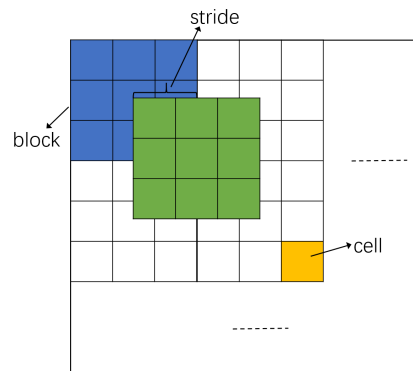


Figure 2. Scheme of SAR-HOG.

3.3. Normalization and Feature Description

Generally, G_m of the SAR image has a large dynamic range, especially for metallic targets in SAR ATR. Therefore, effective local contrast normalization is very important for good performance. Specifically, the cells are grouped into larger spatially-connected blocks (see Figure 2), and the histogram entries of cells in each block are concentrated to be a vector. Then, each vector is normalized to have unit l_2 norm, i.e.,

$$\mathbf{v}_i \leftarrow \frac{\mathbf{v}_i}{\max(\|\mathbf{v}_i\|_2, \varepsilon)} \quad (4)$$

where \mathbf{v}_i denotes the vector corresponding to the i -th block; ε is a small number, which can be chosen to be 0.2-times the mean value of $\|\mathbf{v}_i\|_2$ in all of the blocks [31]. The normalization method here is a little different from that in [9], which can better avoid the poor results for blocks with uniform G_m . In the original HOG; 2–3 cell blocks work best [9]. In Section 5, we can see that block size should be adjusted according to the targets in SAR images.

Finally, all of the normalized vectors corresponding to the blocks are concentrated to yield the SAR-HOG descriptor. It should be noted that, in order to improve the performance, the blocks typically overlap, meaning that each cell contributes more than once to the final descriptor (see Figure 2). In the original HOG, the block overlap (stride) is half of the block size. We find that it is still suitable for the SAR image.

In Section 5, we can see that SAR-HOG can reliably capture the structures of targets in SAR images, and this local feature can directly improve SAR ATR performance.

4. SAR ATR Algorithm via Supervised Discriminative Dictionary Learning of SAR-HOG

In some SAR ATR scenarios, the targets from different classes may have similar physical structures. For example, in military vehicles, SAR ATR [29], the main battle tanks T62 and T72 have similar structures, e.g., tread, turret, armor, etc. Additionally, these similar objects correspond to similar image features. In this paper, we propose the SDDL method to further capture subtle differences among the different classes. In the following, we first review the sparse representation classifier and dictionary learning, then detail the proposed SDDL method and the optimization procedure and, finally, present the complete SAR ATR algorithm.

4.1. Review of the Sparse Representation Classifier and Dictionary Learning

SRC is a generic classifier and has been used in many applications. Given sufficient training samples of K classes, a dictionary $\mathbf{D} = [\mathbf{D}_1 \cdots \mathbf{D}_K] \in \mathbb{R}^{m \times N}$ is constructed by stacking the samples, where $\mathbf{D}_k = [\mathbf{x}_1^k, \cdots, \mathbf{x}_i^k, \cdots, \mathbf{x}_{n_k}^k]$ is the k -th sub-dictionary and \mathbf{x}_i^k represents the i -th training sample

with m dimension in the k -th class, $i \in \{1, \dots, n_k\}$ and $N = \sum_{k=1}^K n_k$. A test sample \mathbf{x}_t is decomposed onto \mathbf{D} by solving the following optimization problem [10]:

$$\min_{\boldsymbol{\alpha}_t \in \mathbb{R}^N} \|\mathbf{x}_t - \mathbf{D}\boldsymbol{\alpha}_t\|_2^2 + \lambda \|\boldsymbol{\alpha}_t\|_1 \quad (5)$$

where $\boldsymbol{\alpha}_t$ is the sparse code of \mathbf{x}_t and λ is the regularization parameter to control the sparse degree of $\boldsymbol{\alpha}_t$; $\|\cdot\|_2$ and $\|\cdot\|_1$ denote the l_2 norm and l_1 norm, respectively.

Then, \mathbf{x}_t is approximated by quantities $\{\hat{\mathbf{x}}_t^k = \mathbf{D}\boldsymbol{\delta}_k(\boldsymbol{\alpha}_t)\}$, where $\boldsymbol{\delta}_k(\boldsymbol{\alpha}_t)$ is a vector whose nonzero entries are the entries in $\boldsymbol{\alpha}_t$ that are associated with the k -th class. Additionally, \mathbf{x}_t is classified by assigning it to the class that minimizes the residual error between \mathbf{x}_t and $\hat{\mathbf{x}}_t^k$ [10]:

$$\hat{k} = \arg \min_{k \in \{1, \dots, K\}} \|\mathbf{x}_t - \hat{\mathbf{x}}_t^k\|_2^2 \quad (6)$$

In order to have a representative and compact dictionary, dictionary learning is applied to learn a dictionary from the training set, which can be formulated as follows [11]:

$$\min_{\mathbf{D}, \{\mathbf{A}_k\}} \sum_{k=1}^K \frac{1}{2} \|\mathbf{X}_k - \mathbf{D}\mathbf{A}_k\|_F^2 + \lambda \psi(\mathbf{A}_k) \quad \text{s.t.} \quad \|\mathbf{D}_k(:, j)\|_2 = 1, \forall k, j \quad (7)$$

where $\mathbf{D} \in \mathbb{R}^{m \times P}$ is the learned dictionary with size of P ; $\mathbf{X}_k \in \mathbb{R}^{m \times n_k}$ and $\mathbf{A}_k \in \mathbb{R}^{P \times n_k}$ denote the k -th test samples matrix and the corresponding sparse codes matrix, respectively; $\psi(\cdot)$ is the sparsity-inducing regularization function, and the l_1 norm is usually used; $\|\cdot\|_F$ denotes the Frobenius norm. To avoid the trivial solution, we constrain each atom to be norm one. Generally, the size of the dictionary is much smaller than the amount of the samples, i.e., $P \ll N$, meaning that the learned dictionary is compact.

The above dictionary learning method is usually used for reconstruction tasks. In fact, for discriminative tasks [24], e.g., SAR ATR, DDL can be used to learn a more discriminative dictionary. In [25], class-specific sub-dictionaries and a shared sub-dictionary are jointly learned from the training samples. Meanwhile, incoherence constraints are enforced on sub-dictionaries. Mathematically, this dictionary learning is formulated as follows:

$$\begin{aligned} \min_{\mathbf{D}_0, \{\mathbf{D}_k\}, \{\mathbf{A}_k\}} & \sum_{k=1}^K \frac{1}{2} \|\mathbf{X}_k - [\mathbf{D}_0, \mathbf{D}_k] \mathbf{A}_k\|_F^2 + \lambda \|\mathbf{A}_k\|_1 + \sum_{k=0}^K \frac{\mu_k}{2} \|\mathbf{D}_k^T \mathbf{D}_k - \mathbf{I}_k\|_F^2 + \frac{\eta_k}{2} \|\mathbf{D}_k^T \mathbf{D}_{\setminus k}\|_F^2 \\ \text{s.t.} & \quad \|\mathbf{D}_k(:, j)\|_2 = 1, \forall k, j \end{aligned} \quad (8)$$

where $\mathbf{D}_0 \in \mathbb{R}^{m \times P_0}$ denotes the shared sub-dictionary, which encodes the common feature among the different classes; $\{\mathbf{D}_k\} \in \mathbb{R}^{m \times P_k}$ denote the class-specific sub-dictionaries, which encode the subtle feature differences among the different classes; the complete dictionary is $\mathbf{D} = [\mathbf{D}_0, \mathbf{D}_1, \dots, \mathbf{D}_K] \in \mathbb{R}^{m \times P}$; m is the feature dimension; $P = P_0 + P_1 + \dots + P_K$ is the size of \mathbf{D} ; $\mathbf{D}_{\setminus k}$ is the sub-matrix by removing \mathbf{D}_k from \mathbf{D} ; $\mathbf{I}_k \in \mathbb{R}^{P_k \times P_k}$ is the identity matrix; $\lambda, \{\mu_k\}, \{\eta_k\}$ are regularization parameters.

In the above formulation, the first two terms represent reconstruction error, which are similar to Equation (7). The third term enforces the self-incoherence on each sub-dictionary in order to make the learned dictionary stable, and the fourth term enforces the incoherence constraint among sub-dictionaries to make the dictionary more discriminative.

Although the dictionary learned by DDL is more discriminative, DDL does not use the class labels in the training set. SDL [12] jointly learns the dictionary and classifier by adding a classification error term to Equation (7), which would require solving:

$$\begin{aligned} \min_{\mathbf{D}, \{\mathbf{A}_k\}, \mathbf{W}} \quad & \sum_{k=1}^K \frac{1}{2} \|\mathbf{X}_k - \mathbf{D}\mathbf{A}_k\|_F^2 + \lambda \psi(\mathbf{A}_k) + \gamma L(\mathbf{Y}_k, \mathbf{W}, \mathbf{A}_k) + \frac{\rho}{2} \|\mathbf{W}\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{D}(:, j)\|_2 = 1, \forall j \end{aligned} \quad (9)$$

where $\mathbf{Y}_k = \mathbf{y}_k \times \mathbf{1}^T \in \mathbb{R}^{K \times n_k}$ is the class labels matrix associated with \mathbf{X}_k and \mathbf{y}_k is usually denoted by a one-of- K indicator vector; $L(\cdot)$ measures the classification error between \mathbf{Y}_k , and the prediction of the classifier with model parameter $\mathbf{W} \in \mathbb{R}^{K \times P}$ based on \mathbf{A}_k , and $L(\cdot)$ can be a logistic loss function, hinge loss function or square loss function; γ and ρ are regularization parameters; the last term is a regularizer for stability reasons.

4.2. Supervised Discriminative Dictionary Learning

Inspired by DDL and SDL, we combine Equations (8) and (9) to propose the supervised discriminative dictionary learning method. Specifically, we represent the dictionary as the concentration of the class-specific sub-dictionaries $\{\mathbf{D}_k\}$ and a shared sub-dictionary \mathbf{D}_0 , i.e., $\mathbf{D} = [\mathbf{D}_0, \mathbf{D}_1, \dots, \mathbf{D}_K]$, enforce the incoherence constraints on sub-dictionaries, adopt the square loss function to measure the classification error and overlook the regularization term $\|\mathbf{W}\|_F^2$ like that in LCK-SVD. Therefore, the dictionary, classifier and sparse codes can be obtained by solving:

$$\begin{aligned} \min_{\mathbf{D}_0, \{\mathbf{D}_k\}, \{\mathbf{A}_k\}, \mathbf{W}} \quad & \sum_{k=1}^K \frac{1}{2} \|\mathbf{X}_k - [\mathbf{D}_0, \mathbf{D}_k] \mathbf{A}_k\|_F^2 + \lambda \|\mathbf{A}_k\|_1 + \sum_{k=0}^K \frac{\mu_k}{2} \|\mathbf{D}_k^T \mathbf{D}_k - \mathbf{I}_k\|_F^2 + \frac{\eta_k}{2} \|\mathbf{D}_k^T \mathbf{D}_{\setminus k}\|_F^2 \\ & + \sum_{k=1}^K \frac{\gamma_k}{2} \|\mathbf{Y}_k - \mathbf{W}\mathbf{A}_k\|_F^2 \\ \text{s.t.} \quad & \|\mathbf{D}_k(:, j)\|_2 = 1, \forall k, j \end{aligned} \quad (10)$$

where $\mathbf{A}_k \in \mathbb{R}^{(P_0+P_k) \times n_k}$, $\{\gamma_k\}$ are regularization parameters.

The above problem can be solved iteratively and alternatively, with the $(2K+1)$ of the unknowns fixed each time and solving for the $(2K+2)$ -th. However, the solution is very likely to get stuck in some local minima. Therefore, we have to optimize the above objective function. Now, it is worth taking a few moments to study the form of Equation (10), as it can provide a new perspective about the classification error. We can see that $\sum_{k=1}^K \frac{\gamma_k}{2} \|\mathbf{Y}_k - \mathbf{W}\mathbf{A}_k\|_F^2 + \lambda \|\mathbf{A}_k\|_1$ in Equation (10) has the same form of dictionary learning (see Equation (7)). Additionally, the parameter of classifier \mathbf{W} can be taken as a dictionary; the class labels $\{\mathbf{Y}_k\}$ can be taken as training samples. Thus, the classification error can also be taken as reconstruction error. It should be noted that $\{\mathbf{Y}_k\}$ are actually class labels, and the columns of $\mathbf{Y}_k = \mathbf{y}_k \times \mathbf{1}^T$ are always denoted by the sparsest one-of- K vector \mathbf{y}_k . Here, we represent $\{\mathbf{y}_k\}$ by a group of dense orthonormal vectors. Based on the above understandings, we further represent \mathbf{W} as the concentration of the class-specific sub-dictionaries $\{\mathbf{W}_k\}$ and a shared sub-dictionary \mathbf{W}_0 , i.e., $\mathbf{W} = [\mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_K]$, and also enforce incoherence constraints on the sub-dictionaries of \mathbf{W} . Because \mathbf{D} and \mathbf{W} are learned from two training sets with different natures, it is reasonable to further enforce incoherence constraints among \mathbf{D} and \mathbf{W} . In addition, we elaborately set the regularization parameters for reasonably simplification. Each γ_k is set to be one, meaning that the reconstruction error

term has the same weight with the classification error term. The incoherence constraint terms about \mathbf{D} and \mathbf{W} have the same weights. Therefore, Equation (10) can be developed into the following problem:

$$\begin{aligned} \min_{\mathbf{D}_0, \{\mathbf{D}_k\}, \mathbf{W}_0, \{\mathbf{W}_k\}, \{\mathbf{A}_k\}} & \sum_{k=1}^K \frac{1}{2} \|\mathbf{X}_k - [\mathbf{D}_0, \mathbf{D}_k] \mathbf{A}_k\|_F^2 + \lambda \|\mathbf{A}_k\|_1 + \sum_{k=0}^K \frac{\mu_k}{2} \|\mathbf{D}_k^T \mathbf{D}_k - \mathbf{I}_k\|_F^2 + \frac{\eta_k}{2} \|\mathbf{D}_k^T \mathbf{D}_{\setminus k}\|_F^2 \\ & + \sum_{k=1}^K \frac{1}{2} \|\mathbf{Y}_k - [\mathbf{W}_0, \mathbf{W}_k] \mathbf{A}_k\|_F^2 + \sum_{k=0}^K \frac{\mu_k}{2} \|\mathbf{W}_k^T \mathbf{W}_k - \mathbf{I}_k\|_F^2 + \frac{\eta_k}{2} \|\mathbf{W}_k^T \mathbf{W}_{\setminus k}\|_F^2 \\ & + \sum_{k=0}^K \mu_k \|\mathbf{D}_k \mathbf{W}_k^T\|_F^2 + \eta_k \text{Tr} \left\{ \mathbf{D}_{\setminus k}^T \mathbf{D}_k \mathbf{W}_k^T \mathbf{W}_{\setminus k} \right\} - \frac{\mu_k P_k}{2} \\ \text{s.t.} \quad & \|\mathbf{D}_k(:, j)\|_2 = 1, \forall k, j \end{aligned} \quad (11)$$

where $\mathbf{W}_0 \in \mathbb{R}^{K \times P_0}$ and $\mathbf{W}_k \in \mathbb{R}^{K \times P_k}$; Tr denotes the trace operation. The third line of Equation (11) denotes the incoherence constraints among \mathbf{D} and \mathbf{W} , and the adoption of such a form is for the following mathematical simplification. Then, we combine the parameters in Equation (11) by:

$$\tilde{\mathbf{X}}_k = \begin{bmatrix} \mathbf{X}_k \\ \mathbf{Y}_k \end{bmatrix}, \quad \tilde{\mathbf{D}}_0 = \begin{bmatrix} \mathbf{D}_0 \\ \mathbf{W}_0 \end{bmatrix}, \quad \tilde{\mathbf{D}}_k = \begin{bmatrix} \mathbf{D}_k \\ \mathbf{W}_k \end{bmatrix} \quad (12)$$

where $\tilde{\mathbf{X}}_k \in \mathbb{R}^{(m+K) \times n_k}$, $\tilde{\mathbf{D}}_0 \in \mathbb{R}^{(m+K) \times P_0}$ and $\tilde{\mathbf{D}}_k \in \mathbb{R}^{(m+K) \times P_k}$. Additionally, the simplified form is obtained as follows:

$$\begin{aligned} \min_{\tilde{\mathbf{D}}_0, \{\tilde{\mathbf{D}}_k\}, \{\mathbf{A}_k\}} & \sum_{k=1}^K \frac{1}{2} \|\tilde{\mathbf{X}}_k - [\tilde{\mathbf{D}}_0, \tilde{\mathbf{D}}_k] \mathbf{A}_k\|_F^2 + \lambda \|\mathbf{A}_k\|_1 + \sum_{k=0}^K \frac{\mu_k}{2} \|\tilde{\mathbf{D}}_k^T \tilde{\mathbf{D}}_k - \mathbf{I}_k\|_F^2 + \frac{\eta_k}{2} \|\tilde{\mathbf{D}}_k^T \tilde{\mathbf{D}}_{\setminus k}\|_F^2 \\ \text{s.t.} \quad & \|\tilde{\mathbf{D}}_k(:, j)\|_2 = 1, \forall k, j \end{aligned} \quad (13)$$

and the complete quasi-dictionary $\tilde{\mathbf{D}} = [\tilde{\mathbf{D}}_0, \tilde{\mathbf{D}}_1, \dots, \tilde{\mathbf{D}}_K] \in \mathbb{R}^{(m+K) \times P}$. We can see that Equation (13) has the same form as Equation (8).

4.3. Optimization Procedure

The objective function in Equation (13) is not convex. We can use the similar method in [25] to solve the problem. The main idea is that each \mathbf{A}_k , $\tilde{\mathbf{D}}_k$ and $\tilde{\mathbf{D}}_0$ is updated alternatively while keeping all the rest of the variables fixed, and the updates iterate until reaching the stopping criterion. Specifically, there are three steps in each iteration as follows.

1. Updating sparse codes $\{\mathbf{A}_k\}$: With all of the other factors fixed in Equation (13), the k -th sparse codes matrix \mathbf{A}_k can be obtained by solving the following problem:

$$\min_{\mathbf{A}_k} \frac{1}{2} \|\tilde{\mathbf{X}}_k - [\tilde{\mathbf{D}}_0, \tilde{\mathbf{D}}_k] \mathbf{A}_k\|_F^2 + \lambda \|\mathbf{A}_k\|_1 \quad (14)$$

This problem is the standard sparse coding problem (see Equation (5)). In this paper, we use the fast implementation of LARS algorithm [31], which is a variant for solving the Lasso.

2. Updating class-specific sub-dictionaries $\{\tilde{\mathbf{D}}_k\}$: With the sparse codes $\{\mathbf{A}_k\}$, the shared sub-dictionary $\tilde{\mathbf{D}}_0$ and the $k-1$ class-specific sub-dictionaries fixed, the k -th class-specific sub-dictionary $\tilde{\mathbf{D}}_k$ can be obtained by solving the following problem:

$$\begin{aligned} \min_{\tilde{\mathbf{D}}_k} & \frac{1}{2} \left\| \left(\tilde{\mathbf{X}}_k - \tilde{\mathbf{D}}_0 \mathbf{A}_k^0 \right) - \tilde{\mathbf{D}}_k \mathbf{A}_k^1 \right\|_F^2 + \frac{\mu_k}{2} \|\tilde{\mathbf{D}}_k^T \tilde{\mathbf{D}}_k - \mathbf{I}_k\|_F^2 + \frac{\eta_k}{2} \|\tilde{\mathbf{D}}_k^T \tilde{\mathbf{D}}_{\setminus k}\|_F^2, \\ \text{s.t.} \quad & \|\tilde{\mathbf{D}}_k(:, j)\|_2 = 1, \forall j \end{aligned} \quad (15)$$

where $\mathbf{A}_k^0 \in \mathbb{R}^{P_0 \times n_k}$, $\mathbf{A}_k^1 \in \mathbb{R}^{P_k \times n_k}$ and $\mathbf{A}_k = [\mathbf{A}_k^0 \ \mathbf{A}_k^1]^T$. We use the gradient descent algorithm to solve the objective function, and the step size is chosen according to the Armijo rule, like that in [13,25].

3. Updating the shared sub-dictionary $\tilde{\mathbf{D}}_0$: When $\{\mathbf{A}_k\}$ and $\{\tilde{\mathbf{D}}_k\}$ are fixed, we can obtain $\tilde{\mathbf{D}}_0$ by solving the following problem:

$$\begin{aligned} \min_{\tilde{\mathbf{D}}_0} \quad & \sum_{k=1}^K \frac{1}{2} \left\| \left(\tilde{\mathbf{X}}_k - \tilde{\mathbf{D}}_k \mathbf{A}_k^1 \right) - \tilde{\mathbf{D}}_0 \mathbf{A}_k^0 \right\|_F^2 + \frac{\mu_k}{2} \left\| \tilde{\mathbf{D}}_0^T \tilde{\mathbf{D}}_0 - \mathbf{I}_0 \right\|_F^2 + \frac{\eta_k}{2} \left\| \tilde{\mathbf{D}}_0^T \tilde{\mathbf{D}}_{\setminus 0} \right\|_F^2, \\ \text{s.t.} \quad & \left\| \tilde{\mathbf{D}}_0(:, j) \right\|_2 = 1, \forall j \end{aligned} \quad (16)$$

We also use the gradient descent algorithm to optimize the above problem, like that in [13,25].

The above iterative process stops when the number of iterations reaches the predefined value M or the respective relative change between two successive estimations of $\tilde{\mathbf{D}}$ and $\{\mathbf{A}_k\}$ are less than the given constants, i.e.,

$$\frac{\left\| \tilde{\mathbf{D}}^s - \tilde{\mathbf{D}}^{s-1} \right\|_F}{\left\| \tilde{\mathbf{D}}^s \right\|_F} < \text{thres}_{\tilde{\mathbf{D}}} \quad \wedge \quad \frac{\sum_{k=1}^K \left\| \mathbf{A}_k^s - \mathbf{A}_k^{s-1} \right\|_F}{\sum_{k=1}^K \left\| \mathbf{A}_k^s \right\|_F} < \text{thres}_{\mathbf{A}} \quad (17)$$

where s denotes the s -th iteration; \wedge denotes the AND operation; $-\text{thres}_{\tilde{\mathbf{D}}}$ and $-\text{thres}_{\mathbf{A}}$ denote the convergence thresholds, which are set to be small numerical constants. Although the solutions obtained by this algorithm are not the exact solutions, they are actually the solutions satisfying the constraints [25]. Additionally, the experiments in Section 5 also demonstrate the good performance of these solutions.

Following the work of [27], once $\tilde{\mathbf{D}}$ is obtained, the final desired dictionary $\hat{\mathbf{D}}$ and the parameter of classifier $\hat{\mathbf{W}}$ can be derived from $\tilde{\mathbf{D}}$ as follows:

$$\hat{\mathbf{D}} = \left[\frac{\mathbf{d}_1}{\left\| \mathbf{d}_1 \right\|_2}, \dots, \frac{\mathbf{d}_j}{\left\| \mathbf{d}_j \right\|_2}, \dots, \frac{\mathbf{d}_P}{\left\| \mathbf{d}_P \right\|_2} \right], \quad \hat{\mathbf{W}} = \left[\frac{\mathbf{w}_1}{\left\| \mathbf{d}_1 \right\|_2}, \dots, \frac{\mathbf{w}_j}{\left\| \mathbf{d}_j \right\|_2}, \dots, \frac{\mathbf{w}_P}{\left\| \mathbf{d}_P \right\|_2} \right] \quad (18)$$

where $\mathbf{d}_j = \tilde{\mathbf{D}}(1:m, j)$, $\mathbf{w}_j = \tilde{\mathbf{D}}(m+1:m+K, j)$, $\hat{\mathbf{D}} \in \mathbb{R}^{m \times P}$ and $\hat{\mathbf{W}} \in \mathbb{R}^{K \times P}$.

As for the initialization of the above iterative process, we initialize $\{\mathbf{D}_k\}$ and $\{\mathbf{A}_k\}$ with K-SVD by using the training samples from different classes, and we initialize \mathbf{D}_0 by using the training samples from all of the classes. For the initialization of \mathbf{W}_0 and $\{\mathbf{W}_k\}$, we can use the ridge regression model to initialize, like that in [27]. As described above, \mathbf{W} can also be taken as a dictionary, so K-SVD can be employed for initialization. In this paper, we use the latter method to initialize \mathbf{W}_0 and $\{\mathbf{W}_k\}$. According to Equation (12), the initialized $\{\tilde{\mathbf{X}}_k\}$, $\tilde{\mathbf{D}}_0$ and $\{\tilde{\mathbf{D}}_k\}$ are obtained.

The parameter setting, performing cross-validation on λ , $\{\mu_k\}$, $\{\eta_k\}$ (regulation parameters) and P (size of dictionary), would be cumbersome. Fortunately, due to the same forms of Equations (8) and (13), we can refer to the method in [25] for parameter setting. For sparse regularization parameter λ , it has been experimentally shown that good performance can be achieved when it is set to be 0.3 [25]. We also set λ to be 0.3 for simplification. In [25], $\{\mu_k\}$ and $\{\eta_k\}$ are set as follows:

$$\mu_k = \frac{a \cdot n_k}{2KP_k^2}, \quad \eta_k = \frac{b \cdot n_k}{P_k(P - P_k)} \quad (19)$$

where a and b are controllable parameters. Such a formulation considers the normalization of the incoherence term by the number of samples and the size of the sub-dictionary in each class. In [25], the ratio between η_k and μ_k is simply set to be two, and b is set to be 0.1. We also take the formulation of Equation (19) and, meanwhile, constrain $\eta_k/\mu_k = 2$. b is set by experiment, which is illustrated in

Section 5. As for the size of the dictionary, [25] shows that a bigger size of class-specific sub-dictionary can lead to better performance, and a moderately-sized shared sub-dictionary leads to the desired performance. A similar rule still holds in our method. It should be noted that P grows linearly with the number of classes. Therefore, we should properly adjust P according to the real SAR ATR problem.

The overall optimization procedure for solving Equation (13) is summarized in Algorithm 1.

Algorithm 1: Supervised discriminative dictionary learning (SDDL).

Input:

feature vectors of the ROIs: $\{\mathbf{X}_k\}$, $k = 1, \dots, K$; class labels $\{\mathbf{Y}_k\}$; regularization parameters, $\lambda, \{\mu_k\}, \{\eta_k\}$; the size of sub-dictionaries, $\{P_k\}$, $k = 0, \dots, K$;
the stopping thresholds $-\text{thres}_{\hat{\mathbf{D}}}$ and $-\text{thres}_{\hat{\mathbf{A}}}$ or maximum iterations M .

Initialization:

Initialize $\{\mathbf{D}_k\}$, \mathbf{D}_0 , $\{\mathbf{W}_k\}$, \mathbf{W}_0 and $\{\mathbf{A}_k\}$ with K-SVD.
Initialize $\{\tilde{\mathbf{X}}_k\}$, $\tilde{\mathbf{D}}_0$ and $\{\tilde{\mathbf{D}}_k\}$ by Equation (12).

Repeat

for $k = 1$ **to** K **do**

Update the sparse codes \mathbf{A}_k of $\tilde{\mathbf{X}}_k$ with LARS algorithm (see Equation (14)).

end for

for $k = 1$ **to** K **do**

Update each \mathbf{D}_k by using gradient descent algorithm (see Equation (15)).

end for

Update \mathbf{D}_0 by using gradient descent algorithm (see Equation (16)).

Until reaching the stopping criterion (see Equation (17)).

Output: The desired dictionary $\hat{\mathbf{D}}$ and the classifier parameter $\hat{\mathbf{W}}$ obtained by Equation (18).

4.4. SAR ATR Algorithm

Given the training slice images of ROIs and relevant class labels from K classes, we first extract features from the slice images. Then, we use the SDDL method to jointly learn a discriminative dictionary $\hat{\mathbf{D}}$ and classifier $\hat{\mathbf{W}}$ using Algorithm 1. Then, we decompose the feature vector of a certain test sample \mathbf{x}_t onto $\hat{\mathbf{D}}$ to obtain its sparse code \mathbf{a}_t by solving the problem of Equation (5), and here, the \mathbf{D} in Equation (5) is replaced by $\hat{\mathbf{D}}$. Finally, we identify \mathbf{x}_t based on \mathbf{a}_t by using the following decision rule:

$$\hat{k} = \arg \min_{k \in \{1, \dots, K\}} \|\mathbf{x}_t - \hat{\mathbf{D}} \delta_k(\mathbf{a}_t)\|_2^2 + \|\mathbf{y}_k - \hat{\mathbf{W}} \delta_k(\mathbf{a}_t)\|_2^2 \quad (20)$$

where $\mathbf{a}_t \in \mathbb{R}^{P_0+P_k}$. By comparing Equations (6) and (20), we can see that both the reconstruction error and the classification error are actually considered by us.

The proposed SAR ATR algorithm via supervised discriminative dictionary learning and sparse representation is summarized in Algorithm 2. Here, the proposed method is denoted by **SDDL SR**.

Algorithm 2: SAR ATR via SDDL SR.

Input:

Slice images of ROIs from the training set and the testing set and class labels in the training set.

Feature extraction:

Compute features of slice images. In this paper, SAR-HOG features are computed following the description in Section 3.

Classification:

Learn the dictionary $\hat{\mathbf{D}}$ and the classifier parameter $\hat{\mathbf{W}}$ using Algorithm 1.

Decompose the test sample \mathbf{X}_t onto $\hat{\mathbf{D}}$ and identify by the decision rule (see Equation (20)).

Output: The identity of \mathbf{X}_t .

5. Experiments and Discussions

In order to verify the effectiveness and robustness of the proposed method, we perform experiments using the Moving and Stationary Target Automatic Recognition (MSTAR) public database [29]. This database is a gallery collected using an X-band HH polarization SAR with $1 \text{ ft} \times 1 \text{ ft}$ resolution for multiple targets. The SAR images are captured at various depression angles ($15^\circ, 17^\circ, 30^\circ, 45^\circ$) over a 0° – 360° range of aspect view. The sizes of the images are all around 128×128 pixels. In this paper, these images are cropped to 64×64 pixels to further avoid the influence of clutter. The performance of SAR ATR obtained by the proposed method is attributed to two factors, i.e., the SAR-HOG feature and the SDDLRS classifier. Therefore, we should demonstrate the effectiveness of the SAR-HOG feature and SDDLRS, respectively.

In the following, we first detail the experiment setup and parameter setting. Then, we evaluate the effectiveness of the SAR-HOG feature by using SDDLRS as the baseline classifier in Section 5.3. In Sections 5.4 and 5.5, we demonstrate the effectiveness of SDDLRS classifier based on the same feature SAR-HOG. Here, some classic classification methods, including SVM, kNN, SRC and the SDL method LCK-SVD, are compared to the proposed SDDLRS classifier. The performance of the proposed method with the reduced size of the training set is evaluated in Section 5.6. The time consumption is illustrated in Section 5.7. All of the experiments are performed by MATLAB, using a common PC with the Intel Core i7 processor with a 3.40-GHz main frequency and an 8.00-GB main memory.

5.1. Experiment Setup

Experiments are performed under standard operating conditions (SOC) and extended operating conditions (EOC) [29]. In SOC scenarios, the testing conditions are very close to the training conditions. However, in EOC scenarios, the operating conditions of testing are always away from the training conditions. Therefore, EOC scenarios are closer to real-world battlefield scenarios.

In the experiments under SOC, the images of ten targets acquired at a 17° depression angle are used to train, while the images acquired at a 15° depression angle are used to test [3,29]. The number of images for training and testing is tabulated in Table 1. It should be noted that BMP2 and T72 have several variants with small structural modifications, and these variants are denoted by different series numbers (SN). Here, only the SN_132 of T72 and SN_9563 of BMP2 acquired at a 17° depression angle are used for training.

Table 1. The number of images for training and testing about the ten targets under standard operating conditions (SOC).

Target	2S1	BRDM2	BTR60	D7	T62	ZIL131	ZSU23/4	BRT70	T72	BMP2
Training (17°)	299	298	256	299	299	299	299	233	(SN_132) 232	(SN_9563) 233
									(SN_132) 196	(SN_9563) 195
Testing (15°)	274	274	195	274	273	274	274	196	(SN_812) 195	(SN_9566) 196
									(SN_s7) 191	(SN_c21) 196

In the experiments under EOC, the EOC difference on the depression angle (denoted by EOC_d) is considered. Four targets, 2S1, BRDM2, ZSU23/4 and T72(SN_A64) at different depression angles under Scene 1 are used [29]. Specifically, the images acquired at a 17° depression angle are used for training, and the images acquired at 30° and 45° depression angles are used for testing. Here, the two cases are denoted by “Training 17° —Testing 30° ” and “Training 17° —Testing 45° ”, respectively. The number of images for training and testing about the four targets is tabulated in Table 2.

Table 2. The number of images for training and testing about the four targets under extended operating conditions (EOC).

Target	2S1	BRDM2	ZSU23/4	T72(SN_A64)
Training (17°)	299	298	299	299
Testing (30°)	288	287	288	288
Testing (45°)	303	303	303	303

For SAR ATR, the training samples acquired from the real battlefield are usually very limited. Therefore, we further study the performance of the SAR ATR methods with a small training set. Specifically, the case “Training 17°—Testing 30°” is considered, and the training samples are gradually reduced. Each time, 50 samples per class are randomly removed from the previous training set, and the left is used as the current training set. For each training size, experiments are repeated 100 times to have the mean performance.

5.2. Parameter Setting

The parameters in the proposed method include the parameters of SAR-HOG and the parameters of SDDL. The setting principles have been given in Sections 3 and 4.3. The parameter setting in SAR-HOG actually depends on SAR image resolution and the sizes of targets in the images. Here, we further illustrated the parameter setting. Additionally, the fixed parameter values are tabulated in Table 3.

Table 3. The parameters used in the proposed method.

Parameter	win	Bins	Cell (Pixels)	Block (Cells)	Stride (Pixels)	λ	b	P_k	M
Value	11	11	8×8	4×4	16	0.3	0.1	96	20

We use the experiment setup “Training 17°—Testing 30°” and the proposed SDDL method to choose the parameter values. The experiment results are recorded in Figures 3 and 4 and Tables 4–6. It should be noted that when the current parameters are tested, the left parameters are fixed at the values in Table 3.

Figure 3a shows that properly increasing the average region size win for computing $M_1(i)$ and $M_2(i)$ can improve the recognition rate up to 11 pixels. Figure 3b shows that setting the number of bins to be 11 can obtain the best performance. Tables 4 and 5 record the recognition rate and time consumption as the cell and block sizes change, respectively. The items for which the corresponding recognition rates are bigger than 0.96 are in bold. We can see that when the cell sizes are set to be 4–8 pixels, the block sizes are set to be 2–7 cells; meanwhile, the block sizes are no more than 32 pixels, and the recognition rates are no less than 0.96. In addition, although the longer SAR-HOG feature length associated with the smaller cell and block sizes tends to lead to a higher recognition rate, the time consumption is more. Therefore, we choose 8×8 pixel cells and 4×4 cell blocks in the experiments to compromise between recognition rate and time consumption. Table 6 shows that setting the stride to be 16 pixels (i.e., half of the block size) can lead to desirable performance. As for the parameter λ and b in SDDL, λ is set to be the empirical value 0.3 [25], and $b = 0.1$ is an ideal value shown in Figure 4a. As for the size of the dictionary, Figure 4b shows that setting $P_k, k = 0, 1, \dots, K$ to be 96 works best. For the stopping criterion, we find that the performance of Algorithm 1 is satisfactory within 20 iterations. In addition, small $-thres_D$ and $-thres_A$, e.g., 10^{-4} can guarantee the convergence of Algorithm 1. However, such a setting generally results in more than one hundred iterations. Therefore, we simply set the max iterations $M = 20$ in the experiments.

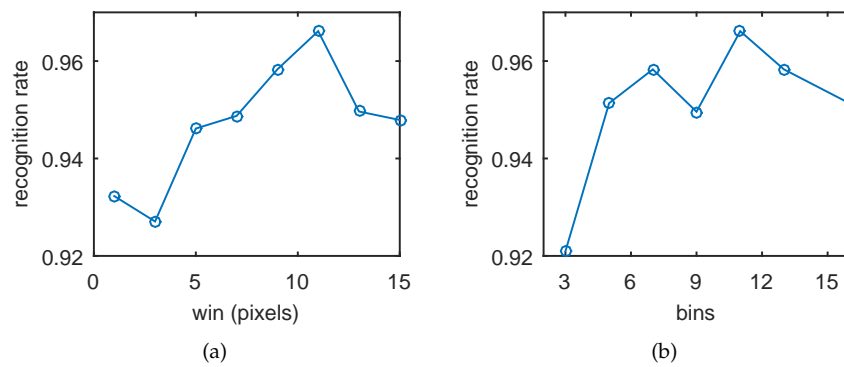


Figure 3. The recognition rate versus the average region size and histogram bins. (a) The recognition rate versus the average region size win . win is 1, 3, 5, 7, 9, 11, 13 and 15. (b) The recognition rate versus the histogram bins. Bins are 3, 5, 7, 9, 11, 13 and 17.

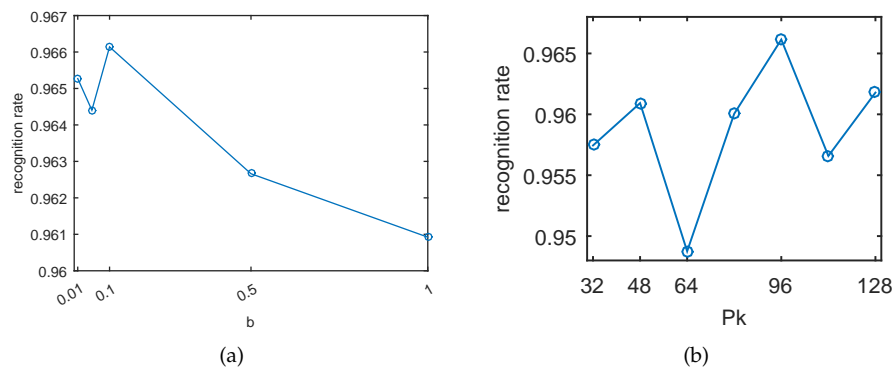


Figure 4. The recognition rate versus b and P_k in supervised discriminative dictionary learning (SDDL). (a) The recognition rate versus b . b is 0.01, 0.05, 0.1, 0.5 and 1. (b) The recognition rate versus P_k . P_k is 32, 48, 64, 80, 96, 112 and 128.

Table 4. The recognition rate versus different (cell, block)s.

		Block (Cells)								
		2	3	4	5	6	7	8	9	10
cell (pixels)	4	0.9661	0.9783	0.9783	0.9739	0.9704	0.9652	0.9513	0.9235	0.9139
	6	0.9609	0.9713	0.9731	0.9661	0.9383	0.9261	0.9304	0.9348	0.9287
	8	0.9661	0.9714	0.9661	0.9522	0.9357	0.9453	0.9470	-	-
	10	0.9505	0.9392	0.9383	0.9427	0.9505	-	-	-	-
	12	0.9392	0.9340	0.9314	0.9349	-	-	-	-	-

“-” denotes that the block size is bigger than the image size. The items for which recognition rates are bigger than 0.96 are in bold.

Table 5. The time consumption (min) per iteration of Algorithm 1 versus different (cell, block)s.

		Block (Cells)								
		2	3	4	5	6	7	8	9	10
cell (pixels)	4	36.5591	24.3237	28.3787	19.0616	16.4551	9.5129	16.6214	5.3417	8.0049
	6	4.8953	5.0265	3.2544	2.5986	1.1269	2.0317	0.2095	0.3557	0.6282
	8	2.0408	1.1931	1.1706	0.5677	0.0744	0.1299	0.2157	-	-
	10	0.5514	0.3604	0.2184	0.0379	0.0747	-	-	-	-
	12	0.2355	0.0817	0.0255	0.0404	-	-	-	-	-

“-” denotes that the block size is bigger than the image size. The items for which the corresponding recognition rates in Table 4 are bigger than 0.96 are in bold.

Table 6. The recognition rate versus the stride.

stride (pixels)	0	8	16	24
recognition rate	0.9505	0.9505	0.9661	0.9653

5.3. Effectiveness of SAR-HOG

The effectiveness of the SAR-HOG feature is first evaluated using the experiment setups under SOC and EOC_d. Specifically, the commonly-used features, i.e., raw intensity images [1] and the wavelet decomposition images [2] are used to compare with the results based on the SAR-HOG feature by using SDDLRS as the baseline classifier. It is easy to comprehend the feature “intensity image”. The features “wavelet decomposition images” are actually the LL, LH and HL (L = low, H = high) sub-bands obtained after a multi-level wavelet decomposition using the 2D reverse biorthogonal wavelet [2]. The parameters in computing SAR-HOG are listed in Table 3. The SDDLRS is implemented in Algorithm 2, the fixed parameters of which are also listed in Table 3.

The results obtained by using Algorithm 2 and taking the intensity image as the input feature under SOC and EOC_d are recorded in Tables 7 and 8, respectively. The results obtained by using Algorithm 2 and taking wavelet decomposition images as the input feature under SOC and EOC_d are recorded in Tables 9 and 10, respectively. The results obtained by using Algorithm 2 and taking SAR-HOG as the input feature under SOC and EOC_d are recorded in Tables 11 and 12, respectively. Results include the confusion matrices and the overall recognition rates. The confusion matrix is a matrix of fractions, with each column representing a possible decision class and each row representing a class that is present to the ATR system. The diagonal elements of the confusion matrix represent the fraction of correct decisions, and the mean value of the diagonal elements is the overall recognition rate.

Table 7. The result of supervised discriminative dictionary learning sparse representation (SDDLRS) based on the raw intensity image under SOC.

	2S1	BRDM2	BTR60	D7	T62	ZIL131	ZSU23/4	BRT70	T72	BMP2
2S1	0.9380	0	0	0	0.0073	0	0.0547	0	0	0
BRDM2	0.0051	0.9590	0	0	0	0	0	0.0308	0	0.0051
BTR60	0	0.0154	0.8769	0	0	0.0103	0	0.0872	0.0051	0.0051
D7	0	0	0	0.9818	0.0109	0.0073	0	0	0	0
T62	0	0	0	0.0440	0.9413	0.0037	0.0110	0	0	0
ZIL131	0	0	0.0036	0.0073	0	0.9891	0	0	0	0
ZSU23/4	0.0146	0	0	0	0	0	0.9854	0	0	0
BRT70	0	0	0.0051	0	0	0	0	0.9898	0	0.0051
T72	0	0.0086	0.0120	0	0	0	0	0.0258	0.8746	0.0790
BMP2	0	0.0221	0.0017	0	0	0	0	0.0358	0.0801	0.8603
recognition rate	0.9406									

Table 8. The result of SDDLRS based on the raw intensity image under EOC_d.

	Training(17°)—Testing(30°)				Training(17°)—Testing(45°)			
	2S1	BRDM2	ZSU23/4	T72(SN_A64)	2S1	BRDM2	ZSU23/4	T72(SN_A64)
2S1	0.9757	0.0243	0	0	0.7459	0.2013	0.0462	0.0066
BRDM2	0.0523	0.8014	0.0070	0.1394	0.0528	0.8218	0.0066	0.1188
ZSU23/4	0.0382	0.0069	0.9514	0.0035	0.1254	0.0792	0.7195	0.0759
T72(SN_A64)	0.0104	0	0.0521	0.9375	0.1155	0.0495	0.2475	0.5875
recognition rate	0.9165				0.7186			

Table 9. The result of SDDLRS based on the wavelet decomposition image under SOC.

	2S1	BRDM2	BTR60	D7	T62	ZIL131	ZSU23/4	BRT70	T72	BMP2
2S1	0.9782	0	0	0	0.0036	0	0.0182	0	0	0
BRDM2	0.0036	0.9307	0.0109	0	0	0	0	0.0109	0.0365	0.0073
BTR60	0	0.0154	0.9538	0	0	0	0	0.0256	0	0.0051
D7	0	0	0	0.9854	0	0.0146	0	0	0	0
T62	0	0	0	0.0256	0.9634	0.0110	0	0	0	0
ZIL131	0	0	0	0.0109	0	0.9891	0	0	0	0
ZSU23/4	0.0073	0	0	0	0	0	0.9927	0	0	0
BRT70	0	0.0051	0.0204	0	0	0	0	0.9745	0	0
T72	0	0.0309	0.0241	0	0	0	0	0.0241	0.8986	0.0223
BMP2	0	0.0716	0.0153	0	0	0	0	0.0937	0.1056	0.7138
recognition rate	0.9380									

Table 10. The result of SDDLRS based on the wavelet decomposition image under EOC_d.

	Training(17°)—Testing(30°)				Training(17°)—Testing(45°)			
	2S1	BRDM2	ZSU23/4	T72(SN_A64)	2S1	BRDM2	ZSU23/4	T72(SN_A64)
2S1	0.9618	0.0208	0.0069	0.0104	0.8548	0.1221	0.0099	0.0132
BRDM2	0.0244	0.9686	0.0035	0.0035	0.0594	0.9340	0.0033	0.0033
ZSU23/4	0.0347	0.0208	0.9271	0.0174	0.1023	0.1584	0.6634	0.0759
T72(SN_A64)	0.0938	0.0243	0.1285	0.7535	0.2739	0.2409	0.1386	0.3465
recognition rate	0.9028				0.6997			

Table 11. The result of SDDLRS based on SAR-HOG under SOC.

	2S1	BRDM2	BTR60	D7	T62	ZIL131	ZSU23/4	BRT70	T72	BMP2
2S1	0.9891	0	0	0	0	0	0.0109	0	0	0
BRDM2	0	0.9635	0.0073	0	0	0	0	0.0182	0.0073	0.0036
BTR60	0.0051	0	0.9333	0	0	0	0	0.0410	0.0103	0.0103
D7	0	0	0	0.9927	0	0.0073	0	0	0	0
T62	0	0	0	0	1	0	0	0	0	0
ZIL131	0	0	0	0	0	1	0	0	0	0
ZSU23/4	0	0	0	0	0	0	1	0	0	0
BRT70	0	0.0051	0.0051	0	0	0	0	0.9694	0	0.0204
T72	0	0.0172	0.0155	0	0	0	0	0.0086	0.8883	0.0704
BMP2	0	0.0290	0.0051	0	0	0	0	0.0239	0.0545	0.8876
recognition rate	0.9624									

Table 12. The result of SDDLRS based on the SAR-HOG under EOC_d.

	Training(17°)—Testing(30°)				Training(17°)—Testing(45°)			
	2S1	BRDM2	ZSU23/4	T72(SN_A64)	2S1	BRDM2	ZSU23/4	T72(SN_A64)
2S1	0.9792	0.0035	0.0104	0.0069	0.8515	0.0495	0.0660	0.0330
BRDM2	0.0070	0.9930	0	0	0.0165	0.9571	0	0.0264
ZSU23/4	0.0035	0.0069	0.9722	0.0174	0.0297	0.0165	0.7954	0.1584
T72(SN_A64)	0.0417	0	0.0382	0.9201	0.1452	0.0495	0.1749	0.6304
recognition rate	0.9661				0.8086			

By comparing the results in Tables 7–12, we can see that the SAR-HOG feature leads to the best performance under SOC and EOC_d by using the same classifier SDDLRS. For the SOC scenario, the recognition rates for using raw intensity images and using wavelet decomposition images are 0.9406 and 0.9380, compared to 0.9624 by using SAR-HOG. For the EOC_d scenario “Training 17°—Testing 30°”, the recognition rates for using raw intensity images and using wavelet decomposition images are 0.9165 and 0.9028, compared to 0.9661 by using SAR-HOG. For the most challenging EOC_d scenario

Table 16. The result of label consistent K(LCK)-SVD based on SAR-HOG under SOC.

	2S1	BRDM2	BTR60	D7	T62	ZIL131	ZSU23/4	BRT70	T72	BMP2
2S1	0.9781	0	0	0	0	0	0.0219	0	0	0
BRDM2	0	0.9453	0.0146	0	0	0	0	0.0219	0.0146	0.0036
BTR60	0	0.0051	0.9436	0	0	0	0	0.0103	0.0256	0.0154
D7	0	0	0	0.9854	0.0036	0.0109	0	0	0	0
T62	0	0	0	0.0073	0.9927	0	0	0	0	0
ZIL131	0	0	0	0	0	1	0	0	0	0
ZSU23/4	0.0036	0	0	0	0.0036	0	0.9927	0	0	0
BRT70	0	0	0.0204	0	0	0	0	0.9694	0	0.0102
T72	0	0.0086	0.0275	0	0	0	0	0.0086	0.8814	0.0739
BMP2	0	0.0290	0.0375	0	0	0	0	0.0426	0.0784	0.8126
recognition rate	0.9501									

By comparing the results in Tables 11 and 13–16, we can see that, SDDLRS achieves the highest recognition rate of 0.9624, which is 2.37%, 1.97%, 1.43% and 1.23% better than SVM, kNN, SRC and LCK-SVD, respectively. These experiments show that SDDL can further capture the subtle differences among the different classes by learning a discriminative dictionary. Additionally, the sparse codes of the testing samples by sparsely decomposing become more distinctive.

By far, the best result under SOC is obtained by sparse representation on Riemannian manifolds (SRCR) (see Table VII in [3]), i.e., the recognition rate for ten targets under SOC is 0.9448. Compared with this result, the recognition rate by our proposed method improves 1.36%.

5.5. Four Targets ATR under EOC

As described in Section 3, SAR images have remarkable aspect sensitivity. The change of depression angle from 17°–30° can lead to the drastic change of the SAR image, not to mention the change from 17°–45°. Therefore, the stable structures of the targets are more difficult to capture; thus, a more efficient feature extraction method should be employed. For the problem of four targets' ATR under EOC, we also take SAR-HOG as the input feature and compare the proposed SDDLRS classifier with the SVM, kNN, SRC and LCK-SVD classifiers. The parameters in computing SAR-HOG and the parameters used in SDDLRS (Algorithm 2) are listed in Table 3. The implementations of SVM, kNN, SRC and LCK-SVD are illustrated in Section 5.7. The experimental results of the proposed SDDLRS method and the reference methods are tabulated in Tables 12 and 17–20.

Table 17. The result of SVM based on SAR-HOG under EOC_d.

	Training(17°)—Testing(30°)				Training(17°)—Testing(45°)			
	2S1	BRDM2	ZSU23/4	T72(SN_A64)	2S1	BRDM2	ZSU23/4	T72(SN_A64)
2S1	0.9549	0	0.0243	0.0208	0.7723	0.1089	0.0264	0.0924
BRDM2	0.0105	0.9895	0	0	0.0363	0.9571	0	0.0066
ZSU23/4	0	0	0.8993	0.1007	0.0099	0.0759	0.5578	0.3564
T72(SN_A64)	0.0625	0	0.1285	0.8090	0.1815	0.1056	0.0957	0.6172
recognition rate	0.9132				0.7261			

Table 18. The result of kNN based on SAR-HOG under EOC_d.

	Training(17°)—Testing(30°)				Training(17°)—Testing(45°)			
	2S1	BRDM2	ZSU23/4	T72(SN_A64)	2S1	BRDM2	ZSU23/4	T72(SN_A64)
2S1	0.9653	0	0.0174	0.0174	0.8152	0.0528	0.0759	0.0561
BRDM2	0.0732	0.8955	0	0.0314	0.0627	0.8383	0.0066	0.0924
ZSU23/4	0.0139	0	0.9583	0.0278	0.0858	0.0594	0.6568	0.1980
T72(SN_A64)	0.0417	0	0.0764	0.8819	0.0891	0.0627	0.2640	0.5842
recognition rate	0.9253				0.7236			

Table 19. The result of SRC based on SAR-HOG under EOC_d.

	Training(17°)—Testing(30°)				Training(17°)—Testing(45°)			
	2S1	BRDM2	ZSU23/4	T72(SN_A64)	2S1	BRDM2	ZSU23/4	T72(SN_A64)
2S1	0.9514	0	0.0139	0.0347	0.8647	0.0726	0.0297	0.0330
BRDM2	0.0488	0.9408	0.0035	0.0070	0.0990	0.8845	0.0033	0.0132
ZSU23/4	0.0035	0.0139	0.9444	0.0382	0.0627	0.0891	0.6007	0.2475
T72(SN_A64)	0.0451	0	0.0694	0.8854	0.0924	0.0660	0.2706	0.5710
recognition rate	0.9305				0.7302			

Table 20. The result of LCK-SVD based on SAR-HOG under EOC_d.

	Training(17°)—Testing(30°)				Training(17°)—Testing(45°)			
	2S1	BRDM2	ZSU23/4	T72(SN_A64)	2S1	BRDM2	ZSU23/4	T72(SN_A64)
2S1	0.9549	0.0104	0.0139	0.0208	0.7360	0.1221	0.0528	0.0891
BRDM2	0.0174	0.9756	0	0.0070	0.0462	0.8581	0.0066	0.0891
ZSU23/4	0	0.0069	0.9375	0.0556	0.0627	0.0264	0.6865	0.2244
T72(SN_A64)	0.0208	0.0035	0.1389	0.8368	0.1320	0.0660	0.1320	0.6700
recognition rate	0.9262				0.7376			

By comparing the results in Tables 12 and 17–20, we can see that, although the experiment setting “Training 17°—Testing 30°” has a 13° depression angle difference, the results based on the SAR-HOG feature are acceptable on the whole. The SDDL SR achieves the best recognition rate of 0.9661, compared to 0.9132 for SVM, 0.9253 for kNN, 0.9305 for SRC and 0.9262 for LCK-SVD, respectively. In addition, it is no surprise that the performance of all of the methods degrades under the experiment setting “Training 17°—Testing 45°”. Additionally, the recognition rate decreases by more than 10% for each method. The proposed method still achieves the highest recognition rate of 0.8086. It is 8.25%, 8.5%, 7.84% and 7.1% better than SVM, kNN, SRC and LCK-SVD, respectively.

By far, the best results under EOC_d are obtained by the TJSR method (see Figure 7 in [3]), i.e., the recognition rates for three targets (2S1, BRDM2 and ZSU23/4) under the EOC_d scenarios “Training 17°—Testing 30°” and “Training 17°—Testing 45°” are 0.9524 and 0.7073, respectively. Compared with these results, the recognition rates by our proposed method improve 1.37% and 10.13%, respectively. The experiments indicate that our method has more advantages to deal with the EOC scenario with a remarkable depression angle difference.

5.6. Experiment on Small Training Set

The recognition rates of different methods versus the reduced training set under “Training 17°—Testing 30°” are shown in Figure 5. We can see that the recognition rates of all of the methods degrade with the reduced size of the training set. When about one third of the original training samples, i.e., 395 training samples, are left, the recognition rate of our method is 0.9314, compared to 0.8633 for SVM, 0.8980 for kNN, 0.9158 for SRC and 0.8874 for LCK-SVD, respectively. The experimental results indicate that the proposed method has the best performance with a reduced training set.

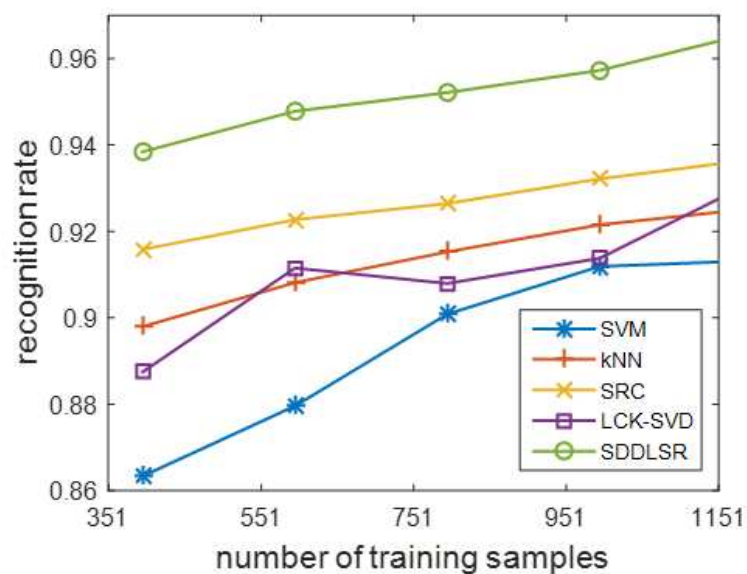


Figure 5. Recognition rates of different methods versus the training set.

5.7. Time Consumption

The time consumption in SAR ATR tasks includes two main parts: the training time and the testing time. Generally, the classifier can be trained off-line. Additionally, the testing time plays an important role in the real-time operation of SAR ATR. For the proposed method, the training time is the time consumption of Algorithm 1. The testing time is the time consumption of sparse representation classification (see Equation (20)). In the proposed method, SAR-HOG calculation and the gradient descent algorithm are written in our unrefined MATLAB code; the LARS algorithm in Algorithm 2 is implemented by the SPAMS software [31]. For the reference methods, SVM is implemented by the well-known LIBSVM software [32]; kNN is implemented by the functions in MATLAB Statistics and the Machine Learning Toolbox; SRC is implemented by the SPAMS software; LCK-SVD is implemented by the code supplied in [27]. The time consumptions of different methods under “Training 17°—Testing 30°” are recorded in Table 21. It can be seen that the computing time of SAR-HOG and the training time of the proposed method are acceptable, and the testing performance is satisfactory. In addition, the training stage of SRC is the dictionary construction by just stacking the training samples; the training stage of LCK-SVD is the discriminative dictionary learning.

Table 21. The time consumption of different methods (s).

Method	Computing Time of SAR-HOG per Image	Training Time	Testing Time
SVM		5.7660	11.2620
kNN		0.6180	1.0440
SRC	0.9660	0.0362	1.1820
LCK-SVD		113.76	0.8160
proposed		1404.72	0.8280

6. Conclusions

In this paper, we proposed an SAR ATR method via supervised discriminative dictionary learning and sparse representation based on the SAR-HOG feature. Firstly, we adapted a local feature named HOG to SAR images by using the ratio-based gradient definition to deal with speckle noise. Then, we combined the merits of discriminative dictionary learning and supervised dictionary learning to propose the supervised discriminative dictionary learning. Using the SDDL, we obtained the discriminative dictionary and the classifier meanwhile. Finally, we sparsely decomposed the test

sample onto the learned dictionary and identified the test sample based on the summation of the reconstruction error and the classification error.

In order to demonstrate the effectiveness and robustness of the proposed method, we performed extensive experiments on the MSTAR database under SOC and EOC. Additionally, we used several classification methods, including SVM, kNN, SRC and LCK-SVD, as reference methods. From the experimental results, we can obtain the following conclusions: (1) SAR-HOG can effectively capture the features embedded in the stable pixels corresponding to stable structures of targets; (2) SDDL can further capture the subtle differences between the different classes; and the proposed strategy can effectively simplify the optimization problem; (3) Compared with the published best results (see Figure 7 and Table VII in [3]), the proposed method can obtain higher recognition rate, especially for the EOC scenario “Training 17°—Testing 45°”.

In the future, the proposed method can be extended to other applications. On the one hand, SAR-HOG can be used as an effective local feature to be applied to target detection, image indexing, SAR image registration, and so on. On the other hand, SDDL can be used to learn a representative and discriminative dictionary for discriminative tasks, and SDDLs can be taken as a general classifier to deal with target detection (two category classification) and SAR image classification problems.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 41171317, in part by the Key Project of the NSFC under Grant No. 61132008 and in part by the Major Research Plan of the NSFC under Grant No. 61490693.

Author Contributions: Shengli Song proposed the general idea of SAR-HOG and SDDL for SAR ATR and performed the experiments. Bin Xu and Jian Yang reviewed the idea and provided many suggestions and much in-depth analysis. This manuscript was written by Shengli Song.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhao, Q.; Principe, J.C. Support vector machine for SAR automatic target recognition. *IEEE Trans. Aerosp. Electron. Syst.* **2001**, *2*, 643–654.
2. Srinivas, U.; Monga, V.; Raj, R.G. SAR automatic target recognition using discriminative graphical models. *IEEE Trans. Aerosp. Electron. Syst.* **2014**, *1*, 591–606.
3. Dong, G.G.; Kuang, G.Y.; Wang, N.; Zhao, L.J.; Lu, J. SAR target recognition via joint sparse representation of monogenic signal. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *7*, 3316–3328.
4. Mishra, A.K. Validation of PCA and LDA for SAR ATR. *IEEE Tencon* **2008**, *10*, 1–6.
5. Huang, X.Y.; Qiao, H.; Zhang, B. SAR target configuration recognition using tensor global and local discriminant embedding. *IEEE Geosci. Remote Sens. Lett.* **2016**, *2*, 222–226.
6. Zhou, J.X.; Shi, Z.G.; Cheng, X.; Fu, Q. Automatic target recognition of SAR images based on global scattering center model. *IEEE Trans. Geosci. Remote Sens.* **2011**, *10*, 3713–3729.
7. Clemente, C.; Pallotta, L.; Proudler, I.; Maio, A.D.; Soraghan, J.J.; Farina, A. Pseudo-Zernike based multi-pass automatic target recognition from multi-channel SAR. *IET Radar Sonar Navig.* **2015**, *9*, 457–466.
8. Ding, J.; Chen, B.; Liu, H.W.; Huang, M.Y. Convolutional neural network with data augmentation for SAR target recognition. *IEEE Geosci. Remote Sens. Lett.* **2016**, *3*, 1–5.
9. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), San Diego, CA, USA, 25 June 2005.
10. Wright, J.; Yang, A.; Ganesh, A.; Sastry, S.; Ma, Y. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *2*, 210–227.
11. Olshausen, B.A.; Field, D.J. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature* **1996**, *6*, 607–609.
12. Mairal, J.; Ponce, J.; Sapiro, G.; Zisserman, A.; Bach, F. Supervised dictionary learning. In Proceedings of the Neural Information Processing Systems 21, Vancouver, BC, Canada, 8–10 December 2008.

13. Ramirez, I.; Sprechmann, P.; Sapiro, G. Classification and clustering via dictionary learning with structured incoherence and shared features. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010.
14. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *10*, 91–110.
15. Weinmann, M. Visual features: From early concepts to modern computer vision. In *Advanced Topics in Computer Vision, Advances in Computer Vision and Pattern Recognition*; Farinella, G.M., Battiato, S., Cipolla, R., Eds.; Springer-Verlag: London, UK, 2013; pp. 1–34.
16. Tuytelaars, T.; Mikolajczyk, K. Local invariant feature detectors: A survey. *Found. Trends Comput. Graph. Vis.* **2008**, *1*, 177–280.
17. Dai, D.; Yang, W.; Sun, H. Multilevel local pattern histogram for SAR image classification. *IEEE Geosci. Remote Sens. Lett.* **2011**, *3*, 225–229.
18. Cui, S.; Dumitru, C.O.; Datcu, M. Ratio-Detector-Based feature extraction for very high resolution SAR image patch indexing. *IEEE Geosci. Remote Sens. Lett.* **2013**, *9*, 1175–1179.
19. Dellinger, F.; Delon, J.; Gousseau, Y.; Michel, J.; Tupin, F. SAR-SIFT: A SIFT-Like algorithm for SAR images. *IEEE Trans. Geosci. Remote Sens.* **2015**, *1*, 453–466.
20. Yang, F.; Gao, W.; Xu, B.; Yang, J. Multi-Frequency polarimetric SAR classification based on riemannian manifold and simultaneous sparse representation. *Remote Sens.* **2015**, *7*, 8469–8488.
21. Sun, X.; Nasrabadi, N.M.; Tran, T.D. Task-Driven dictionary learning for hyperspectral image classification with structured sparsity constraints. *IEEE Trans. Geosci. Remote Sens.* **2015**, *8*, 4457–4471.
22. Aharon, M.; Elad, M.; Bruckstein, A. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **2006**, *11*, 4311–4322.
23. Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G. Online dictionary learning for sparse coding. *Proc. ICML* **2009**, *6*, 689–696.
24. Mairal, J.; Bach, F.; Ponce, J. Task-driven dictionary learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *4*, 791–804.
25. Gao, S.; Tsang, I.W.H.; Ma, Y. Learning category-specific dictionary and shared dictionary for fine-grained image categorization. *IEEE Trans. Image Process.* **2014**, *2*, 623–634.
26. Zhang, Q.; Li, B. Discriminative K-SVD for dictionary learning in face recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010.
27. Jiang, Z.L.; Lin, Z.; Davis, L.S. Label consistent K-SVD: Learning a discriminative dictionary for recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *11*, 2651–2664.
28. Gangeh, M.J.; Farahat, A.K.; Ghodsi, A.; Kamel, M.S. Supervised dictionary learning and sparse representation—A Review. *Comput. Sci.* **2015**, *2*, 1–56.
29. Keydel, E.; Lee, S.; Moore, J. MSTAR extended operating conditions: A tutorial. *Proc. SPIE* **1997**, *4*, 228–242.
30. Touzi, R.; Lopes, A.; Bousquet, P. A statistical and geometrical edge detector for SAR images. *IEEE Trans. Geosci. Remote Sens.* **1988**, *11*, 764–773.
31. Mairal, J.; Bach, F.; Ponce, J. Sparse modeling for image and vision processing. *Found. Trends Comput. Graph. Vis.* **2014**, *12*, 85–283.
32. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *3*, 1–27.



© 2016 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<http://creativecommons.org/licenses/by/4.0/>).