Modified Diversity of Class Probability Estimation Co-training for Hyperspectral Image Classification

Yan Ju, Lingling Li, *Member, IEEE*, Licheng Jiao, *Fellow, IEEE*, Zhongle Ren, *Student Member, IEEE*,
Biao Hou, *Member, IEEE*, Shuyuan Yang, *Senior Member, IEEE*

Abstract—**Due to the limited amount and imbalanced classes of labeled training data, the conventional supervised learning can not ensure the discrimination of the learned feature for hyperspectral image (HSI) classification. In this paper, we propose a modified diversity of class probability estimation (MDCPE) with two deep neural networks to learn spectral-spatial feature for HSI classification. In co-training phase, recurrent neural network (RNN) and convolutional neural network (CNN) are utilized as two learners to extract features from labeled and unlabeled data. Based on the extracted features, MDCPE selects most credible samples to update initial labeled data by combining k-means clustering with the traditional diversity of class probability estimation (DCPE) co-training. In this way, MDCPE can keep new labeled data class-balanced and extract discriminative features for both the minority and majority classes. During testing process, classification results are acquired by co-decision of the two learners. Experimental results demonstrate that the proposed semi-supervised co-training method can make full use of unlabeled information to enhance generality of the learners and achieve favorable accuracies on all three widely used data sets: Salinas, Pavia University and Pavia Center.**

Index Terms—**Co-training, recurrent neural network (RNN), convolutional neural network (CNN), spectral-spatial feature learning, hyperspectral image classification**

I. Introduction

WITH the development of satellite imaging technique, HSI with high resolution in spectral as well as spatial is available. Observed by thousands of continuous electromagnetic spectrums, each pixel of HSI contains abundant terrain information. Hence, automatic interpretation of HSI has great potential in urban development [1], [2], change detection of landscape [3], [4], scene interpretation [5] and regulation of natural resource [6], [7], etc. HSI classification is an essential task and lays the ground for the applications mentioned above.

Numerous traditional methods are proposed for HSI classification during past years, such as decision trees [8], random forest [9], support vector machine (SVM) [10], [11], etc. Nevertheless, these approaches utilize spectral information but ignore abundant spatial information [12], which has been proved to be useful in improving the interpretation of HSI data [13]. Thus, more and more methods based on spectral-spatial feature are presented and improve accuracy significantly [14]–[17] for HSI classification.

With the increase of spectral channels and spatial variability of spectral signature, the shallow-layer classifiers mentioned above can not ensure the validity of the learned feature. As the most prosperous machine learning method nowadays, deep learning (DL) are able to extract more discriminative feature and achieve better performance than traditional shallower classifiers for HSI classification [12], [18], [19]. Mou *et al.* [18] proposed a novel RNN with a new activation function and modified gated recurrent unit (GRU) for HSI classification. This model takes the spectra as a sequential signal for the first time and acquires good performance only based on spectral feature. Spectral-spatial feature is proved to be effective in HSI classification, which promotes the study of classification models based on deep spectral-spatial feature of HSI [19]–[22]. For example, Zhao *et al.* [20] combined the CNN-based spatial feature and the BLDE-based spectral feature into image interpretation, and proposed a spectral-spatial feature-based classification (SSFC) method, which achieved good classification performance. A 3-D CNN-based feature extraction model is also proposed in [21] in order to extract discriminative spectral-spatial feature of HSI and the classification results also show fully using of spectral-spatial information can improve accuracy considerably.

The deep models mentioned above are supposed to be trained with large number of labeled data in a supervised way. As the depth of the deep model increases, more and more parameters need to be trained and more labeled data is demanded to prevent overfitting. While in practical classification issues, labeled data is much less than unlabeled data, as manual labeling of remote sensing data is costly in finance and manpower. Furthermore, utilizing costly labeled data while neglecting accessible unlabeled data is a great waste. In such case, semi-supervised learning (SSL) is proposed to enlarge the labeled data by recovering unlabeled samples.

There are many classic methods for SSL, such as generative model with EM [23], self-training [24], co-training [25], transductive SVM [26], graph based method [27], etc. Recently, plenty of semi-supervised models have been proposed for HSI classification [28]–[31]. In [29], a modified co-training method for HSI classification is presented. This method extracts the spectral feature and 2-D Gabor feature from spatial domain in order to co-train two classifiers from distinct views. Shallow feature is extracted to handle complex classification problems in these semi-supervised models, which is not as robust and

This work was supported in part by the Major Research Plan of the National Natural Science Foundation of China (No. 91438201 and No. 91438103), the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048), the Fundamental Research Funds for the Central Universities(No. XJS17108) and the China Postdoctoral Fund (No. 2017M613081)

Y. Ju, L. Li, L. Jiao, Z. Ren, B. Hou and S. Yang are with the Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an, Shaanxi Province, 710071, China (Email: yan.juchn@gmail.com).

discriminative as deep feature is. Hence, Ratle *et al.* [32] combined deep learning with semi-supervised learning and proposed a semi-supervised neural networks (SSNNs) to deal with HSI classification problems. By adding a regularizer to the loss function for training neural networks, this method promotes classification accuracy considerably. However, without studying the class balance, it fails to represent the distributive characteristics of HSI data and provide poor accuracies across all classes [33].

In conclusion, there are two main difficulties in fulfilling accurate HSI classification [33], [34]: 1) classifiers are prone to be overfitting and can not extract discriminative feature because of limited labeled HSI data with many spectral channels and great spatial variability; 2) classifiers tend to provide a severely imbalanced accuracies across classes, with the majority class having high accuracy and the minority class having abysmal accuracy if labeled data is class-imbalanced.

To solve these problems, a semi-supervised method is proposed based on MDCPE co-training of RNN and CNN for HSI classification. Inspired by Mou *et al.* [18], we take all spectra of a pixel as sequence to train the RNN, which is good at exploiting spectral feature of HSI. Spatial feature is extracted with CNN after reducing dimension of HSI by principal component analysis (PCA). Based on the extracted spectral and spatial features, MDCPE chooses credible samples from each class to update labeled data. Then two networks are trained with new updated labeled data iteratively and co-decide the classification results. The three major contributions of the proposed method are listed as follows.

1) A semi-supervised co-training MDCPE is proposed in this paper, which presents two advantages: a) MDCPE enlarges insufficient labeled data by recovering unlabeled samples, which avoids of wasting unlabeled HSI information and enhances the robustness of extracted feature; b) k-means clustering is adopted to keep the updated data class-balanced, which helps the learners extract feature of all classes comprehensively and thus acquire a balanced accuracy across classes.
2) RNN and CNN cooperate as two learners of MDCPE to extract spectral-spatial feature of HSI. Taking all HSI spectra as sequential signals, RNN is expert in exploring relationship between spectra and extracting discriminative spectral feature. CNN is utilized to learn robust and effective spatial feature as its deep layer structure and superiority in learning of neighbor information.
3) In this paper, RNN and CNN are trained with MDCPE co-training from spectral and spatial views. Deep neural networks, intergraded with co-training method, not only adept in extracting robust and effective feature, but also promote accuracy and strengthen generation ability by making full use of unlabeled data.

The remainder of this paper is organized as follows. Section II briefly introduces the preliminaries of related methods. The proposed method for HSI classification, including MDCPE co-training and two learners (RNN, CNN) are described in detail in section III. Network configurations, experimental results, and discussion are provided in section IV. Finally, section V concludes this paper briefly.

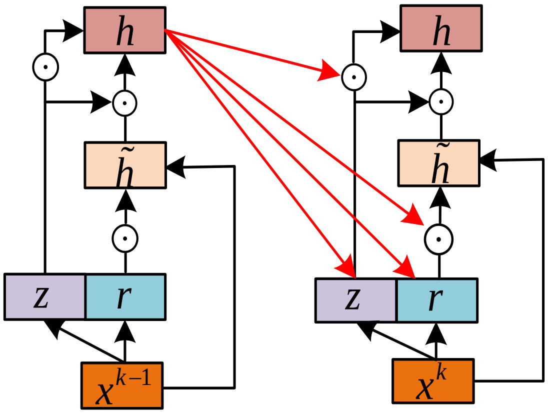

Fig. 1. Graph model of GRU. z and r are the update gate and reset gate. The new memory content is denoted by $\tilde{h}$.

II. Preliminaries

In this section, we mainly recall background information of traditional DCPE co-training and RNN, CNN classification models.

A. DCPE Co-training

Co-training, as a complex generative branch of SSL model, is firstly proposed for classification of web pages [25]. In traditional co-training approach, two classifiers are trained separately based on two sufficient and redundant feature subsets (views), and then select unlabeled data with the most credible predicted labels as the new labeled data [28].

DCPE is first proposed in [35], in which the diversity based class probability estimation from two classifiers are utilized. The samples that have same labels and biggest difference class probability are chose to update the labeled data in DCPE. New recovered data has the decision agreement and the largest diversity in class probability estimation between two learning algorithms [35].

Suppose that there are k classes $c_i, i = 1, 2, \cdots, k$ in classification issue. For each data x, the class probability estimation for each class is defined as $P(c_i|x)$. Data x will be allocated the most credible label, $L(x) = argmax(P(c_i|x))$. In DCPE, the samples that have same predicted labels of two classifiers are allocated the predicted labels. Labels of the samples that have biggest difference class probability are calculated by

$$label(x) = \arg\max(P_2(c_i|x)) - P_1(c_i|x)) \qquad (1)$$

$$label(x) = \arg\max(P_1(c_i|x)) - P_2(c_i|x)) \qquad (2)$$

where $P_1(c_i|x)$ and $P_2(c_i|x)$ denote the class probability estimation for each category of classifier 1 and 2. Equations (1), (2) are utilized to update the training samples of classifier 1 and 2, respectively. When we ensemble two classifiers given each testing data during the testing process, the combined classifier computes the probability $P(c_i|x)$ of class c_i via

$$P(c_i|x) = P_1(c_i|x)P_2(c_i|x) \qquad (3)$$

Then the labels of the testing data is decided by $argmax(P(c_i|x))$.

B. Recurrent Neural Network

RNN is a class of artificial neural network that extends conventional feedforward neural network with loops in connections [18]. RNN network is able to process sequential signal by using a recurrent hidden state that is activated at each step depending on the previous hidden state [36], [37]. In this way, it can exploit temporal connection between input units and induce output of next step. RNN have gained significant attention for solving many challenging problems involving sequential data analysis, such as language modeling [38], machine translation [39], and speech recognition [40], [41].

In the traditional RNN model [36], for a sequential input $x = (x_1, x_2, \cdots, x_T)$, where x_i represents the input of ith time step. Its hidden vector sequence $h = (h_1, h_2, \cdots, h_T)$ is calculated by:

$$h_t = \varphi(W_h \odot h_{t-1} + U_h \cdot h_{t-1}) \tag{4}$$

where t ranges from 1 to T. W_h, U_h are coefficient matrixes for the input at present step and the state of hidden unit at previous step. $\varphi(\cdot)$ is nonlinear activation function of hidden layer and $y = (y_1, y_2, \cdots, y_T)$ is the output vector. In some cases such as classification of HSI, y_T is the only output of RNN as predicted label.

Gated recurrent unit (GRU) is introduced to learn long-term dependencies and alleviate gradient vanishing problem [42], [43]. The graph model of GRU is shown in Fig. 1. Moreover, GRU has few parameters and is more suitable for small number of samples. By constructing some gates, GRU decides whether to update the cell state or to forget the former cell memory. Traditional RNN calculates hidden state by Equation (4), which merely calculates the weighted sum of inputs and activates with a nonlinear function. However, a GRU-based recurrent layer calculates the activation of gated recurrent units by

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{5}$$

where z_t denotes update gate that determines which part of the activation will be updated and $\odot$ means an elementwise multiplication. h_{t-1} is the activation of gated recurrent units at previous step, and $\tilde{h}_t$ represents candidate activation. The update gate is calculated by Equation (6), and candidate activation is obtained by Equation (7).

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{6}$$

$$\tilde{h}_t = \tanh(W x_t + r_t \odot U h_{t-1}) \tag{7}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{8}$$

where $\sigma(\cdot)$ denotes a logistic sigmoid function and $\tanh(\cdot)$ is the hyperbolic tangent function. W and U terms are weight matrixes, and r_t represents the reset gate that determine whether to abandon the previous activation or not when generating candidate activation. r_t is calculated by Equation (8).

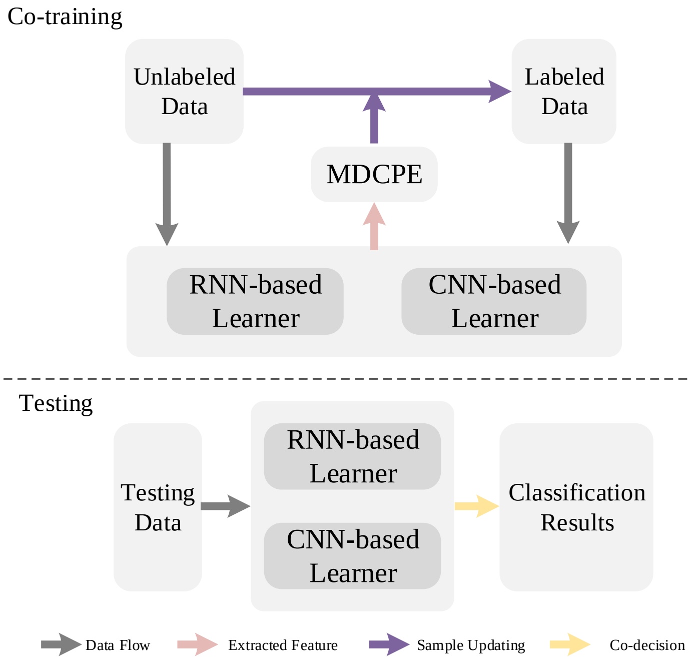

Fig. 2. Flowchart of the proposed method.

C. Convolutional Neural Network

A complete CNN structure consists of convolutional layers and pooling layers, usually followed several fully connected layers and a softmax classification layer [44]. 3-D convolutional kernel is utilized in 3-D CNN [21], which is often used to extract spectral and spatial features simultaneously in HSI classification models.

In the ith layer, the value of a neuron n_{ij}^{xyz} at the position of the jth feature map in 3-D CNN is calculated by Equation (9)

$$n_{ij}^{xyz} = f(\sum_{m} \sum_{h}^{H_i-1} \sum_{l}^{L_i-1} \sum_{d}^{D_i-1} w_{ijm}^{hld} v_{(i-1)m}^{(x+h)(y+l)(z+d)} + b_{ij}) \tag{9}$$

where m denotes the feature maps in the $(i-1)$th layer that are connected to the jth feature map. H_i, L_i and D_i denote the height, the width and the dimension (along the spectral dimension) of the convolutional kernel. w_{ijm}^{hld} means the weight at the position (h, l, d) that is connected to the mth feature map, and b_{ij} is the bias of the jth feature map in the ith layer. $f(\cdot)$ denotes nonlinear activation function, which is sigmoid activation function commonly:

$$f(x) = (1 + \exp^{(-x)})^{-1} \tag{10}$$

III. Proposed Method

The flowchart of the proposed MDCPE co-training is illustrated in Fig. 2. During training process, as shown on top, RNN and CNN are pre-trained on labeled data and then are used to extract spectral-spatial feature of unlabeled data. Based on the feature, MDCPE is utilized to select the most credible unlabeled samples to replenish labeled data. New labeled data is used to train the classifiers again and loop this process until MDCPE meets the stop criteria. In addition, part of labeled data are utilized as validation data to monitor training process for saving the model with highest validation accuracy. During testing process, as shown in bottom of Fig. 2, classification results are acquired by co-decision of co-trained RNN and CNN with Equation (3).

It is obvious that three sections play important roles in the proposed method: MDCPE co-training, RNN-based spectral feature learner and CNN-based spatial feature learner. We will introduce our innovative work on three parts in section III-A, III-B and III-C, respectively.

A. *MDCPE Co-training*

Traditional DCPE is a single view method, which is unnecessary to split feature to different groups. Rather than learning on the same feature, MDCPE expands DCPE to spectral and spatial views to extract discriminative spectral-spatial feature with RNN and CNN, which enhances effectiveness of extracted feature and promotes HSI classification results.

Furthermore, traditional DCPE updates labeled data without considering unbalance between classes in updated data, which diminish the comprehensiveness of extracted features and the overall accuracy. The classes that are allocated too many training samples will be over-trained and get high accuracies, while the performance is poor when the classes have inadequate updated samples. In MDCPE, k-means clustering is combined with DCPE to ensure the balance in updated data. The most credible samples selected to replenish labeled data are determined by results of two learners and k-means clustering.

MDCPE co-training is described in the Algorithm 1 in detail. Different learners have diverse calculation mechanisms of class probability. In MDCPE co-training, we utilize Softmax layers of deep neural network (RNN, CNN) to estimate the class probability.

Algorithm 1 The proposed MDCPE co-training.

```
Input: $C_1$: learner 1; $C_2$: learner 2; $D_t$: labeled data set; $D_u$:
    unlabeled data set; $c_i$: class $c_{i,i=1,2,\cdots,k}$
Output: $\widetilde{C_1}$: co-trained learner 1; $\widetilde{C_2}$: co-trained learner 2
allocate $S_1 = D_t$, $S_2 = D_t$;
train $C_1$, $C_2$ on $S_1$, $S_2$;
record predicted label, class probability estimation and
    extracted feature of two learners as $L_1(\cdot)$, $L_2(\cdot)$ , $P_1(\cdot)$,
    $P_2(\cdot)$ and $F_1(\cdot)$, $F_2(\cdot)$;
record predicted label of k-means clustering as $L_{km}(\cdot)$;
repeat
    choose $u_s$,$u_{1m}$,$u_{2m}$ from $D_u$ to satisfy equations (11),
        (12) and (13);
    select one sample belonged to class $c_i$ from $D_t$ randomly to form cluster center $O = \{o_i\}$, $i = 1, 2, \cdots, k$;
    centered on $F_1(O)$, cluster $F_1(u_s)$, $F_1((u_{1m})$ with k-means clustering;
    similarly, cluster $F_2(u_s)$, $F_2((u_{2m})$, centered on
        $F_2(O)$;
    update $S_1 = S_1 + D_1$, which satisfies equations (14);
    update $S_2 = S_2 + D_2$, which satisfies equations (17);
    eliminate $D_1$, $D_2$ from $D_u$;
    train $C_1$, $C_2$ on new labeled data $S_1$, $S_2$.
until $D_u = \varnothing$
```

In this algorithm, we predict all unlabeled data in each iteration rather than choosing some samples pool as DCPE does, because we want to enlarge the searching range and ensure there are enough samples in each class to be selected. During selecting samples to update labeled data, the sample u_s that satisfies Equation (11) is selected.

$$L_1(u_s) = L_2(u_s) \tag{11}$$

Then we need to select samples that have highest diversity of class probability. When updating samples of RNN, labels of samples that have diversity of class probability estimation(CPE) are calculated by

$$L_1(u_{1m}) = \arg\max(P_2(u_{1m}) - P_1(u_{1m})) \tag{12}$$

Similarly, when updating samples of CNN, labels of the samples are calculated by

$$L_2(u_{2m}) = \arg\max(P_1(u_{2m}) - P_2(u_{2m})) \tag{13}$$

In order to keep classes of updated samples balanced, we utilize k-means clustering to classify $F_1(u_s)$, $F_1((u_{1m})$, $F_2((u_{2m})$, from which we select fixed number of data from each class to update the labeled data. For RNN, updated data set D_1 that owns same labels and highest CPE is calculated by

$$D_1 = \{d_{1s_i}\} \bigcup \{d_{1m_i}\} \tag{14}$$

where $i = 1, 2, \cdots, k$. d_{1s_i} and d_{1m_i} are selected to update the labeled data of RNN learner and their labels are decided by

$$L_1(d_{1s_i}) = L_{km}(d_{1s_i}) = c_i \tag{15}$$

$$L_1(d_{1m_i}) = L_{km}(d_{1m_i}) = c_i \tag{16}$$

When selecting updated data for CNN, samples in D_2 data set are supposed to satisfy the follows:

$$D_2 = \{d_{2s_i}\} \bigcup \{d_{2m_i}\} \tag{17}$$

where $i = 1, 2, \cdots, k$ and d_{2s_i}, d_{2m_i} are computed as follows:

$$L_2(d_{2s_i}) = L_{km}(d_{2s_i}) = c_i \tag{18}$$

$$L_2(d_{2m_i}) = L_{km}(d_{2m_i}) = c_i \tag{19}$$

Rather than choosing cluster centers randomly, we select one labeled sample from each class, denoted as O and extract spectral feature $F_1(O)$ and spatial feature $F_2(O)$ as clustering centers of two classifiers. By adding prior labeled information of classes, we alleviate the influence of choosing initial cluster centers randomly and thus can get better clustering performance. After clustering, choosing the samples whose predicted labels of classifiers are same with results of k-means clustering from each class to update labeled data.

It's important to note that the selected unlabeled data will be remove from original unlabeled data set. New recovered data selected by the MDCPE has the decision agreement and the largest diversity in class probability estimation between two classifiers. In addition, updated data is balanced between classes, which contributes to improve overall classification results.

B. RNN-based Spectral Feature Learner

Since the temporal variability of a sequential signal is similar to the spectral variability of a hyperspectral pixel, the same idea can be transferred to hyperspectral pixel vector [18]. Considering all spectra of a hyperspectral pixel as a sequence, we design a RNN learner, as shown in Fig. 3, to extract spectral feature of HSI. Input of the classifier is a spectral vector of a HSI pixel x, where its kth spectral band is defined as x^k. Output is the predicted label of pixel x. The detailed process is summarized as follows.

1) Feed x^k into the input layer of network.
2) The recurrent layer calculates the hidden state of current spectral band k after receiving the x^k and saves this information.
3) Next band x^{k+1} is input to recurrent layer. In order to calculate the activation of next $k+1$ spectral band, the candidate activation and the activation of previous band k was linear summed with a controlling parameter of the update gate.
4) Train the learner on labeled training data.
5) Extract spectral feature of training data from FC2 layer after removing the Softmax layer.
6) Input testing pixel sequences into the learner to get classification results.

Considering of insufficient labeled training data and huge data complexity in hundreds of spectral bands, the proposed RNN with GRU classifier has fewer parameters than traditional LSTM unit so it can alleviate overfitting. In addition, by using gates, the GRU will preserve the errors that can be back propagated through sequences and layers. Hence, RNN can learn over many bands of hyperspectral pixels without the risk of the vanishing gradient [18]. Moreover, as a learner of the proposed MDCPE co-training, RNN is good at capturing intrinsic characteristic and extract discriminative spectral feature of HSI.

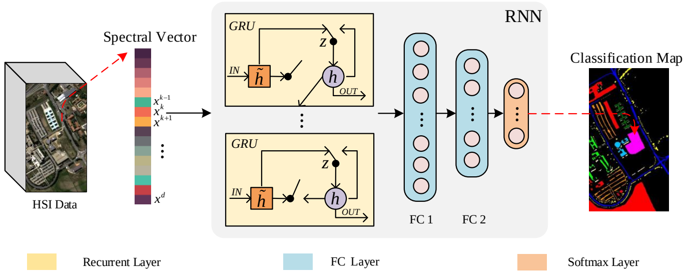

Fig. 3. RNN-based spectral feature learner.

C. CNN-based Spatial Feature Learner

CNN is adopted as another learner of MDCPE co-training to learn effective spatial feature of HSI. Architecture of this learner is shown in Fig. 4. The size of 3-D convolutional kernels of C1 and C2 layers are set as $5 \times 5 \times 5$, $3 \times 3 \times 3$, respectively. Strides of max pooling layers P1, P2 are 2. The fully connected layer FC1 owns 1024 units and the number of units FC2 has is equal to the classes of HSI data. Inputting data of CNN is generated by segmenting data blocks with fixed size after reducing dimensions of HSI by PCA. Spatial feature is extracted form FC2 layer by dismissing the Softmax layer of CNN learner.

As our CNN learner mainly aims at extracting robust spatial feature, dimensions of HSI is first reduced by principal component analysis (PCA), which could decrease the spectral redundancy and save computational time with a large margin [20]. Furthermore, Credibility of samples selected to update labeled data has a big impact on co-training of learner and thus classification accuracy. The 3-D CNN, as a base learner of MDCPE, can extract more discriminative spatial feature and enhance the confidence of updated samples.

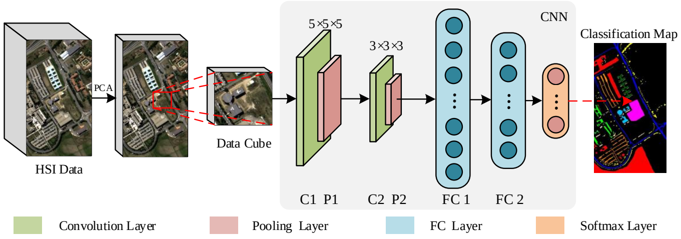

Fig. 4. CNN-based spatial feature learner.

IV. Experiment Results and Discussion

In this section, we introduce three data sets used in our experiment and the configuration of the proposed MDCPE co-training. In addition, classification results based on the proposed method and other comparative methods are presented. All the experiments are conducted on HP z840 workstation equipped with an Intel Xeon E5 with 2.40 GHz and NVIDIA Titan X GPU. TensorFlow 1.2.0 and Keras 2.1.3 are utilized as software platforms.

A. Data Description

1) Salinas Data (SA): SA was collected by the 224-band AVIRIS sensor and is characterized by 3.7-meter spatial resolution. 20 water absorption bands were discarded, with 204 bands left to be applied in the experiment. This data set includes 16 classes ground-truths, with the size of 512×217 pixels. Table I lists all classes and their corresponding training and test samples.

2) Pavia University (PU): This data set is acquired by reflective optics system imaging spectrometer (ROSIS). The ROSIS-03 sensor comprises 115 spectral channels ranging from 430 to 860nm. Once 12 noisy channels have been removed, the remaining 103 spectral channels are adopted in

TABLE I
NUMBER OF TRAINING AND TESTING SAMPLES IN SALINAS DATA SET

Class No.	Class Name	Training			Testing
		Labeled	Validation	Unlabeled	
1	Brocoli_green_weeds_1	15	45	56975	1949
2	Brocoli_green_weeds_2	15	45		3666
3	Fallow	15	45		1916
4	Fallow_rough_plow	15	45		1334
5	Fallow_smooth	15	45		2618
6	Stubble	15	45		3899
7	Celery	15	45		3519
8	Grapes_untrained	50	100		11121
9	Soil_vinyard_develop	30	60		6113
10	Corn_senesced_green_weeds	15	45		3218
11	Lettuce_romaine_4wk	15	45		1008
12	Lettuce_romaine_5wk	15	45		1867
13	Lettuce_romaine_6wk	15	45		856
14	Lettuce_romaine_7wk	15	45		1010
15	Vinyard_untrained	30	60		7178
16	Vinyard_vertical_trellis	15	45		1747
	TOTAL	**305**	**805**	**56975**	**53019**

this paper. The image has 610×340 pixels with a spatial resolution of $1.3m$ per pixel. The available training samples of this data set contains nine urban land-cover types. Table II provides information about different classes and corresponding training and test samples.

3) Pavia Center (PC): PC is also acquired by ROSIS. In this data set, 13 noisy channels have been removed and the remaining 102 spectral channels are utilized in this paper. The image is of 1096×715 pixels covering the center of Pavia, which spatial resolution is 1.3 m per pixel. The available training samples contains nine urban land-cover classes and Table III provides information about training and test samples of different classes.

TABLE II
NUMBER OF TRAINING AND TESTING SAMPLES IN PAVIA UNIVERSITY DATA SET

Class No.	Class Name	Training			Testing
		Labeled	Validation	Unlabeled	
1	Asphalt	35	105	164624	6491
2	Meadows	90	180		18379
3	Gravel	20	40		2039
4	Trees	20	40		3004
5	Painted metal sheets	15	45		1285
6	Bare Soil	35	105		4889
7	Bitumen	15	45		1270
8	Self-Blocking Bricks	20	40		3622
9	Shadows	10	40		897
	TOTAL	**260**	**640**	**164624**	**41876**

We mainly utilize the three data sets mentioned above to evaluate the performance of the proposed method. These data sets are selected for several reasons: firstly, they both have high spatial resolution and cover wide range of real cases, which make the method difficult to classify; secondly, they have different scenery: SA data set is obtained in a mixed vegetation site, while PU and PC data sets are acquired in an urban site. These characters facilitate us to assess the effectiveness and robustness of the proposed method.

TABLE III
NUMBER OF TRAINING AND TESTING SAMPLES IN PAVIA CENTER DATA SET

Class No.	Class Name	Training			Testing
		Labeled	Validation	Unlabeled	
1	Water	285	700	635488	64986
2	Trees	42	140		7416
3	Asphalt	17	50		3023
4	Self-Blocking Bricks	17	50		2618
5	Bitumen	42	140		6402
6	Tiles	42	140		9066
7	Shadows	42	140		7105
8	Meadows	285	700		41841
9	Bare Soil	17	50		2796
	TOTAL	**789**	**2110**	**635488**	**145253**

In our experiment, data set is split randomly into two groups as Fig. 5 shown, training data and testing data (98 % of given labeled data). Training data consists of three parts, labeled samples (about 0.5% of given labeled data), all unlabeled samples and validation data (1.5% of given labeled data).

In SA data set, we choose 3 principal components that could consist of 99% information of the original HSI, and then extract image blocks of when extracting spatial feature based on CNN. Similarly, 4 principal components are chosen in PU data set and 3 components for PC data set. In addition, all training and testing samples are standardized by min-max normalization. In the kth spectral channel, the value of a pixel after normalization $x^{'k}_{ij}$ at the position of the ith line, jth column is calculated by:

$$x^{'k}_{ij} = \frac{x^k_{ij} - \min(x^k_{ij})}{\max(x^k_{ij}) - \min(x^k_{ij})} \tag{20}$$

where $\min(\cdot)$ and $\max(\cdot)$ means the minimum and maximum in all pixels of kth spectral channel, respectively.

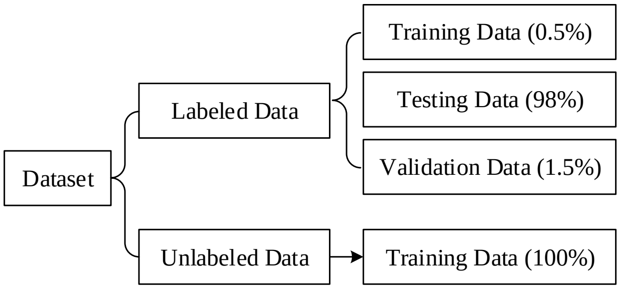

Fig. 5. Data splitting configuration for experiment.

B. Parameter Setting

As two neural network learners of MDCPE co-training, RNN and CNN are all trained by stochastic gradient descent (SGD) algorithm. Because labeled data sets are small, 32 is set as the batch size of two classifiers for SA and PU data sets, and 64 is chose for PC data set. Then we analyze several parameters that impact training process and classification accuracy, such as learning rate, patch size of input block, and iterations of MDCPE, etc. The models with the highest classification accuracy on validation data were saved during the training process of each configuration.

1) Learning rate: Learning rate controls the learning step during training process. Too big learning rate will cause vibration and tend to converge on local optimum, while network converges too slow if it too small. The learning rate of RNN and CNN are chose from $[0.01, 0.008, 0.005, 0.003, 0.001, 0.0006, 0.0003]$ and $[0.001, 0.0008, 0.0005, 0.0003, 0.0001, 0.00006, 0.00003]$, respectively. Based on the classification accuracy, the optimal learning rate of RNN and CNN for SA data set are 0.001 and 0.0003, 0.003 and 0.0001 for PU data set and 0.001 and 0.0001 for PC data set. During the co-training of two learners with MDCPE, the learning rate is set as optimal learning rate and remains unchanged.

TABLE IV
OA OF CNN WITH DIFFERENT INPUT SIZES

Spatial Size	Accuracy (%)		
	SA	PU	PC
15×15	68.21	54.49	72.24
19×19	71.56	58.16	73.76
23×23	74.81	63.92	75.89
27×27	75.49	65.91	75.97

2) Patch size: CNN, as a learner of MDCPE, is utilized to extract spatial feature and its performance badly depends on the size of input block. In order to evaluate the influence of spatial size of input blocks during training of CNN, we set the size of input block 15×15, 19×19, 23×23, 27×27 to extract discriminative spatial feature. Table IV shows classification result on two validation data sets. These results are acquired in 500-epoch training processes for each setting in three data sets. In all data sets, the classification results increase with the spatial size of input cubes, because larger input blocks are, more spatial information is offered. Hence, we fixed the spatial size of input cube to make a fair comparison between different classification methods.

TABLE V
OA OF CNN WITH DIFFERENT REGULARIZERS. THE BEST ACCURACY IN EACH ROW IS SHOWN IN BOLD

Dropout	Accuracy (%)		
	SA	PU	PC
0.2	73.41	63.47	73.58
0.3	**76.12**	**66.63**	73.32
0.4	72.85	65.81	**74.76**
0.5	73.49	55.62	68.73
0.7	42.91	49.84	59.25

3) Dropout: Dropout is adopted to regularize the CNN training process and keep from overfitting. Table V shows several groups of evaluation results with different dropout proportion. These results are also acquired in 500-epoch training processes for each setting in three data sets. As Table V shows, 30% dropout is selected for SA and PU data sets and 40% for PC data set to get the highest accuracy.

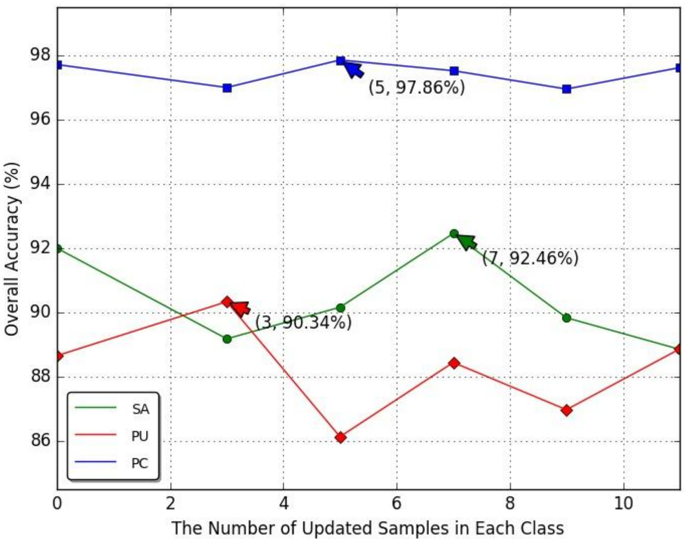

Fig. 6. OA (%) of MDCPE with different numbers of updated sample.

4) Number of updated samples in each class: In the analysis of co-training [45], it claims that the size of recovering data in each iteration should not affect the original data distribution. So in our experiment, the number of new labeled sample in each iteration is fixed and far less than the original number of labeled samples. MDCPE is adopted to select the unlabeled sample from each class and recover these samples with a predicted label. These updated data, with the initial labeled data together, are used to train the learners in next iteration. Fig. 6 presents performance of different number of updated samples in each class. These results are acquired in 5 iterations with MDCPE for each setting in three valid data sets. As Fig. 6 shows, In SA data set, 7 samples are selected from each class in every iteration of MDCPE co-training, 3 samples in PU data set and 5 samples in PC data set.

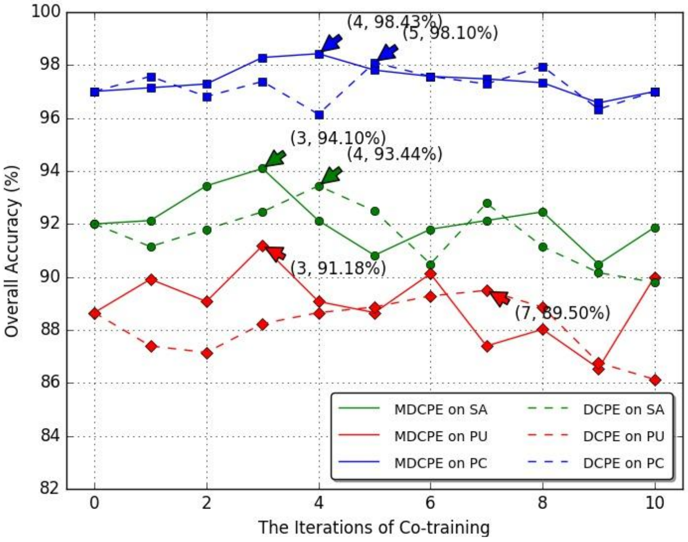

Fig. 7. OA (%) of MDCPE and DCPE with different iterations.

TABLE VI
CLASSIFICATION RESULTS OF DIFFERENT METHODS FOR THE SA DATA SET. THE BEST ACCURACY IN EACH ROW IS SHOWN IN BOLD

Class No.	**Class Name**	**RF**	**SVM**	**RNN**	**CNN**	**SSRN**	**DCPE-RNN-CNN**	**MDCPE-RNN-CNN**
1	Brocoli_green_weeds_1	97.88	**98.48**	96.61	92.93	1	97.93	97.32
2	Brocoli_green_weeds_2	99.68	93.15	98.3	70.89	**99.73**	87.99	91.51
3	Fallow	93.88	71.84	97.94	80.68	**99.69**	98.87	98.51
4	Fallow_rough_plow	96.11	99.71	99.71	98.39	43.36	**1**	**1**
5	Fallow_smooth	96.98	98.15	89.35	95.13	**1**	94.66	98.98
6	Stubble	99.24	99.39	96.74	94.6	86.84	98.22	**99.7**
7	Celery	97.24	99.27	97.52	89.74	**1**	97.6	97.44
8	Grapes_untrained	83.41	**97.78**	79.87	84.02	93.08	86.55	81.36
9	Soil_vinyard_develop	98.18	97.38	95.85	94.04	**99.97**	93.01	96.97
10	Corn_senesced_green_weeds	80.94	74.91	82.51	90.02	**98.34**	92.86	96.31
11	Lettuce_romaine_4wk	86.99	87.38	91.91	92.39	**98.11**	94.7	95.09
12	Lettuce_romaine_5wk	96.99	98.15	98.42	95.47	99.81	**99.89**	96.78
13	Lettuce_romaine_6wk	97.86	**98.08**	97.74	95.71	60.81	95.71	97.86
14	Lettuce_romaine_7wk	92.21	90.67	92.02	85.38	**1**	94.33	92.21
15	Vinyard_untrained	32.03	0	50.65	52.71	**92.36**	49.71	72.94
16	Vinyard_vertical_trellis	90.6	85.26	91.56	90.94	**1**	96.17	91.02
OA	**-**	84.35	81.49	85.67	83.71	**91.86**	87.76	90.21
AA	-	90.01	86.85	91.04	87.69	92.01	92.76	**94.01**
Kappa	-	0.8248	0.7913	0.8402	0.8181	**0.9093**	0.8633	0.891

TABLE VII
CLASSIFICATION RESULTS OF DIFFERENT METHODS FOR THE PU DATA SET. THE BEST ACCURACY IN EACH ROW IS SHOWN IN BOLD

Class No.	**Class Name**	**RF**	**SVM**	**RNN**	**CNN**	**SSRN**	**DCPE-RNN-CNN**	**MDCPE-RNN-CNN**
1	Asphalt	**86.99**	77.01	74.02	70.02	66.41	82.91	83.48
2	Meadows	**98.66**	90.64	90.27	91.72	98.34	94.24	96.35
3	Gravel	51.05	58.95	79.84	68.91	**97.66**	68.27	78.47
4	Trees	65.11	78.36	82.09	80.46	88.8	96.5	**96.87**
5	Painted metal sheets	**99.84**	99.53	99.3	99.14	86.47	**99.84**	**99.84**
6	Bare Soil	7.77	46.74	71.03	89.9	**95.52**	90.71	90.73
7	Bitumen	28.35	54.65	88.82	81.57	**99.07**	85.83	86.54
8	Self-Blocking Bricks	56.07	67.26	63.53	52.95	**91.24**	72.86	85.78
9	Shadows	99.44	**1**	99.89	90.76	82.28	99.1	**1**
OA	**-**	75.73	78.32	82.53	82.78	89.47	89.14	**91.84**
AA	-	65.92	74.79	83.2	80.6	89.53	87.81	**90.9**
Kappa	-	0.6541	0.709	0.7685	0.773	0.8585	0.8563	**0.8917**

C. *Classification Results*

1) Iterations: In the proposed MDCPE co-training, as the fixed number of samples are selected from unlabeled data to update the labeled data, the unlabeled data is decreasing with the increase of iterations. If there are not enough unlabeled data to replenish the labeled data, the semi-supervised learning tend to be overfitting [46]. So we set the maximum of iterations to control the training process and prevent overfitting. Co-training process will not stop until the iterations equal to the maximum of iteration. However, last iteration is not always the best result [35]. So, validation data is used to control the iterations and monitor the training process in order to end in highest accuracy. In each iteration, we calculate the accuracy of validation data based on trained learner and save the learner with highest accuracy for next training iteration. Accuracy of validation data in each iteration is shown in Fig. 7, in which the performance of DCPE and MDCPE are shown together. The final output learners, which have the highest accuracy among all iterations on validation data, are selected and applied on testing data. As shown inn Fig. 7, The highest accuracy of DCPE and MDCPE for SA data set occurs in 4th and 3rd iterations, 7th and 3rd iterations for PU data set, 5th and 4th iterations for PC data set, respectively.

2) Comparative Methods: To evaluate effectiveness of proposed method, we compared it with vector-based models: random forest (RF), support vector machine (SVM) and state-of-

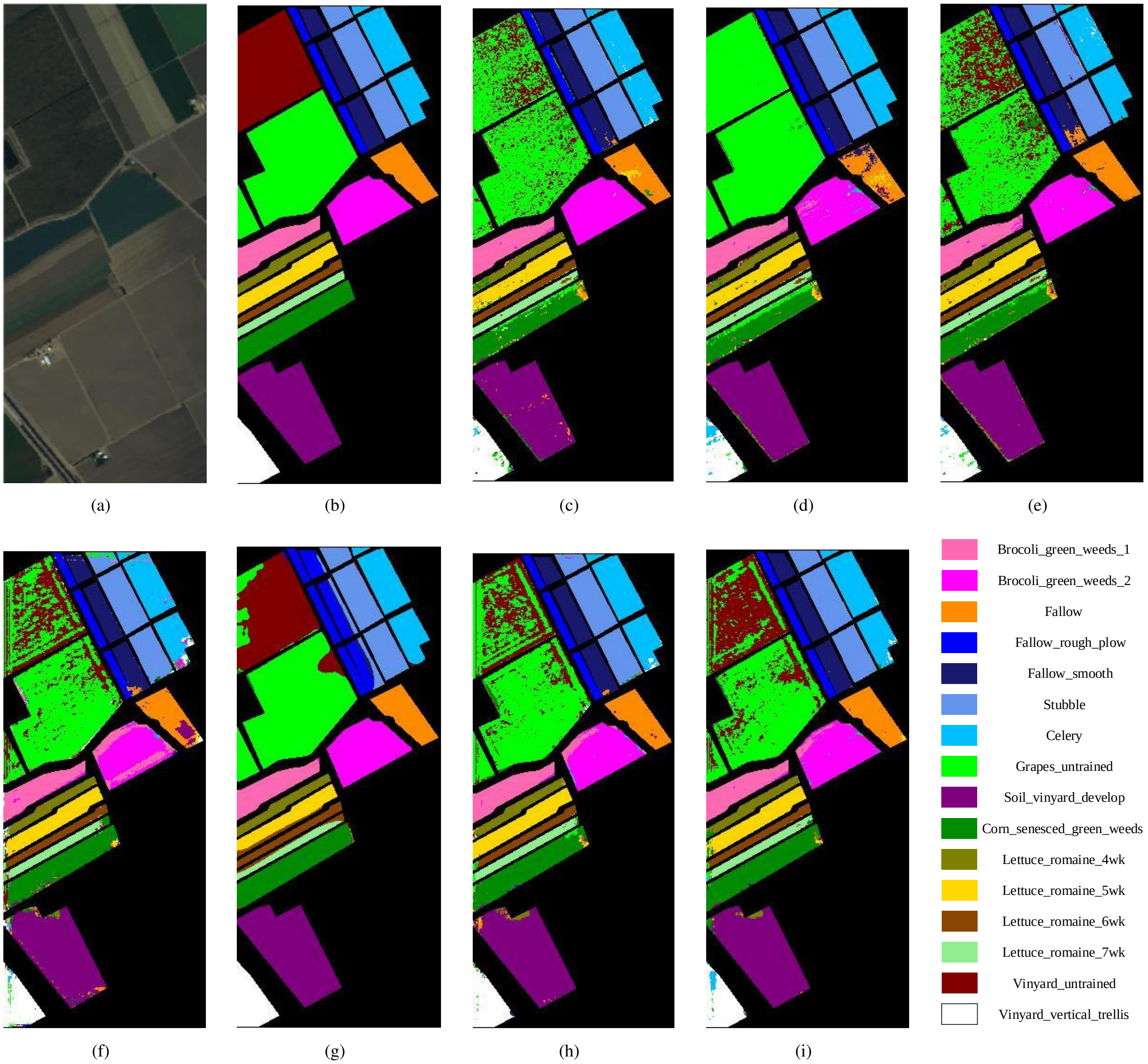

Fig. 8. Classification results obtained by different methods for the Salinas scene. (a) False color image of HS data. (b) Ground-truth labels. (c) RF. (d) SVM. (e) RNN. (f) CNN. (g) SSRN. (h) DCPE-RNN-CNN. i) MDCPE-RNN-CNN.

the-art deep learning models, such as RNN, CNN and SSRN. Moreover, in order to validate the superiority of MDCPE, we design DCPE-RNN-CNN, a semi-supervised classification model based on DCPE co-training of CNN and RNN learners. The contrastive methods are summarized as follows.

1) RF: Random forest with 150 trees.
2) SVM: SVM with RBF kernel.
3) RNN: GRU-based recurrent neural network, which owns similar structure and configuration with the RNN-based learner used in MDCPE. Learning rate and the number of time steps are optimized on validation data.
4) CNN: Three dimensional convolutional neural network, which shares same structure and convolutional kernels with the CNN-based learner adopted in our method. Learning rate and spatial size of input block are optimized to fulfill the highest classification accuracy.
5) SSRN: An end-to-end spectral-spatial residual network proposed in [22].
6) DCPE-RNN-CNN: RNN and CNN are taken as two learners of DCPE co-training. They share same structure with the learners in MDCPE.
7) MDCPE-RNN-CNN: The proposed co-training method based on MDCPE with RNN and CNN.

To make a fair comparison, we utilize the same training data set and testing data set for all used methods and tuned these contrasts to their optimal settings in SA, PU and PC

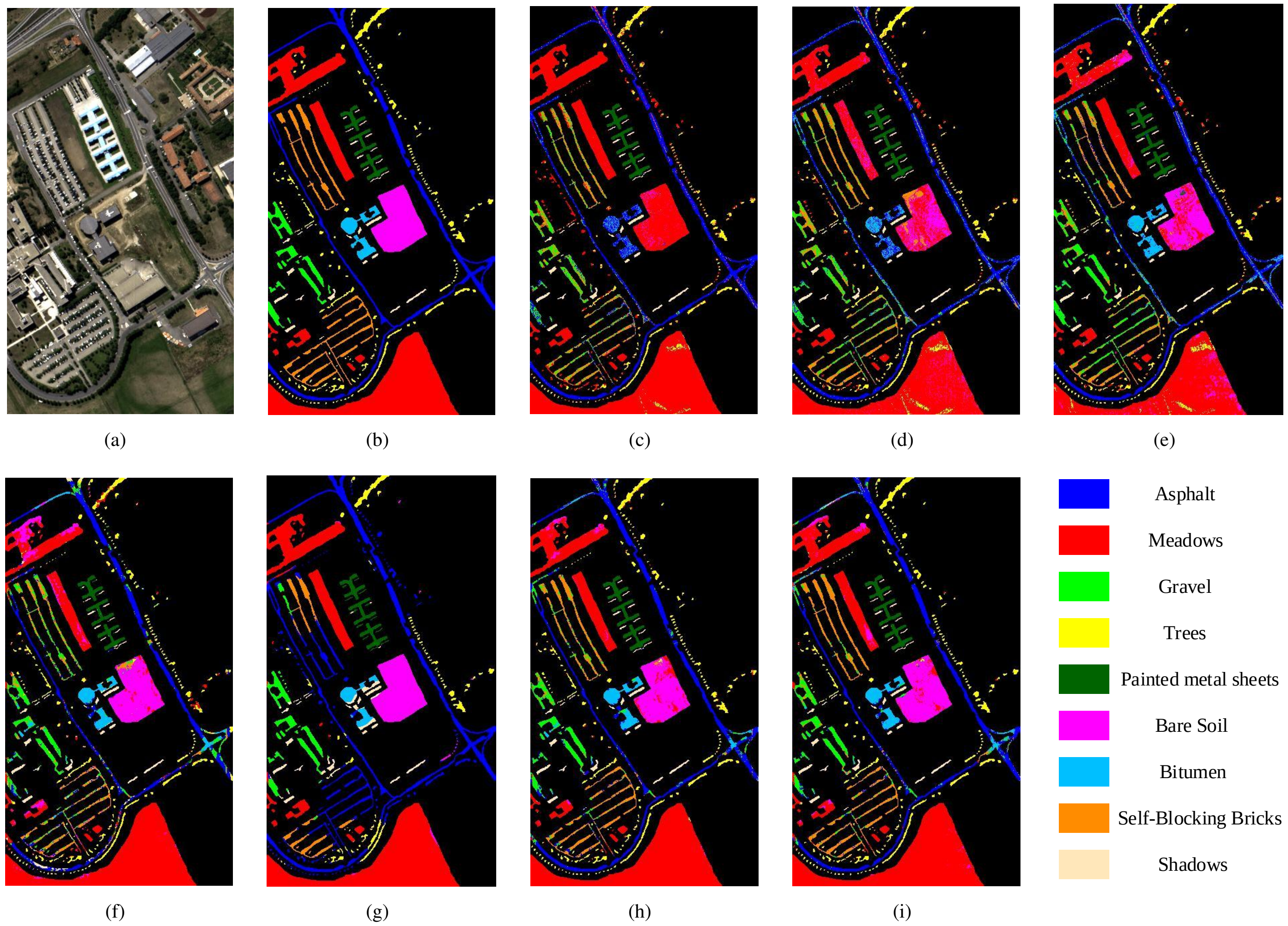

Fig. 9. Classification results obtained by different methods for the Pavia University scene. (a) False color image of HS data. (b) Ground-truth labels. (c) RF. (d) SVM. (e) RNN. (f) CNN. (g) SSRN. (h) DCPE-RNN-CNN. i) MDCPE-RNN-CNN.

data sets. Overall accuracy (OA), average accuracy (AA) and Kappa coefficient (K) are all calculated as classification indexes during analyzing results.

3) Results on SA Data Set: Table VI shows the classification indexes on SA data set, including OAs, AAs, Kappa coefficients, and the classification accuracy of each class. It's clear that SSRN and the proposed method have better performance than others. Traditional random tree (RF) and support vector machine (SVM) merely take advantage of the shallow spectral feature but ignoring abundant spatial feature, which leads to modest classification performance. However, SVM has the best performance on class 1, 8 and 13. RNN and CNN are utilized as learners to extract spectral or spatial feature separately. All of their performance are unfavorable because only spectral or spatial feature is insufficient for HSI classification. Comparing with DCPE-RNN-CNN, the prominent advantage of proposed method is that it enhances the accuracy of indistinguishable classes. For instance, table VI shows that the accuracy of MDCPE-RNN-CNN on class 15 is 72.94%, which is far higher than that of DCPE-RNN-CNN. After improving the performance on the classes that are hard to differentiate, our method increases the accuracy by 2.45% of OA, 1.25% of AA and 0.0277 of Kappa coefficient compared with DCPE-RNN-CNN. SSRN has the best accuracy on several classes and thus slightly breach the proposed method on OA and Kappa coefficient. However, MDCPE-RNN-CNN surpass a lot than SSRN on class 4 and 13, reflecting that the proposed method is more homogeneous and robust, not only achieves good performance in large area but also identifies small and indistinctive objects.

The classification maps of the best trained models on SA data set are visualized in Figs. 8. The false color images of original HSI data and their corresponding ground-truth maps are also shown along side with classification maps to offer direct comparison. Some classes that are hard to be correctly classified challenges the robustness and effectiveness of classifier. One possible reason is that these indistinguishable classes have similar spectral and spatial feature with other classes due to their complex and confusing ground-truth. In Fig. 8, All methods mistake the area of class Vinyard_untrained for class Grapes_untrained, as these two classes are mix up and hard to classify in true surfaces. Although further improvement is needed for SSRN and MDCPE-RNN-CNN in this class, more pixels are classified correctly and most

TABLE VIII
CLASSIFICATION RESULTS OF DIFFERENT METHODS FOR THE PC DATA SET. THE BEST ACCURACY IN EACH ROW IS SHOWN IN BOLD

Class No.	**Class Name**	**RF**	**SVM**	**RNN**	**CNN**	**SSRN**	**DCPE-RNN-CNN**	**MDCPE-RNN-CNN**
1	Water	99.83	99.65	99.99	99.74	99.89	1	**1**
2	Trees	89.32	88.21	87.28	93.02	88.76	**97.37**	95.62
3	Asphalt	87.68	82.11	77.88	49.3	**96.7**	47.58	80.13
4	Self-Blocking Bricks	49.75	49.45	75.53	63.06	**89.62**	71.05	84.97
5	Bitumen	85.24	92.14	87.04	79.83	92.79	91.24	**94.64**
6	Tiles	90.04	**95.8**	93.37	76.37	82.51	84.75	91.2
7	Shadows	80.68	84.98	87.02	70.96	**99.91**	94.11	91.61
8	Meadows	97.71	98.54	99.39	97.25	97.02	99.85	**99.89**
9	Bare Soil	99.92	96.24	99.96	69.64	**1**	97.38	97.28
OA	**-**	95.3	96.12	96.65	92.64	96.61	96.52	**97.81**
AA	-	86.7	87.46	89.72	77.69	**94.13**	87.04	92.81
Kappa	-	0.9338	0.945	0.9524	0.8947	0.9517	0.9506	**0.9688**

similar classification map with ground-truth labels is obtained by this two methods. Classification map of SSRN seems more homogeneous and smooth than other maps in most classes, but it has poor performance on class Fallow_rough_plow and Lettuce_romaine_6wk.

4) Results on PU Data Set: Quantitative classification results of PU data set are shown in Table VII. It's obvious that the proposed MDCPE-RNN-CNN, DCPE-RNN-CNN and SSRN surpass others. SSRN owns highest accuracy on class 3, 6, 7, and 8 but it has poor performance on class 1 and 9. Such unbalanced classification accuracies between classes impairs overall effectiveness of the method. DCPE-RNN-CNN and MDCPE-RNN-CNN have better performance than others because 1) these two co-training methods resort to two deep neural networks(RNN, CNN) as two learners to extract more discriminative feature from spectral and spatial views, 2) rich and valuable unlabeled data are adopted to enhance the generality and robustness of learners. DCPE-RNN-CNN has low precision on class 3 and 8 even it acquires highest accuracy on class 5. The proposed MDCPE-RNN-CNN manifests the highest accuracy and well-balanced accuracy between classes. Comparing to SSRN and DCPE-RNN-CNN, MDCPE-RNN-CNN increases the accuracy significantly by 2.37, 2.7% of OA, 1.37, 3.09% of AA and 0.0332, 0.0354 of Kappa coefficient, respectively.

Fig. 9 shows the classification maps of the PU data set qualitatively. The proposed method achieves the better performance on class Trees, Painted Metal Sheets and Shadows. These classes occupy either large areas of corridor or piecemeal mapped to ground-truths, for which they are easily perturbed by noise and other neighboring classes during the process of spectral imaging. Another reason is that the classifiers fail to learn differentiable feature of such classes due to few and unbalanced labeled training data. Although it's hard to learn the features from such classes and to classify them correctly, our method surpass other approaches in acquiring more homogeneous and well-bounded classification maps.

5) Results on PC Data Set: The classification maps of the PC data set obtained by the vector-based models, deep neural networks and our method are shown in Fig. 10, and the corresponding accuracy indexes are presented in Table VIII. Kappa is thought to be a more persuasive index than a simple percent agreement calculation, since it considers the agreement occurring by chance [46]. As table VIII shows, MDCPE-RNN-CNN generates highest Kappa and the best classification results. RF, SVM, and RNN have very low accuracy on class 4, and CNN, DCPE-RNN-CNN shows low-precision on class 3 and 4. Although SSRN acquires highest AA among other methods, its OA and Kappa coefficients are 1.2 percent and 0.0171 lower than the proposed method.

As Fig. 10 shows, classification maps of other methods present many noisy points and confuse class Self-Blocking Bricks with class Meadows and Bitumen. But the proposed method and SSRN acquire more smooth and homogeneous results on class Self-Blocking Bricks and Meadows. Moreover, Results on class Shadows of other methods are messy and show severe misclassification, but the proposed method display more accurate and uniform performance on this class. Taking accuracies of all classes into consideration, our method shows more robust results even though small amount of samples and unbalance between classes.

V. Conclusions

In this paper, a semi-supervised co-training method is proposed to extract spectral-spatial feature based on MDCPE of RNN and CNN for HSI classification. RNN and CNN are utilized as two learners of MDCPE co-training to extract spectral and spatial features, respectively. MDCPE is then adopted to co-train two learners based on extracted spectral-spatial feature, which is conducive to keep updated data balanced between classes and promotes the generality of learners with unlabeled data. Analyzing experiment results on three widely studied HSI cases, our method not only acquires higher and more balanced accuracy than other contrast methods on the quantitative, but also presents more homogeneous and refined

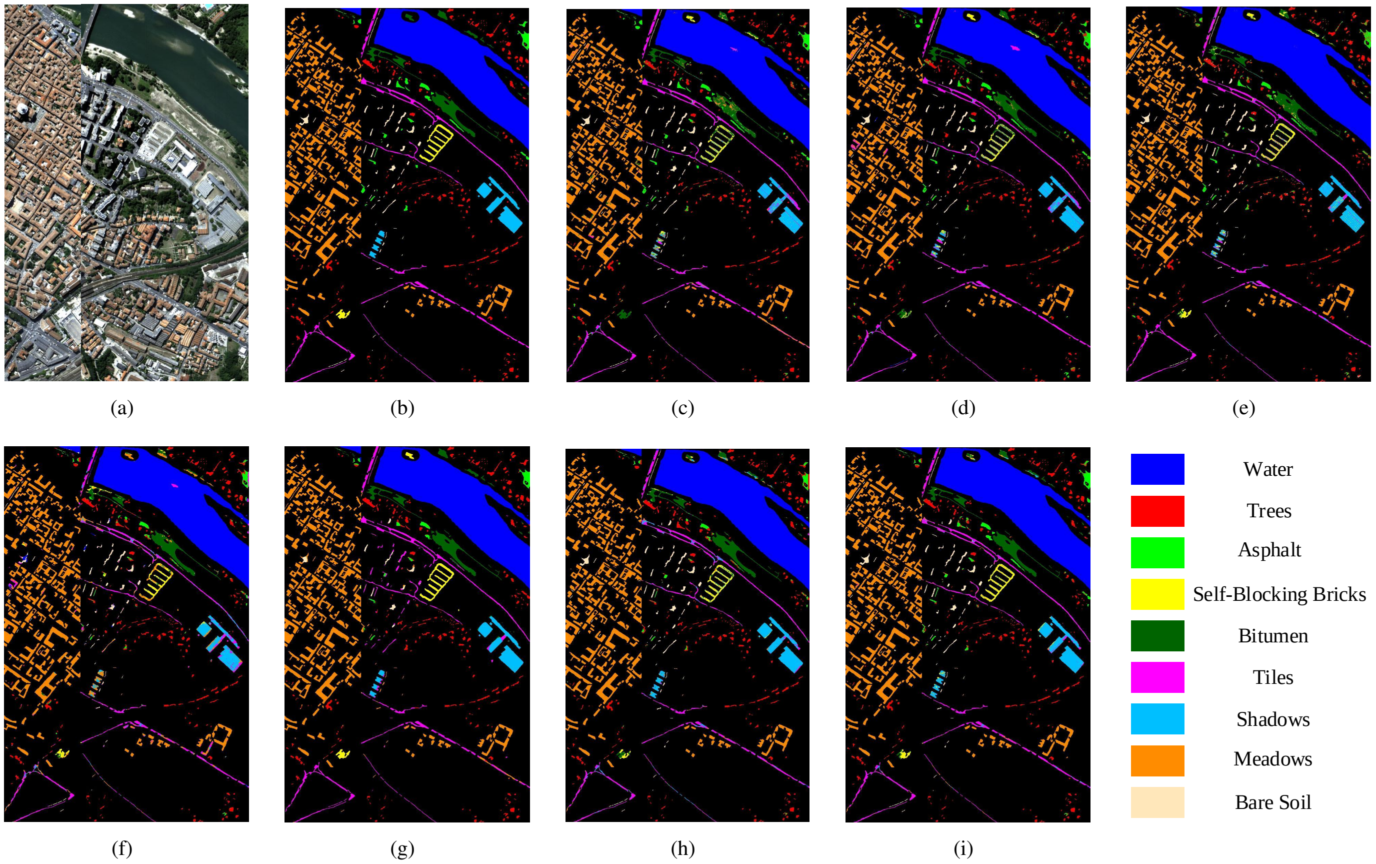

Fig. 10. Classification results obtained by different methods for the Pavia Center scene. (a) False color image of HS data. (b) Ground-truth labels. (c) RF. (d) SVM. (e) RNN. (f) CNN. (g) SSRN. (h) DCPE-RNN-CNN. i) MDCPE-RNN-CNN.

classification maps in visual. Moreover, experiments results prove that balanced training data and unlabeled data have great potential in improving the classification accuracy, which motivates us to make full use of unlabeled data and ensure the balance across classes in further study. Considering the favorable performance with limited and imbalanced labeled data, we believe that the proposed method can outperform other methods in more of HSI classification cases.

Even though the proposed MDCPE co-training surpasses comparative methods as shown in experiment results, further study is deserved to promote performance on the classes that are indiscriminating and can not be right classified in MDCPE. Besides, two neural networks are adopted as learners of MDCPE co-training to extract spectral and spatial features separately, which is considerably complicated and time-consuming. A intergraded learner is expected to come out, which can extract spectral and spatial features simultaneously and use unlabeled data to enhance the classification performance when labeled data is inadequate and imbalanced.

References

[1] Y. Gu, T. Liu, X. Jia, J. A. Benediktsson, and J. Chanussot, "Nonlinear multiple kernel learning with multiple-structure-element extended morphological profiles for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3235–3247, 2016.

[2] J. Li, M. Khodadadzadeh, A. Plaza, X. Jia, and J. M. Bioucas-Dias, "A discontinuity preserving relaxation scheme for spectral–spatial hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 2, pp. 625–639, 2016.

[3] C. Wu, B. Du, and L. Zhang, "Slow feature analysis for change detection in multispectral imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2858–2874, 2014.

[4] H. Lyu, H. Lu, and L. Mou, "Learning a transferable change rule from a recurrent neural network for land cover change detection," *Remote Sensing*, vol. 8, no. 6, p. 506, 2016.

[5] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sensing*, vol. 7, no. 11, pp. 14 680–14 707, 2015.

[6] L. G. Olmanson, P. L. Brezonik, and M. E. Bauer, "Airborne hyperspectral remote sensing to assess spatial distribution of water quality characteristics in large rivers: The mississippi river and its tributaries in minnesota," *Remote Sensing of Environment*, vol. 130, pp. 254–265, 2013.

[7] M. S. Moran, Y. Inoue, and E. Barnes, "Opportunities and limitations for image-based remote sensing in precision crop management," *Remote sensing of Environment*, vol. 61, no. 3, pp. 319–346, 1997.

[8] S. Delalieux, B. Somers, B. Haest, T. Spanhove, J. V. Borre, and C. Mücher, "Heathland conservation status mapping through integration of hyperspectral mixture analysis and decision tree classifiers," *Remote sensing of environment*, vol. 126, pp. 222–231, 2012.

[9] J. Ham, Y. Chen, M. M. Crawford, and J. Ghosh, "Investigation of the random forest framework for classification of hyperspectral data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 492–501, 2005.

[10] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Transactions on geoscience and remote sensing*, vol. 42, no. 8, pp. 1778–1790, 2004.

[11] J. Gualtieri and S. Chettri, "Support vector machines for classification of hyperspectral data," in *Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International*, vol. 2. IEEE, 2000, pp. 813–815.

[12] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE Journal of Selected topics in applied earth observations and remote sensing*, vol. 7, no. 6, pp. 2094–2107, 2014.

[13] A. Plaza, J. Plaza, and G. Martin, "Incorporation of spatial constraints into spectral mixture analysis of remotely sensed hyperspectral data," in *Machine Learning for Signal Processing, 2009. MLSP 2009. IEEE International Workshop on*. IEEE, 2009, pp. 1–6.

[14] J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geoscience and remote sensing magazine*, vol. 1, no. 2, pp. 6–36, 2013.

[15] M. Fauvel, J. A. Benediktsson, J. Chanussot, and J. R. Sveinsson, "Spectral and spatial classification of hyperspectral data using svms and morphological profiles," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 46, no. 11, pp. 3804–3814, 2008.

[16] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 2, pp. 844–856, 2013.

[17] J. Liu, Z. Wu, Z. Wei, L. Xiao, and L. Sun, "Spatial-spectral kernel sparse representation for hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 6, no. 6, pp. 2462–2471, 2013.

[18] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 7, pp. 3639–3655, 2017.

[19] Y. Chen, X. Zhao, and X. Jia, "Spectral–spatial classification of hyperspectral data based on deep belief network," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 6, pp. 2381–2392, 2015.

[20] W. Zhao and S. Du, "Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 8, pp. 4544–4554, 2016.

[21] Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[22] Z. Zhong, J. Li, Z. Luo, and M. Chapman, "Spectral–spatial residual network for hyperspectral image classification: A 3-d deep learning framework," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 56, no. 2, pp. 847–858, 2018.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the royal statistical society. Series B (methodological)*, pp. 1–38, 1977.

[24] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proceedings of the 33rd annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1995, pp. 189–196.

[25] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the eleventh annual conference on Computational learning theory*. ACM, 1998, pp. 92–100.

[26] T. D. Bie and N. Cristianini, "Convex methods for transduction," in *Advances in neural information processing systems*, 2004, pp. 73–80.

[27] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," 2001.

[28] I. Dópido, J. Li, P. R. Marpu, A. Plaza, J. M. B. Dias, and J. A. Benediktsson, "Semisupervised self-learning for hyperspectral image classification," *IEEE transactions on geoscience and remote sensing*, vol. 51, no. 7, pp. 4032–4044, 2013.

[29] X. Zhang, Q. Song, R. Liu, W. Wang, and L. Jiao, "Modified co-training with spectral and spatial views for semisupervised hyperspectral image classification," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 6, pp. 2044–2055, 2014.

[30] S. Samiappan and R. J. Moorhead, "Semi-supervised co-training and active learning framework for hyperspectral image classification," in *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*. IEEE, 2015, pp. 401–404.

[31] G. Camps-Valls, T. V. B. Marsheva, and D. Zhou, "Semi-supervised graph-based hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 10, pp. 3044–3054, 2007.

[32] F. Ratle, G. Camps-Valls, and J. Weston, "Semisupervised neural networks for efficient hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 48, no. 5, pp. 2271–2282, 2010.

[33] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[34] G. Camps-Valls and L. Bruzzone, "Kernel-based methods for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 6, pp. 1351–1362, 2005.

[35] J. Xu, H. He, and H. Man, "Dcpe co-training for classification," *Neurocomputing*, vol. 86, pp. 75–85, 2012.

[36] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural computation*, vol. 1, no. 2, pp. 270–280, 1989.

[37] P. Rodriguez, J. Wiles, and J. L. Elman, "A recurrent neural network that learns to count," *Connection Science*, vol. 11, no. 1, pp. 5–40, 1999.

[38] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent lstm neural networks for language modeling," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 3, pp. 517–529, 2015.

[39] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[40] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *International Conference on Machine Learning*, 2014, pp. 1764–1772.

[41] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[42] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[43] Y. Gal and Z. Ghahramani, "A theoretically grounded application of dropout in recurrent neural networks," in *Advances in neural information processing systems*, 2016, pp. 1019–1027.

[44] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[45] W. Wang and Z.-H. Zhou, "Analyzing co-training style algorithms," in *European Conference on Machine Learning*. Springer, 2007, pp. 454–465.

[46] D. Schuurmans and F. Southey, "Metric-based methods for adaptive model selection and regularization," *Machine Learning*, vol. 48, no. 1-3, pp. 51–84, 2002.