remote sensing

MDPI

Article

Ship Classification Based on MSHOG Feature and Task-Driven Dictionary Learning with Structured Incoherent Constraints in SAR Images

Huiping Lin, Shengli Song and Jian Yang *

Department of Electronic Engineering, Tsinghua University, Beijing 100084, China; linhp15@mails.tsinghua.edu.cn (H.L.); ssl_1980@126.com (S.S.)

* Correspondence: yangjian_ee@mails.tsinghua.edu.cn; Tel.: +86-10-6279-4726

Received: 9 December 2017; Accepted: 24 January 2018; Published: 27 January 2018

Abstract: In this paper, we present a novel method for ship classification in synthetic aperture radar (SAR) images. The proposed method consists of feature extraction and classifier training. Inspired by SAR-HOG feature in automatic target recognition, we first design a novel feature named MSHOG by improving SAR-HOG, adapting it to ship classification, and employing manifold learning to achieve dimensionality reduction. Then, we train the classifier and dictionary jointly in task-driven dictionary learning (TDDL) framework. To further improve the performance of TDDL, we enforce structured incoherent constraints on it and develop an efficient algorithm for solving corresponding optimization problem. Extensive experiments performed on two datasets with TerraSAR-X images demonstrate that the proposed method, MSHOG feature and TDDL with structured incoherent constraints, outperforms other existing methods and achieves state-of-art performance.

Keywords: ship classification; task-driven dictionary learning; structured incoherent constraints; sparse representation; manifold learning; histogram of oriented gradients (HOG)

1. Introduction

Synthetic aperture radar (SAR) plays an indispensable role in maritime surveillance for its independence of meteorological conditions [1]. As one of the most important steps, ship classification has attracted much interest. Early ship classification research is usually based on simulative SAR ship samples [2]. Subsequently, relatively medium resolution SAR images appeared. Margarit et al. [3] identified ships' classes in ENVISAT 30 m resolution SAR images.

With the launching of high-resolution SAR sensors, SAR images with high resolution make it possible to extract discriminative features. Zhang et al. [4] explored geometric features, focusing on width ratio of minimum enclosing rectangle (MER), and the ratio of ship and non-ship points on the principal axis. Jiang et al. [5] and Xing et al. [6] tried to present a deep understanding of ship properties from superstructure view. 2D comb feature was proposed by Leng et al. [7] based on scattering distributions of ship target. Hierarchical structure was introduced in Lang et al. [8] to further improve the performance of scattering features. However, geometric features may fail when the ships from different classes share similar geometric shape. Superstructure and scattering features can be disturbed by the imaging incident angle. One possible way to solve this problem is to adopt feature selection strategy, and then combine the selected features. Lang et al. [9] proposed a joint feature and classifier selection method, and Chen et al. [10] developed a two-stage feature selection approach. Obviously, the performance of the methods can be limited by the feature set for selection, which means that, if the feature set is not friendly to classification task, we cannot get the desired effect through selection. Another drawback comes from convergence property of the method, as there is a chance that the solution is trapped around the local optima during the selection. With the development of deep learning, researchers employed deep neural networks to extract features for SAR images [11,12]. Bentes et al. [13] proposed a multiple input resolution convolution neural network model (CNN-MR) and demonstrated its effectiveness. However, its application is limited, because we cannot afford the large amount of data required by this method in most cases. Recently, local statistical features were applied to SAR images and showed great potential [11,14]. Song et al. [15] proposed a HOG like feature named SAR-HOG for target classification in SAR images, which improved classification performance greatly. Nevertheless, SAR-HOG feature suffers high dimensionality, which brings great obstacles to feature computation and classifier training. Manifold learning [16–22], as a nonlinear dimensionality reduction method, reveals low dimensional manifolds that are not detected by classic linear method, such as principle component analysis (PCA) [23]. Inspired by this idea, we propose a manifold-learning SAR-HOG feature (MSHOG) by improving SAR-HOG feature for ship classification task and employing manifold learning to implement dimensionality reduction.

With respect to the classifier, sparse representation [24,25] has achieved great success on face recognition [26,27], hyperspectral image classification [28,29] and automatic target recognition (ATR) [30]. Xing et al. [31] applied sparse representation in feature space, and developed a classification method based on the sparse codes, achieving high accuracy in TerraSAR-X images. In the original sparse representation, the dictionary is constructed by simply stacking the whole training samples. Subsequently, dictionary learning methods were proposed to achieve better representation, such as online dictionary learning [32], and K-SVD [33]. Class-specific dictionary was further proposed in Ramirez et al. [34] to capture the difference of categories. The classical dictionary learning approach concerned reconstruction error. As research moved along, researchers realized that a lower reconstruction error in dictionary learning does not necessarily lead to better classification performance and classification performance can even be improved by sacrificing some signal reconstruction performance [35–37]. It was pointed out that better results could be obtained when the dictionary was tuned to the specific task (and not just data) it is intended for. Mairal et al. [38] presented a general formulation for supervised dictionary learning adapted to a wide variety of task, which was referred to as task-driven dictionary learning (TDDL). However, training a universal dictionary for all classes limited the performance of the original TDDL, as shown in Section 5.4. In this paper, we train the dictionary and classifier jointly in TDDL framework. To amplify the difference between classes and suppress the common features, we design class-specific sub-dictionaries, and impose intrinsic incoherent constraints [34,39]. To further encourage sub-dictionaries associated to different classes to be as independent as possible, we enforce direct incoherent constraints on TDDL method, which is proposed for the first time. Finally, corresponding algorithm for solving the optimization problem is developed based on fixed point differentiation and gradient descent (GD) algorithm.

The main contributions of this paper can be summarized as follows: (1) We present a novel local feature named MSHOG for SAR image by fusing improved SAR-HOG with manifold learning. (2) We propose a new dictionary learning algorithm for TDDL with structured incoherent constraints to increase discriminability between different classes of ships. (3) We also describe an optimization algorithm for solving sparse recovery problem with structured incoherent constraints. (4) We show experimentally that the proposed method obtains state-of-art results and has a significantly better performance than other existing methods.

The remainder of this paper is organized as follows. In Section 2, a brief review of SAR-HOG and TDDL is given. In Section 3, we improve SAR-HOG feature, combine the improved one with manifold learning, and propose a novel feature named MSHOG. TDDL with structures incoherent constraints and corresponding algorithm are detailed in Section 4. In Section 5, we show that the proposed method is superior to other ship classification methods in SAR image. Finally, we conclude our work in Section 6.

2. Related Work

In this section, we briefly introduce the original SAR-HOG feature, followed by the modeling and expression of TDDL method. Implementation details can be found in Song et al. [15] and Mairal [38].

2.1. SAR-HOG

SAR-HOG computation includes three steps: gradient computation, orientation binning, and normalization [15].

In gradient computation, ratio-based gradient definition is used, and horizontal and vertical gradient are defined as

$$G_H = \log(\frac{M_{left}}{M_{right}}), \quad G_V = \log(\frac{M_{up}}{M_{down}}) \tag{1}$$

where M denotes the local means on corresponding side of the current pixel. Furthermore, the gradient magnitude and orientation can be computed by:

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

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

Then, orientation binning is employed. Concretely, the SAR image is divide into small regions (called cells). For all the pixels within a cell, the orientations are quantized into a fixed number of angular bins, and the magnitudes are accumulated into orientation bins. The cells are grouped into larger blocks, and the histogram entries of cells in each block are concentrated to be a vector. The vector normalization can be expressed as:

$$\mathbf{v}_i \leftarrow \frac{\mathbf{v}_i}{\max(||\mathbf{v}_i||_2, \varepsilon)} \tag{3}$$

where $\mathbf{v}_i$ denotes the vector corresponding to the ith block; and ε is a small number, whose value is always 0.2-times the mean value of $||\mathbf{v}_i||_2$ in all of the blocks.

The effectiveness of SAR-HOG has been verified in Song et al. [15]. However, its dimensionality is usually very high, especially for large SAR images.

2.2. TDDL

Consider a pair of training samples $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} \in \mathbb{R}^M$ is the some feature extracted from SAR image, and $\mathbf{y} \in \mathbb{R}^K$ is a binary vector representation of corresponding class label. Given some dictionary $\mathbf{D} \in \mathbb{R}^{M \times P}$, $\mathbf{x}$ can be represented as a sparse vector $\boldsymbol{\alpha}(\mathbf{x}, \mathbf{D}) \in \mathbb{R}^P$, defined as the solution of an elastic-net problem [40]:

$$\boldsymbol{\alpha}(\mathbf{x}, \mathbf{D}) = \arg\min_{\alpha \in \mathbb{R}^P} \frac{1}{2}||\boldsymbol{x} - \boldsymbol{D\alpha}||_2^2 + \lambda_1||\boldsymbol{\alpha}||_1 + \frac{\lambda_2}{2}||\boldsymbol{\alpha}||_2^2 \tag{4}$$

where λ_1 and λ_2 are the regularization parameters.

For classification task, TDDL uses the sparse vector $\boldsymbol{\alpha}(\mathbf{x}, \mathbf{D})$ in a classical expected risk minimization formulation:

$$\min_{\text{D,W}} L(\mathbf{D}, \mathbf{W}, \mathbf{x}) = \min_{D,W} f(\mathbf{D}, \mathbf{W}, \mathbf{x}) + \frac{\mu}{2}||\mathbf{W}||_F^2 \tag{5}$$

where $\mathbf{W}$ is the parameter matrix of the classifier, μ is a classifier regularization parameter to avoid the overfitting of classifier [41], and $f(\mathbf{D}, \mathbf{W}, \mathbf{x})$ is a convex function defined as

$$f(\mathbf{D}, \mathbf{W}, \mathbf{x}) = E_{\mathbf{y},\mathbf{x}}[l_s(\mathbf{y}, \mathbf{W}, \boldsymbol{\alpha}(\mathbf{x}, \mathbf{D}))]. \tag{6}$$

In this equation, $E_{\mathbf{y},\mathbf{x}}$ denotes the expectation taken relative to the probability distribution $p(\mathbf{x},\ \mathbf{y})$, l_s is a convex loss function that measures how well one can predict $\mathbf{y}$ by observing $\boldsymbol{\alpha}(\mathbf{x},\mathbf{D})$ given the parameter matrix $\mathbf{W}$, which can be the square, logistic, or hinge loss from SVM [42].

Stochastic gradient descent (SGD) algorithm is used to update the dictionary $\mathbf{D}$ and the parameter matrix $\mathbf{W}$. The update rules are as follows.

$$\begin{cases} \mathbf{D}^{t+1} = \mathbf{D}^t - \rho^t \cdot \frac{\partial L^t}{\partial \mathbf{D}} \\ \mathbf{W}^{t+1} = \mathbf{W}^t - \rho^t \cdot \frac{\partial L^t}{\partial \mathbf{W}} \end{cases} \tag{7}$$

where t is the iteration index and ρ is the step size. The equation for updating $\mathbf{W}$ is straightforward since $L(\mathbf{D},\ \mathbf{W},\ \mathrm{x})$ is both smooth and convex with respect to $\mathbf{W}$. We have

$$\frac{\partial L}{\partial \mathbf{W}} = (\mathbf{W}\boldsymbol{\alpha} - \mathbf{y})\boldsymbol{\alpha}^T + \mu\mathbf{W}. \tag{8}$$

According to the chain rule, we have

$$\frac{\partial L}{\partial \mathbf{D}} = \frac{\partial L}{\partial \boldsymbol{\alpha}} \cdot \frac{\partial \boldsymbol{\alpha}}{\partial \mathbf{D}}. \tag{9}$$

The main difficulty comes from $\partial\boldsymbol{\alpha}/\partial\mathbf{D}$, since the optimization problem in Equation (4) is not smooth [43]. Mairal [38] used fixed point differentiation to solve the problem [44]. The detailed derivation of the algorithm can be found in the Appendix of Mairal et al. [38]. We put the main propositions as follows.

$$\begin{cases} \nabla_{\mathbf{W}} f(\mathbf{D},\ \mathbf{W},\ \mathbf{x}) = E_{\mathbf{y},\mathbf{x}}[\nabla_{\mathbf{W}} l_s(\mathbf{y},\ \mathbf{W},\ \boldsymbol{\alpha})] \\ \nabla_{\mathbf{D}} f(\mathbf{D},\ \mathbf{W},\ \mathbf{x}) = E_{\mathbf{y},\mathbf{x}}[-\mathbf{D}\boldsymbol{\beta}\boldsymbol{\alpha}^T + (\mathbf{x} - \mathbf{D}\boldsymbol{\alpha})\boldsymbol{\beta}^T] \end{cases} \tag{10}$$

where $\boldsymbol{\beta}$ is a vector in $\mathbb{R}^P$ that depends on $\mathbf{y}, \mathbf{x}, \mathbf{W}, \mathbf{D}$ with

$$\boldsymbol{\beta}_{\Lambda^c} = 0 \text{ and } \boldsymbol{\beta}_{\Lambda} = (\mathbf{D}_{\Lambda}^T \mathbf{D}_{\Lambda} + \lambda_2 \mathbf{I})^{-1} \nabla_{\boldsymbol{\alpha}_{\Lambda}} l_s(\mathbf{y}, \mathbf{W}, \boldsymbol{\alpha}) \tag{11}$$

where Λ and Λ_c denote the indices of the nonzero and zero coefficients of $\boldsymbol{\alpha}(\mathbf{x},\mathbf{D})$, respectively.

3. MSHOG Feature

In this section, a novel feature named MSHOG is proposed and analyzed. We introduce the structures of ships first, as a motivation of proposing MSHOG feature. Then, the details of MSHOG feature are described.

The bulk carrier, container ship and oil tanker are the study object in this paper, which constitute approximately 70%–80% of the ships worldwide [4]. The bulk carrier has shorter hull, compared with container ship and oil tanker. The cargo hold is a flat deck with a wide hatch, and the hatch coaming is tall. The container ship has a slender shape, a single plate, and a double or three row of cargo port, and the cabin is grid structure. As for the oil tanker, there is an oil pipeline, along the fore-and-aft line on the upper deck, which is designed for oil handling. Optical images and SAR images of these three ship types are shown in Figure 1.

Unique structures result in unique features, as shown in the second column of Figure 1. For bulk carriers, we can see two bright lines along the fore-and-aft orientation, produced by the tall hatch coaming. The area related to the flat deck is much darker. For container ships, duplicate texture can be noticed, as a result of the grid structure of the cabin. Additionally, container ships maintain high length-width ratio. For oil tankers, we can observe a bright line produced by the pipeline, and darker areas produced by the deck. Classical features, such as geometric feature, structure features and scattering features, are mainly based on the image characteristic mentioned above, obtaining good performance in previous work [4–9]. However, interaction of strong scatters on board and electromagnetic reflection between hull and sea surface blur the imaging, making the effective information in SAR images unavailable, as shown in the third and fourth column of Figure 1. This phenomenon is very common in our datasets. It fails the classic features, promoting us to design a more compact and effective feature. Faced with this dilemma, we use SAR-HOG feature as a prototype and adapt it to our classification task, since SAR-HOG reliably captures the structures of targets in SAR images [15]. Considering the high dimensionality in SAR-HOG feature, we employ manifold learning to achieve dimensionality reduction. Finally, we obtain a compact and effective feature named MSHOG.

MSHOG computation includes four steps: gradient computation, orientation binning, normalization and descriptor blocks, and dimensionality reduction.

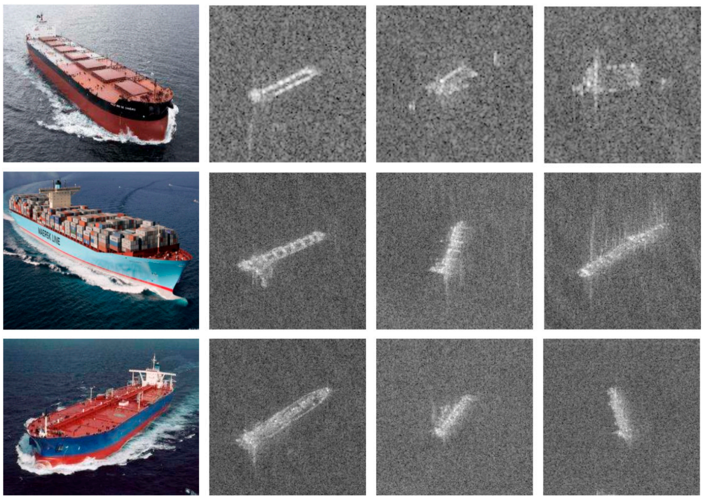


Figure 1. Samples of: bulk carrier (**top**); container ship (**middle**); and oil tanker (**bottom**) in TerraSAR-X images.

3.1. Gradient Computation

Theoretically, speckle in SAR image can be characterized by a multiplicative noise model. In this step, we use the ratio-based gradient definition, introduced in SAR-HOG feature. The horizontal and vertical gradient are defined as in Equation (1). Then, the gradient magnitude $\mathbf{G}_m$ and orientation $\mathbf{G}_\theta$ can be computed by:

$$\mathbf{G}_m = \sqrt{\mathbf{G}_H^2 + \mathbf{G}_V^2} \tag{12}$$

$$\mathbf{G}_\theta = \text{angle}(\frac{\mathbf{G}_V}{\mathbf{G}_H}) \tag{13}$$

where $\mathbf{G}_H$ and $\mathbf{G}_V$ are the gradient matrices in horizontal and vertical direction, respectively. The function angle$(\cdot)$ returns the phase angles, ranging from 0° to 360°. The division, square and square root of the matrices should be understood as element-by-element operations.

3.2. Orientation Binning

Divide the SAR image into small regions (called cells) similar to that in Dalal et al. [45]. We divide the angular space into bins, and then choose the magnitude itself as the weight to vote in the orientation bins corresponding to gradient orientation at each pixel over the cells.

SAR-HOG claims that smaller cells work better, and angular bins should be a bit bigger, which is also identified in MSHOG. The difference between SAR-HOG and MSHOG in orientation binning is that the orientation bins of MSHOG are spaced over 0° to 360° (“signed” gradient) to extract more structure information, while those of SAR-HOG are spaced over 0° to 180° ("unsigned" gradient). The reason the "signed" gradient cannot improve the performance in Song et al. [15] may lie in the wide range of target attitude and background which probably makes the signs of contrast uninformative.

3.3. Normalization and Descriptor Blocks

Several cells constitute a block. The normalization in SAR-HOG is also used in MSHOG, which is described in Equation (3). Next, we concentrate the normalized vectors over blocks and obtain the candidate of MSHOG.

3.4. Dimensionality Reduction

All HOG-like features suffer high dimensionality. For instance, given a SAR image with a size of 128-by-64, we gain a vector with 3780 dimensions with the following parameters setting:

$$\begin{aligned} \text{block\_size} &= 16 \times 16, \text{ block\_stride} = 8, \\ \text{cell\_size} &= 8 \times 8, \quad \text{num\_bins} = 9. \end{aligned} \tag{14}$$

High dimensionality can bring overfitting, when the number of samples is insufficient to support high dimensional classification task. Besides, much computation power is necessary for high dimensionality computation problem. Thus, dimensionality reduction is urgent in small-sample condition. PCA is a classic linear dimensionality reduction method, widely used in image processing. However, PCA has a limited performance when dealing with the data which lie on low dimensional manifolds. That is to say, PCA cannot reveal the nonlinear structure hidden in data. Manifold learning was proposed for this situation. In this paper, maximum variance unfolding (MVU) [16] is employed as the manifold learning method to reveal low dimensional manifolds. PCA aims to preserve Euclidean distances between all pair of vectors, while MVU considers only preserving the geometric properties of local neighbors. MVU claims that low dimensional manifold can be obtained by spectral decomposition of the inner product matrix **K**, which is given by a semidefinite programming (SDP) problem [16]. The SDP problem can be expressed as follows.

$$\boxed{\begin{array}{l} \text{Maxmize trace}(\mathbf{K}) \text{ subject to} \\ (1)\ \mathbf{K} \geq 0. \\ (2)\ \sum_{ij} \mathrm{K}_{ij} = 0 \\ (3)\ \mathrm{K}_{ii} - 2\mathrm{K}_{ij} + \mathrm{K}_{jj} = ||\mathbf{s}_i - \mathbf{s}_j||_2 \text{for all } (i,j) \text{ with} \\ \quad \varphi_{ij} = 1 \end{array}}$$

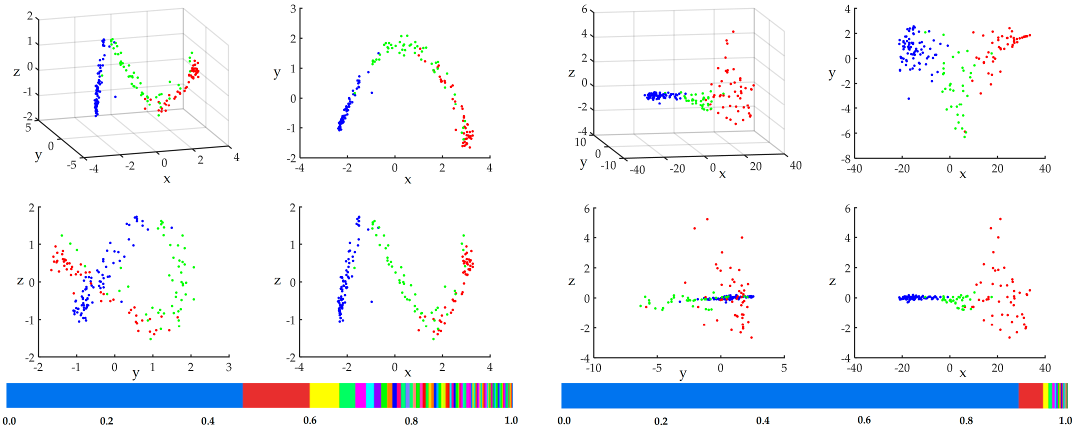
where $\{\mathbf{s}_i\}_{i=1}^{N}$ is the high dimensional input dataset, and $\varphi_{ij} \in \{0,\ 1\}$ indicates whether there is an edge between $\mathbf{s}_i$ and $\mathbf{s}_j$ in the graph formed by pairwise connecting all q-nearest neighbors. The problem above is a classic instance of SDP and can be solved by many mature algorithms as well as several general-purpose toolboxes. We use the CSDP v4.9 toolbox in MATLAB to solve it [46]. The low dimensional output $\{\mathbf{x}_i\}_{i=1}^{N}$ of MVU is given by the eigenvalues and eigenvectors of the inner product matrix **K**. We have

$$x_{\alpha i} = \sqrt{\xi_\alpha V_{\alpha i}} \tag{15}$$

where $x_{\alpha i}$ denotes the αth element of $\mathbf{x}_i$, and $V_{\alpha i}$ denotes the ith element of the αth eigenvector, with eigenvalue ξ_α.

To compare PCA and MVU intuitively, we apply PCA and MVU to TerraSAR-X images (DS1, introduced in Section 5), and visualize the three-dimensional results and eigenvalues in Figure 2 (higher dimensional results cannot be visualized easily). The top and middle panel present the dimensionality reduction results, including the three-dimensional results and their projection on two-dimensional subspaces. The blue, red and green dots represent bulk carrier samples, container ship samples, and oil tanker samples, respectively. We can notice that the distinctiveness of the three-dimensional results in MVU is more pronounced and three-dimensional results in PCA themselves lie on a two-dimensional manifold. Besides, just as the eigenvalue spectrum of the covariance matrix in PCA indicates the dimensionality of a subspace, the eigenvalue spectrum of the inner product matrix in MVU reflects the dimensionality of an underlying manifold. Dominant eigenvalues indicate dominant components in dimensionality. We sort the eigenvalues from the matrices of PCA and MVU in descending order, and visualize them as a fraction of the traces in the bottom panel of Figure 2. The first three eigenvalues in PCA account for 65.71% of the total, whereas the first three eigenvalues in MVU account for 96.15%, which means that MVU more reliably captures the generalized principle components than PCA. Furthermore, we find that the first 20 eigenvalues in MVU account for 99.99% and the first 20 eigenvalues in PCA account for 85.32%. Therefore, we can conclude that MVU outperforms PCA for HOG-like features.

Considering the compactness and effectiveness, we set the dimensionality of the low dimensional manifold as 20. Then, we obtain MSHOG feature by unfolding the low dimensional manifold through MVU.



(**a**) (**b**)

Figure 2. Three-dimensional principle components computed by PCA and three-dimensional embedding computed by MVU on DS1: (**a**) visualization of the three-dimensional principle components, their projections on two-dimensional subspace, and eigenvalues from the matrices of PCA; and (**b**) visualization of the three-dimensional embedding, their projections on two-dimensional subspace, and eigenvalues from the matrices of MVU.

4. TDDL with Structured Incoherent Constraints

4.1. Basic Intuition

TDDL method provides us a supervised dictionary learning framework to learn dictionaries adapted to various tasks instead of only adapted to data reconstruction [38]. For classification task, TDDL method can learn discriminative dictionary and improve classification performance. Ramirez et al. [34] introduce class-specific dictionary learning and incoherent constraints, which can further magnify the differences between classes, compared to a universal dictionary trained on all the samples. The drawback of the method in Ramirez et al. [34] lies in the incoherent constraints, which are unsupervised constraints and are not enough to obtain expected structure on testing set (see Section 5.4). Therefore, we combine the advantages of both methods and impose more effective constraints. First, we design class-specific sub-dictionaries, concentrate them and obtain a big dictionary. Then, intrinsic and direct incoherent constraints are imposed. Finally, we learn the dictionary and classifier jointly in TDDL framework.

4.2. Formulation

Denote training samples of k classes as $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_N] \in \mathbb{R}^{M \times N}$, where $\mathbf{x}_i \in \mathbb{R}^M$ is a SAR image feature. Denote the dictionary as $\mathbf{D} = [\mathbf{D}_1, \mathbf{D}_2, \cdots, \mathbf{D}_l, \cdots, \mathbf{D}_k] \in \mathbb{R}^{M \times P}$ ($P = P_1 + P_2 + \cdots + P_k$), where $\mathbf{D}_l = [\mathbf{d}_1^l, \mathbf{d}_2^l, \ldots, \mathbf{d}_j^l, \ldots, \mathbf{d}_{P_l}^l] \in \mathbb{R}^{M \times P_l}$ is the lth sub-dictionary and $\mathbf{d}_j^l$ is the jth atom with M dimensions in lth sub-dictionary. The sample $\mathbf{x}_i$ can be represented as a sparse code $\boldsymbol{\alpha}_\mathbf{i}(\mathbf{x}, \mathbf{D}) \in \mathbb{R}^P$ by solving the optimization:

$$\boldsymbol{\alpha}_i(\mathbf{x}, \mathbf{D}) = \underset{\alpha_i \in \mathbb{R}^P}{\arg\min} ||\mathbf{x}_i - \mathbf{D}\boldsymbol{\alpha}_i||_2^2 + \lambda_1 ||\boldsymbol{\alpha}_i||_1 + \frac{1}{2}\lambda_2 ||\boldsymbol{\alpha}_i||_2^2 \tag{16}$$

where λ_1 and λ_2 are the regularization parameters. λ_1 controls the sparsity of $\boldsymbol{\alpha}_i(\mathbf{x}, \mathbf{D})$, and λ_2 is to avoid poor convergence of algorithm.

The training samples can be represented by the linear combination of the sub-dictionaries, i.e.,

$$\mathbf{x}_i \approx \mathbf{D}_1\boldsymbol{\alpha}_1^i + \mathbf{D}_2\boldsymbol{\alpha}_2^i + \cdots \mathbf{D}_l\boldsymbol{\alpha}_l^i + \cdots + \mathbf{D}_k\boldsymbol{\alpha}_k^i \tag{17}$$

where $\boldsymbol{\alpha}_l^i \in \mathbb{R}^{P_l}$ denotes the lth sub-sparse-code corresponding to the sub-dictionary $\mathbf{D}_l$. For classification task, it is desirable that class-specific samples are encoded as class-specific sparse code on class-specific dictionary. That means that the samples from class l can be almost completely represented by $\boldsymbol{\alpha}_l^i$ with respect to $\mathbf{D}_l$, i.e.,

$$\mathbf{x}_i^l \approx \mathbf{D}_l\boldsymbol{\alpha}_l^i \tag{18}$$

where $\mathbf{x}_i^l$ denotes the SAR image samples from class l.

1. Intrinsic Constraints

Two sub-dictionaries $\mathbf{D}_{l_1}$ and $\mathbf{D}_{l_2}$ are assumed to be orthogonal, i.e.,

$$\mathbf{D}_{l_1}^T\mathbf{D}_{l_2} = \mathbf{0},\ 1 \le l_1 \le k,\ 1 \le l_2 \le k, \text{ and } l_1 \ne l_2 \tag{19}$$

where $\mathbf{0}$ denotes the matrix of all zeros, T denotes the transposition. Then, the linear space spanned by the atoms of $\mathbf{D}_{l_1}$ is orthogonal with the space spanned by the atoms of $\mathbf{D}_{l_2}$. In other words, if sample $\mathbf{x}_i \in \mathbb{R}^M$ can be represented by a linear combination of atoms of $\mathbf{D}_{l_1}$, the projection of $\mathbf{x}_i$ on the space of $\mathbf{D}_{l_2}$ is zero. Therefore, we can achieve the purpose in Equation (18) by restricting the coherence between class-specific sub-dictionaries to

$$\min_{D \in \mathbb{R}^{M \times P}} \sum_{l=1}^{k} ||\mathbf{D}_l^T\mathbf{D}_{-l}||_F^2. \tag{20}$$

2. Direct Constraints

The intrinsic constraints are unsupervised constraints. To explore the priori information in class labels, we propose direct constraints as supervised constraints. The two kinds of constraints complement each other, obtaining the ideal structure in Equation (18) and improving the classification performance significantly.

Basically, the update of the dictionary should be driven by the minimization of the distance between the ideal structure of sparse codes and the current structure of those. Since the sparsity of sparse codes is guaranteed by Equation (16), it is unnecessary to control the sparsity of $\boldsymbol{\alpha}^{i}{}_{-l}$ via *L1*-norm. In addition, the smoothness of *L2*-norm can benefit the differentiation in the update of the dictionary. Therefore, *L2*-norm, as an approximation of *L1*-norm, is chosen to characterize the distance. We have:

$$\min_{\mathrm{D}} ||\boldsymbol{\alpha}^{i}{}_{-l}||_2^2 \tag{21}$$

where and $\boldsymbol{\alpha}^{i}{}_{-l}$ denotes the sub-sparse-vector by removing $\boldsymbol{\alpha}_l^i$ from $\boldsymbol{\alpha}_i$. For the brevity of expression, let $\mathbf{s}_i \in \mathbb{R}^P$ denote the supervising vector for sample $\mathbf{x}_i \in \mathbb{R}^M$ from class l, i.e.,

$$\mathbf{s}_i = \begin{bmatrix} \mathbf{s}_{i1}^T & \mathbf{s}_{i2}^T & \cdots & \mathbf{s}_{ij}^T & \cdots & \mathbf{s}_{ik}^T \end{bmatrix}^T \tag{22}$$

where $\mathbf{s}_{ij} = (1 - \delta_{jl}) \cdot \mathbf{1}_{P_j \times 1} \in \mathbb{R}^{P_j} (1 \leq j \leq k)$ is a sub-vector of $\mathbf{s}_i$. $\mathbf{1}_{P_j \times 1}$ is the vector of all ones whose size is $P_j \times 1$, and δ_{jl} is defined as:

$$\delta_{jl} = \begin{cases} 1 & \text{if } j = l \\ 0 & \text{else.} \end{cases} \tag{23}$$

Thus, Equation (21) can be rewritten as:

$$\min_{\mathrm{D}} ||\mathbf{s}_i . * \boldsymbol{\alpha}_i||_2^2, \ s.t.\ \boldsymbol{\alpha}_i(\mathbf{x}, \mathbf{D}) = \arg\min_{\mathrm{z}} ||\mathbf{x}_i - \mathbf{D}\mathbf{z}||_2^2 + \lambda_1 ||\mathbf{z}||_1 + \frac{1}{2}\lambda_2 ||\mathbf{z}||_2^2 \tag{24}$$

where $.*$ denotes element-by-element multiplication.

Given the training samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_i, \cdots, \mathbf{x}_N] \in \mathbb{R}^{M \times N}$ and corresponding label vectors $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_i, \cdots, \mathbf{y}_N] \in \mathbb{R}^{k \times N}$, we update the dictionary $\mathbf{D}$ and the classifier's parameters $\mathbf{W}$ jointly in TDDL framework with the constraints introduced above. Mathematically, the objective function can be formulated as follows:

$$\begin{aligned} \min_{\mathrm{D,W}} L(\mathbf{D}, \mathbf{W}, \mathbf{X}) = \min_{\mathrm{D,W}} \tfrac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{A}||_F^2 + \tfrac{\mu}{2}||\mathbf{W}||_F^2 + \tfrac{\eta_1}{2}\sum_{l=1}^{k} \tfrac{1}{P_l^2}||\mathbf{D}_l^T\mathbf{D}_l - \mathbf{I}_{P_l}||_F^2 \\ + \tfrac{\eta_2}{2}\sum_{l=1}^{k} \tfrac{1}{2P_l(P-P_l)}||\mathbf{D}_l^T\mathbf{D}_{-l}||_F^2 + \tfrac{\nu}{2}||\mathbf{S}. * \mathbf{A}||_F^2 \end{aligned} \tag{25}$$

where μ, η_1, η_2 and ν are the regularization parameters; $\mathbf{W} \in \mathbb{R}^{k \times P}$ is the parameters of the linear classifier; $\mathbf{D}_{-l}$ is denoted as the sub-dictionaries by removing $\mathbf{D}_l$ from $\mathbf{D}$; $\mathbf{I}_{P_l}$ is an identity matrix whose size is $P_l \times P_l$; $\mathbf{S} = [\mathbf{s}_1,\ \mathbf{s}_2, \ldots, \mathbf{s}_i, \ldots, \mathbf{s}_N] \in \mathbb{R}^{P \times N}$ is the supervising matrix associated with training samples' class labels; and $\mathbf{A} \in \mathbb{R}^{P \times N}$ is the solution of the following problem:

$$\mathbf{A} = \underset{\mathrm{A}}{\operatorname{argmin}} ||\mathbf{X} - \mathbf{D}\mathbf{A}||_F^2 + \lambda_1 \sum_{i=1}^{N} ||\boldsymbol{\alpha}_i||_1 + \frac{\lambda_2}{2}||\mathbf{A}||_F^2 \tag{26}$$

where $\boldsymbol{\alpha}_i$ is the ith column of $\mathbf{A} \in \mathbb{R}^{P \times N}$, λ_1 and λ_2 are the regularization parameters. In Equation (25), the term $||\mathbf{Y} - \mathbf{W}\mathbf{A}||_F^2$ describes the classification error, the term $||\mathbf{W}||_F^2$ is to avoid overfitting of the classifier, the term $||\mathbf{D}_l^T\mathbf{D}_l - \mathbf{I}_l||_F^2$ as self-incoherence is to stabilize the learned dictionary for each class [47], term $||\mathbf{D}_l^T\mathbf{D}_{-l}||_F^2$ as cross-incoherence is to enforce class-specific sub-dictionaries incoherency, and the term $||\mathbf{S}. * \mathbf{A}||_F^2$ is to enforce the sparse codes to ideal structure. The coefficients $1/P_l^2$ and $1/2P_l(P - P_l)$ in Equation (25) are to reduce the influence of the sub-dictionary size and make the learned dictionary more stable for classification, which was introduced by Gao et al. [48].

Compared with Song et al. [15] and Remirez [34], we learn the dictionary and the classifier's parameter in a task-driven way. Unlike the original TDDL [38], we take structured incoherent constraints to obtain abundant discriminability. We take TDDL-SIC as shorthand for TDDL with structured incoherent constraints in the following. Related experiments will be performed and discussed in Section 5.

4.3. Optimization

The objective function $L(\mathbf{D}, \mathbf{W}, \mathbf{X})$ in Equation (25) can be further represented by two parts, L_1 and L_2, which are defined as follows:

$$L_1 = \frac{1}{2}||\mathbf{Y} - \mathbf{W}\mathbf{A}||_F^2 + \frac{\mu}{2}||\mathbf{W}||_F^2 + \frac{\nu}{2}||\mathbf{S}. * \mathbf{A}||_F^2 \tag{27}$$

$$L_2 = \frac{\eta_1}{2}\sum_{l=1}^{k}\frac{1}{P_l^2}||\mathbf{D}_l^T\mathbf{D}_l - \mathbf{I}_{P_l}||_F^2 + \frac{\eta_2}{2}\sum_{l=1}^{k}\frac{1}{2P_l(P-P_l)}||\mathbf{D}_l^T\mathbf{D}_{-l}||_F^2. \tag{28}$$

Gradient descent algorithm is used to update the classifier's parameter $\mathbf{W}$ and the dictionary $\mathbf{D}$. It is simple to obtain the gradient with respect to $\mathbf{W}$, i.e.,

$$\frac{\partial L}{\partial \mathbf{W}} = (\mathbf{W}\mathbf{A} - \mathbf{Y})\mathbf{A}^T + \mu\mathbf{W}. \tag{29}$$

We apply chain rule to compute the gradient with respect to the dictionary $\mathbf{D}$:

$$\frac{\partial L}{\partial \mathbf{D}} = \frac{\partial L_1}{\partial \mathbf{A}}\frac{\partial \mathbf{A}}{\partial \mathbf{D}} + \frac{\partial L_2}{\partial \mathbf{D}}. \tag{30}$$

Since there is no explicit expression of $\mathbf{D}$ for the sparse codes $\mathbf{A}$, it is difficult to compute the derivative $\partial\mathbf{A}/\partial\mathbf{D}$. Applying fixed point differentiation [38] to Equation (26), we have:

$$\left.\frac{\partial||\mathbf{X} - \mathbf{D}\mathbf{A}||_F^2}{\partial \mathbf{A}}\right|_{\mathbf{A}=\hat{\mathbf{A}}} = -\lambda_1\sum_{i=1}^{N}\frac{\partial||\boldsymbol{\alpha}_i||_1}{\partial \mathbf{A}} - \left.\frac{\lambda_2}{2}\frac{\partial||\mathbf{A}||_F^2}{\partial \mathbf{A}}\right|_{\mathbf{A}=\hat{\mathbf{A}}} \tag{31}$$

where $\hat{\mathbf{A}}$ is the optimal point of Equation (26). Then, we have:

$$2\mathbf{D}^T(\mathbf{X} - \mathbf{D}\mathbf{A})\Big|_{\mathbf{A}=\hat{\mathbf{A}}} = \lambda_1 \cdot \mathrm{sign}(\mathbf{A}) + \lambda_2\mathbf{A}|_{\mathbf{A}=\hat{\mathbf{A}}}. \tag{32}$$

We compute the differentiation when the element of $\mathbf{A}$ are non-zeros, since the function sign$(\cdot)$ is non-differentiable at zero points.

$$\frac{\partial \mathbf{A}_\Lambda}{\partial \mathrm{D}_{mn}} = (\mathbf{D}_\Lambda^T\mathbf{D}_\Lambda)^{-1}(\frac{\partial \mathbf{D}_\Lambda^T\mathbf{X}}{\partial \mathrm{D}_{mn}} - \frac{\partial \mathbf{D}_\Lambda^T\mathbf{D}_\Lambda}{\partial \mathrm{D}_{mn}}\mathbf{A}_\Lambda) \tag{33}$$

$$\frac{\partial \mathbf{A}_{\Lambda^c}}{\partial \mathrm{D}_{mn}} = 0 \tag{34}$$

where Λ is defined as the active set of $\mathbf{A}$,

$$\Lambda = \{i : \ \mathrm{vec}(\mathbf{A})_\mathrm{i} \neq 0, i \in \{1, \cdots, NP\}\}. \tag{35}$$

vec$(\cdot)$ denotes vectorization operator. From Equation (27), we have:

$$\frac{\partial L_1}{\partial \mathbf{A}} = \mathbf{W}^T(\mathbf{W}\mathbf{A} - \mathbf{Y}) + \nu\mathbf{S}. * \mathbf{A}. \tag{36}$$

By uniting Equations (33), (34) and (36), we reach the analytic form of $\partial L_1/\partial \mathbf{D}$.

Define $\mathbf{E} = [\mathbf{E}_1, \cdots, \mathbf{E}_l, \cdots, \mathbf{E}_k] \in \mathbb{R}^{M \times P}$, and $\mathbf{F} = [\mathbf{F}_1, \cdots, \mathbf{F}_l, \cdots, \mathbf{F}_k] \in \mathbb{R}^{M \times P}$. The lth sub-matrix of $\mathbf{E}$ and $\mathbf{F}$ is defined as follows:

$$\mathbf{E}_l = \frac{1}{2P_l^2} \frac{\partial ||\mathbf{D}_l^T \mathbf{D}_l - \mathbf{I}_{P_l}||_F^2}{\partial \mathbf{D}_l^T \mathbf{D}_l} \frac{\partial \mathbf{D}_l^T \mathbf{D}_l}{\partial \mathbf{D}_l} \tag{37}$$

$$\mathbf{F}_l = \frac{1}{2P_l(P-P_l)} \frac{\partial ||\mathbf{D}_l^T \mathbf{D}_{-l}||_F^2}{\partial \mathbf{D}_l}. \tag{38}$$

Obviously, $\partial L_2/\partial \mathbf{D}$ can be expressed as

$$\frac{\partial L_2}{\partial \mathbf{D}} = \eta_1 \mathbf{E} + \eta_2 \mathbf{F}. \tag{39}$$

Expanding $\partial \mathbf{D}_l^T \mathbf{D}_l / \partial \mathbf{D}_l$ in Equation (37), we have

$$\frac{\partial \mathbf{D}_l^T \mathbf{D}_l}{\partial \mathrm{D}_{l_{mn}}} = (\mathbf{D}_l^T \mathbf{U}_l^{mn}) + (\mathbf{D}_l^T \mathbf{U}_l^{mn})^T \tag{40}$$

where $\mathbf{U}_l^{mn}$ is defined as

$$\mathbf{U}_l^{mn} = \Big\{ \mathbf{U} \in \mathbb{R}^{M \times P_l} \Big| u_{ij} = \delta_{mi}\delta_{nj},\ \forall i,\ j,\ 1 \le i \le M,\ 1 \le j \le P_l \Big\}. \tag{41}$$

Based on Equations (37), (38), (40) and (41), we obtain the analytic form of $\mathbf{E}_l$ and $\mathbf{F}_l$ as follows:

$$\mathbf{E}_l = \frac{1}{P_l^2}(\mathbf{B}_l - 2\mathbf{D}_l) \tag{42}$$

$$\mathbf{F}_l = \frac{1}{P_l(P-P_l)} \mathbf{D}_{-l} \mathbf{D}_{-l}^T \mathbf{D}_l \tag{43}$$

where $\mathbf{B}_l$ is defined as

$$\mathbf{B}_l = \{ \mathbf{B} \in \mathbb{R}^{M \times P_l} \Big| b_{ij} = \mathrm{sum}(\mathbf{D}^T\mathbf{D} .* (\mathbf{D}^T \mathbf{U}_l^{ij} + (\mathbf{D}^T \mathbf{U}_l^{ij})^T)),\ \forall i,\ j,\ 1 \le i \le M,\ 1 \le j \le P \}. \tag{44}$$

The function $\mathrm{sum}(\cdot)$ computes the sum of matrix elements. Finally, we reach the analytic form of $\partial L_2/\partial \mathbf{D}$.

Here, we conclude the derivation results as follows:

$$\frac{\partial L}{\partial \mathbf{D}} = -\mathbf{D}\boldsymbol{\beta}\mathbf{A}^T + (\mathbf{X} - \mathbf{D}\mathbf{A})\boldsymbol{\beta}^T + \eta_1 \mathbf{E} + \eta_2 \mathbf{F} \tag{45}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{P \times N}$ is defined as

$$\mathrm{vec}(\boldsymbol{\beta})_{\Lambda^c} = 0 \tag{46}$$

$$\mathrm{vec}(\boldsymbol{\beta})_{\Lambda} = (\mathbf{I}_N \otimes \mathbf{D}^T\mathbf{D} + \lambda_2 \mathbf{I}_{NP})_{\Lambda,\Lambda}^{-1} \mathrm{vec}(\mathbf{W}^T(\mathbf{W}\mathbf{A} - \mathbf{Y}) + \nu \mathbf{S} .* \mathbf{A})_{\Lambda} \tag{47}$$

and $\mathbf{E}$ and $\mathbf{F}$ are given by Equations (42), (43) and (44).

We summarize the overall optimization for TDDL-SIC in Algorithm 1.

```
Algorithm 1: Gradient descent algorithm for TDDL with structured incoherent constraints
Input:
    The feature vectors of $k$ classes' samples: $\mathbf{X} \in \mathbb{R}^{M \times N}$, the class labels: $\mathbf{Y} \in \mathbb{R}^{k \times N}$.
    Initial dictionary $\mathbf{D} \in \mathbb{R}^{M \times P}$ and classifier $\mathbf{W} \in \mathbb{R}^{k \times P}$.
    Regularization parameter $\lambda_1, \lambda_2, \eta_1, \eta_2, \nu \in \mathbb{R}$.
    Number of iterations $T$, parameter $t_0$, learning rate parameter $\rho$
Repeat:
    for $t = 0$ to $T$ do
        Compute sparse code A according to Equation (26).
        Compute the active set $\Lambda$ according to Equation (35).
        Compute the matrix $\boldsymbol{\beta}$, E and F according to Equations (42), (43), (44), (46) and (47).
        Choose the learning rate
    $\rho_t \leftarrow \min(\rho, \rho t_0/t)$
    normally, set $t_0 = T/10$.
        Update the dictionary D and classifier W
    $\mathbf{W} \leftarrow \mathbf{W} - \rho_t((\mathbf{WA} - \mathbf{Y})\mathbf{A}^T + \mu\mathbf{W})$
    $\mathbf{D} \leftarrow \mathbf{D} - \rho_t(-\mathbf{D}\boldsymbol{\beta}\mathbf{A}^T + (\mathbf{X} - \mathrm{DA})\boldsymbol{\beta}^T + \eta_1\mathbf{E} + \eta_2\mathbf{F})$
    and normalize each column of D with respect to $L2$-norm.
    end for
Output: D and W
```

Initialization strategy for Algorithm 1: We initialize the dictionary **D** via unsupervised dictionary learning method. Concretely, given feature vectors of lth class $\mathbf{X}_l$, we compute the class-specific sub-dictionary by solving

$$\begin{array}{l} \min\limits_{\mathbf{D}_l,\mathbf{A}} ||\mathbf{X}_l - \mathbf{D}_l\mathbf{A}||_F^2 + \lambda_1 \sum\limits_{i=1}^{N} ||\boldsymbol{\alpha}_i||_1 + \frac{\lambda_2}{2}||\mathbf{A}||_F^2 \\ \text{s.t. } ||\mathbf{d}_j^l||_F^2 = 1,\ \forall j,\ l,\ 1 \leq j \leq P_l,\ 1 \leq l \leq k \end{array} \tag{48}$$

where $\mathbf{D}_l$ denotes the class-specific sub-dictionary, $\mathbf{d}_j^l$ is the jth atom in $\mathbf{D}_l$, and $\boldsymbol{\alpha}_i$ is the ith column of **A**. Then, we concentrate the sub-dictionaries and obtain the initialization of **D**. Based on the initial dictionary **D**, we compute the initial classifier by solving

$$\min_{\mathrm{W}} ||\mathbf{Y} - \mathbf{WA}||_2^2 + \frac{\mu}{2}||\mathbf{W}||_F^2 \tag{49}$$

where **Y** is the class labels, and **A** is the solution of Equation (26). Hence, with the strategy above, (**D**, **W**) is obtained as a pair of input parameters to initialize Algorithm 1. The SPAM software [32] is used to implement the initialization in TDDL-SIC.

Variants of Algorithm 1: For large scale classification task, one can speed up of our algorithm with a minibatch strategy—that is, by drawing $N_{batchsize} < N$ samples randomly at each iteration instead of N samples. It should be noted that too small batch-size will affect the convergence of the algorithm. In practice, the value $N_{batchsize} = 50$ has given good results in our experiments.

5. Experiments and Discussions

To demonstrate the effectiveness of the proposed method, we design and perform experiments on two SAR datasets. The first dataset (DS1), which is provided by Xiangwei Xing and Kefeng Ji [6,10,31], consists of image chips collected from six TerraSAR-X images of Hong Kong. The image acquisition dates vary from 13 May 2008 to 4 December 2010. The other dataset (DS2) is collected from ten TerraSAR-X images of Zhoushan, Zhejiang Province. The image acquisition dates vary from 1 May 2015 to 21 July 2017. The image details of DS1 and DS2 are shown in Table 1. With the aid of automatic identification system (AIS), the samples in two datasets are labeled precisely. DS1 contains 150 bulk carriers (BC), 50 container ships (CS) and 50 oil tankers (OT). The number of three types of ships in DS2 are all 150.

Table 1. The image details of DS1 and DS2.

Dataset	Sensor	Model	Polarization	Azimuth Resolution	Range Resolution
DS1	TerraSAR-X	stripmap	VV	2.0 m	1.0 m
DS2	TerraSAR-X	spotlight	VV	1.0 m	1.0 m

In the following, we first introduce the experiment setup and detail the parameter setting. Then, the effectiveness of MSHOG feature will be discussed in Section 5.3. We demonstrate the effectiveness of TDDL-SIC based on MSHOG feature in Section 5.4. In Sections 5.5 and 5.6, we evaluate the classification performance of the proposed method on two datasets. Comparison between our method and other existing methods will also be included. All experiments are performed by MATLAB, using a common PC with the Intel Core i7 processor with a 3.60-GHz main frequency and 8.00-Gb main memory.

5.1. Experiment Setup

We randomly divide the dataset into two parts: training set and testing set. The numbers of ship samples in training set and testing set of DS1 and DS2 are listed in Table 2.

According to Lang et al. [9], image processing is necessary for feature extraction. In this paper, we also take the preprocessing method in Lang et al. [9], which includes three steps. In the first step, we transform SAR images (see Figure 3a) into binary images, estimate the angles through Radon transform, and rotate the ships to horizontal direction (see Figure 3b). In the second step, we calculate the accumulations in the x and y directions (see Figure 3c,e), set empirical thresholds to locate the positions of the bounding boxes, and obtain the bounding boxes around the ships (see Figure 3d). The size of bounding boxes is set uniformly as 30 pixels × 200 pixels. Finally, the ship image chips are cut along the bounding boxes, which are used for subsequent experiments (see Figure 3f).

Table 2. The numbers of ship samples in training set and testing set.

	DS1			DS2		
	BC	CS	OT	BC	CS	OT
Training Set	75	25	25	75	75	75
Testing Set	75	25	25	75	75	75

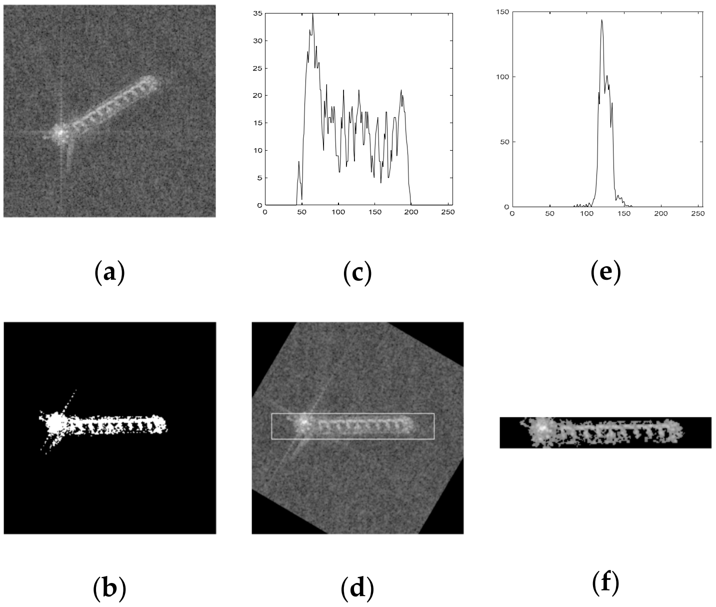


Figure 3. Image preprocessing by the three-step segmentation: (**a**) the original SAR image; (**b**) the binary image after the binaryzation and rotation; (**c**) the cumulations curve along x direction; (**d**) the bounding box around the ship; (**e**) the cumulations curve along y direction; and (**f**) the SAR image chip.

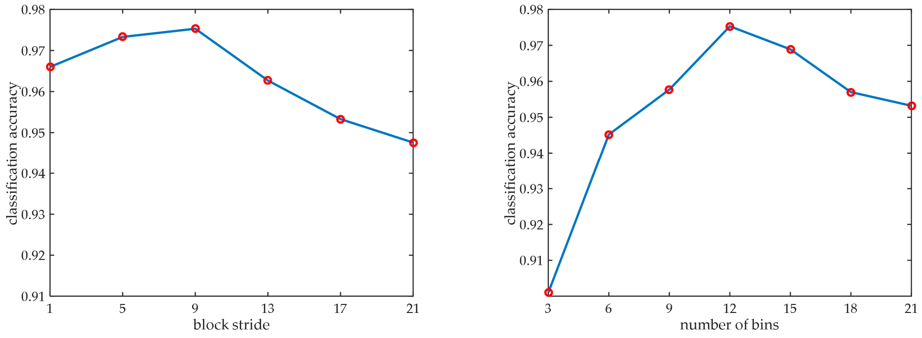
5.2. Parameter Setting

The parameters in the proposed method include the parameters of MSHOG and the parameters of TDDL-SIC. Performing cross validation on the parameters would be cumbersome and we optimize the parameters according to following strategies. The parameters in MSHOG actually depend on SAR image resolution and the sizes of targets in the images. We refer to the method in Song et al. [15] to optimize the parameters of MSHOG feature. We set the parameters according to parameters in SAR-HOG [15], and then optimize each parameter by keeping others fixed. Although the method in Song et al. [15] may be trapped by local optimum solution, it does work in our experiments. For the parameters of TDDL-SIC, we use a few simple heuristics to reduce the search space, which are used in many TDDL-like methods [38,49,50].

We list the parameters used in MSHOG feature in Table 3. Figure 4a,b gives the results of classification accuracy versus the block stride and number of bins, respectively. It shows that relatively small block stride can improve the classification accuracy slightly and the classification accuracy reaches its maximum when the number of bins is set as 12. In addition, Figure 5 shows the optimal cell size and block size. The dimension of MSHOG feature is given by the eigenvalues of the inner product matrix, as mentioned in Section 3.4.

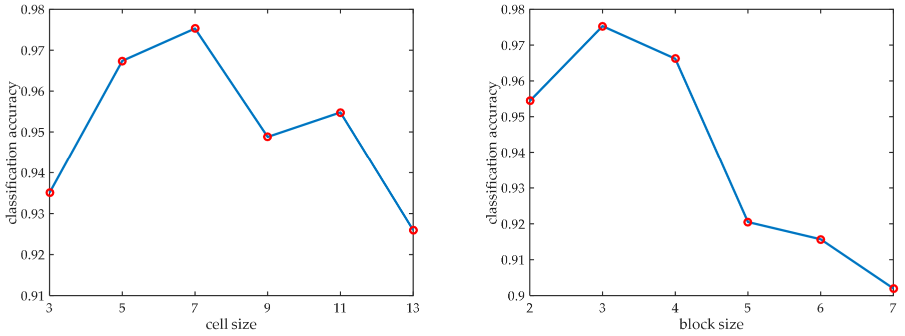
Table 3. The parameters used in MSHOG feature.

Parameter	Cell Size (Pixels)	Block Size (Cells)	Block Stride (Pixels)	Number of Bins	Dimensions
Value	7×7	3×3	9	12	20



(**a**) (**b**)

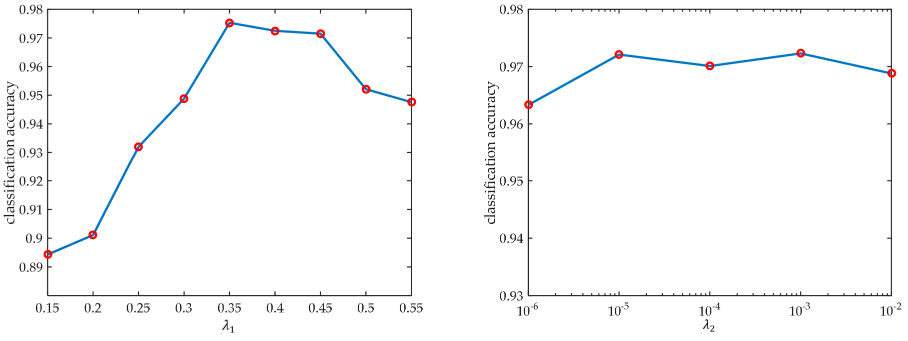
Figure 4. The classification accuracy versus the block stride and number of bins: (**a**) the classification accuracy versus the block stride, where we set the block stride to be 1, 5, 9, 13, 17 and 21; and (**b**) the classification accuracy versus the number of bins, where we set the number of bins to be 3, 6, 9, 12, 15, 18 and 21.



(**a**) (**b**)

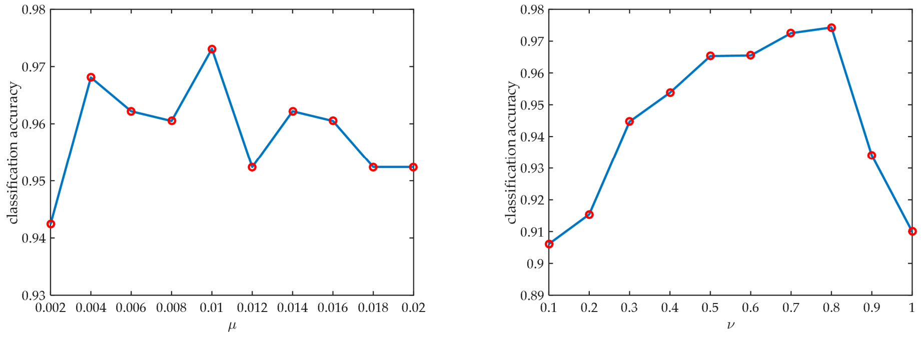
Figure 5. The classification accuracy versus the cell size and block size: (**a**) the classification accuracy versus the cell size, where we set the cell size to be 3, 5, 7, 9, 11 and 13; and (**b**) the classification accuracy versus the block size, where we set the block size to be 2, 3, 4, 5, 6 and 7. When the block size is bigger than the image size, we decrease the corresponding cell size to adapt to the image size.

For TDDL-SIC method, we try the parameters $\lambda_1 = 0.35 + 0.05j$, with $j \in \{-4, \ldots, 4\}$, and λ_2 is chosen in $\{10^{-2}, 10^{-3}, \ldots, 10^{-6}\}$. The candidate parameters of μ and ν are $\{0.002, 0.004, \ldots, 0.02\}$ and $\{0.1, 0.2, \ldots, 1\}$, respectively. Additionally, the candidate parameters of η_1 and η_2 are $\{0, 0.1, \ldots, 1\}$ and $\{0, 0.025, \ldots 0.25\}$, respectively. The candidate sub-dictionary sizes are from 4 to 11 atoms. The choice of the parameters depends on the classification performance. Figure 6 presents the classification performance versus the regularization parameter λ_1 and λ_2. The classification performance versus the regularization parameter μ and ν is demonstrated in Figure 7. Moreover, Figures 8a,b and 9 record the classification performance as the regularization parameter η_1, η_2 and the sub-dictionary size P_l change, respectively. Based on the figures, we obtain optimal parameters in TDDL-SIC, which are listed in Table 4.



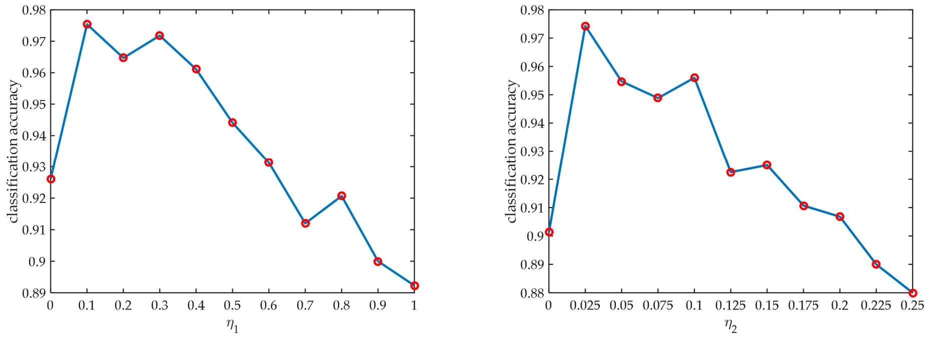
(**a**) (**b**)

Figure 6. The classification accuracy versus the regularization parameters λ_1 and λ_2: (**a**) the classification accuracy versus λ_1, where we set λ_1 to be 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5 and 0.55; and (**b**) the classification accuracy versus λ_2, where we set λ_2 to be 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3} and 10^{-2}.



(**a**) (**b**)

Figure 7. The classification accuracy versus the regularization parameters μ and ν: (**a**) the classification accuracy versus μ, where we set μ to be 0.002, 0.004, . . . , 0.018 and 0.02; and (**b**) the classification accuracy versus ν, where we set ν to be 0.1, 0.2, . . . , 0.9 and 1.



(**a**) (**b**)

Figure 8. The classification accuracy versus the regularization parameters η_1 and η_2: (**a**) the classification accuracy versus η_1, where we set η_1 to be 0, 0.1, . . . , 0.9 and 1; and (**b**) the classification accuracy versus η_2, where we set η_2 to be 0, 0.025, . . . , 0.225 and 0.25.

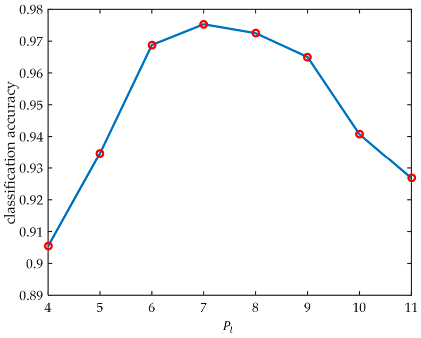


Figure 9. The classification accuracy versus the sub-dictionary size P_l. We set the sub-dictionary size P_l to be 4, 5, 6, 7, 8, 9, 10 and 11.

Table 4. The parameters used in TDDL-SIC.

Parameter	λ_1	λ_2	μ	ν	η_1	η_2	P_l
Values	0.35	0.001	0.01	0.8	0.1	0.025	**7**

5.3. Effectiveness of MSHOG Feature

We evaluate the effectiveness of MSHOG feature by using SVM as the baseline classifier. MSHOG feature is compared to the features in previous work, including 2D comb feature (2DC) [7], selected features (SF) [10], superstructure scattering features (SS) [5], local RCS density features associated with geometric features (LRCSG) [31], and SAR-HOG feature with PCA dimensionality reduction (PSHOG). The parameters of these features are set up according to the original paper. Please refer to [5,7,10,31] for more details. We conduct experiments of each feature twenty times to reduce the disturbance of stochastic factors in single experiment and present the average performance of each feature.

Table 5 illustrates the comparison on classification accuracy of 2DC, SF, SS, LRCSG, PSHOG, and MSHOG on DS1. We can see that all the methods provide high classification accuracy of bulk carriers. The classification accuracy of oil tankers using MSHOG achieves 92.3%, which is significantly higher than that of oil tankers using 2DC, SF, SS, LRCSG and PSHOG, indicating that MSHOG feature can capture the discriminability of oil tankers, and distinguish oil tankers from bulk carriers and container ships. It is also observed that MSHOG feature outperforms the other methods and yields the best overall classification accuracy as high as 94.8%. This performance validates that the MSHOG feature is an effective feature for ship classification.

Furthermore, Table 6 illustrates the comparison on classification accuracy of 2DC, SF, SS, LRCSG, PSHOG, and MSHOG on DS2. The overall classification accuracy of MSHOG feature is 91.6%, which is higher by 4.2%, 4.1%, 7.1%, 1.6%, and 2.1% than that of 2DC, SF, SS, LRCSG and PSHOG, respectively. The improvement of the classification accuracy of oil tankers using MSHOG is 3.4% at least, compared to using other methods. For the classification accuracy of bulk carriers and container ships, MSHOG shows pretty good performance. Thus, we can conclude that MSHOG feature improves ship classification performance and has remarkable merit.

Table 5. Comparison on classification accuracy (%) of 2DC, SF, SS, LRCSG, PSHOG and MSHOG on DS1 with SVM classifier.

	2DC	SF	SS	LRCSG	PSHOG	MSHOG
BC	94.8	98.2	94.4	97.4	96.3	96.1
CS	92.1	80.5	91.8	92.3	91.1	93.5
OT	81.7	75.9	81.3	76.2	82.8	92.3
Total	91.6	90.2	91.2	92.1	92.5	94.8

Table 6. Comparison on classification accuracy (%) of 2DC, SF, SS, LRCSG, PSHOG and MSHOG on DS2 with SVM classifier.

	2DC	SF	SS	LRCSG	PSHOG	MSHOG
BC	96	96.7	90	100	96.3	98.1
CS	93.4	79.3	89.4	96.2	91.7	92.2
OT	72.8	77.6	81.2	73.8	80.4	84.6
Total	87.4	84.5	86.9	90.0	89.5	91.6

5.4. Effectiveness of TDDL-SIC

In this section, we first focus on the effectiveness of structured incoherent constraints by visualizing the distribution of the sparse codes and comparing the performance of TDDL with different constraints. Experiments are performed on DS1 with the parameters in Tables 3 and 4. Then we illustrate the effectiveness of TDDL-SIC based on MSHOG feature by comparing the performance of TDDL-SIC and other classifiers, including SVM, K-NN and SRC. For these reference classifier, SVM, K-NN and SRC are implemented by LIBSVM software [51], Statistics and the Machine Learning Toolbox of MATLAB and the SPAMS software [32], respectively. We conduct experiments of each classifier twenty times on both datasets to reduce the disturbance of stochastic factors in single experiment and present the average performance.

In Figure 10, we visualize the distribution of the sparse codes with respect to the dictionary, which is learnt by TDDL (without incoherent constraints). The sparse codes of the training set are presented in Figure 10a, while the sparse codes of the testing set are presented in Figure 10b. For each figure, the X- and Y-axes stand for the sample number and atom number, respectively. We can see that the sparse codes of both training and testing set in Figure 10 do not maintain the expected structure. Figure 11 presents the distribution of the sparse codes with respect to the dictionary, which is learned by TDDL with intrinsic constraints only. It can be noticed that the sparse codes of both sets show better structure property compared with Figure 10. However, the sparse codes of the testing set still do not maintain the expected structure. Figure 12 presents the distribution of the sparse codes with respect to the dictionary, which is learnt by TDDL-SIC. Apparently, the sparse codes of both sets show a good aggregation property. For the sparse codes of lth class's samples, almost all nonzero elements are associated to lth sub-dictionary.

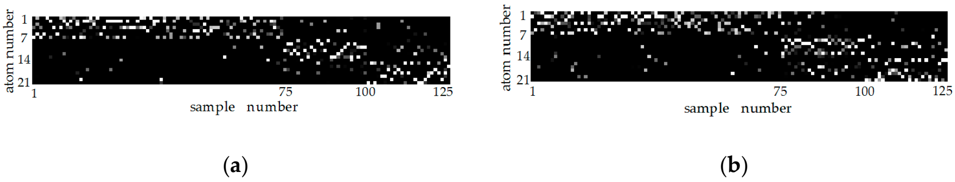


Figure 10. The distribution of sparse codes with respect to the dictionary, which is learnt by TDDL: (**a**) the distribution of sparse codes of the training set; and (**b**) the distribution of sparse codes of the testing set.

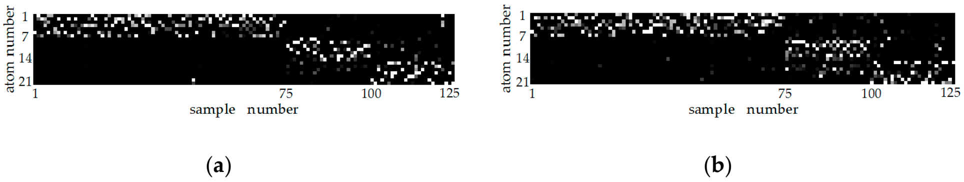


Figure 11. The distribution of sparse codes with respect to the dictionary, which is learnt by TDDL with intrinsic constraints only: (**a**) the distribution of sparse codes of the training set; and (**b**) the distribution of sparse codes of the testing set.

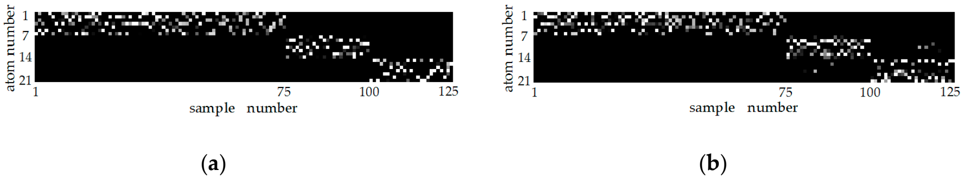


Figure 12. The distribution of sparse codes with respect to the dictionary, which is learnt by TDDL-SIC: (**a**) the distribution of sparse codes of the training set; and (**b**) the distribution of sparse codes of the testing set.

The comparison of classification accuracy of TDDL, TDDL with intrinsic constraints only and TDDL-SIC are presented in Table 7. For container ships and oil tankers, TDDL with intrinsic constraints only achieves higher classification accuracy than TDDL, and TDDL-SIC achieves higher classification accuracy than TDDL with intrinsic constraints only. For bulk carriers, the classification accuracy of all methods is 100%. The overall classification accuracy of TDDL-SIC is 98.4%, which is higher by 4.7% and 2.4% than that of TDDL and TDDL with intrinsic constraints only, respectively. Similar results can also be obtained on DS2. We will not repeat them here. Therefore, we can conclude that structured incoherent constraints improve classification performance.

Table 7. Comparison of the classification accuracy (%) of TDDL, TDDL with intrinsic constraints only and TDDL-SIC on DS1.

	TDDL			**TDDL with Intrinsic Constraints Only**			**TDDL-SIC**		
	BC	CS	OT	BC	CS	OT	BC	CS	OT
BC	100	0	0	100	0	0	100	0	0
CS	4.2	87.6	8.2	0	88.3	11.7	0	96.5	3.5
OT	3	15.9	81.1	0.4	7.8	91.8	0.1	4.2	95.7
Total		93.7			96			98.4	

The comparison of classification accuracy of SVM, K-NN, SRC and TDDL-SIC are presented in Tables 8 and 9. It can be noticed that TDDL-SIC achieve higher classification accuracy than other classifiers for each class of ships on two datasets. The overall classification accuracy using TDDL-SIC is higher than those using other classifiers by 3.5% at least on DS1. For DS2, the improvement of overall classification accuracy is 4.1% at least. Therefore, we can conclude that TDDL-SIC is superior to other classifiers on DS1 and DS2.

Table 8. Comparison on classification accuracy (%) of SVM, K-NN, SRC and TDDL-SIC with MSHOG feature on DS1.

	SVM	**K-NN**	**SRC**	**TDDL-SIC**
BC	96.1	94.8	98.2	100
CS	93.5	91.4	93.1	96.5
OT	92.3	86.6	87.0	95.7
Total	94.8	92.5	94.9	98.4

Table 9. Comparison on classification accuracy (%) of SVM, K-NN, SRC and TDDL-SIC with MSHOG feature on DS2.

	SVM	**K-NN**	**SRC**	**TDDL-SIC**
BC	98.1	97.4	100	100
CS	92.2	90.5	93.1	96.5
OT	84.6	88.9	87.3	96.1
Total	91.6	92.3	93.4	97.5

5.5. Classification Performance on Dataset 1

We first perform the ship classification on DS1. The classification methods that are tested and compared are feature selection (FS) [10], superstructure scattering analysis (SCA) [5], feature space based sparse representation (FSSR) [31], joint feature and classifier selection (JFCS) [9], and MSHOG feature and TDDL-SIC ("MSHOG+TDDL-SIC"). We perform experiments twenty times for each method in accordance with convention.

Table 10 demonstrates the classification accuracy obtained by all the methods on DS1, including the classification accuracy of each class and overall classification accuracy. The proposed method, "MSHOG+TDDL-SIC", obtains the best classification accuracy, including the classification accuracy of bulk carriers, container ships, oil tankers, and the overall classification accuracy. The overall classification accuracy of the proposed method is as high as 98.4%, which outperforms the second highest by 4.2%. Moreover, we can find that the classification accuracy of bulk carriers, container ships and oil tankers in the proposed method is higher than that of bulk carriers, container ships and oil tankers in other competitors by at least 1.8%, 3.8% and 10.4%, respectively. As expected, the proposed method presents significant improvements in the classification of container ships and oil tankers. Therefore, we can conclude that the proposed method classifies container ships and oil tankers more precisely and yields best performance.

Table 10. Comparison of the classification accuracy (%) of FS, SCA, FSSR, JFCS and "MSHOG+TDDL-SIC" on DS1.

	FS	**SCA**	**FSSR**	**JFCS**	**MSHOG+TDDL-SIC**
BC	98.2	94.4	97.8	98.1	100
CS	80.5	91.8	92.7	89.6	96.5
OT	75.9	81.3	85.1	85.2	95.7
Total	90.2	91.2	94.2	93.9	98.4

5.6. Classification Performance on Dataset 2

We perform ship classification on DS2 to analyze the performance of classification methods further. Table 11 illustrates the classification accuracy obtained by all the methods on DS2. We can see that the proposed method achieves the highest overall classification accuracy of 97.5%, which is better than the competitors by 3.4% at least. Additionally, the proposed method also outperforms other methods in classification accuracy of each class. The proposed method reaches 100% in classification accuracy of bulk carriers, and have at least 2.2% superiority over other methods in classification accuracy of container ships. The improvement of the proposed method in classification accuracy of oil tankers is as high as 9.3% and 7.1%, compared to FSSR and JFCS, respectively. None of others reaches classification accuracy over 90% for oil tankers. These results support our assertion that MSHOG feature and TDDL-SIC provide strong discriminability and greatly improve the classification performance.

Table 11. Comparison of the classification accuracy (%) of FS, SCA, FSSR, JFCS and "MSHOG+TDDL-SIC" on DS2.

	FS	**SCA**	**FSSR**	**JFCS**	**MSHOG+TDDL-SIC**
BC	96.7	90	100	100	100
CS	79.3	89.4	95	93.3	96.5
OT	77.6	81.2	86.8	89	96.1
Total	84.5	86.9	93.9	94.1	97.5

6. Conclusions

In this paper, we proposed a novel method for ship classification in SAR image—MSHOG feature and TDDL with structured incoherent constraints ("MSHOG+TDDL-SIC"). We adapt SAR-HOG to our ship classification task, use manifold learning for dimensionality reduction, and obtain MSOHG feature. Then, we jointly optimize the dictionary and the classifier parameter in TDDL framework. Intrinsic and direct constraints are employed to learn a dictionary with elegant structures. The corresponding optimization algorithm is developed using fixed point differentiation and gradient descent. Finally, we conduct various experiments to verify the effectiveness of MSHOG feature, explore the coherence of the dictionary in relation to the constraints, and evaluate the performance of our method. The experiment results verify the superior performance of our method on two datasets quantitatively. Our method outperforms other methods and achieves the best classification accuracy, as high as 98.4% and 97.5% on DS1 and DS2, respectively.

In the future, we would like to apply the proposed method to general classification task in SAR image. In addition, our method can also be extended to PolSAR image classification.

Acknowledgments: The work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 61771043, in part by the Key Project of NSFC under Grant No. 61132008 and in part by the Major Research Plan of the NSFC under Grant No. 61490693. The authors would like to thank K. Ji and X. Xing, the authors of [6,10,31], for providing the TerraSAR-X dataset (DS1). The authors would like to thank H. Lang, the author of [8,9], for precious suggestions about data preprocessing.

Author Contributions: Huiping Lin proposed the general idea of MSHOG and TDDL with structured incoherent constraints in SAR image, and designed and performed the experiments. Shengli Song and Jian Yang gave many suggestions and deep-going analysis. The manuscript was written by Huiping Lin.

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

References

1. Brusch, S.; Lehner, S.; Fritz, T.; Soccorsi, M.; Soloviev, A.; van Schie, B. Ship Surveillance with TerraSAR-X. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 1092–1103. [CrossRef]
2. Osman, H.M.; Pan, L.; Blostein, S.D.; Gagnon, L. Classification of ships in airborne SAR imagery using backpropagation neural networks. In Proceedings of the Radar Processing, Technology, and Applications II, San Diego, CA, USA, 24 September 1997; pp. 126–136.
3. Margarit, G.; Tabasco, A. Ship Classification in Single-Pol SAR Images Based on Fuzzy Logic. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3129–3138. [CrossRef]
4. Zhang, H.; Tian, X.; Wang, C.; Wu, F.; Zhang, B. Merchant Vessel Classification Based on Scattering Component Analysis for COSMO-SkyMed SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1275–1279. [CrossRef]
5. Jiang, M.; Yang, X.; Dong, Z.; Fang, S.; Meng, J. Ship Classification Based on Superstructure Scattering Features in SAR Images. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 616–620. [CrossRef]
6. Xing, X.W.; Ji, K.F.; Chen, W.T.; Zou, H.X.; Sun, J.X. Superstructure scattering distribution based ship recognition in TerraSAR-X imagery. In Proceedings of the IOP Conference Series: Earth and Environmental Science, Kuala Lumpur, Malaysia, 22–23 April 2014; pp. 682–691.
7. Leng, X.; Ji, K.; Zhou, S.; Xing, X.; Zou, H. 2D comb feature for analysis of ship classification in high-resolution SAR imagery. *Electron. Lett.* **2017**, *53*, 500–502. [CrossRef]
8. Lang, H.; Meng, J. Hierarchical ship detection and recognition with high-resolution polarimetric synthetic aperture radar imagery. *J. Appl. Remote Sens.* **2014**, *8*. [CrossRef]
9. Lang, H.; Zhang, J.; Zhang, X.; Meng, J. Ship Classification in SAR Image by Joint Feature and Classifier Selection. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 212–216. [CrossRef]
10. Chen, W.T.; Ji, K.F.; Xing, X.W.; Zou, H.X.; Sun, H. Ship recognition in high resolution SAR imagery based on feature selection. In Proceedings of the International Conference on Computer Vision in Remote Sensing, Xiamen, China, 16–18 December 2012; pp. 301–305.
11. Fernandez Arguedas, V.; Velotto, D.; Tings, B.; Greidanus, H.; Bentes da Silva, C.A. Ship classification in high and very high resolution satellite SAR imagery. In Proceedings of the Security Research Conference, 11th Future Security, Berlin, Germany, 13–14 September 2016; pp. 347–354.
12. Bentes, C.; Velotto, D.; Lehner, S. Target classification in oceanographic SAR images with deep neural networks: Architecture and initial results. In Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS), Milan, Italy, 26–31 July 2015; pp. 3703–3706.

13. Bentes, C.; Velotto, D.; Tings, B. Ship Classification in TerraSAR-X Images with Convolutional Neural Networks. *IEEE J. Ocean. Eng.* **2017**, *43*, 258–266. [CrossRef]
14. Santamaria, C.; Stasolla, M.; Argentieri, P.; Alvarez, M.; Greidanus, H. *Sentinel-1 Maritime Surveillance. Testing and Experiences with Long-Term Monitoring*; JRC Science and Policy Reports; Publications Office of the European Union: Luxembourg, 2015; pp. 54–61.
15. Song, S.; Xu, B.; Yang, J. SAR Target Recognition via Supervised Discriminative Dictionary Learning and Sparse Representation of the SAR-HOG Feature. *Remote Sens.* **2016**, *8*, 683. [CrossRef]
16. Weinberger, K.Q.; Saul, L.K. Unsupervised learning of image manifolds by semidefinite programming. *Int. J. Comput. Vis.* **2006**, *70*, 77–90. [CrossRef]
17. Tenenbaum, J.B.; De Silva, V.; Langford, J.C. A global geometric framework for nonlinear dimensionality reduction. *Science* **2000**, *290*, 2319–2323. [CrossRef] [PubMed]
18. Belkin, M.; Niyogi, P. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.* **2003**, *15*, 1373–1396. [CrossRef]
19. Roweis, S.T.; Saul, L.K. Nonlinear dimensionality reduction by locally linear embedding. *Science* **2000**, *290*, 2323–2326. [CrossRef] [PubMed]
20. Donoho, D.L.; Grimes, C. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proc. Natl. Acad. Sci. USA* **2003**, *100*, 5591–5596. [CrossRef] [PubMed]
21. Brand, M. Charting a manifold. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2003; pp. 985–992.
22. Zhang, Z.; Zha, H. Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Sci. Comput.* **2004**, *26*, 313–338. [CrossRef]
23. Wold, S.; Esbensen, K.; Geladi, P. Principal component analysis. *Chemom. Intell. Lab. Syst.* **1987**, *2*, 37–52. [CrossRef]
24. Baraniuk, R.G.; Cevher, V.; Duarte, M.F.; Hegde, C. Model-based compressive sensing. *IEEE Trans. Inf. Theory* **2010**, *56*, 1982–2001. [CrossRef]
25. Chen, S.S.; Donoho, D.L.; Saunders, M.A. Atomic decomposition by basis pursuit. *SIAM Rev.* **2001**, *43*, 129–159. [CrossRef]
26. Wright, J.; Yang, A.Y.; Ganesh, A.; Hegde, C. Robust face recognition via sparse representation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *31*, 210–227. [CrossRef] [PubMed]
27. Huang, J.B.; Yang, M.H. Fast sparse representation with prototypes. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3618–3625.
28. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3973–3985. [CrossRef]
29. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification via kernel sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 217–231. [CrossRef]
30. Zhang, H.; Nasrabadi, N.M.; Zhang, Y.; Huang, T.S. Multi-view automatic target recognition using joint sparse representation. *IEEE Trans. Aerosp. Electron. Syst.* **2012**, *48*, 2481–2497. [CrossRef]
31. Xing, X.; Ji, K.; Zou, H.; Chen, W.; Sun, J. Ship Classification in TerraSAR-X Images with Feature Space Based Sparse Representation. *IEEE Geosci. Remote Sens. Lett.* **2013**, *10*, 1562–1566. [CrossRef]
32. Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G. Online dictionary learning for sparse coding. In Proceedings of the 26th Annual International Conference on Machine learning, Montreal, QC, Canada, 14–18 June 2009; pp. 689–696.
33. Aharon, M.; Elad, M.; Bruckstein, A. rmK-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **2006**, *54*, 4311–4322. [CrossRef]
34. Ramirez, I.; Sprechmann, P.; Sapiro, G. Classification and clustering via dictionary learning with structured incoherence and shared features. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3501–3508.
35. Mairal, J.; Bach, F.; Ponce, J.; Sapiro, G.; Zisserman, A. Discriminative learned dictionaries for local image analysis. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008; pp. 1–8.
36. Mairal, J.; Ponce, J.; Sapiro, G.; Zisserman, A.; Bach, F.R. Supervised dictionary learning. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2009; pp. 1033–1040.

37. Duarte-Carvajalino, J.M.; Sapiro, G. Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. *IEEE Trans. Image Process.* **2009**, *18*, 1395–1408. [CrossRef] [PubMed]
38. Mairal, J.; Bach, F.; Ponce, J. Task-driven dictionary learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 791–804. [CrossRef] [PubMed]
39. Elad, M. *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing*; Springer: New York, NY, USA, 2010; Volume 2, pp. 1094–1097.
40. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **2005**, *67*, 301–320. [CrossRef]
41. Alpaydin, E. *Introduction to Machine Learning*; Pitman: London, UK, 1988.
42. Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*; China Machine Press: Beijing, China, 2005.
43. Bradley, D.M.; Bagnell, J.A. Differentiable sparse coding. In Proceedings of the International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–11 December 2008; Curran Associates Inc.: Red Hook, NY, USA, 2008; pp. 113–120.
44. Yang, J.; Yu, K.; Huang, T. Supervised translation-invariant sparse coding. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 3517–3524.
45. 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, 20–25 June 2005; Volume 1, pp. 886–893.
46. Borchers, B. CSDP, A C library for semidefinite programming. *Optim. Methods Softw.* **1999**, *11*, 613–623. [CrossRef]
47. Ramírez, I.; Lecumberry, F.; Sapiro, G. Universal priors for sparse modeling. In Proceedings of the 2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), Aruba, The Netherlands, 13–16 December 2009; pp. 197–200.
48. 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**, *23*, 623–634. [PubMed]
49. 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**, *53*, 4457–4471. [CrossRef]
50. Wang, Z.; Nasrabadi, N.M.; Huang, T.S. Semisupervised hyperspectral classification using task-driven dictionary learning with Laplacian regularization. *IEEE Trans. Geosci. Remote Sens.* **2015**, *53*, 1161–1173. [CrossRef]
51. Chang, C.C.; Lin, C.J. LIBSVM: A library for support vector machines. In *ACM Transactions on Intelligent Systems and Technology (TIST)*; ACM: New York, NY, USA, 2011; Volume 2, p. 27.