remote sensing

MDPI

Article

Weakly Supervised Classification of Hyperspectral Image Based on Complementary Learning

Lingbo Huang, Yushi Chen * and Xin He

School of Electronics and Information Engineering, Harbin Institute of Technology, Harbin 150001, China;
20s005093@stu.hit.edu.cn (L.H.); 19b905007@stu.hit.edu.cn (X.H.)
* Correspondence: chenyushi@hit.edu.cn

Abstract: In recent years, supervised learning-based methods have achieved excellent performance for hyperspectral image (HSI) classification. However, the collection of training samples with labels is not only costly but also time-consuming. This fact usually causes the existence of weak supervision, including incorrect supervision where mislabeled samples exist and incomplete supervision where unlabeled samples exist. Focusing on the inaccurate supervision and incomplete supervision, the weakly supervised classification of HSI is investigated in this paper. For inaccurate supervision, complementary learning (CL) is firstly introduced for HSI classification. Then, a new method, which is based on selective CL and convolutional neural network (SeCL-CNN), is proposed for classification with noisy labels. For incomplete supervision, a data augmentation-based method, which combines mixup and Pseudo-Label (Mix-PL) is proposed. And then, a classification method, which combines Mix-PL and CL (Mix-PL-CL), is designed aiming at better semi-supervised classification capacity of HSI. The proposed weakly supervised methods are evaluated on three widely-used hyperspectral datasets (i.e., Indian Pines, Houston, and Salinas datasets). The obtained results reveal that the proposed methods provide competitive results compared to the state-of-the-art methods. For inaccurate supervision, the proposed SeCL-CNN has outperformed the state-of-the-art method (i.e., SSDP-CNN) by 0.92%, 1.84%, and 1.75% in terms of OA on the three datasets, when the noise ratio is 30%. And for incomplete supervision, the proposed Mix-PL-CL has outperformed the state-of-the-art method (i.e., AROC-DP) by 1.03%, 0.70%, and 0.82% in terms of OA on the three datasets, with 25 training samples per class.

Keywords: complementary learning; convolutional neural network (CNN); deep learning; hyperspectral image classification; weakly supervised learning

check for updates

Citation: Huang, L.; Chen, Y.; He, X. Weakly Supervised Classification of Hyperspectral Image Based on Complementary Learning. *Remote Sens.* **2021**, *13*, 5009. https://doi.org/10.3390/rs13245009

Academic Editors: Amin Zehtabian, Saeid Homayouni, Pedram Ghamisi, Fardin Mirzapour and Ali Mousivand

Received: 3 November 2021
Accepted: 7 December 2021
Published: 9 December 2021

1. Introduction

Hyperspectral remote sensing obtains the spatial and spectral information from objects with hundreds of narrow spectral bands. The obtained hyperspectral image (HSI) contains abundant spectral and spatial information, therefore, HSI has a wide variety of applications such as agriculture [1], mineralogy [2], surveillance [3], physics [4], astronomy [5], chemical imaging [6], and environmental sciences [7].

In order to fully explore the usage of HSI, many data processing techniques have been proposed and classification is one of them [8]. The classification of HSI tries to assign a label to each pixel in the scene and it is the basic of many applications [9]. Most of existing HSI classification methods belong to supervised classification, where each training sample has a corresponding label indicating it's ground truth. It is a very active and hot topic, and a great many methods have been proposed [10–12].

In the early stage of HSI supervised classification, most of classifiers do not classify HSI in a deep manner. The typical HSI feature extraction and classification techniques include support vector machine (SVM), morphological operation, and sparse representation [13]. For example, support vector machine (SVM) exhibits its low sensitivity to high dimensionality, therefore, SVM-based methods have obtained good performance for HSI classification [14]. In order to extract the spatial features of HSI, many morphological operations including morphological profiles (MPs) [15] and extended multi-attribute profile (EMAP) [16] have been proposed for HSI classification. Another important technique is sparse representation, which generates a dictionary from inputs. And many sparse representation-based methods have been successfully explored for HSI classification [17,18]. In recent years, deep learning-based methods, especially convolutional neural networks (CNNs), have shown their powerfulness in many research fields including HSI classification [19–21]. Deep CNNs (DCNNs) hierarchically extract discriminate features of HSI and then obtain better classification performance compared with shallow models [22].

Although DCNN-based methods have achieved great progress in HSI classification, accurate classification is still challenging in real practice. For example, to proper train a large number of parameters in DCNNs, sufficient labeled samples are usually needed. However, the collecting of labeled training samples is expensive, daunting, and time-consuming. Therefore, the problem of learning with limited labeled samples should be solved in CNN-based methods [23]. Furthermore, there are incorrect labeled samples when labeling HSI training samples, which does great harm to classification performance [24]. However, traditional methods did not pay much attention to noisy labels in classification.

It is desirable to develop a new kind of classification mechanism which depends on less support and weakly supervised classification is a proper method. Weakly supervised learning covers a wide range of studies including incomplete supervision (i.e., only a subset of training samples is labeled), inexact supervision (i.e., only coarse-grained labeled), and inaccurate supervision (i.e., the given labels are not always right which are usually noisy labels) [25]. For the classification of HSI, there are usually two types of weakly supervised classification: semi-supervised HSI classification and HSI classification with noisy label.

Most of existing weakly supervised methods in HSI classification require discriminative features [26–28]. However, the handcrafted features limited the classification performance with weakly supervision. Therefore, we consider using deep CNN in the presence of weak supervision. Meanwhile, complementary learning (CL) strategy is proper to prevent CNN from being overfitting to weak supervision [29,30]. In CL, each training example is supplied with a complementary label. It is an indirect learning method for training CNN that “input image does not belong to this complementary label.” In this manner, the noisy-labeled samples can contribute to training CNN by providing the “right” information.

Due to the advantages of deep CNN-based methods, the property of CL and the necessary of weak supervision in real practice, weakly supervised deep learning based on CL is investigated in this study. Two kinds of weakly supervised classification, i.e., inaccurate supervision and incomplete supervision are addressed. The main contributions of this study are summarized as follows.

(1) Complementary learning is introduced for HSI classification for the first time. Compared to traditional supervised learning, complementary learning has the advantages of using less supervised information, which makes it proper for weakly supervised classification.
(2) An improved complementary learning strategy, which is based on selective CL (SeCL), is proposed for HSI classification with noisy labels. The SeCL uses CL to filter noisy-labeled samples out and uses selective CL to accelerate the training process.
(3) A method, i.e., Pseudo-Label, combined with mixup (Mix-PL), is proposed for semi-supervised HSI classification. The usage of Mix-PL makes the training process more stable and achieves better classification performance.
(4) SeCL is combined with Mix-PL (Mix-PL-CL) for further improving the performance of HSI semi-supervised classification, owing to the SeCL’s capacity for filtering noisy-labeled samples.

The rest of this paper is organized as follows. Section 2 presents the related works of this study. Sections 3 and 4 introduce the proposed inaccurate and incomplete supervision-based HSI classification methods, respectively. Section 5 presents comprehensive experiments including data description, results, and analysis. Finally, Section 6 summarizes the main conclusion of this study.

2. Related Works

2.1. DCNN-Based HSI Classification

In recent years, DCNN-based methods have achieved significant breakthroughs in HSI classification [31]. Compared to the traditional methods, DCNNs have been used to automatically learn high-level features from HSI in a hierarchical manner, which have achieved state-of-the-art performance. CNN-based methods for HSI classification can be roughly divided into two branches: modified CNN [32,33] and CNN combined with other machine learning techniques [34,35].

For the modified CNN methods for HSI classification, most works aim to modify the architecture of CNN for HSI classification. For example, the authors in [36] proposed a deep contextual CNN with residual learning and multi-scale convolution to explore the spatial-spectral features of HSI. In [37], CNN was used to extract the pixel-pair features for following HSI classification. In addition, due to the fact that the input of HSI should be a 3D cube, 3D convolution is used for HSI classification [38].

Many works have combined CNN with other machine learning techniques for HSI classification, such as transfer learning [39], ensemble learning [40], and few shot learning [41]. In addition, to fully extract the spatial features of HSI, morphological profiles were conducted on principal components and then followed by CNN to finish HSI classification task [42,43]. Very recently, Transformer has been investigated for HSI classification with CNN to extract spectral-spatial features [44]. However, the above approaches obtained superior performance heavily depend on enough and correctly labeled samples.

2.2. Weakly Supervised Learning-Based Classification

In weakly supervised learning, two types of weak supervision are often discussed, including inaccurate and incomplete supervision.

For inaccurate supervision, there are noisy-labeled training samples whose given labels don't indicate their ground-truth. Three major strategies dealing with label noise are widely explored: robust model architecture, robust loss, and sample selection. Noise adaptation layer is often used in robust model design to estimate the noise transition matrix [45]. In addition, designing robust losses is also a hot topic for learning with noisy labels. Ref. [46] combined mean absolute error and cross-entropy loss to design a noise-robust loss, which achieved good classification performance. Besides, sample selection is a promising way to cope with label noise. For example, Co-teaching [47] utilized two DNNs, each DNN selects a certain number of small-loss examples as clean samples and feeds them to another DNN for further training. And many works based on co-teaching were proposed for learning with noisy labels [48,49].

For incomplete supervision, there are not enough labeled training samples to train a good classifier. Semi-supervised learning is a major technique for solving this problem, which attempts to exploit unlabeled training samples to improve performance without human intervention [50]. Specifically, graph-based methods mainly focus on the construction of graph with different properties [51]. Ref. [52] introduced a new sparse graph construction method that integrates manifold constraints on the unknown sparse codes as a graph regularizer. Apart from graph-based methods, self-training is also a popular strategy. Ref. [53] proposed Pseudo-Label for semi-supervised learning, which used the labels predicted for unlabeled samples in the last epoch to train model. The authors in [54] utilized the features extracted by CNN to conduct label propagation algorithm, and got the pseudo labels for unlabeled samples.

2.3. Weakly Supervised Learning-Based HSI Classification

There are usually two types of weakly supervised HSI classification: HSI classification with noisy label and semi-supervised HSI classification.

For the HSI classification with noisy label approaches, many researchers are mainly focused on cleaning of mislabeled samples [55,56]. For example, the authors in [57] used spatial-spectral information extraction method to improve the separability of features, and then a target detection method was utilized to find noisy-labeled samples and correct their labels. Ref. [58] designed a noisy labels detection algorithm based on density peak algorithm. Training samples whose computed local densities below the threshold were removed from the training set. After cleaning, SVM would be trained on the less noisy training set. The above works used handcrafted features which would limit the classification performance, and it is an open question to construct a deep model robust to noisy labels.

A great many of methods have been proposed for HSI semi-supervised classification, including graph-based methods [59,60], Self-Organizing Maps [61] and self-training methods [62,63]. Several studies based on self-training are related to our work. For example, the authors in [64] utilized simple linear iterative cluster segmentation method to extract spatial information, and multiple classifiers were assembled to find the most confident pseudo-labeled samples. Of particular interest, [65] used the cluster results based on deep features and classification results based on the output of deep model, to determine whether to select the confident samples or not. The semi-supervised methods form a promising research direction in HSI classification with application-realistic assumption of limited availability of labeled samples.

3. CL-Based HSI Classification with Noisy Labels

CNN-based methods are quite powerful for classifying HSI if the labels are all correct. Unfortunately, the process of labeling training samples with no error is not only time-consuming but also sometimes impossible. If inaccurate labels are used in training stage, the classification performance will be severely degraded. In this section, complementary learning-based method is investigated for HSI classification with noisy labels.

3.1. CL-Based Deep CNN for HSI Classification

In supervised learning, each training sample contains an example (i.e., image) and its corresponding label. For example, if a classification model receives a 3D hyperspectral cube of a tree and a label "tree", the supervised classifier will be trained to acknowledge that the input cube is a tree.

For complementary learning, every training sample contains an image and a complementary label that the image does not belong to. For example, the model may receive a 3D hyperspectral cube of a tree and a label "not soil". Complementary label is relatively easy to obtain and it can be used for weakly supervised learning.

In a c-class classification task $f : \boldsymbol{X} \to \boldsymbol{Y}$, $\boldsymbol{x} \in \boldsymbol{X}$ and $y \in \boldsymbol{Y} = \{1, \cdots, c\}$ are the input image and the corresponding label of a training sample, respectively. The complementary label $\overline{y}$ of the sample can be obtained by:

$$\overline{y} = Radom\ selection\ from\ \{1, \cdots, c\} \backslash \{y\}. \tag{1}$$

In practice, $y, \overline{y} \in \{0, 1\}^c$ are the one-hot vector of the training sample. For traditional supervised deep learning-based classification, cross entropy is a widely-used loss function:

$$\mathcal{L}(f, \boldsymbol{x}, y) = -\textstyle\sum_{k=1}^{c} \boldsymbol{y}_k \log \boldsymbol{p}_k, \tag{2}$$

where $\boldsymbol{p}$ is a c-dimension vector outputted by CNN and $\boldsymbol{p}_k$ is the k-th element of $\boldsymbol{p}$, representing the probability that $\boldsymbol{x}$ belongs to class k. Cross entropy loss forces the output of model to meet the true distribution. It works well if the labels are all correct.

For CL-based learning, cross entropy loss is calculated as follows:

$$\mathcal{L}(f,x,\,\overline{y}) = -\sum_{k=1}^{c} \overline{y}_k \log(1-p_k). \tag{3}$$

Equation (3) enables the probability value of the given complementary label (i.e., $\overline{y}$) to be optimized towards zero, resulting in an increase in the probability values of other classes.

The framework of CL-based deep CNN for HSI classification is shown in Figure 1. In the figure, a neighboring region of the pixel to be classified is obtained as input. Then, a well-designed CNN is used for feature extraction and softmax is used to finish the HSI classification task. In the training procedure, the complementary labels for training samples are firstly obtained by Equation (1), and then CL-based loss, i.e., the loss in Equation (3), is used to train the parameters in CNN based on back-propagation.

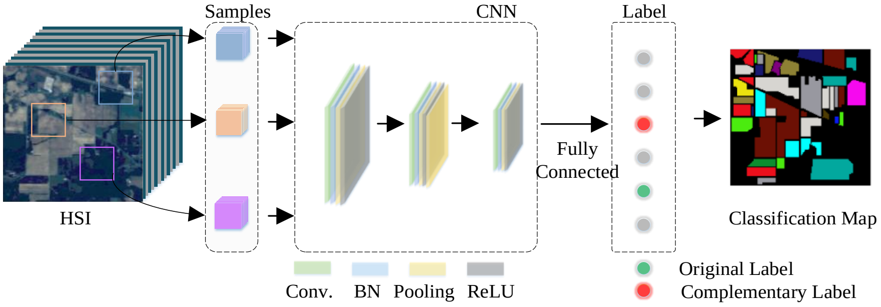


Figure 1. Complementary learning-based CNN for HSI classification.

3.2. CL-Based HSI Classification with Noisy Labels

Complementary learning can reduce the probability of wrong labeled training samples and therefore it can prevent the deep learning methods from overfitting to noisy data, which is useful for a supervised classification task with noisy labels.

Figure 2 demonstrates the proposed CL-based HSI classification method with noisy labels (SeCL-CNN). Due to the powerfulness and good performance, CNN is used as the basis of the classification system.

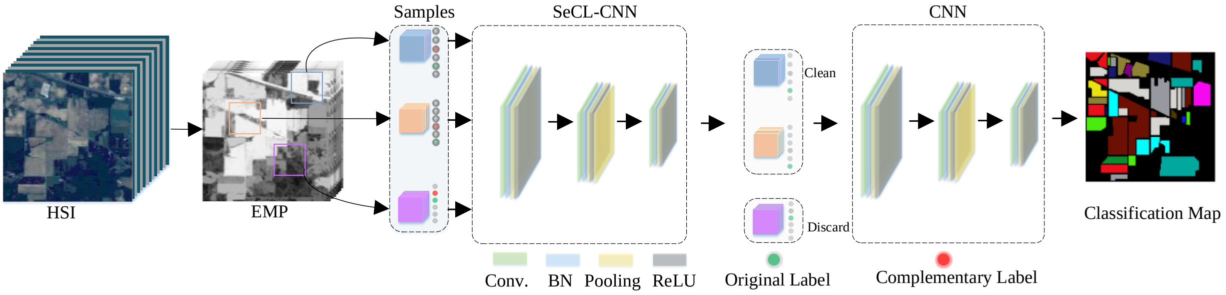


Figure 2. The framework of the complementary learning-based HSI classification with noisy labels.

In order to reduce the computational complexity of HSI classification, extended morphological profile (EMP) [15] is used as a pre-processing step of CL-CNN-based classification.

In general, there are two stages in the whole method: detection stage using SeCL, and classification stage using CNN. In the detection stage, the proposed SeCL firstly uses the CL strategy to train a CNN by minimizing Equation (3). Then, the CNN is trained using selective CL strategy, which only select the samples whose p_y are larger than $1/c$, for faster and better convergence. In the classification stage, the training samples whose p_y are larger than 0.5 is selected, and then they are treated as clean samples to train a classifier (i.e., CNN) using Equation (2).

In a nutshell, the overall flowchart of the proposed SeCL-CNN is shown in Algorithm 1. Steps 4 and 5 mean the procedure of complementary learning and selective complementary learning, respectively. Then step 6 selects clean-label samples from training set. And steps 7–9 use selected samples to train the CNN model for final classification.

Algorithm 1 SeCL-CNN for HSI classification with noisy labels

```
1. Begin
2.  Input: Noisy training samples $(x, y) \in (\boldsymbol{X}, \boldsymbol{Y})$, where $x$ is a 3D cube from EMPs
           of HSI and $y$ is the corresponding label
3.  Initialize network $f$
4.  For $t$ = 1 to $T_1$ do:
           Batch $(\boldsymbol{X}_B, \boldsymbol{Y}_B)$ = sample $(x, y)$ from $(\boldsymbol{X}, \boldsymbol{Y})$
           For each $x \in \boldsymbol{X}_B$ do:
               Get complementary label $\overline{y}$ using Equation (1)
               Calculate $\mathcal{L}(f, x, \overline{y})$ by Equation (3)
           Update $f$ by minimizing $\sum_{x \in \boldsymbol{X}_B} \mathcal{L}(f, x, \overline{y})$
5.  For $t$ = 1 to $T_2$ do:
           Batch $(\boldsymbol{X}_B, \boldsymbol{Y}_B)$ = sample $(x, y)$ from $(\boldsymbol{X}, \boldsymbol{Y})$, if $\boldsymbol{p}_y > 1/c$
           For each $x \in \boldsymbol{X}_B$ do:
               Get complementary label $\overline{y}$ using Equation (1)
               Calculate $\mathcal{L}(f, x, \overline{y})$ by Equation (3)
           Update $f$ by minimizing $\sum_{x \in \boldsymbol{X}_B} \mathcal{L}(f, x, \overline{y})$
6.  $(\boldsymbol{X}_{clean}, \boldsymbol{Y}_{clean})$ = sample $(x, y)$ from $(\boldsymbol{X}, \boldsymbol{Y})$, if $\boldsymbol{p}_y > 0.5$
7.  Initialize network $f$
8.  For $t$ = 1 to $T_3$ do:
           Batch $(\boldsymbol{X}_B, \boldsymbol{Y}_B)$ = sample $(x, y)$ from $(\boldsymbol{X}_{clean}, \boldsymbol{Y}_{clean})$
           For each $x \in \boldsymbol{X}_B$ do:
               Calculate $\mathcal{L}(f, x, y)$ by Equation (2)
           Update $f$ by minimizing $\sum_{x \in \boldsymbol{X}_B} \mathcal{L}(f, x, y)$
9.  Output: network $f$ and filtered dataset $(\boldsymbol{X}_{clean}, \boldsymbol{Y}_{clean})$
10. End
```

4. CL-Based Semi-Supervised HSI Classification

The collection of labeled training samples is not only costly but also time-consuming. In addition, there are tremendous unlabeled samples. How to effectively utilize both the labeled and unlabeled samples is an urgent task in HSI classification. In this section, a semi-supervised HSI classification method, which combines complementary learning, Pseudo-Label, and mixup, is proposed for the task.

Incomplete supervised HSI classification concerns the situation with a small amount of labeled data, which is insufficient to train a classifier well, and a large number of unlabeled data. For incomplete supervision, the task is to learn $f = \boldsymbol{X} \rightarrow \boldsymbol{Y}$ from labeled and unlabeled training set. The labeled training dataset $\boldsymbol{D}_l$ and unlabeled $\boldsymbol{D}_u$ can be denoted as:

$$\boldsymbol{D}_l = \{(x_1, y_1), (x_2, y_2), \cdots, (x_l, y_l), \cdots, (x_m, y_m)\}, \tag{4}$$

$$\boldsymbol{D}_u = \{x_{m+1}, x_{m+2}, \cdots, x_u, \cdots, x_{m+n}\}. \tag{5}$$

There are m samples with cubes x_l and their corresponding labels y_l $(l = 1 : m)$ in labeled training dataset. Furthermore, $\boldsymbol{D}_u$ has n unlabeled training samples x_u $(u = m + 1 : m + n)$.

4.1. Pseudo-Label for HSI Semi-Supervised Classification

Pseudo-Label (PL) is a simple but efficient method which can exploit both labeled and unlabeled samples. It just picks up the class which has the maximum predicted probability, and uses them as if they were true labels.

In PL, a CNN g is trained in a supervised fashion with labeled and unlabeled data simultaneously. For an unlabeled samples $\boldsymbol{x}_u$ in the current training epoch, its pseudo label $\hat{y}_u$ has been obtained by

$$\hat{y}_u = g(\boldsymbol{x}_u), \tag{6}$$

in the last epoch. And then $\hat{y}_u$ is used to calculate the cross entropy loss for unlabeled samples.

The overall loss function is

$$\mathcal{L}_{total} = \mathcal{L}_s + \rho(t) * \mathcal{L}_u, \tag{7}$$

$$\mathcal{L}_s = \frac{1}{B_1} \sum\nolimits_{l \in B_1} \mathcal{L}(g, \boldsymbol{x}_l, y_l), \tag{8}$$

$$\mathcal{L}_u = \frac{1}{B_2} \sum\nolimits_{u \in B_2} \mathcal{L}(g, \boldsymbol{x}_u,\ \hat{y}_u), \tag{9}$$

where $\mathcal{L}_s$ and $\mathcal{L}_u$ are supervised loss generated by labeled samples and unsupervised loss generated by unlabeled samples, respectively. And $\rho(t)$ is a balancing coefficient, varying with epoch represented by t, to weight the importance of unsupervised loss. B_1 and B_2 are batch-size for each kind of loss. $\mathcal{L}(\cdot)$ is the cross-entropy loss defined by Equation (2).

4.2. Combining Mixup and Pseudo-Label for HSI Semi-Supervised Classification

As is introduced in Section 4.1, PL trains a CNN by using pseudo labels as if they were true labels. In order to alleviate the negative impact caused by incorrect pseudo labels and regularize the model for better convergence, PL combined with mixup [66], abbreviated as Mix-PL, is proposed for HSI semi-supervised classification.

Given a mixup operation:

$$\begin{cases} \widetilde{\boldsymbol{x}} = \lambda \boldsymbol{x}' + (1-\lambda)\boldsymbol{x}'' \\ \widetilde{\boldsymbol{y}} = \lambda \boldsymbol{y}' + (1-\lambda)\boldsymbol{y}'' \end{cases}, \tag{10}$$

where $(\boldsymbol{x}', \boldsymbol{y}')$ and $(\boldsymbol{x}'', \boldsymbol{y}'')$ are randomly selected from training set while using corresponding one-hot label vector. The decision boundary is pushed by enforcing the prediction model to behave linearly in-between training examples. The parameter $\lambda \sim \mathrm{Beta}(\alpha,\ \alpha)$, with $\alpha \in (0, \infty)$. $\mathrm{Beta}(\alpha,\ \alpha)$ represents the Beta distribution in probability theory. And the hyper parameter α controls the strength of interpolation for mixup.

From Equation (10), it can be seen that labels are needed for mixup. Here we extend mixup to the semi-supervised learning setting by using the pseudo label for unlabeled samples:

$$\begin{cases} \widetilde{\boldsymbol{x}}_u = \lambda \boldsymbol{x}_{u_1} + (1-\lambda)\boldsymbol{x}_{u_2} \\ \widetilde{\boldsymbol{y}}_u = \lambda \hat{\boldsymbol{y}}_{u_1} + (1-\lambda)\hat{\boldsymbol{y}}_{u_2} \end{cases}, \tag{11}$$

where $\boldsymbol{x}_{u_1}, \boldsymbol{x}_{u_2}$ are sampled from unlabeled dataset, and $\hat{\boldsymbol{y}}_{u_1}, \hat{\boldsymbol{y}}_{u_2}$ are corresponding one-hot pseudo labels which are generated by Equation (6).

The unsupervised loss can be calculated by:

$$\mathcal{L}_{mu} = \frac{1}{B_2} \sum\nolimits_{u \in B_2} \mathcal{L}(g, \widetilde{\boldsymbol{x}}_u,\ \hat{\boldsymbol{y}}_u), \tag{12}$$

where $\mathcal{L}(\cdot)$ is the cross entropy loss and $\widetilde{\boldsymbol{x}}_u,\ \hat{\boldsymbol{y}}_u$ are generated by Equation (11). $\mathcal{L}_{total}$ is revised as:

$$\mathcal{L}_{total} = \mathcal{L}_s + \rho(t) * \mathcal{L}_{mu}. \tag{13}$$

4.3. Combining CL and Mix-PL for HSI Semi-Supervised Classification

Considering the excellent performance in the presence of label noise, we further combine Mix-PL with SeCL to filter out some incorrect labels and propose Mix-PL-CL method based on self-training.

Figure 3 illustrates the proposed Mix-PL-CL for semi-supervised HSI classification. Specifically, we firstly train a CNN denoted by g using Mix-PL. And then the classifier is used to make predictions on abundant unlabeled samples. This process can be described by:

$$\hat{y}_u = g(\boldsymbol{x}_u),\ u = m+1 : m+n, \tag{14}$$

$$\boldsymbol{D}_{noisy} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_{m+n}, \hat{y}_{m+n})\}. \tag{15}$$

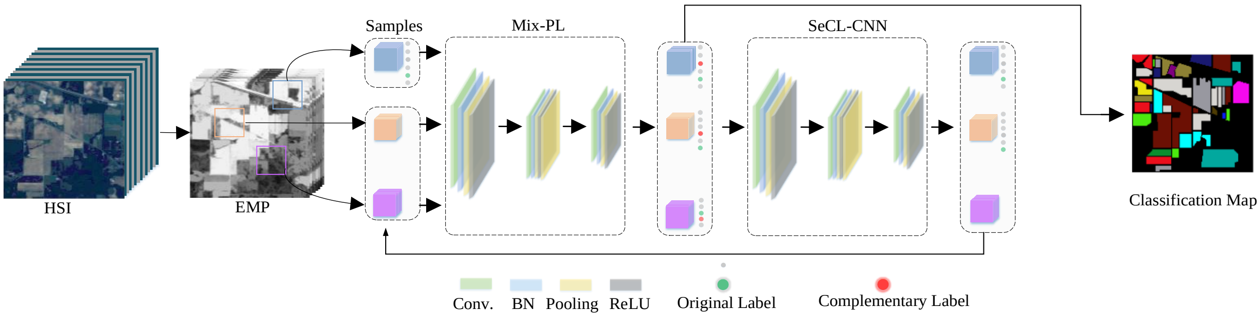


Figure 3. The framework of the complementary learning-based HSI semi-supervised classification.

The predicted pseudo-labels are not absolutely correct, so $\boldsymbol{D}_{noisy}$ is used here to denote the labeled and pseudo-labeled samples. We would like to select pseudo-labeled samples that are most likely to be correct, treat them as truly labeled, and add them to the labeled training set. This can be accomplished by SeCL-CNN, which has been introduced in previous section:

$$\boldsymbol{D}_{clean} = select\left(\boldsymbol{D}_{noisy}\right), \tag{16}$$

where $select(\cdot)$ means using SeCL to choose less noisy samples.

Iterating this procedure is an alternative way of improving the quality of pseudo-labels and finally obtain better classification performance.

Algorithm 2 shows the overall process of the proposed semi-supervised classification method.

Algorithm 2 Mix-PL-CL for HSI semi-supervised classification

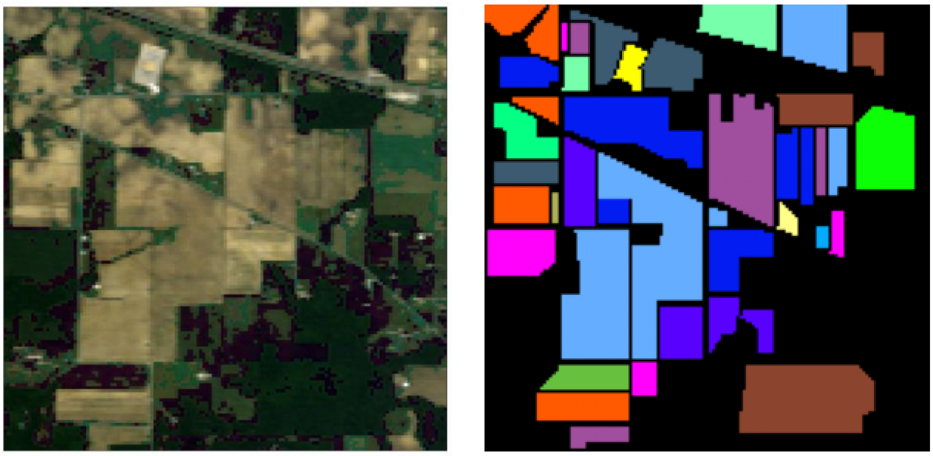
```
1. Begin
2.   Input: labeled training set $\boldsymbol{D}_l$, unlabeled training set $\boldsymbol{D}_u$
3.   Initialize network $g$
4.   For $i$ = 1 to $T_4$. do:
5.     For $t$ = 1 to $T_5$ do:
            For each $\boldsymbol{x}_u \in \boldsymbol{D}_u$ do:
                $\hat{y}_u = g(\boldsymbol{x}_u)$
            $\hat{\boldsymbol{D}}_u = \{(\boldsymbol{x}_u, \hat{y}_u)\}_{u=m+1}^{m+n}$
            Sample $\{(\boldsymbol{x}_l, y_l)\}_{l=1}^{B_1}$ from $\boldsymbol{D}_l$
            Calculate supervised loss $\mathcal{L}_s$ by Equation (8)
            Sample $\{(\boldsymbol{x}_{u_1}, \hat{y}_{u_1})\}_{u_1=1}^{B_2}$ from $\hat{\boldsymbol{D}}_u$
            $\{(\boldsymbol{x}_{u_2}, \hat{y}_{u_2})\}_{u_2=1}^{B_2}$ = permutation $(\{(\boldsymbol{x}_{u_1}, \hat{y}_{u_1})\}_{u_1=1}^{B_2})$
            Get $\{(\tilde{\boldsymbol{x}}_u, \tilde{y}_u)\}_{u=1}^{B_2}$ by Equation (11)
            Calculate unsupervised loss $\mathcal{L}_{mu}$ by Equation (12)
            Update $g$ by minimizing Equation (13)
6.     For each $\boldsymbol{x}_u \in \boldsymbol{D}_u$ do:
            $\hat{y}_u = g(\boldsymbol{x}_u)$
7.     $\boldsymbol{D}_{noisy} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_{m+n}, \hat{y}_{m+n})\}$
8.     $\boldsymbol{D}_l = \boldsymbol{D}_{clean} = select\left(\boldsymbol{D}_{noisy}\right), \boldsymbol{D}_u = \boldsymbol{D}_{noisy} - \boldsymbol{D}_l$
9.   Output: network $g$
10. End
```

5. Results

5.1. Datasets Description

To evaluate the performance of the proposed methods, three widely-used hyperspectral datasets, including Indian Pines, Houston, Salinas Valley, were employed in the experiments, which are described as follows.

(1) Indian Pines: This dataset was acquired by the Airborne Visible/Infrared Imaging Spectrometer sensor in June 1992, covering the agricultural fields in Indian, USA. The scene consists of 145 × 145 pixels with a spatial resolution of 20 m × 20 m and has 220 bands covering the range from 400 nm to 2500 nm. In this paper, 20 low signal to noise ratio (SNR) bands were removed and a total of 200 bands were reserved for classification. Figure 4 illustrated the false color composite images and corresponding ground-truth map of the Indian Pines dataset. The numbers of samples for each class were listed in Table 1.



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

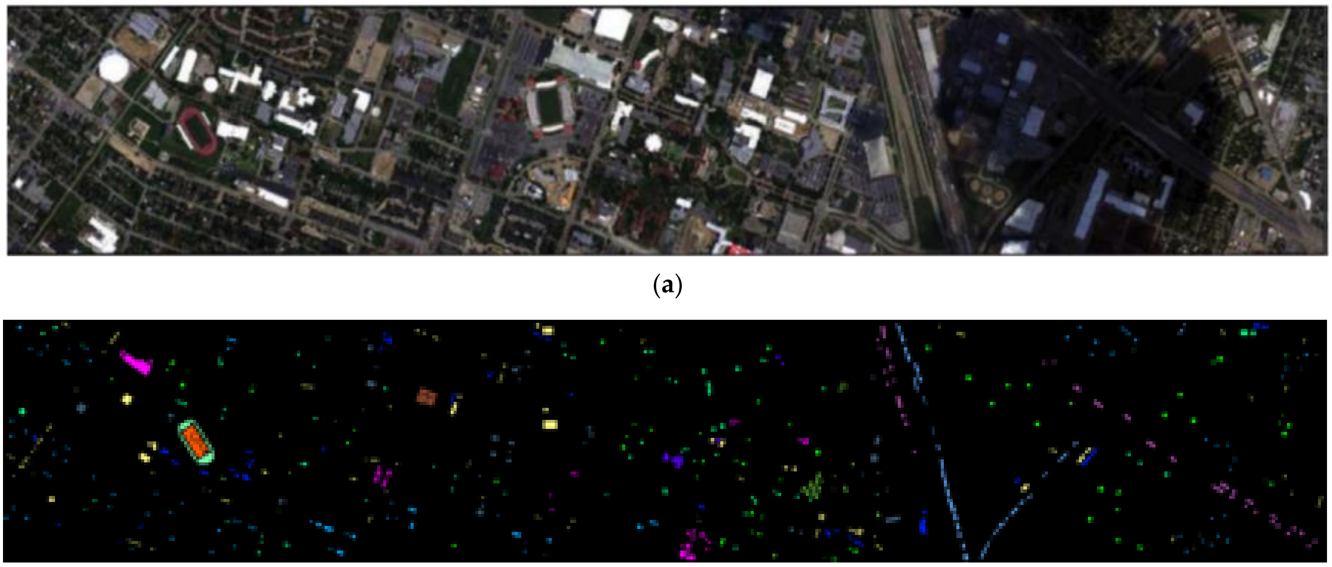
Figure 4. Indian Pines dataset: (**a**) false color map; (**b**) ground-truth map.

Table 1. Land cover classes and numbers of samples in the Indian Pines dataset.

No.	Color	Class Name	Number
1	🟨	Alfalfa	46
2	🟦	Corn-notill	1428
3	🟧	Corn-mintill	830
4	🟩	Corn	237
5	🟪	Grass-pasture	483
6	🟪	Grass-trees	730
7	🟦	Grass-pasture-mowed	28
8	🟩	Hay-windrowed	478
9	🟩	Oats	20
10	🟪	Soybean-notill	972
11	🟦	Soybean-mintill	2455
12	⬛	Soybean-clean	593
13	🟩	Wheat	205
14	🟫	Woods	1265
15	🟩	Buildings-Grass-Trees	386
16	🟨	Stone-Steel-Towers	93
	Total		10,249

(2) Houston: The Houston dataset was acquired over the Houston University campus and its neighboring area, by an ITRES-CASI 1500 sensor. It had been used in the 2013 GRSS Data Fusion Contest. The dataset contains 144 spectral bands ranging from 380 nm to 1050

nm region, and 349 × 1905 pixels with a spatial resolution of 2.5 m. This dataset is an urban dataset whose most of the land covers are man-made objects. It contains fifteen classes. Figure 5 illustrated the false color composite images and corresponding ground-truth maps. The numbers of samples for each class were listed in Table 2.


(**a**)

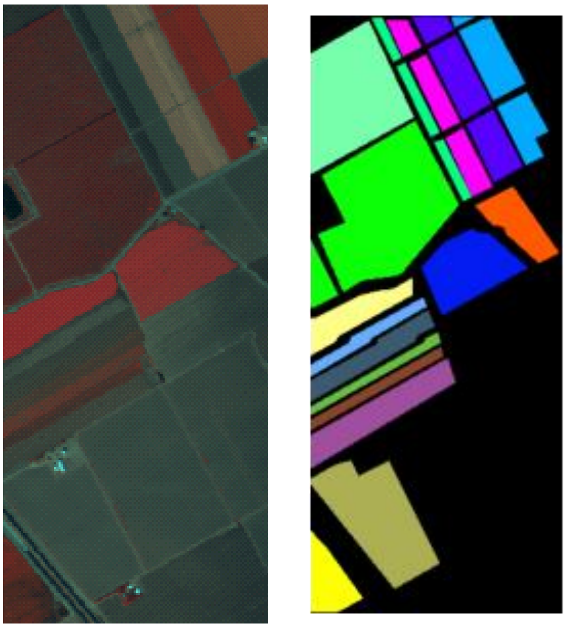
(**b**)

Figure 5. Houston dataset: (**a**) false color map; (**b**) ground-truth map.

Table 2. Land cover classes and numbers of samples in the Houston dataset.

No.	Color	Class Name	Number
1	🟨	Grass-healthy	1251
2	🟦	Grass-stressed	1254
3	🟧	Grass-synthetic	697
4	🟩	Tree	1244
5	🟪	Soil	1242
6	🟪	Water	325
7	🟦	Residential	1268
8	🟩	Commercial	1244
9	🟨	Road	1252
10	🟪	Highway	1227
11	🟦	Railway	1235
12	⬛	Parking-lot-1	1233
13	🟩	Parking-lot-2	469
14	🟫	Tennis-court	428
15	🟩	Running-track	660
	Total		15,029

(3) Salinas: This dataset was acquired by the 224-band AVIRIS sensor, capturing an area over Salinas Valley, CA, USA. The dataset consists of 204 spectral channels after the removal of 20 water absorption bands (108–112, 154–167, and 224), ranging from 400 to 2500 nm. 512 × 217 pixels are included with a spatial resolution of 3.7 m. In this dataset, there are approximately 54,129 labeled pixels with 16 classes sampled from the ground-truth map. Figure 6 demonstrated the false color composite images and corresponding ground-truth maps. The numbers of samples for each class were listed in Table 3.



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

Figure 6. Salinas dataset: (**a**) false color map; (**b**) ground-truth map.

Table 3. Land cover classes and numbers of samples in the Salinas dataset.

No.	Color	Class Name	Number
1	🟨	Brocoli-green-weeds-1	2009
2	🟦	Brocoli-green-weeds-2	3726
3	🟧	Fallow	1976
4	🟩	Fallow-rough-plow	1394
5	🟪	Fallow-smooth	2678
6	🟪	Stubble	3959
7	🟦	Celery	3579
8	🟩	Grapes-untrained	11,271
9	🟫	Soil-vineyard-develop	6203
10	🟪	Corn-senesced-green-weeds	3278
11	🟦	Lettuce-romaine-4wk	1068
12	🟦	Lettuce-romaine-5wk	1927
13	🟩	Lettuce-romaine-6wk	916
14	🟫	Lettuce-romaine-7wk	1070
15	🟩	Vineyard-untrained	7268
16	🟨	Vineyard-vertical-trellis	1807
		Total	54,129

5.2. Experimental Setup

For the three datasets, the samples were divided into two subsets which contained the training and testing samples, respectively.

(1) Experimental Setup for Classification with Noisy Labels: In the training process with noisy labels, 30 samples were chosen randomly for each class and only 15 labeled samples were chosen if the corresponding class had less than 30 samples.

For each training sample $\boldsymbol{x}_i$, the potential noisy label $\widetilde{y}_i$ could be generated as follows:

$$p(\widetilde{y}_i = k | y_i = j, \boldsymbol{x}_i) = p(\widetilde{y}_i = k | y_i = j) = \eta_{jk} \,, \tag{17}$$

$$\eta_{jk} = \begin{cases} 1-\eta, & j = k \\ \frac{\eta}{C-1}, & j \neq k \end{cases}, \tag{18}$$

where y_i represented the correct label, whose value was j, and it had the probability η_{jk} to become the noisy label k. From Equation (17), one could see that the noise added in labels was independent of individual samples. And Equation (18) showed that the probability of label transition from one class to the other was constant. This type of label noise was called symmetric noise. Following most related works, we used symmetric label noise in the experiments.

In the experiments, the general noise ratio η was set to be 0.1, 0.2, and 0.3 for exploring the performance of different leraning algorithms.

As a noisy label detection method, the proposed SeCL-CNN was compared with density peak based mehtods including DP, KSDP [58], and SSDP [56]. And a method with noise-robust loss function, denoted as Lq-CNN, was used for comparison [46]. Besides, traditional classification methods were also conducted, such as SVM, EMP-SVM, CNN, and MCNN-CP [67]. Among these methods, SVM, EMP-SVM, CNN, MCNN-CP, and CNN-Lq were end-to-end classification methods, while DP, KSDP, SSDP, and SeCL-CNN would filter out noisy samples firstly, and then use the rest samples to train CNNs for classification.

In SVM-based methods, we adopted grid search together with five-fold cross validation to find the proper C and γ $\left(C = 10^{-4}, 10^{-3}, \ldots, 10^{3}, \gamma = 10^{-4}, 10^{-3}, \ldots, 10^{3}\right)$. When using EMP, the first four principle components (PCs) were used. For each PC, three openings and closings by reconstruction were conducted with a circular structuring element whose initial size was four and step size increment was two.

The architecture of the CNN used in the experiments was shown in Table 4. It contained three convolutional layers with rectified linear unit (ReLU), three batch normalization layers, and two pooling layers. In order to use spatial information, the 27 × 27 image regions corresponding to a center pixel were fed to the 2D CNN.

Table 4. Architecture of CNN.

No.	Convolution	ReLU	Pooling	Padding	Stride	BN
1	4 × 4 × 32	YES	2 × 2	NO	1	YES
2	5 × 5 × 32	YES	2 × 2	NO	1	YES
3	4 × 4 × 64	YES	NO	NO	1	YES

For CNN and Lq-CNN, the initial learning rates were set to 0.01, and it was divided by 10 every 50 epochs. The number of epochs for training was set to 150.

The initial learning rate for SeCL-CNN was set to 0.01, and it was divided by 10 at the 400th and 800th epochs. The complementary learning was conducted in the first 800 epochs, followed by selective complementary learning in the next 1000 epochs. The last 200 epochs were used for conducting traditional learning. The batch-size of the deep learning-based methods was set to 128.

In the experiments, the classification performance was mainly evaluated using overall accuracy (OA), average accuracy (AA), and Kappa coefficient (K). Besides, the area under ROC curve (AUC) was also adopted to evaluate the detection ability of different methods. Experiments were repeated ten times.

(2) Experimental Setup for Semi-Supervised Classification: In semi-supervised classification, 8000 samples were chosen randomly as the unlabeled samples and they were also the testing samples. 20, 25, and 30 training samples (denoted by N) for each class were selected as the labeled training set for exploring the classification performance of different methods, but only 15 labeled examples were chosen if the corresponding class has less than 30 samples.

The proposed Mix-PL-CL method was compared with popular semi-supervised classification methods, such as label propagation (LP), Laplacian support vector machine

(LapSVM), EMP-LapSVM, pseudo-label (PL) AROC-DP [65], and proposed Mix-PL. Besides, supervised methods like EMP-CNN and MCNN-CP were also considered.

In LapSVM based methods, we adopted grid search method with five-fold cross validation to find the proper γ_A and γ_I $\left(\gamma_A = 10^{-5}, 10^{-4}, \ldots, 10^1, \gamma = 10^{-5}, 10^{-4}, \ldots 10^1\right)$. Besides, a one-against-one multiclass strategy which involved a parallel architecture consisting of $c(c-1)/2$ different SVMs was adopted, where c is the number of classes. In the graph-based method like Label Propagation, we used a RBF kernel to construct a graph, and the clamping factor α was set to be 0.2, which represented that the 80 percent of original label distribution was always reserved and it changed the confidence of the distribution within 20 percent. The parameter of the kernel was chosen from $\{10^{-3}, \ldots, 10^3\}$. LP iterated on a modified version of the original graph and normalizes the edge weights by computing the normalized graph Laplacian matrix, besides, it minimized a loss function that has regularization properties to make classification performance robust against noise.

When training g, the initial learning rate was set to 0.001, and it was divided by ten after 60 epochs. The number of epochs, denoted by T_5, was set to 450. The hyper parameters α used in mixup was fixed to one and balancing coefficient $\rho(t)$ was obtained by Equation (19). In the experiments, t_1 and t_2 were set to be 120 and 300, respectively. And ρ_{end} was set to be two. Figure 7 illustrated the $\rho(t)$ set in the experiments. The influence of ρ_{end} and α would analyzed later. The number of iterations, denoted by T_4, was set to be two. That meant Mix-PL was used twice and CL-CNN was used once in the iteration. T_4 would have a great impact on classification performance, and it would be analyzed in the experiments.

$$\rho(t) = \begin{cases} 0 & t < t_1 \\ \frac{t-t_1}{t_2-t_1}\rho_{end} & t_1 \leq\ t \leq t_2 \\ \rho_{end} & t > t_2 \end{cases}, \tag{19}$$

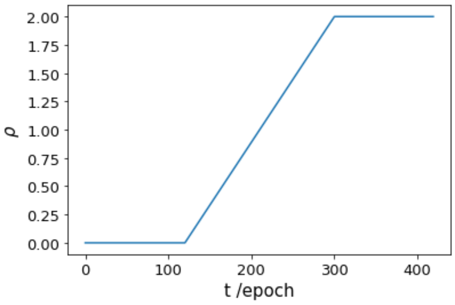
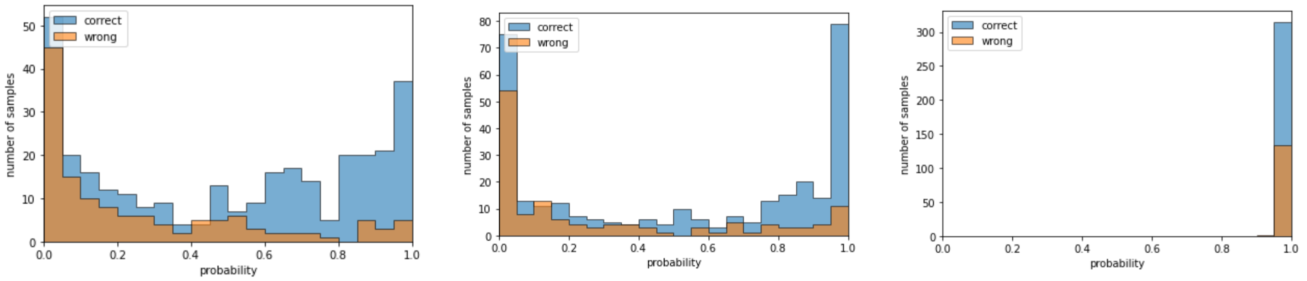


Figure 7. Balancing coefficient $\rho(t)$.

5.3. Results of HSI Classification with Noisy Labels

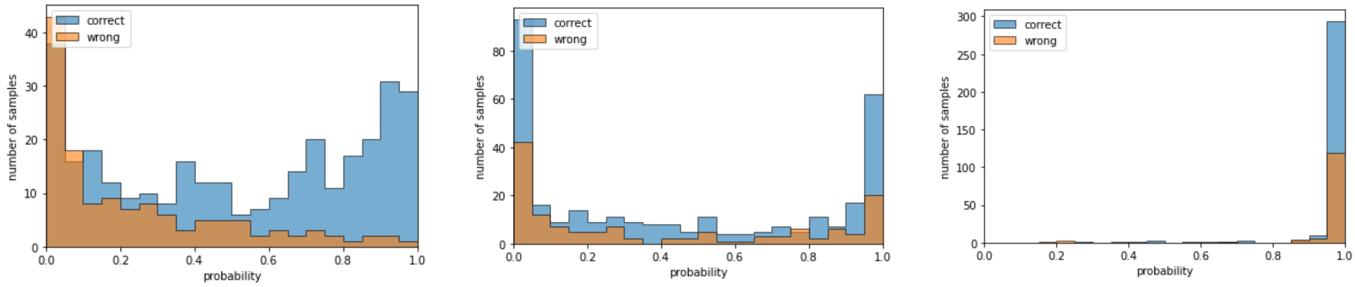
(1) Training Process of CNN Using CE Loss in the Presence of Label Noise: Generally, CNNs were capable of memorizing completely random labels and exhibited poor generalization capability in the presence of noisy labels.

Figures 8–10 showed the distribution of training data in different learning stages with 30% label noise, according to probability $\boldsymbol{p}_y$. From Figures 8a, 9a and 10a, one could see that a large number of clean samples together with few noisy samples lay in the right of the graphs. This meant that they were firstly learned by the deep model in the early training stage. With training going on, noisy samples would move toward to the right, indicating that the model was becoming overfitting to noisy samples, just like Figures 8b, 9b and 10b showed. When the training was completed, most of training samples had large values of $\boldsymbol{p}_y$, meaning that the model had memorized most of the noisy training set, just like Figures 8c, 9c and 10c showed.



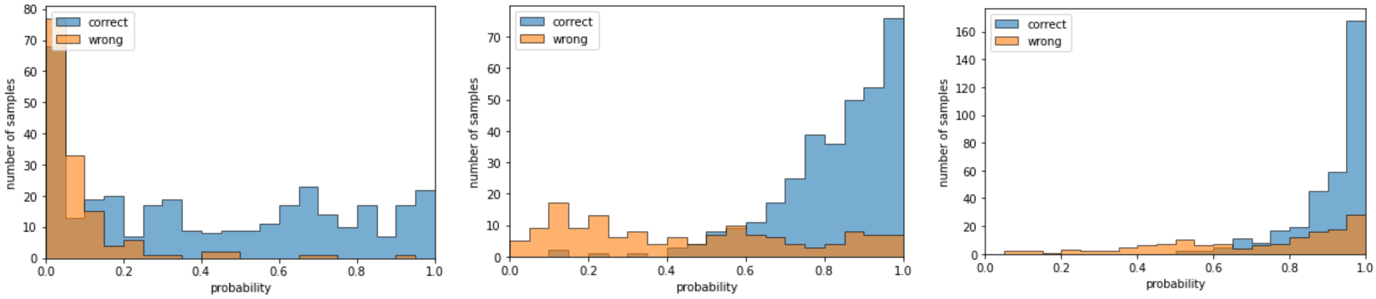
(**a**) (**b**) (**c**)

Figure 8. The distribution of Indian Pines training data in different learning stages with 30% label noise, according to probability p_y. (**a**) early stage of learning; (**b**) middle stage of learning; (**c**) late stage of learning.



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

Figure 9. The distribution of Houston training data in different learning stages with 30% label noise, according to probability p_y. (**a**) early stage of learning; (**b**) middle stage of learning; (**c**) late stage of learning.



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

Figure 10. The distribution of Salinas training data in different learning stages with 30% label noise, according to probability p_y. (**a**) early stage of learning; (**b**) middle stage of learning; (**c**) late stage of learning.

(2) Training Process of CNN Using CL in the Presence of Label Noise: Figures 11–13 showed the distribution of training data using different learning methods with 30% label noise, according to probability p_y Figures 11a, 12a, 13a and 11b, 12b, 13b respectively showed the histogram of the training data after traditional learning (using CE) and CL. Different from the fact that the probability p_y of both clean and noisy samples seemed large in PL, the probability of noisy samples was much lower than those of clean samples in CL, indicating the CL's capability to prevent the CNN from overfitting to noisy samples. After CL, noisy samples and clean samples could be separated, which could be seen in Figures 11b, 12b and 13b clearly.

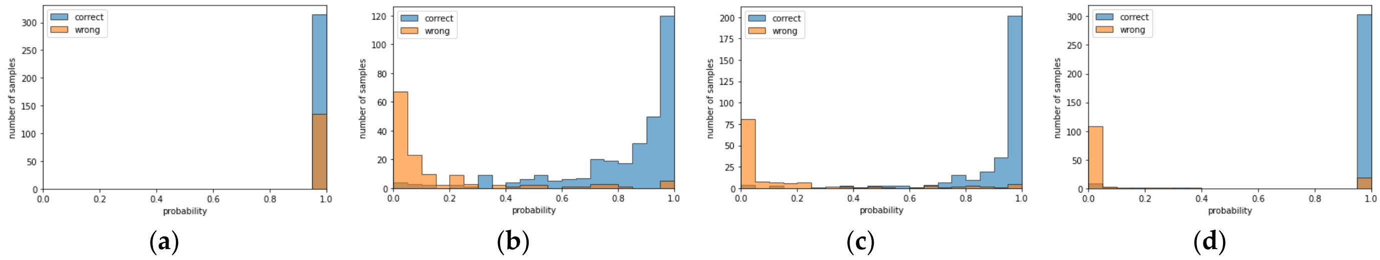


Figure 11. The distribution of Indian Pines training data in different learning strategies with 30% label noise, according to probability p_y. (**a**) traditional learning; (**b**) complementary learning; (**c**) selective complementary learning following CL; (**d**) traditional learning using samples whose $p_y > 0.5$.

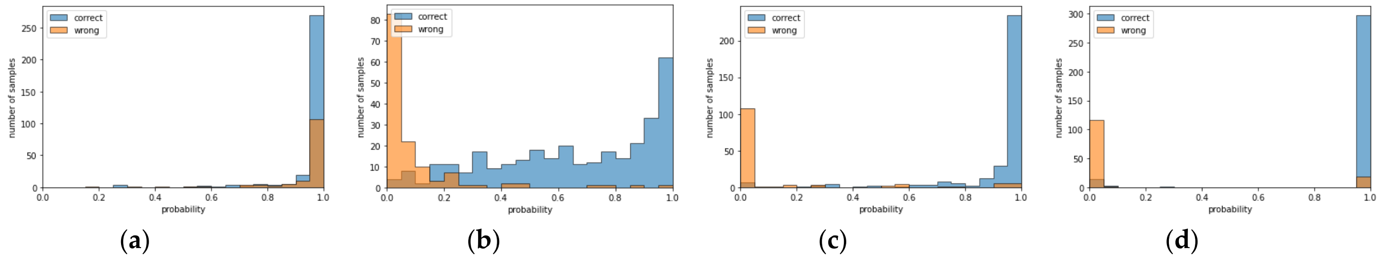


Figure 12. The distribution of Houston training data in different learning strategies with 30% label noise, according to probability p_y. (**a**) traditional learning; (**b**) complementary learning; (**c**) selective complementary learning following CL; (**d**) traditional learning using samples whose $p_y > 0.5$.

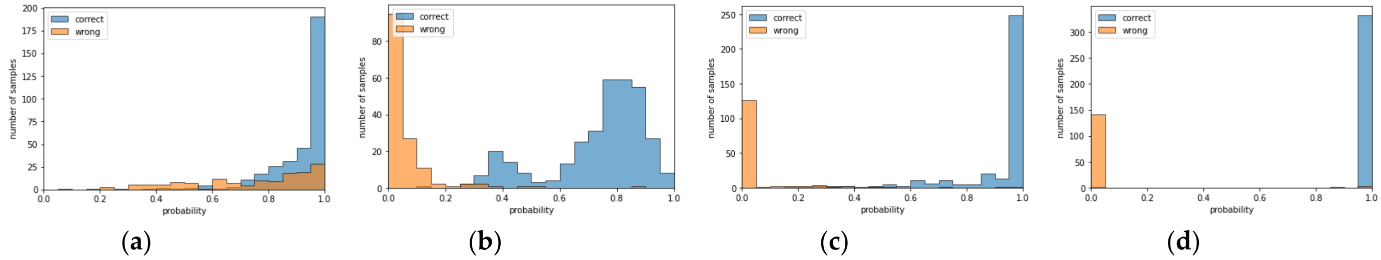


Figure 13. The distribution of Salinas training data in different learning strategies with 30% label noise, according to probability p_y. (**a**) traditional learning; (**b**) complementary learning; (**c**) selective complementary learning following CL; (**d**) traditional learning using samples whose $p_y > 0.5$.

However, there was still an overlap between the distributions of clean and noisy samples, which could be seen in Figures 11b, 12b and 13b. And most of noisy samples had the output p_y less than $1/c$, which was align with cognition. Figures 11c, 12c and 13c showed that there was smaller overlap after training the CNN only with the data having p_y over $1/c$. Using thresholds, the samples involved in training tended to be less noisy than before, and thus improved the convergence of the CNN.

Figures 11c, 12c and 13c showed that noisy samples could be detected by judging if the values of p_y were smaller than 0.5 for simplicity and universality, which meant samples having p_y less than 0.5 would likely to be noisy samples. After training CNN only with samples having probability p_y larger than 0.5, almost all clean samples exhibited high p_y, which could be seen from Figures 11d, 12d and 13d.

(3) Detection Performance Compared with Other Methods: Table 5 showed the AUC of different noisy labels detection methods. From Table 5, one could see the proposed CL performed best on three datasets, compared with DP, KSDP, and SSDP. And it could work well in noisy datasets with different noise ratios. The results showed that the proposed method had better detection ability.

(4) Classification Performance Compared with Other Methods: Tables 6–8 showed the classification results of different methods on three datasets. And the detailed classification results with 30% label noise could be seen in Tables A1–A3, Appendix A.

Table 5. AUC of detection results on the three datasets.

Dataset	10%				20%				30%			
	DP	**KSDP**	**SSDP**	**SeCL**	**DP**	**KSDP**	**SSDP**	**SeCL**	**DP**	**KSDP**	**SSDP**	**SeCL**
Indian Pines	0.9027	0.9281	0.9411	0.9756	0.8994	0.9277	0.9391	0.9778	0.8988	0.9248	0.9386	0.9672
Houston	0.9130	0.9262	0.9353	0.9503	0.9007	0.9123	0.9285	0.9449	0.8875	0.8932	0.9124	0.9404
Salinas	0.9679	0.9786	0.9861	0.9951	0.9681	0.9776	0.9844	0.9956	0.9678	0.9751	0.9806	0.9955

Table 6. Testing data classification results (mean ± standard deviation) on the Indian Pines dataset.

Noise Ratio		RBF-SVM	EMP-SVM	CNN	MCNN-CP	CNN-Lq	DP-CNN	KSDP-CNN	SSDP-CNN	SeCL-CNN
10%	OA (%)	62.25 ± 2.65	74.52 ± 2.52	76.84 ± 2.04	83.94 ± 1.76	82.51 ± 1.85	79.55 ± 1.72	80.01 ± 1.74	81.86 ± 1.68	82.70 ± 1.96
	AA (%)	73.50 ± 1.51	83.17 ± 1.13	83.32 ± 0.96	87.77 ± 1.65	89.59 ± 1.63	86.20 ± 1.57	86.60 ± 1.03	88.25 ± 1.67	89.36 ± 1.73
	K × 100	57.57 ± 2.85	71.23 ± 2.73	73.88 ± 2.23	81.81 ± 1.95	80.18 ± 2.06	76.88 ± 1.88	77.40 ± 1.87	79.45 ± 1.92	80.35 ± 2.20
20%	OA (%)	59.55 ± 1.99	71.16 ± 2.70	67.45 ± 2.52	76.91 ± 2.16	78.19 ± 2.61	72.81 ± 4.56	76.79 ± 2.59	78.81 ± 1.94	79.98 ± 2.40
	AA (%)	70.98 ± 1.71	80.29 ± 1.91	73.19 ± 2.01	79.98 ± 1.07	85.26 ± 1.44	82.66 ± 1.86	84.79 ± 0.78	86.11 ± 1.44	88.04 ± 1.54
	K × 100	84.58 ± 1.95	67.48 ± 2.82	63.48 ± 2.70	73.97 ± 2.30	75.33 ± 2.83	69.43 ± 4.99	73.79 ± 2.79	76.35 ± 2.12	77.28 ± 2.63
30%	OA (%)	55.38 ± 4.53	67.11 ± 3.54	57.34 ± 2.87	68.16 ± 3.27	66.36 ± 5.14	70.43 ± 2.82	72.22 ± 2.64	72.88 ± 2.47	73.90 ± 2.94
	AA (%)	66.90 ± 2.77	76.60 ± 2.19	63.48 ± 1.97	72.21 ± 2.06	75.32 ± 3.01	78.27 ± 1.68	81.00 ± 1.73	82.62 ± 1.24	83.44 ± 2.07
	K × 100	49.94 ± 4.53	62.92 ± 3.70	52.56 ± 3.00	64.30 ± 3.48	62.30 ± 5.51	66.72 ± 3.01	68.66 ± 2.84	69.86 ± 2.67	70.51 ± 3.21

Table 7. Testing data classification results (mean ± standard deviation) on the Houston dataset.

Noise Ratio		RBF-SVM	EMP-SVM	CNN	MCNN-CP	CNN-Lq	DP-CNN	KSDP-CNN	SSDP-CNN	SeCL-CNN
10%	OA (%)	82.81 ± 2.14	85.65 ± 1.91	82.03 ± 1.42	88.01 ± 1.59	86.47 ± 1.62	84.96 ± 1.42	85.76 ± 1.04	86.29 ± 1.37	86.95 ± 2.18
	AA (%)	82.96 ± 1.81	86.11 ± 1.71	82.94 ± 1.43	89.09 ± 1.35	87.82 ± 1.45	86.26 ± 1.38	87.02 ± 0.96	88.25 ± 1.45	88.42 ± 2.04
	K × 100	81.40 ± 2.31	84.48 ± 2.06	80.60 ± 1.54	87.05 ± 1.72	85.38 ± 1.75	83.76 ± 1.53	84.62 ± 1.12	79.45 ± 1.89	85.89 ± 2.36
20%	OA (%)	79.92 ± 0.85	82.26 ± 0.85	71.29 ± 0.92	82.13 ± 2.46	81.97 ± 1.50	80.00 ± 2.64	81.25 ± 1.60	82.43 ± 1.96	83.68 ± 2.57
	AA (%)	80.65 ± 0.70	83.09 ± 0.80	72.17 ± 0.95	83.18 ± 2.49	83.02 ± 1.60	81.55 ± 2.31	82.66 ± 1.11	83.91 ± 1.88	85.01 ± 2.58
	K × 100	78.29 ± 0.92	80.81 ± 0.92	69.03 ± 0.99	80.70 ± 2.66	80.52 ± 1.63	78.41 ± 2.85	79.74 ± 1.71	81.15 ± 2.03	82.37 ± 2.77
30%	OA (%)	77.05 ± 1.91	78.88 ± 1.62	62.05 ± 1.96	75.58 ± 2.63	74.44 ± 2.12	75.25 ± 2.36	76.65 ± 2.27	78.16 ± 2.00	80.00 ± 2.51
	AA (%)	77.87 ± 1.33	79.96 ± 1.54	62.33 ± 1.92	76.02 ± 2.33	75.21 ± 2.28	76.93 ± 2.22	78.36 ± 2.26	79.49 ± 1.49	81.41 ± 2.45
	K × 100	75.18 ± 2.06	77.16 ± 1.75	59.09 ± 2.11	73.63 ± 2.84	72.41 ± 2.29	73.29 ± 2.53	74.77 ± 2.93	76.40 ± 2.13	78.39 ± 2.72

Table 8. Testing data classification results (mean ± standard deviation) on the Salinas dataset.

Noise Ratio		RBF-SVM	EMP-SVM	CNN	MCNN-CP	CNN-Lq	DP-CNN	KSDP-CNN	SSDP-CNN	SeCL-CNN
10%	OA (%)	87.01 ± 1.92	90.09 ± 0.89	88.06 ± 2.03	92.68 ± 1.28	92.14 ± 2.29	90.90 ± 1.86	91.80 ± 2.64	92.24 ± 2.56	92.57 ± 2.45
	AA (%)	92.61 ± 0.93	94.45 ± 0.46	92.06 ± 1.14	94.34 ± 0.96	96.10 ± 0.89	95.17 ± 0.54	95.44 ± 1.25	95.86 ± 1.67	96.39 ± 1.26
	K × 100	85.57 ± 2.09	88.97 ± 0.98	86.78 ± 2.24	91.62 ± 1.42	91.28 ± 2.53	89.91 ± 2.04	90.91 ± 2.91	91.32 ± 2.82	91.74 ± 2.70
20%	OA (%)	85.80 ± 2.17	88.22 ± 2.01	81.36 ± 3.04	88.43 ± 2.12	91.73 ± 2.10	89.92 ± 2.04	90.62 ± 1.78	91.31 ± 1.80	92.13 ± 1.62
	AA (%)	91.79 ± 0.86	93.33 ± 0.98	85.01 ± 1.80	89.85 ± 2.03	95.69 ± 1.26	94.42 ± 0.98	94.56 ± 1.31	95.01 ± 1.21	95.88 ± 0.75
	K × 100	84.23 ± 2.34	86.912.21	79.39 ± 3.32	87.15 ± 2.35	90.81 ± 2.34	88.81 ± 2.24	89.57 ± 1.99	90.35 ± 2.03	91.27 ± 1.79
30%	OA (%)	85.59 ± 2.05	86.85 ± 2.02	72.36 ± 2.30	84.53 ± 2.79	89.99 ± 1.92	87.10 ± 2.38	88.35 ± 3.32	89.76 ± 1.67	91.51 ± 2.31
	AA (%)	91.24 ± 1.09	92.23 ± 1.35	75.44 ± 1.24	85.27 ± 2.95	93.77 ± 1.57	90.79 ± 2.01	92.25 ± 1.17	92.86 ± 1.40	95.07 ± 1.48
	K × 100	83.98 ± 2.22	85.38 ± 2.21	69.54 ± 2.48	82.84 ± 3.07	88.89 ± 2.13	85.70 ± 2.63	87.09 ± 2.53	88.62 ± 1.85	90.57 ± 2.55

From these results, one could see that though CNN-based models performed well in traditional HSI classification tasks. However, they exhibited poor generalization capability when noisy labels existed. For example, CNN and MCNN-CP achieved excellent classification results with non-noisy labels or less noisy labels, compared with EMP-SVM. When the noise ratio was 10%, MCNN-CP maintained highest classification results. But CNN-based models' OA, AA, and K decreased drastically as the noise ratio increased. And they couldn't perform as well as EMP-SVM in the case of higher noise ratio. It behaved the same on the other two datasets. One could also see that the accuracies of EMP-SVM didn't decrease as drastically as CNN when the noise ratio increased, but EMP-SVM's classification performance was limited by handcrafted features.

The proposed SeCL-CNN outperformed other methods in terms of OA, AA, and K. For example, in Table 6, one could see that the OA of SeCL-CNN was 73.90% when the noise ratio was 30%. This accuracy was higher than the ones obtained by the other methods. SeCL-CNN outperformed SSDP-CNN by 1.02%, 0.82%, and 0.0065 in terms of OA, AA, and K, respectively. When noise ratio was 10% and 20%, SeCL-CNN gained better classification results than compared methods in terms of OA, AA and K, except that the accuracies were slightly lower than the ones obtained by MCNN-CP. However, our proposed method mainly focused on the cleaning of noisy-labeled samples and could be combined with any classifier, including MCNN-CP, to complete the final classification.

(5) Ablation Studies: Table 9 showed the results obtained by ablation studies, when the noise ratio was 30%. From Table 9, one could see that without EMP, the OA decreased 1.67%, 1.38% and 1.29 on the three datasets, respectively. And the AUC decreased 0.0146, 0.0058, 0.0028. It showed that the use of EMP enhanced the capacity of noisy labels detection and finally improved the classification performance. And it was similar to selective CL. Without selective CL, the AUC and OA both decreased, which meant the importance of selective CL.

Table 9. Ablation studies for inaccurate supervision on the three datasets (30%label noise).

Dataset	Metric	SeCL-CNN	Without EMP	Without Selective CL
Indian	OA (%) AUC	73.90 0.9672	72.23 0.9526	72.98 0.9559
Houston	OA (%) AUC	80.00 0.9404	78.62 0.9346	79.07 0.9373
Salinas	OA (%) AUC	91.51 0.9955	90.20 0.9927	90.62 0.9915

5.4. Results of HSI Semi-Supervised Classification

(1) Classification Performance Compared with Other Methods: Tables 10–12 showed the classification results of different supervised and semi-supervised classification methods on the three datasets. And the detailed semi-supervised classification results with 25 labeled training samples per class are reported in Tables A4–A6, Appendix A.

From Table 10, one could see that the proposed Mix-PL-CL achieved the best performance compared with other methods. Mix-PL-CL outperformed AROC-DP by 1.03%, 0.19%, and 0.00115 in terms of OA, AA, and K when the number of samples per class was 25, and the accuracies of each class obtained by Mix-PL-CL demonstrated good performance compared with other compared methods, including supervised methods like MCNN-CP. Besides, the accuracies gained by different classification methods increased as the number of labeled training samples per class grew, and the proposed methods, i.e., Mix-PL and Mix-PL-CL, still achieved higher classification accuracies, which showed the superior classification ability.

Table 11 showed classification results of different methods on Houston dataset. The usage of mixup helped PL improved classification accuracies. And, Mix-PL-CL achieved better classification results when compared with Mix-PL, which showed the importance of CL. On Houston dataset, Mix-PL-CL outperformed AROC-DP by 0.70%, 0.55%, and 0.0086 in terms of OA, AA and K when the number of samples per class was 25. And the highest classification accuracies were obtained by the proposed method in the case of different numbers of labeled training samples.

From Table 12, one could see that the proposed Mix-PL-CL still achieved superior performance on Salinas dataset with different numbers of labeled training samples.

Table 10. Testing data classification results (mean ± standard deviation) on the Indian Pines dataset.

N		EMP-CNN	MCNN-CP	LP	LapSVM	EMP-LapSVM	PL	AROC-DP	Mix-PL	Mix-PL-CL
20	OA (%)	88.67 ± 1.99	89.97 ± 1.43	55.96 ± 2.15	59.02 ± 1.89	84.10 ± 2.58	89.78 ± 2.00	90.73 ± 1.68	91.65 ± 2.11	92.54 ± 1.93
	AA (%)	93.00 ± 1.01	94.76 ± 0.76	66.93 ± 1.56	70.27 ± 1.44	89.86 ± 1.29	93.99 ± 1.08	94.47 ± 0.87	94.48 ± 1.21	94.67 ± 1.21
	K × 100	87.08 ± 2.24	88.11 ± 1.60	50.45 ± 2.28	53.89 ± 2.05	81.96 ± 2.92	88.36 ± 2.26	89.45 ± 1.88	90.48 ± 2.39	91.45 ± 2.20
30	OA (%)	92.83 ± 1.45	93.88 ± 1.46	59.49 ± 1.25	63.52 ± 1.07	86.55 ± 2.39	93.56 ± 1.57	93.69 ± 1.80	94.41 ± 1.39	94.82 ± 1.32
	AA (%)	95.84 ± 0.75	96.35 ± 0.56	68.47 ± 0.81	73.38 ± 2.11	91.37 ± 1.37	96.06 ± 0.74	96.11 ± 0.85	96.33 ± 0.58	96.72 ± 0.64
	K × 100	91.82 ± 1.64	92.99 ± 1.66	54.28 ± 1.30	58.89 ± 1.19	84.70 ± 2.68	92.63 ± 1.77	92.46 ± 2.04	93.57 ± 1.56	94.06 ± 1.50
25	OA (%)	91.78 ± 2.22	92.74 ± 1.49	58.12 ± 1.33	61.27 ± 1.27	85.09 ± 2.34	92.87 ± 2.30	92.30 ± 1.72	93.12 ± 3.28	93.33 ± 2.29
	AA (%)	94.95 ± 1.20	96.19 ± 0.74	67.86 ± 1.27	71.60 ± 1.64	90.57 ± 1.43	95.35 ± 1.26	95.55 ± 0.77	95.37 ± 1.27	95.74 ± 1.11
	K × 100	90.60 ± 2.52	91.71 ± 1.69	52.73 ± 1.40	56.26 ± 1.46	83.07 ± 2.61	91.83 ± 2.62	91.20 ± 1.95	92.12 ± 2.71	92.35 ± 2.61

Table 11. Testing data classification results (mean ± standard deviation) on the Houston dataset.

N		EMP-CNN	MCNN-CP	LP	LapSVM	EMP-LapSVM	PL	AROC-DP	Mix-PL	Mix-PL-CL
20	OA (%)	90.48 ± 0.97	92.53 ± 1.27	78.21 ± 0.99	80.63 ± 1.06	85.63 ± 1.53	91.52 ± 0.98	92.90 ± 0.86	92.89 ± 1.12	93.39 ± 1.06
	AA (%)	91.38 ± 0.81	93.66 ± 1.10	78.91 ± 0.83	81.12 ± 1.21	86.75 ± 1.37	92.15 ± 0.84	93.82 ± 0.73	93.44 ± 0.92	94.29 ± 0.87
	K × 100	89.71 ± 1.05	91.93 ± 1.38	76.45 ± 1.08	79.06 ± 1.16	84.47 ± 1.66	90.86 ± 0.92	92.33 ± 0.93	92.01 ± 1.25	92.86 ± 1.14
30	OA (%)	93.34 ± 0.86	94.34 ± 0.80	81.04 ± 0.89	83.49 ± 1.08	88.13 ± 1.26	94.12 ± 1.05	94.59 ± 0.68	94.82 ± 1.28	95.62 ± 0.98
	AA (%)	94.13 ± 0.67	95.32 ± 0.71	81.47 ± 0.79	83.69 ± 1.02	88.89 ± 1.13	94.86 ± 0.89	95.56 ± 0.69	95.49 ± 1.12	96.36 ± 0.81
	K × 100	92.80 ± 0.93	93.88 ± 0.87	79.50 ± 0.96	82.14 ± 1.16	87.17 ± 1.36	93.52 ± 1.12	94.12 ± 1.12	94.41 ± 1.39	95.26 ± 1.06
25	OA (%)	92.05 ± 0.82	93.44 ± 0.99	79.86 ± 0.88	82.30 ± 1.04	86.52 ± 1.24	93.39 ± 1.33	93.48 ± 1.15	93.77 ± 0.95	94.18 ± 0.82
	AA (%)	92.86 ± 0.76	94.53 ± 0.98	80.37 ± 0.85	82.55 ± 1.18	87.54 ± 1.24	94.23 ± 1.28	94.43 ± 1.16	94.75 ± 0.89	94.98 ± 0.86
	K × 100	91.42 ± 0.89	92.91 ± 1.07	78.22 ± 0.94	80.86 ± 1.13	85.43 ± 1.34	92.86 ± 1.44	92.95 ± 1.24	93.27 ± 1.02	93.71 ± 0.89

Table 12. Testing data classification results (mean ± standard deviation) on the Salinas dataset.

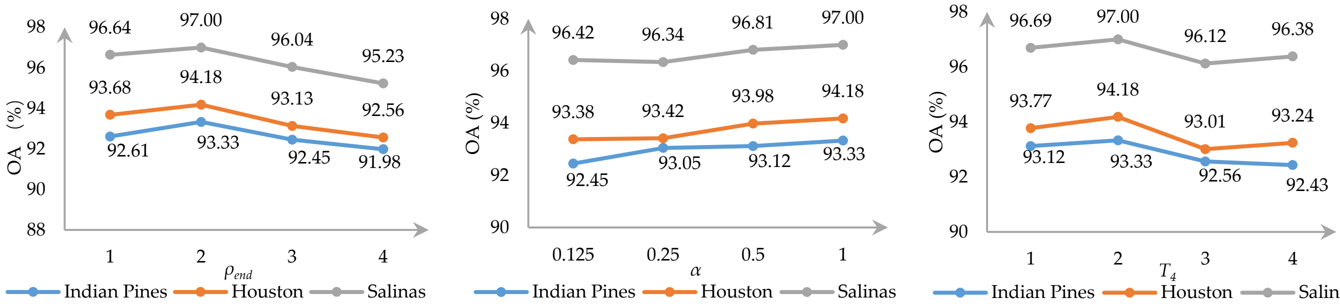
N		EMP-CNN	MCNN-CP	LP	LapSVM	EMP-LapSVM	PL	AROC-DP	Mix-PL	Mix-PL-CL
20	OA (%)	94.60 ± 3.32	95.77 ± 1.50	83.77 ± 0.88	85.37 ± 1.92	91.38 ± 1.62	95.41 ± 1.37	95.47 ± 1.86	95.94 ± 1.68	96.20 ± 1.13
	AA (%)	97.88 ± 1.80	98.12 ± 0.38	91.25 ± 0.46	91.43 ± 1.25	94.81 ± 1.03	97.99 ± 0.48	98.26 ± 0.58	98.02 ± 0.75	98.29 ± 0.51
	K × 100	94.00 ± 3.81	95.30 ± 1.55	82.00 ± 0.98	83.76 ± 2.12	90.41 ± 1.81	94.90 ± 1.51	94.97 ± 2.05	95.45 ± 2.02	95.77 ± 1.25
30	OA (%)	95.72 ± 1.37	96.44 ± 0.67	84.30 ± 0.76	86.14 ± 1.41	92.90 ± 0.94	96.36 ± 2.59	96.95 ± 1.13	96.85 ± 1.60	97.18 ± 0.84
	AA (%)	98.41 ± 0.48	98.43 ± 0.45	91.83 ± 0.31	92.43 ± 0.91	95.85 ± 0.63	98.73 ± 0.78	98.91 ± 0.40	98.80 ± 0.78	98.83 ± 0.44
	K × 100	95.25 ± 1.51	96.04 ± 0.95	82.60 ± 0.81	84.61 ± 1.54	92.10 ± 1.04	95.97 ± 2.85	96.67 ± 1.26	96.50 ± 1.80	96.87 ± 0.93
25	OA (%)	94.95 ± 2.46	96.17 ± 0.98	84.13 ± 1.19	86.12 ± 1.96	91.93 ± 1.71	95.97 ± 2.25	96.18 ± 1.72	96.69 ± 0.71	97.00 ± 0.85
	AA (%)	98.24 ± 0.80	98.37 ± 0.37	91.91 ± 0.44	92.01 ± 1.01	95.18 ± 1.10	98.49 ± 0.82	98.63 ± 0.44	98.82 ± 0.22	98.91 ± 0.30
	K × 100	94.40 ± 2.70	95.64 ± 0.85	82.40 ± 1.30	84.59 ± 2.15	91.02 ± 1.91	95.53 ± 2.48	95.76 ± 1.34	96.33 ± 0.78	96.67 ± 0.94

(2) Ablation Studies: Table 13 showed the results obtained by ablation studies, when the number of training samples per class was 25. From Table 13, one could see that every module contributed to the final classification results. (1) EMP was used in CL-Mix-PL to reduce the computational complexity of HSI classification, which made the model be less overfitting. And without EMP, the OA on the three dataset decreased. (2) Without PL, the model (CL-CNN) only used ordinary CNN to generate pseudo labels for unlabeled samples, which were less accurate than the ones generated by Mix-PL-CL. And one could see that the OA on the three dataset decreased 0.97%, 1.37%, 1.86%, compared to Mix-PL-CL. (3) CL was used in Mix-PL-CL to filter the noisy pseudo-labels generated by Mix-PL. And one could see that the OA of Mix-PL was lower than that of Mix-PL-CL on three datasets, which showed the importance of CL. (4) Without mixup, the model (CL-PL) only used Pseudo-Label method to generate pseudo labels for unlabeled samples. the results showed that the use of mixup led to gains of OA over three datasets.

Table 13. The OA (%) of ablation studies for semi-supervised classification on the three datasets (N = 25).

Dataset	Mix-PL-CL	Without EMP	Without PL	Without CL	Without Mixup
Indian	93.33	92.15	92.36	93.12	92.98
Houston	94.18	92.76	92.81	93.77	93.75
Salinas	97.00	96.05	95.14	96.69	96.63

Figure 14 showed the influence of ρ_{end}, T_4, and α, respectively. From Figure 14a, one could see that it was better to set the value of ρ_{end} to 2. A higher ρ_{end} would make the model quickly be overfitting to the noisy pseudo-labeled samples and degrade the accuracy. On the contrary, the lower ρ_{end} would make model learn fewer from the unlabeled samples and get lower classification results. Figure 14b showed that α was better set to one. Figure 14c showed that T_4 was better set to two. When T_4 was one, the model was actually Mix-PL. With T_4 increased, the model would gradually be overfitting to the pseudo-labeled samples and degrade the accuracy.



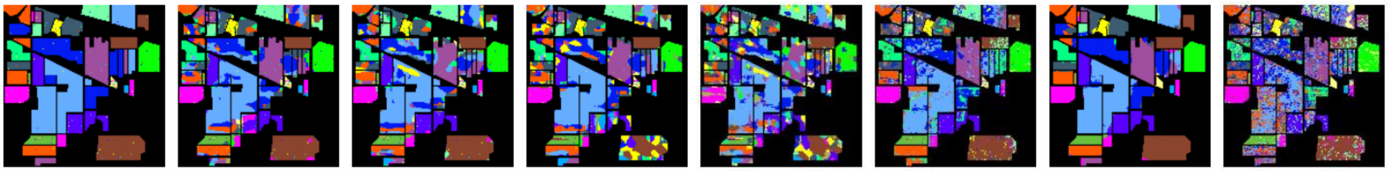
(**a**) (**b**) (**c**)

Figure 14. The influence of ρ_{end} and α on OA with N = 25. (**a**) OA with different values of ρ_{end}, while $\alpha = 1.0$, $T_4 = 2$; (**b**) OA with different values of α, while $\rho_{end} = 2$, $T_4 = 2$; (**c**) OA with different values of T_4, while $\alpha = 1.0$, $\rho_{end} = 2$.

5.5. Classification Maps of Different Classification Methods

Figures 15–17 showed the classification maps of different methods, including classification methods in the presence of noisy labels and semi-supervised methods, on the three datasets.

From these maps, one could clearly see the differences. For example, it showed that pixels near some noisy samples were misclassified, while getting correct labels in anti-label noise methods such as SeCL-CNN and KSDP-CNN. And SeCL-CNN performed better than methods for comparison. And for semi-supervised classification, the proposed Mix-PL-CL achieved better classification results.



(**a**) (**b**) (**c**) (**d**) (**e**) (**f**) (**g**) (**h**)

Figure 15. Indian Pines. (**a**) The ground-truth map with noisy training samples, the classification map using (**b**) SeCL-CNN; (**c**) KSDP-CNN; (**d**) CNN-Lq; (**e**) CNN; (**f**) EMP-SVM; (**g**) Mix-PL-CL; (**h**) LP.

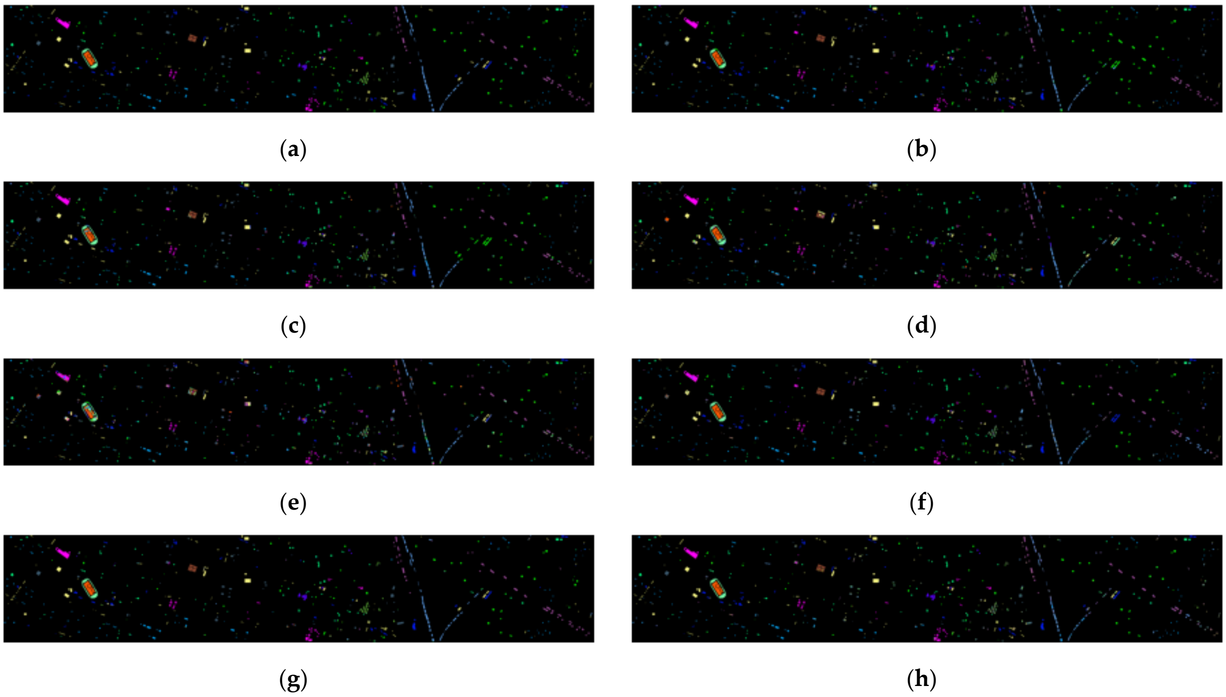


Figure 16. Houston. (**a**) The ground-truth map with noisy training samples, the classification map using (**b**) SeCL-CNN; (**c**) KSDP-CNN; (**d**) CNN-Lq; (**e**) CNN; (**f**) EMP-SVM; (**g**) Mix-PL-CL; (**h**) LP.

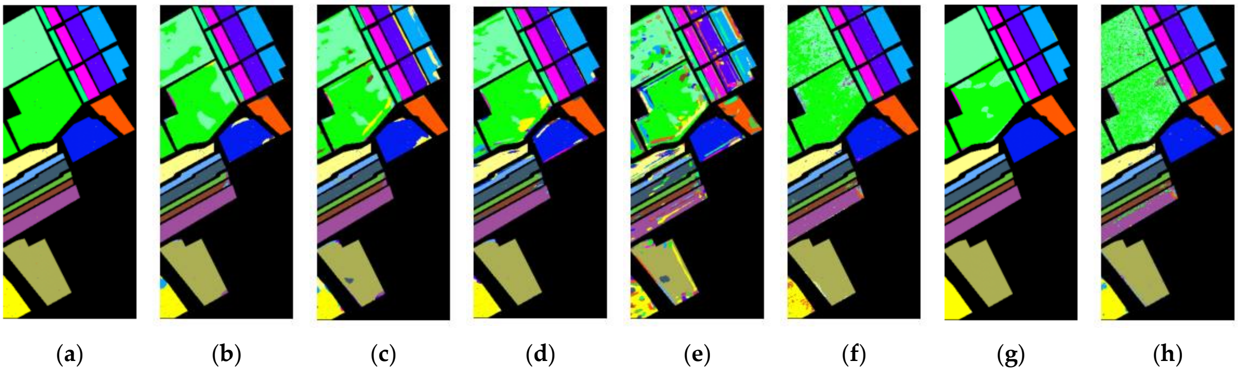


Figure 17. Salinas. (**a**) The ground-truth map with noisy training samples, the classification map using (**b**) SeCL-CNN; (**c**) KSDP-CNN; (**d**) CNN-Lq; (**e**) CNN; (**f**) EMP-SVM; (**g**) Mix-PL-CL; (**h**) LP.

6. Conclusions

In this study, the strategy of complementary learning was explored for hyperspectral weakly supervised classification. For inaccurate supervision, complementary learning method was proposed for HSI classification. Then SeCL, using selective CL, was proposed for classification in the presence of noisy labels. For incomplete supervision, Mix-PL was proposed, which combines mixup and Pseudo-Label method. And then Mix-PL-CL was designed aiming at better semi-supervised HSI classification capacity.

Experimental conclusions can be drawn for the three widely used datasets (i.e., Indian Pines, Houston, and Salinas datasets): (1) The CL strategy can prevent DCNNs from being overfitting to noisy labels and can be used to detect noisy-labeled training samples. The proposed SeCL can further improve the ability of dealing with label noise. (2) According to the experimental results, the proposed Mix-PL can achieve good semi-supervised classification results. And the use of CL (Mix-PL-CL) further improves the classification performance. (3) The classification results on the three datasets demonstrate that the proposed methods for inaccurate and incomplete supervised classification outperformed other studied state-of-the-art methods as well as the conventional techniques. This research provides guidance for further studies to explore complementary learning and weakly supervised learning in the field of HSI classification.

Author Contributions: Conceptualization: Y.C.; methodology: L.H. and Y.C.; writing—original draft preparation: L.H., Y.C. and X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of China under the Grant 61971164 and the Grant U20B2041.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The Houston dataset is available at: https://hyperspectral.ee.uh.edu/. The Indian Pines and Salinas datasets are available at: http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes (accessed on 1 September 2020).

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

Appendix A. Detailed Classification Results

Table A1. Detailed classification results (mean ± standard deviation) with 30% label noise on the Indian Pines dataset.

Noise Ratio	Class	RBF-SVM	EMP-SVM	CNN	MCNN-CP	CNN-Lq	DP-CNN	KSDP-CNN	SSDP-CNN	SeCL-CNN
30%	OA (%)	55.38 ± 4.53	67.11 ± 3.54	57.34 ± 2.87	68.16 ± 3.27	66.36 ± 5.14	70.43 ± 2.82	72.22 ± 2.64	72.88 ± 2.47	73.90 ± 2.94
	AA (%)	66.90 ± 2.77	76.60 ± 2.19	63.48 ± 1.97	72.21 ± 2.06	75.32 ± 3.01	78.27 ± 1.68	81.00 ± 1.73	82.62 ± 1.24	83.44 ± 2.07
	K × 100	49.94 ± 4.53	62.92 ± 3.70	52.56 ± 3.00	64.30 ± 3.48	62.30 ± 5.51	66.72 ± 3.01	68.66 ± 2.84	69.86 ± 2.67	70.51 ± 3.21
	🟨	83.75 ± 11.92	93.13 ± 9.04	74.38 ± 10.25	73.21 ± 11.23	93.75 ± 9.27	88.13 ± 9.46	97.50 ± 5.00	95.62 ± 5.63	99.38 ± 1.88
	🟦	32.80 ± 8.13	49.13 ± 7.34	49.24 ± 5.36	61.56 ± 4.78	54.43 ± 7.74	58.35 ± 5.87	56.36 ± 7.61	58.36 ± 5.73	58.50 ± 6.82
	🟧	43.24 ± 10.50	64.74 ± 9.97	51.08 ± 5.48	57.28 ± 10.34	55.53 ± 9.00	58.86 ± 8.87	61.74 ± 10.58	63.62 ± 8.71	64.35 ± 15.91
	🟩	62.61 ± 9.38	69.61 ± 10.14	69.37 ± 6.21	77.94 ± 6.91	79.28 ± 6.68	83.48 ± 9.03	83.24 ± 8.50	85.51 ± 4.41	89.66 ± 8.20
	🟪	76.05 ± 4.99	82.32 ± 2.86	59.45 ± 7.01	75.77 ± 6.69	71.59 ± 8.30	78.94 ± 11.77	75.96 ± 10.95	80.04 ± 10.93	77.68 ± 9.93
	🟪	80.21 ± 8.03	87.07 ± 7.45	52.74 ± 5.83	69.90 ± 6.54	65.66 ± 6.43	75.26 ± 9.35	79.70 ± 12.68	79.06 ± 10.69	74.74 ± 19.23
	🟦	89.23 ± 7.05	94.62 ± 3.53	63.85 ± 22.31	67.56 ± 18.22	86.15 ± 18.46	81.54 ± 20.12	86.15 ± 16.06	89.24 ± 15.07	99.23 ± 2.31
	🟩	82.25 ± 4.80	94.40 ± 4.29	75.27 ± 8.68	81.64 ± 5.52	89.04 ± 5.94	88.26 ± 8.17	95.20 ± 3.32	95.31 ± 5.07	93.82 ± 4.75
	🟫	80.00 ± 20.00	84.00 ± 14.97	76.00 ± 24.98	88.17 ± 19.84	84.00 ± 23.32	88.00 ± 13.27	94.00 ± 9.17	96.00 ± 8.00	98.00 ± 6.00
	🟪	59.47 ± 13.28	63.95 ± 10.70	59.00 ± 10.12	69.47 ± 8.12	67.19 ± 10.39	68.75 ± 8.95	68.96 ± 6.75	72.65 ± 7.27	72.40 ± 7.08
	🟦	44.27 ± 18.99	56.66 ± 15.36	51.51 ± 9.11	62.66 ± 11.56	60.40 ± 15.31	64.11 ± 8.87	66.89 ± 9.60	69.47 ± 9.33	69.62 ± 8.31
	⬛	40.25 ± 9.41	63.18 ± 11.00	58.37 ± 5.50	67.21 ± 4.62	67.55 ± 7.59	67.94 ± 6.94	71.97 ± 8.46	70.94 ± 6.03	67.67 ± 6.76
	🟩	95.83 ± 2.14	97.43 ± 1.72	68.97 ± 10.11	75.16 ± 8.37	88.80 ± 10.19	88.69 ± 4.93	92.00 ± 5.01	91.60 ± 6.86	95.49 ± 5.61
	🟫	79.03 ± 8.75	84.08 ± 5.56	67.22 ± 9.75	79.68 ± 8.67	77.06 ± 11.75	84.55 ± 7.13	86.04 ± 6.02	90.56 ± 5.63	88.93 ± 5.16
	🟩	36.94 ± 10.52	54.75 ± 10.89	70.51 ± 8.88	72.77 ± 9.95	78.99 ± 9.70	80.76 ± 7.93	84.92 ± 7.88	85.45 ± 7.05	88.76 ± 9.43
	🟨	84.44 ± 9.14	86.51 ± 8.93	68.73 ± 9.94	75.34 ± 13.95	85.71 ± 5.72	96.67 ± 4.23	95.40 ± 4.51	93.65 ± 4.32	96.83 ± 4.65

Table A2. Detailed classification results (mean ± standard deviation) with 30% label noise on the Houston dataset.

Noise Ratio	Class	RBF-SVM	EMP-SVM	CNN	MCNN-CP	CNN-Lq	DP-CNN	KSDP-CNN	SSDP-CNN	SeCL-CNN
30%	OA (%)	77.05 ± 1.91	78.88 ± 1.62	62.05 ± 1.96	75.58 ± 2.63	74.44 ± 2.12	75.25 ± 2.36	76.65 ± 2.27	78.16 ± 2.00	80.00 ± 2.51
	AA (%)	77.87 ± 1.33	79.96 ± 1.54	62.33 ± 1.92	76.02 ± 2.33	75.21 ± 2.28	76.93 ± 2.22	78.36 ± 2.26	79.49 ± 1.49	81.41 ± 2.45
	K × 100	75.18 ± 2.06	77.16 ± 1.75	59.09 ± 2.11	73.63 ± 2.84	72.41 ± 2.29	73.29 ± 2.53	74.77 ± 2.93	76.40 ± 2.13	78.39 ± 2.72
	🟨	89.70 ± 7.10	89.74 ± 7.70	70.77 ± 8.39	72.72 ± 14.54	85.18 ± 7.02	83.16 ± 7.17	82.70 ± 7.49	78.76 ± 7.19	86.47 ± 7.49
	🟦	87.92 ± 6.67	82.71 ± 8.59	60.17 ± 8.57	76.07 ± 11.70	68.24 ± 13.05	74.78 ± 9.09	74.71 ± 8.93	71.43 ± 5.64	76.09 ± 8.26
	🟧	99.10 ± 0.67	99.24 ± 1.05	72.59 ± 10.69	85.27 ± 5.64	87.45 ± 9.46	84.38 ± 7.56	88.32 ± 7.03	88.85 ± 7.78	94.86 ± 5.75
	🟩	91.92 ± 3.12	91.28 ± 3.06	66.96 ± 8.32	83.55 ± 6.19	82.00 ± 5.55	88.66 ± 5.86	88.57 ± 5.24	84.76 ± 2.86	87.20 ± 2.70
	🟪	92.95 ± 4.59	94.56 ± 3.98	70.14 ± 5.00	85.28 ± 5.68	85.72 ± 8.44	88.68 ± 6.37	89.84 ± 5.52	93.01 ± 5.57	93.20 ± 4.57
	🟪	85.32 ± 10.14	85.39 ± 8.33	56.58 ± 5.96	70.15 ± 7.72	67.73 ± 8.18	74.82 ± 6.80	74.98 ± 5.17	72.95 ± 6.82	74.14 ± 13.12

Table A2. *Cont.*

Noise Ratio	Class	RBF-SVM	EMP-SVM	CNN	MCNN-CP	CNN-Lq	DP-CNN	KSDP-CNN	SSDP-CNN	SeCL-CNN
	🟦	76.45 ± 6.74	83.65 ± 9.29	54.29 ± 5.81	68.15 ± 8.11	67.81 ± 5.72	71.75 ± 5.64	73.55 ± 7.71	75.54 ± 4.45	75.98 ± 7.18
	🟩	50.44 ± 7.20	51.46 ± 6.74	49.89 ± 7.38	63.07 ± 5.14	57.35 ± 10.62	57.30 ± 5.83	60.72 ± 7.89	67.91 ± 8.33	55.03 ± 8.90
	🟨	74.17 ± 5.44	76.35 ± 8.98	54.18 ± 8.36	67.24 ± 8.96	64.61 ± 7.05	65.00 ± 8.57	65.96 ± 7.81	68.35 ± 11.48	75.63 ± 8.14
	🟪	65.33 ± 17.97	64.70 ± 11.38	65.36 ± 12.98	73.32 ± 9.21	76.33 ± 9.34	65.90 ± 13.12	69.86 ± 13.90	72.90 ± 12.55	75.74 ± 16.92
	🟦	72.37 ± 6.40	84.25 ± 5.59	62.76 ± 5.35	84.00 ± 5.82	73.70 ± 7.49	67.47 ± 6.30	67.16 ± 7.31	74.06 ± 7.25	77.82 ± 11.42
	⬛	52.81 ± 10.50	51.79 ± 13.84	59.80 ± 8.06	73.71 ± 10.65	71.78 ± 8.75	70.12 ± 9.86	72.19 ± 7.80	77.22 ± 6.83	78.26 ± 8.47
	🟩	37.15 ± 5.51	49.25 ± 5.46	64.31 ± 11.14	77.16 ± 9.11	82.07 ± 7.64	76.63 ± 10.97	77.27 ± 6.31	79.68 ± 8.32	91.50 ± 3.00
	🟫	96.41 ± 1.03	97.66 ± 2.09	64.48 ± 9.56	78.30 ± 10.50	80.53 ± 11.25	95.95 ± 7.96	97.29 ± 4.45	95.58 ± 5.17	91.83 ± 7.64
	🟩	96.08 ± 2.09	97.33 ± 2.90	62.78 ± 5.81	82.31 ± 4.69	77.628.58	89.33 ± 8.18	92.35 ± 4.38	91.33 ± 5.98	87.38 ± 8.52

Table A3. Detailed classification results (mean ± standard deviation) with 30% label noise on the Salinas dataset.

Noise Ratio	Class	RBF-SVM	EMP-SVM	CNN	MCNN-CP	CNN-Lq	DP-CNN	KSDP-CNN	SSDP-CNN	SeCL-CNN
30%	OA (%)	85.59 ± 2.05	86.85 ± 2.02	72.36 ± 2.30	84.53 ± 2.79	89.99 ± 1.92	87.10 ± 2.38	88.35 ± 3.32	89.76 ± 1.67	91.51 ± 2.31
	AA (%)	91.24 ± 1.09	92.23 ± 1.35	75.44 ± 1.24	85.27 ± 2.95	93.77 ± 1.57	90.79 ± 2.01	92.25 ± 1.17	92.86 ± 1.40	95.07 ± 1.48
	K × 100	83.98 ± 2.22	85.38 ± 2.21	69.54 ± 2.48	82.84 ± 3.07	88.89 ± 2.13	85.70 ± 2.63	87.09 ± 2.53	88.62 ± 1.85	90.57 ± 2.55
	🟨	98.18 ± 0.93	99.02 ± 0.46	75.16 ± 9.77	88.88 ± 7.36	97.64 ± 2.95	96.38 ± 3.05	97.18 ± 2.74	97.29 ± 3.31	99.95 ± 0.08
	🟦	97.18 ± 3.13	98.28 ± 1.68	73.40 ± 6.30	89.40 ± 4.16	94.50 ± 5.84	92.77 ± 5.83	93.14 ± 6.44	93.43 ± 4.84	96.68 ± 4.05
	🟧	85.99 ± 12.47	89.90 ± 13.79	69.30 ± 8.63	84.47 ± 9.66	91.51 ± 8.41	88.24 ± 9.30	89.90 ± 9.23	89.95 ± 8.86	92.97 ± 8.47
	🟩	98.91 ± 0.52	98.56 ± 1.95	82.03 ± 7.27	85.03 ± 8.80	97.69 ± 2.00	96.06 ± 3.27	95.84 ± 3.28	94.40 ± 5.75	99.24 ± 1.10
	🟪	95.32 ± 3.70	95.07 ± 4.70	81.02 ± 8.93	88.11 ± 3.09	97.26 ± 4.77	93.15 ± 6.80	94.59 ± 6.22	91.81 ± 5.91	96.15 ± 6.76
	🟪	96.75 ± 3.02	96.13 ± 3.73	75.74 ± 9.07	87.78 ± 9.04	96.70 ± 3.33	93.63 ± 8.22	94.35 ± 7.82	97.01 ± 6.42	97.89 ± 3.50
	🟦	98.94 ± 0.49	98.96 ± 0.62	71.68 ± 9.23	84.18 ± 10.03	93.18 ± 7.71	89.96 ± 8.66	92.33 ± 6.36	92.30 ± 5.07	94.69 ± 3.30
	🟩	72.64 ± 12.11	72.40 ± 11.75	61.23 ± 7.26	81.05 ± 4.70	76.94 ± 6.45	73.06 ± 6.14	74.22 ± 10.33	77.68 ± 4.84	78.37 ± 7.55
	🟨	98.41 ± 1.32	99.24 ± 0.96	84.76 ± 5.54	91.37 ± 8.36	98.49 ± 1.91	93.96 ± 3.53	95.78 ± 2.99	96.88 ± 3.44	97.89 ± 2.00
	🟪	87.29 ± 4.01	89.37 ± 3.57	68.77 ± 11.85	80.59 ± 10.50	91.24 ± 10.35	88.63 ± 5.63	90.77 ± 6.38	92.30 ± 5.07	93.99 ± 8.42
	🟦	90.52 ± 3.91	93.70 ± 1.09	74.33 ± 4.90	81.09 ± 8.77	94.19 ± 7.21	90.28 ± 6.15	92.36 ± 5.72	93.63 ± 6.53	96.85 ± 5.56
	⬛	99.42 ± 0.52	99.92 ± 0.12	82.38 ± 11.29	87.65 ± 4.66	98.38 ± 2.19	95.59 ± 7.88	97.54 ± 4.08	96.85 ± 6.34	98.96 ± 1.85
	🟩	98.51 ± 0.67	98.12 ± 0.75	84.60 ± 8.03	87.10 ± 7.72	97.88 ± 2.17	94.45 ± 7.71	97.90 ± 3.16	97.42 ± 3.86	98.83 ± 1.55
	🟫	92.26 ± 2.51	90.62 ± 7.17	81.74 ± 7.58	84.48 ± 8.40	96.88 ± 7.94	92.97 ± 6.87	94.69 ± 6.02	94.28 ± 5.18	96.67 ± 7.26
	🟩	57.38 ± 10.00	63.62 ± 9.53	67.82 ± 5.42	78.56 ± 6.13	82.27 ± 10.23	82.31 ± 4.92	82.46 ± 4.99	85.07 ± 8.89	86.53 ± 6.53
	🟨	92.19 ± 5.09	92.69 ± 4.83	73.06 ± 10.22	84.53 ± 9.32	95.53 ± 4.36	91.22 ± 4.91	92.98 ± 3.49	95.48 ± 2.51	95.43 ± 3.50

Table A4. Detailed semi-supervised classification results (mean ± standard deviation) on the Indian Pines dataset (N = 25).

N	Class	EMP-CNN	MCNN-CP	LP	LapSVM	EMP-LapSVM	PL	AROC-DP	Mix-PL	CL-MixPL
25	OA (%)	91.78 ± 2.22	92.74 ± 1.49	58.12 ± 1.33	61.27 ± 1.27	85.09 ± 2.34	92.87 ± 2.30	92.30 ± 1.72	93.12 ± 3.28	93.33 ± 2.29
	AA (%)	94.95 ± 1.20	96.19 ± 0.74	67.86 ± 1.27	71.60 ± 1.64	90.57 ± 1.43	95.35 ± 1.26	95.55 ± 0.77	95.37 ± 1.27	95.74 ± 1.11
	K × 100	90.60 ± 2.52	91.71 ± 1.69	52.73 ± 1.40	56.26 ± 1.46	83.07 ± 2.61	91.83 ± 2.62	91.20 ± 1.95	92.12 ± 2.71	92.35 ± 2.61
	🟨	100.0 ± 0.00	100.0 ± 0.00	86.30 ± 10.34	88.31 ± 11.63	98.18 ± 2.80	99.00 ± 3.00	100.0 ± 0.00	98.50 ± 3.20	100.0 ± 0.00
	🟦	80.86 ± 7.85	88.62 ± 3.78	31.90 ± 5.17	40.10 ± 4.65	79.24 ± 3.76	83.79 ± 6.37	85.36 ± 3.30	84.28 ± 6.57	86.62 ± 5.90
	🟧	91.79 ± 4.89	90.33 ± 4.92	42.32 ± 6.27	50.78 ± 7.29	84.04 ± 4.62	90.59 ± 7.37	89.28 ± 5.67	90.96 ± 7.20	90.23 ± 7.45
	🟩	98.42 ± 2.03	99.25 ± 0.92	63.26 ± 6.25	68.26 ± 10.47	92.61 ± 7.07	98.45 ± 1.62	99.25 ± 1.26	98.25 ± 1.54	99.47 ± 0.88
	🟪	90.89 ± 6.28	95.67 ± 2.89	79.28 ± 4.95	78.79 ± 3.89	86.59 ± 2.98	89.86 ± 5.47	91.98 ± 3.07	89.93 ± 5.47	91.51 ± 4.71
	🟪	98.13 ± 1.94	97.85 ± 1.61	85.64 ± 4.07	84.50 ± 5.19	90.28 ± 5.56	98.60 ± 2.12	95.59 ± 14.26	98.63 ± 2.13	96.04 ± 3.62
	🟦	100.0 ± 0.00	100.0 ± 0.00	92.76 ± 6.54	92.92 ± 4.71	94.87 ± 5.16	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
	🟩	100.0 ± 0.00	99.95 ± 0.14	80.95 ± 3.24	85.99 ± 3.15	99.78 ± 0.38	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
	🟨	100.0 ± 0.00	100.0 ± 0.00	69.50 ± 20.91	87.00 ± 14.0	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
	🟪	87.75 ± 4.33	92.98 ± 5.08	61.58 ± 7.58	60.81 ± 5.43	86.82 ± 3.81	85.67 ± 6.03	88.69 ± 5.52	85.16 ± 5.71	86.96 ± 5.66
	🟦	92.51 ± 5.45	87.79 ± 3.82	55.30 ± 5.84	55.53 ± 4.01	78.82 ± 6.77	95.12 ± 5.21	91.03 ± 4.21	95.81 ± 5.08	95.73 ± 5.17
	⬛	86.31 ± 6.53	90.56 ± 2.33	39.49 ± 4.26	43.06 ± 7.25	83.10 ± 5.44	92.87 ± 4.82	90.57 ± 2.54	93.39 ± 4.86	90.71 ± 4.19
	🟩	99.93 ± 0.21	99.41 ± 1.42	93.22 ± 2.07	93.42 ± 3.11	97.46 ± 0.94	99.83 ± 0.37	99.88 ± 0.36	99.89 ± 0.34	100.0 ± 0.00
	🟫	97.95 ± 3.78	98.82 ± 1.83	76.00 ± 8.64	79.62 ± 8.09	89.93 ± 8.17	97.59 ± 2.11	98.81 ± 0.78	97.60 ± 2.12	96.94 ± 2.53
	🟩	95.59 ± 4.25	98.10 ± 2.92	33.66 ± 3.73	44.90 ± 7.45	89.82 ± 3.19	96.54 ± 3.66	99.10 ± 1.26	96.46 ± 3.59	98.74 ± 2.00
	🟨	99.13 ± 1.16	99.67 ± 0.66	96.30 ± 3.41	91.60 ± 5.44	97.58 ± 3.51	97.73 ± 2.27	99.18 ± 1.10	97.12 ± 2.39	98.94 ± 0.97

Table A5. Detailed semi-supervised classification results (mean ± standard deviation) on the Houston dataset (N = 25).

N	Class	EMP-CNN	MCNN-CP	LP	LapSVM	EMP-LapSVM	PL	AROC-DP	Mix-PL	Mix-PL-CL
25	OA (%)	92.05 ± 0.82	93.44 ± 0.99	79.86 ± 0.88	82.30 ± 1.04	86.52 ± 1.24	93.39 ± 1.33	93.48 ± 1.15	93.77 ± 0.95	94.18 ± 0.82
	AA (%)	92.86 ± 0.76	94.53 ± 0.98	80.37 ± 0.85	82.55 ± 1.18	87.54 ± 1.24	94.23 ± 1.28	94.43 ± 1.16	94.75 ± 0.89	94.98 ± 0.86
	K × 100	91.42 ± 0.89	92.91 ± 1.07	78.22 ± 0.94	80.86 ± 1.13	85.43 ± 1.34	92.86 ± 1.44	92.95 ± 1.24	93.27 ± 1.02	93.71 ± 0.89
	🟨	90.86 ± 5.39	91.90 ± 5.62	94.15 ± 4.56	94.42 ± 4.26	92.58 ± 4.72	93.05 ± 3.99	91.96 ± 5.11	91.43 ± 5.30	91.85 ± 4.73
	🟦	87.25 ± 8.23	97.15 ± 2.41	95.70 ± 1.64	94.38 ± 2.90	95.13 ± 2.34	87.23 ± 8.91	94.66 ± 5.64	85.33 ± 8.43	88.51 ± 7.70
	🟧	98.74 ± 0.96	99.33 ± 0.45	98.14 ± 1.30	97.27 ± 2.02	97.86 ± 2.19	99.53 ± 0.66	98.95 ± 0.66	99.71 ± 0.34	99.82 ± 0.19
	🟩	94.14 ± 2.28	98.65 ± 1.71	97.10 ± 2.69	95.96 ± 3.26	92.03 ± 3.28	95.36 ± 5.07	96.59 ± 2.91	97.46 ± 3.02	97.72 ± 2.33
	🟪	98.75 ± 1.13	99.94 ± 0.18	96.65 ± 1.24	96.62 ± 1.09	97.59 ± 1.75	99.61 ± 0.69	99.72 ± 0.75	99.82 ± 0.55	99.80 ± 0.55
	🟪	93.13 ± 5.02	98.07 ± 3.88	95.33 ± 2.76	93.12 ± 3.02	96.95 ± 3.07	95.54 ± 4.21	97.28 ± 3.87	96.94 ± 3.93	96.87 ± 4.00
	🟦	85.42 ± 3.02	85.81 ± 4.21	71.25 ± 5.58	77.62 ± 6.82	84.94 ± 3.26	91.86 ± 4.41	89.32 ± 1.67	91.64 ± 3.08	91.95 ± 3.21
	🟩	82.42 ± 3.06	79.82 ± 5.79	65.64 ± 4.19	65.37 ± 8.14	75.27 ± 4.68	79.65 ± 6.82	80.04 ± 6.81	83.18 ± 5.60	81.97 ± 5.40
	🟨	90.48 ± 3.76	87.50 ± 5.33	66.94 ± 3.89	74.87 ± 6.84	80.51 ± 3.80	92.66 ± 5.88	91.96 ± 4.00	94.98 ± 2.78	95.56 ± 3.08
	🟪	97.69 ± 2.59	96.90 ± 2.21	74.52 ± 3.93	80.49 ± 5.65	86.56 ± 6.10	99.07 ± 1.05	96.49 ± 7.75	98.09 ± 2.30	97.50 ± 3.95
	🟦	93.43 ± 3.20	96.39 ± 2.60	67.87 ± 3.59	72.00 ± 3.75	79.13 ± 3.20	96.05 ± 2.63	93.93 ± 3.98	96.15 ± 2.29	97.03 ± 1.99
	⬛	90.25 ± 4.98	89.27 ± 4.15	57.47 ± 5.50	62.59 ± 6.42	67.76 ± 7.25	89.80 ± 5.89	89.99 ± 4.89	89.10 ± 6.43	91.05 ± 5.10
	🟩	92.30 ± 3.86	97.23 ± 2.82	28.41 ± 5.02	39.92 ± 8.41	70.69 ± 4.74	94.21 ± 5.23	95.61 ± 3.49	97.43 ± 1.61	95.10 ± 4.54
	🟫	99.76 ± 0.50	100.0 ± 0.00	97.07 ± 2.29	95.32 ± 4.30	96.55 ± 2.41	99.92 ± 0.16	100.0 ± 0.00	100.0 ± 0.00	99.95 ± 0.15
	🟩	98.19 ± 2.70	100.0 ± 0.00	99.33 ± 0.72	98.33 ± 1.00	99.58 ± 0.59	100.0 ± 0.00	99.92 ± 0.19	99.98 ± 0.05	99.98 ± 0.05

Table A6. Detailed semi-supervised classification results (mean ± standard deviation) on the Salinas dataset (N = 25).

N	Class	EMP-CNN	MCNN-CP	LP	LapSVM	EMP-LapSVM	PL	AROC-DP	Mix-PL	CL-MixPL
25	OA (%)	94.95 ± 2.46	96.17 ± 0.98	84.13 ± 1.19	86.12 ± 1.96	91.93 ± 1.71	95.97 ± 2.25	96.18 ± 1.72	96.69 ± 0.71	97.00 ± 0.85
	AA (%)	98.24 ± 0.80	98.37 ± 0.37	91.91 ± 0.44	92.01 ± 1.01	95.18 ± 1.10	98.49 ± 0.82	98.63 ± 0.44	98.82 ± 0.22	98.91 ± 0.30
	K × 100	94.40 ± 2.70	95.64 ± 0.85	82.40 ± 1.30	84.59 ± 2.15	91.02 ± 1.91	95.53 ± 2.48	95.76 ± 1.34	96.33 ± 0.78	96.67 ± 0.94
	🟨	99.99 ± 0.03	100.0 ± 0.00	98.07 ± 1.04	97.04 ± 1.87	99.24 ± 0.60	99.89 ± 0.22	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
	🟦	98.79 ± 2.60	99.92 ± 0.19	99.56 ± 0.37	96.99 ± 1.59	97.95 ± 2.51	97.00 ± 5.46	99.52 ± 1.78	99.02 ± 1.39	98.67 ± 2.67
	🟧	99.97 ± 0.08	99.99 ± 0.02	95.57 ± 3.00	94.29 ± 3.21	99.59 ± 0.45	99.83 ± 0.27	99.37 ± 1.04	99.94 ± 0.17	99.92 ± 0.17
	🟩	99.89 ± 0.31	99.43 ± 0.42	98.96 ± 1.41	98.86 ± 0.86	99.02 ± 1.00	99.96 ± 0.13	99.94 ± 0.14	99.974 ± 0.10	99.96 ± 0.07
	🟪	99.13 ± 0.73	97.38 ± 2.04	95.41 ± 2.37	95.03 ± 2.01	96.25 ± 2.71	99.12 ± 1.04	99.10 ± 1.00	99.07 ± 1.02	99.50 ± 0.56
	🟪	100.0 ± 0.00	99.98 ± 0.12	99.36 ± 0.32	98.27 ± 0.97	98.60 ± 1.10	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
	🟦	99.80 ± 0.39	99.94 ± 0.12	99.38 ± 0.38	96.88 ± 3.71	97.74 ± 2.45	100.0 ± 0.00	99.55 ± 0.96	99.95 ± 0.05	100.0 ± 0.01
	🟩	79.71 ± 11.1	88.55 ± 3.07	58.83 ± 7.24	70.19 ± 9.31	82.59 ± 5.53	84.16 ± 9.52	84.89 ± 4.90	87.19 ± 2.88	89.03 ± 3.28
	🟨	99.93 ± 0.20	100.0 ± 0.00	95.61 ± 1.38	95.85 ± 2.14	97.98 ± 1.28	99.93 ± 0.20	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00
	🟪	99.98 ± 0.06	99.09 ± 0.98	86.46 ± 2.36	85.37 ± 4.36	95.64 ± 2.34	99.95 ± 0.09	99.85 ± 0.17	99.88 ± 0.17	99.99 ± 0.02
	🟦	99.85 ± 0.16	99.97 ± 0.06	93.51 ± 2.25	93.10 ± 2.59	96.05 ± 2.69	99.86 ± 0.14	99.88 ± 0.10	99.91 ± 0.13	99.88 ± 0.14
	⬛	99.95 ± 0.12	99.78 ± 0.52	99.56 ± 0.53	99.47 ± 1.02	100.0 ± 0.00	100.0 ± 0.00	100.0 ± 0.00	99.99 ± 0.02	100.0 ± 0.00
	🟩	99.73 ± 0.60	100.0 ± 0.00	96.89 ± 1.95	97.71 ± 1.65	97.77 ± 1.52	99.99 ± 0.03	99.89 ± 0.23	99.93 ± 0.12	99.99 ± 0.03
	🟫	99.87 ± 0.21	99.70 ± 0.39	92.95 ± 2.80	92.96 ± 3.66	92.91 ± 3.26	99.45 ± 1.06	99.88 ± 0.22	99.86 ± 0.27	99.89 ± 0.26
	🟩	95.31 ± 2.43	91.35 ± 5.08	62.89 ± 5.20	64.97 ± 7.42	79.42 ± 4.63	96.80 ± 3.55	96.26 ± 2.58	99.37 ± 2.05	95.76 ± 3.13
	🟨	99.95 ± 0.13	98.86 ± 1.01	97.52 ± 1.41	95.14 ± 3.05	92.04 ± 5.66	99.92 ± 0.24	99.94 ± 0.14	99.99 ± 0.02	99.95 ± 0.09

References

1. Gevaert, C.M.; Suomalainen, J.; Tang, J.; Kooistra, L. Generation of spectral–temporal response surfaces by combining multispectral satellite and hyperspectral UAV imagery for precision agriculture applications. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2015**, *8*, 3140–3146. [CrossRef]
2. Murphy, R.J.; Schneider, S.; Monteiro, S.T. Consistency of Measurements of Wavelength Position from Hyperspectral Imagery: Use of the Ferric Iron Crystal Field Absorption at ~900 nm as an Indicator of Mineralogy. *IEEE Trans. Geosci. Remote Sens.* **2013**, *52*, 2843–2857. [CrossRef]
3. Koz, A. Ground-Based Hyperspectral Image Surveillance Systems for Explosive Detection: Part I—State of the Art and Challenges. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 4746–4753. [CrossRef]
4. Qu, D.-X.; Berry, J.; Calta, N.P.; Crumb, M.F.; Guss, G.; Matthews, M.J. Temperature Measurement of Laser-Irradiated Metals Using Hyperspectral Imaging. *Phys. Rev. Appl.* **2020**, *14*, 014031. [CrossRef]
5. Berné, O.; Helens, A.; Pilleri, P.; Joblin, C. Non-negative matrix factorization pansharpening of hyperspectral data: An application to mid-infrared astronomy. In Proceedings of the 2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, Reykjavìk, Iceland, 14–16 June 2010; pp. 1–4.

6. Cheng, J.-H.; Sun, D.-W.; Pu, H.; Zhu, Z. Development of hyperspectral imaging coupled with chemometric analysis to monitor K value for evaluation of chemical spoilage in fish fillets. *Food Chem.* **2015**, *185*, 245–253. [CrossRef]
7. Chang, C.-I. *Hyperspectral Data Exploitation: Theory and Applications*; John Wiley & Sons: Hoboken, NJ, USA, 2007.
8. Bioucas-Dias, J.M.; Plaza, A.; Camps-Valls, G.; Scheunders, P.; Nasrabadi, N.; Chanussot, J. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geosci. Remote Sens. Mag.* **2013**, *1*, 6–36. [CrossRef]
9. Ghamisi, P.; Yokoya, N.; Li, J.; Liao, W.; Liu, S.; Plaza, J.; Rasti, B.; Plaza, A. Advances in hyperspectral image and signal processing: A comprehensive overview of the state of the art. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 37–78. [CrossRef]
10. Fauvel, M.; Tarabalka, Y.; Benediktsson, J.A.; Chanussot, J.; Tilton, J.C. Advances in spectral-spatial classification of hyperspectral images. *Proc. IEEE* **2012**, *101*, 652–675. [CrossRef]
11. Ghamisi, P.; Plaza, J.; Chen, Y.; Li, J.; Plaza, A.J. Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geosci. Remote Sens. Mag.* **2017**, *5*, 8–32. [CrossRef]
12. He, L.; Li, J.; Liu, C.; Li, S. Recent advances on spectral–spatial hyperspectral image classification: An overview and new guidelines. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 1579–1597. [CrossRef]
13. Sowmya, V.; Soman, K.; Hassaballah, M. Hyperspectral image: Fundamentals and advances. In *Recent Advances in Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 401–424.
14. Melgani, F.; Bruzzone, L. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Trans. Geosci. Remote Sens.* **2004**, *42*, 1778–1790. [CrossRef]
15. Benediktsson, J.A.; Palmason, J.A.; Sveinsson, J.R. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Trans. Geosci. Remote Sens.* **2005**, *43*, 480–491. [CrossRef]
16. Dalla Mura, M.; Villa, A.; Benediktsson, J.A.; Chanussot, J.; Bruzzone, L. Classification of hyperspectral images by using extended morphological attribute profiles and independent component analysis. *IEEE Geosci. Remote Sens. Lett.* **2010**, *8*, 542–546. [CrossRef]
17. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification using dictionary-based sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2011**, *49*, 3973–3985. [CrossRef]
18. Huang, S.; Zhang, H.; Pižurica, A. A robust sparse representation model for hyperspectral image classification. *Sensors* **2017**, *17*, 2087. [CrossRef] [PubMed]
19. Li, S.; Song, W.; Fang, L.; Chen, Y.; Ghamisi, P.; Benediktsson, J.A. Deep learning for hyperspectral image classification: An overview. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 6690–6709. [CrossRef]
20. Mou, L.; Ghamisi, P.; Zhu, X.X. Deep recurrent neural networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3639–3655. [CrossRef]
21. Chen, Y.; Lin, Z.; Zhao, X.; Wang, G.; Gu, Y. Deep learning-based classification of hyperspectral data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2094–2107. [CrossRef]
22. Zhao, W.; Du, S. Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 4544–4554. [CrossRef]
23. Haut, J.M.; Paoletti, M.E.; Plaza, J.; Li, J.; Plaza, A. Active learning with convolutional neural networks for hyperspectral image classification using a new bayesian approach. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 6440–6461. [CrossRef]
24. Jiang, J.; Ma, J.; Wang, Z.; Chen, C.; Liu, X. Hyperspectral image classification in the presence of noisy labels. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 851–865. [CrossRef]
25. Zhou, Z.-H. A brief introduction to weakly supervised learning. *Natl. Sci. Rev.* **2018**, *5*, 44–53. [CrossRef]
26. Leng, Q.; Yang, H.; Jiang, J. Label noise cleansing with sparse graph for hyperspectral image classification. *Remote Sens.* **2019**, *11*, 1116. [CrossRef]
27. Zhang, X.; Song, Q.; Liu, R.; Wang, W.; Jiao, L. Modified co-training with spectral and spatial views for semisupervised hyperspectral image classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2014**, *7*, 2044–2055. [CrossRef]
28. Yang, L.; Yang, S.; Jin, P.; Zhang, R. Semi-supervised hyperspectral image classification using spatio-spectral Laplacian support vector machine. *IEEE Geosci. Remote Sens. Lett.* **2013**, *11*, 651–655. [CrossRef]
29. Yu, X.; Liu, T.; Gong, M.; Tao, D. Learning with biased complementary labels. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 68–83.
30. Feng, L.; Kaneko, T.; Han, B.; Niu, G.; An, B.; Sugiyama, M. Learning with multiple complementary labels. In Proceedings of the International Conference on Machine Learning, Online, 13–18 July 2020; pp. 3072–3081.
31. Yuan, Y.; Wang, C.; Jiang, Z. Proxy-Based Deep Learning Framework for Spectral-Spatial Hyperspectral Image Classification: Efficient and Robust. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5501115. [CrossRef]
32. Paoletti, M.E.; Haut, J.M.; Fernandez-Beltran, R.; Plaza, J.; Plaza, A.; Li, J.; Pla, F. Capsule networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 2145–2160. [CrossRef]
33. Yu, C.; Han, R.; Song, M.; Liu, C.; Chang, C.-I. A simplified 2D-3D CNN architecture for hyperspectral image classification based on spatial–spectral fusion. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 2485–2501. [CrossRef]
34. Alam, F.I.; Zhou, J.; Liew, A.W.-C.; Jia, X.; Chanussot, J.; Gao, Y. Conditional random field and deep feature learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *57*, 1612–1628. [CrossRef]
35. Yu, C.; Zhao, M.; Song, M.; Wang, Y.; Li, F.; Han, R.; Chang, C.-I. Hyperspectral image classification method based on CNN architecture embedding with hashing semantic feature. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 1866–1881. [CrossRef]

36. Lee, H.; Kwon, H. Going deeper with contextual CNN for hyperspectral image classification. *IEEE Trans. Image Process.* **2017**, *26*, 4843–4855. [CrossRef]
37. Li, W.; Wu, G.; Zhang, F.; Du, Q. Hyperspectral image classification using deep pixel-pair features. *IEEE Trans. Geosci. Remote Sens.* **2016**, *55*, 844–853. [CrossRef]
38. Zhong, Z.; Li, J.; Luo, Z.; Chapman, M. Spectral–spatial residual network for hyperspectral image classification: A 3-D deep learning framework. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 847–858. [CrossRef]
39. He, X.; Chen, Y.; Ghamisi, P. Heterogeneous transfer learning for hyperspectral image classification based on convolutional neural network. *IEEE Trans. Geosci. Remote Sens.* **2019**, *58*, 3246–3263. [CrossRef]
40. Fang, L.; Zhao, W.; He, N.; Zhu, J. Multiscale CNNs Ensemble Based Self-Learning for Hyperspectral Image Classification. *IEEE Geosci. Remote Sens. Lett.* **2020**, *17*, 1593–1597. [CrossRef]
41. Gao, K.; Liu, B.; Yu, X.; Qin, J.; Zhang, P.; Tan, X. Deep relation network for hyperspectral image few-shot classification. *Remote Sens.* **2020**, *12*, 923. [CrossRef]
42. Roy, S.K.; Mondal, R.; Paoletti, M.E.; Haut, J.M.; Plaza, A. Morphological Convolutional Neural Networks for Hyperspectral Image Classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2021**, *14*, 8689–8702. [CrossRef]
43. Aptoula, E.; Ozdemir, M.C.; Yanikoglu, B. Deep learning with attribute profiles for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2016**, *13*, 1970–1974. [CrossRef]
44. He, X.; Chen, Y.; Lin, Z. Spatial-Spectral Transformer for Hyperspectral Image Classification. *Remote Sens.* **2021**, *13*, 498. [CrossRef]
45. Cheng, L.; Zhou, X.; Zhao, L.; Li, D.; Shang, H.; Zheng, Y.; Pan, P.; Xu, Y. Weakly supervised learning with side information for noisy labeled images. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 306–321.
46. Zhang, Z.; Sabuncu, M.R. Generalized cross entropy loss for training deep neural networks with noisy labels. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS), Montréal, QC, Canada, 3–8 December 2018.
47. Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I.; Sugiyama, M. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Proceedings of the 32nd Conference on Neural Information Processing Systems (NeurIPS), Montréal, QC, Canada, 3–8 December 2018; pp. 8527–8537.
48. Yu, X.; Han, B.; Yao, J.; Niu, G.; Tsang, I.; Sugiyama, M. How does disagreement help generalization against label corruption? In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; pp. 7164–7173.
49. Wei, H.; Feng, L.; Chen, X.; An, B. Combating noisy labels by agreement: A joint training method with co-regularization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 13726–13735.
50. Zhu, X.; Goldberg, A.B. Introduction to semi-supervised learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **2009**, *3*, 1–130. [CrossRef]
51. Wang, C.-P.; Zhang, J.-S.; Du, F.; Shi, G. Symmetric low-rank representation with adaptive distance penalty for semi-supervised learning. *Neurocomputing* **2018**, *316*, 376–385. [CrossRef]
52. Dornaika, F.; Weng, L. Sparse graphs with smoothness constraints: Application to dimensionality reduction and semi-supervised classification. *Pattern Recognit.* **2019**, *95*, 285–295. [CrossRef]
53. Lee, D.-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In Proceedings of the Workshop on Challenges in Representation Learning, ICML, Atlanta, GA, USA, 16–21 June 2013; p. 896.
54. Iscen, A.; Tolias, G.; Avrithis, Y.; Chum, O. Label propagation for deep semi-supervised learning. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 5070–5079.
55. Bahraini, T.; Azimpour, P.; Yazdi, H.S. Modified-mean-shift-based noisy label detection for hyperspectral image classification. *Comput. Geosci.* **2021**, *155*, 104843. [CrossRef]
56. Tu, B.; Zhou, C.; He, D.; Huang, S.; Plaza, A. Hyperspectral classification with noisy label detection via superpixel-to-pixel weighting distance. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 4116–4131. [CrossRef]
57. Kang, X.; Duan, P.; Xiang, X.; Li, S.; Benediktsson, J.A. Detection and correction of mislabeled training samples for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 5673–5686. [CrossRef]
58. Tu, B.; Zhang, X.; Kang, X.; Wang, J.; Benediktsson, J.A. Spatial density peak clustering for hyperspectral image classification with noisy labels. *IEEE Trans. Geosci. Remote Sens.* **2019**, *57*, 5085–5097. [CrossRef]
59. Camps-Valls, G.; Marsheva, T.V.B.; Zhou, D. Semi-supervised graph-based hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2007**, *45*, 3044–3054. [CrossRef]
60. Wang, L.; Hao, S.; Wang, Q.; Wang, Y. Semi-supervised classification for hyperspectral imagery based on spatial-spectral label propagation. *ISPRS J. Photogramm. Remote Sens.* **2014**, *97*, 123–137. [CrossRef]
61. Riese, F.M.; Keller, S.; Hinz, S. Supervised and semi-supervised self-organizing maps for regression and classification focusing on hyperspectral data. *Remote Sens.* **2020**, *12*, 7. [CrossRef]
62. Tan, K.; Zhu, J.; Du, Q.; Wu, L.; Du, P. A novel tri-training technique for semi-supervised classification of hyperspectral images based on diversity measurement. *Remote Sens.* **2016**, *8*, 749. [CrossRef]
63. Zhang, Y.; Liu, K.; Dong, Y.; Wu, K.; Hu, X. Semisupervised classification based on SLIC segmentation for hyperspectral image. *IEEE Geosci. Remote Sens. Lett.* **2019**, *17*, 1440–1444. [CrossRef]

64. Ji, X.; Cui, Y.; Wang, H.; Teng, L.; Wang, L.; Wang, L. Semisupervised hyperspectral image classification using spatial-spectral information and landscape features. *IEEE Access* **2019**, *7*, 146675–146692. [CrossRef]
65. Fang, B.; Li, Y.; Zhang, H.; Chan, J.C.-W. Collaborative learning of lightweight convolutional neural network and deep clustering for hyperspectral image semi-supervised classification with limited training samples. *ISPRS J. Photogramm. Remote Sens.* **2020**, *161*, 164–178. [CrossRef]
66. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. *arXiv* **2017**, arXiv:1710.09412.
67. Zheng, J.; Feng, Y.; Bai, C.; Zhang, J. Hyperspectral image classification using mixed convolutions and covariance pooling. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 522–534. [CrossRef]