SUPERVISED SPATIALLY COHERENT NONLINEAR DIMENSIONALITY REDUCTION FOR HYPERSPECTRAL IMAGE CLASSIFICATION

Sergio D. Manzanarez and Vidya Manian, Member IEEE

ABSTRACT

This paper presents a supervised hyperspectral image classification method using spatially coherent nonlinear dimensionality reduction method. Local linear embedding is used to extract spatially and spectrally coherent regions in the image for feature extraction. A loss function is used to optimize the mapping of the original dataset to a d-dimensional manifold. Combining the modified Local linear embedding method with support vector machine gives improved classification accuracies.

Index Terms— Classification, hyperspectral images (HSI), segmentation, spatial coherence, local linear Embedding (LLE), supervised LLE (SLLE), spatial Coherence LLE (CLLE), supervised spatial coherence LLE (SCLLE).

1. INTRODUCTION

Analysis of hyperspectral images(HSIs) attracts considerable attention because it contains information of ground surface in a set of hundreds of narrow and contiguous spectral bands, which usually range from $0.4\mu m$ to $2.5\mu m$, spanning from visible to near-infrared region of the electromagnetic spectrum. This spectral information contained in a hyperspectral image enables better characterization and identification of targets. HSIs are widely used in a variety of environmental, agricultural, biological, geological, and surveillance applications, among others [1–3]. The limited availability of training samples present in hyperspectral images drastically impacts the performances of the classifier due to the high dimensionality of HSIs, which generates a problem for designing robust statistical estimations from HSIs. The feature extraction technique can be used as a solution to the posed problem [4]. The key idea of feature extraction is finding a set of vectors that have low dimensionality compared with the input data by transforming the hyperspectral dataset linearly or nonlinearly to another domain and extracting informative features in the new domain. In other words, given the input $X = \{\vec{\boldsymbol{x}}_1, \vec{\boldsymbol{x}}_2, \cdots, \vec{\boldsymbol{x}}_{\text{n\_pixels}} \in \mathbb{R}^{\text{n\_bands}}\}$ feature extraction seeks the low dimensional output $\widetilde{X} = \{\vec{\widetilde{\boldsymbol{x}}}_1, \vec{\widetilde{\boldsymbol{x}}}_2, \cdots, \vec{\widetilde{\boldsymbol{x}}}_{\text{n\_pixels}} \in \mathbb{R}^d\}$, where $d \ll$ n_bands, preserving the information of the classes. Some feature extraction methods combat the *curse of dimensionality*, by having more training samples to well fit a model in a higher-dimensional space. Feature extraction methods can be divided into two main categories; unsupervised and supervised feature extraction. The unsupervised feature extraction is usually used for data representation while supervised feature extraction is considered for solving the Hughes phenomenon [5].

2. METHODOLOGY

In this paper, an LLE embedding algorithm based feature extraction method using Spatial coherence in the image is applied. The idea of this work is to combine the ideas proposed on [6] and [7] with the aim of improving the classification accuracies of hyperspectral images.

2.1. Locally Linear Embedding

Locally Linear Embedding (LLE) [8] is an unsupervised nonlinear dimensionality reduction method based on the assumption that each sample $\vec{\boldsymbol{x}} \in \mathbb{R}^{\text{n\_bands}}$ can be approximated by a linear combination of some of its local neighbors. LLE seeks to extract the intrinsic geometry information of the data by reducing dimensionality while preserving local distances between its nearest neighbors. The locally linear embedding algorithm can be summarized in three steps:

1. Finding K neighbors or every vector $\vec{\boldsymbol{x}}_i$, i.e. compute $\{\vec{\boldsymbol{x}}_{i,1}, \vec{\boldsymbol{x}}_{i,2}, \cdots, \vec{\boldsymbol{x}}_{i,K} \in \mathbb{R}^{\text{n\_bands}}\}$

2. Computing local reconstruction weighted matrix $\boldsymbol{W}$ by minimizing cost function:

$$\varepsilon(\boldsymbol{W}) = \sum_{i=1}^{n\_pixels} \left\| \vec{\boldsymbol{x}}_i - \sum_{j=1}^{K} w_{ij} \vec{\boldsymbol{x}}_{i,j} \right\| \tag{1}$$

   such that $\sum_j w_{ij} = 1$.

3. Mapping the original dataset to d-dimensional manifold coordinates. Finding the vectors $\vec{\widetilde{\boldsymbol{x}}}_i$ to minimize the following cost function:

$$\frac{1}{2} \sum_{i=1}^{\text{n\_pixels}} \left\| \vec{\widetilde{\boldsymbol{x}}}_i - \sum_{j=1}^{K} w_{ij} \vec{\widetilde{\boldsymbol{x}}}_j \right\|^2 \tag{2}$$

subject to

$$\frac{1}{N}\sum_{i=1}^{\text{n\_pixels}} \vec{\tilde{\boldsymbol{x}}}_i \vec{\tilde{\boldsymbol{x}}}_i^T = I \quad \text{and} \quad \sum_{i=1}^{\text{n\_pixels}} \vec{\tilde{\boldsymbol{x}}}_i = \vec{0}.$$

2.2. Spatial Coherence in Locally Linear Embedding

This algorithm was studied and presented by A. Mohan, G. Sapiro and E. Bosh in [7]. The authors extend the LLE algorithm by introducing a spatial coherence in the computation of high dimensional neighborhoods. In their algorithm, pixels are compared based on their local $n_{\text{sur}} \times n_{\text{sur}}$ immediate surrounding patch in the image domain. The authors mention that in classical LLE, the Euclidean distance is used to compute the K closest local neighbors $\{\vec{\boldsymbol{x}}_{i,1}, \vec{\boldsymbol{x}}_{i,2}, \cdots, \vec{\boldsymbol{x}}_{i,K} \in \mathbb{R}^{\text{n\_bands}}\}$ to a point $\vec{\boldsymbol{x}}_i$. Instead of using the individual pixels of each band to compute Euclidean distance, they propose to use all the $n_{\text{sur}} \times n_{\text{sur}}$ surrounding pixels (an $n_{\text{sur}} \times n_{\text{sur}}$ square in the image plane with $\vec{\boldsymbol{x}}_i$ as its centers). They propose that in the first step of LLE each pixel $\vec{\boldsymbol{x}}_i$ be replaced by a higher-dimensional vector for computing its K-neighbors $\{\vec{\boldsymbol{x}}_{i,1}, \vec{\boldsymbol{x}}_{i,2}, \cdots, \vec{\boldsymbol{x}}_{i,K} \in \mathbb{R}^{\text{n\_bands}}\}$. The new vector $\vec{\boldsymbol{x}}_i^{\text{new}}$ is built by replacing each band of $\vec{\boldsymbol{x}}_i$ for an n_{sur}^2 dimensional vector which is formed by its $n_{\text{sur}} \times n_{\text{sur}}$ surrounding pixels. They claim that if two pixels $\vec{\boldsymbol{x}}_i$ and $\vec{\boldsymbol{x}}_j$ are very similar, but their respective surrounding $n_{\text{sur}} \times n_{\text{sur}}$ image patches are dissimilar, then the distance between the two new pixel vectors $\vec{\boldsymbol{x}}_i^{\text{new}}$, $\vec{\boldsymbol{x}}_j^{\text{new}}$ would be high, and the original pixels would not be considered neighbors and will not be jointly used to estimate the LLE. Also, the new distance between the new vectors take into account the spatial coherence between two pixel vectors".

2.3. Supervised Locally Linear Embedding

Both de Ridder et al in [9] and Zhang et al in [10] proposed supervised versions of LLE. In this paper, we have focused on a version of supervised LLE (SLLE) presented by Thomas Boucher, C. Carey, S. Giguere, S. Mahadevan, M. Dyar in [6]. In general, the idea of supervised LLE is to add a penalty term to the loss function in the third step (equation 2) which draws spectra of similar compositions closer together in the embedding space.

In this paper, we have used the Supervised LLE version in [6] but with a modification. The SLLE studied shares the same first two steps as LLE, i.e every sample's nearest neighbors are calculated and then the sample is linearly reconstructed from those neighbors. The difference is that, we do one hot encoding for the labels of the hyperspectral image. The RBF kernel matrix $K \in \mathbb{R}^{n_{\text{pixels}} \times n_{\text{pixels}}}$ is then defined as:

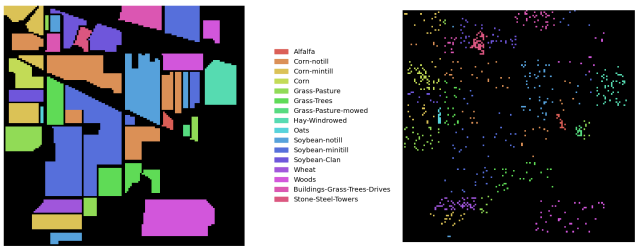
$$K(i.j) = \begin{cases} 1 & \text{if} \quad y_i = y_j \\ \alpha \quad \text{s.t.} \quad 0 < \alpha \ll 1 & \text{if} \quad y_i \neq y_j \end{cases} \tag{3}$$

The authors agree that K can be viewed as a similarity score between two samples within the range $(0, 1]$. The projection step is defined in a similar way. The vectors $\vec{\tilde{x}}_i$ that minimize the following loss function are found:

$$\min_{\widetilde{\boldsymbol{X}}} \frac{1}{2}\sum_{i=1}^{\text{n\_pixels}} \left\| \vec{\tilde{\boldsymbol{x}}}_i - \sum_{j=1}^{K} w_{ij} \vec{\tilde{\boldsymbol{x}}}_j \right\|^2 + \frac{\mu}{2} \sum_{i=1}^{\text{n\_pixels}} \sum_{j=1}^{\text{n\_pixels}} \left\| \vec{\tilde{x}}_i - \vec{\tilde{x}}_j \right\|^2 K(i,j)$$

3. HYPERSPECTRAL DATASETS

The experiments were conducted on the Indian Pines and Pavia University dataset. **Indian Pines** hyperspectral image dataset was captured using the Airborne/Visible Infrared Imaging Spectrometer (AVIRIS) sensor over the Indian Pines test site in northwestern Indiana. The HSI consists of 224 spectral reflectance bands of 145×145 pixels each, in the wavelength range 0.4 to 2.5μm. The number of bands is reduced to 200 by removing bands covering regions of high water absorption: 104-108, 150-163, and 220. The ground truth image consists of a total of 10,249 pixels, distributed between sixteen classes and are not all mutually exclusive, as shown in **Figure 1** . **Pavia University** was acquired by reflective optics system imaging spectrometer (ROSIS). The image is of 610×340 pixels covering the Engineering School at the University of Pavia, which was collected under the HySens project managed by the German Aerospace Agency (DLR). The ROSIS-03 sensor comprises 115 spectral channels ranging from 430 to 860 nm. In this dataset, 12 noisy channels have been removed and the remaining 103 spectral channels are used. The spatial resolution is 1.3 m per pixel. The available training samples of this dataset cover nine classes of interests, as shown in **Figure 2**. The number of training and test samples are displayed in Table 1 for Indian Pines and Pavia University datasets.

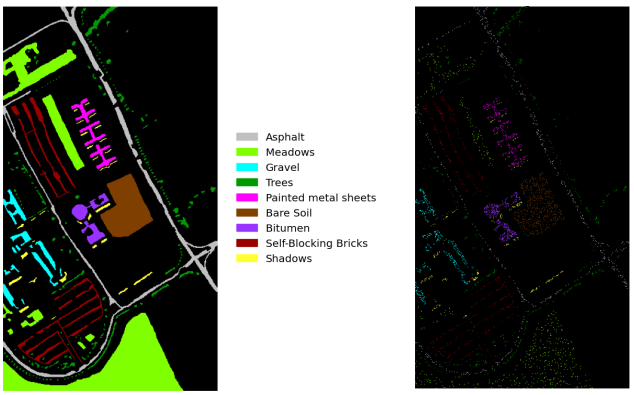


(a) Ground Truth (b) Training data

Fig. 1: Indian Pine Ground-Truth map containing 16 land-cover classes and the training dataset used.

4. EXPERIMENTAL RESULTS

Similar to [7] the Indian Pines image has been projected to 25 and 100 bands. The number of LLE neighbors are varied



(a) Ground Truth (b) Training data

Fig. 2: Pavia Univerity Ground-Truth map containing 9 land-cover classes and the training dataset used.

Table 1: Number of training and test samples of dataset.

Indinan Pines			Pavia University		
Class Name	Training	Test	Class Name	Training	Test
Alfalfa	15	31	Asphalt	548	6083
Corn-notill	50	1378	Meadows	540	18109
Corn-mintill	50	780	Gravel	392	1707
Corn	50	187	Trees	524	2540
Grass-Pasture	50	433	Painted metal sheets	265	1080
Grass-Trees	50	680	Bare Soil	532	4497
Grass-Pasture-mowed	15	13	Bitumen	375	955
Hay-Windrowed	50	428	Self-Blocking Bricks	514	3168
Oats	15	5	Shadows	231	716
Soybean-notill	50	922	-	-	-
Soybean-minitill	50	2405	-	-	-
Soybean-Clan	50	543	-	-	-
Wheat	50	155	-	-	-
Woods	50	1215	-	-	-
Buildings-Grass-Trees-Drives & 50	50	336	-	-	-
Stone-Steel-Towers	50	43	-	-	-
Total	695	9554	Total	3921	38855

between 10 and 25. In Table 2 to 5 the classification performance of SVM on Indian Pines dataset can be seen. First, the feature extraction methods was applied to the whole labeled data, then the whole raw data was transformed with the embeddings generated with this feature method and the SVM classifier was trained with the corresponding training samples.

Table 3: Accuracy comparison for Indian Pines using Support Vector Machine and LLE parameter $K = 10, d = 100$

Class No.	Class Name	LLE+SVM	SLLE+SVM	CLLE+SVM			SCLLE+SVM		
				$n_{sur} = 3$	$n_{sur} = 5$	$n_{sur} = 7$	$n_{sur} = 3$	$n_{sur} = 5$	$n_{sur} = 7$
1	Alfalfa	25.8	9.7	16.1	40.9	0	16.1	6.5	9.7
2	Corn-notill	36.1	100	35.1	46.6	54.6	100	100	100
3	Corn-mintill	48.7	99	47.8	50.9	70.5	100	100	100
4	Corn	66.8	66.3	65.8	73.3	72.7	81.8	92.0	88.2
5	Grass-Pasture	79.9	99.1	92.6	89.1	87.3	99.3	99.1	100
6	Grass-Trees	84.7	99.9	85.9	86.2	69.3	100	100	100
7	Grass-Pasture-mowed	69.2	23.1	0	84.6	92.3	15.4	7.7	15.4
8	Hay-Windrowed	81.1	98.4	96.7	90.2	100	99.8	100	100
9	Oats	60.0	20.0	40.0	80.0	80.0	40.0	20.0	80.0
10	Soybean-notill	45.2	100	53.1	56.9	62.4	100	100	100
11	Soybean-minitill	32.6	100	60.7	71.9	78.0	100	100	100
12	Soybean-Clan	44.9	98.5	53.4	53.2	72.4	99.3	100	100
13	Wheat	95.5	79.4	98.7	98.1	98.7	85.2	63.2	72.9
14	Woods	88	100	91.4	71.1	65.8	100	100	100
15	Buildings-Grass-Trees-Drives	23.2	90.5	28.3	27.4	61.0	95.2	96.7	98.8
16	Stone-Steel-Towers	83.7	90.7	81.4	90.7	100	100	100	100
OA	-	56.06	97.90	62.98	65.43	70.92	98.73	98.62	98.87
AA	-	60.30	79.70	59.20	69.5	72.80	83.30	80.30	85.30
Kappa	-	0.4761	0.9758	0.5798	0.6034	0.6671	0.9854	0.9841	0.9870

Table 2: Accuracy comparison for Indian Pines using Support Vector Machine and LLE parameter $K = 10, d = 25$

Class No.	Class Name	LLE+SVM	SLLE+SVM	CLLE+SVM			SCLLE+SVM		
				$n_{sur} = 3$	$n_{sur} = 5$	$n_{sur} = 7$	$n_{sur} = 3$	$n_{sur} = 5$	$n_{sur} = 7$
1	Alfalfa	41.9	32.3	0	16.1	0	61.3	100	96.8
2	Corn-notill	28.2	78.0	14.6	21.8	28.7	99.6	100	100
3	Corn-mintill	23.5	63.8	41.7	27.1	49.1	100	100	100
4	Corn	30.5	83.4	56.1	59.9	71.1	98.9	99.5	99.5
5	Grass-Pasture	77.8	99.8	85.9	31.2	86.6	99.8	99.5	100
6	Grass-Trees	83.1	99.6	83.8	86.6	91.3	100	100	100
7	Grass-Pasture-mowed	69.2	38.5	0.0	84.6	92.3	100	84.6	92.3
8	Hay-Windrowed	66.1	97.4	99.5	100	100	100	100	100
9	Oats	0.0	100	60.0	0.0	0.0	100	80.0	80.0
10	Soybean-notill	55.4	93.3	45.2	60.1	77.1	100	100	100
11	Soybean-minitill	25.4	99.9	63.7	60.1	17.2	100	100	100
12	Soybean-Clan	34.1	77.3	33	28.9	47.1	99.1	100	100
13	Wheat	94.2	99.4	98.7	98.1	98.7	99.4	100	99.4
14	Woods	73.8	100	80.3	90	56	100	100	100
15	Buildings-Grass-Trees-Drives	14.0	72	14.3	22.3	17.9	85.1	98.2	99.4
16	Stone-Steel-Towers	81.4	90.7	79.1	93	100	95.3	100	100
OA	-	44.58	90.10	55.89	55.57	48.83	99.17	99.87	99.93
AA	-	49.80	82.8	53.5	55.00	58.30	96.20	97.60	98.00
Kappa	-	0.3830	0.8853	0.4965	0.4920	0.4341	0.9905	0.9986	0.9992

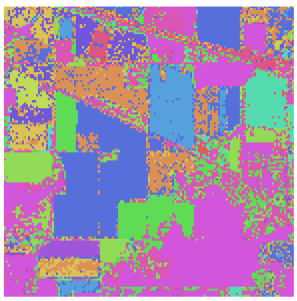
Table 4: Accuracy comparison for Indian Pines using Support Vector Machine and LLE parameter $K = 25, d = 25$

Class No.	Class Name	LLE+SVM	SLLE+SVM	CLLE+SVM			SCLLE+SVM		
				$n_{sur} = 3$	$n_{sur} = 5$	$n_{sur} = 7$	$n_{sur} = 3$	$n_{sur} = 5$	$n_{sur} = 7$
1	Alfalfa	0	54.8	0	0	64.5	80.6	71.0	90.3
2	Corn-notill	25.5	12.5	43.1	23.4	27.0	99.9	99.9	100
3	Corn-mintill	46.3	38.5	42.1	37.4	43.7	99.9	100	99.9
4	Corn	48.1	92.0	63.6	73.8	61.5	98.9	99.5	98.9
5	Grass-Pasture	77.8	99.8	89.8	73.9	74.6	99.3	99.8	99.8
6	Grass-Trees	87.6	99.9	91.9	93.4	95.9	99.9	100	100
7	Grass-Pasture-mowed	23.1	30.8	76.9	30.8	69.2	92.3	92.3	76.9
8	Hay-Windrowed	98.4	99.8	95.6	99.3	99.3	100	100	100
9	Oats	0	40	80	100	0	80	100	100
10	Soybean-notill	44.3	94.4	43.6	38.9	48.3	99.8	100	100
11	Soybean-minitill	26.8	99.0	55.3	65.2	67.0	100	100	100
12	Soybean-Clan	30.9	82.0	44.4	30	36.3	99.4	100	99.4
13	Wheat	94.8	99.4	98.7	98.1	99.4	99.4	100	100
14	Woods	79.6	100	87.0	78.0	85.6	100	100	100
15	Buildings-Grass-Trees-Drives	21.7	97.6	17.3	23.2	19.3	96.4	98.5	99.1
16	Stone-Steel-Towers	86	95.3	86.0	90.7	90.7	97.7	100	100
OA	-	48.19	79.97	60.24	57.03	60.80	99.63	99.81	99.83
AA	-	49.40	77.20	63.50	59.80	61.40	96.5	97.60	97.80
Kappa	-	0.4248	0.7647	0.5501	0.5084	0.5503	0.9958	0.9978	0.9981

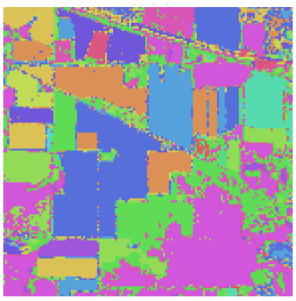
Table 5: Accuracy comparison for Indian Pines using Support Vector Machine and LLE parameter $K = 25, d = 100$

Class No.	Class Name	LLE+SVM	SLLE+SVM	CLLE+SVM			SCLLE+SVM		
				$n_{sur} = 3$	$n_{sur} = 5$	$n_{sur} = 7$	$n_{sur} = 3$	$n_{sur} = 5$	$n_{sur} = 7$
1	Alfalfa	19.4	9.7	25.8	61.3	77.4	9.7	12.9	9.7
2	Corn-notill	35.2	100	43.6	41.9	50.5	100	100	100
3	Corn-mintill	44.1	99.2	55.8	51.0	61.4	100	100	99.7
4	Corn	66.3	87.2	74.9	77.0	89.8	82.4	94.1	88.8
5	Grass-Pasture	80.1	99.3	88.2	94.2	91.5	99.1	99.5	99.5
6	Grass-Trees	80.0	100	90.9	91.0	91.9	99.7	100	100
7	Grass-Pasture-mowed	53.8	0.0	69.2	84.6	92.3	7.7	30.8	46.2
8	Hay-Windrowed	86.0	98.8	91.8	94.2	97.7	99.8	99.8	100
9	Oats	60.0	20.0	60.0	100	80.0	60.0	20.0	100
10	Soybean-notill	48.6	99.5	50.1	53.0	59.1	99.8	100	100
11	Soybean-minitill	45.4	100	55.0	68.0	69.3	100	100	100
12	Soybean-Clan	45.5	99.1	51.0	60.4	56.2	99.3	100	99.4
13	Wheat	94.2	69.7	99.4	97.4	99.4	80.6	74.2	99.4
14	Woods	86.5	100	91.2	73.8	78.3	100	100	100
15	Buildings-Grass-Trees-Drives	44.0	93.8	31.0	31.5	36.3	94.3	96.4	96.7
16	Stone-Steel-Towers	90.7	95.3	93.0	100	100	97.7	100	97.7
OA	-	56.52	98.29	63.40	65.24	69.20	98.56	98.89	99.20
AA	-	61.20	79.50	66.90	73.70	77.00	83.10	83.00	89.80
Kappa	-	0.5085	0.9804	0.5851	0.6027	0.6481	0.9834	0.9872	0.9908

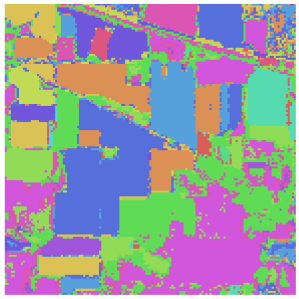
The classification maps of Indian Pines for the feature extraction methods are shown in Fig 3. An analysis of classification accuracy indicate that SCLLE improve the SCLLE by almost 10% for both Overall Accuracy (OA) and Average Accuracy (AA). With the application of spatial coherence, it is seen that large values for n_{sur} tend to make the classified image more smooth with lesser misclassification errors.



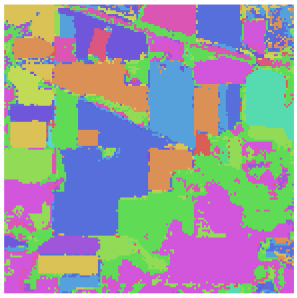
(a) $K = 10, d = 25$



(b)$K = 10, d = 25, n_{\text{sur}} = 3$

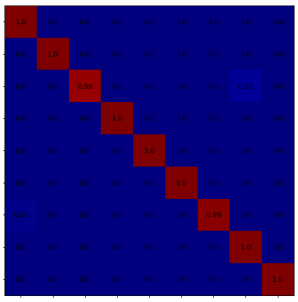


(c) $K = 10, d = 25, n_{\text{sur}} = 5$

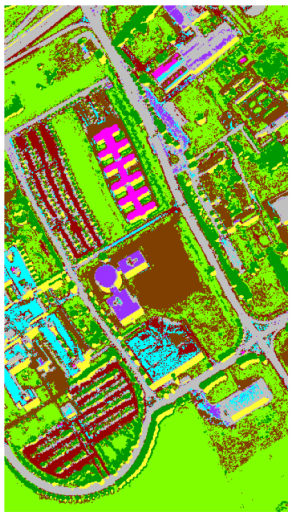


(d)$K = 10, d = 25, n_{\text{sur}} = 7$

Fig. 3: Classification results obtained by different feature extraction methods from Indian Pine scene. (a) SLLE + SVM , (b)-(d) SCLLE + SVM .



(a) Confusion Matrix



(b)$K = 10, d = 25, n_{\text{sur}} = 3$

Fig. 4: Classification results obtained by different feature extraction methods from Pavia University scene. (a) Confusion Matrix (b) SCLLE + SVM.

5. CONCLUSIONS

A spatially coherent LLE method is applied for feature extraction followed by optimization using a loss function for supervised hyperspectral image classification. This method has given improved classification accuracies for the the Indian Pine and Pavia datasets. The algorithm is computationally intensive and future work will involve speeding up the algorithm. Also, constrained optimization will be used for faster convergence of the method.

6. REFERENCES

[1] Laura M Dale, André Thewis, Christelle Boudry, Ioan Rotar, Pierre Dardenne, Vincent Baeten, and Juan A Fernández Pierna, "Hyperspectral imaging applications in agriculture and agro-food product quality and safety control: a review," *Applied Spectroscopy Reviews*, vol. 48, no. 2, pp. 142–159, 2013.

[2] Rick L Lawrence, Shana D Wood, and Roger L Sheley, "Mapping invasive plants using hyperspectral imagery and breiman cutler classifications (randomforest)," *Remote Sensing of Environment*, vol. 100, no. 3, pp. 356–362, 2006.

[3] Lorenzo Bruzzone and Roberto Cossu, "A multiple-cascade-classifier system for a robust and partially unsupervised updating of land-cover maps," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 9, pp. 1984–1996, 2002.

[4] Behnood Rasti, Danfeng Hong, Renlong Hang, Pedram Ghamisi, Xudong Kang, Jocelyn Chanussot, and Jon Atli Benediktsson, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep," *arXiv preprint arXiv:2003.02822*, 2020.

[5] Gordon Hughes, "On the mean accuracy of statistical pattern recognizers," *IEEE transactions on information theory*, vol. 14, no. 1, pp. 55–63, 1968.

[6] Thomas Boucher, C. Carey, S. Giguere, S. Mahadevan, and M. Dyar, "Manifold learning for regression of mars spectra," in ., 2015.

[7] Anish Mohan, Guillermo Sapiro, and Edward Bosch, "Spatially coherent nonlinear dimensionality reduction and segmentation of hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 4, no. 2, pp. 206–210, 2007.

[8] Sam T Roweis and Lawrence K Saul, "Nonlinear dimensionality reduction by locally linear embedding," *science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[9] Dick De Ridder, Olga Kouropteva, Oleg Okun, Matti Pietikäinen, and Robert PW Duin, "Supervised locally linear embedding," in *Artificial Neural Networks and Neural Information Processing—ICANN/ICONIP 2003*, pp. 333–341. Springer, 2003.

[10] Lingxiao Zhao and Zhenyue Zhang, "Supervised locally linear embedding with probability-based distance for classification," *Computers & Mathematics with Applications*, vol. 57, no. 6, pp. 919–926, 2009.