Deep nonsmooth nonnegative matrix factorization network with semi-supervised learning for SAR image change detection

Heng-Chao Li, Gang Yang*, Wen Yang, Qian Du, William J. Emery

1. Introduction

As one of key technologies of Earth observation applications, change detection of synthetic aperture radar (SAR) images aims at detecting the changed areas from multitemporal SAR images, acquired over the same geographical area but at different times (Bovolo and Bruzzone, 2005; Gong et al., 2014; Luo et al., 2018). Considering that SAR can provide high-resolution images of the Earth surface at anytime and under all-weather conditions, a tremendous number of multitemporal SAR images are acquired by active sensors on board of airborne and spaceborne platforms, which have been demonstrated to be the effective sources for change detection. By now, to automatically detect the changed areas, many change detection methods have been reported, which mainly include Markov field model (Wang et al., 2013; Benedeka et al., 2015), finite mixture models (Bruzzone and Prieto, 2000; Celik, 2011), thresholding (Patra et al., 2011; Xiong et al., 2012), and clustering (Gong et al., 2012; Li et al., 2017), etc.

In this paper, we focus on researching the clustering-based change detection in view of its simplicity and effectiveness. In Celik (2009), the effective features were firstly extracted with principle component analysis (PCA) from a difference image, and then a binary change mask (CM) was obtained by partitioning the above extracted features with the k-means clustering, referred to as PCAKM. Furthermore, Volpi et al. (2012) and Jia et al. (2016) utilized kernel k-means and multiple kernel k-means to detect the changed areas from the original images, respectively. In order to extract more useful features, Wu et al. (2014) and Wu et al. (2017) utilized slow feature analysis (SFA) and kernel SFA, while Gao et al. (2016) and Li et al. (2019) employed PCANet originally proposed by Chan et al. (2015) to perform change detection of SAR images, respectively. Overall, the PCA algorithm learns the holistic representations, which generally lacks intuitive physical meaning. By contrast, as one of low-rank matrix approximations, nonnegative matrix factorization (NMF) imposes the nonnegativity constraint on the decomposed matrices (i.e., basis matrix and coefficient matrix), thus providing a useful tool for part-based representation learning that is valuable for clustering or classification (Lee and Seung, 1999; Lee et al., 2010). However, for the NMF algorithm, there is a high degree of overlapping among basis vectors that contradict the intuitive nature of the “parts” (Mel, 1999). Therefore, a nonsmooth NMF (nsNMF) model was proposed in Pascual-Montano et al. (2006) to learn the localized...
and less overlapped representations. From the geometric perspective, since the real data is sampled generally from a low-dimensional manifold embedded in a high-dimensional ambient space, Cai et al. (2011) proposed graph regularized NMF (GNMF) to explicitly exploit the local invariance. In Ding et al. (2010), NMF was extended into semi-nonnegative matrix factorization (semi-NMF), which allowed the observation matrix and basis matrix to have mixed signs, and only kept the coefficient matrix as nonnegative. As such, in order to perform detection task, semi-NMF was introduced into Li et al. (2014a) to cluster the PCA-extracted features, called as PCASNMF. To further consider spatial contextual information and enhance the robustness against the noise and outliers, a robust semi-NMF with total variation was built for change detection of SAR images, referred to as RSNMF-TV (Li et al., 2018).

NMF and its variations are regarded as a single-layer learning structure, thus only learning the basic and low-level features, while they cannot exploit more useful hierarchical information from rather complex data (Qian et al., 2016). Recently, deep learning (DL) proposed by Hinton et al. (2006) has been widely applied to remote sensing image processing (Gong et al., 2016; Liu et al., 2018; Zhang et al., 2016; Wu et al., 2018; Paoletti et al., 2018; Feng et al., 2018). Hence, by adopting the DL’s idea, a multilayer NMF (MLNMF) model was proposed in Rajabi and Ghassemian (2015), which factorized the non-negative observation matrix into the product of multiple basis matrices and one coefficient matrix. To further learn discriminative features, Tong et al. (2018) constructed a deep discriminative and robust NMF network with soft label constraint (Deep DRNMFN_SLC). However, these models only factorize the resulting coefficient matrices layer by layer, referred to as pretrained stage in DL. As a result, to reflect the total reconstruction error and better learn the decomposed matrices, it is necessary to fine-tune the designed model via back-propagation (BP) algorithm. Specifically, Trigeorgis et al. (2017) extended semi-NMF into deep semi-NMF structure, whose learning process included pretraining stage and fine-tuning stage, thus obtaining satisfactory performance on face recognition. Subsequently, the deep semi-NMF model is further explored on the PCA-extracted features (called as PCADSNMF) to achieve change detection of remote sensing images (Yang et al., 2018). Similarly, by enforcing L12 constraint and total variation regularizer, we proposed a sparsity-constrained deep NMF with total variation (SDNMF-TV) model for hyperspectral unmixing (Feng et al., 2018). Yu et al. (2018) extended nsNMF into deep nsNMF model to learn hierarchical and localized data representations. However, for MLNMF, deep NMF, and deep nsNMF, they only linearly decompose the resulting nonnegative matrices layer by layer, which may fail to characterize nonlinearities in complex data. Inspired by neurophysiology paradigms and DL (Riesenhuber and Poggio, 1999; Hinton et al., 2006; Trigeorgis et al., 2017), the nonlinear activation function is introduced into the deep nsNMF model to learn nonlinear data representation. As such, a nonlinear deep nsNMF model is first built to learn hierarchical, nonlinear, and localized features from rather complex data.

In most of the existing change detection methods, e.g., PCAKM (Celik, 2009), PCASNMF (Li et al., 2014a), RSNMF-TV (Li et al., 2018), and PCADSNMF (Yang et al., 2018), feature learning and clustering are two separate steps. In other words, a clustering algorithm is further utilized on the learned features to partition the pixels into the changed and unchanged classes. Besides, Zafeiriou et al. (2006) demonstrated that supervised NMF may perform better in classification task. Subsequently, in order to make the learned features suitable for classification, Lee et al. (2010) and Jiang et al. (2013) both adopt a simple linear classifier, which is difficult to effectively partition a real observation data. By contrast, extreme learning machine (ELM) proposed by Huang et al. (2006) is able to produce good generalization performance with lower computational complexity. Therefore, different from Yu et al. (2018), by employing the nonlinear deep nsNMF model and the ELM classifier, we construct a deep nsNMF network to learn nonlinear, localized, and discriminative features from complex data. Besides, labeled samples are required to train the proposed network, but whose acquisition usually demands time and cost. In such situation, semi-supervised learning can be of great practical value. As such, a semi-supervised learning strategy is adopted for deep nsNMF network to incorporate partially the information on class labels. The main contributions of this paper are threefold.

1. To characterize the nonlinearities of the observation data, a nonlinear deep nsNMF model is first constructed by introducing nonlinear activation function. Meanwhile, in order to make the learned features suitable for classification, the ELM classifier is stacked into the nonlinear deep nsNMF model to construct deep nsNMF network, which can learn hierarchical, nonlinear, localized, and discriminative data representations.

2. In practical applications of remote sensing, the acquisition of labeled data is generally difficult. In order to combine partial labeled samples and unlabeled samples, the semi-supervised learning strategy is proposed to train the proposed deep nsNMF network.

3. The learning process of the proposed network composes of pre-training stage and fine-tuning stage, in which the former pretrains all decomposed matrices layer by layer and the latter aims at reducing the total reconstruction error by using the mini-batch gradient descent algorithm. In addition, the optimization of these two stages is also deduced in detail.

The rest of this paper is organized as follows. Section 2 gives some related work. Section 3 describes the proposed network in detail. Experimental results and discussion on four pairs of SAR images are presented in Section 4. Section 5 concludes the paper with some remarks.

2. Related work

2.1. Nonnegative matrix factorization

NMF is a popular method for low-rank matrix approximations due to its clear physical meaning and simplicity, which factorizes the non-negative observation matrix $X \in \mathbb{R}^{D \times N}$ into the product of two non-negative matrices $Z \in \mathbb{R}^{D \times K}$ and $F \in \mathbb{R}^{K \times N}$, representing basis matrix and coefficient matrix (Lee and Seung, 1999), respectively, i.e.,

$$X \approx ZF.$$  

Given $X$, in order to estimate $Z$ and $F$, cost function based on Euclidean distance is defined as

$$C = \frac{1}{2} \| X - ZF \|_F^2, \quad \text{s.t.} \quad Z \succeq 0, F \succeq 0,$$  

where $\| \cdot \|_F$ is Frobenius norm of a matrix. To minimize (2), multiplicative update rules (MURs) are employed to alternatively update $Z$ and $F$ (Lee and Seung, 2001) as follows

$$Z \leftarrow Z \odot (XF^T) \rho (ZF F^T),$$  

$$F \leftarrow F \odot (Z^TX) \rho (Z^TZF),$$  

in which $(\cdot)^T$ represents the transpose operation to a matrix, $\odot$ and $\rho$ denote the element-wise multiplication and division, respectively.

2.2. Nonsmooth nonnegative matrix factorization

The nsNMF model proposed by Pascual-Montano et al. (2006) can learn localized and less overlapped data representation, whose cost
function is given by
\[ C = \frac{1}{2} \| X - \text{ZSF} \|_F^2, \quad \text{s. t. } Z \geq 0, F \geq 0, \] (4)
where \( S \in \mathbb{R}^{K \times K} \) is smoothing matrix, i.e.,
\[ S = \left( 1 - \frac{d}{K} \right) I + \frac{d}{K} I^T, \] (5)
where \( I \) is an identity matrix, \( I \) is a vector of ones, and \( d \) controls the extent of smoothness of the matrix \( S \), being in the range of \([0, 1]\). If \( d = 0 \), nSNMF boils down to the basic NMF. Similar to the optimization of (2), \( Z \) and \( F \) are updated iteratively according to
\[ Z \leftarrow Z \odot \left[ X (\text{ZSF})^T \right] \odot [\text{ZSF}(\text{ZSF})^T]. \] (6a)
\[ F \leftarrow F \odot [\text{ZSF}^T X] \odot [\text{ZSF}^T \text{ZSF}], \] (6b)
until a stopping condition is reached.

### 2.3. Deep NMF and deep nSNMF

Both NMF and nSNMF are regarded as a single-layer learning structure, which only learn the basic and low-level features from original data. In order to explore the hierarchical and high-level features embedded in complex observation, a deep NMF model is proposed for hyperspectral unmixing, which factorizes the nonnegative observation matrix \( X \) into multiple nonnegative basis matrices and one coefficient matrix (Feng et al., 2018), i.e.,
\[ X \approx Z_1 Z_2 \cdots Z_L F_L, \] (7)
where \( L \) is the number of layers, \( K_i, F_i \in \mathbb{R}^{F_{i-1} \times K_i} \), and \( F_L \in \mathbb{R}^{F_{i-N}} \) are the number of dimensions, basis matrix, and coefficient matrix of the \( l \)-th layer, respectively. Note that \( K_{i-1} \) equals \( D \) if and only if \( l = 1 \). Similar to the deep NMF model, Yu et al. (2018) linearly extended nSNMF into deep structure, and proposed the deep nSNMF model for learning the hierarchical and localized data representations, given as
\[ X \approx S_1 S_2 \cdots S_L F_L, \] (8)
where \( S_i \in \mathbb{R}^{F_i \times K_i} \) is the smoothing matrix of the \( l \)-th layer.

### 3. Methodology

Consider two coregistered multitemporal SAR images
\[ Y_1 = [y_i(j)] 1 \leq i \leq L, 1 \leq j \leq J \] and \( Y_2 = [y_j(i)] 1 \leq i \leq L, 1 \leq j \leq J \), acquired over the same geographical area but at different times (i.e., \( t_i \) and \( t_j \)). Here \( I \) and \( J \) are the numbers of row and column of these input images, respectively. In this paper, a SAR image change detection method based on deep nSNMF network with semi-supervised learning is proposed, whose schematic diagram is shown in Fig. 1.

#### 3.1. Generation of neighborhood features

The neighborhood features \( X \) can be generated by using these two considered images \( Y_1 \) and \( Y_2 \). First of all, in order to guarantee that the value for each element of \( Y_i \) is in the range of \([0, 1]\), \( Y_i \) is normalized according to
\[ Y_i = \frac{Y_i - \min(Y_i)}{\max(Y_i) - \min(Y_i)}, t = \{1, 2\}. \] (9)
where \( \min(Y_i) \) and \( \max(Y_i) \) return the minimum and maximum of \( Y_i \), respectively. Then, a \( \rho \times \rho \) local sliding window is enforced on \( Y_t \) as a neighborhood centered at pixel \((i, j)\) with \( \rho \geq 3 \). Furthermore, the neighborhood feature \( y_t(i, j) \) for an arbitrary pixel is constructed by using pixel values of the position \((i, j)\) and its neighbourhood (Celik, 2011), i.e.,
\[ y_t(i, j) = [y_t(i - \lfloor \rho/2 \rfloor, j - \lfloor \rho/2 \rfloor), \ldots, y_t(i, j), \ldots, y_t(i + \lfloor \rho/2 \rfloor, j + \lfloor \rho/2 \rfloor)]^T, \] (10)
where \( \lfloor \rho/2 \rfloor \) rounds the nearest integer towards minus infinity, e.g., \( \lfloor 1.5 \rfloor = 1 \), and \( y_t(i, j) \in \mathbb{R}^{D \times 1} \). \( y_t \) and \( y_2 \) are concatenated into a feature vector \( x \), i.e., \( x = [y_1^T, y_2^T] \), which is treated as a column of \( X \in \mathbb{R}^{D \times N} \) with \( D = 2\rho^2 \) and \( N = L \times J \). Moreover, in order to adequately exploit the change information from these two SAR images, a weighted strategy is developed to enhance the difference between neighborhood features corresponding to the changed and unchanged pixels. Specifically, a diagonal weight matrix \( \Lambda \) with \( N \times N \) is designed by using the corresponding pixel intensity differences for these two input images, whose each diagonal element is defined as
\[ \lambda_{nn} = 1 - \left( 1 - \frac{(Y_1 \odot \max(Y_2))^2}{\max(Y_2)} \right)^2, \] (11)
where \( Y_2 \) is generated by the logarithm-ratio operator (Bazi et al., 2009), i.e., \( Y_2 = [\log(Y_1) - \log(Y_2)] \), \( \log(\cdot) \) and \( \max(\cdot) \) are the absolute-value and logarithm operators, respectively. As such, \( X \) can be changed according to \( X \leftarrow X \Lambda \).

![Fig. 1. Schematic diagram of the proposed deep nSNMF network with semi-supervised learning for SAR image change detection.](image-url)
3.2. Proposed deep nsNMF network with semi-supervised learning

The deep NMF and deep nsNMF models can learn hierarchical features, but they may fail to characterize nonlinearities in real data. Inspired by neurophysiology paradigms and DL (Riesenhuber and Poggio, 1999; Hinton et al., 2006; Trigeorgis et al., 2017), we introduce a nonlinear activation function into the deep nsNMF model, given as

\[ X \approx Z_1 S_1 \sigma(Z_2 S_2 \sigma(\cdots \sigma(Z_l S_l F_l)))) \]

where \( \sigma(\cdot) \) is the nonlinear activation function, defined as

\[ \sigma^{-1}(\cdot) \approx Z_1 S_1 F_1 \]

with \( \sigma^{-1}(\cdot) \) being the inverse function of \( \sigma(\cdot) \). In order to achieve the favorable solutions, according to Hinton et al. (2006), Trigeorgis et al. (2017), Fang et al. (2018), the learning process of the proposed deep nsNMF network is divided into two stages: pretraining stage and fine-tuning stage.

(1) Pretraining Stage: This stage aims at pretraining all factors layer by layer. Specifically, in the first layer, the observation matrix \( X \) is factorized into \( Z_1 \) and \( F_1 \), and \( S_1 \) controls the sparsity on \( Z_1 \) and \( F_1 \), i.e., \( X \approx Z_1 S_1 F_1 \). Subsequently, \( \sigma^{-1}(\cdot) \) is stacked into the output of the first layer, and the resulting \( \sigma^{-1}(F_1) \) is regarded as the input of the second layer, followed by factorizing \( \sigma^{-1}(F_1) \approx Z_2 S_2 F_2 \). The same process is continued until the maximum number of layers is reached. In addition, similar to (4), the Euclidean-based cost function of the \( l \)-th layer is given as

\[ C = \frac{1}{2} \| \sigma^{-1}(F_{l-1}) - Z_l S_l F_l \|_F^2, \quad \text{s. t.} \quad Z_l \geq 0, \quad F_l \geq 0, \]

where \( \sigma^{-1}(F_1) = X \) if and only if \( l = 1 \).

(2) Fine-Tuning Stage: After pretraining the factors of all layers, this stage is performed by minimizing the following cost function:

\[ J = \frac{1}{2} \| X - Z_1 S_1 \sigma(Z_2 S_2 \sigma(\cdots \sigma(Z_l S_l F_l)))) \|_F^2, \quad \text{s. t.} \quad Z_l \geq 0, \quad F_l \geq 0 \]

to reduce the total reconstruction error of the proposed model, thus learning better factorized matrices. Moreover, Zaefirou et al. (2006) demonstrated that supervised NMF can perform better in classification task. Therefore, as in Lee et al. (2010), Li et al. (2013), in order to make the resulting features suitable for classification, the classification error is contained into the cost function given in (15) as a supplementary term, i.e.,

\[ J = \frac{1}{2} \| X - Z_1 S_1 \sigma(Z_2 S_2 \sigma(\cdots \sigma(Z_l S_l F_l)))) \|_F^2 + \frac{\gamma}{2} \| \mathbf{T} - AF \|_F^2 \]

where \( \gamma \) is a regularization parameter, \( A \) is a linear classifier parameter, and \( \mathbf{T} \in \mathbb{R}^{N \times 1} \) is the label vector with \( N \) being the total number of labeled samples. However, (16) utilizes the simple linear classifier, i.e., \( \sigma(\mathbf{A} \mathbf{f}) = \mathbf{A} \mathbf{f} \), which is difficult to effectively deal with various real data. By contrast, ELM is a single-hidden layer feedforward neural network and is able to produce good generalization performance with lower computational complexity (Huang et al., 2006; Huang et al., 2012). As such, the ELM classifier is adopted to replace the linear classifier of (16), and a novel deep nsNMF network is constructed, whose cost function in fine-tuning stage is defined as

\[ J = \frac{1}{2} \| X - Z_1 S_1 \sigma(Z_2 S_2 \sigma(\cdots \sigma(Z_l S_l F_l)))) \|_F^2 + \frac{\gamma}{2} \| \mathbf{T} - H\mathbf{F} \|_F^2 + \frac{\alpha}{2} \| \mathbf{F} \|_F^2, \quad \text{s. t.} \quad Z_l \geq 0, \quad F_l \geq 0, \]

where \( \gamma \) and \( \alpha \) are also a regularized parameter, and \( \beta \in \mathbb{R}^{N \times 1} \) is the output weight vector connecting the hidden layer and the output layer. In addition, \( H \in \mathbb{R}^{N \times M} \) is the output matrix of the hidden layer in terms of the 1th features \( F_1 \) with \( M \) being the number of hidden nodes, i.e., \( H = (h_{nm}) \), and \( h_{nm} \) is defined as

\[ h_{nm} = g[(f_{l,n})], w_{mn} = g[(F_l^T W_{nm})], \quad n = 1, 2, \ldots, N, \quad m = 1, 2, \ldots, M, \]

where \( g(\cdot) \) denotes Sigmoid function, \( f_{l,n} \) is the \( n \)-th column vector of \( F_l \), and \( W_{mn} \) is the column vector of weight matrix \( W \) of size \( K_l \times M \) between the hidden and input layers.

For (17), the labeled samples are used to train the deep nsNMF network, but whose acquisition often requires much time and cost in change detection application. Considering that semi-supervised learning makes use of partial labeled data and some unlabeled data for training, we propose a deep nsNMF network with semi-supervised learning for SAR image change detection. Furthermore, the cost function given in (17) can be transformed into

\[ J = \frac{1}{2} \| X - Z_1 S_1 \sigma(Z_2 S_2 \sigma(\cdots \sigma(Z_l S_l F_l)))) \|_F^2 + \frac{\gamma}{2} \| \mathbf{T} - H\mathbf{F} \|_F^2 + \frac{\alpha}{2} \| \mathbf{F} \|_F^2, \quad \text{s. t.} \quad Z_l \geq 0, \quad F_l \geq 0, \]

with \( \mathbf{T} \) and \( \mathbf{0} \) being row vectors of ones and of zeros, respectively. Here, in order to simplify the following derivation, (19) is rewritten as

\[ J = \frac{1}{2} \| X - Z_1 S_1 \sigma(Z_2 S_2 \sigma(\cdots \sigma(Z_l S_l F_l)))) \|_F^2 + \frac{1}{2} \| \mathbf{T} - H\mathbf{F} \|_F^2 + \frac{\gamma}{2} \| \mathbf{F} \|_F^2, \quad \text{s. t.} \quad Z_l \geq 0, \quad F_l \geq 0, \]

where \( \Gamma \) is a diagonal matrix with \( (n_l + n_u) \times (n_l + n_u) \), and its first row diagonal elements equal \( \gamma \) and the rest is set as zero.

3.3. Optimization

In this subsection, the cost functions given in (14) and (21) require to be optimized, whose corresponding optimization processes are as follows.

(1) Optimization for Pretraining Stage: Similar to (6), by minimizing (14), \( Z_l \) and \( F_l \) are updated iteratively according to

\[ Z_l \leftarrow Z_l \ominus \left[ \sigma^{-1}(F_{l-1})(S_l F_l)^T \right] \forall (Z_l S_l F_l(S_l F_l)^T), \]

(22a)

\[ F_l \leftarrow F_l \ominus \left[ (Z_l S_l)^T \sigma^{-1}(F_{l-1}) \right] \forall (Z_l S_l)^T Z_l S_l F_l, \]

(22b)

until stopping condition is reached.

(2) Optimization for Fine-Tuning Stage: This stage aims at minimizing (21) in terms of \( Z_l, H \), and \( \beta \) as follows:

- By making use of the chain rule, the derivative of (21) with respect to \( Z_l \) is computed by

\[ \frac{\delta J}{\delta Z_l} = \frac{\delta J}{\delta F_{l-1}} \odot \nabla_{\sigma(Z_l S_l F_l)} (S_l F_l)^T, \quad l = 2, \ldots, L, \]

(23)
\[
\frac{\delta J}{\delta Z_i} = (Z_i S_i F_i - X) (S_i F_i)^T.
\] (24)

• Similarly, from the second layer to the penultimate layer, the derivative of (21) in terms of \( F_i \) can be calculated as

\[
\frac{\delta J}{\delta F_i} = (Z_i S_i)^T \bigg[ \frac{\delta J}{\delta F_{i-1}} \bigg] \ast V \sigma \left( Z_i S_i F_i \right), \quad i = \{2, \ldots, L - 1\},
\] (25)

and for the first layer,

\[
\frac{\delta J}{\delta F_1} = (Z_i S_i)^T \left( Z_i S_i F_i - X \right).
\] (26)

Since the final-layer output features \( F_L \) are the input of the ELM classifier, the derivation of (21) in terms of \( F_L \) is different from those of the first \( L - 1 \) layers, derived as

\[
\frac{\delta J}{\delta F_L} = (Z_i S_i)^T \bigg[ \frac{\delta J}{\delta F_{L-1}} \bigg] \ast V \sigma \left( Z_i S_i F_L \right) + W \left[ \frac{\delta J}{\delta H} \bigg] \ast V_l (F_L^T W) \right]^T,
\] (27)

where we have

\[
\frac{\delta J}{\delta H} = -\Gamma (\Gamma^T - H \beta) \beta^T.
\] (28)

• The derivative of (21) with respect to \( \beta \) is calculated as

\[
\frac{\delta J}{\delta \beta} = \alpha \beta - H^T \Gamma (\Gamma^T - H \beta).
\] (29)

After achieving the above derivatives in terms of \( Z_i, F_i \), and \( \beta \), the mini-batch gradient descent algorithm (Li et al., 2014b) is employed to deduce the update rules. Specifically, this kind of method utilizes a batch of samples to update the parameters in a loop, which can facilitate effectively the convergence of the proposed network according to

\[
Z_i \leftarrow Z_i - \xi \frac{\delta J}{\delta Z_i};
\] (30a)

\[
F_i \leftarrow F_i - \xi \frac{\delta J}{\delta F_i};
\] (30b)

where both \( \xi \) and \( \xi \) are step size (called the learning rate). It should be noted that the resulting features of the \( l \)th layer \( F_i \) consist of \( F_{i-1} \) and \( F_{i-2}^T \), i.e., \( F_i = [F_{i-1}, F_{i-2}^T] \). Here, \( F_{i-1} \) and \( F_{i-2} \) are the features learned from the labeled data \( X_l \) and unlabeled data \( X_u \), respectively.

In addition, by setting \( \frac{\delta J}{\delta \beta} \) to be zero, the update rule for \( \beta \) is obtained by

\[
\beta = (\alpha I + H^T \Gamma H)^{-1} H^T \Gamma \tilde{Y}.
\] (31)

According to Huang et al. (2012), if the number of the hidden layer is greater than that of the training samples, \( \beta \) is updated by

\[
\beta = H^T (\alpha I + \Gamma HH^T)^{-1} \Gamma \tilde{Y}.
\] (32)

3.4. Generation of change mask

Once \( F_i \) and \( \beta \) are obtained, the intermediate class \( \omega_\circ \) can be further partitioned into the two classes (i.e., \( \omega_\circ \) and \( \omega_\circ \) according to

\[
CM(n_u) = \frac{1}{2} \text{sign}(\langle H^T g \rangle_{n_u}) + \frac{1}{2}, \quad n_u = \{1, 2, \ldots, N_u\},
\] (33)

where \( \text{sign}(\cdot) \) is sign function. For each element \( x \) of the matrix, \( \text{sign}(x) \) returns 1 if the element is greater than zero, 0 if it equals zero, and \(-1\) if it is less than zero. \( H^T \) is the output matrix of the hidden layer with respect to \( F_{i-1}^T \), being calculated by (18). Furthermore, the final binary CM is formed by combining (33) with the changed and unchanged classes generated by the pre-classification algorithm.

Algorithm 1. Proposed deep nsNMF network with semi-supervised learning

<table>
<thead>
<tr>
<th>Input:</th>
<th>SAR images ( Y_1 ) and ( Y_2 ), parameters ( \rho, \mu, L, \theta, \alpha, \beta, \gamma, ) and ( \epsilon ).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initialize:</td>
<td>( W ) by using the random initialization.</td>
</tr>
<tr>
<td>Step 1:</td>
<td>Generate neighborhood features ( X ).</td>
</tr>
<tr>
<td>Step 2:</td>
<td>Obtain the labeled samples ( X_l, T ) and the unlabeled data ( X_u ) based on the pre-classification method.</td>
</tr>
<tr>
<td>Step 3:</td>
<td>Train the proposed network:</td>
</tr>
<tr>
<td>* Pretraining stage:</td>
<td></td>
</tr>
<tr>
<td>for ( l = 1, 2, \ldots, L )</td>
<td></td>
</tr>
<tr>
<td>Initialize ( Z_l ) and ( F_l ) by using the NNDSVD algorithm</td>
<td></td>
</tr>
<tr>
<td>repeat</td>
<td></td>
</tr>
<tr>
<td>Update ( Z_l ) by (22a)</td>
<td></td>
</tr>
<tr>
<td>Update ( F_l ) by (22b)</td>
<td></td>
</tr>
<tr>
<td>until stopping condition is met</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>* Fine-tuning stage:</td>
<td></td>
</tr>
<tr>
<td>repeat</td>
<td></td>
</tr>
<tr>
<td>Update ( Z_l ) by (30a) with (24)</td>
<td></td>
</tr>
<tr>
<td>Update ( F_l ) by (30b) with (26)</td>
<td></td>
</tr>
<tr>
<td>for ( l = 1, 2, \ldots, L )</td>
<td></td>
</tr>
<tr>
<td>Update ( Z_l ) by (30a) with (23)</td>
<td></td>
</tr>
<tr>
<td>Update ( F_l ) by (30b) with (25)</td>
<td></td>
</tr>
<tr>
<td>if ( N &gt; M ), then update ( \beta ) by (31)</td>
<td></td>
</tr>
<tr>
<td>if ( N &lt; M ), then update ( \beta ) by (32)</td>
<td></td>
</tr>
<tr>
<td>until stopping condition is met</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td></td>
</tr>
<tr>
<td>Step 4:</td>
<td>Obtain the binary CM by combining the results obtained by (33) with the pre-classification algorithm.</td>
</tr>
<tr>
<td>Output:</td>
<td>The binary CM.</td>
</tr>
</tbody>
</table>

3.5. Implementation issues

The complete SAR image change detection method is shown in Algorithm 1, in which several implementation issues are discussed as follows.

• The first issue is initialization of \( Z_l \) and \( F_l \) for each layer in pre-training stage. In general, \( Z_l \) and \( F_l \) are initialized randomly, leading to different local optimal solutions in each execution. Moreover, to speed up the convergence of nsNMF, nonnegative double singular value decomposition (NNDSVD) (Boutsidis and Gallopoulos, 2008) is employed to initialize \( Z_l \) and \( F_l \) for each layer.

• The second issue is to generate the labeled samples by using the pre-classification method. Specifically, by considering spatial neighborhood information of each pixel, the neighborhood-based ratio (NR) operator is employed to yield a difference image (Gong et al., 2012). Subsequently, the fuzzy c-means (FCM)-based clustering strategy presented in Li et al. (2015) and Gao et al. (2016) is utilized to assign the difference image pixels into three categories, representing the changed class \( \omega_\circ \), the intermediate class \( \omega_\circ \), and the unchanged class \( \omega_\circ \), respectively. Next, \( \alpha_\circ \) and \( \omega_\circ \) are selected as the cluster labels of the labeled samples, and the labeled samples are denoted by \( X_l, T \), where each pixel vector \( x_i \) in \( X_l \) belongs to \( \omega_\circ \) or \( \omega_\circ \). Meanwhile, the rest is taken as the unlabeled data \( X_u \) in R^R^N^u with \( N = N_l + N_u \).

• The third issue is stopping condition. If the number of iterations \( t \) is up to maximum number \( t_{\text{max}} \) or stopping criteria, i.e.,

\[
|C^{(t+1)} - C^{(t)}| < \epsilon \|C^{(t)}\|,
\] (34a)

\[
|f^{(t+1)} - f^{(t)}| < \epsilon \|f^{(t)}\|,
\] (34b)

are met, the iterative update will be stopped, where (34a) and (34b) are utilized for pre-training stage and fine-tuning stage, respectively.
Fig. 2. Visualized results of the input SAR images and ground-truth images. Foshan data set: (a) $Y_1$ acquired in May 24, 2008, (b) $Y_2$ acquired in December 19, 2008, (c) ground-truth image between (a) and (b). San Francisco data set: (d) $Y_1$ acquired in August 2003, (e) $Y_2$ acquired in May 2004, (f) ground-truth image between (d) and (e). Florence data set: (g) $Y_1$ acquired in July 21, 2004, (h) $Y_2$ acquired in September 30, 2004, (i) ground-truth image between (g) and (h). Yingshang data set: (j) $Y_1$ acquired in June 11, 2016, (k) $Y_2$ acquired in July 5, 2016, (l) ground-truth image between (j) and (k).
4. Experimental results and discussion

In order to verify the effectiveness of the proposed change detection algorithm, the experiments are performed on four pairs of SAR images. Experimental codes are developed in MATLAB R2015b, and are executed on desktop computer with Intel(R) Core(TM) i3-4170 CPU @ 3.70 GHz, 8.00 GB RAM. Detailed descriptions of data sets, quantitative measures, parameter analysis, and experimental results are provided in the following.

4.1. Experimental data sets and quantitative measures

In the experiments, four pairs of data sets are used to assess the effectiveness of the proposed change detection algorithm.

(1) Data Set A: The Foshan data set is made up of a pair of SAR images, acquired over the Foshan City (Guangdong Province, China) on May 24, 2008 and December 19, 2008 by TerraSAR-X to survey the changed urban building areas. The data set with 450 × 300 pixels is shown in Fig. 2(a) and (b). Fig. 2(c) displays the ground-truth image between Fig. 2(a) and (b).

(2) Data Set B: the San Francisco data set is a pair of SAR images with 512 × 512 pixels, acquired on August 2003 and May 2004 over the city of San Francisco (California) by the ERS-2 SAR sensor. Fig. 2(d) and (e) show the San Francisco data set. Meanwhile, Fig. 2(f) presents the ground-truth image between Fig. 2(d) and (e).

(3) Data Set C: The Florence data set with 400 × 400 pixels is a pair of SAR images, shown in Fig. 2(g) and (h), and is extracted from the scene over the city of Florence, Italy (originally image with 7641 × 7441 were acquired by the ERS-2 SAR sensor respectively on July 21, 2004 and September 30, 2004). It is available at http://earth.esa.int/ers/ers/ersaction. Moreover, Fig. 2(i) presents the ground-truth image between Fig. 2(g) and (h).

(4) Data Set D: The Yingshang dataset consists of a couple of SAR images having the size of 800 × 900 pixels, respectively acquired on June 11, 2016 and July 5, 2016 over the Yingshang County by Sentinel-1A satellite to observe the changed water area, shown in Fig. 2(j) and (k). Moreover, their ground-truth image is presented in Fig. 2(l).

In addition, both qualitative and quantitative measures are performed on the real SAR images. Specifically, qualitative analysis is achieved by visual inspection of the resulting CM, while the adopted quantitative measures are as follows.

(1) False alarm (FA): the number of the unchanged pixels that are mistakenly detected as changed pixels. False alarm rate is given by $P_{FA} = FA/N_0 \times 100\%$.

(2) Missed detection (MD): the number of the changed pixels that are incorrectly detected as unchanged pixels. Missed detection rate is described as $P_{MD} = MD/N_0 \times 100\%$.

(3) Total error (TE): the number of the pixels that are incorrectly detected, i.e., the sum of FA and MD. Total error rate is defined as $P_{TE} = (MD + FA)/(N_0 + N_1) \times 100\%$.

(4) Kappa coefficient ($\kappa$): a similarity measure between the ground-truth image and the resulting CM. Kappa coefficient is described as $\kappa = (P_2 - P_1)/(1 - P_2) \times 100\%$, $P_2 = ((N_1 - FA) + (N_0 - MD))/N$, and $P_1 = ((N_0 - MD + FA) \times N_0 + (N_0 - FA + MD) \times N_1)/N^2$, where $N_1$ and $N_0$ are the total numbers of changed pixels and unchanged pixels in the ground-truth image, respectively.

4.2. Parameter analysis

Before proceeding to the experiments, we first analyze the influence of the number of layers $L$, smoothness parameter $\theta$, and regularized parameters $\gamma$ and $\alpha$. In order to avoid the descriptive redundancy, we only take the Foshan data set, shown in Fig. 2(a)-(c), as an example to perform the parameter analysis.

(1) Influence of the Number of Layers $L$: For the proposed network, the number of layers $L$ is an important parameter, which can directly influence the effectiveness of the extracted features. To this end, $L$ is set as $1, 2, 3$, and $4$ to discuss the effect of parameter $L$ on Foshan data set. From Fig. 3 and Table 1, it can be seen that the one-layer structure (i.e., $L = 1$) achieves the worst performance compared with the deep structure, which may fail to learn the useful features. By contrast, the two-layer or three-layer nNMF network performs much better than the single-layer one, indicating that deep-layer network is beneficial to explore more effective features. However, with the further increase of $L$, the detection performance decreases according to Table 1. Overall, when $L$ equals 2, the resulting CM obtained by the proposed network is the most similar to the ground-truth image, which is also evidenced according to the smallest value of $P_{TE}$ and largest value of $\kappa$ in Table 1. This may be the reason that as $L$ continues to increase, the network will contain more tunable parameters, thus leading to higher complexity and even over-fitting.

(2) Influence of Parameters $\theta$, $\gamma$, and $\alpha$: In the proposed network, $\theta$ controls the extent of smoothness of the matrix $S$, and $\gamma$ and $\alpha$ define the relative contributions of the corresponding terms. Hence, these parameters can be divided into two groups. First, we discuss the influence of parameter $\theta$ under $L = 2$, $\gamma = 2$, and $\alpha = 0.5$, and $\theta$ is defined in a finite set $\{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Form Fig. 4, it is apparent that the plot for $P_{TE}$ descends as $\theta$ grows in the beginning, and then progressively rises with the further increase of $\theta$. When $\theta$ is in the vicinity of 0.5, the proposed method performs much better according to the smallest value of $P_{TE}$ and largest value of $\kappa$. More importantly, if $\theta$ equals zero, the proposed network is degenerated into semi-supervised deep NMF network, whose cost function is given as

$$J = \frac{1}{2} \|X - Z_0^k \cdot (Z_0^k) \cdot (Z_0^k) \cdot (\cdots (Z_0^k)(E_k))\|_F \cdot (\gamma \cdot (\gamma + 1) \text{tr}(H_k^2)), \frac{1}{2} \|Z_0^k\|_2, \text{ s. t. } Z_0^k \geq 0, F_k \geq 0.$$

(35)

Furthermore, it can be seen from Fig. 4 that (35) achieves the larger value of $P_{TE}$ and smaller value of $\kappa$ (i.e., $P_{TE} = 1.74$% and $\kappa = 91.60$%), implying that the importance of introducing the smoothness matrix in the proposed network. For another group, the influence of parameters $\gamma$ and $\alpha$ also requires to be investigated. Specifically, fixing $L = 2$ and $\theta = 0.5$, $\gamma$ and $\alpha$ are tuned together and defined in the finite set $\{0.01, 0.01, 0.1, 0.2, 0.5, 1, 2, 5, 10\}$ and $\{0.001, 0.01, 0.1, 0.2, 0.5, 1, 2\}$, respectively. Fig. 5 gives the performance in terms of $\gamma$ and $\alpha$ on Foshan data set. It is easy to find that the proposed network achieves the slightly fluctuant performance in the vicinity of $\gamma = 2$ and $\alpha = 0.5$.

4.3. Experimental results and discussion

In the experiments, five existing detection algorithms, i.e., LiRCD (Xiong et al., 2012), PCAKM (Celik, 2009), GWDM (Li et al., 2017), PCANet (Gao et al., 2016), and NR-ELM (Gao et al., 2016) are utilized for comparison purpose. Among them, LiRCD utilizes a $3 \times 3$ window to represent a homogenous area, while PCAKM employs the default parameters, i.e., $p = 4$, $S = 3$. GWDM is implemented with $U_1 = 0.05$ and $U_0 = 0.4$, and leaves the total numbers of scale and orientation (i.e., $S$ and $O$) to be tuned, where $U_1$ and $U_0$ denote the lower and upper center frequencies of interest. For the PCANet algorithm, its
neighborhood window size is set to be $5 \times 5$ for all the considered data sets. In NR-ELM, the neighborhood window size is set as $3 \times 3$ for Foshan and Florence data sets and $7 \times 7$ for San Francisco and Yingshang data set, respectively. It should be noted that here PCANet and NR-ELM employ the same pre-classification algorithm as the proposed method for a fair comparison. As far as the proposed method (called nsNMFNet) is concerned, we respectively set $L, \theta, \gamma$, and $\alpha$ to be $2, 0.5, 2,$ and $0.01$, respectively. The ground-truth image and resulting CMs achieved

### Table 1
Quantitative measures for different values of $L$ on Foshan data set.

<table>
<thead>
<tr>
<th>Number of layers $L$</th>
<th>$FA, %$</th>
<th>$P_{TA}, %$</th>
<th>$MD, %$</th>
<th>$P_{MD}, %$</th>
<th>$TE, %$</th>
<th>$P_{TE}, %$</th>
<th>$\kappa, %$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 1$</td>
<td>379</td>
<td>0.32</td>
<td>3461</td>
<td>21.53</td>
<td>3840</td>
<td>2.84</td>
<td>85.21</td>
</tr>
<tr>
<td>$L = 2$</td>
<td>982</td>
<td>0.83</td>
<td>1117</td>
<td>6.95</td>
<td>2099</td>
<td>1.55</td>
<td>92.56</td>
</tr>
<tr>
<td>$L = 3$</td>
<td>931</td>
<td>0.78</td>
<td>1354</td>
<td>8.42</td>
<td>2285</td>
<td>1.69</td>
<td>91.84</td>
</tr>
<tr>
<td>$L = 4$</td>
<td>1276</td>
<td>1.07</td>
<td>1280</td>
<td>7.96</td>
<td>2556</td>
<td>1.89</td>
<td>90.97</td>
</tr>
</tbody>
</table>

Fig. 3. Visualized results with respect to the different values of $L$ on Foshan data set, (a) $L = 1$, (b) $L = 2$, (c) $L = 3$, and (d) $L = 4$.

Fig. 4. Performance about $P_{TE}$ and $\kappa$ acquired by utilizing different values of $\theta$ on Foshan data set.

4.3.1. Experiments on Foshan data set

In this experiment, the effectiveness of the proposed algorithm is verified on Foshan data set. When $L, \theta, \gamma$, and $\alpha$ equal $2, 0.5, 2,$ and $0.01$, respectively, it achieves the quasi-optimal performance. Fig. 6 gives the ground-truth image and resulting CMs achieved by the various detection methods. Additionally, Table 2 summarizes the $FA, P_{TA}, MD, P_{MD}, TE, P_{TE}$, and $\kappa$ values for quantitative comparison. In LiRCD, the detection result seems to be extremely poor, yielding the largest $P_{TE}$ value of $4.29\%$ and the smallest $\kappa$ value of $76.10\%$. For the PCAKM and GWDM algorithms, although the neighborhood information is considered by using sliding window and Markov random field (MRF), respectively, they still generate many false alarms. By contrast, the PCANet algorithm gives better detection performance in terms of $P_{FA}$. NR-ELM employs the supervised ELM classifier to partition the intermediate pixels into the changed and unchanged classes, obtaining a smaller false alarm rate. Although there are respective advantages in $P_{TA}$ and $P_{MD}$ for each method, nsNMFNet achieves the smallest $P_{TE}$ value of $1.55\%$ and the greatest $\kappa$ value of $92.56\%$. This is because nsNMFNet can learn nonlinear, localized, and discriminative features from the original SAR images.

4.3.2. Experiments on San Francisco data set

For the experiments on San Francisco data set, in order to achieve the quasi-optimal performance, $L, \theta, \gamma$, and $\alpha$ are set to be $2, 0.5, 1,$ and $0.01$, respectively. The ground-truth image and resulting CMs achieved

(a)  
(b)
by all considered methods are shown in Fig. 7. By visual comparison, we can find that the LiRCD algorithm has the highest missed detection, and the corresponding CM is the more dissimilar to the ground-truth image. PCAKM further reduces the missed detection, but leads to the largest false alarm. For the GWDM, PCANet, and NR-ELM algorithms, they yield similar detection performance on San Francisco data set according to Table 3. By contrast, although there are respective advantages in $P_{FA}$ and $P_{MD}$ for the compared methods, nsNMFNet achieves better values in metrics such as $P_{TE}$ and $\chi$, and the resulting CM is also much closer to the ground-truth image. Therefore, nsNMFNet outperforms other methods both in the qualitative analysis and quantitative measures on the San Francisco data set.

### 4.3.3. Experiments on Florence data set

For the experiments on Florence data set, $L$, $\delta$, $\gamma$, and $\alpha$ are set as 2, 0.6, 0.1, and 2, respectively. Fig. 8 presents the ground-truth image and resulting CMs achieved by the various detection methods. From visual comparison in Fig. 8, LiRCD fails to detect the changed pixels, while PCAKM, GWDM, PCANet, and NR-ELM achieve much better change detection results. Here, PCANet generates less noise spots, thus giving the smaller false alarm rate. This may because PCANet possesses the hierarchical architecture of traditional convolutional neural networks (CNN) by using the PCA algorithm to construct the parameters of convolution filters. In addition, Table 4 summarizes the $P_{FA}$, $P_{MD}$, $P_{TE}$, and $\chi$ values of various methods for quantitative comparison, respectively. It is obvious from Table 4 that LiRCD produces the $P_{TE}$ value of 2.21% and the $\chi$ value of 87.39%. PCAKM, GWDM, PCANet, and NR-ELM achieve the similar performance, i.e., 1.82%, 1.79%, 1.90%, and 1.78% about $P_{TE}$ and 89.92%, 90.13%, 89.42%, and 90.23% about $\chi$, respectively. However, nsNMFNet performs the best by visual comparison with the above methods. Specifically, nsNMFNet achieves the $P_{TE}$ and $\chi$ gains of 0.07 and 0.38 in percentage over the NR-ELM algorithm, respectively. This is due to the fact that nsNMFNet can transform the original image into a suitable latent feature subspace, and the effective information is extracted. As a result, the quantitative comparison demonstrates the effectiveness of the proposed method.

### 4.3.4. Experiments on Yingshang data set

For the experiments on Yingshang data set, $L$, $\delta$, $\gamma$, and $\alpha$ are set as 2, 0.3, 0.1, and 1, respectively. The ground-truth image and resulting

---

**Table 2**

Quantitative measures for all considered methods on Foshan data set.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>$P_{FA}$, %</th>
<th>$P_{MD}$, %</th>
<th>$P_{TE}$, %</th>
<th>$\chi$, %</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiRCD</td>
<td>0.16</td>
<td>5592</td>
<td>34.79</td>
<td>57.85</td>
</tr>
<tr>
<td>PCAKM</td>
<td>3.20</td>
<td>929</td>
<td>5.78</td>
<td>47.31</td>
</tr>
<tr>
<td>GWDM</td>
<td>4.07</td>
<td>385</td>
<td>2.40</td>
<td>50.10</td>
</tr>
<tr>
<td>PCANet</td>
<td>2.45</td>
<td>1033</td>
<td>6.43</td>
<td>39.49</td>
</tr>
<tr>
<td>NR-ELM</td>
<td>2.05</td>
<td>1188</td>
<td>7.39</td>
<td>36.24</td>
</tr>
<tr>
<td>nsNMFNet</td>
<td>0.83</td>
<td>1117</td>
<td>6.95</td>
<td>20.99</td>
</tr>
</tbody>
</table>

CMs achieved by the various detection methods are presented in Fig. 9. From visual comparison in Fig. 9, LiRCD generates the worst detection performance. PCAKM, GWDM, and PCANet further decrease the false alarm rate by using feature extraction methods based on PCA, Gabor wavelet, and PCANet, respectively. NR-ELM achieves much better change detection results. However, nsNMFNet performs the best by visual comparison with the above methods, which can be also evidenced according to the smallest \( P_{TE} \) value of 2.68% and the greatest value of 94.48% in Table 5.

4.4. Influence of nonlinear activation function and computational time

As is well known, nonlinear activation function between the layers of neural network is important to learn the nonlinear representation from observation data. As such, nsNMFNet with linear activation function (called nsNMFNet-LAF) is added for a comparison. Besides, this subsection also gives the computational time of the proposed nsNMFNet method as well as other compared ones. Similarly, we also take the Foshan data set as an example to conduct the experiments for avoiding the descriptive redundancy. From Section 4.2, when the number of layers \( L \) equals 2, the proposed method achieves the best detection performance, so we discuss the performance of nsNMFNet-LAF only under some fixed parameters \( L = 2, \theta = 0.5 \). According to Fig. 5, different values of \( \gamma \) and \( \alpha \) evidently influence the final performance. Hence, in order to better demonstrate the effect of linear activation function, the experiments are conducted under the different values of \( \gamma \) and \( \alpha \), defined in the finite set \{0.001, 0.01, 0.1, 0.2, 0.5, 1, 2, 5, 10\} and \{0.001, 0.01, 0.1, 0.2, 0.5, 1, 2\}, respectively. Fig. 10 shows the detection performance of nsNMFNet-LAF about (a) \( P_{TE} \) and (b) \( \alpha \) in terms of \( \gamma \) and \( \alpha \) on Foshan data set. It is obvious that the curve’s tendency is similar with that of Fig. 5, and the best detection performance is shown in the vicinity of \( \gamma = 5 \) and \( \alpha = 0.01 \). Overall, the performance of nsNMFNet-LAF is poorer than that of nsNMFNet, implying that nonlinear activation function is beneficial to learn nonlinear features from observation data.

Table 6 shows computational time for all considered methods on Foshan data set. It can be observed that the LiRCD and PCAKM algorithms have a lower computational time compared with the NR-ELM, GWDM, and PCANet algorithms. By contrast, the proposed method not only achieves better detection performance, but also has the higher computational efficiency than the GWDM and PCANet algorithms. nsNMFNet requires to be trained according to pretraining stage and fine-tuning stage, so its computational time is more than that of NR-ELM. In addition, it is obvious that its computational time also generally depends on the numbers of layers and iterations, and the size of the input images.

5. Conclusion

In this paper, we have proposed a SAR image change detection method based on deep nsNMF network with semi-supervised learning. The deep nsNMF network is first constructed by introducing the nonlinear activation function and the ELM classifier to learn nonlinear, localized, and discriminative data representations from complex real data. In addition, semi-supervised learning strategy is adopted into deep nsNMF network to use partial labeled samples and unlabeled samples for training. The learning process of the proposed network consists of pretraining stage and fine-tuning stage, whose corresponding update rules are also developed in detail. The experimental results on four pairs of SAR images demonstrate that nsNMFNet can achieve better detection results than other compared methods.
Table 4
Quantitative measures for all considered methods on Florence data set.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>FA</th>
<th>Pfa, %</th>
<th>MD</th>
<th>Pmd, %</th>
<th>TE</th>
<th>Pte, %</th>
<th>x</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiRCD</td>
<td>286</td>
<td>0.20</td>
<td>3245</td>
<td>19.13</td>
<td>3531</td>
<td>2.21</td>
<td>87.39</td>
<td></td>
</tr>
<tr>
<td>PCAKM</td>
<td>483</td>
<td>0.34</td>
<td>2421</td>
<td>14.28</td>
<td>2904</td>
<td>1.82</td>
<td>89.92</td>
<td></td>
</tr>
<tr>
<td>GWDM</td>
<td>548</td>
<td>0.38</td>
<td>2309</td>
<td>13.62</td>
<td>2857</td>
<td>1.79</td>
<td>90.13</td>
<td></td>
</tr>
<tr>
<td>PCANet</td>
<td>479</td>
<td>0.33</td>
<td>2555</td>
<td>15.07</td>
<td>3034</td>
<td>1.90</td>
<td>89.42</td>
<td></td>
</tr>
<tr>
<td>R-ELM</td>
<td>693</td>
<td>0.48</td>
<td>2158</td>
<td>12.72</td>
<td>2851</td>
<td>1.78</td>
<td>90.23</td>
<td></td>
</tr>
<tr>
<td>nsNMFNet</td>
<td>613</td>
<td>0.43</td>
<td>2122</td>
<td>12.51</td>
<td>2735</td>
<td>1.71</td>
<td>90.61</td>
<td></td>
</tr>
</tbody>
</table>

Table 5
Quantitative Measures for All Considered Methods on Yingshang Data Set

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>FA</th>
<th>Pfa, %</th>
<th>MD</th>
<th>Pmd, %</th>
<th>TE</th>
<th>Pte, %</th>
<th>x</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiRCD</td>
<td>25859</td>
<td>6.28</td>
<td>2423</td>
<td>0.79</td>
<td>28282</td>
<td>3.93</td>
<td>92.05</td>
<td></td>
</tr>
<tr>
<td>PCAKM</td>
<td>2634</td>
<td>0.64</td>
<td>19601</td>
<td>6.36</td>
<td>22235</td>
<td>3.09</td>
<td>93.65</td>
<td></td>
</tr>
<tr>
<td>GWDM</td>
<td>3989</td>
<td>0.97</td>
<td>20978</td>
<td>6.82</td>
<td>24967</td>
<td>3.47</td>
<td>92.86</td>
<td></td>
</tr>
<tr>
<td>PCANet</td>
<td>3002</td>
<td>0.73</td>
<td>18197</td>
<td>5.91</td>
<td>21199</td>
<td>2.94</td>
<td>93.95</td>
<td></td>
</tr>
<tr>
<td>R-ELM</td>
<td>3051</td>
<td>0.74</td>
<td>17868</td>
<td>5.80</td>
<td>20919</td>
<td>2.91</td>
<td>94.03</td>
<td></td>
</tr>
<tr>
<td>nsNMFNet</td>
<td>2056</td>
<td>0.50</td>
<td>17274</td>
<td>5.61</td>
<td>19330</td>
<td>2.68</td>
<td>94.48</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 8. Visualized results for (a) the ground-truth image and various detection methods on Florence data set: (b) LiRCD, (c) PCAKM, (d) GWDM ($S = 4$, $O = 8$), (e) PCANet, (f) NR-ELM, and (g) nsNMFNet.

Fig. 9. Visualized results for (a) the ground-truth image and various detection methods on Yingshang data set: (b) LiRCD, (c) PCAKM, (d) GWDM ($S = 4$, $O = 6$), (e) PCANet, (f) NR-ELM, and (g) nsNMFNet.
Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank European Space Agency for providing some experimental data sets. They would also like to thank the Editor-in-Chief, the Associate Editor, and the anonymous reviewers for their careful reading and valuable comments, which have greatly improved the technical quality and presentation of this paper. This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 61871335 and 61771351), in part by the Frontier Intersection Basic Research Project for the Central Universities (Grant No. A0920502051814-5), and in part by the Doctoral Innovation Fund Program of Southwest Jiaotong University (Grant No. D-CX201925).

Appendix A

In the appendix, we focus on the derivations of Eqs. (23), (25), and (27) as follows.

\[
\frac{\delta J}{\delta l} = \frac{\delta J}{\delta Z} (s_i F_i)^T = \left[ \frac{\delta J}{\delta Z} (s_i F_i) \right]_{ij} = \frac{\delta J}{\delta Z} (s_i F_i)^T,
\]

where \( F_i \) and we have

\[
\frac{\delta J}{\delta Z} (s_i F_i) = \sum_j \frac{\delta J}{\delta Z} (s_i F_i)_{ij} = \left[ \frac{\delta J}{\delta Z} (s_i F_i)_{ij} \right]_{ij}.
\]

Similarly, Eq. (25) can be obtained. By combining Eq. (25), Eq. (27) is given by

\[
\frac{\delta J}{\delta F_i} = (Z_i S_i)^T \left[ \frac{\delta J}{\delta Z} (Z_i S_i) \right] + W \left[ \frac{\delta J}{\delta F_i} \right]^T,
\]

where \( H = g(F_i^T W) \).

References


