

Article

## A Dimension Reduction Framework for HSI Classification Using Fuzzy and Kernel NFLE Transformation

Ying-Nong Chen <sup>1</sup>, Cheng-Ta Hsieh <sup>1</sup>, Ming-Gang Wen <sup>2</sup>, Chin-Chuan Han <sup>3,\*</sup> and Kuo-Chin Fan <sup>1</sup>

<sup>1</sup> Department of Computer Science and Information Engineering, National Central University, Taoyuan 32001, Taiwan; E-Mails: yingnong1218@gmail.com (Y.-N.C.); hsieh.chengta@gmail.com (C.-T.H.); kcfan@csie.ncu.edu.tw (K.-C.F.)

<sup>2</sup> Department of Information Management, National United University, Miaoli 36063, Taiwan; E-Mail: mgwen@nuu.edu.tw

<sup>3</sup> Department of Computer Science and Information Engineering, National United University, Miaoli 36063, Taiwan

\* Author to whom correspondence should be addressed; E-Mail: cchan@nuu.edu.tw; Tel.: +886-37-382610.

Academic Editors: Yuei-An Liou, Chyi-Tyi Lee, Yaoming Ma, Takashi Oguchi, Indrajeet Chaubey, Giles M. Foody and Prasad S. Thenkabail

Received: 23 May 2015 / Accepted: 22 October 2015 / Published: 29 October 2015

---

**Abstract:** In this paper, a general nearest feature line (NFL) embedding (NFLE) transformation called fuzzy-kernel NFLE (FKNFLE) is proposed for hyperspectral image (HSI) classification in which kernelization and fuzzification are simultaneously considered. Though NFLE has successfully demonstrated its discriminative capability, the non-linear manifold structure cannot be structured more efficiently by linear scatters using the linear NFLE method. According to the proposed scheme, samples were projected into a kernel space and assigned larger weights based on that of their neighbors. The within-class and between-class scatters were calculated using the fuzzy weights, and the best transformation was obtained by maximizing the Fisher criterion in the kernel space. In that way, the kernelized manifold learning preserved the local manifold structure in a Hilbert space as well as the locality of the manifold structure in the reduced low-dimensional space. The proposed method was compared with various state-of-the-art methods to evaluate the performance using three benchmark data sets. Based on the experimental results: the proposed FKNFLE outperformed the other, more conventional methods.

**Keywords:** hyperspectral image classification; manifold learning; nearest feature line embedding; kernelization; fuzzification

---

## 1. Introduction

Dimensionality reduction (DR) in hyperspectral image (HSI) classification is a critical issue during data analysis because most multispectral, hyperspectral, and ultraspectral images generate high-dimensional spectral images with abundant spectral bands and data. However, it is challenging to classify these spectral data because a vast amount of samples has to be collected for training beforehand. Besides, the spectral properties of land covers are too similar to clearly separate them. Hence, an effective DR is an essential step to extract the salient features for classification.

Recently, a number of DR methods have been proposed that can be classified into three categories: linear analysis, manifold learning, and kernelization. Those using linear analysis try to model the linear variation of samples and find a transformation to maximize or minimize the scatter matrix, e.g., principal component analysis (PCA) [1], linear discriminant analysis (LDA) [2], and discriminant common vectors (DCV) [3]. Sample scatters are represented in the global Euclidean structure in these methods. They work well for DR or classification if samples are linearly separated or are distributed in a Gaussian function. However, when samples are distributed in a manifold structure, the local structure of a sample in a high-dimensional space is not apparent when using global measurement. In addition, the classification performance in the case of linear analysis methods would deteriorate when the decision boundaries are predominantly nonlinear [4]. Manifold learning methods have been proposed to reveal the local structure of samples. He *et al.* [5] propose the locality preserving projection (LPP) method to preserve the local structure of training samples for face recognition. Since LPP presents sample scatter using the relationship between neighbors, the local manifold structure is preserved and the performance is more effective than in the case of the linear analysis methods. Tu *et al.* [6] used the Laplacian eigenmap (LE) method for land cover classification using polarimetric synthetic aperture radar data. The LE algorithm reduces the dimensions of features from a high-dimensional polarimetric manifold space to an intrinsic low-dimensional manifold space. Wang and He [7] investigated the LPP for DR in HSI classification. Kim *et al.* [8] utilized the locally linear embedding (LLE) method to reduce the dimensionality of HSIs. Li *et al.* [9,10] used the local Fisher discriminant analysis (LFDA) method which integrates the properties of LDA and LPP to reduce the dimensionality of HSI data. Luo *et al.* [11] propose a discriminative and supervised neighborhood preserving embedding (NPE) method for feature extraction in HSI classification. Zhang *et al.* [12] propose a manifold regularized sparse low-rank approximation, which treats the hyperspectral image as a data cube for HSI classification. These manifold learning methods all preserve the local structure of samples and improve on the performance of conventional linear analysis methods. However, according to Boots and Gordon [13], the applicability of linear manifold learning is limited to noises. Generally, the discriminative salient features of training samples are extracted using certain evaluation processes. An appropriate kernel function could improve the performance for the given method [14]. The kernelization approaches have been proposed for improving the performance of HSI classification. Boots and Gordon [13] introduced a kernelization

method to alleviate the limitation of manifold learning. Scholkopf *et al.* [15] propose a kernel PCA (KPCA) method for nonlinear DR. KPCA generates a high-dimensional Hilbert space to extract the nonlinear structure that is missed by PCA. Furthermore, Lin *et al.* [16] propose a general framework for multiple kernel learning during DR. They unify the multiple kernel representation, and the multiple feature representations of data are consequently revealed in a low dimension. On the other hand, a composite kernel scheme, which is a linear combination of multiple kernels, extracts both spectral and spatial data [17]. Chen *et al.* [18] present a sparse representation of kernels for HSI classification. A query sample is represented via all training samples in an induced kernel space. Moreover, pixels within a local neighborhood are also represented by the combination of training samples. Similar to the idea of multiple kernels, Zhang *et al.* [19] proposed a multiple-features combination method for HSI classification, which combined spectral, texture, and shape features to increase the HSI classification performance.

In the previous works, the nearest feature line (NFL) strategy was embedded in the linear transformation for dimension reduction on face recognition [20] and HSI classification [21]. However, the nonlinear and non-Euclidean structures were not efficiently extracted using the linear transformation. Fuzzification and kernelization are two efficient tools for enhancement in nonlinear spaces. The fuzzy methodology was further adopted in previous work [26]. In this study, a general NFLE transformation, called fuzzy-kernel NFLE, was extended for feature extraction in which kernelization and fuzzification were simultaneously considered. In addition, more experimental analysis was conducted in this study. Three benchmark data sets were evaluated in this work instead of one set as in [26]. The proposed method was compared with state-of-the-art algorithms for performance evaluation.

The rest of this paper is organized as follows: Some related works are reviewed in Section 2. In Section 3, the kernelization and fuzzification strategies are introduced and incorporated into the NFLE algorithm. Several experiments were conducted to show the effectiveness of the proposed method as reported in Section 4. Furthermore, the comparisons with several state-of-the-art HSI classification methods are given. Finally, conclusions are given in Section 5.

## 2. Related Works

In this study, three approaches, nearest feature line embedding (NFLE) [20,21], kernelization [15], and fuzzy  $k$  nearest neighbor (FKNN) [22], were considered to reduce the feature dimensions for HSI classification. Before the proposed methods, brief reviews of NFLE and kernelization methods are presented in the following: given  $N$   $d$ -dimensional training samples  $X = [x_1, x_2, \dots, x_N] \in R^{d \times N}$  consisting of  $N_C$  land-cover classes  $C_1, C_2, \dots, C_{N_C}$ . The new samples in a low-dimensional space were obtained by the linear projection  $y_i = w^T x_i$ , where  $w$  is a found linear projection matrix for DR.

### 2.1. Nearest Feature Line Embedding (NFLE)

NFLE is a linear transformation for DR. The sample scatters are represented in a Laplacian matrix form by using the point-to-line strategy which originated from the nearest linear combination (NLC) approach [23]. The objective function is defined and minimized as follows:

$$\begin{aligned}
 O &= \sum_i \left( \sum_{i \neq m \neq n} \|y_i - L_{m,n}(y_i)\|^2 l_{m,n}(y_i) \right) \\
 &= \sum_i \left\| y_i - \sum_j M_{i,j} y_j \right\|^2 \\
 &= \text{tr} \left( Y(I - M)^T (I - M) Y \right) = \text{tr} \left( w^T X(D - W) X^T w \right) \\
 &= \text{tr} \left( w^T X L X^T w \right).
 \end{aligned} \tag{1}$$

Here, point  $L_{m,n}(y_i)$  is a projection point on line  $L_{m,n}$  for point  $y_i$ , and weight  $l_{m,n}(y_i)$  (being 1 or 0) represents the connectivity relationship from point  $y_i$  to a feature line  $L_{m,n}$  that passes through two points  $y_m$  and  $y_n$ . The projection point  $L_{m,n}(y_i)$  is represented as a linear combination of points  $y_m$  and  $y_n$ :  $L_{m,n}(y_i) = y_m + t_{m,n}(y_n - y_m)$ , in which  $t_{m,n} = (y_i - y_m)^T (y_m - y_n) / (y_m - y_n)^T (y_m - y_n)$ , and  $i \neq m \neq n$ . Using simple algebra operations, the discriminant vector from point  $y_i$  to the projection point  $L_{m,n}(y_i)$  can be represented as  $y_i - \sum_j M_{i,j} y_j$ , in which two values in the  $i$ th row in matrix  $M$  are set as  $M_{i,m} = t_{n,m}$ ,  $M_{i,n} = t_{m,n}$ , and  $t_{n,m} + t_{m,n} = 1$ , when weight  $l_{m,n}(y_i) = 1$ . The other values in the  $i$ th row are set as zero, if  $j \neq m \neq n$ . The mean squared distance in Equation (1) for all training points to their NFLs is next obtained as  $\text{tr}(w^T X L X^T w)$ , in which  $L = D - W$ , and matrix  $D$  is a matrix of the column sums of the similarity matrix  $W$ . From the results of Yan *et al.* [24], matrix  $W$  is defined as  $W_{i,j} = (M + M^T - M^T M)_{i,j}$  when  $i \neq j$ , and is zero otherwise;  $\sum_j M_{i,j} y_j = 1$ . Matrix  $L$  in Equation (1) is represented as a Laplacian matrix. For more details, refer to [20,21].

Considering the class labels in supervised classification, two parameters  $K_1$  and  $K_2$  are manually determined in calculating the within-class scatter  $S_w$  and the between-class scatter  $S_b$ , respectively:

$$S_w = \sum_{k=1}^{N_C} \left( \sum_{x_i \in C_k} \sum_{L_{m,n} \in F_{K_1}(x_i, C_k)} (x_i - L_{m,n}(x_i))(x_i - L_{m,n}(x_i))^T \right), \text{ and} \tag{2}$$

$$S_b = \sum_{k=1}^{N_C} \left( \sum_{x_i \in C_k} \sum_{l=1, l \neq k}^{N_C} \sum_{L_{m,n} \in F_{K_2}(x_i, C_l)} (x_i - L_{m,n}(x_i))(x_i - L_{m,n}(x_i))^T \right) \tag{3}$$

$F_{K_1}(x_i, C_k)$  indicates the set of  $K_1$  NFLs within the same class,  $C_k$ , of point  $x_i$ , *i.e.*,  $l_{m,n}(y_i) = 1$ , and  $F_{K_2}(x_i, C_l)$  is a set of  $K_2$  NFLs belonging to the different classes of point  $x_i$ . The Fisher criterion  $\text{tr}(w^T S_b w / w^T S_w w)$  is then maximized to find the projection matrix  $w$ , which is composed of the eigenvectors with the corresponding largest eigenvalues. A new sample in the low-dimensional space can be obtained by the linear projection  $y = w^T x$ , and the nearest neighbor (one-NN) matching rule is applied for template matching.

### 2.2. Kernelization of LDA

In kernel LDA, considering the nonlinear mapping function from a space  $X$  to a Hilbert space  $H$ ,  $\phi: x \in X \rightarrow \phi(x) \in H$ , the within-class and between-class scatter in space  $H$  are calculated as

$$\mathbf{S}_w^\phi = \sum_{k=1}^{N_c} \left( \sum_{x_i \in C_k} (\phi(x_i) - \bar{\phi}_k)(\phi(x_i) - \bar{\phi}_k)^T \right), \text{ and} \quad (4)$$

$$\mathbf{S}_b^\phi = \sum_{k=1}^{N_c} (\bar{\phi}_k - \bar{\phi})(\bar{\phi}_k - \bar{\phi})^T \quad (5)$$

Here,  $\bar{\phi}_k = \frac{1}{n_k} \sum_{i=1}^{n_k} \phi(x_i)$  and  $\bar{\phi} = \frac{1}{N} \sum_{i=1}^N \phi(x_i)$  represent the class mean and the population mean in space  $H$ , respectively. To generalize LDA to the nonlinear case, the dot product trick is exclusively used. The expression of dot product on the Hilbert space  $H$  is given by the following kernel function:  $k(x_i, x_j) = k_{i,j} = \phi^T(x_i)\phi(x_j)$ . Let the symmetric matrix  $K$  of  $N$  by  $N$  be a matrix composed of dot product in feature space  $H$ , i.e.,  $K(x_i, x_j) = \langle \phi(x_i) \cdot \phi(x_j) \rangle = (k_{i,j})$  and,  $i, j = 1, 2, \dots, N$ . The kernel operator  $K$  makes it possible for the construction of the linear separating function in space  $H$  to be equivalent to that of the nonlinear separating function in space  $X$ . Kernel LDA also maximizes the between-class scatter and minimizes the within-class scatter, i.e.,  $\max(w^T S_b^\phi w / w^T S_w^\phi w)$ . This maximization is equivalent to the following eigenvector resolution:  $\lambda S_w^\phi w = S_b^\phi w$ . There is a set of coefficients  $\alpha$  for  $w = \sum_{i=1}^N \alpha_i \phi(x_i)$  such that the largest eigenvalue gives the maximum of the scatter quotient  $\lambda = w^T S_b^\phi w / w^T S_w^\phi w$ .

### 3. Fuzzy Kernel Nearest Feature Line Embedding (FKNFLE)

According to the analyses above, a training DR scheme effectively extracts the discriminant features from the non-Euclidean and non-linear space. To this end, fuzzy kernel nearest feature line embedding (FKNFLE) is proposed for HSI classification. The idea of FKNFLE is to incorporate the fuzziness and kernelization into the manifold learning method. The kernel function not only generates a non-linear feature space for discriminant analysis, but also increases the robustness to noise during the training phase. Manifold learning methods preserve the local structure of samples in the Hilbert space. On the other hand, the fuzzy kernel nearest neighbor method extracts the non-Euclidean structures of training samples to enhance discriminative capability. NFLE has been successfully applied in HSI classification. Noise variations and high-degree non-linear data distributions limit the performance of manifold learning. A kernel trick is used to alleviate this problem as introduced in the following.

#### 3.1. Kernelization of NFLE

The kernelization function adopted in this study was inspired by that in [15]. Let  $\phi: x \in X \rightarrow \phi(x) \in H$  be a nonlinear mapping from a low-dimensional space to a high-dimensional Hilbert space  $H$ . The mean squared distance for all training points to their NFLs in the Hilbert space is written as follows:

$$\begin{aligned}
 \sum_i \|\phi(y_i) - L_{m,n}(\phi(y_i))\|^{(2)} &= \sum_i \left\| \phi(y_i) - \sum_j M_{i,j} \phi(y_j) \right\|^2 \\
 &= \text{tr}(\phi^T(Y)(I - M)^T(I - M)\phi(Y)) \\
 &= \text{tr}(\phi^T(Y)(D - W)\phi(Y)) \\
 &= \text{tr}(w^T \phi(X)L\phi^T(X)w).
 \end{aligned} \tag{6}$$

Then, the object function in Equation (6) is minimized and expressed as a Laplacian matrix. The eigenvector problem of kernel NFLE in the Hilbert space is expressed as:

$$[\phi(X)L\phi^T(X)]w = \lambda[\phi(X)D\phi^T(X)]w \tag{7}$$

To extend NFLE to its kernel version, the implicit feature vector,  $\phi(x)$ , does not need to be obtained explicitly. The dot product expression of two samples is exclusively applied in the Hilbert space with a kernel function as follows:  $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$ . The eigenvectors of Equation (7) are represented by the linear combinations  $\phi(x_1), \phi(x_2), \dots, \phi(x_N)$ . The coefficient  $\alpha_i$  is  $w = \sum_{i=1}^N \alpha_i \phi(x_i) = \phi(X)\alpha$  where  $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_N]^T \in R^N$ . Then, the eigenvector problem is as follows:

$$K L K \alpha = \lambda K D K \alpha. \tag{8}$$

Let the coefficient vectors,  $\alpha^1, \alpha^2, \dots, \alpha^N$ , be the solutions of Equation (8) in a column format. Given a testing point,  $z$ , the projections onto the eigenvectors,  $w^k$ , are obtained as follows:

$$(w^k \cdot \phi(z)) = \sum_{i=1}^N \alpha_i^k \langle \phi(z), \phi(x_i) \rangle = \sum_{i=1}^N \alpha_i^k K(z, x_i), \tag{9}$$

where  $\alpha_i^k$  is the  $i$ th element of the coefficient vector,  $\alpha^k$ . The kernel function RBF (radial basis function) is used in this study. Thus, the within-class and between-class scatters in a kernel space are defined as follows:

$$\mathbf{S}_w^\phi = \sum_{k=1}^{N_c} \left( \sum_{\phi(x_i) \in C_k} \sum_{L_{m,n} \in F_{K_1}(\phi(x_i), C_k)} (\phi(x_i) - L_{m,n}(\phi(x_i))) (\phi(x_i) - L_{m,n}(\phi(x_i)))^T \right), \text{ and} \tag{10}$$

$$\mathbf{S}_b^\phi = \sum_{k=1}^{N_c} \left( \sum_{\phi(x_i) \in C_k} \sum_{l=1, l \neq k}^{N_c} \sum_{L_{m,n} \in F_{K_2}(\phi(x_i), C_l)} (\phi(x_i) - L_{m,n}(\phi(x_i))) (\phi(x_i) - L_{m,n}(\phi(x_i)))^T \right). \tag{11}$$

The kernelized manifold learning preserves the non-linear local structure in a Hilbert space. The distances in the NFLE approach are calculated by the Euclidean distance-based measurement. On the other hand, the non-Euclidean structure of training samples can be further extracted by fuzzification. The FKNN algorithm [22] enhances the discriminant power among samples by assigning the higher membership grades to the samples whose neighbors are within the same class. By doing so, the non-Euclidean structures are extracted, and the discriminative power of samples can be enhanced.

3.2. Fuzzification of NFLE

Consider  $N$  samples in the reduced space  $Y = [y_1, y_2, \dots, y_N]$  and their corresponding fuzzy membership grades,  $\pi(y_i)$ , for each sample,  $y_i$ . The objective function is re-defined as follows:

$$\begin{aligned}
 O &= \sum_i \pi(y_i) \left( \sum_{i \neq m \neq n} \|y_i - L_{m,n}(y_i)\|^2 l_{m,n}(y_i) \right) \\
 &= \sum_i \pi(y_i) \left\| y_i - \sum_j M_{i,j} y_j \right\|^2 \\
 &= \text{tr} \left( Y^T (FEI - FEM)^T (FEI - FEM) Y \right) \\
 &= \text{tr} \left( Y^T (FED - FEW) Y \right) \\
 &= \text{tr} \left( Y^T (D_{fuzzy} - W_{fuzzy}) Y \right) \\
 &= \text{tr} \left( w^T X L_{fuzzy} X^T w \right)
 \end{aligned} \tag{12}$$

Here, each sample is assigned a fuzzy grade,  $\pi(y_i)$ . Element  $M_{i,j}$  denotes the connectivity relationship between point  $y_i$  and line  $L_{m,n}$  which is the same as that in Equation (1). Two non-zero terms,  $M_{i,n} = t_{m,n}$  and  $M_{i,m} = t_{n,m}$ , are set, and  $\sum_j M_{i,j} = 1$ . Using simple algebra operations, the objective function with fuzzification is represented in a Laplacian matrix in which the fuzzy terms,  $\pi(y_i)$ , constitute the column vector,  $F$ , with size  $N \times 1$ , and  $E$  is a row vector of all those with size  $1 \times N$ .

Similarly, given  $N$  samples  $\phi(X) = \{\phi(x_1), \phi(x_2), \dots, \phi(x_N)\}$  in a Hilbert space, the membership grade of a specified sample,  $\phi(x_i)$ , and its  $K_3$  neighbors, is designed in the following equation for computing the within-class scatter:

$$\pi(x_i) = \begin{cases} 0.51 + (0.49 * (q_i / K_3)), & \text{if } q_i \geq \theta_{within}; \\ 0.49 * (q_i / K_3) & \text{otherwise.} \end{cases} \tag{13}$$

Here, value  $q_i$  is the number of samples whose labels are the same as that of  $\phi(x_i)$  among  $K_3$  nearest neighbors, and  $\theta_{within}$  is a manual threshold. If  $q_i = K_3$ , then  $\pi(x_i)$  returns to 1, *i.e.*, all neighbors are in the same class. Adding the fuzzy term  $\pi(x_i)$ , the within-class scatter matrix becomes:

$$\mathbf{S}_w^{\phi^F} = \sum_{k=1}^{N_C} \left( \sum_{\phi(x_i) \in C_k} \pi(x_i) \times \sum_{L_{m,n} \in F_{K_1}(\phi(x_i), C_k)} (\phi(x_i) - L_{m,n}(\phi(x_i))) (\phi(x_i) - L_{m,n}(\phi(x_i)))^T \right) \tag{14}$$

Similarly, a fuzzy term  $\lambda(x_i)$  is also adopted to evaluate the membership grade of  $\phi(x_i)$  and its neighbors during the computation of between-class scatter as follows:

$$\lambda(x_i) = \begin{cases} 0.51 + (0.49 \times (p_i / K_4)) & \text{if } p_i \geq \theta_{between}; \\ 0.49 \times (p_i / K_4) & \text{otherwise.} \end{cases} \tag{15}$$

Here, value  $p_i$  is the number of samples with labels different from  $\phi(x_i)$  among  $K_4$  nearest neighbors, and  $\theta_{between}$  is a given threshold. If  $p_i = K_4$ , term  $\lambda(x_i)$  returns to 1. That means that all neighbors have labels different from  $\phi(x_i)$ . The fuzzy term  $\lambda(x_i)$  is added into the between-class scatter matrix to generate a new one as:

$$\mathbf{S}_b^{\phi^F} = \sum_{k=1}^{N_c} \left( \sum_{\phi(x_i) \in C_k} \lambda(x_i) \times \sum_{l=1, l \neq k}^{N_c} \sum_{L_{m,n} \in F_{K_2}(\phi(x_i), C_l)} (\phi(x_i) - L_{m,n}(\phi(x_i))) (\phi(x_i) - L_{m,n}(\phi(x_i)))^T \right). \quad (16)$$

Hence, kernelization and fuzzification are simultaneously integrated into the NFLE transformation for feature extraction. The pseudo-codes of algorithm FKNFLE are listed in Table 1. It is proposed in this paper that a general format for the NFLE learning method using kernelization and fuzzification be used for DR. The advantages of the proposed method are threefold: the kernelization strategy generates a non-linear feature space for the discriminant analysis and increases the robustness to noise for manifold learning; the kernelized manifold learning preserves the local manifold structure in a Hilbert space as well as the locality of the manifold structure in the reduced low-dimensional space; non-Euclidean structures are extracted for improving discriminative abilities using the FKNN strategy.

**Table 1.** The pseudo-codes of FKNFLE (fuzzy-kernel nearest feature line) training algorithm.

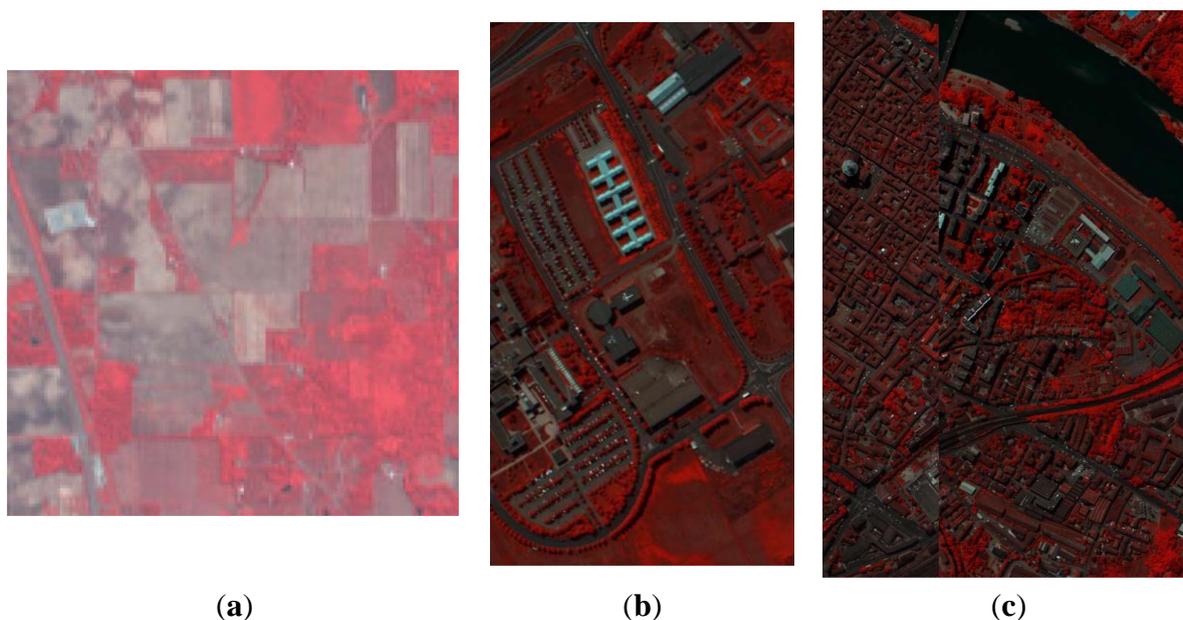
<b>Input:</b>	<b>A <math>d</math>-dimensional training set <math>X = [x_1, x_2, \dots, x_N]</math> consists of <math>N_c</math> classes projected into a Hilbert space <math>\phi(X) = [\phi(x_1), \phi(x_2), \dots, \phi(x_N)]</math>, and parameters <math>K_1, K_2, K_3, K_4</math>.</b>
Output:	The projection transformation $w$ .
Step 1:	PCA projection: Samples are transformed from a high-dimensional space into a low-dimensional subspace by matrix $w_{PCA}$ .
Step 2:	Computation of the within-class scatter: The possible feature lines $L_{m,n}$ are generated from the samples within the same class for a specified point $\phi(x_i)$ . Find $K_3$ nearest neighbors of point $\phi(x_i)$ to calculate the fuzzy membership values $\pi(x_i)$ by Equation (13). Select $K_1$ vectors $\phi(x_i) - L_{m,n}(\phi(x_i))$ with the smallest distances, and compute the within-class scatter $\mathbf{S}_w^{\phi^F}$ by Equation (14).
Step 3:	Computation of the between-class scatter: Generate the feature lines from the samples whose labels are different from that of point $\phi(x_i)$ . Find $K_4$ nearest neighbors of point $\phi(x_i)$ to calculate the fuzzy membership values $\lambda(x_i)$ by Equation (15). Select $K_2$ discriminant vectors $\phi(x_i) - L_{m,n}(\phi(x_i))$ with the smallest distances from point $\phi(x_i)$ to the feature lines. The between-class scatter $\mathbf{S}_b^{\phi^F}$ is obtained from Equation (16).
Step 4:	Fisher criterion maximization: The Fisher criterion $w^* = \arg \max \mathbf{S}_b^{\phi^F} / \mathbf{S}_w^{\phi^F}$ is maximized to obtain the best transformation matrix, which is composed of $\gamma$ eigenvectors with the largest eigenvalues.
Step 5:	Output the final transformation matrix: $w = w_{PCA} w^*$ .

## 4. Experimental Results

### 4.1. Description of Data Sets

In this section, the experimental results are discussed to demonstrate the effectiveness of the proposed method for HSI classification. Three HSI benchmarks are given for evaluation. The first data set, Indian

Pines Site (IPS) image, was generated from AVIRIS (Airborne Visible/Infrared Imaging Spectrometer), which was captured by the Jet Propulsion Laboratory and NASA/Ames in 1992. The IPS image was captured from six miles in the western area of Northwest Tippecanoe County (NTC). A false color IR image of dataset IPS is shown in Figure 1a. The IPS dataset contained 16 land-cover classes with 220 bands, e.g., Alfalfa(46), Corn-notill(1428), Corn-mintill(830), Corn(237), Grass-pasture(483), Grass-trees(730), Grass-pasture-mowed(28), Hay-windrowed(478), Oats(20), Soybeans-notill(972), Soybeans-mintill(2455), Soybeans-cleantill(593), Wheat(205), Woods(1265), and Bldg-Grass-Tree-Drives(386), and Stone-Steel-Towers(93). The numbers in parentheses were the collected pixel numbers in the dataset. The ground truths in dataset IPS of 10,249 pixels were manually labeled for training and testing. In order to analyze the performance of various algorithms, 10 classes of more than 300 samples were adopted in the experiments, e.g., a subset IPS-10 of 9620 pixels. Nine hundred training samples of 10 classes in subset IPS-10 were randomly chosen from 9,620 pixels, and the remaining samples were used for testing.



**Figure 1.** False color of IR images for datasets (a) Indian Pines Site (IPS); (b) Pavia University; and (c) Pavia City Center.

The other two HSI data sets adopted in the experiments were obtained from the Reflective Optics System Imaging Spectrometer (ROSIS) instrument covering the City of Pavia, Italy. Two scenes, the university area and the Pavia city center, contained 103 and 102 data bands, both with a spectral coverage from 0.43 to 0.86  $\mu\text{m}$  and a spatial resolution of 1.3 m. The image sizes of these two areas were  $610 \times 340$  and  $1096 \times 715$  pixels, respectively. Figure 1b,c show the false color IR image of these two data sets. Nine land-cover classes were available in each data set, and the samples in each data set were separated into two subsets, *i.e.*, one training and one testing set. Given the Pavia University data set, 90 training samples per class were randomly collected for training, and the 8046 remaining samples were tested for performance evaluation. Similarly, the numbers of training and testing samples used for the Pavia City Center data set were 810 and 9529, respectively.

4.2. A Toy Example

Two toy examples are given to illustrate the discriminative power of FKNFLE in the following. Firstly, 561 samples with 220 dimensions of the three classes (Grass/pasture, Woods, and Grass/trees) were collected from a hyperspectral image. The samples were projected onto the first three axes using eight algorithms: PCA, LDA, supervised LPP, LFDA [28], NFLE, FNFLE, KNFLE, and FKNFLE, as shown in Figure 2. These class samples are represented by green triangles (class G), blue stars (class B), and red circles (class R). A simple analysis was done by observing the sample distributions in the reduced spaces. Since the global Euclidean structure criterion was considered during the PCA and LDA training phases, the samples from three classes in the reduced spaces were mixed after the PCA and LDA projections as shown in Figure 2a,b. Since the samples were distributed in a manifold structure in the original space, the manifold learning algorithms, e.g., supervised LPP, LFDA, and NFLE, were executed to preserve the local structure of the samples. The sample distributions projected by supervised LPP, LFDA, and NFLE are displayed in Figure 2c–e, respectively. Three classes were efficiently separated and contrasted with those in Figure 2a,b. The class boundaries, however, were unclear due to the non-linear and non-Euclidean sample distributions in the original space. Kernelization and fuzzification were pre-performed to extend the original non-Euclidean and non-linear space to a higher linear space. Consider the sample distributions in Figure 2e,h, the boundaries of classes G and R in Figure 2e being still unclear using the NFLE transformation. The sample distributions of FNFLE and KNFLE as shown in Figure 2f,g were the results when the kernelization and fuzzification strategies were used, respectively. Obviously, classes G and R were more effectively separated than those in Figure 2e. The local structures of the samples from the observed sample distribution were preserved, and the class separability improved. Several points located at the boundaries were misclassified in these cases. When both strategies were further adopted in FKNFLE, only one red point was mis-located at class G, and classes G and R were clearly separated. From the analysis, both fuzzification and kernelization strategies enhanced the discriminative power of manifold learning methods.

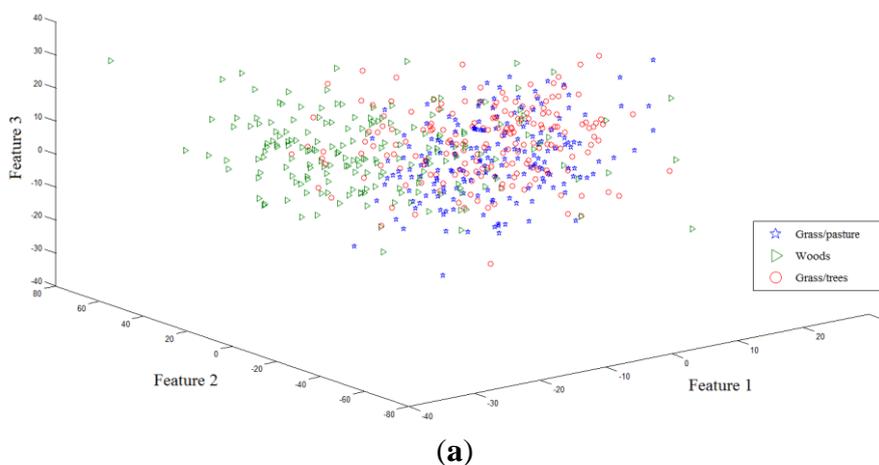
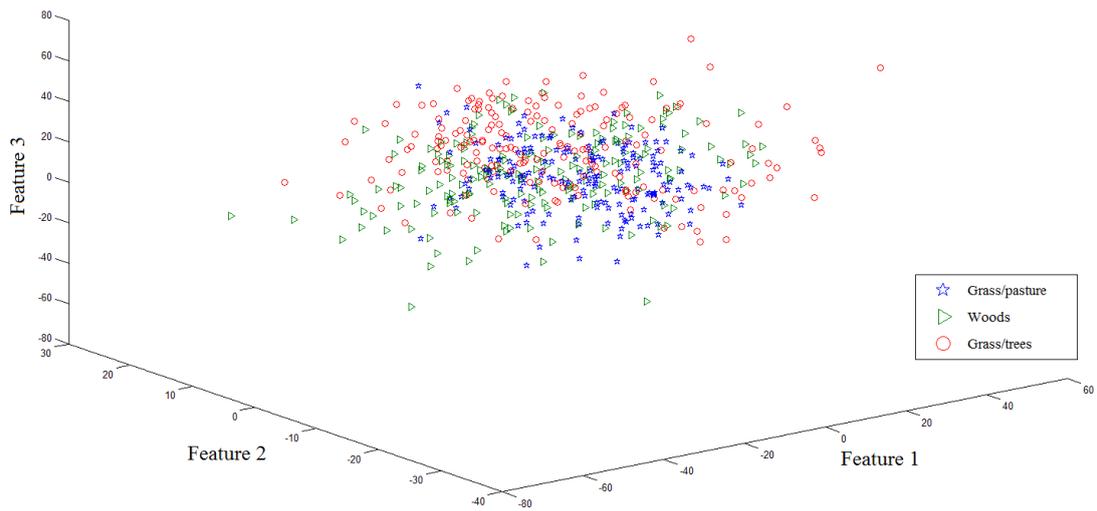
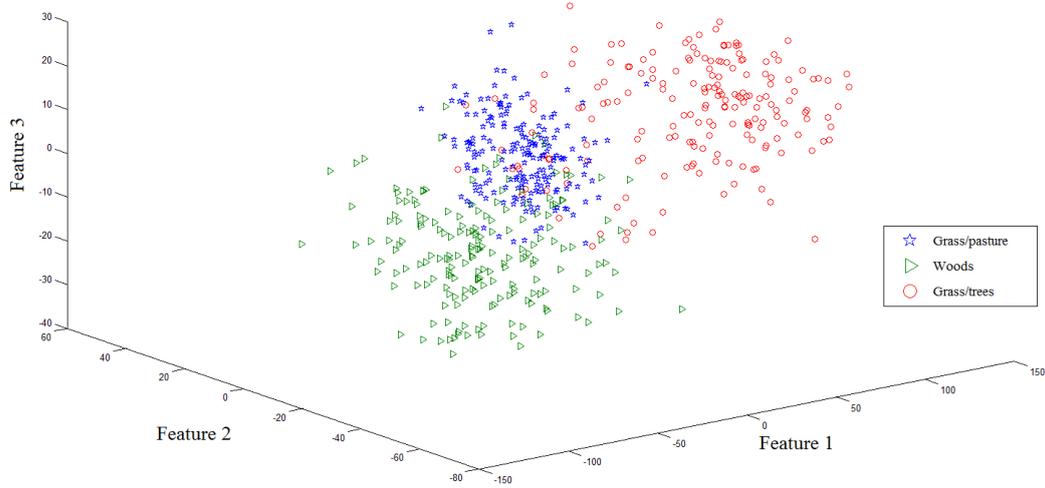


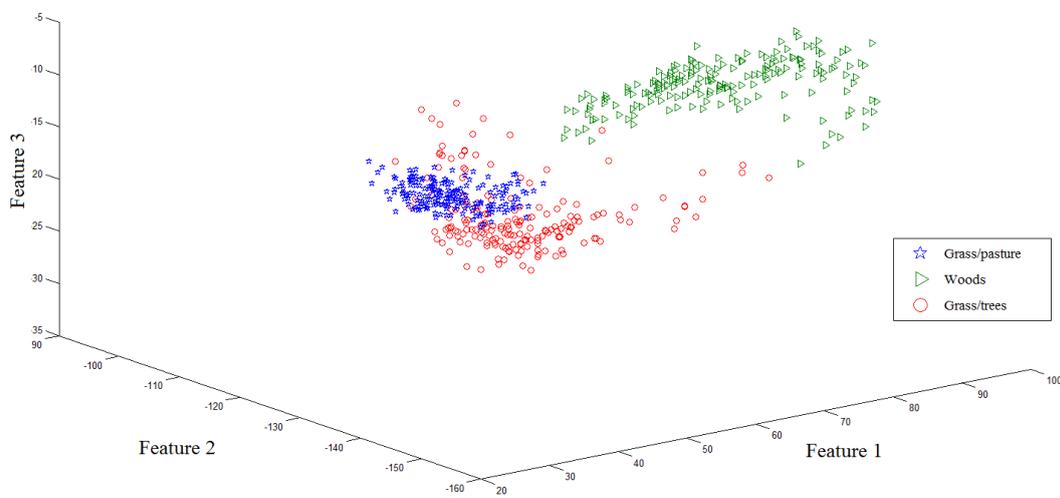
Figure 2. Cont.



(b)



(c)



(d)

Figure 2. Cont.

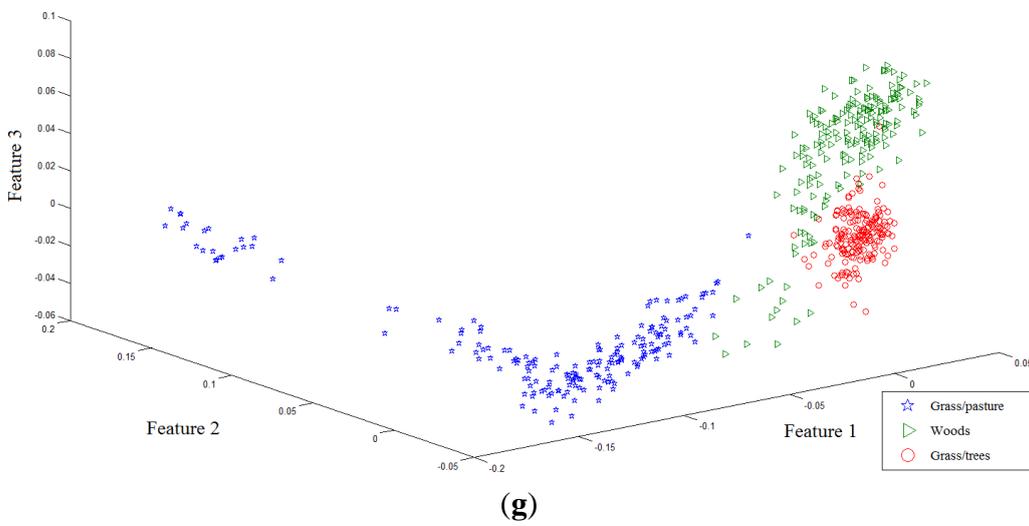
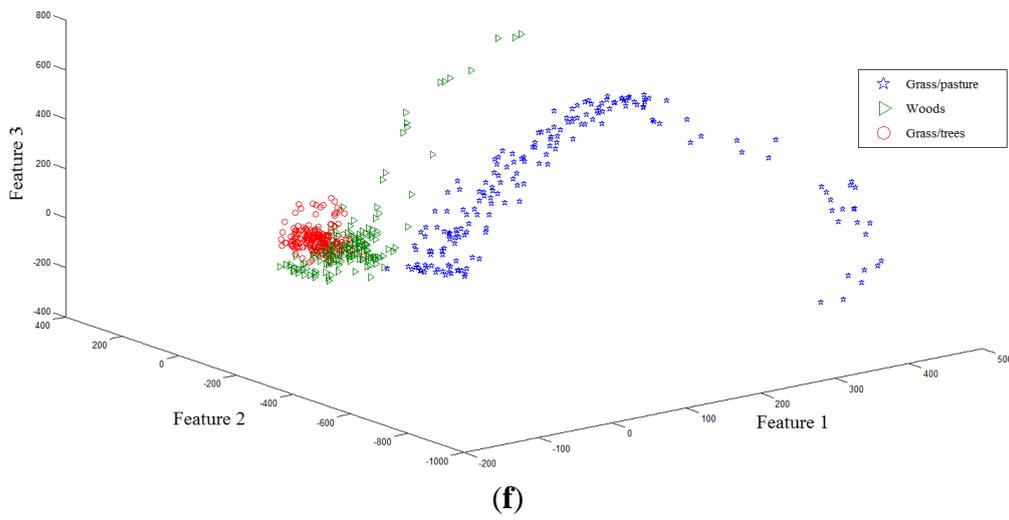
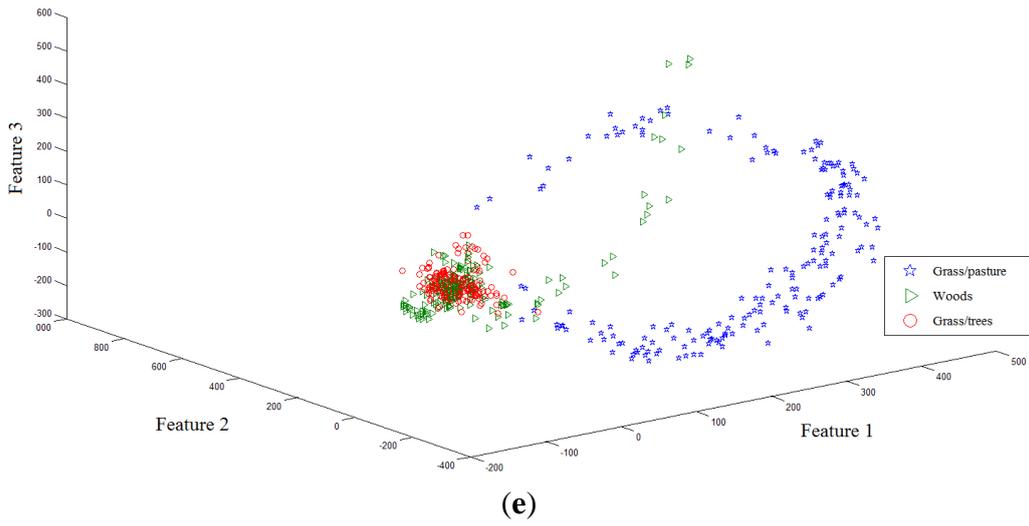
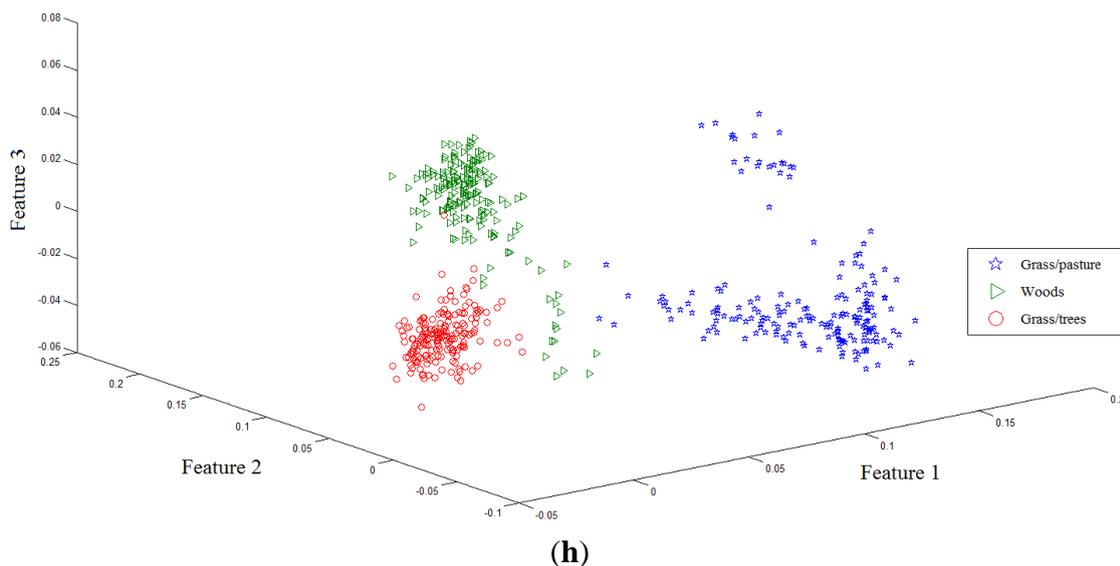
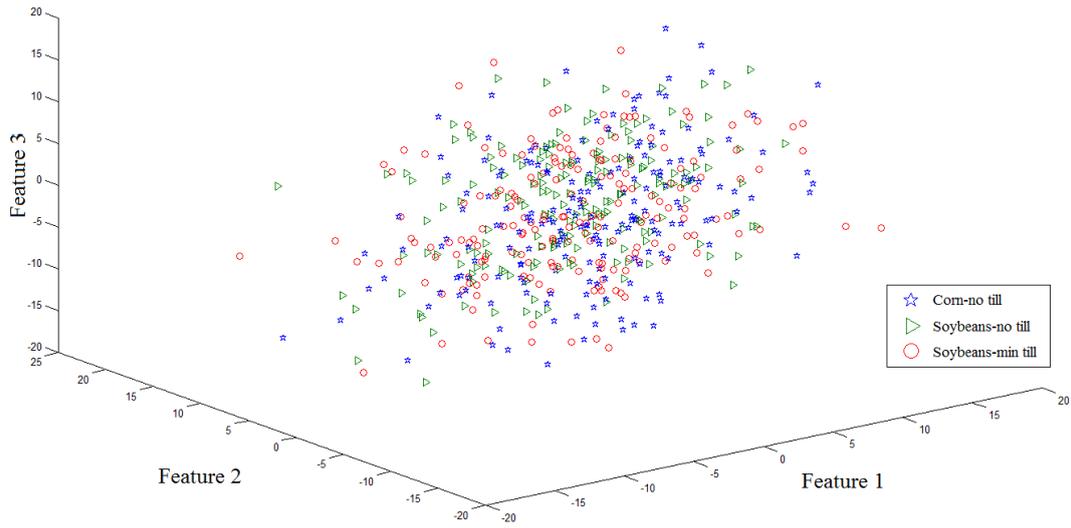


Figure 2. Cont.

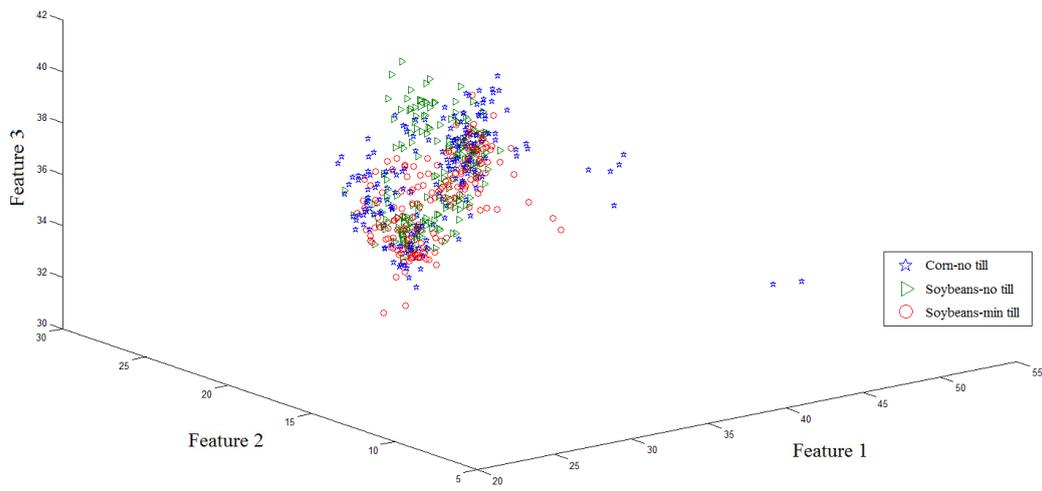


**Figure 2.** The first toy sample distributions projected on the first three axes using algorithms (a) PCA (principal component analysis); (b) LDA (linear discriminant analysis); (c) supervised LPP (locality preserving projection); (d) LFDA (local Fisher discriminant analysis); (e) NFLE (nearest feature line (NFL) embedding); (f) FNFLE (fuzzy nearest feature line embedding); (g) KNFLE (kernel nearest feature line embedding); and (h) FKNFLE (fuzzy-kernel nearest feature line).

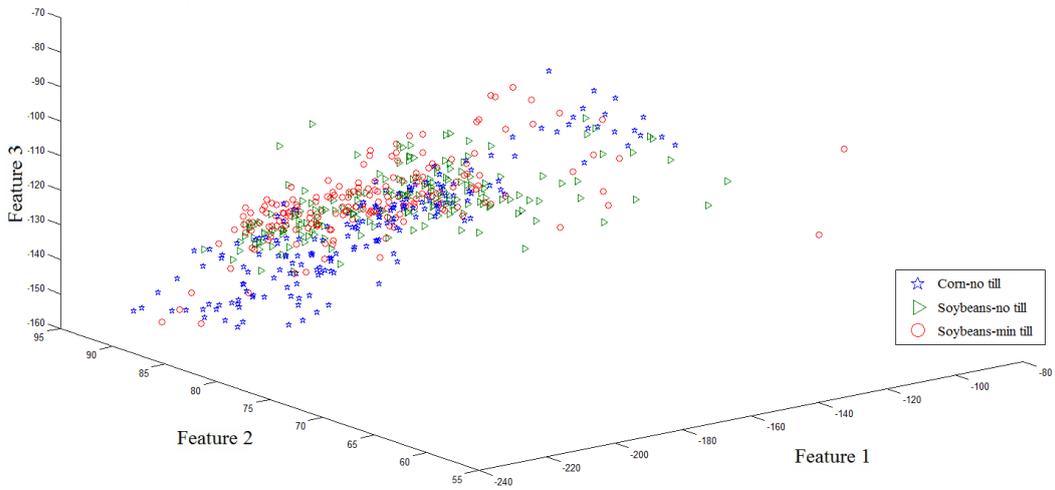
Secondly, 561 samples with 220 dimensions of the three classes (Corn-no till, Soybeans-min till, Soybeans-no till) were collected from a hyperspectral image. The samples were projected onto the first three axes by eight algorithms: PCA, LDA, supervised LPP, LFDA, NFLE, FNFLE, KNFLE, and FKNFLE, as shown in Figure 3. These class samples are also represented by green triangles (class G), blue stars (class B), and red circles (class R). A simple analysis was also done by observing the sample distributions in the reduced spaces. Since the global Euclidean structure criterion was considered during the PCA and LDA training phases, the samples of three classes in the reduced spaces were mixed after the PCA and LDA projections as shown in Figure 3a,b. Since the samples were distributed in a manifold structure in the original space, the manifold learning algorithms, e.g., supervised LPP, LFDA, and NFLE, were executed to preserve the local structure of the samples. The sample distributions projected by supervised LPP, LFDA, and NFLE are displayed in Figure 3c–e, respectively. Due to the strong overlapping in classes G, R, and B, they were mixed, and the separation was relatively low compared with those in Figure 2c–e. However, when the kernelization and fuzzification strategies were used, class B was more effectively separated than those shown in Figure 3c–e. According to the analysis, in the case of strong overlapping, both fuzzification and kernelization strategies enhanced the discriminative power of manifold learning methods.



(a)

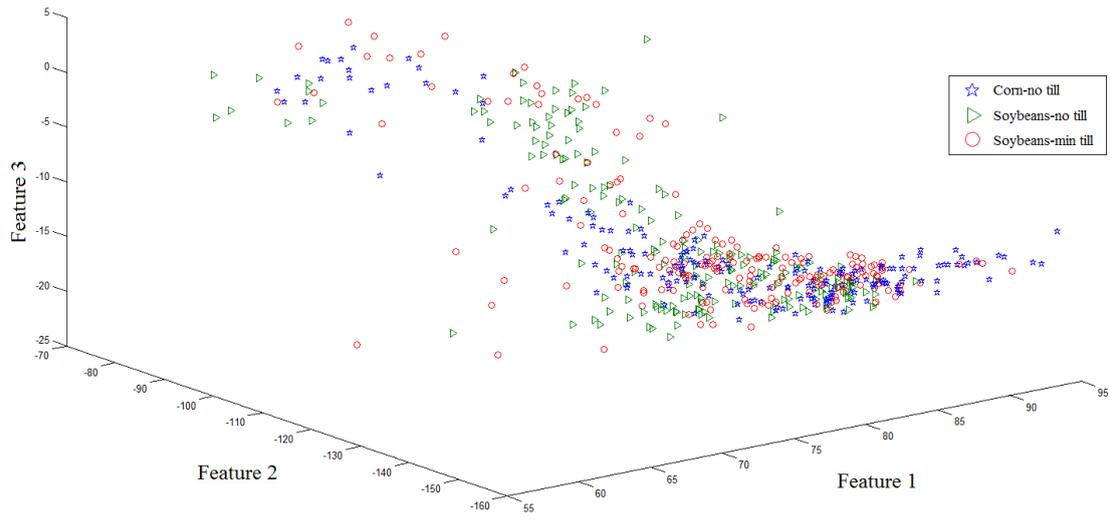


(b)

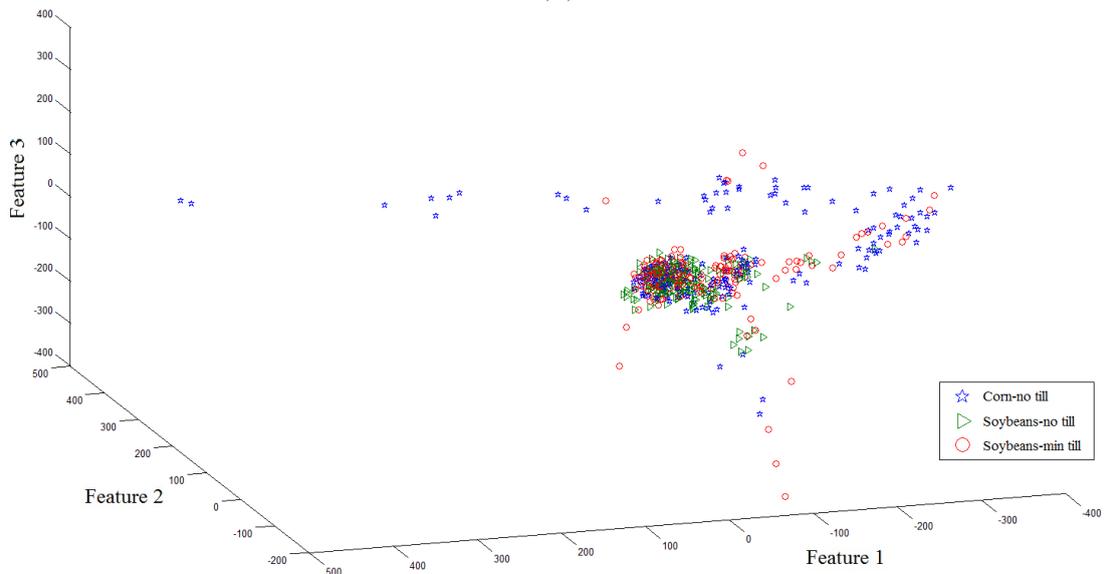


(c)

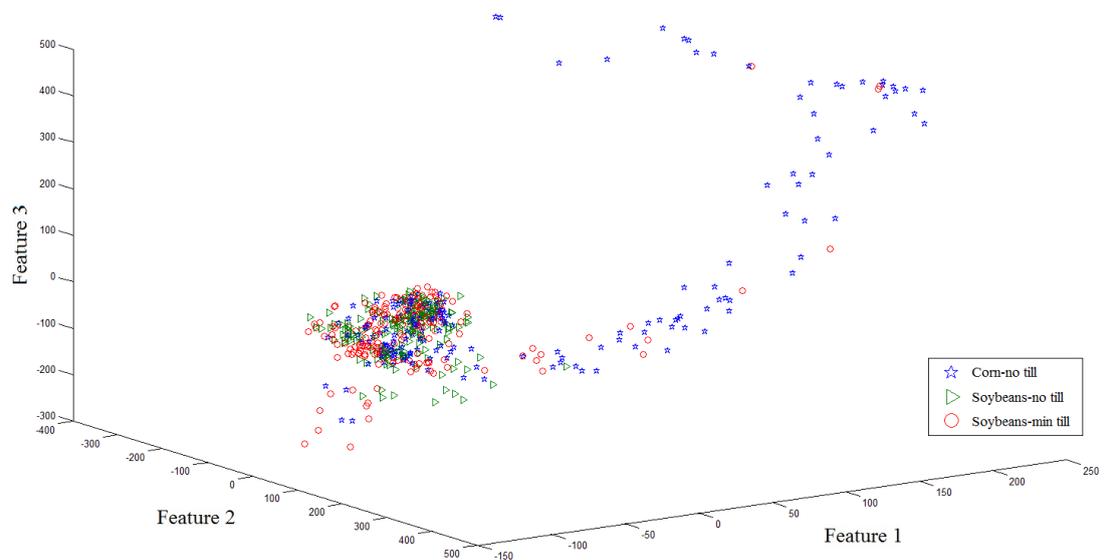
Figure 3. Cont.



(d)

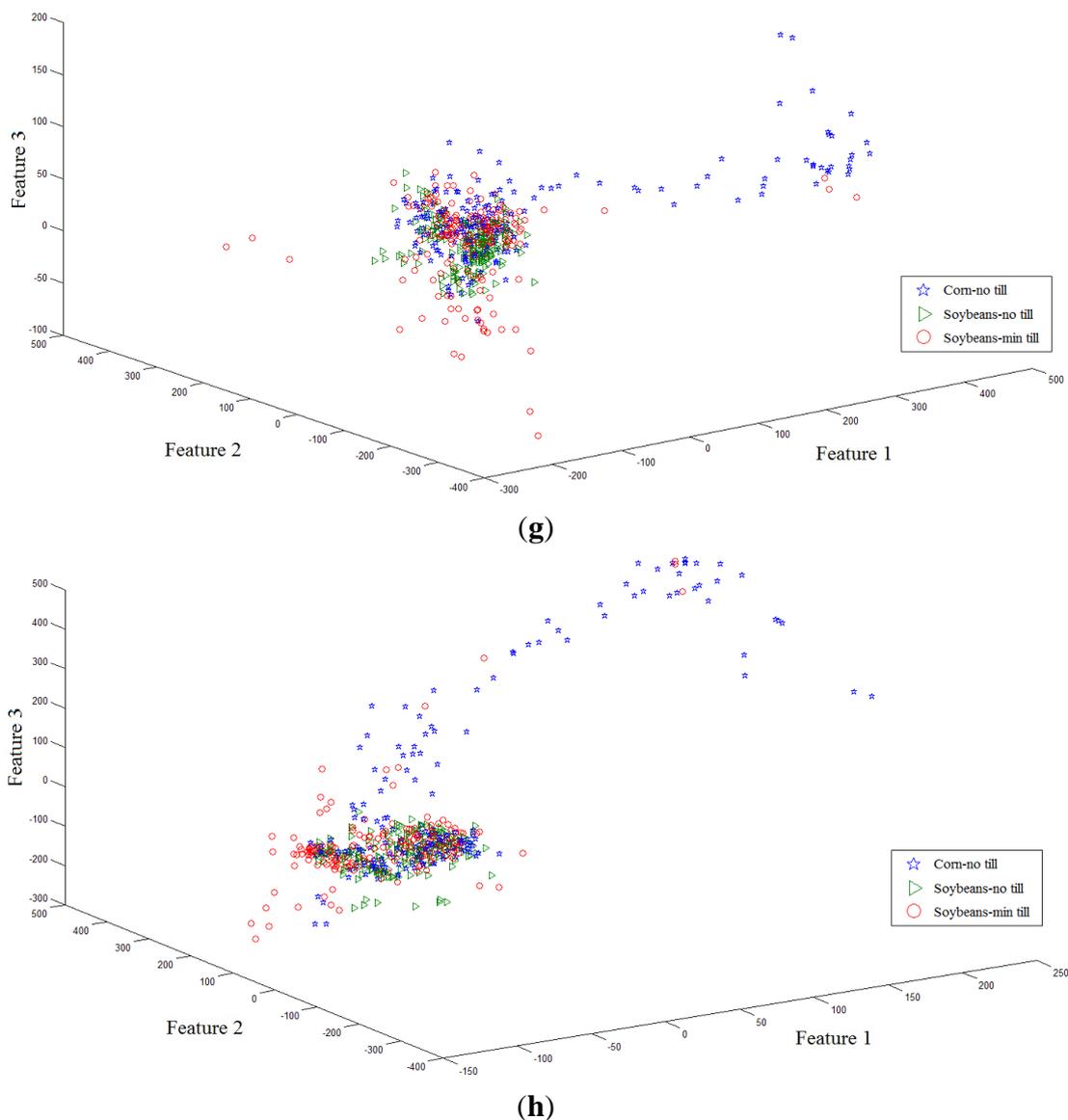


(e)



(f)

Figure 3. Cont.

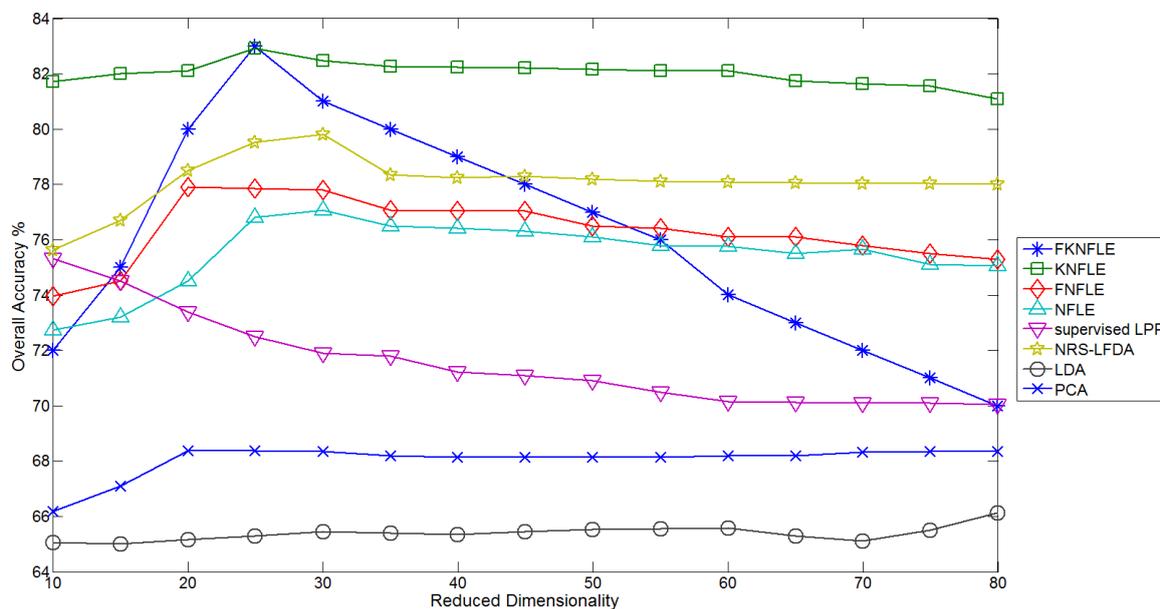


**Figure 3.** The second toy sample distributions projected on the first three axes using algorithms (a) PCA; (b) LDA; (c) supervised LPP; (d) LFDA; (e) NFLE; (f) FNFLE; (g) KNFLE; and (h) FKNFLE.

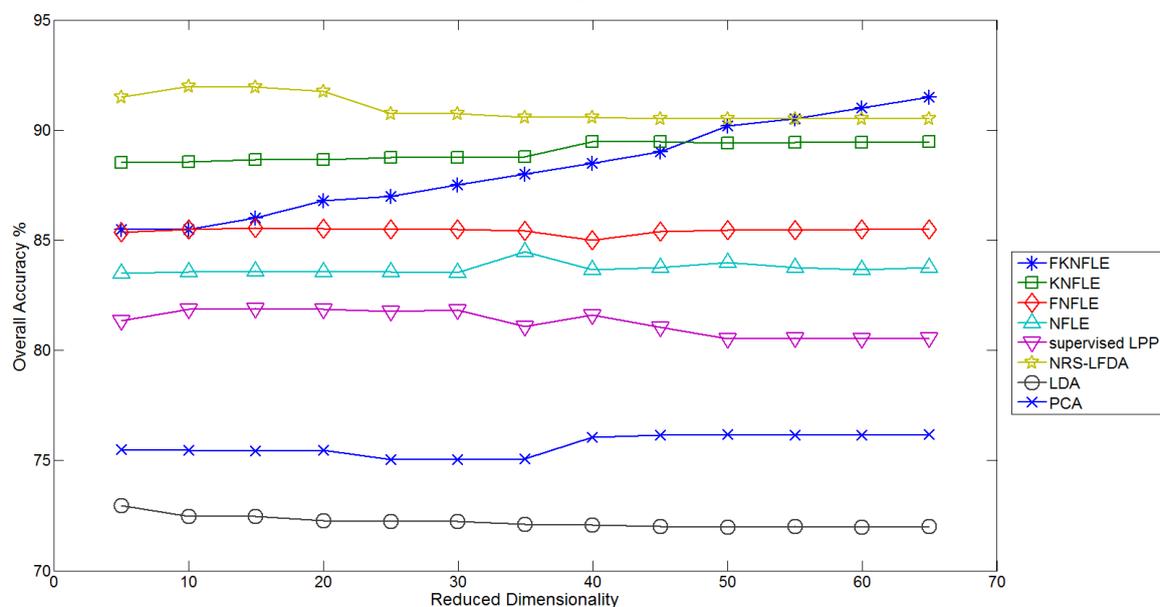
#### 4.3. Classification Results

The proposed methods, NFLE [20,21], KNFLE, FNFLE [26], and FKNFLE, were compared with two state-of-the-art algorithms, *i.e.*, nearest regularized subspace (NRS) [25] and NRS-LFDA [25]. The parameter configurations for both algorithms NRS [29] and NRS-LFDA were as seen in [25]. The gallery samples were randomly chosen for training the transformation matrix, and the query samples were matched with the gallery samples using the nearest neighbor (NN) matching rule. Each algorithm was run 30 times to obtain the average rates. To obtain the appropriate reduced dimensions of FKNFLE, the available training samples were used to evaluate the overall accuracy (OA) *versus* the reduced dimensions in the benchmark datasets. As shown in Figure 4, the best dimensions of algorithm FKNFLE for datasets IPS-10, Pavia University, and Pavia City Center were 25, 50, and 50, respectively. The proposed FKNFLE and KNFLE algorithms are both extended from algorithm NFLE. From the

classification results as shown in Figure 4, though FKNFLE achieves the best results at the specific reduced dimensions on three datasets, the high variant OA rates are obtained. Moreover, two additional parameters  $K_3$  and  $K_4$  were needed for training during the fuzzification. On the other hand, the performance of KNFLE is more robust than that of FKNFLE. KNFLE usually achieves a higher performance even at low reduced dimensions, e.g., five or 10. It also outperforms the other algorithms at all reduced dimensions on datasets IPS-10 and Pavia City Center. Compared with NRS-LDA, slightly reduced OA rates were obtained on dataset Pavia University. From this analysis, algorithm KNFLE is more competitive than FKNFLE in HSI classification.

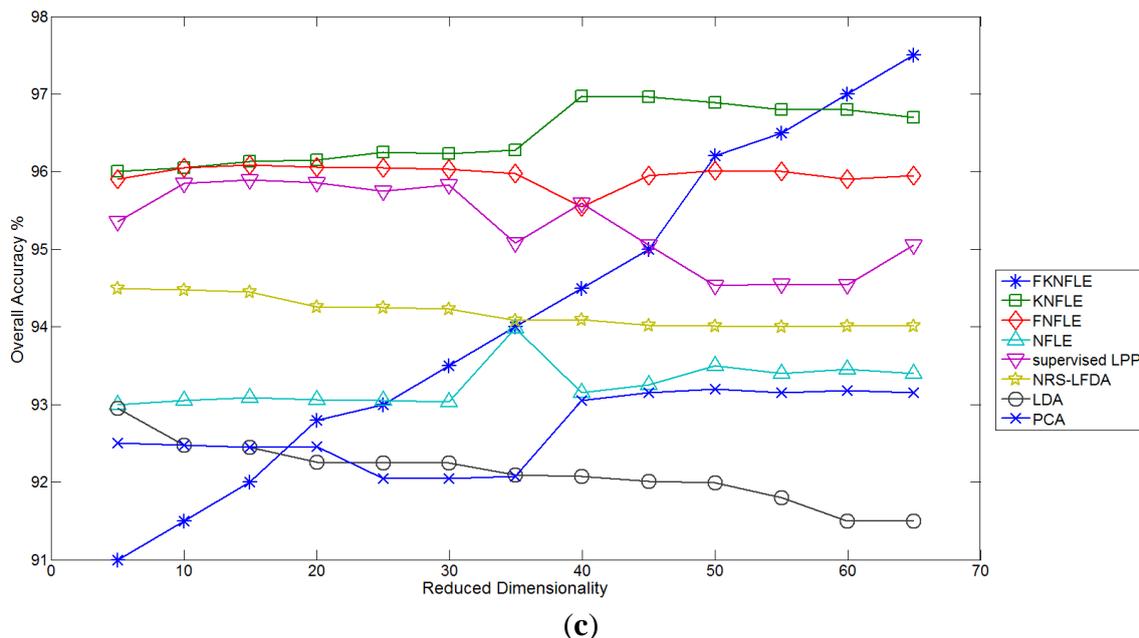


(a)



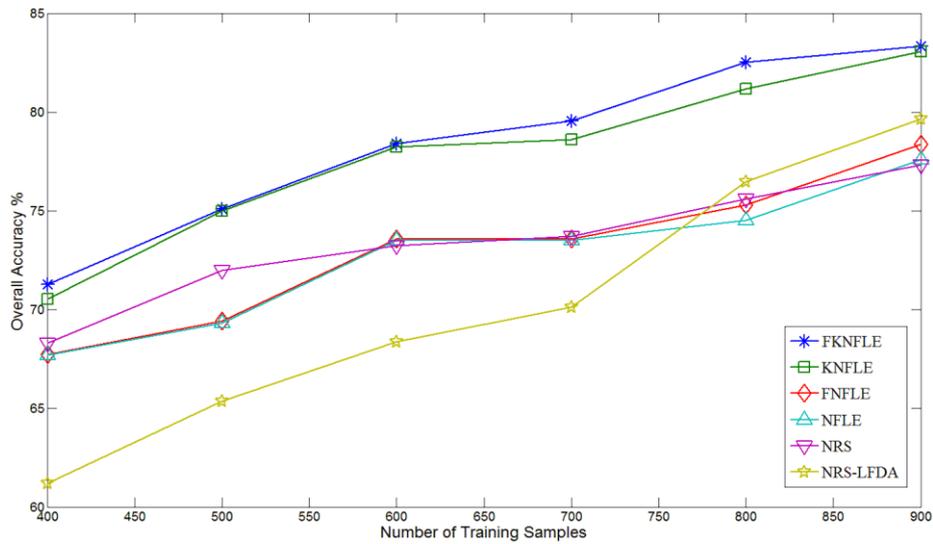
(b)

Figure 4. Cont.

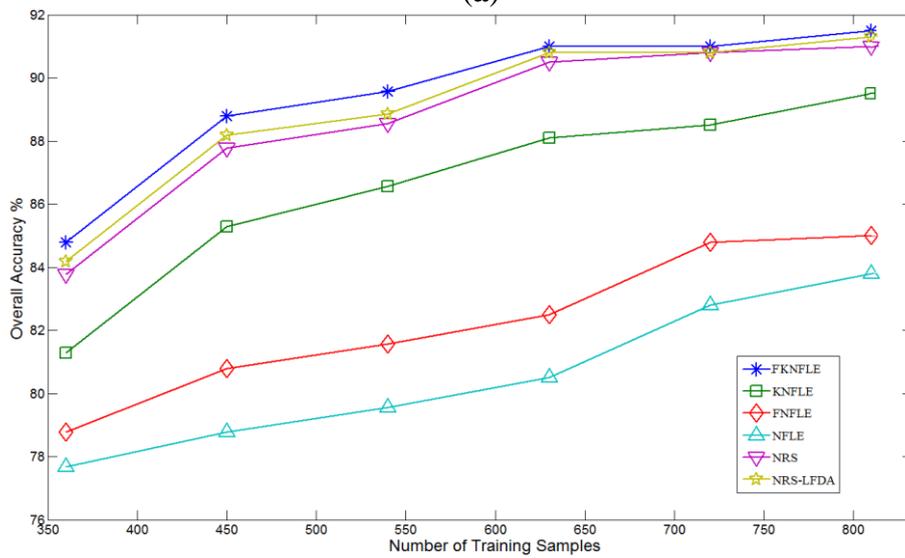


**Figure 4.** The classification accuracy *versus* the reduced dimension on three benchmark datasets using the various algorithms: (a) IPS-10; (b) Pavia University; (c) Pavia City Center.

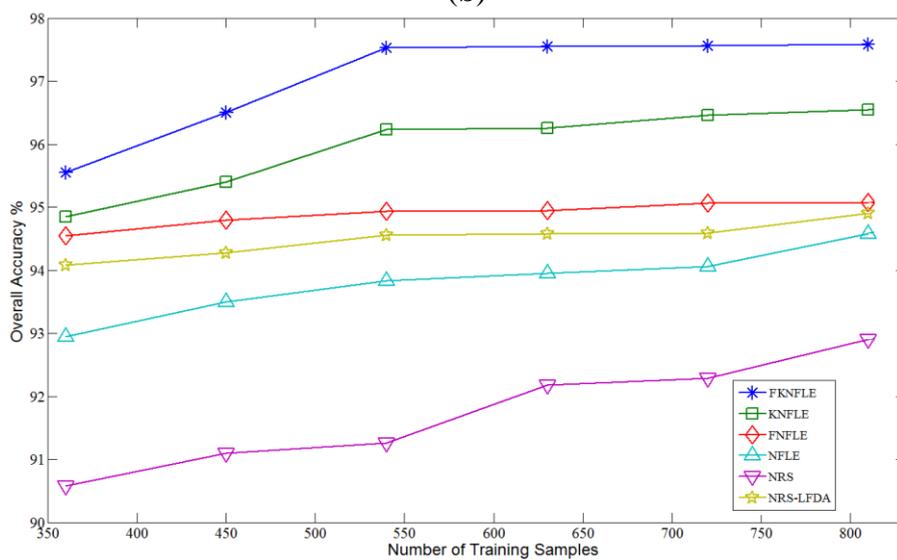
The average classification rates *versus* the number of training samples on dataset IPS-10 are shown in Figure 5a; algorithms FKNFLE and KNFLE outperformed the other methods. The accuracy rate of FKNFLE was 4% higher than that of FNFLE. The kernelization strategy effectively enhanced the discriminative power. The performance of FKNFLE was better than that of KNFLE to a value of 0.8%, and the rate of FNFLE was higher than that of NFLE to a value of 0.7%. This shows that the fuzzification strategy slightly enhanced the performance. Figure 5b,c also demonstrates the overall accuracy *versus* the number of training samples in the benchmark datasets of Pavia University and Pavia City Center, respectively. According to the classification rates in these two datasets, algorithm FKNFLE outperformed the other methods. In addition, the classification results were insensitive to the number of training samples. Next, the maps of the classification results for the dataset IPS-10 are given in Figure 6. The classification results of algorithms FKNFLE, KNFLE, FNFLE, NFLE, NRS, and NRS-LFDA are given based on the maps of  $145 \times 145$  pixels depicting the ground truth. The speckle-like errors of FKNFLE were fewer than those of the other algorithms. Figures 6–8 give the maps of the classification results for datasets Pavia University and Pavia City Center, respectively. Once again, the speckle-like errors of FKNFLE were fewer than in the case of the other algorithms. In addition, the thematic maps of Pavia University and Pavia City Center are shown in Figure 9a,b, respectively, using the proposed FKNFLE method. Observing the results in Figure 9a, the roads, buildings, and the areas in University were clearly classified even though there was some speckle-like noise in the images. The roads, rivers, buildings, small islands, and the areas in the city were classified in the same way. See Figure 9b. Algorithm FKNFLE effectively classified the land cover even in the limited training samples.



(a)



(b)



(c)

**Figure 5.** The accuracy rates *versus* the number of training samples for datasets (a) IPS-10; (b) Pavia University; and (c) Pavia City Center.

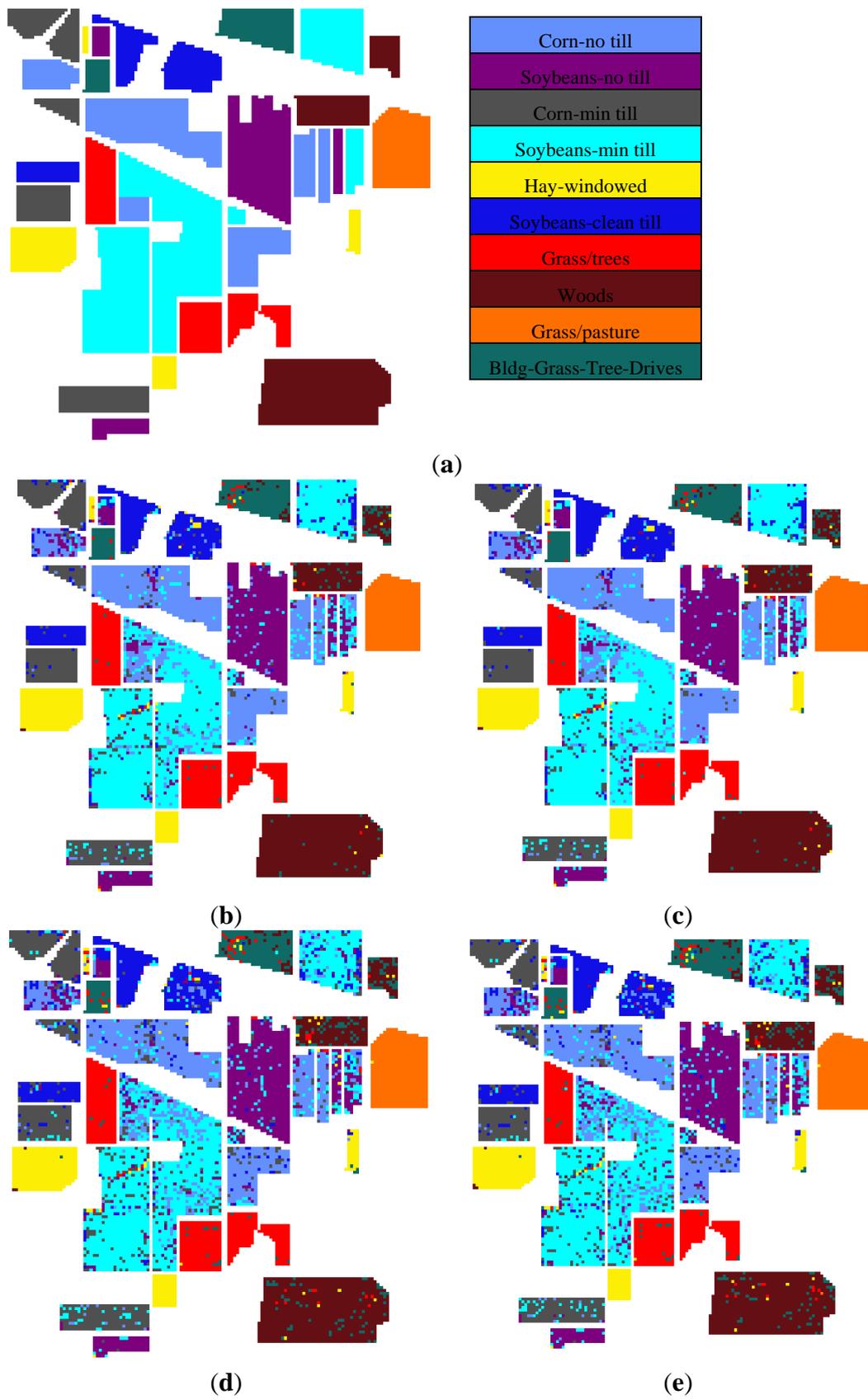
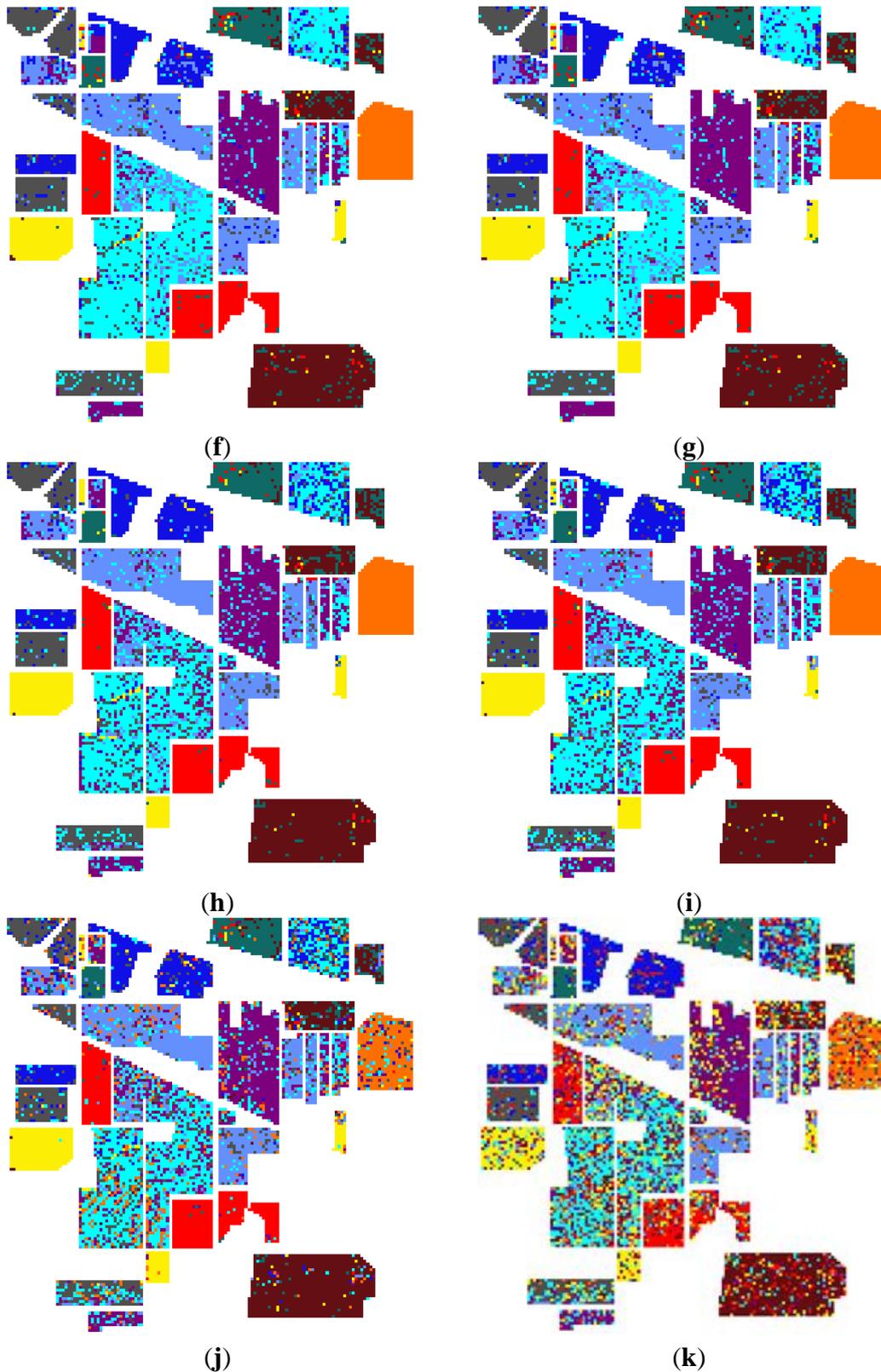


Figure 6. Cont.



**Figure 6.** The classification maps of dataset IPS using various algorithms: (a) The ground truth; (b) FKNFLE; (c) KNFLE; (d) FNFLE; (e) NFLE; (f) NRS (nearest regularized subspace); (g) LFDA-NRS (local Fisher discriminant analysis-NRS); (h) LFDA; (i) supervised LPP; (j) LDA; and (k) PCA.

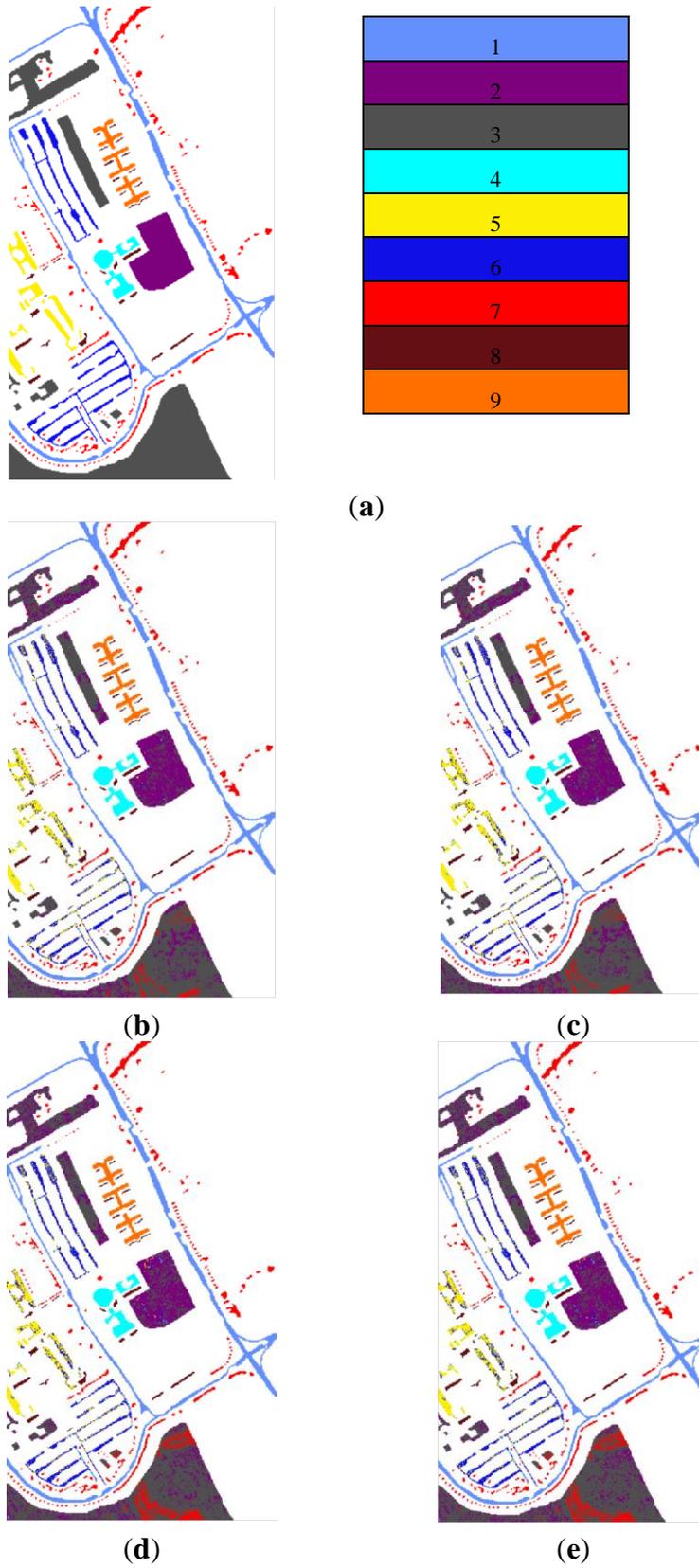
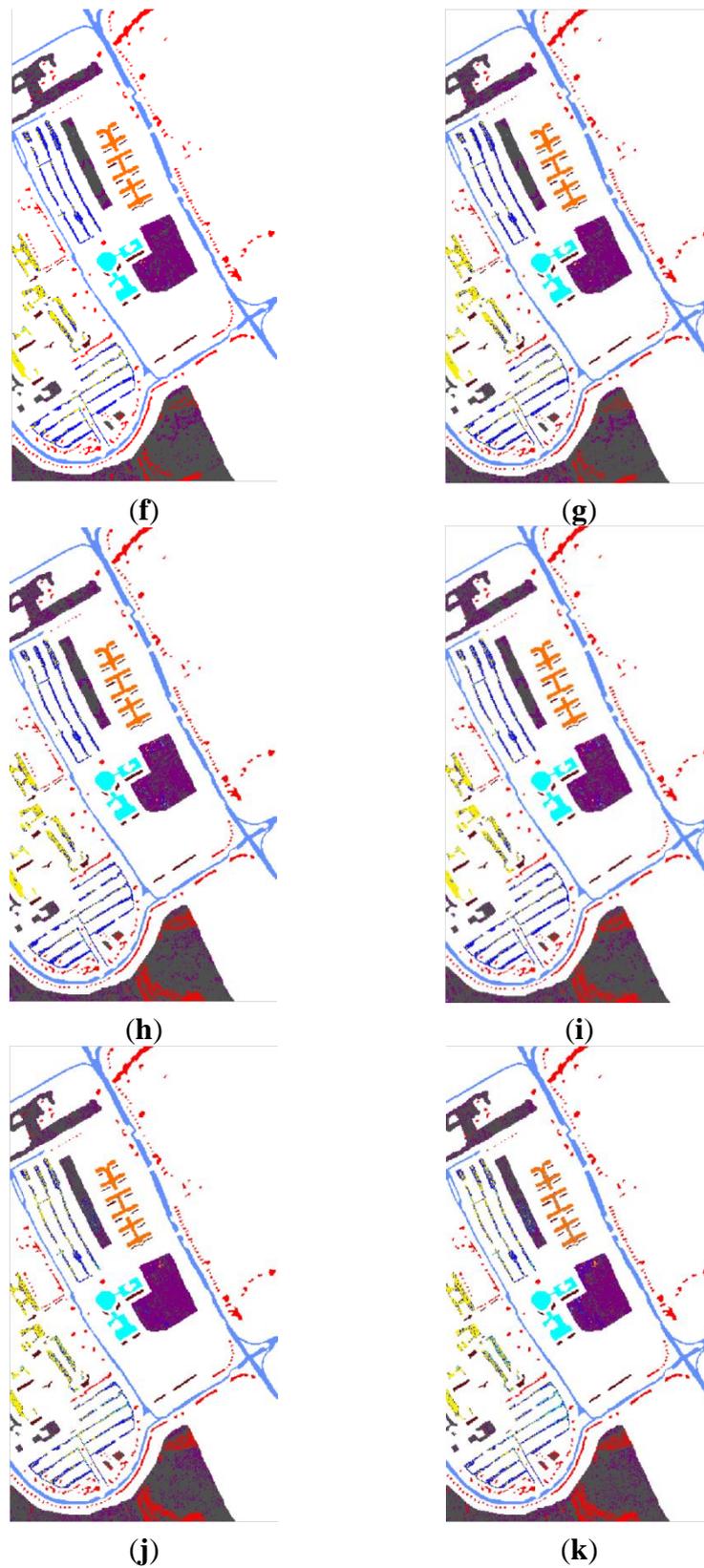


Figure 7. Cont.



**Figure 7.** The classification maps of dataset Pavia University using various algorithms: (a) The ground truth; (b) FKNFLE; (c) KNFLE; (d) FNFLE; (e) NFLE; (f) NRS; (g) LFDA-NRS; (h) LFDA; (i) supervised LPP; (j) LDA; and (k) PCA.

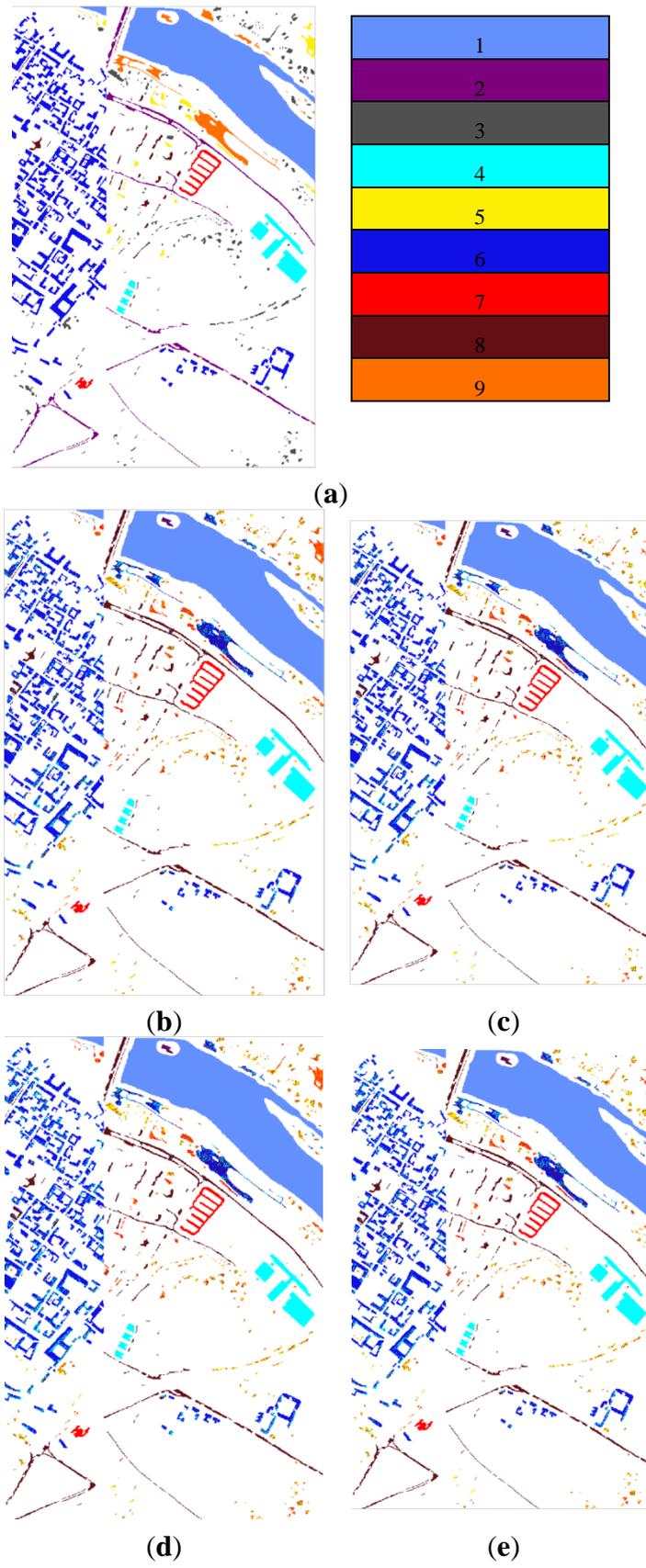
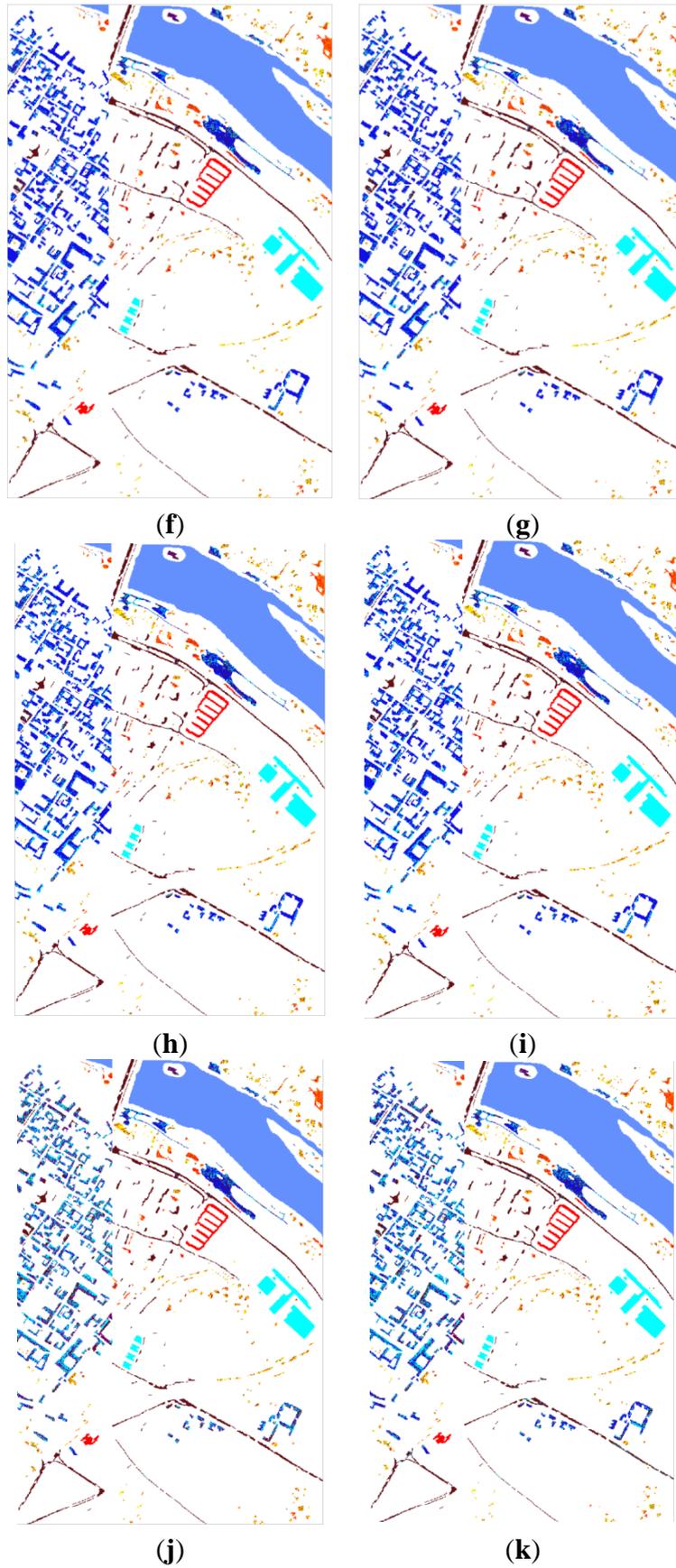
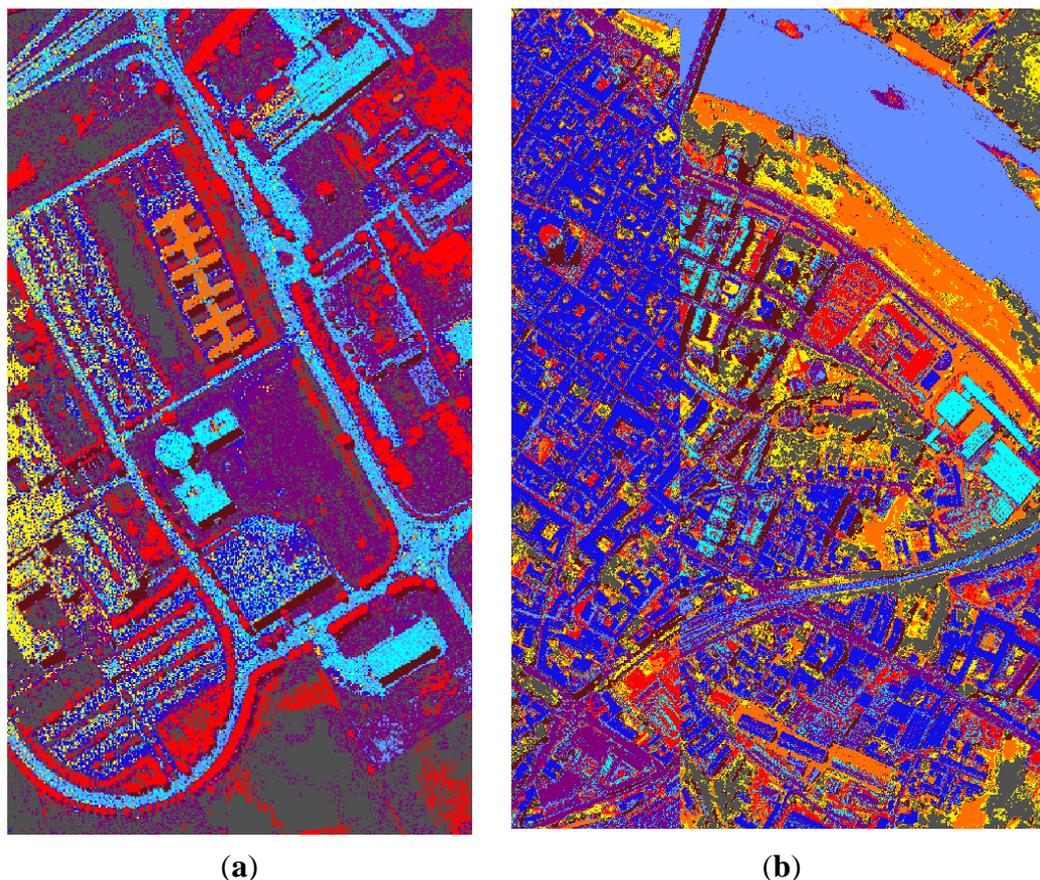


Figure 8. Cont.



**Figure 8.** The classification maps of dataset Pavia City Center using various algorithms: (a) The ground truth, (b) FKNFLE; (c) KNFLE; (d) FNFLE; (e) NFLE; (f) NRS; (g) LFDA-NRS; (h) LFDA; (i) supervised LPP; (j) LDA; and (k) PCA.



**Figure 9.** The thematic maps of (a) Pavia University, and (b) Pavia City Center using the proposed FKNFLE algorithm.

The proposed method was compared with various classification methods on computational time. All methods were implemented by MATLAB codes on a personal computer with an i7 2.93-GHz CPU and 12.0 gigabyte RAM. The comparisons of various algorithms on computational time were tabulated in Table 2 for the IPS-10, Pavia University, and Pavia City Center datasets. Considering the training time, the proposed FKNFLE algorithm was generally faster than NRS and NRS-LFDA by two times and 15 times, respectively. Due to the fuzzification process, algorithms FKNFLE and FNFLE were slower than KNFLE and NFLE, by 13 times and 15 times, respectively.

**Table 2.** The training and testing times of various algorithms for the benchmark datasets (s).

Datasets	IPS-10		Pavia University		Pavia City Center	
	Training	Testing	Training	Testing	Training	Testing
	900	8720	810	8046	810	9529
NFLE-NN	10	18	9	16	9	20
KNFLE-NN	12	18	11	16	11	20
FNFLE-NN	155	18	140	16	140	20
FKNFLE-NN	156	18	141	16	141	20
NRS	326	326	294	300	294	351
LFDA-NRS	2331	327	2098	301	2098	352

From Tables 3–5, the producer’s accuracy, overall accuracy, kappa coefficients, and user’s accuracy defined by the error matrices (or confusion matrices) [27] were calculated for performance evaluation. They are briefly defined in the following. The user’s accuracy and the producer’s accuracy are two widely used measures for class accuracy. The user’s accuracy is defined as the ratio of the number of correctly classified pixels in each class by the total pixel number classified in the same class. The user’s accuracy is a measure of commission error, whereas the producer’s accuracy measures the errors of omission and indicates the probability that certain samples of a given class on the ground are actually classified as such. The kappa coefficient, also called the kappa statistic, is defined to be a measure of the difference between the actual agreement and the changed agreement. The overall accuracies of the proposed method were 83.34% in IPS-10, 91.31% in Pavia University, and 97.59% in Pavia City Center with the kappa coefficients of 0.821, 0.910, and 0.971, respectively. Subset IPS-10 of 10 classes is used for fair comparisons with other algorithms. Another alternative classification on the whole IPS dataset of 16 classes was performed. Ten percent training samples of each class were randomly chosen from 10,249 pixels except for class Oats. Three training samples were randomly chosen from class Oats because of few samples in this data set. The remaining samples were used for testing. The classification error matrix is given in Table 6 in which the overall accuracy and kappa coefficient are 83.85% and 0.826, respectively.

**Table 3.** The classification error matrix for data set IPS-10 (in percentage).

Classes	Reference Data										User’s Accuracy
	1	2	3	4	5	6	7	8	9	10	
1	79.20	3.43	0.28	0.35	0	5.46	9.73	1.54	0	0	79.20
2	5.90	81.81	0	0.12	0	1.33	6.39	4.34	0	0.12	81.81
3	0	0	97.49	1.46	0.21	0.42	0	0.21	0.42	0.84	97.49
4	0	0	0.27	96.30	0	0	0	0	0	3.42	96.30
5	0	0	0.42	0	99.58	0	0	0	0	0	99.58
6	5.14	0.21	0.10	0.41	0	88.89	4.42	0.72	0	0.10	88.89
7	10.59	5.58	0.29	0.33	0.04	9.78	69.98	3.30	0	0.12	69.98
8	1.35	4.05	1.52	0.34	0	1.69	1.85	88.53	0	0.67	88.53
9	0	0	3.32	0.16	0	0	0	0	90.83	5.69	90.83
10	0	0	3.89	5.70	0	0	0	0.26	10.88	79.27	79.27
Producer’s Accuracy	77.51	86.04	90.62	91.57	99.75	82.63	75.76	89.51	88.94	87.85	
Kappa Coefficient: <b>0.821</b>						Overall Accuracy: <b>83.34%</b>					

**Table 4.** The classification error matrix for data set Pavia University (in percentage).

Classes	Reference Data									User’s Accuracy
	1	2	3	4	5	6	7	8	9	
1	90.18	3.15	0	0	0	3.24	1.35	1.26	0.81	90.18
2	2.31	92.50	0	2.31	0	1.85	0	1.01	0	92.50
3	0	0	90.07	2.38	1.58	0.99	2.97	0.99	0.99	90.07
4	0	1.23	2.84	90.24	1.42	1.42	1.51	1.32	0	90.24
5	0.63	1.13	0.75	1.26	91.91	0.63	1.64	0.88	1.13	91.91
6	1.10	1.19	1.38	1.56	1.19	92.54	0.55	0.46	0	92.54

Table 4. Cont.

Classes	Reference Data									User's Accuracy
	1	2	3	4	5	6	7	8	9	
7	0	1.12	0.51	0.61	2.24	0	93.25	1.22	1.02	93.25
8	0.47	1.42	0.95	1.42	2.38	1.90	0	90.76	0.66	90.76
9	1.14	0	2.15	2.01	0	2.29	0	2.15	90.22	90.22
Producer's Accuracy	94.10	90.92	91.30	88.65	91.25	88.25	92.08	90.71	95.14	
Kappa Coefficient: <b>0.910</b>						Overall Accuracy: <b>91.31%</b>				

Table 5. The classification error matrix for data set Pavia City Center (in percentage).

Classes	Reference Data									User's Accuracy
	1	2	3	4	5	6	7	8	9	
1	98.61	0.17	0.51	0.34	0.34	0	0	0	0	98.61
2	1.04	97.47	0.43	0	0	0.34	0.17	0.52	0	97.47
3	0.59	0.82	96.23	0.69	0.99	0	0	0	0.69	96.23
4	0	0.56	0.66	96.68	0.37	0.47	0.66	0.56	0	96.68
5	0	0	0.43	0.34	97.73	0.26	0.34	0.34	0.52	97.73
6	0.35	0.26	0.61	0	0	98.15	0	0.26	0.35	98.15
7	0.35	0.26	0	0.35	0	0.44	98.23	0.35	0	98.23
8	0	0	0.37	0.30	0.37	0.52	0.45	97.43	0.52	97.43
9	0.39	0.59	0.79	0.29	0.29	0	0	0	97.60	97.60
Producer's Accuracy	97.32	97.34	96.20	97.67	97.64	97.97	98.38	97.96	97.91	
Kappa Coefficient: <b>0.971</b>						Overall Accuracy: <b>97.59%</b>				

Table 6. The classification error matrix for data set IPS of 16 classes (in percentage).

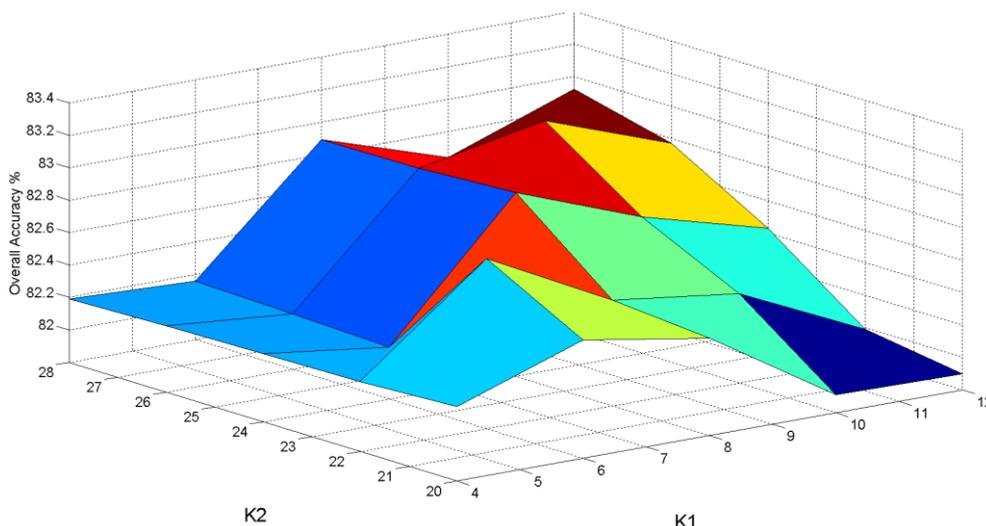
	Reference Data																UA
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1	78.22	0	0	0	4.35	0	0	17.43	0	0	0	0	0	0	0	0	78.22
2	0	77.15	2.22	0.69	0	0.18	0	0	0.07	5.17	13.26	1.19	0	0	0.07	0	77.15
3	0	3.32	73.03	3.04	0	0	0	0	0	0.71	15.15	4.75	0	0	0	0	73.03
4	0	13.91	8.84	65.83	0.42	0	0	0.82	0	1.29	7.59	1.29	0	0	0	0	65.83
5	0	0.21	0.23	0.24	94.61	0.22	0	0	0	0.80	0.81	1.04	0	1.85	0	0	94.61
6	0	0.12	0.14	0	0.19	97.11	0	0	0	0.68	0	0	0.58	1.18	0	0	97.11
7	0	0	0	0	3.61	0	92.81	3.58	0	0	0	0	0	0	0	0	92.81
8	1.81	0	0	0	0	0	0	98.19	0	0	0	0	0	0	0	0	98.19
9	0	0	0	0	0	0	0	0	94.99	0	0	0	5.01	0	0	0	94.99
10	0	3.83	0.31	0	0.32	0.31	0	0	0.11	81.74	12.95	0.43	0	0	0	0	81.74
11	0	4.62	3.52	0.22	0.32	0.31	0	0	0.08	5.45	83.95	1.37	0	0	0.16	0	83.95
12	0	4.93	7.93	0.61	0.12	0.14	0	0	0	2.05	9.79	74.25	0	0	0.17	0	74.25
13	0	0	0	0	0	0	0	0	0	0	0	0.49	99.51	0	0	0	99.51
14	0	0	0	0	0.47	0.08	0	0	0	0	0	0	0.08	96.03	3.34	0	96.03
15	0	0	0.54	0.54	7.25	15.02	0	0	0.20	1.85	2.59	0.25	0.25	16.85	54.66	0	54.66
16	0	1.03	0	0	0	0	0	0	0	1.06	3.28	0	0	0	0	94.63	94.63
PA	97.73	70.70	75.47	92.49	84.73	85.65	1	81.81	99.51	81.64	56.21	87.29	94.90	83.27	91.74	1	
Kappa Coefficient: <b>0.826</b>						Overall Accuracy: <b>83.85%</b>											

UA: User's Accuracy, PA: Producer's Accuracy

In this study, since we focused on the performance of kernelization and fuzzification, the  $k$ -NN classifier was adopted rather than the complex support vector machine (SVM) classifier. An analysis of various  $k$  values is given to demonstrate the performance of the  $k$ -NN classifier as shown in Table 7. Here, value  $k$  was set as values 1, 3, and 4, and the voting strategy was used in this analysis. Obviously, an adaptive higher value of the  $k$ -NN classifier can achieve more competitive performances. Next, the empirical parameters  $K_1, K_2, K_3$ , and  $K_4$  were properly determined by a cross-validation technique. Training samples were separated into two groups: the training and validation subsets, where, for example, 50% of the samples for training and the other for validation. The validation results were generated under various parameters, and the proper setting was determined by selecting the best results. From the cross-validation experiment, the proper parameters  $K_1 = 8$ ,  $K_2 = 28$ ,  $K_3 = 14$ , and  $K_4 = 28$  were chosen. After that, the transformation was obtained from the whole training set. A sensitivity analysis on four parameters  $K_1, K_2, K_3$ , and  $K_4$  was done as shown in Figure 10. In Figure 10a, the variances of classification rates were relatively low for parameters  $K_1$  versus  $K_2$ . In contrast, from Figure 10b–f, parameters  $K_3$  and  $K_4$  resulted in a higher variance of classification rates. In other words, the NFLE parameters  $K_1$  and  $K_2$  were not sensitive to the classification rates, and the parameters  $K_3$  and  $K_4$  of fuzzy  $k$  nearest neighbor were sensitive to the classification rates. According to the results of the sensitivity analysis of the four parameters, the parameters selected in the proposed algorithm were  $K_1 = 8$ ,  $K_2 = 28$ ,  $K_3 = 14$ , and  $K_4 = 28$ , which are consistent with the parameters in the cross-validation test.

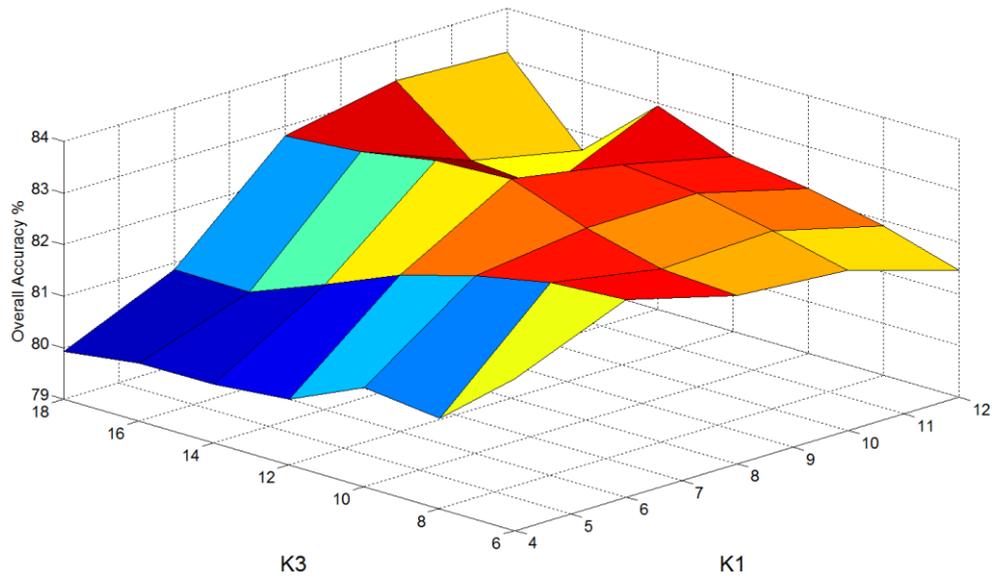
**Table 7.** The classification performance using various  $k$ -NN for data set IPS-10 (in percentage).

	FKNFLE			KNFLE			FNFLE			NFLE		
	<i>k</i> -Value			<i>k</i> -Value			<i>k</i> -Value			<i>k</i> -Value		
	1	3	4	1	3	4	1	3	4	1	3	4
IPS-10	83.34	84.19	85.11	83.07	83.55	84.19	78.37	78.98	79.10	77.59	78.89	78.93
Pavia City Center	97.59	98.18	98.24	96.55	96.84	96.88	95.08	95.32	95.51	94.58	95.26	95.41
Pavia University	91.31	92.13	92.36	89.50	90.04	90.19	85.10	86.05	86.57	83.80	84.63	85.05

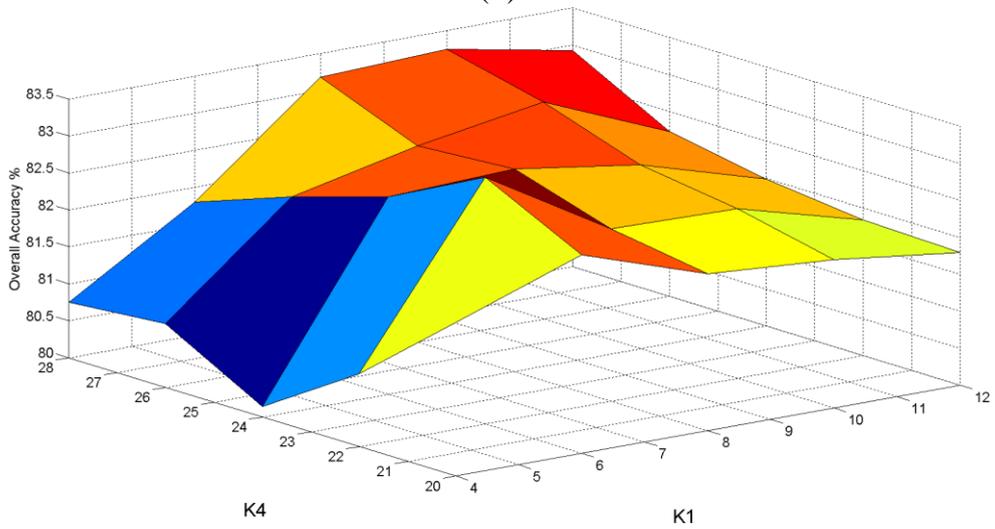


(a)

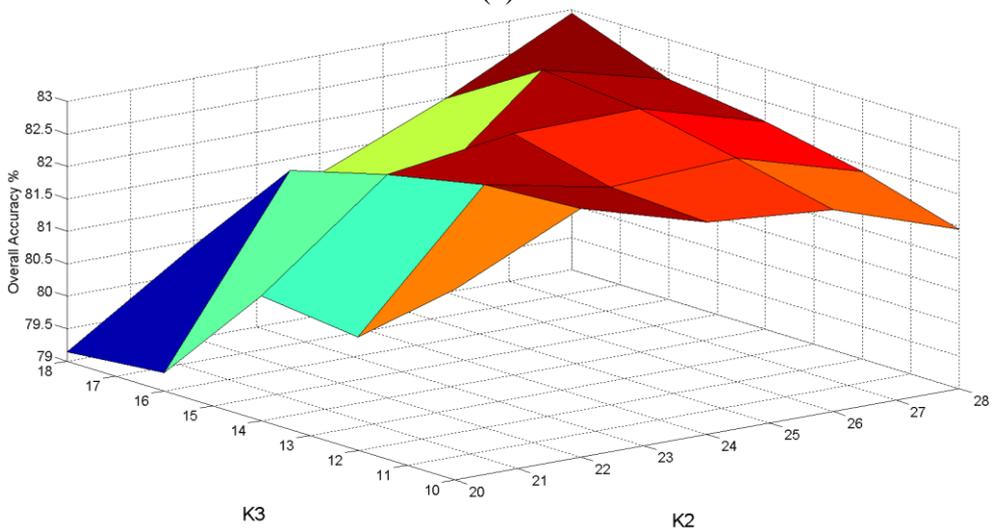
**Figure 10.** Cont.



(b)

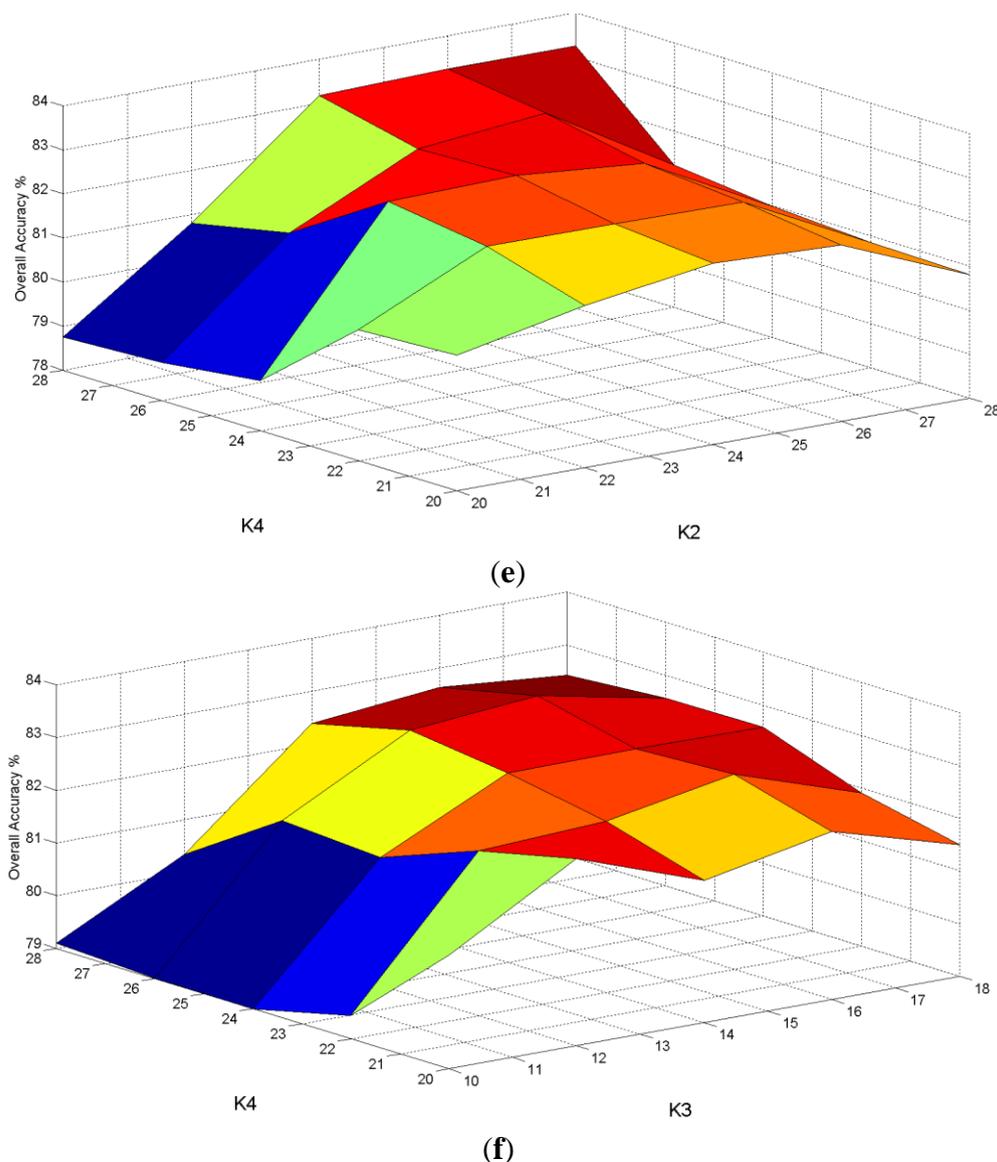


(c)



(d)

Figure 10. Cont.



**Figure 10.** The sensitivity analysis of four parameters  $K_1$ ,  $K_2$ ,  $K_3$ ,  $K_4$ . (a)  $K_1$  vs.  $K_2$ ; (b)  $K_1$  vs.  $K_3$ ; (c)  $K_1$  vs.  $K_4$ ; (d)  $K_2$  vs.  $K_3$ ; (e)  $K_2$  vs.  $K_4$ ; (f)  $K_3$  vs.  $K_4$ .

Furthermore, due to the proposed algorithm being based on kernelization and fuzzification, the performance comparison between the proposed algorithms and the well-known kernelization-based algorithm GCK-MLR (Generalized composite kernel-multinomial logistic regression) [17,30] is illustrated in Tables 8 and 9. Basically, algorithm GCK-MLR is a multinomial logistic regression (MLR)-based classifier of composite kernels in which four kernels, spectral, spatial, spectral-spatial cross information, and spatial-spectral cross information kernels, deeply impact the classification results. The training configurations in [17] were quite different from ours. Besides, it is unfair for comparing the results of a single kernel (KNFLE) method with those of multi-kernels (GCK-MLR). In the experiment, we re-trained the classifier using the same configurations of [17]. The training configurations and classification results have directly been referred from [17]. Moreover, only the results using a single spectral kernel  $K^o$  were used for the fair comparison. Datasets IPS of 16 classes and Pavia University were evaluated as shown in Tables 8 and 9, respectively. Considering the IPS dataset of 16 classes in Table 8, algorithm GCK-MLR outperforms the proposed method at the overall accuracy index, while

the average accuracy rate is lower than those of algorithms FKNFLE and KNFLE. In Table 9, the overall accuracy and average accuracy rates of the proposed method are both higher than those of algorithm GCK-MLR.

**Table 8.** The comparison between algorithm GCK-MLR ( $K^\omega$ ) and the proposed method for dataset IPS of 16 classes (in percent).

Class	Number of Samples		GCK-MLR ( $K^\omega$ )	FKNFLE	KNFLE
	Train	Test			
Alfalfa	3	51	47.06 ± 15.41	65.22 ± 15.32	56.52 ± 16.42
Corn-no till	71	1363	78.24 ± 3.01	70.66 ± 3.05	67.44 ± 3.03
Corn-min till	41	793	64.17 ± 3.01	67.71 ± 3.04	71.08 ± 3.05
Corn	11	223	48.211 ± 1.76	43.88 ± 11.54	47.68 ± 12.14
Grass/pasture	24	473	87.76 ± 2.27	84.47 ± 2.18	87.16 ± 2.58
Grass/tree	37	710	95.13 ± 1.40	96.58 ± 1.42	94.79 ± 1.32
Grass/pasture-mowed	3	23	53.04 ± 11.74	92.86 ± 11.88	92.82 ± 10.68
Hay-windrowed	24	465	98.84 ± 0.61	97.28 ± 0.59	98.12 ± 0.62
Oats	3	17	68.82 ± 17.33	65.10 ± 16.31	70.12 ± 15.35
Soybeans-no till	48	920	68.42 ± 5.22	70.27 ± 5.12	66.87 ± 5.42
Soybeans-min till	123	2245	82.56 ± 1.26	77.43 ± 1.31	73.93 ± 1.25
Soybeans-clean till	30	584	74.52 ± 5.35	62.56 ± 5.32	61.89 ± 5.52
Wheat	10	202	99.36 ± 0.52	91.71 ± 0.54	94.63 ± 0.51
Woods	64	1230	95.46 ± 1.53	96.60 ± 1.49	97.15 ± 1.54
Bldg-grass-tree-drives	19	361	50.75 ± 3.49	38.34 ± 3.18	48.19 ± 3.38
Stone-steel towers	4	91	62.09 ± 6.95	82.80 ± 6.89	83.87 ± 6.59
Overall accuracy			80.16 ± 0.73	77.19 ± 0.71	76.43 ± 0.73
Average accuracy			73.40 ± 1.26	75.22 ± 1.21	75.76 ± 1.25

**Table 9.** The comparison between algorithm GCK-MLR ( $K^\omega$ ) and the proposed method for dataset Pavia University of nine classes (in percent).

Class	Number of Samples		GCK-MLR ( $K^\omega$ )	FKNFLE	KNFLE
	Train	Test			
Asphalt	548	6631	82.64	83.14	82.64
Bare soil	540	18,649	68.62	82.89	82.07
Bitumen	392	2099	75.04	81.75	79.32
Bricks	524	3064	97.00	93.21	92.95
Gravel	265	1345	99.41	99.93	99.93
Meadows	532	5029	93.88	80.47	79.48
Metal Sheets	375	1330	90.08	92.26	92.41
Shadows	514	3682	91.36	85.61	85.17
Trees	231	947	97.57	99.89	99.89
Overall accuracy			80.34	85.76	84.04
Average accuracy			88.40	88.79	88.20

## 5. Conclusions

In this paper, a general NFLE transformation, FKNFLE, for HSI classification has been proposed. Kernelization and fuzzification in NFLE were both considered in order to extract non-linear and non-Euclidean structures. Three state-of-the-art algorithms, NFL, NRS and NRS-LFDA, were compared with the proposed FKNFLE. Three land-cover benchmarks, IPS-10, Pavia University, and Pavia City Center, were tested for performance evaluation. From the experimental results, algorithm FKNFLE outperformed the other algorithms. More specifically, using the 1-NN classifier, the rates of FKNFLE were higher than those of NFLE to the value of 5.75%, 3.01%, and 7.51% for datasets IPS-10, Pavia City Center, and Pavia University, respectively. Though FKNFLE had high classification rates using the features on a single pixel, there was some speckle-like noise in the image segmentation results. In the future, the features of spatial neighbors will be adopted for better classification and segmentation.

## Acknowledgement

The work was supported by Ministry of Science and Technology of Taiwan under Grant nos. MOST104-2221-E-008-030-MY3 and MOST103-2221-E-008-058-MY3.

## Author Contributions

The idea was conceived by Ying-Nong Chen and Chin-Chuan Han, performed by Ying-Nong Chen, Cheng-Ta Hsieh, and Ming-Gang Wen, analyzed by Chin-Chuan Han, Ying-Nong Chen, Cheng-Ta Hsieh, and Ming-Gang Wen, written by Chin-Chuan Han and Ying-Nong Chen, and revised by Ying-Nong Chen, Chin-Chuan Han and Kuo-Chin Fan.

## Conflicts of Interest

We have no conflict of interest to declare.

## References

1. Turk, M.; Pentland, A.P. Face recognition using eigenfaces. In Proceedings of the 1991 Proceedings CVPR '91. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Maui, HI, USA, 3–6 June 1991; pp. 586–591.
2. Belhumeur, P.N.; Hespanha, J.P.; Kriegman, D.J. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1997**, *19*, 711–720.
3. Cevikalp, H.; Neamtu, M.; Wikes, M.; Barkana, A. Discriminative common vectors for face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 4–13.
4. Prasad, S.; Mann Bruce, L. Information fusion in kernel-induced spaces for robust subpixel hyperspectral ATR. *IEEE Trans. Geosci. Remote Sens. Lett.* **2009**, *6*, 572–576.
5. He, X.; Yan, S.; Ho, Y.; Niyogi, P.; Zhang, H.J. Face recognition using Laplacianfaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 328–340.
6. Tu, S.T.; Chen, J.Y.; Yang, W.; Sun, H. Laplacian eigenmaps-based polarimetric dimensionality reduction for SAR image classification. *IEEE Trans. Geosci. Remote Sens.* **2011**, *50*, 170–179.

7. Wang, Z.; He, B. Locality preserving projections algorithm for hyperspectral image dimensionality reduction. In Proceedings of the 2011 19th International Conference on Geoinformatics, Shanghai, China, 24–26 June 2011; pp. 1–4.
8. Kim, D.H.; Finkel, L.H. Hyperspectral image processing using locally linear embedding. In Proceedings of the 1st International IEEE EMBS Conference on Neural Engineering, Italy, 20–22 March 2003; pp. 316–319.
9. Li, W.; Prasad, S.; Fowler, J.E.; Bruce, L.M. Locality-preserving discriminant analysis in kernel-induced feature spaces for hyperspectral image classification. *IEEE Geosci. Remote Sens. Lett.* **2011**, *8*, 894–898.
10. Li, W.; Prasad, S.; Fowler, J.E.; Bruce, L.M. Locality-preserving dimensionality reduction and classification for hyperspectral image analysis. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 1185–1198.
11. Luo, R.B.; Liao, W.Z.; Pi, Y.G. Discriminative supervised neighborhood preserving embedding feature extraction for hyperspectral-image classification. *Telkomnika* **2012**, *10*, 1051–1056.
12. Zhang, L.; Zhang, Q.; Zhang, L.; Tao, D.; Huang, X.; Du, B. Ensemble manifold regularized sparse low-rank approximation for multi-view feature embedding. *Pattern Recognit.* **2015**, *48*, 3102–3112.
13. Boots, B.; Gordon, G.J. Two-manifold problems with applications to nonlinear system Identification. In Proceedings of the 29th International Conference on Machine Learning, Edinburgh, UK, 26 June–1 July 2012.
14. Odone, F.; Barla, A.; Verri, A. Building kernels from binary strings for image matching. *IEEE Trans. Image Process.* **2005**, *14*, 169–180.
15. Scholkopf, B.; Smola, A.; Muller, K.R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Comput.* **1998**, *10*, 1299–1319.
16. Lin, Y.Y.; Liu, T.L.; Fuh, C.S. Multiple kernel learning for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1147–1160.
17. Li, J.; Reddy Marpu, P.; Plaza, A.; Bioucas-Dias, J.M.; Atli Benediktsson, J. Generalized composite kernel framework for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 4816–4829.
18. Chen, Y.; Nasrabadi, N.M.; Tran, T.D. Hyperspectral image classification via kernel sparse representation. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 217–231.
19. Zhang, L.; Zhang, L.; Tao, D.; Huang, X. On combining multiple features for hyperspectral remote sensing image classification. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 879–893.
20. Chen, Y.N.; Han, C.C.; Wang, C.T.; Fan, K.C. Face recognition using nearest feature space embedding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 1073–1086.
21. Chang, Y.L.; Liu, J.N.; Han, C.C.; Chen, Y.N. Hyperspectral image classification using nearest feature line embedding approach. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 278–287.
22. Keller, J.J.M.; Gray, M.R.; Givens, J.A., Jr. A fuzzy k-nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern.* **1985**, *15*, 580–585.
23. Li, S.Z. Face recognition based on nearest linear combinations. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Santa Barbara, CA, USA, 23–25 June 1998; pp. 839–844.

24. Yan, S.; Xu, D.; Zhang, B.; Zhang, H.J.; Yang, Q.; Lin, S. Graph embedding and extensions: a framework for dimensionality reduction. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *29*, 40–51.
25. Li, W.; Tramel, E.W.; Prasad, S.; Fowler, J.E. Nearest regularized subspace for hyperspectral classification. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 477–489.
26. Chen, Y.N.; Han, C.C.; Fan, K.C. Use fuzzy nearest feature line embedding for hyperspectral image classification. In Proceedings of the 4th International Conference Earth Observations and Societal Impacts, Miaoli, Taiwan, 22–24 June 2014.
27. Lillesand, T.M.; Kiefer, R.W. *Remote Sensing and Image Interpretation*; Wiley: New York, NY, USA, 2000.
28. Sugiyama-Sato Lab at the University of Tokyo. Available online: <http://www.ms.k.u-tokyo.ac.jp/software.html> (accessed on 26 October 2015).
29. Github. Available online: <https://github.com/eric-tramel/NRSClassifier> (accessed on 15 May 2015).
30. IEEE Publications. Available online: <http://www.lx.it.pt/~jun/publications.html> (accessed on 15 May 2015).

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).