CrossMark

Spectral–spatial multi-feature-based deep learning for hyperspectral remote sensing image classification

Lizhe Wang[1] · **Jiabin Zhang**[2] · **Peng Liu**[3] · **Kim-Kwang Raymond Choo**[4] · **Fang Huang**[5]

Abstract Hyperspectral remote sensing has a strong ability in information expression, so it provides better support for classification. The methods proposed to deal the hyperspectral data classification problems were build one by one. However, most of them committed to spectral feature extraction that means wasting some valuable information and poor classification results. Thus, we should pay more attention to multi-features. And on the other hand, due to extreme requirements for classification accuracy, we should hierarchically explore more deep features. The first thought is machine learning, but the traditional machine learning classifiers, like the support vector machine, are not friendly to larger inputs and features. This paper introduces a hybrid of principle component analysis (PCA), guided filtering, deep learning architecture into hyperspectral data classification. In detail, as a mature dimension reduction architecture, PCA is capable of reducing the redundancy of hyperspectral information. In addition, guided filtering provides a passage to spatial-dominated information concisely and effectively. According to the stacked autoencoders which is a efficient deep learning architecture, deep-level multi-features are not in mystery. Two public data set PaviaU and Salinas are used to test the proposed algorithm. Experimental results demonstrate that the proposed spectral–spatial hyperspectral image classification method can show competitive performance. Multi-feature learning based on deep learning exhibits a great potential on the classification of hyperspectral images. When the number of samples is 30 % and the iteration number is over 1000, the accuracy rates for both of the two data set are over 99 %.

Keywords Multi-feature learning · Hyperspectral image classification · Deep learning · Remote sensing

✉ Peng Liu
liupeng@radi.ac.cn

1 School of Computer Science, China University of Geoscience, No. 388 Lumo Road, Wuhan 430074, People's Republic of China

2 School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, People's Republic of China

3 Institute of Remote Sensing and Digital Earth, Chinese Academy of Sciences, No. 9 Dengzhuang South Road, Beijing 100094, People's Republic of China

4 School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, Australia

5 School of Resources and Environment, University of Electronic Science and Technology of China, Chengdu 611731, People's Republic of China

1 Introduction

With the development of hyperspectral sensors and commercial systems, the hyperspectral image data have became the very important data source of remote sensing. Hyperspectral image usually has more than 100 spectral bands which represent the reflection of physical characteristics of different materials, which we called spectral information. With the development of science and technology, hyperspectral data mining has become a new multilateral area about observation and analysis of the Earth's surface (Richards and Jia 2013), including ecological environment, agriculture, mineralogy, physics, chemical imaging, astronomy and other aspects.

Pixel-wise classification was applied to many classification framework, and each pixel is a high-dimensional vector which contains a lot of spectral information. However, each coin has two sides; high dimensional also accompanies Hughes phenomenon (Ko et al. 2005). Feature extraction

reduces the dimension effectively, such as principal component analysis (PCA) and independent component analysis (ICA). PCA (Pearson 1901) retains most of the principal components (PCs), and ICA focuses on the independence of components. They retain the vast majority of spectral information and greatly reduce the dimension. So far, we have not considered the spatial context between pixel and pixel. Incorporating the spatial contextual information, we can research the strong relationship of pixels in a small window. Chen et al. (2014) proposed a "flatten neighbor region" model. It used all pixels in window as the spatial information of the center pixel. Useful but inefficient, due to the cost of "flatten" is a disaster when we choose a slightly larger window, despite the PCA out there. Recently, edge-preserving filtering (Kang et al. 2014) (EPF) has gained a lot of attention, since it smoothed the image under the premise of preserving edge information. This method can be used on each hyperspectral dimension and provide competitive performance on spatial information extraction. However, the filtering focuses on the results of SVM, not real raw spectral data. That makes EPF depending on the complexity of the image, and avoids the spatial information of pixels. Spatial information has been the important position in the classification of hyperspectral image. Tarabalka et al. (2010), Li et al. (2013), spatial–spectral representation is applied to hyperspectral data classification effectively. Multi-feature classification method provides significant improvement. Joint spectral and spatial information bring higher dimensions and complexity that proved the importance of reducing the dimension of data once again.

Traditional machine learning classifiers such as linear SVM (Melgani and Bruzzone 2004) (a single-layer classifier) and neural network (Mavrovouniotis and Yang 2015), Azar and Hassanien (2015) with very few layers are difficult to express complex features since they do not have sufficient hidden layers. Even for its improvement functions (Baassou et al. 2013), the overall structures are significantly limited. Furthermore, under finite samples and computing units, representation of the complex function is limited. The generalization ability of the complex classification problem is restricted. Thus, a deep architecture (Li et al. 2015) based on artificial neural networks has been proposed in recent years, with combinations of bottom level features. Deep learning Chen et al. (2014) is a new field in machine learning research. The motivation is to build and simulate the human brain to analyze and study the neural network, which imitates the human brain to interpret the data.

In this paper, our work focuses on applying guided filtering (He et al. 2013) to exploit spatial-dominated information and integrating spectral and spatial information. Then, we use multilayer fine-tuning stacked autoencoder(FSAE) to learn hyperspectral data features. The method is a member of the deep architecture-based models, combined with unsupervised deep feature learning and supervised feature fine

Table 1 Definition of all variables

Variables	Definition
$\mathbf{W}$	The weights of SAE
$\mathbf{W}'$	The weights of SAE
$\mathbf{b}_x$	The bias of SAE
$\mathbf{b}_y$	The bias of SAE
$\mathbf{h}$	The hidden layers
$\mathbf{x}$	The input vector data of AE
$\mathbf{y}$	The reconstructed vector data of AE
ω	The neighborhood window for guided filter
r	The semi-radius for window ω
$\mathbf{g}$	The guidance image for a guided filter
$\mathbf{G}$	$\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_N]$ are the guidance images when extracting spatial features
$\mathbf{U}$	$\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_N]$ are output image for a guided filter when extracting spatial features
Λ	The input data when extracting spatial features. $\Lambda = [\mathbf{I}_1, \ldots, \mathbf{I}_k, \ldots, \mathbf{I}_K]$, where $\mathbf{I}_k = \{I_{k,1}, \ldots, I_{k,n} \ldots, I_{k,N}\}$
$\mathbf{a}$	The linear coefficient for guide filter. It is vector if the input image is vector
c	The linear coefficient for guide filter. $\mathbf{c}$ is its vector form.
$\mathbf{P}$	The hyperspectral data set, and $\mathbf{x}_n \in \mathbf{P}$ is a pixel vector
S	The number of band of $\mathbf{P}$
s	The index of number of band of $\mathbf{P}$
$\mathbf{Z}$	It is equal to $\mathbf{P}$, and $\mathbf{z}_s \in \mathbf{Z}$ is a band image
ϵ	The regularization parameter for guided filter
$\mathbf{z}_s$	The sth band of hyperspectral data $\mathbf{Z}$
$\mathbf{d}_s$	The sth principal component of hyperspectral data $\mathbf{Z}$
$\mathbf{I}$	The input image for guided filter
I_i	The ith pixel input image $\mathbf{I}$
$f(x)$	The sigmoid function

Table 2 Number of training iterations comparison with the University of Pavia and the Salinas

Accuracy (%)	Number of training iterations					
Data sets	10	100	200	300	500	1000
PaviaU	82.52	99.21	99.03	99.57	99.49	99.51
Salinas	81.58	98.51	98.60	99.15	97.47	99.64

tuning. Finally, we establish a novel classification framework including previously mentioned methods. All the definitions of variables are listed in Table 1. Experiments show that the method has a high classification accuracy with much less computation burden (Tables 2, 3 and 4).

2 Related works

This high-resolution imagery obtained by remote sensors contains more detailed spectral and spatial information.

However, its higher resolutions do not directly result in higher classification accuracies. How to comprehensively explore and utilize the spectral–spatial information is still an open problem. For most current spectral–spatial methods, they usually have to face three major problems: how to extract spectral features, how to extract spatial features and how to combine with the multi-features.

About how to extract spectral features, there are the popular methods such as nonnegative matrix factorization (Wen et al. 2013), independent component analysis (Falco et al. 2014), rotation forest (Xia et al. 2014). For hyperspectral image, the redundancy of the spectral feature is usually high. Therefore, the extraction of spectral feature mainly involves in dimension reduction and feature selection schemes such as PCA, ICA, manifold learning (Lunga et al. 2014). In the current study of remote sensing classification, the spatial features are receiving more and more attentions. For spatial feature extraction methods, there is co-occurrence matrix (Pesaresi et al. 2008; Ouma et al. 2008) which is a texture extraction and representation method and differential morphological proles (Fauvel et al. 2008) which is based on open or close operator of structuring elements. Actually, most of the popular spatial domain filter such as guided filter (Kang et al. 2014), Markov random fields (Sun et al. 2015), nonlocal means (Liu et al. 2012) and wavelet (Quesada-Barriuso et al. 2014) all can be easily introduced into spectral–spatial classification of multi-channel remote sensing image.

It is a open problem about how to combine with spatial and spectral features in practice, although both of them are available. Usually, a combination of multiple features may show better classification performance. How to integrate multiple features for image classification is usually a feature selection problem. The most widely used multi-feature fusion approach is to concatenate multiple features into one vector and then interpret the vector via a classier (Huang and Zhang 2013). There is also some research that takes spectral and spatial features as different subspace (Yuan and Tang 2016). Most studies mentioned above are based on SVM classifier. Some of them may not be suitable for a deep architecture. For the popular deep leaning method, a framework that is a hybrid of principal component analysis, hierarchical learning-based feature extraction and logistic regression Chen et al. (2015) is proposed to promote the classification accuracy. It seems that spectral–spatial classification is very promising when it is introduced into deep architecture.

3 Sparse autoencoders (SAEs)

One of the fundamental difference between deep learning and the conventional artificial network model is how to construct its hidden layers. In this paper, SAE (Vincent et al. 2008)
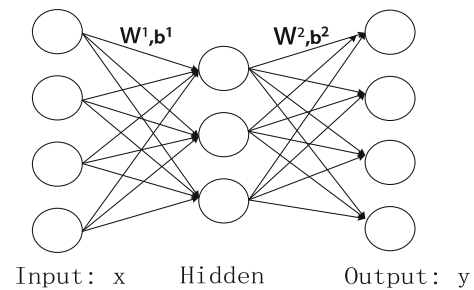

Fig. 1 AE with single layer. Hidden features are learned by input "*x*" and output "*y*"

as one of the popular autoencoders (AEs) is promoted and applied to spectral–spatial feature-based classification.

Autoencoder Tan and Eswaran (2008) is the extension of neural networks which is composed of many hierarchical single neurons. Figure 1 is an example of AE. It has three levels: the left side is the input layer with visible inputs, the middle is a hidden layer with appropriate hidden units, and the right is the output layer with outputs.

We define $\mathbf{P} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n, \ldots, \mathbf{x}_N\}$ as the hyperspectral data set, where $\mathbf{x}_n = \{x_{n,1}, \ldots, x_{n,S}\}$ is the nth pixel with S bands. To be convenient, we eliminate the index and use $\mathbf{x}$ when explaining SAE. In the training of SAE, first we "code" the input $\mathbf{x} \in R^S$ and obtain the activation $\mathbf{h}$. Then "decode" the $\mathbf{h}$ and obtain $\mathbf{y} \in R^S$. These two processes can be described as

$$\mathbf{h} = f\left(\mathbf{Wx} + \mathbf{b}_x\right) \tag{1}$$
$$\mathbf{y} = f\left(\mathbf{W'h} + \mathbf{b}_y\right) \tag{2}$$

where $\mathbf{W}$ and $\mathbf{W}'$ are weight associated with connection between the two neighbor layers. $\mathbf{b}_x$ and $\mathbf{b}_y$ are the bias. The activation function f in this paper is sigmoid function.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

The purpose is to find the most appropriate $\mathbf{W}$ and $\mathbf{b}_x$ to reduce the gap between input and output. In the pre-training stage, an autoencoder takes an input $\mathbf{x}$, then maps it to a hidden representation $\mathbf{h}$, using Eq. (1). Then the input is reconstructed using Eq. (2) where $\mathbf{y}$ is the reconstruction of $\mathbf{x}$ by $\mathbf{h}$. The cost function is:

$$L(\mathbf{x}, \mathbf{y}) = ||\mathbf{y} - \mathbf{x}||_2^2 + \lambda ||\mathbf{W}||_2^2 \tag{4}$$

The first term is a sum of squares error. The second term is regularization, and λ is the regularization parameter. The main differences between SAE and traditional ANN is the pre-training stage. Based on the basic autoencoder in Bengio et al. (2007) and KL-divergence regularization, the sparse
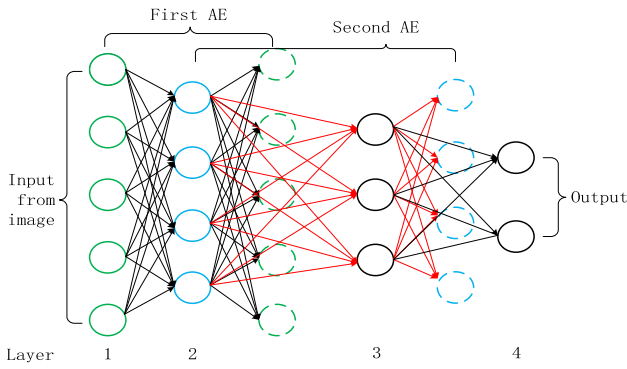
Fig. 2 A model of SAE with four layers: one input layer, two hidden layers and one output layer

coding was proposed by Vincent et al. (2008). The object function becomes

$$L(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_2^2 + \lambda||\mathbf{W}||_2^2 + \beta KL(\mathbf{x}||\mathbf{y}) \tag{5}$$

where

$$KL(\mathbf{x}||\mathbf{y}) = \sum_{k=1}^{d} [x_k \log y_k + (1 - x_k) \log(1 - y_k)] \tag{6}$$

β controls the weight of the sparsity penalty term. In the iteration, the gradient descent is used to update the parameters $\mathbf{W}$ and $\mathbf{b}_x$.

As shown in Fig. 2, the SAE is stacked by several AEs. Considering the concept of greedy layer-wise training, it means that the front AE's "activation" of hidden layer is taken as the next AE's inputs, so that a set of deeper parameters network can be obtained. Figure 2 shows a simple of SAE architecture. Therefore, the training process of SAE has two stages: unsupervised pre-training and supervised fine tuning. In the unsupervised pre-training, each layer is trained separately, which has input and hidden representation. Once we complete the pre-training of all layers, the fine-tuning step should be performed, which is a back-propagating (BP) (Rumelhart et al. 1986) step using supervised learning.

4 Guided filter

SAE deep network has good feature learning abilities. For hyperspectral remote sensing image, apart from spectral features we can also introduce spatial features into the SAE deep network classification. Therefore, how to select and organize the spatial features of a hyperspectral image is the key problem in utilizing multi-features in deep network. The guided filter has the characteristic of edge-preserving and focuses on a local linear model which considers that a point and its vicinity (in one function) have the linear relationship. Thus,

we assume in a local window with the radius r, then the size of the window is $\omega = (2r + 1) \times (2r + 1)$. Based on the idea of guided filtering (He et al. 2013), the pixel in guidance image $\mathbf{g}_i$ (the first three principal components are used in this paper) and the output single band image I_i has linear relationship in the local window ω. For the location with a local window ω, based on the linear relationship the output pixel u_i in the ω is

$$u_i = \mathbf{a}\mathbf{g}_i + c, \quad \forall i \in \omega \tag{7}$$

where $\mathbf{a}$ and c is the linear coefficient that transforms $\mathbf{g}_i$ to u_i. Since there is $\nabla u \approx a \nabla \mathbf{g}$, it proves the characteristic of edge-preserving. In order to get parameter $\mathbf{a}$ and c within the window ω, we hope the filtering output is as near as the input image pixel I_i in the corresponding location. It is to search the minimum value of following function:

$$E(\mathbf{a}, c) = \sum_{i \in \omega} ((\mathbf{a}\mathbf{g}_i + c - I_i)^2 + \epsilon \mathbf{a}^2) \tag{8}$$

where ϵ is the regularization parameter. After obtaining the linear coefficients $(\mathbf{a}, c)$, we can get the final u_i as

$$u_i = \frac{1}{|\omega|} \sum_{i \in \omega} (\mathbf{a}\mathbf{g}_i + c) \tag{9}$$

where $|\omega|$ means number of elements in ω. Guided filter is a good scheme for edge reservation. In constructing the input feature vector for deep network, the guided filter can strengthen the continuity of the spectral features based on the spatial texture of the hyperspectral image.

5 Spectral–spatial feature-based classification

Since $\mathbf{x}$ in Eq. (1) only contains spectral information, we construct a new $\mathbf{x}'$ to employ both spectral and spatial information. In this paper, the spatial information is extracted by guided filter. Before using guided filter, we need a multi-channel guidance image with spatial features and a group of input image with spectral information. Then the image data set needs to be decomposed into different components. PCA is used to decompose the whole image data set. After that, we only keep a small amount of principle components

$$[\mathbf{d}_1, \ldots, \mathbf{d}_S] = PCA(\mathbf{Z}) \tag{10}$$

where $\mathbf{Z}$ is equal to $\mathbf{P}$, but it is in its bands form of $\mathbf{Z} = \{\mathbf{z}_1, \ldots, \mathbf{z}_s, \ldots, \mathbf{z}_S\}$ which is the total image with all bands. $\mathbf{z}_s$ is the vector unfolded by a single band image. S is the number of bands. The guidance image $\mathbf{G}$ is constructed by the first three principal components
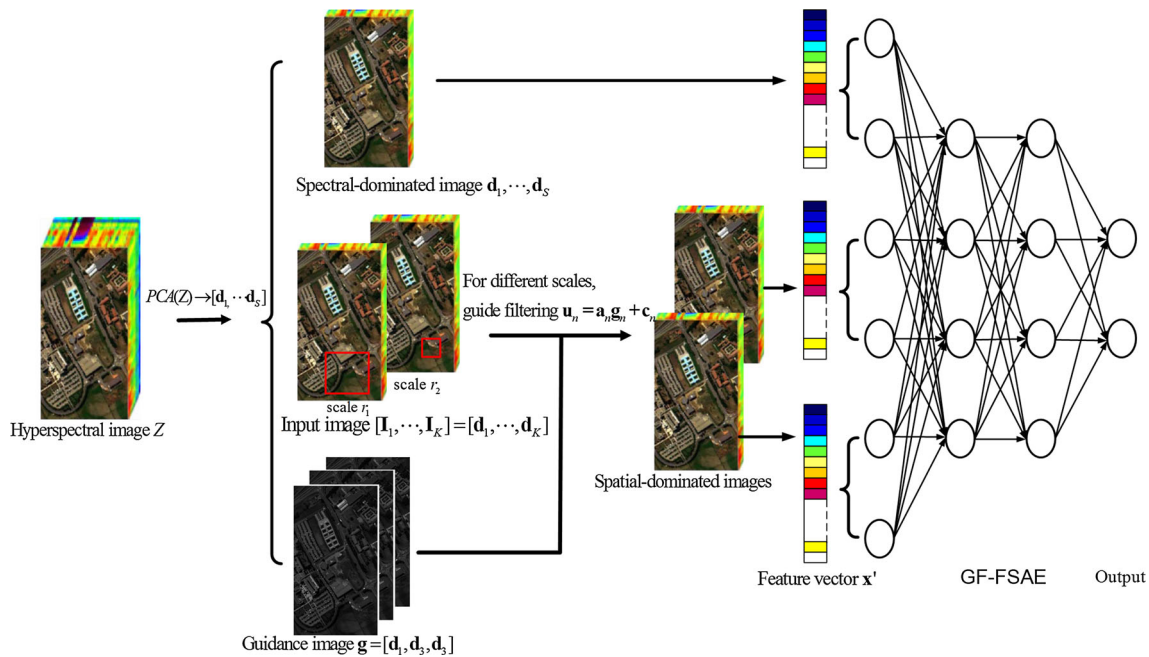
Fig. 3 Spectral–spatial feature classification framework. The original data are decomposed by PCA, and then, we obtain the spectral feature vector, input images for guided filter and guidance images for guided filter. Based on input images, guidance images and Eq. (8) we obtain

$$\mathbf{G} = [\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3] \tag{11}$$

Since $\mathbf{G}$ is $3 \times N$ matrix. We can also rewrite it as $\mathbf{G} = [\mathbf{g}_1, \ldots, \mathbf{g}_N]$. To extract spatial information, the first K principal components are selected as the input images Λ for the guided filter

$$\Lambda = [\mathbf{d}_1, \ldots, \mathbf{d}_K], \quad K < S \tag{12}$$

We can also rewrite $\Lambda = [\mathbf{I}_1, \ldots, \mathbf{I}_k, \ldots, \mathbf{I}_K]$, where image vector $\mathbf{I}_k = \{I_{k,1}, \ldots, I_{k,n} \ldots, I_{k,N}\}$. Based on Eq. (8), using input image Λ and guide image $\mathbf{G}$ we can obtain all the parameters $\{\mathbf{a}_{k,n}, c_{k,n}\}$. For each pixel vector $\mathbf{g}_n$, within its small neighborhood window ω_n, we have the output pixel $u_{k,n}$

$$u_{k,n} = \mathbf{a}_{k,n}\mathbf{g}_n + c_{k,n} \tag{13}$$

To be convenient, we eliminate k and use vector notation, and there is

$$\mathbf{u}_n = \mathbf{a}_n\mathbf{g}_n + \mathbf{c}_n \tag{14}$$

Now we have the output feature map $\mathbf{U} = \{\mathbf{u}_1, \ldots, \mathbf{u}_n, \ldots, \mathbf{u}_N\}$ by the input map Λ and guide map $\mathbf{G}$.

Since the number of component for filtered map $\mathbf{U}$ is $K \times N$ matrix, $\mathbf{u}_n = \{u_{n,1}, \ldots, u_{n,K}\}$ is the spatial feature vector. Similarly, we select the first F principal components from $[\mathbf{d}_1, \ldots, \mathbf{d}_S]$ as spectral feature vector which is $F \times N$ matrix

linear regression parameter $\mathbf{a}$ and $\mathbf{c}$. The spatial feature maps $\mathbf{u}$ are produced by the guidance image $\mathbf{g}$ and regression parameter $\mathbf{a}$ and $\mathbf{c}$. At last, integrate the spatial feature and spectral feature into $\mathbf{x}'$ and take it as the input of GF-FSAE

$\mathbf{B}$. We can rewrite it as $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n, \ldots, \mathbf{b}_N\}$, where $\mathbf{b}_n = \{b_{n,1}, \ldots, b_{n,F}\}$ is the spectral feature vector. Then the spectral–spatial feature vector is defined as

$$\mathbf{P}' = \left\{ \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{b}_1 \end{pmatrix}, \ldots, \begin{pmatrix} \mathbf{u}_n \\ \mathbf{b}_n \end{pmatrix}, \ldots, \begin{pmatrix} \mathbf{u}_N \\ \mathbf{b}_N \end{pmatrix} \right\} \tag{15}$$

To better utilize spatial features, we introduce spatial information with two different scales, which can give us more abundant spatial information. The different scales depend on the different window sizes r, so that we get two spatial feature matrixes with different neighbor information.

For the images of different scenes, we need to select different number F of the principal components as spatial information map and guidance image. After the integration, $\mathbf{x}'_n = [\mathbf{u}_n \quad \mathbf{b}_n]^{\mathrm{T}}$ contains spectrum information and spatial information. At last, we put the vectors of $\mathbf{x}'$ as SAE's input, training with labeled pixels. A deep classification framework is integrated and constructed. The whole procedure is shown in Fig. 3.

6 Experiments and results

6.1 Data description and experiment design

In our study, two hyperspectral data sets, i.e., the University of Pavia, the Salinas image, with different environmental
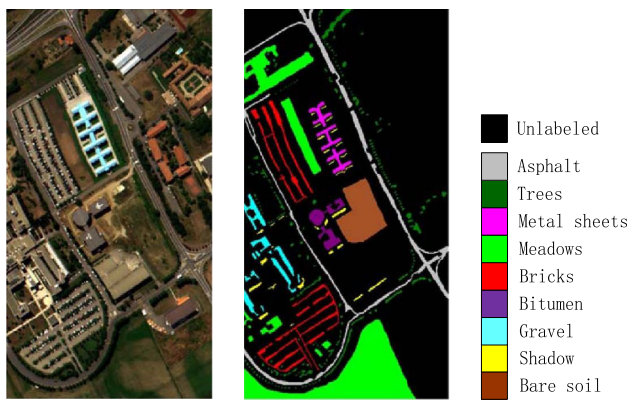
Fig. 4 Pavia, Italy, *three-band color* composite image and ground truth data image of the University of Pavia (color figure online)

Fig. 5 Salinas, USA, *three-band color* composite image and ground truth data image of the Salinas Valley (color figure online)

parameters are applied to validate proposed methods. The University of Pavia image was recorded by the reflective optics system imaging spectrometer (ROSIS-3) satellite sensor over the University of Pavia, with 610×340 pixels and 103 bands.

The image has a spatial resolution of 1.3 m per pixel and collected in the 0.43–0.86 μm range of spectral coverage. Nine material classes are labeled, which are shown in Fig. 4.

The Salinas image was acquired from Salinas Valley by the AVIRIS sensor, with 512×217 pixels and 224 bands, No. 108–112 154–167, and 224 were discarded because we can't handle the water absorption. Therefore, the bands of Salinas image now is 204. The image has a spatial resolution of 3.7 m per pixel. Figure 5 shows the composite of Salinas image and sixteen labeled material classes.

6.2 Classification results

In this section, the proposed GF-FSAE method is compared with some widely used and recently published classifica-
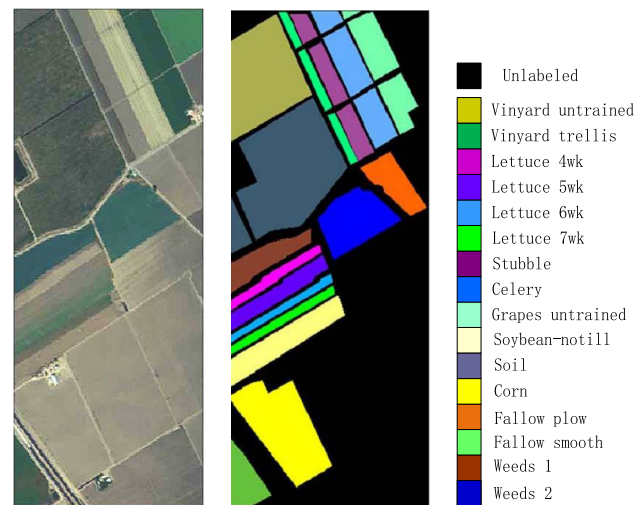
tion methods, SVM (Melgani and Bruzzone 2004), edge-preserving filtering (EPF) (Kang et al. 2014), SAE and SAE-LR (Chen et al. 2014). LIBSVM library (Chang and Lin 2011) is a mature SVM framework so that we can generate SVM classification results. The EPF method takes neighborhood information into account by directly filtering the results of SVM. EPF can ensure the smoothed probabilities aligning with real object boundaries and improve the classification accuracy significantly. The SAE-LR algorithm also uses the spectral–spatial thinking based on deep learning. In both images, we part the labeled samples into training data with 30 % randomly components and testing data with other 70 % components. The data set of training, testing samples and the classification results are detailed in Tables 3 and 4. We compare the capabilities of the proposed framework by three quality indexes, overall accuracy (OA), average accuracy (AA) and kappa coefficient. They show the accuracy

Table 3 Number of training and testing samples of the Pavia image and the classification accuracies

Class	Train	Test	SVM	EPF	SAE	SAE-LR	GF-FSAE
Asphalt	1989	4642	95.37	98.64	96.62	97.39	**99.29**
Meadow	5594	13,055	97.44	99.83	97.85	98.77	**99.88**
Gravel	629	1470	86.82	82.60	66.84	94.86	**98.63**
Tree	919	2145	97.64	97.77	96.68	99.43	**99.44**
Metal sheet	403	942	99.89	99.68	99.15	**1**	**1**
Bare soil	1508	3521	94.41	94.75	96.62	97.89	**99.77**
Bitumen	399	931	89.83	89.62	87.25	91.55	**99.24**
Bricks	1104	2578	89.02	91.33	93.55	94.37	**99.14**
Shadows	284	663	**1**	99.85	99.70	99.40	**1**
OA	–	–	95.44	99.28	95.78	97.76	**99.51**
AA	–	–	94.49	99.25	94.46	97.07	**99.35**
Kappa	–	–	93.95	99.04	94.40	97.03	**99.35**

Bold values indicate the best results for different methods

Table 4 Number of training and testing samples of the Salinas image and the classification accuracies

Class	Train	Test	SVM	EPF	SAE	SAE-LR	GF-FSAE
Weeds 1	602	1407	**1**	**1**	99.64	**1**	**1**
Weeds 2	1117	2609	99.85	**1**	99.58	99.89	99.92
Fallow	592	1384	**1**	**1**	99.57	98.78	**1**
Fallow P	418	976	99.39	98.49	98.58	**99.59**	99.08
Fallow S	803	1875	99.41	**1**	98.94	99.46	99.52
Stubble	1187	2772	**1**	**1**	**1**	**1**	**1**
Celery	1073	2506	**1**	**1**	99.96	99.76	**1**
Grapes	3381	7890	86.11	96.52	86.91	94.89	**99.66**
Soil	1860	4343	99.54	99.72	99.52	**1**	99.66
Corn	983	2295	98.59	99.52	96.37	98.31	**99.63**
Lettuce 4wk	320	748	98.93	**1**	99.43	99.60	**1**
Lettuce 5wk	578	1349	98.78	**1**	98.83	99.48	99.85
Lettuce 6wk	274	642	99.38	**1**	98.75	99.69	99.53
Lettuce 7wk	321	749	98.79	**99.73**	97.54	99.46	99.19
Vinyard U	2180	5088	85.06	99.57	85.47	91.77	**99.78**
Vinyard T	542	1265	99.61	**1**	99.76	99.84	99.61
OA	–	–	94.84	99.09	94.78	97.57	**99.64**
AA	–	–	97.78	99.60	97.43	98.78	**99.66**
Kappa	–	–	94.25	98.99	94.18	97.30	**99.59**

Bold values indicate the best results for different methods

Table 5 Parameters for different methods and different data sets

EPF		r_{EPF}	–	–	–	–
	PaviaU	2	–	–	–	–
	Salinas	2	–	–	–	–
SVM		Kernal	g	c	–	–
	PaviaU	RBF	1	100	–	–
	Salinas	RBF	0.125	100	–	–
SAE		–	–	Layers	Units	Iteration
	PaviaU	–	–	4	60	300
	Salinas	–	–	5	40	300
SAE-LR		r_{SAE-LR}	d	Layers	Units	Iteration
	PaviaU	2	3	4	60	300
	Salinas	3	4	5	40	300
GF-FSAE		r_1	r_2	Layers	Units	Iteration
	PaviaU	10	30	6	200	300
	Salinas	20	50	6	180	300

rates of classifications and the consistency between the classification results and real labels.

The parameters have a very direct impact on each model. We summarize all parameters in Table 5. In SVM, the kernel function is radial basis function (RBF), the semi-radius of the kernel function $g = 1$ and penalized parameters $c = 100$. For EPF algorithm, the window size of its spatial filter is $r_{EPF} = 2$ for both the two data sets. SAE-LR algorithm is based on deep learning, and it has propounded a neighbor region to obtain the spatial information. Reducing the dimension of original hyperspectral image by PCA, we can obtain a d-dimensional principal component where d is relatively small and accorded to the image attribute. Then, we define the neighbor region r_{SAE-LR}, also accorded to the image attribute. For the data of Pavia, we configure them as $r_{SAE-LR} = 2$ and $d = 3$, in data Salinas $r_{SAE-LR} = 3$ and $d = 4$. In training part, the neural networks for Pavia data have four layers and each layer has 60 units. The neural networks for Salinas data have five layers, and each layer has 40 units. For GF-FSAE algorithm, we determine 60
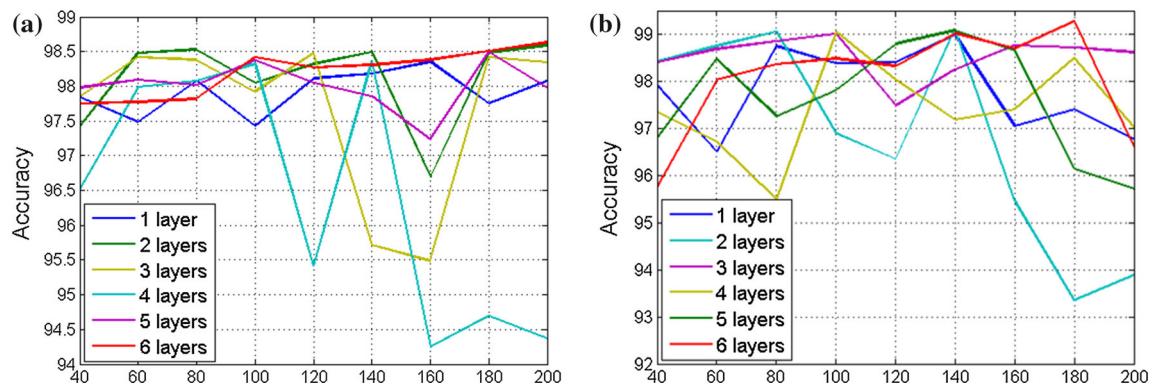
Fig. 6 Influence of the number of hidden layers for the proposed GF-FSAE method. The horizon axis is iteration number. The vertical axis is accuracy. **a** Is the result of proposed GF-FSAE method with differ- ent hidden layers for the Pavia data, and **b** is the result of proposed GF-FSAE method with different hidden layers for the Salinas data

spatial dimensions. On the other hand, the value of two windows radius r_1 and r_2 in guided filtering is 10 and 30 for the Pavia data and 20 and 50 for the Salinas data, both 30 dimensions.

Table 3 shows the classification accuracies under designated training and testing samples of the Pavia data. The Salinas data results are presented in Table 4. Table 2 summarizes the accuracy with different number of training iterations from 1 to 1000 of two images. Through Tables 3 and 4, we can gain a full view that the GF-FSAE algorithm shows more competitive in OA, AA, kappa under a small training set. More importantly, this framework permits great expansion in multi-features learning and provides a great help for the future work.

The influence of hidden layers number is shown in Fig. 6 where the hidden layers are from 1 to 6.

7 Conclusions

In this paper, we combine the SAE deep learning classification framework and guided filtering. Both spatial and spectral features are efficiently explored in constructing feature vectors. The proposed method promotes the hyperspectral data classification accuracy by introducing the filtering of the local pixel information and utilizing multi-feature in deep feature learning. An advantage is the framework controls spatial information skillfully before classification, not focusing on denoising the result. The proposed multi-feature deep learning shows better results when compared with some other methods.

The proposed method also can be applied to classification of the data acquired by Internet of Things (IoT) sensors and from social media. For variety of IoT sensors, we often obtain multi-source data such as image, sound, velocity, temperature, air pressure. For social media data, most of them are images, sound, video and texts. Both IoT sensors and social media data are multi-source or multi-modal data. It is similar to the characteristic of multi-band in hyperspectral data in remote sensing field. That image data are correlated with sound data is just analogous to that the red band is correlated with infrared band. Some of the general characters extracted by PCA from the IoT sensors data or social media data could be used as the guidance data are analog to the guidance image in the proposed method. And then, as an extension of the proposed method, the multi-feature could be better utilized for the classification problem in IoT sensors and social media data.

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Ethical standard This article does not contain any studies with human participants or animals performed by any of the authors.

References

Azar AT, Hassanien AE (2015) Dimensionality reduction of medical big data using neural-fuzzy classifier. Soft Comput 19(4):1115–1127

Baassou B, He M, Mei S (2013) An accurate SVM-based classification approach for hyperspectral image classification. In: 21st International conference on geoinformatics, geoinformatics 2013, Kaifeng, China, June 20–22, 2013, pp 1–7

Bengio Y, Lamblin P, Popovici D, Larochelle H (2007) Greedy layer-wise training of deep networks. Adv Neural Inf Process Syst 19:153

Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM TIST 2(3):27

Chen Y, Lin Z, Zhao X, Wang G, Yanfeng G (2014) Deep learning-based classification of hyperspectral data. IEEE J Sel Top Appl Earth Obs Remote Sens 7(6):2094–2107

Chen Y, Zhao X, Jia X (2015) Spectral–spatial classification of hyperspectral data based on deep belief network. IEEE J Sel Top Appl Earth Obs Remote Sens 8(6):2381–2392

Falco N, Benediktsson JA, Bruzzone L (2014) A study on the effectiveness of different independent component analysis algorithms for hyperspectral image classification. IEEE J Sel Top Appl Earth Obs Remote Sens 7(6):2183–2199

Fauvel M, Benediktsson JA, Chanussot J, Sveinsson JR (2008) Spectral and spatial classification of hyperspectral data using SVMs and morphological profiles. IEEE Trans Geosci Remote Sens 46(11):3804–3814

He K, Sun J, Tang X (2013) Guided image filtering. IEEE Trans Pattern Anal Mach Intell 35(6):1397–1409

Huang X, Zhang L (2013) An SVM ensemble approach combining spectral, structural, and semantic features for the classification of high-resolution remotely sensed imagery. IEEE Trans Geosci Remote Sens 51(1):257–272

Kang X, Li S, Benediktsson JA (2014) Spectral–spatial hyperspectral image classification with edge-preserving filtering. IEEE T Geosci Remote Sens 52(5):2666–2677

Ko L-W, Kuo B-C, Lin C-T (2005) An optimal nonparametric weighted system for hyperspectral data classification. In: Proceedings of the knowledge-based intelligent information and engineering systems, 9th international conference, KES 2005, Melbourne, Australia, September 14–16, 2005, Part I, pp 866–872

Li J, Bioucas-Dias JM, Plaza A (2013) Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning. IEEE Trans Geosci Remote Sens 51(2):844–856

Li J, Bruzzone L, Liu S (2015) Deep feature representation for hyperspectral image classification. In: 2015 IEEE international geoscience and remote sensing symposium (IGARSS 2015), Milan, Italy, July 26–31, 2015, pp 4951–4954

Liu P, Sun S, Li G, Xie J, Zeng Y (2012) Unsupervised change detection on remote sensing images using non-local information and Markov random field models. In: 2012 IEEE international geoscience and remote sensing symposium (IGARSS). IEEE, pp 2245–2248

Lunga D, Prasad S, Crawford MM, Ersoy O (2014) Manifold-learning-based feature extraction for classification of hyperspectral data: a review of advances in manifold learning. IEEE Signal Process Mag 31(1):55–66

Mavrovouniotis M, Yang S (2015) Training neural networks with ant colony optimization algorithms for pattern classification. Soft Comput 19(6):1511–1522

Melgani F, Bruzzone L (2004) Classification of hyperspectral remote sensing images with support vector machines. IEEE Trans Geosci Remote Sens 42(8):1778–1790

Ouma YO, Tetuko J, Tateishi R (2008) Analysis of co-occurrence and discrete wavelet transform textures for differentiation of forest and non-forest vegetation in very-high-resolution optical-sensor imagery. Int J Remote Sens 29(12):3417–3456

Pearson K (1901) On lines and planes of closest fit to systems of points in space. Philos Mag 2(11):559–572

Pesaresi M, Gerhardinger A, Kayitakire F (2008) A robust built-up area presence index by anisotropic rotation-invariant textural measure. IEEE J Sel Top Appl Earth Obs Remote Sens 1(3):180–192

Quesada-Barriuso P, Arguello F, Heras DB (2014) Spectral–spatial classification of hyperspectral images using wavelets and extended morphological profiles. IEEE J Sel Top Appl Earth Obs Remote Sens 7(4):1177–1185

Richards JA, Jia X (2013) Remote sensing digital image analysis: an introduction. Springer, Berlin

Rumelhart DE, Hinton GE, Williams RJ, Hinton GE (1986) Learning representations by back-propagating errors. Nature 323:533–536

Sun L, Wu Z, Liu J, Xiao L, Wei Z (2015) Supervised spectral–spatial hyperspectral image classification with weighted Markov random fields. IEEE Trans Geosci Remote Sens 53(3):1490–1503

Tan CC, Eswaran C (2008) Reconstruction of handwritten digit images using autoencoder neural networks. In: Canadian conference on electrical and computer engineering, 2008 (CCECE 2008), pp 000465-000470

Tarabalka Y, Benediktsson JA, Chanussot J, Tilton JC (2010) A multiple classifier approach for spectral–spatial classification of hyperspectral data. In: Proceedings of the IEEE international geoscience & remote sensing symposium (IGARSS 2010), July 25–30, 2010, Honolulu, Hawaii, USA, pp 1410–1413

Vincent P, Larochelle H, Bengio Y, Manzagol P-A (2008) Extracting and composing robust features with denoising autoencoders. In: Proceedings of the 25th international conference on machine learning (ICML'08), pp 1096–1103

Wen J, Tian Z, Liu X, Lin W (2013) Neighborhood preserving orthogonal pnmf feature extraction for hyperspectral image classification. IEEE J Sel Top Appl Earth Obs Remote Sens 6(2):759–768

Xia J, Du P, He X, Chanussot J (2014) Hyperspectral remote sensing image classification based on rotation forest. IEEE Geosci Remote Sens Lett 11(1):239–243

Yuan H, Tang YY (2016) Spectral–spatial shared linear regression for hyperspectral image classification. IEEE Trans Cybern PP(99):1–12