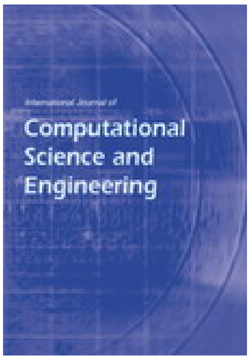


International Journal of Computational Science and Engineering

ISSN online: 1742-7193 - ISSN print: 1742-7185
https://www.inderscience.com/ijcse

Texture-based superpixel segmentation algorithm for classification of hyperspectral images

Subhashree Subudhi, Ramnarayan Patro, Pradyut Kumar Biswal, Kanhu Charan Bhuyan

DOI: 10.1504/IJCSE.2024.10061863

Article History:

Received:	11 July 2022
Last revised:	28 September 2022
Accepted:	13 October 2022
Published online:	25 January 2024

Texture-based superpixel segmentation algorithm for classification of hyperspectral images

Subhashree Subudhi*, Ramnarayan Patro and Pradyut Kumar Biswal

Department of Electronics and Telecommunication,
International Institute of Information Technology,
Bhubaneswar, Odisha, India
Email: ssubudhi84@gmail.com
Email: ram_patro@rediffmail.com
Email: pradyut@iiit-bh.ac.in
*Corresponding author

Kanhu Charan Bhuyan

Department of Electronics and Instrumentation Engineering,
Odisha University of Technology and Research (OUTR),
Bhubaneswar, Odisha, India
Email: kanhu2006@gmail.com

Abstract: To increase classification accuracy, a variety of feature extraction techniques have been presented. A pre-processing method called superpixel segmentation divides an image into meaningful sub-regions, which simplifies the image. This substantially reduces single-pixel misclassification. In this work, a texture-based superpixel segmentation technique is developed for the accurate classification of hyperspectral images (HSI). Initially, the local binary pattern and Gabor filters are employed to extract local and global image texture information. The extracted texture features are then provided as input to the simple linear iterative clustering (SLIC) algorithm for segmentation map generation. The final classification map is constructed by utilising a majority vote strategy between the superpixel segmentation map and the pixel-wise classification map. The proposed method was validated on standard HSI datasets. In terms of classification performance, it outperformed other state-of-the-art algorithms. Furthermore, the algorithm may be incorporated into the UAV's onboard camera to automatically classify HSI.

Keywords: hyperspectral image classification; superpixel segmentation; simple linear iterative clustering; SLIC; spatial-spectral feature extraction.

Reference to this paper should be made as follows: Subudhi, S., Patro, R., Biswal, P.K. and Bhuyan, K.C. (2024) 'Texture-based superpixel segmentation algorithm for classification of hyperspectral images', *Int. J. Computational Science and Engineering*, Vol. 27, No. 1, pp.103–121.

Biographical notes: Subhashree Subudhi is currently pursuing her PhD from the International Institute of Information Technology, Bhubaneswar. Currently, she is working as an Assistant Professor in the Odisha University of Technology and Research (OUTR), Bhubaneswar. Her research interests include hyperspectral image processing, machine learning, deep learning, and remote sensing.

Ramnarayan Patro received his PhD degree from the International Institute of Information Technology, Bhubaneswar in 2021. Currently, he is working in MathWorks as a Senior Technical Content Writer. His research interests include embedded system, hyperspectral image processing, band selection, machine learning and remote sensing.

Pradyut Kumar Biswal received his PhD from the Indian Institute of Technology, Kharagpur in 2011. Currently, he is working as an Assistant Professor in the International Institute of Information Technology, Bhubaneswar, India. His research interest includes VLSI architecture design for signal and image processing algorithms, biomedical signal and image processing, and hyperspectral image processing.

Kanhu Charan Bhuyan received his PhD from NIT Rourkela in 2014. He is currently working as an Associate Professor at the Instrumentation and Electronics Department, Odisha University of Technology and Research (OUTR), Bhubaneswar. His research interests include renewable energy, IOT, machine learning, and remote sensing.

1 Introduction

In applications such as urban planning, ecology, precision farming, defence, mining, and space explorations, accurate classification of remote sensing images is a very crucial task. With recent advancements in the hyperspectral (HS) imaging system, now for each spatial location in the image, it is possible to record the electromagnetic spectrum in various narrow contiguous spectral bands. In addition, small spatial structures in the image can also be analysed by HS sensors of very high spatial resolution. However, there are several challenges involved in the classification of hyperspectral image (HSI). It suffers from the Hughes effect or curse of dimensionality as it has a limited proportion of reference samples over hundreds of spectral bands. Also, there exists high spatial variability of spectral signatures. Many HSI classification techniques have been developed, which only exploits the rich spectral information of HSI, and neglects the spatial details. Some of the popular traditional classification methods are maximum likelihood, neural network, k-nearest neighbours, logistic regression and support vector machine (SVM). But, the classification performance by using only spectral features is not at all encouraging because of the large spectral variability produced by materials properties and environmental factors (Tao et al., 2022). In HSI, significant improvement in the classification performance can be achieved by considering the homogeneous spatial distribution of surface materials (Li et al., 2020). Therefore, in recent years more emphasis is laid on the spatialspectral feature extraction (FE) step and some of such approaches are discussed further (Venkatesan and Prabu, 2022). Mathematical morphology based approaches are extensively applied by researchers for FE. Benediktsson et al. (2005) first introduced the concept of extended morphological profiles (EMP) for FE in HSI. It utilises morphological closing and opening operations to extract spatial features. Later, Dalla Mura et al. (2010) proposed morphological attribute filters (MAP) for the spatial FE. From that point onwards, several variations of attribute profiles (AP) were created. Ghamisi et al. (2015) conducted a comprehensive survey on the evolution in AP. Texture descriptors like wavelet transform (Guo et al., 2014), Gray-level co-occurrence matrix (GLCM) (Huang et al., 2014), local binary patterns (LBP) (Ye et al., 2017) and Gabor filters (Jia et al., 2015) are also used in literature for spatial FE.

The filters that are usually incorporated for noise removal can also be utilised for spatial-spectral FE (Yang et al., 2016). The spatial distribution of neighbourhood pixels is of greater significance in the filtration process as it carries crucial edge information. Hence, several edge-preserving filters like domain transform recursive filters (Kang et al., 2014a), bilateral filters (Tomasi and Manduchi, 1998), guidance filters (Kang et al., 2014b), trilateral filters (Sun and Messinger, 2011), etc. were introduced. For minimisation of noise and texture variations, smoothening operations are performed by these filters. Along with that, they also preserve the most crucial attributes like lines, edges, and other features that help in image interpretation.

Segmentation is another popular approach for inclusion of spatial features (Vantaram et al., 2015). With the traditional fixed window-based methods, the occurrence of salt and pepper noise in the classification result is quite prevalent. Also, it is difficult to adaptively capture the information regarding the changing shape and size of the structural object (Guo et al., 2021). To deal with the above challenges, the concept of superpixel segmentation has emerged as a new option recently (Stutz et al., 2018). It creates smaller meaningful patches that can adhere to the object boundaries by grouping pixels that have homogeneous properties. As superpixels reduce the redundancy and complexity in the image, the performance of the subsequent processing steps can be enhanced significantly. Hence, superpixels can be employed to compute the local image features (Samson and Gabbar, 2022). Due to the numerous advantages of superpixels, they are now widely used for effective FE in HSI as well. In Subudhi et al. (2021), the authors concentrated on the various ways in which established surperpixel techniques might be employed as a pre-processing step for HSI analysis, with a particular focus on classification. There is no dedicated superpixel segmentation algorithms specifically designed for HSI. The segmentation algorithms which are primarily designed for natural RGB images are also utilised for analysing HSI data. But as HS images contain rich amount of information it is not advisable to apply superpixel segmentation directly on raw HSI. The quality of generated superpixel map greatly depends on the base image (Subudhi et al., 2021) on which the algorithms are applied. Hence, in this work the SLIC superpixel segmentation algorithm is applied on the feature space rather than the PCA image in order to accurately extract the contextual features in the image.

In this paper, texture-based superpixel segmentation algorithms are proposed for HSI classification. Extracting useful information from HSI is quite challenging. Hence, Gabor and LBP features are first applied on HSI image, to highlight the key discriminant features. Further, the popular simple linear iterative clustering (SLIC) algorithm is applied on the extracted texture features to acquire more enhanced features which aids in the classification process. The final classification map is regularised with the help of the generated superpixel segmentation map by using the majority voting strategy. The obtained result clearly reveals the superiority of the proposed LBP-SLIC method against other state-of-the-art algorithms.

The major contributions of the proposed approach are highlighted below:

- A novel approach for integration of SLIC superpixel segmentation with the texture descriptors such as LBP and Gabor is presented.
- A modified SLIC algorithm with the following changes are made.

a SLIC is applied on high dimensional texture features to get more highlighted spatial structural information.

b Hexagonal grid is defined for cluster initialisation so that spatial groups with improved homogeneity can be obtained.

c A new distance measure is incorporated as here SLIC is applied on the texture features.

The manuscript is organised as follows: A concise description of various superpixel segmentation algorithms is discussed in Section 2. Next, the proposed work is explained in Section 3. The experimental results are presented in Section 4. Finally, the conclusion is provided in Section 5.

2 Related work

Superpixels which serve as a precursor to image segmentation task can be described as an unsupervised oversegmentation of an image into several semantic sub-regions, bearing similar characteristic features. This concept was initially introduced in the year 2003 by Ren and Malik (2003). Using superpixels for segmentation has several advantages:

1 Features can be computed on more meaningful regions instead of acting on individual pixels.

2 Computational complexity reduces drastically as the input entries for subsequent algorithms reduce significantly.

A superpixel must have certain desirable properties as described by Machairas et al. (2014). They are:

- Homogeneity: the generated superpixels must have uniform pixel values.
- Boundary adherence: superpixel boundaries must match the object boundaries.
- Regularity: superpixels must be placed in a regular pattern in the image.
- Time complexity: the generated superpixels should have lower computational complexity and higher efficiency.
- Connected partition: superpixels consist of a connected set of pixels and the overlapping of superpixels must not exist.

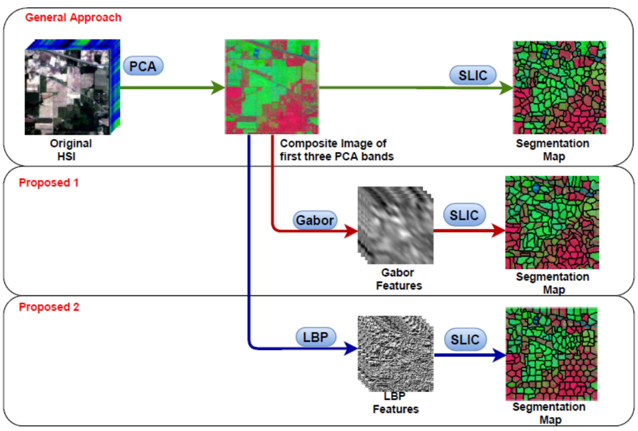
In the area of computer vision, the concept of superpixel segmentation has already gained a lot of popularity due to the aforementioned properties of superpixels. In such applications segmentation is mostly performed over the colour and greyscale images. A detailed survey on the state-of-the-art superpixel segmentation algorithms for colour images is provided by Stutz et al. (2018). The author has broadly categorised the superpixel algorithms into seven categories: density-based, watershed-based, graph-based, path-based, contour evolution-based, energy optimisation-based and clustering-based approaches.

Recently superpixel segmentation algorithms have been incorporated for HSI classification, as it can very well represent the spatial regularity of the surface materials in HSI. These segmentation algorithms partition the HS image into several homogeneous subregions, thereby reducing the computational complexity in the subsequent image processing tasks drastically. In Subudhi et al. (2021), a comprehensive review of superpixel segmentation methods for HSI classification is presented. The article also analysed different superpixel creation algorithms and post-processing frameworks for using superpixels in HSI. Chen and Wang (2014) proposed a spatial-spectral classification framework where superpixel segmentation and pixel-wise classification results are merged using a fuzzy-logic combination rule. Segmentation can be used to improve the classification results of SVM classifier by applying majority voting inside each segment (Jiménez et al., 2015). A superpixel-based Markov random field model for HSI classification was presented by Li et al. (2013). Later, Fang et al. (2015) adopted superpixels to exploit spatial-spectral information via multiple kernels. A multiscale superpixel-level subspace-based SVM for HSI classification was then proposed by Yu et al. (2017). The pixel correlation within each superpixel was also considered by Tu et al. (2018) in order to exploit spatial-spectral features. Liu et al. (2017) developed a multi-morphological superpixel model in for HS image classification. In remote sensing community, entropy rate superpixels (ERS) (Tu et al., 2018) and SLIC (Liu et al., 2017), algorithms are vastly applied for superpixel segmentation. This is mainly because both these algorithms are faster and they can generate compact superpixels which adhere well with the object boundaries (Tang et al., 2015). Watershed segmentation algorithm was applied by Zhang et al. (2015) on a fused gradient image generated from multispectral bands. This helps in overcoming the oversegmentation problem in watershed segmentation.

3 Proposed method

The authors in Subudhi et al. (2021) have mentioned the different ways in which superpixel algorithms can be applied as a pre-processing step for HSI analysis along with the pros and cons of each of the approaches. Hence, by taking the motivation from the article (Subudhi et al., 2021), in this paper we decided to incorporate SLIC on the feature space to create superpixels. Superpixel segmentation when applied on the feature space generates enhanced spatial structure and reduces the incorrect boundary regions (Liu et al., 2017). The framework for superpixel generation is presented in Figure 1. In most of the existing approaches, SLIC is applied on the composite image of first three PCA bands as the reduced image becomes similar to the natural RGB image hence, the SLIC algorithm originally developed for general images can be applied easily. But by reducing the dimension of HSI, significant amount of information is lost. So, to extract enhanced spatial structures and correct boundary information, feature extractors like LBP and Gabor can be applied on the PCA image. In the first approach, Gabor filter is applied on the PCA image to extract the global texture information. The extracted Gabor features is then provided as an input to SLIC for segmentation map generation. In the second approach, to effectively capture the local texture information LBP is applied over the PCA image and the resultant features is fed as an input to SLIC for superpixel map generation. The framework for the validation of the proposed superpixel generation method is presented in Figure 2. A pixel-wise classification map is first produced by directly employing the SVM classifier on the initial HSI image. The derived spatial information from the superpixel segmentation map is next incorporated in spatial-spectral classification by employing the majority voting strategy (Tarabalka et al., 2010). This technique is mostly used here to regularise/optimise the classification map with the guidance of the segmentation map. It is based on the assumption that an unlabeled pixel, after initial classification, that has the same class label of neighbouring pixels, is reliable. Figure 3 shows an illustrative example of the combination of spectral and spatial information using the majority voting classification method. A detailed explanation of the modified SLIC algorithm and the texture descriptors used in this work is presented in the below subsection.

Figure 1 Proposed framework for superpixel generation (see online version for colours)



3.1 SLIC algorithm

SLIC (Achanta et al., 2012) algorithm is the most widely used method for grouping set of similar pixels into a region. It is a popular gradient-ascent-based superpixel segmentation approach, where an initially defined tentative set of cluster points are iteratively refined using a gradient ascent method until some convergence criteria are met. This algorithm has lower computational complexity as it applies the *k*-means method locally. The algorithm includes four key steps: cluster centre initialisation, cluster assignment, cluster centre updation, and post-processing. In this work, a modified SLIC algorithm with the following changes is presented so as to obtain improved segmentation results:

1 hexagonal grids for cluster centre initialisation

2 application of SLIC on texture features

3 incorporation of a new distance measure. Each of these changes are explained in details in below section.

To generate initial clusters, the standard SLIC algorithm uses a square grid. But, in the proposed method hexagonal grids are defined for cluster centre initialisation. Figure 4 contains the architecture of the basic hexagonal grid structure where each corner and each edge are shared by three and two hexagons respectively. There are two main advantages of using hexagonal grids over square grids:

1 more number of off-diagonal neighbours is available for each hexagon hence; the surrounding spatial information can be learned more accurately

2 less distance distortion of boundary pixels is provided by hexagonal grids.

Hence, with the modified SLIC algorithm, spatial groups with improved homogeneity can be obtained. Figure 4 contains the structure of a basic hexagonal grid. To describe the hexagon's size height h and width w is computed. To represent the spacing between the adjacent hexagons vertical distance V and horizontal distance T is used.

The centre of the hexagon can be calculated using the following simple matrix multiplication [equation (1)].

$$\begin{bmatrix} x_i \\ y_i \end{bmatrix} = \begin{bmatrix} T & 0 \\ V & h \end{bmatrix} \begin{bmatrix} r_i \\ c_i \end{bmatrix} \tag{1}$$

where $[x_i,\ y_i]$ represents the i^{th} cluster centre's spatial coordinates. $[r_i, c_i]$ are the row and column indices of the i^{th} superpixel. The segmentation process is initialised in accordance with the width and height of the hexagon. The central pixel coordinates are used as the initial coordinates and the origin is at the upper left corner of the image. The average spectrum is based on the spectrum of the centre pixel. Even though, the superpixel's spatial extent is assumed to be a hexagonal region, the search operation is performed on a region of size $2U \times 2U$ around the superpixel in order to find similar pixels. Hence, with regard to the number of pixels, the computational complexity is still linear.

The original SLIC algorithm was developed for natural images having red, green and blue channels. But in case of HSI, it is not advisable to directly apply SLIC as the red, green and blue channels are not covered by some HSIs. To overcome this problem, often the initial three principal components (PCs) are utilised to create pseudocolour images (Jia et al., 2018). But several issues still exist. The underlying significant spectral information of HSI can not be explored completely by the SLIC as after applying PCA, the image dimension is reduced drastically. The SLIC algorithm is modified in this paper, so that it can utilise the key discriminative features of all the spectral bands along with the spatial information of the surface material to produce the superpixel segmentation map. But superpixel segmentation when applied directly on the raw HSI may result in over-segmentation. To overcome this problem in this paper, superpixel segmentation is performed on the extracted texture features. This in turn results in more highlighted spatial structural information, thereby minimising the influence of faulty region boundaries, decreasing disparity within the same class, and inhibiting over-segmentation. The quality of the generated superpixels greatly depends upon the features over which it is applied. In this paper two popular texture descriptors: LBP and Gabor filters are employed to derive local and global image texture information. Highly distinguished surface materials can be obtained by these texture descriptors. Let the extracted texture features are represented as $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\} \in \mathbb{R}^{B \times N}$ with N pixels and B number of feature band. Each pixel can be represented as $\mathbf{A}_i = [\mathbf{f}_i, \mathbf{x}_i, \mathbf{y}_i]^T$ where $\mathbf{f}_i^T = [\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_B]^T$ is the feature vector and $[\mathbf{x}_i, \mathbf{y}_i]^T$ is the position vector. The K number of initial cluster centres $\mathbf{C}_j = [\mathbf{f}_j, \mathbf{x}_j, \mathbf{y}_j]^T$ are sampled on a regular hexagonal grid and are thus equally spaced apart.

After initialisation of cluster centres, the next step is the cluster assignment step, where each pixel is assigned to the nearby cluster centre based on the computed distance measure D. Distance is computed within a $2U \times 2U$ window around the cluster centre. The distance between the cluster centre C_j and pixel A_i is calculated as follows [equation (2)]:

$$D = D_{feature} + \frac{W}{U} D_{spatial} \tag{2}$$

where W is the weighting factor between spectral and spatial features. Note that the distance is identical to one in the original SLIC algorithm. The only modification is new definition of $D_{feature}$. In this paper, as we are applying SLIC on the texture features, the colour space is getting changed. Hence, there is a need to update the corresponding spectral distance measure to spectral information divergence (SID) (Chang, 2000). SID is one of the most popular measures to compute spectral similarity between two pixels by measuring the discrepancy between them. The dissimilarity measure among the pair of feature vectors can be represented as in equation (3):

$$D_{feature} = \sum_{i,j=1}^{N} f_i \log\left(f_i / f_j\right) + \sum_{i,j=1}^{N} f_j \log\left(f_j / f_i\right) \tag{3}$$

where $D_{feature}$ is the measure of homogeneity within the superpixels. f_i and f_j are the features vectors at pixel i and j respectively.

The spatial distance between feature vectors A_i and A_j are represented as in equation (4).

$$D_{spatial} = \sqrt{\left(x_i - x_j\right)^2 + \left(y_i - y_j\right)^2} \tag{4}$$

where (x, y) denotes the location of pixel i in superpixel. The spatial distance $D_{spatial}$ ensures regularity and compactness in the generated superpixels.

The cluster assignment step is next followed by the cluster centre updation step. Here, the superpixel centre coordinates and the centre spectrum are updated. The centre coordinates takes the average coordinates of all pixels in the superpixel and the centre spectrum takes average spectrum of all pixels in the superpixel. The cluster assignment and updation steps are iteratively repeated until convergence criterion is met.

In the final step, post-processing is performed to enforce connectivity by reassigning disjoint pixels to nearby superpixels.

3.2 Texture descriptors

3.2.1 Gabor

Gabor filter is a widely used texture feature extractor and edge detector (Jia et al., 2015). It is basically generated by multiplying a Gaussian kernel with a sinusoidal wave. Mathematical representation of a two-dimensional Gabor filter is as follows:

$$G_{\lambda,\theta,\varphi,\sigma,\gamma}(m, n) = \exp\left(\frac{-m'^2 + \gamma^2 n'^2}{2\sigma^2}\right) \exp\left(j\left(2\pi \frac{m'}{\lambda} + \varphi\right)\right)$$

where

$$m' = m\cos\theta + n\sin\theta \tag{5}$$

$$n' = -m\sin\theta + n\cos\theta \tag{6}$$

where λ represents the sinusoidal factor wavelength, θ specifies the Gabor kernel's orientation, φ is the phase shift, σ represents the Gaussian kernel's standard deviation and γ denotes the Gabor function's spatial aspect ratio. The Gabor filter's real and the imaginary parts are returned by $\varphi = 0$ and $\varphi = \pi/2$ respectively.

The standard deviation σ is determined with the help of λ and spatial frequency bandwidth BW as:

$$\sigma = \frac{\lambda}{\pi} \sqrt{\frac{\ln(2)}{2}} \frac{2^{BW} + 1}{2^{BW} - 1} \tag{7}$$

The Gabor features are extracted by applying the Gabor filter on the pseudocolour image of first three PCA bands (Figure 1). These extracted texture features are further used for performing the superpixel segmentation.

Figure 2 Validation of the proposed approach (see online version for colours)

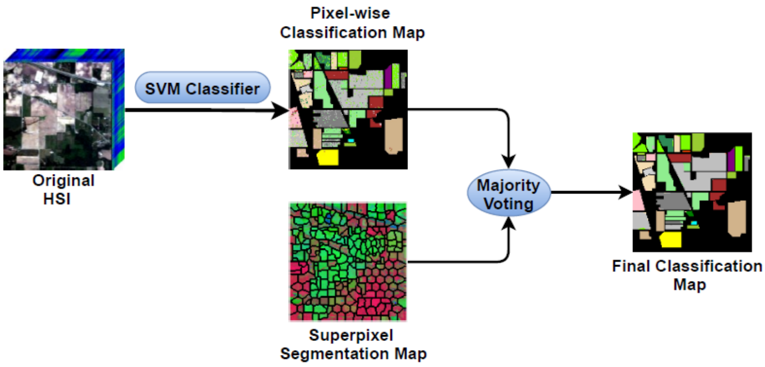


Figure 3 Majority voting strategy (see online version for colours)

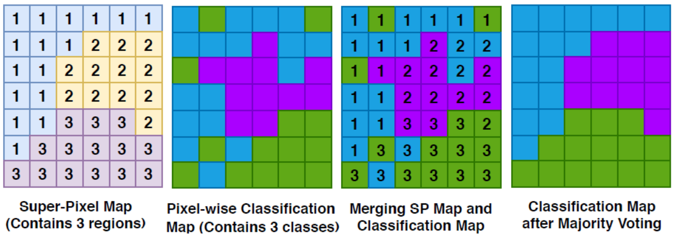


Figure 4 Hexagonal grid structure (see online version for colours)

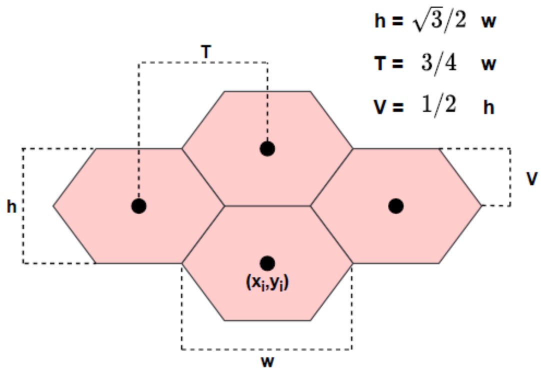
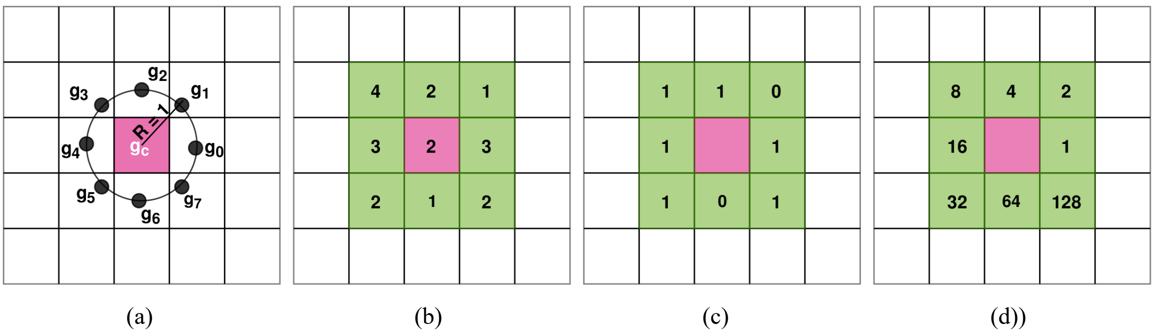


Figure 5 (a) Texture unit with radius $r = 1$, centre pixel g_c and its neighbours g_i, (b) Sample 3 × 3 block, (c) Binary labels for neighbouring points, (d) Weights (see online version for colours)



3.2.2 *Local binary pattern*

Another popular texture descriptor is a LBP which can effectively summarise the local structures in an image. The computation of LBP involves two key steps: encoding each point in a block as a pattern and gathering statistics of LBP occurrences in the form of a histogram. For the first step, LBP operator is applied on a texture unit. A texture unit is a basic element for LBP encoding. Figure 5 illustrates the encoding process. For the centre pixel g_c, its local neighbourhood g_i is a set of evenly spaced sampling points P located in a circle of radius R (Figure 5). If the pixel value g_i of the neighbouring pixels is greater than the centre pixel, it is assigned a value 1, otherwise 0. The *LBP* code can be mathematically represented as:

$$LBP_{P,R}(g_c) = \sum_{i=0}^{P-1} t(g_i - g_x) 2^i \quad (8)$$

$$t(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases} \quad (9)$$

Hence, the generated LBP code for g_c is 189. However, the range of LBP code varies from 0 to 255, which have several disadvantages. The 256-level LBP is ineffective in case of noisy images. Also, the time and space complexity of the FE procedure increases drastically as the feature dimension is $256 \times B$, where B represents the available spectral bands. The stability of the generated LBP code is also not good enough. Hence, in order to resolve the aforementioned problems, a uniform LBP (ULBP) code is utilised for texture FE (Li et al., 2015). In ULBP, if the number of bitwise transitions ($0 \rightarrow 1$ or $1 \rightarrow 0$) is less than or equal to 2, it is considered as a uniform sample and is assigned with a unique index (from 0 to 57). In case of more than two transitions, an index 58 is assigned. The mathematical formulation for ULBP is as follows:

$$LBP_{P,R}(g_c) = \begin{cases} \sum_{i=0}^{P-1} t(g_i - g_c), & \text{if } U(LBP_{P,R}(g_c)) \leq 2 \\ [P(P-1)+3]-1, & \textit{Otherwise} \end{cases}$$

where

$$U(LBP_{P,R}(g_c)) = t(g_{P-1} - g_c) + \sum_{i=1}^{P-1} t(g_i - g_c) - t(g_{i-1} - g_c)$$

Finally, after obtaining the LBP code, the LBP occurrence statistics is gathered in the form of a histogram.

In this work, ULBP is applied on the PCA reduced HSI in order to extract LBP codes. Then, ULBP histogram is computed for each pixel in an image patch around a pixel, in order to construct ULBP features of each principal component in HSI. The extracted ULBP features are displayed in Figure 1. Finally, the extracted LBP texture features serve as a base image upon which superpixel segmentation is applied.

4 Experimental result

4.1 Experimental data

To experimentally validate the performance of the proposed method, four popular datasets namely Indian Pines, Pavia University, Houston 2013, and Houston 2018 were utilised. These datasets have different characteristic behaviour in terms of spectral and spatial resolution. The basic information regarding the datasets is presented in a tabular form in Table 1. In Figures 6, 7, 8, and 9 the false-colour composite image, ground truth image, and class names for Indian Pines, Pavia University, Houston 2013, and Houston 2018 datasets respectively are provided.

4.2 Experimental setup

To validate the effectiveness of the proposed superpixel algorithm, it is compared with other state-of-the-art algorithms like SVM (Melgani and Bruzzone, 2014), EMP (Benediktsson et al., 2005), edge preserving filter (EPF) (Kang et al., 2013), SLIC (Achanta et al., 2012), superpixel-based classification via multiple kernels (SCMK) (Fang et al., 2015), superpixelwise PCA (SuperPCA) (Jiang et al., 2018), adjacent superpixel-based generalised spatial-spectral kernel (ASGSSK) (Sun et al., 2019), convolutional neural network-pixel pair features (CNN-PPF) (Li et al., 2016), and 3DCNN (Chen et al., 2016). For training, 3%, 2%, and 0.2% of samples were randomly selected from each class for the Indian Pines, Pavia University, and Houston 2018 datasets respectively. In the case of the Houston 2013 dataset, 30 samples were chosen randomly for training. All the experiments were independently repeated for ten iterations with different train/test sets to generate results that are statistically more remarkable, and the mean classification accuracy along with standard deviation values are finally reported. With random train/test split, the systematic errors and random discrepancies can be avoided easily and unbiased results can be produced.

The performance of different algorithms are evaluated by using five popular performance metrics, i.e., overall accuracy (OA), average accuracy (AA), Kappa coefficient, class-by-class accuracy, and computation time. All the experiments were performed using MATLAB R2018b software installed on a computer having Intel core i5-6200 CPU 2.30 GHz and 16 GB RAM.

The SVM classifier adopts a one-vs.-one multiclass approach for classification. The LIBSVM package was utilised for the implementation of SVM. The regularisation parameter C and RBF kernel parameter γ are determined by using a five-fold cross-validation approach. To obtain a fair comparison for the EMP, EPF, SLIC, SC-MK, SuperPCA, ASGSSK, CNN-PPF, and 3D-CNN methods the default parameter settings provided in the corresponding publications were utilised.

The value of parameter K has a significant influence on the classification performance. Depending on the structural and texture information available in the dataset, the value of K must be decided. In Figure 10(b), a plot is shown to demonstrate the effect of the number of superpixels K on the classification accuracy for the proposed method when K is varied from 100 to 1,600. In the proposed work, the value of K is taken as 300 for the Indian Pines dataset as it contains a large portion of homogeneous regions. Hence, a lower value of K is preferred. Whereas for the Pavia University, Houston 2013, and Houston 2018 datasets, a larger value of K, i.e., 1,000 is chosen as these datasets contain more detailed structural and texture information. Hence, there is a need to devote a higher number of superpixels to effectively capture the available information.

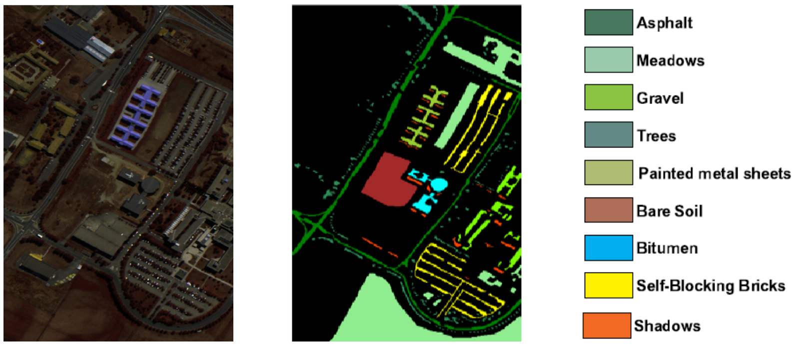
Table 1 Dataset description

Dataset	*Spectral range (nm)*	*Total bands*	*Reduced band*	*Image size (pixels)*	*Classes*	*Spatial resolution (m/pixels)*
Indian Pines	400–2,500	224	200	145 × 145	16	20
Pavia University	430–860	115	103	610 × 340	9	1.3
Houston 2013	380–1,050	144	144	349 × 1,905	15	2.5
Houston 2018	380–1,050	50	50	601 × 2,384	20	1

Figure 6 (a) False-colour composite image, (b) Ground truth image and (c) Class names for Indian Pines dataset (see online version for colours)

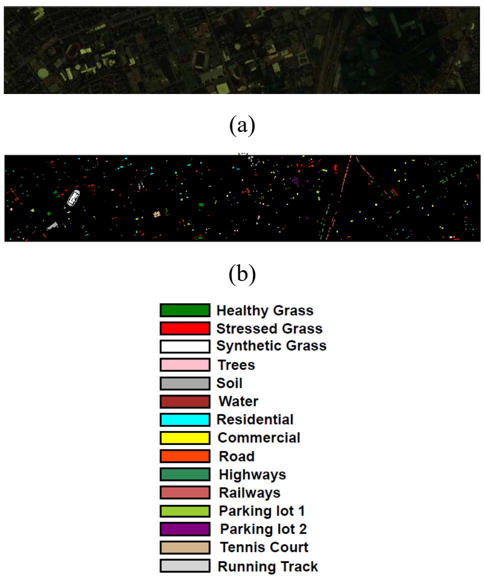
(a) (b) (c)

Figure 7 (a) False-colour composite image, (b) Ground truth image and (c) Class names for Pavia University dataset (see online version for colours)



(a) (b) (c)

Figure 8 (a) False-colour composite image, (b) Ground truth image and (c) Class names for Houston-2013 dataset (see online version for colours)

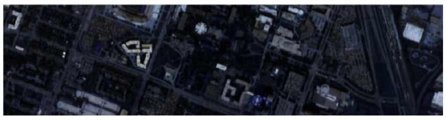


(a)

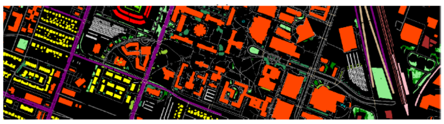
(b)

(c)

Figure 9 (a) False-colour composite image, (b) Ground truth image and (c) Class names for Houston-2018 dataset (see online version for colours)



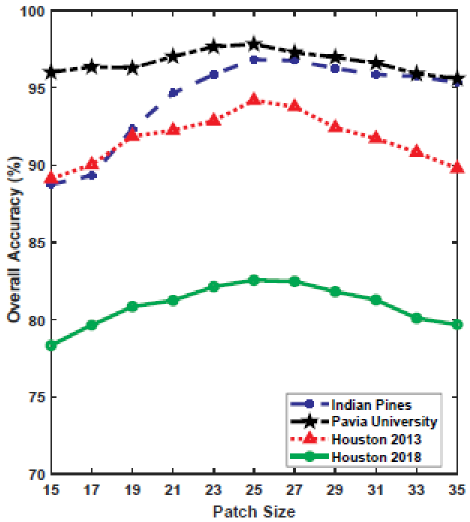
(a)



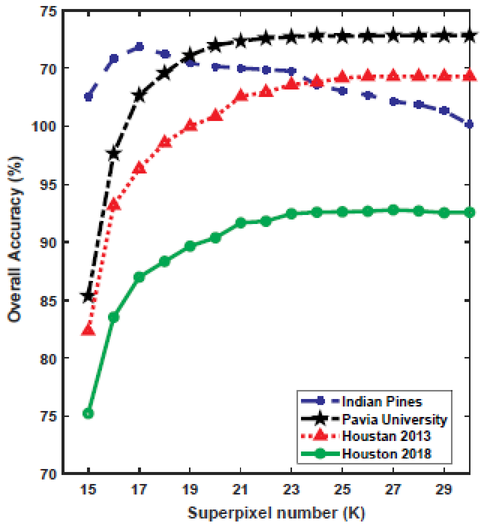
(b)

(c)

Figure 10 (a) Effect of block size on the classification accuracy for proposed LBP-SLIC method, (b) Effect of number of superpixels *K* on the classification accuracy for the proposed LBP-SLIC method (see online version for colours)



(a)



(b)

In LBP-SLIC and Gabor-SLIC, texture features are first extracted using ULBP and Gabor filters respectively. For ULBP, the considered parameters are: radius (R) is taken as 1 with 8 evenly spaced sampling points *P*. The block size of 21 × 21 is used to compute the LBP. Figure 10(a) contains a plot showing the effect of variation in block size on the classification accuracy. It can be observed that variation in block size has not much influence on the classification accuracy for the four experimented datasets. Hence, we may use any block size. For all the test cases, a block size of 25 × 25 is considered for evaluation. Whereas for the Gabor filter the bandwidth BW was taken as 1 for the Indian Pines dataset. But for the Pavia University, Houston 2013, and Houston 2018 datasets BW was considered to be 5. The value of wavelength λ was chosen as 16. As a result, eight orientations, i.e., [0, $\pi/8$, $\pi/4$, $3\pi/8$, $\pi/2$, $5\pi/8$, $3\pi/4$, $7\pi/8$] were considered. The default value for the Gabor function's aspect ratio was selected as 0.5.

Figure 11 Effect of training sample variation on classification performance for (a) Indian Pines, (b) Pavia University, (c) Houston 2013 datasets, (d) Houston 2018 dataset (see online version for colours)

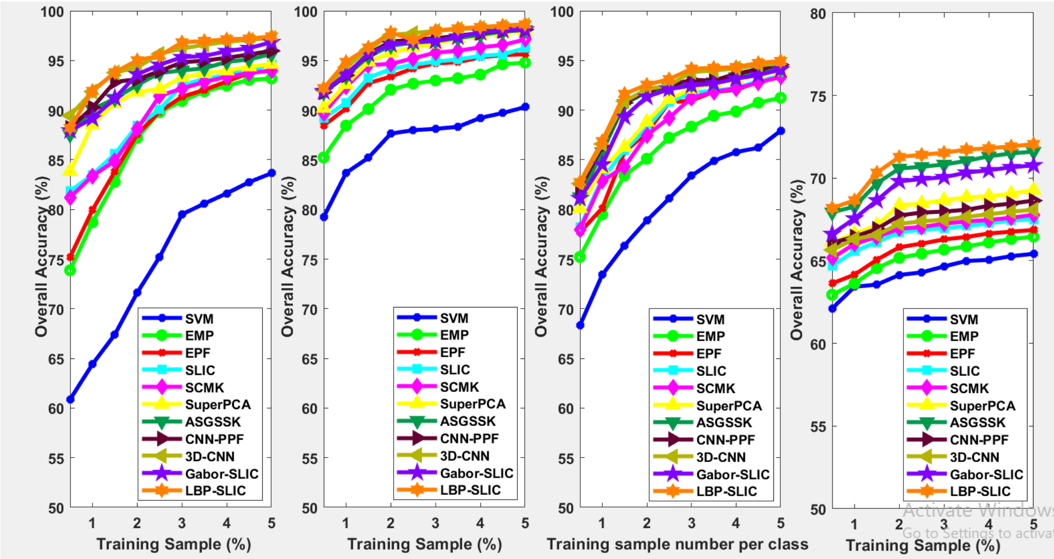


Table 2 Classification result for Indian Pines dataset with 3% training samples from each class for SVM, EMP, EPF, SLIC, SCMK, SuperPCA, ASGSSK, CNN-PPF, 3D-CNN, Gabor-SLIC, and LBP-SLIC algorithms

Class	*Samples*	*SVM (Melgani and Bruzzone, 2004)*	*EMP (Benediktsson et al., 2005)*	*EPF (Kang et al., 2013)*	*SLIC (Achanta et al., 2012)*	*SCMK (Fang et al., 2015)*
1	46	63.89(4.29)	65.12(3.45)	81.4(1.96)	95.35(1.28)	97.67(0.91)
2	1,428	72.97(6.82)	83.31(2.62)	86.96(2.31)	84.34(1.72)	91.09(3.04)
3	830	78.2(7.63)	91.79(2.21)	90.26(3.45)	94.56(2.37)	95.02(1.58)
4	237	46.56(10.33)	86.04(4.12)	98.65(2.15)	90.00(3.05)	97.25(1.69)
5	483	95.85(1.53)	88.99(3.43)	89.87(2.31)	95.10(0.48)	90.11(2.37)
6	730	97.44(2.55)	99.27(0.21)	99.42(0.32)	98.08(0.59)	96.73(1.24)
7	28	75.45(3.42)	84.62(2.13)	99.48(0.16)	85.19(3.48)	100.00(0.00)
8	478	98.69(1.34)	98.89(2.21)	100.00(0.00)	94.16(2.78)	99.54(0.27)
9	20	62.5(11.32)	100.00(0.00)	84.21(4.56)	57.89(7.61)	55.56(6.74)
10	972	78.12(4.61)	87.51(2.34)	88.5(1.86)	93.24(1.81)	93.97(0.97)
11	2,455	68.38(8.31)	93.07(3.44)	91.63(2.94)	94.92(1.37)	93.14(2.12)
12	593	75.95(6.81)	80.65(5.31)	73.84(3.86)	73.37(4.82)	78.90(2.27)
13	205	98.79(0.84)	91.19(2.54)	96.89(1.93)	90.05(2.74)	89.42(3.41)
14	1,265	93.08(1.64)	97.31(1.37)	96.30(2.21)	96.51(1.81)	97.08(0.87)
15	386	71.75(3.57)	90.08(2.31)	98.90(2.57)	97.49(3.24)	97.46(2.84)
16	93	93.33(2.11)	95.45(1.65)	77.01(3.57)	95.40(1.32)	91.86(2.31)
OA:		79.51(1.21)	90.92(0.66)	91.35(1.21)	92.19(0.75)	93.30(1.50)
AA:		75.68(1.34)	89.65(1.12)	90.86(0.98)	89.73(1.52)	91.55(0.87)
Kappa:		74.81(0.64)	89.66(0.81)	90.14(0.58)	91.11(0.88)	92.36(0.21)
Time (sec):		5.65	16.82	11.34	10.51	13.22

Class	*SuperPCA (Jiang et al., 2018)*	*ASGSSK (Sun et al., 2019)*	*CNN-PPF (Li et al., 2016)*	*3D-CNN (Chen et al., 2016)*	*Gabor-SLIC*	*LBP-SLIC*
1	87.8(4.23)	97.44(0.58)	86.05(3.49)	92.86(3.27)	82.93(2.83)	89.19(5.21)
2	90.51(2.76)	92.83(1.18)	89.03(4.32)	91.01(2.43)	91.01(2.61)	93.87(1.75)
3	94.65(1.27)	97.62(0.85)	95.55(1.66)	90.96(4.58)	95.62(1.83)	98.34(0.65)
4	92.86(2.24)	96.57(0.94)	89.95(5.31)	92.2(2.86)	95.69(0.67)	90.48(3.42)

Note: Numbers in parentheses denote the standard deviation of the accuracies obtained in repeated experiments.

Table 2 Classification result for Indian Pines dataset with 3% training samples from each class for SVM, EMP, EPF, SLIC, SCMK, SuperPCA, ASGSSK, CNN-PPF, 3D-CNN, Gabor-SLIC, and LBP-SLIC algorithms (continued)

Class	*SuperPCA (Jiang et al., 2018)*	*ASGSSK (Sun et al., 2019)*	*CNN-PPF (Li et al., 2016)*	*3D-CNN (Chen et al., 2016)*	*Gabor-SLIC*	*LBP-SLIC*
5	94.94(0.88)	97.84(0.76)	97.30(3.54)	95.96(2.31)	96.95(0.45)	96.11(0.77)
6	97.87(1.53)	98.89(0.98)	99.85(0.23)	97.02(1.18)	99.07(0.12)	98.63(0.84)
7	100.00(0.00)	100.00(0.00)	98.68(1.21)	99.94(0.01)	87.50(3.54)	100.00(0.00)
8	99.53(0.31)	99.03(0.11)	99.77(0.18)	95.91(2.67)	99.29(0.38)	100.00(0.00)
9	94.44(2.13)	100.00(0.00)	68.42(8.94)	78.95(6.82)	94.12(4.21)	93.75(3.87)
10	90.63(3.75)	96.77(0.64)	91.72(5.67)	93.85(3.57)	93.92(1.83)	96.66(0.63)
11	94.30(1.62)	96.16(0.88)	93.36(2.31)	89.86(4.82)	95.05(1.62)	96.59(0.86)
12	93.81(3.51)	89.61(2.83)	88.81(4.73)	86.97(3.21)	90.60(3.11)	94.73(0.88)
13	100.00(0.00)	96.61(0.75)	88.30(5.22)	95.24(2.71)	91.71(2.56)	95.12(1.21)
14	98.68(0.65)	98.34(0.34)	98.02(1.96)	96.13(2.27)	99.73(0.05)	99.80(0.11)
15	99.71(0.21)	99.40(0.14)	98.03(1.27)	95.49(2.59)	98.82(0.35)	99.68(0.13)
16	96.39(1.22)	81.01(3.15)	97.67(2.69)	84.88(7.35)	95.06(1.53)	94.59(2.11)
OA:	94.82(1.12)	96.20(0.79)	94.01(0.68)	92.52(1.02)	95.33(0.61)	96.83(0.86)
AA:	92.51(0.68)	96.13(0.51)	92.62(1.35)	92.33(0.86)	94.19(1.05)	96.10(0.51)
Kappa:	94.10(0.39)	95.67(0.72)	93.17(1.02)	91.49(0.75)	94.68(0.82)	96.38(0.61)
Time (sec):	15.37	18.54	265.46	210.34	42.35	54.81

Note: Numbers in parentheses denote the standard deviation of the accuracies obtained in repeated experiments.

Table 3 Classification result for Pavia University dataset with 2% training samples from each class for SVM, EMP, EPF, SLIC, SCMK, SuperPCA, ASGSSK, CNN-PPF, 3D-CNN, Gabor-SLIC, and LBP-SLIC algorithms

Class	*Samples*	*SVM (Melgani and Bruzzone, 2004)*	*EMP (Benediktsson et al., 2005)*	*EPF (Kang et al., 2013)*	*SLIC (Achanta et al., 2012)*	*SCMK (Fang et al., 2015)*
1	6,631	88.16(3.57)	89.94(4.28)	95.69(0.67)	93.18(1.59)	96.45(1.72)
2	18,649	94.75(4.71)	97.29(1.08)	96.57(1.17)	99.27(0.03)	99.08(0.14)
3	2,099	69.4(3.79)	75.7(2.58)	74.39(3.64)	78.47(2.44)	81.09(1.65)
4	3,064	86.37(2.87)	78.49(1.54)	89.59(1.43)	92.82(0.87)	88.99(1.05)
5	1,345	99.72(0.05)	96.19(1.27)	95.12(2.31)	99.92(0.02)	98.12(0.88)
6	5,029	68.76(6.64)	92.78(2.41)	90.91(3.47)	92.8(1.87)	95.51(1.34)
7	1,330	82.33(4.24)	93.72(2.81)	89.23(3.14)	78.55(3.84)	94.38(1.31)
8	3,682	82.28(3.71)	87.54(3.89)	90.82(2.73)	90.34(2.51)	91.72(1.45)
9	947	99.87(0.02)	90.01(1.56)	95.2(1.24)	97.77(1.23)	96.38(0.87)
OA:		87.64(1.07)	92.07(0.67)	93.38(1.08)	94.66(0.72)	95.78(0.59)
AA:		85.74(1.26)	89.07(1.01)	90.84(1.13)	91.46(0.64)	93.52(0.51)
Kappa:		83.47(0.92)	89.46(1.05)	91.23(0.69)	92.89(0.58)	94.39(0.48)
Time (sec):		9.5	33.69	21.57	20.68	25.67

Class	*SuperPCA (Jiang et al., 2018)*	*ASGSSK (Sun et al., 2019)*	*CNN-PPF (Li et al., 2016)*	*3D-CNN (Chen et al., 2016)*	*Gabor-SLIC*	*LBP-SLIC*
1	97.23(1.12)	97.29(0.47)	96.05(1.06)	94.35(2.24)	94.55(1.22)	98.29(0.65)
2	99.6(0.17)	99.77(0.08)	98.59(1.24)	97.62(1.56)	99.81(0.08)	99.73(0.05)
3	87.41(0.98)	88.46(1.24)	86.81(4.35)	85.41(3.17)	92.76(0.24)	87.49(1.51)
4	95.07(1.14)	96.97(0.98)	94.13(2.31)	92.84(3.14)	94.54(0.87)	95.55(1.08)
5	97.5(1.19)	97.32(1.21)	98.5(0.82)	98.58(1.46)	98.25(1.67)	99.89(0.04)
6	96.25(1.27)	97.66(0.89)	93.55(1.42)	91.73(2.94)	96.96(1.61)	98.53(0.25)
7	97.39(0.76)	96.05(0.64)	97.49(2.67)	94.62(2.73)	89.03(2.88)	94.91(0.83)

Note: Numbers in parentheses denote the standard deviation of the accuracies obtained in repeated experiments.

Table 3 Classification result for Pavia University dataset with 2% training samples from each class for SVM, EMP, EPF, SLIC, SCMK, SuperPCA, ASGSSK, CNN-PPF, 3D-CNN, Gabor-SLIC, and LBP-SLIC algorithms (continued)

Class	*SuperPCA (Jiang et al., 2018)*	*ASGSSK (Sun et al., 2019)*	*CNN-PPF (Li et al., 2016)*	*3D-CNN (Chen et al., 2016)*	*Gabor-SLIC*	*LBP-SLIC*
8	91.96(0.97)	93.42(0.72)	91.41(1.34)	83.73(3.68)	90.08(1.66)	94.62(0.64)
9	90.2(2.31)	96.95(0.52)	93.28(1.88)	94.05(2.34)	96.66(0.75)	96.59(0.53)
OA:	96.91(0.34)	97.58(0.27)	95.93(1.21)	94.14(1.35)	96.64(0.63)	97.82(0.62)
AA:	94.73(0.26)	95.99(0.12)	94.42(1.18)	92.55(1.24)	94.74(0.81)	96.19(0.47)
Kappa:	95.91(0.68)	96.79(0.71)	94.61(1.32)	92.24(1.91)	95.54(0.73)	97.1(0.56)
Time (sec):	23.52	36.21	298.21	369.45	98.34	87.62

Note: Numbers in parentheses denote the standard deviation of the accuracies obtained in repeated experiments.

Figure 12 (a) Ground truth image, classification maps of (b) SVM, (c) EMP, (d) EPF, (e) SLIC, (f) SCMK, (g) SuperPCA, (h) ASGSSK, (i) CNN-PPF, (j) 3D-CNN, (k) Gabor-SLIC, (l) LBP-SLIC for Indian Pines dataset (see online version for colours)

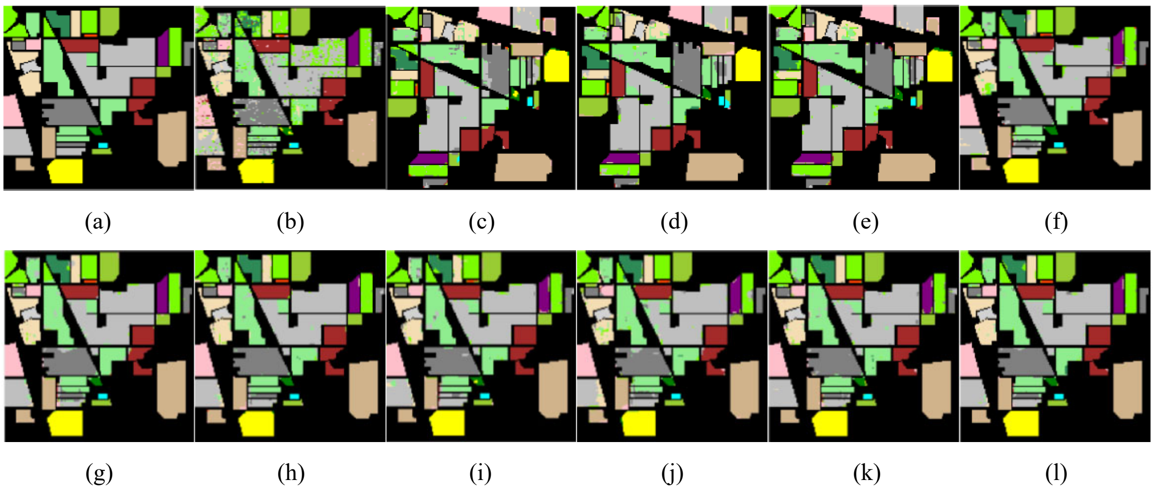


Figure 13 (a) Ground truth image, classification maps of (b) SVM, (c) EMP, (d) EPF, (e) SLIC, (f) SCMK, (g) SuperPCA, (h) ASGSSK, (i) CNN-PPF, (j) 3D-CNN, (k) Gabor-SLIC, (l) LBP-SLIC for Pavia University dataset (see online version for colours)

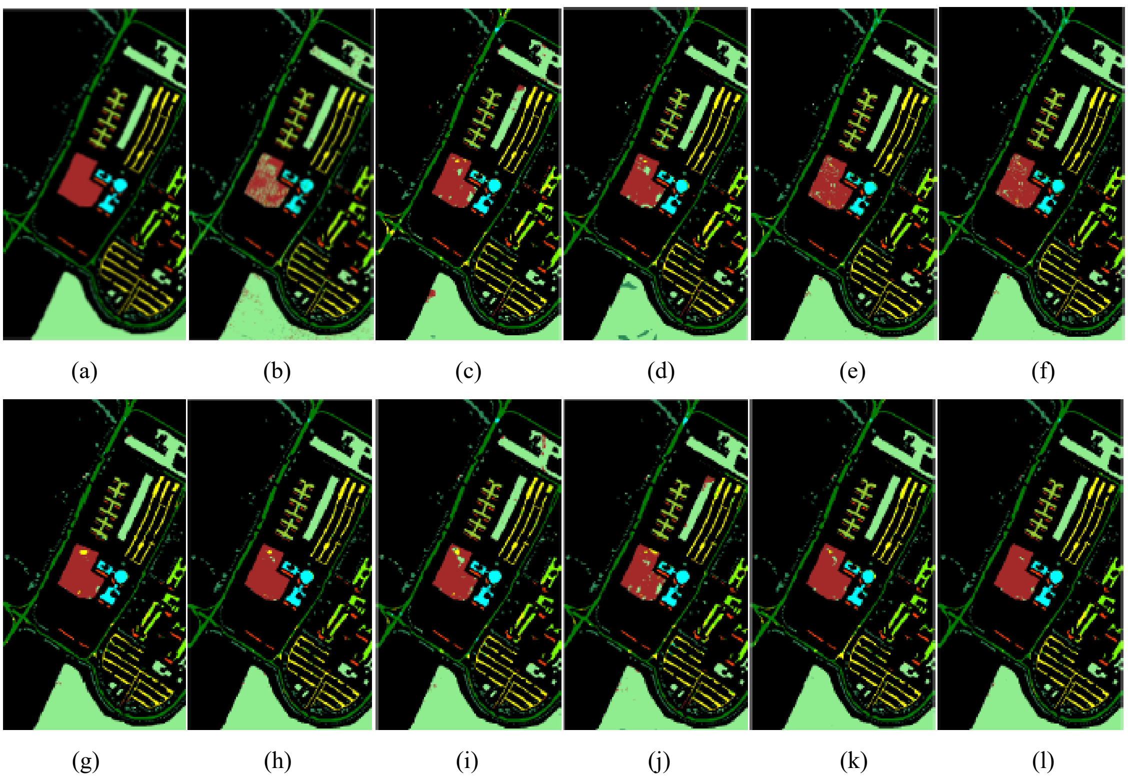
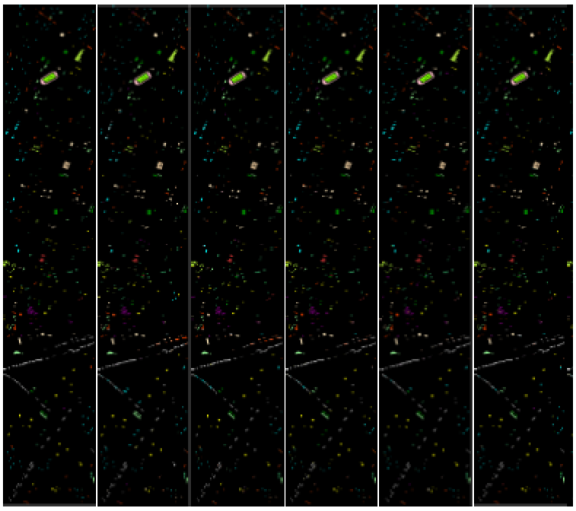
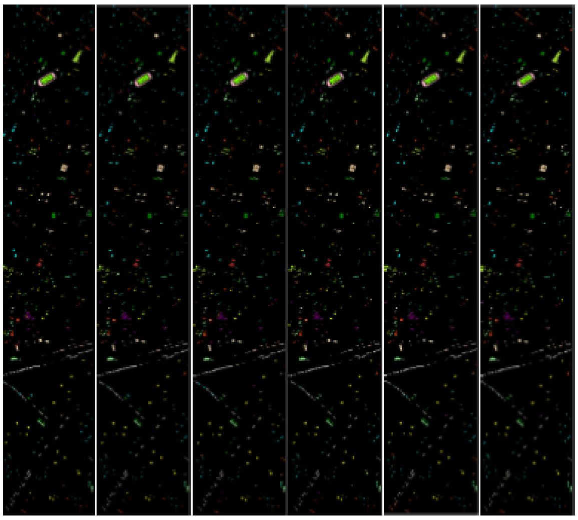


Figure 14 (a) Ground truth image, classification maps of (b) SVM, (c) EMP, (d) EPF, (e) SLIC, (f) SCMK, (g) SuperPCA, (h) ASGSSK, (i) CNN-PPF, (j) 3D-CNN, (k) Gabor-SLIC, (l) LBP-SLIC for Houston 2013 dataset (see online version for colours)



(a) (b) (c) (d) (e) (f)



(g) (h) (i) (j) (k) (l)

Table 4 Classification result for Houston 2013 dataset with 30 training samples from each class for SVM, EMP, EPF, SLIC, SCMK, SuperPCA, ASGSSK, CNN-PPF, 3D-CNN, Gabor-SLIC, and LBP-SLIC algorithms

Class	*Samples*	*SVM (Melgani and Bruzzone, 2004)*	*EMP (Benediktsson et al., 2005)*	*EPF (Kang et al., 2013)*	*SLIC (Achanta et al., 2012)*	*SCMK (Fang et al., 2015)*
1	1,251	83.17(10.21)	97.16(2.65)	94.84(3.62)	97.13(1.84)	91.65(4.51)
2	1,254	94.75(4.63)	96.27(3.21)	97.63(1.86)	94.85(3.52)	97.88(1.36)
3	697	93.84(5.81)	99.41(0.21)	98.35(1.25)	98.95(1.11)	100.00(0.00)
4	1,244	87.06(3.87)	94.44(4.27)	93.41(3.29)	96.13(1.74)	95.47(2.57)
5	1,242	100(0.00)	97.3(2.44)	96.62(2.83)	97.69(3.12)	99.92(0.02)
6	3,25	81.61(5.98)	94.75(1.62)	93.9(2.46)	87.46(4.53)	96.95(2.33)
7	1,268	75.26(8.21)	86.3(5.16)	89.98(2.48)	92.57(3.56)	89.82(4.28)
8	1,244	66.23(2.54)	69.53(3.28)	76.94(2.88)	65.32(1.98)	79.00(3.07)
9	1,252	69.93(3.57)	74.92(1.54)	86.82(3.22)	78.31(4.76)	84.62(2.53)
10	1,227	71.2(6.82)	77.63(4.65)	87.13(2.58)	89.72(1.97)	93.98(2.25)
11	1,235	84.67(2.43)	85.6(2.79)	89.21(0.86)	95.44(1.07)	94.85(2.47)

Note: Numbers in parentheses denote the standard deviation of the accuracies obtained in repeated experiments.

Table 4 Classification result for Houston 2013 dataset with 30 training samples from each class for SVM, EMP, EPF, SLIC, SCMK, SuperPCA, ASGSSK, CNN-PPF, 3D-CNN, Gabor-SLIC, and LBP-SLIC algorithms (continued)

Class	*Samples*	*SVM (Melgani and Bruzzone, 2004)*	*EMP (Benediktsson et al., 2005)*	*EPF (Kang et al., 2013)*	*SLIC (Achanta et al., 2012)*	*SCMK (Fang et al., 2015)*
12	1,233	92.69(1.45)	86.56(3.57)	87.03(4.15)	95.18(1.77)	83.62(3.52)
13	469	86.78(3.49)	90.87(2.46)	87.47(2.59)	89.98(1.76)	93.17(2.15)
14	428	98.06(1.02)	98.77(1.31)	99.25(0.21)	99.75(0.37)	96.73(1.14)
15	660	90.08(1.13)	98.75(1.28)	98.10(0.87)	99.84(0.24)	98.25(0.86)
OA:		83.89(1.09)	88.34(1.54)	90.97(0.87)	91.24(0.72)	92.13(0.81)
AA:		85.02(1.24)	89.88(0.67)	91.78(0.24)	91.89(1.15)	93.06(0.48)
Kappa:		82.59(0.94)	87.4(1.01)	90.23(0.38)	90.53(0.84)	91.49(0.69)
Time (sec):		48.56	57.63	55.73	42.17	51.38

Class	*SuperPCA (Jiang et al., 2018)*	*ASGSSK (Sun et al., 2019)*	*CNN-PPF (Li et al., 2016)*	*3D-CNN (Chen et al., 2016)*	*Gabor-SLIC*	*LBP-SLIC*
1	96.44(1.46)	98.76(1.28)	99.43(0.65)	96.67(2.94)	97.93(2.12)	96.92(1.05)
2	95.21(2.13)	95.78(3.21)	96.73(1.44)	96.35(2.34)	94.13(3.08)	99.17(0.81)
3	98.93(1.05)	100.00(0.00)	98.2(0.85)	98.76(0.29)	99.23(0.37)	99.07(0.21)
4	96.34(3.81)	95.9(3.95)	93.33(4.21)	97.91(3.18)	98.25(2.51)	96.82(3.73)
5	96.75(1.76)	99.92(0.01)	99.42(0.11)	99.83(0.09)	99.75(0.13)	99.58(0.21)
6	96.47(1.84)	98.57(2.11)	97.63(2.63)	89.82(6.21)	98.57(1.53)	96.73(2.34)
7	91.44(3.42)	91.99(2.59)	91.76(1.46)	83.09(4.52)	89.53(3.11)	95.73(1.76)
8	87.69(2.52)	89.07(1.62)	79.08(4.15)	81.32(8.62)	83.9(3.59)	80.57(5.17)
9	87.44(1.86)	79.70(5.43)	87.07(3.11)	83.03(6.57)	88.4(2.87)	82.2(54.93)
10	92.66(1.37)	92.05(0.58)	84.96(1.67)	91.33(1.88)	91.12(0.84)	93.29(1.21)
11	88.35(3.14)	95.38(1.19)	86.56(4.27)	85.23(3.53)	90.76(2.45)	96.46(0.75)
12	91.77(2.81)	92.85(0.84)	88.2(2.28)	89.94(1.86)	81.14(4.27)	94.84(1.01)
13	89.46(5.34)	91.04(2.17)	91.57(2.49)	88.31(4.43)	91.98(2.14)	90.69(3.46)
14	97.67(1.83)	96.08(2.15)	97.74(1.87)	100(0.00)	97.65(1.28)	98.94(0.82)
15	97.41(1.34)	99.51(0.11)	98.57(1.12)	99.02(0.37)	98.86(0.75)	99.18(0.48)
OA:	93.06(1.53)	93.84(1.09)	91.72(0.76)	91.38(0.58)	92.48(1.21)	94.17(0.88)
AA:	93.6(1.15)	94.44(0.46)	92.68(0.41)	92.04(0.78)	93.41(0.98)	94.68(0.57)
Kappa:	92.49(2.07)	93.34(0.81)	91.05(0.67)	90.68(0.71)	91.87(1.53)	93.69(0.34)
Time (sec):	49.78	68.21	414.17	458.32	237.42	314.62

Note: Numbers in parentheses denote the standard deviation of the accuracies obtained in repeated experiments.

Table 5 Classification result for Houston 2018 dataset with 0.2% training samples from each class for SVM, EMP, EPF, SLIC, SCMK, SuperPCA, ASGSSK, CNN-PPF, 3D-CNN, Gabor-SLIC, and LBP-SLIC algorithms

Class	*Samples*	*SVM (Melgani and Bruzzone, 2004)*	*EMP (Benediktsson et al., 2005)*	*EPF (Kang et al., 2013)*	*SLIC (Achanta et al., 2012)*	*SCMK (Fang et al., 2015)*
1	9,799	64.07(11.21)	73.29(3.45)	65.71(6.2)	70.43(12.3)	70.31(4.28)
2	32,502	81.5(0.89)	82.91(2.5)	85.08(1.45)	85.07(3.81)	86.83(1.67)
3	684	100.00(0.19)	100(0.00)	98.83(1.21)	100(0.00)	98.67(1.02)
4	13,588	78.97(4.59)	74.04(6.26)	73.99(4.29)	78.04(13.12)	84.77(8.39)
5	5,048	38.26(13.24)	41.89(8.66)	36.75(5.37)	46.84(7.26)	44.6(16.17)
6	4,516	75.33(6.89)	87.19(3.47)	82.67(7.69)	82.47(5.73)	88.61(4.18)
7	266	15.09(5.23)	61.89(0.65)	67.8(0.89)	66.67(1.11)	68.18(3.14)
8	39,762	72.37(2.98)	74.83(4.59)	77.29(3.62)	76.86(4.27)	78.48(3.08)

Note: Numbers in parentheses denote the standard deviation of the accuracies obtained in repeated experiments.

Table 5 Classification result for Houston 2018 dataset with 0.2% training samples from each class for SVM, EMP, EPF, SLIC, SCMK, SuperPCA, ASGSSK, CNN-PPF, 3D-CNN, Gabor-SLIC, and LBP-SLIC algorithms (continued)

Class	*Samples*	*SVM (Melgani and Bruzzone, 2004)*	*EMP (Benediktsson et al., 2005)*	*EPF (Kang et al., 2013)*	*SLIC (Achanta et al., 2012)*	*SCMK (Fang et al., 2015)*
9	223,684	83.42(0.54)	84.43(1.13)	85.51(0.85)	87.2(2.04)	87.11(0.67)
10	45,810	35.41(2.23)	41.71(4.37)	43.7(3.14)	45.5(4.13)	45.42(3.48)
11	34,002	33.69(5.07)	34.84(3.73)	35.3(2.82)	34.65(3.22)	34.1(2.07)
12	1,516	10.19(0.21)	5.17(1.15)	9.02(6.81)	4.78(3.17)	6.3(3.86)
13	46,358	51.73(5.37)	53.77(4.33)	55.91(4.98)	60(8.12)	60.74(6.52)
14	9,849	55.81(8.64)	57.86(7.82)	60.35(3.65)	67.08(4.93)	64.68(11.32)
15	6,937	72(5.22)	78.06(6.17)	85.83(2.61)	81.15(3.68)	82.05(5.63)
16	11,475	60(3.48)	63.51(2.41)	63.2(2.53)	75.48(4.52)	69.41(8.18)
17	149	55.98(15.21)	63.51(5.28)	83.11(6.48)	37.84(15.61)	85.14(7.65)
18	6,578	48.3(8.85)	40.84(7.05)	47(7.15)	44.97(6.89)	51.64(3.46)
19	5,365	39.2(10.51)	48.32(7.56)	40.54(5.34)	45.36(6.24)	56.48(2.96)
20	6,824	77.34(4.75)	83.27(5.58)	93.25(2.62)	90.57(3.47)	86.22(5.42)
OA:		68.29(0.81)	70.36(0.51)	71.64(0.72)	73.46(0.48)	73.89(0.31)
AA:		54.63(1.68)	62.57(1.23)	64.54(1.76)	64.05(2.15)	67.49(2.34)
Kappa:		59.08(1.21)	61.77(0.52)	63.41(0.91)	65.62(1.05)	66.21(0.61)
Time (sec):		635.621	6,528.6681	3,957.48	1,078.37	4,978.61

Class	*SuperPCA (Jiang et al., 2018)*	*ASGSSK (Sun et al., 2019)*	*CNN-PPF (Li et al., 2016)*	*3D-CNN (Chen et al., 2016)*	*Gabor-SLIC*	*LBP-SLIC*
1	79.44(3.88)	76.81(10.2)	72.89(10.22)	77.34(8.26)	84.66(4.12)	84.09(2.25)
2	85.61(0.49)	89.22(4.21)	86.22(3.03)	86.09(2.71)	88.8(6.87))	90.55(1.22)
3	98.23(1.92)	100(0.00)	99.56(0.51)	99.26(0.86)	83.6(3.58)	98.51(0.78)
4	86.91(12.1)	85.61(8.82)	82.06(3.87)	80.64(5.76)	91.41(2.17)	91.15(4.63)
5	53.97(3.88)	68.75(8.76)	50.00(1.67)	53.53(10.2)	40.98(12.31)	51.14(10.34)
6	85.79(1.82)	94.51(0.87)	88.91(3.56)	92.15(3.97)	78.03(5.41)	79.6(11.38)
7	71.86(2.54)	86.59(1.81)	18.56(3.68)	14.77(7.13)	44.49(2.09)	74.71(1.57)
8	80.21(0.77)	85.24(1.54)	78.86(4.39)	77.99(2.56)	84.14(0.84)	85.77(5.15)
9	88.84(1.34)	91.76(0.17)	88.45(1.43)	87.64(0.62)	95.14(0.81)	95.98(0.16)
10	50.13(2.75)	57.63(2.67)	49.35(1.39)	46.10(5.31)	47.06(2.34)	56.84(3.45)
11	39.03(1.16)	45.31(3.48)	36.59(1.77)	35.81(0.59)	41.19(2.58)	46.79(5.31)
12	13.27(0.51)	12.18(7.79)	9.18(2.25)	8.57(1.35)	5.4(4.62)	2.69(6.13)
13	63.01(4.83)	71.63(1.16)	61.49(2.37)	62.38(3.62)	68.92(3.84)	74.26(5.28)
14	65.7(3.56)	75(5.61)	65.71(3.63)	61.24(10.12)	59.54(8.34)	67.79(6.18)
15	93.51(1.16)	96.59(0.65)	90.32(2.63)	92.06(5.04)	90.55(3.48)	91.87(7.31)
16	77.66(2.94)	82.14(3.74)	73.15(2.79)	66.28(7.35)	69.36(2.66)	71.12(4.14)
17	76.87(2.27)	81.51(4.82)	34.46(12.52)	87.76(4.73)	22.45(18.72)	33.56(14.24)
18	61.35(1.48)	62.05(3.18)	58.1(9.83)	53.83(7.37)	63.35(6.84)	61.91(9.57)
19	63.29(12.91)	78.7(8.67)	57.61(5.36)	51.85(3.54)	42.15(7.19)	61.46(4.82)
20	96.43(0.78)	96.68(0.42)	94.11(1.06)	86.93(3.12)	81.44(5.58)	84.2(6.28)
OA:	76.7(0.62)	81.17(0.96)	75.49(1.06)	74.48(0.94)	79.62(0.38)	82.56(0.56)
AA:	71.56(1.43)	76.9(1.65)	64.78(1.02)	66.11(0.98)	64.13(1.44)	70.2(2.21)
Kappa:	69.93(0.82)	75.61(0.31)	68.26(0.64)	67.01(1.1)	72.94(1.51)	76.92(0.84)
Time (sec):	1,254.28	5,629.37	9,147.32	8,762.41	7,532.1	6,937.71

Note: Numbers in parentheses denote the standard deviation of the accuracies obtained in repeated experiments.

Figure 15 (a) Ground truth image, classification maps of (b) SVM, (c) EMP, (d) EPF, (e) SLIC, (f) SCMK, (g) SuperPCA, (h) ASGSSK, (i) CNN-PPF, (j) 3D-CNN, (k) Gabor-SLIC, (l) LBP-SLIC for Houston 2018 dataset (see online version for colours)

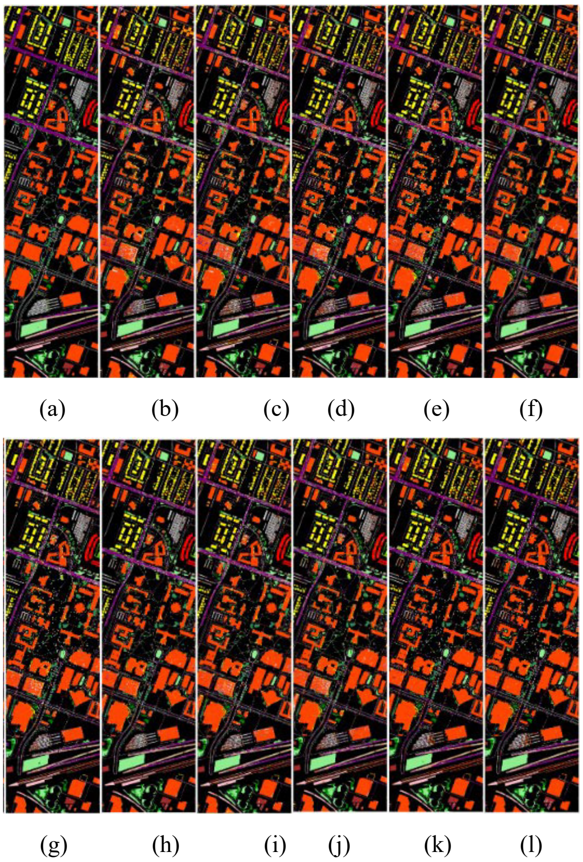


Table 6 Statistics of McNemar test for Indian Pines, Pavia University, Houston 2013, and Houston 2018 dataset

Z	*IP*	*PU*	*Houston 2013*	*Houston 2018*	*IP*	*PU*	*Houston 2013*	*Houston 2018*
	Proposed method (Gabor-SLIC)				*Proposed method (LBP-SLIC)*			
SVM	28.421	54.321	21.548	49.342	37.548	65.642	24.328	72.312
EMP	25.324	49.821	17.682	43.467	32.714	51.432	19.954	65.248
EPF	19.532	37.354	15.932	41.219	29.143	45.423	17.782	61.372
SLIC	12.278	34.439	9.197	38.334	16.753	36.214	13.457	55.225
SC-MK	15.351	22.649	10.216	35.614	18.783	34.294	11.423	51.673
SuperPCA	11.678	21.782	8.672	15.154	15.647	27.649	9.845	29.867
ASGSSK	8.345	11.649	6.792	13.534	13.045	24.314	8.237	18.431
CNN-PPF	5.732	17.354	7.321	28.624	11.468	19.971	7.114	32.614
3D-CNN	7.324	19.723	8.631	30.627	9.379	15.367	4.982	46.342

4.3 Result and discussion

The quantitative and qualitative performance of the proposed method along with the other nine compared approaches is provided in this section, using four popular HSI datasets. Figure 11 shows the influence of training sample variance on classification performance for various approaches. From Figure 11, it can be observed that the increment in the number of training samples has a positive influence on the classification performance.

To objectively assess the performance of the suggested strategy, the classification accuracies and computation time for the proposed technique and similar approaches for the four datasets are reported in Tables 2–5. Figures 12–15 show the classification maps for the proposed approach as well as the other comparable methods.

4.3.1 Indian Pines

The classification result with 3% training samples from each class is provided in Table 2. The proposed LBP-SLIC technique achieves higher classification accuracy for the majority of the classes. In contrast to the SVM approach, the suggested LBP-SLIC and Gabor-SLIC methods increase

OA by around 17.32% and 15.82%, respectively. In general, superpixelbased algorithms give better classification results because superpixels efficiently capture local spatial information. The suggested method outperforms deep learning approaches (3D-CNN and CNN-PPF), which require more training data (Srivastava and Biswas, 2020). When just the spectral characteristics are evaluated, as seen in Figure, 12, the SVM approach produces an extremely noisy map. However, the EMP and EPF algorithms provide a significantly smoother categorisation map when spatial information is included. However, in the detailed and edge areas, all techniques fail to reliably categorise the pixels. Meanwhile, increased performance is found for the superpixel-based algorithms examined (SLIC, SCMK, SuperPCA, and ASGSSK). The proposed LBP-SLIC and Gabor-SLIC algorithms outperform or are equivalent to other current superpixel classification techniques. Specifically, the LBP-SLIC technique outperforms existing approaches. As LBP features are integrated into the SLIC algorithm while constructing superpixels, it is capable of correctly capturing local texture information.

4.3.2 Pavia University

The classification result using 2% training samples from each class is provided in Table 3, and the related classification map is shown in Figure 13. The classification map for SVM is still rather noisy in this scenario. Whereas EMP and EPF approaches provide significantly smoother maps with less salt and pepper noise. The use of superpixel-based algorithms minimises misclassification even further. Table 3 clearly shows that the proposed LBP-SLIC method still outperforms all other tested approaches. In comparison to the baseline SVM methodology, the LBP-SLIC and Gabor-SLIC algorithms enhance OA by around 10.18% and 9%, respectively.

4.3.3 Houston 2013

To evaluate the performance of the investigated approaches, 30 labelled samples from each class were selected as training samples. In Table 4, the classification results are presented. A visual comparison of different methods performance is displayed in Figure 14. From Table 4, it can be clearly observed that the proposed LBP-SLIC method outperforms all the compared methods in terms of OA, AA, and Kappa. Also, the classification map closely resembles the ground truth image for the proposed method.

4.3.4 Houston 2018

The quantitative results for Houston 2018 dataset with 0.2% training samples from each class are presented in Table 5. The corresponding classification map is shown in Figure 15. From Table 5, it can be observed that the proposed methods are robust enough and achieve good classification results even for challenging scenes. Even in the presence of limited training samples, the classifier performance is significantly improved as the spatial contextual features are accurately incorporated using superpixels. Figure 15 also highlights the superiority of the proposed method. The salt and pepper noise is reduced by a greater extent and a smoother classification map is produced with the proposed method.

4.3.4.1 Limitations of proposed method

- The computational cost of the proposed method is higher than most of the compared methods (Tables 2–5). However, the best classification result is provided by the proposed method at the expense of higher computational cost.
- The performance of the proposed method mainly depends on the filter parameters (LBP and Gabor). So these parameters must be chosen carefully. As superpixel segmentation is performed on these extracted features proper tuning of the filter parameters is crucial for efficient spatial-spectral FE.

4.4 Statistical evaluation

The effectiveness of the proposed method was statistically evaluated using the McNemar's test. The classification results for all the test cases are compared using this test. McNemar's test is defined as in equation (10).

$$Z = \frac{f_{12} - f_{21}}{\sqrt{f_{12} + f_{21}}} \tag{10}$$

where f_{12} indicates the number of samples correctly classified by Algorithm 1 and incorrectly classified by Algorithm 2. The performance of Algorithm 1 is better than Algorithm 2 if $Z > 0$. The difference between Algorithm 1 and 2 are statistically significant if $Z > 1.96$. McNemar's test between the proposed LBP-SLIC algorithm and other compared algorithms for Indian Pines, Pavia University, Houston 2013, and Houston 2018 dataset are provided in Table 6. The test result clearly reveals that classification result for the proposed method is statistically significant than other approaches.

5 Conclusions

The primary motivation for this work is to construct a superpixel segmentation framework for the accurate classification of HSI. Superpixels create semantic subregions from the image. So, instead of operating on individual pixels, classification is performed on more meaningful sub-regions. In this work, texture-based superpixel segmentation algorithms are proposed for the classification of HSI. Texture features like LBP and Gabor are utilised for the construction of superpixels using SLIC. Several modifications were made in the SLIC algorithm before utilising the texture features as input. Some of the key changes are: the utilisation of hexagonal grid structure for cluster initialisation, and the introduction of SID distance measure for accurate feature distance measurement.

Later a majority voting strategy was carried out between the obtained superpixel map and pixel-wise classification map to produce the final classification map. The obtained results reveal the supremacy of the proposed LBP-SLIC approach against other state-of-the-art segmentation techniques.

The proposed approach may be implemented by utilising FPGA for real-time classification of HSI. Aside from that, the created algorithm might be implemented inside the UAV's onboard camera to conduct automated HSI classification. The proposed algorithm can swiftly examine vast amounts of data from drones. Using these strategies, local administrators may swiftly get insights from data gathered by drones and address challenges that their citizens are encountering.

References

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P. and Süsstrunk, S. (2012) 'SLIC superpixels compared to state-of-the-art superpixel methods', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 34, No. 11, pp.2274–2282.

Benediktsson, J.A., Palmason, J.A. and Sveinsson, J.R. (2005) 'Classification of hyperspectral data from urban areas based on extended morphological profiles', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 43, No. 3, pp.480–491.

Chang, C-I. (2000) 'An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis', *IEEE Transactions on Information Theory*, Vol. 46, No. 5, pp.1927–1932.

Chen, Y., Jiang, H., Li, C., Jia, X. and Ghamisi, P. (2016) 'Deep feature extraction and classification of hyperspectral images based on convolutional neural networks', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 54, No. 10, pp.6232–6251.

Chen, Z. and Wang, B. (2014) 'An improved spectralspatial classification framework for hyperspectral remote sensing images', in *2014 International Conference on Audio, Language and Image Processing*, IEEE, pp.532–536.

Dalla Mura, M., Benediktsson, J.A., Waske, B. and Bruzzone, L. (2010) 'Morphological attribute profiles for the analysis of very high resolution images', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 48, No. 10, pp.3747–3762.

Fang, L., Li, S., Duan, W., Ren, J. and Benediktsson, J.A. (2015) 'Classification of hyperspectral images by exploiting spectral-spatial information of superpixel via multiple kernels', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 53, No. 12, pp.6663–6674.

Ghamisi, P., Dalla Mura, M. and Benediktsson, J.A. (2015) 'A survey on spectral-spatial classification techniques based on attribute profiles', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 53, No. 5, pp.2335–2353.

Guo, J., Jiang, Z. and Jiang, D. (2021) 'Multi-level spatial attention network for image data segmentation', *International Journal of Embedded Systems*, Vol. 14, No. 3, pp.289–299.

Guo, X., Huang, X. and Zhang, L. (2014) 'Three dimensional wavelet texture feature extraction and classification for multi/hyperspectral imagery', *IEEE Geoscience and Remote Sensing Letters*, Vol. 11, No. 12, pp.2183–2187.

Huang, X., Liu, X. and Zhang, L. (2014) 'A multichannel gray level co-occurrence matrix for multi/hyperspectral image texture representation', *Remote Sensing*, Vol. 6, No. 9, pp.8424–8445.

Jia, S., Shen, L. and Li, Q. (2015) 'Gabor feature-based collaborative representation for hyperspectral imagery classification', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 53, No. 2, pp.1118–1129.

Jia, S., Wu, K., Zhu, J. and Jia, X. (2018) 'Spectral-spatial gabor surface feature fusion approach for hyperspectral imagery classification', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 57, No. 2, pp.1142–1154.

Jiang, J., Ma, J., Chen, C., Wang, Z., Cai, Z. and Wang, L. (2018) 'SuperPCA: a superpixelwise PCA approach for unsupervised feature extraction of hyperspectral imagery', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 56, No. 8, pp.4581–4593.

Jiménez, L.I., Plaza, A., Ayma, V.A., Achanccaray, P., Costa, G.A.O.P. and Feitosa, R.Q. (2015) 'Segmentation as postprocessing for hyperspectral image classification', in *IEEE EUROCON 2015 – International Conference on Computer as a Tool (EUROCON)*, IEEE, pp.1–4.

Kang, X., Li, S. and Benediktsson, J.A. (2013) 'Spectral-spatial hyperspectral image classification with edge-preserving filtering', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 52, No. 5, pp.2666–2677.

Kang, X., Li, S. and Benediktsson, J.A. (2014a) 'Feature extraction of hyperspectral images with image fusion and recursive filtering', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 52, No. 6, pp.3742–3752.

Kang, X., Li, S. and Benediktsson, J.A. (2014b) 'Spectral-spatial hyperspectral image classification with edge-preserving filtering', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 52, No. 5, pp.2666–2677.

Li, M., Deng, S., Wang, L. and Ye, J. (2020) 'Probabilistic rough set-based band selection method for hyperspectral data classification', *International Journal of Computational Science and Engineering*, Vol. 21, No. 1, pp.38–48.

Li, S., Jia, X. and Zhang, B. (2013) 'Superpixel-based markov random field for classification of hyperspectral images', in *2013 IEEE International Geoscience and Remote Sensing Symposium-IGARSS*, IEEE, pp.3491–3494.

Li, W., Chen, C., Su, H. and Du, Q. (2015) 'Local binary patterns and extreme learning machine for hyperspectral imagery classification', *IEEE Trans. Geoscience and Remote Sensing*, Vol. 53, No. 7, pp.3681–3693.

Li, W., Wu, G., Zhang, F. and Du, Q. (2016) 'Hyperspectral image classification using deep pixelpair features', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 55, No. 2, pp.844–853.

Liu, T., Gu, Y., Chanussot, J. and Dalla Mura, M. (2017) 'Multimorphological superpixel model for hyperspectral image classification', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 55, No. 12, pp.6950–6963.

Machairas, V., Decencière, E. and Walter, T. (2014) 'Waterpixels: superpixels based on the watershed transformation', in *2014 IEEE (ICIP)*, IEEE, pp.4343–4347.

Melgani, F. and Bruzzone, L. (2004) 'Classification of hyperspectral remote sensing images with support vector machines', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 42, No. 8, pp.1778–1790.

Ren, X. and Malik, J. (2003) 'Learning a classification model for segmentation', in *IEEE International Conference on Computer Vision*, IEEE Computer Society, Vol. 2, p.10.

Samson, M.I. and Gabbar, H.A. (2022) 'Defect detection through customised reduction and hybrid convolution classification over super-pixel clusters', *International Journal of Reasoning-based Intelligent Systems*, Vol. 14, No. 1, pp.1–7.

Srivastava, V. and Biswas, B. (2020) 'CNN-based salient features in HSI image semantic target prediction', *Connection Science*, Vol. 32, No. 2, pp.113–131.

Stutz, D., Hermans, A. and Leibe, B. (2018) 'Superpixels: an evaluation of the state-of-the-art', *Computer Vision and Image Understanding*, January, Vol. 166, No. 1, pp.1–27.

Subudhi, S., Patro, R.N., Biswal, P.K. and Dell'Acqua, F. (2021) 'A survey on superpixel segmentation as a preprocessing step in hyperspectral image analysis', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 14, pp.5015–5035.

Sun, L., Ma, C., Chen, Y., Shim, H.J., Wu, Z. and Jeon, B. (2019) 'Adjacent superpixel-based multiscale spatial-spectral kernel for hyperspectral classification', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 12, No. 6, pp.1905–1919.

Sun, W. and Messinger, D.W. (2011) 'Trilateral filter on multispectral imagery for classification and segmentation', in *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XVII*, International Society for Optics and Photonics, Vol. 8048, p.80480Y.

Tang, X., Chen, J., Liu, Y. and Sun, Q-S. (2015) 'Hyperspectral image classification by fusing sparse representation and simple linear iterative clustering', *Journal of Applied Remote Sensing*, Vol. 9, No. 1, p.095977.

Tao, H., Xie, X., Tang, R., Hou, Y., Li, J., Feng, W., Chen, Y. and Xu, G. (2022) 'Adaptive group riemannian manifold learning for hyperspectral image classification', *International Journal of Wireless and Mobile Computing*, Vol. 22, No. 3–4, pp.300–309.

Tarabalka, Y., Chanussot, J. and Benediktsson, J.A. (2010) 'Segmentation and classification of hyperspectral images using watershed transformation', *Pattern Recognition*, Vol. 43, No. 7, pp.2367–2379.

Tomasi, C. and Manduchi, R. (1998) 'Bilateral filtering for gray and color images', in *Sixth International Conference on Computer Vision*, IEEE, pp.839–846.

Tu, B., Yang, X., Li, N., Ou, X. and He, W. (2018) 'Hyperspectral image classification via superpixel correlation coefficient representation', *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, Vol. 11, No. 11, pp.4113–4127.

Vantaram, S.R., Piramanayagam, S., Saber, E. and Messinger, D. (2015) 'Spatial segmentation of multi/hyperspectral imagery by fusion of spectral-gradient-textural attributes', *Journal of Applied Remote Sensing*, Vol. 9, No. 1, p.095086.

Venkatesan, R. and Prabu, S. (2022) 'Profound feature extraction and classification of hyperspectral images using deep learning', *International Journal of Intelligent Enterprise*, Vol. 9, No. 3, pp.318–331.

Yang, J., Fan, J., Ai, D., Wang, X., Zheng, Y., Tang, S. and Wang, Y. (2016) 'Local statistics and non-local mean filter for speckle noise reduction in medical ultrasound image', *Neurocomputing*, 26 June, Vol. 195, No. 29, pp.88–95.

Ye, Z., Fowler, J.E. and Bai, L. (2017) 'Spatial-spectral hyperspectral classification using local binary patterns and Markov random fields', *Journal of Applied Remote Sensing*, Vol. 11, No. 3, p.035002.

Yu, H., Gao, L., Liao, W., Zhang, B., Pižurica, A. and Philips, W. (2017) 'Multiscale superpixel-level subspace-based support vector machines for hyperspectral image classification', *IEEE Geoscience and Remote Sensing Letters*, Vol. 14, No. 11, pp.2142–2146.

Zhang, G., Jia, X. and Hu, J. (2015) 'Superpixel-based graphical model for remote sensing image mapping', *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 53, No. 11, pp.5861–5871.