Shearlet-based Structure-Aware Filtering for Hyperspectral and LiDAR Data Classification

Sen Jia[1,2], Zhangwei Zhan[1], and Meng Xu[1,2*]

[1]College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China.

[2]SZU Branch, Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen, China.

[*]Corresponding author. Email: m.xu@szu.edu.cn

Abstract

The joint interpretation of hyperspectral images (HSIs) and light detection and ranging (LiDAR) data has developed rapidly in recent years due to continuously evolving image processing technology. Nowadays most feature extraction methods are carried out by convolving the raw data with fixed-size filters, whereas the structural and texture information of objects in multiple scales can not be sufficiently exploited. In this article, a shearlet-based structure-aware filtering approach, abbreviated as ShearSAF, is proposed for HSI and LiDAR feature extraction and classification. Specifically, superpixel-guided kernel principal component analysis (KPCA) is firstly adopted on raw HSIs to reduce the dimensions. Then, the KPCA-reduced HSI and LiDAR data are converted to the shearlet domain for texture and area feature extraction. In contrast, superpixel segmentation algorithm utilizes the raw HSI data to obtain the initial oversegmentation map. Subsequently, by utilizing a well-designed minimum merging cost that fully considers spectral (HSI and LiDAR data), texture and area features, a region merging procedure is gradually conducted to produce a final merging map. Further, a scale map that locally indicate the filter size is achieved by calculating the edge distance. Finally, the KPCA-reduced HSI and LiDAR data are convolved with the locally adaptive filters for feature extraction, and a random forest (RF) classifier is thus adopted for classification. The effectiveness of our ShearSAF approach is verified on three real-world datasets, and the results show that the performance of ShearSAF can achieve an accuracy higher than that of comparison methods when exploiting small-size training sample problems. The codes of this work will be available at http://jiasen.tech/papers/ for the sake of reproducibility.

1 Introduction

Recently, the continuously evolving remote sensing sensor technologies have contributed to the capture of multisource data in the same area [1, 2]. Among those numerous remote sensing data,

hyperspectral images (HSIs) contain joint spectrum and space information, providing a distinctive discriminating ability for Earth's surface objects. HSIs have hundreds or thousands of narrow spectral bands, covering the spectral region from the visible to the infrared field [3]. In particular, HSIs have both spatial and spectral smoothness, which not only produces detailed and accurate descriptions of objects but also results in a high correlation between adjacent bands [4, 5, 6]. Based on the above reasons, some obstacles and challenges exist regarding the interpretation of HSI information. Specifically, HSIs are prone to include information redundancy as a result of high correlation and the Hughes phenomenon caused by a high spectral dimension [7, 8]. In addition, environmental factors, such as clouds, noise, etc., will also cause information confusion when the remote sensor captures scene data [9]. Compared with HSI, LiDAR integrates a laser ranging system, a global positioning system, and an inertial navigation system, so that it can collect the position and intensity information of objects in a three-dimensional space [10, 11, 12]. However, LiDAR works in a single band and lacks semantic information; thus, it has poor discriminative ability in distinguishing targets with similar heights but different spectra [13].

Many works in the literature have proven the effectiveness of combined HSI and LiDAR interpretation, indicating that the intensity information provided by LiDAR can supplement the HSI deficiencies regarding the target height and shape information [14, 15, 16, 17, 18]. In 2013, the HSI and LiDAR data fusion competition, organized by IEEE Geoscience and Remote Sensing Society, greatly promoted the research on HSI and LiDAR data fusion methods for classification [19]. In general, these fusion methods can be roughly divided into three categories: pixel-level fusion, feature-level fusion and decision-level fusion. The strategy of pixel-level fusion relies on concatenating multisource data directly on the original data, which requires geometric registration. Feature-level fusion is considered as a better approach in that it can achieve better classification performance in most cases [20, 21, 22, 23]. It conducts the feature extraction for each source individually and then combines them. Decision-level fusion methods aim to integrate several rough classification results of multisource data into the final classification [24, 25]. Although the computational complexity is relatively low, it relies heavily on the original classification results and integration strategies. In fact, due to the inherent shortcomings of single data fusion methods, there are many articles exploring the classification framework that combines feature-level fusion and decision-level fusion [26, 27, 28].

In addition, it must be mentioned that deep learning-based approaches is also widely favored in recent years for hyperspectral feature extraction and classification. The pioneering work is the stacked auto-encoders proposed by Chen et al., which has been used for hyperspectral high-level features extraction [29]. Subsequently, convolutional neural network (CNN) [30, 31, 32] and recurrent neural network (RNN) [33, 34] has been widely developed. Among them, the representative 3D-CNN can effectively extract the joint spatial-spectral information and has shown good performance [35]. Furthermore, the deep learning-based framework, that combine traditional features and network structure, has also been explored [36, 37]. However, the model parameter optimization of these deep learning-based approaches generally relies on a large number of training samples, which greatly limits its applicable ability due to the difficulty of sample labelling in remote sensing field. Inspired by the small sample set circumstance, some new strategies have been developed. For example, Yu et al. proposed a novel CNN model by combining data augmentation and 1×1 convolutional kernel [38].

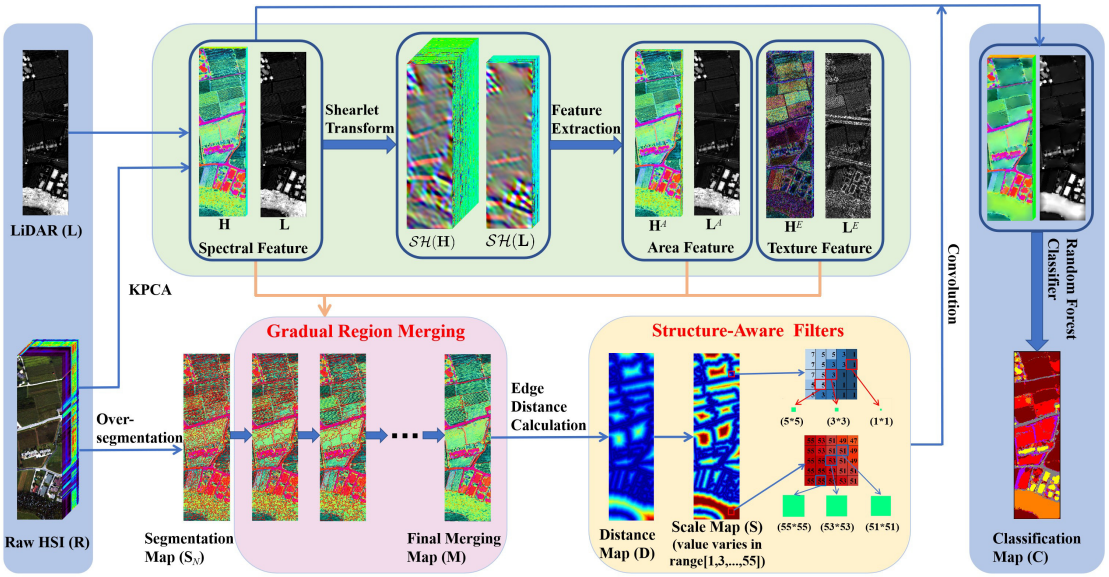


Figure 1: Pipeline of the shearlet-based structure-aware filtering framework for hyperspectral and LiDAR data classification.

More recently, semi-supervised CNN and PCANet-derivative methods have also received constant attention because of their performance with limited training samples [39, 40, 41, 42].

Alternatively, wavelet analysis is an important mathematical tool because of its optimal approximate fitness in signal processing. However, in the case of multidimensional images with a discontinuity curve, traditional wavelet loses its sparsity on the edge response [43, 44]. Thus, the multidimensional directional wavelet is required. Gabor, which is widely used in texture analysis and feature extraction, can be considered as an early directional wavelet [45, 46]. Its inherent drawback is that directions are restricted on each scale once sampled. There has been an emerging series of directional wavelets in the past few decades, such as contourlets, bandlets, curvelets, and shearlets [47, 48, 49, 50]. They all provide a flexible framework in mathematical theory while capturing the geometric features in applications. Among these methods, the shearlet possesses remarkable properties: it accurately captures the edge direction, has an optimal sparse representation for multidimensional data, uses a well-organized multiscale structure, and exhibits fast algorithm implementation and efficient calculation [51, 52, 53]. In its simplest form, the shearlet starts with the construction of the so-called mother function, and then it adopts three basic operations (scaling, translation and direction) to provide a derivative with more shape and direction. Two well-known properties of shearlets are highlighted as follows: 1) If a point is far away from the edge, then its shearlet coefficient decays rapidly as the scale decreases; 2) If a point is an edge point or a corner point, then its shearlet coefficient decays slowly in the normal direction, while it decays rapidly in other directions. Therefore, shearlets have been widely used for edge and corner detection[54, 55, 56]. In addition, due to its frequency domain division, there are also some pioneering works using it for denoising, feature extraction and data fusion [57, 58, 59, 60, 61].

During the past two decades in the computer vision field, another emerging and rapidly spreading

concept is the use of superpixel segmentation. Specifically, a superpixel is considered as a homogeneous area containing some texture or structural similar pixels [62, 63, 64]. The edge of the superpixel is a closed curve with continuity, which is different from scenarios encountered with edge extraction algorithms in which continuous scattered points may exist. Moreover, the superpixel should also possess region compactness, shape regularity and boundary smoothness [65]. Currently, superpixel algorithms are roughly divided into three categories: cluster-based approaches represented by simple linear iterative clustering (SLIC) and simple noniterative clustering (SNIC) [66, 67], graph-based approaches represented by entropy rate superpixel segmentation (ERS) and normalized nuts (NCut) [68, 69], and gradient-based methods represented by spatial-constrained watershed (SCoW) and superpixels using the shortest gradient distance (SSGD) [70, 71]. Notably, fuzzy superpixels were proposed in the past two years and have further enriched the content of superpixels, especially in cases of low spatial resolution such as in remote images [72, 73]. However, regardless of the kind of superpixel algorithm, direct or indirect spatial constraints exist for compactness requirements. For example, SLIC directly adds spatial distance to the clustering metric, and ERS requires each superpixel to be as close to the same size as possible. This spatial constraint inevitably leads to conflicts between oversegmentation and undersegmentation. Moreover, the situation worsens since the resistance of this constraint grows as a power series. In other words, due to larger space constraints, a large homogeneous region has to be divided into several small superpixels, and a small area may contain several objects because there is almost no spatial restriction. Nevertheless, some heuristic solutions have been proposed for superpixel number selection [74, 75], but this inherent property of superpixels has not been slackened effectively.

In particular, filters play an extremely important role in image processing. There are already many filters based on different applications, such as the mean filter, Gaussian filter, Gabor filter, etc., for feature extraction [76, 77] and the Laplacian, Sobel, Laplacian of Gaussian (LoG), etc., for edge capture. However, fixed size filter does not has the ability to obtain the best description of surface objects with various scales, and thus it is undeniably difficult for a uniform size filter to achieve globally satisfactory results. In other words, near the edge, a larger size of a filter will cause more confusing information, whereas in the center of the region, a smaller size of the filter hardly works well when abnormal points exist. Some researchers have made some attempts in this area, such as the multiscale spectral-spatial classification method with adaptive filtering [78, 79, 80] and the spatial adaptive multiscale filtering technique [81, 82]. However, the features or classification results in multiple filtering scales were simply concatenated or combined, and it is more desirable to take full advantage of internal structure of objects and achieve local structure-aware filtering (i.e., automatically adjusting the size of the filter kernel according to the local position).

In this article, we innovatively propose a shearlet-based structure-aware filtering framework, abbreviated as ShearSAF, for HSI and LiDAR data classification with the help of the above tools. First, superpixel-guided kernel principal component analysis (KPCA) is adopted on raw HSIs for dimension reduction and information focus, which is greatly helpful for subsequent calculation and processing. Then, shearlet transform is implemented on KPCA-reduced HSI and LiDAR, and structural description in frequency domain is achieved, i.e., the high frequency and low frequency respectively contains the region information and texture information. They are further processed by energy superposition

and time-frequency conversion to attain region features and texture features. Second, a gradual region merging procedure is developed to alleviate the superpixel spatial constraints and enhance the robustness of the proposed ShearSAF method. Specifically, the SNIC superpixel algorithm acts on the raw HSI to address serious oversegmentation and ensure the homogeneity of each small region, and then the superpixels are progressively combined together according to a well-designed merging criterion that takes all the spectral information, region information and texture information into account, eventually achieving the final merging map. Third, by locating the edge in the final merging map and calculating the shortest distance between each pixel and the edge, we can obtain a distance map to reflect the relationship between points and edges. Through geometry optimization and threshold processing, a scale map is also extracted to control the filter size, in which the point value indicates the size of the convolution kernel. Thus, the adaptive-size filter for each spatial pixel is obtained. Finally, KPCA-reduced HSI and LiDAR are convolved by these structure-aware mean filters for feature extraction and classification to verify the effectiveness of the proposed ShearSAF approach. For a better description, the detailed process of our ShearSAF is shown in Figure 1, and the main contributions of this article are summarized as follows:

- First of all, we design a structure-aware filtering scheme for HSI and LiDAR feature extraction. These locally adaptive-size filters have a small-size kernel near the edge to protect the information from being disturbed by nearby objects, while the kernel size is larger at the center of the area to filter noise and abnormal points. Since structure-aware filter size could reflect the spatial structure of objects more precisely, the convolutional procedure can be more elegant and the discriminability of extracted features can be promoted. To our best knowledge, this is the first time a method of extracting structure-aware features for HSI and LiDAR data processing.

- Second, shearlet transform is employed for structure description on both HSI and LiDAR data. After dividing the shearlet features into low-frequency and high-frequency energy parts, they are converted into area feature and texture feature, respectively. Since the local region and edge structure of objects can be well characterized by the extracted area and texture features, both features are taken into account (conventional methods only use either feature for edge detection and noise reduction), which provide a valuable guidance for the subsequent superpixel fusion.

- Third, we proposed an elaborate design to effectively combine HSI and LiDAR information in a distance measurement for gradual region merging. The well-designed evaluation criterion integrates the spectral (from Euclidean distance viewpoint), area and texture (from statistical distance viewpoint) features, which provides a more comprehensive description of the real scene and could greatly increase the structural representation ability of objects.

- Finally, all the parameters can be either preset and kept unchanged for different HSI and LIDAR data sets or heuristically determined, therefore the robustness and generalization ability of the proposed ShearSAF method can be ensured. Meanwhile, since the structure-aware feature extraction procedure is unrelated to the training samples and only needed to be calculated

once, our ShearSAF also has high efficiency. The source codes of this work will be available at http://jiasen.tech/papers/ for the sake of reproducibility.

We would like to point out that the structure-aware filtering design presented here is essentially very general and can be easily utilized for other features (such as morphological and attribute features). Experimental results with several state-of-the-art methods on three real data sets demonstrate the effectiveness of the proposed ShearSAF approach.

The organization of this paper is as follows. Section 2 introduces related works about shearlet transform, SNIC superpixel algorithm and KPCA. Section 3 describes the process of designing the shearlet-based structure-aware filtering in detail. Section 4 presents the experimental data and two ablation experiments. The experimental results of the proposed ShearSAF method with a number of alternatives are given in Section 5. Finally, Section 6 provides the conclusions of this paper.

2 Related Works

This section introduces the theory of shearlet transform, SNIC superpixel algorithm and KPCA.

2.1 Shearlet Transform

Shearlets have received great attention for their optimal approximation properties in representing images and were first introduced in [83, 44]. The shearlet is regarded as a multiscale representation system and possesses the ability to capture the direction and geometric features [54, 51]. Suppose there exist a dilation matrix $A_a = \begin{pmatrix} a & 0 \\ 0 & \sqrt{a} \end{pmatrix}$ and a shear matrix $S_s = \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix}$ (where $a \in \mathbb{R}^+$ and $s \in \mathbb{R}$ are called the dilation factor and shear factor, respectively), the shearlet mother function is expressed as:

$$\psi_{a,s,t}(x) = a^{-3/4}\psi(A_a^{-1}S_s^{-1}(x-t)) \tag{1}$$

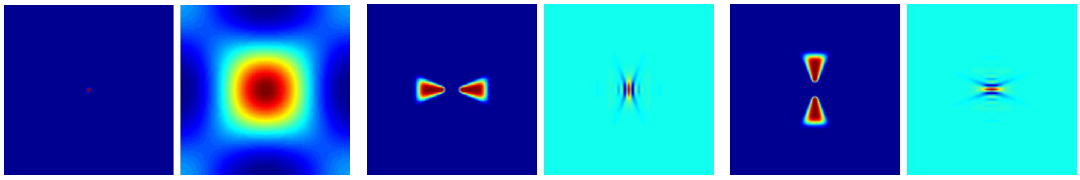
where $t \in \mathbb{R}^2$ is a translation factor, and $x \in \mathbb{R}^2$ is the coordinate in the spatial domain. a and s respectively controls the scale and orientation of the shearlet. In the frequency domain, function ψ is written as $\hat{\psi}$ and can be factorized as:

$$\hat{\psi}(\omega_1, \omega_2) = \hat{\psi}_1(\omega_1)\hat{\psi}_2(\frac{\omega_2}{\omega_1}) \tag{2}$$

where ω_1 and ω_2 are the two coordinate in the frequency domain. Consequently, the $\psi_{a,s,t}(x)$ in the frequency domain can be expressed as follows:

$$\hat{\psi}_{a,s,t}(\omega) = a^{\frac{3}{4}}e^{-2\pi i<\omega,t>}\hat{\psi}_1(a\omega_1)\hat{\psi}_2(a^{-\frac{1}{2}}(\frac{\omega_2}{\omega_1}+s)) \tag{3}$$

where $\omega \in \mathbb{R}^2$ is the coordinate in the frequency domain, which is just the concatenation of ω_1 and ω_2. $\hat{\psi}_1$ and $\hat{\psi}_2$ are the continuous wavelet function and bump function, respectively, meeting a certain support domain.



(a) (b) (c)

Figure 2: Shearlet in different frequency domain supports (left) and the corresponding time domain supports (right). (a) low frequency, (b) horizontal cone and (c) vertical cone.

In fact, the shearlet compact framework in the frequency domain can be divided into three parts: the low-frequency region, horizontal cone and vertical cone. Notably, the above factorization (2) and equation (3) are used for the horizontal cone (In the following section, we renamed $\hat{\psi}_{a,s,t}(\omega)$ in equation (3) as $\hat{\psi}^h_{a,s,t}(\omega)$). Alternatively, in the vertical cone, they are denoted as:

$$\hat{\psi}(\omega_1, \omega_2) = \hat{\psi}_1(\omega_2)\hat{\psi}_2(\frac{\omega_1}{\omega_2}) \tag{4}$$

$$\hat{\psi}^v_{a,s,t}(\omega) = a^{\frac{3}{4}} e^{-2\pi i<\omega,t>}\hat{\psi}_1(a\omega_2)\hat{\psi}_2(a^{-\frac{1}{2}}(\frac{\omega_1}{\omega_2} + s)) \tag{5}$$

For the low-frequency domain, the shearlet has neither dilation nor shear, so it can be written as:

$$\hat{\phi}_{a,s,t}(\omega) = \hat{\phi}_t(\omega) = e^{-2\pi i<\omega,t>}\hat{\phi}(\omega) \tag{6}$$

where $\hat{\phi}$ is called the scaling function. The frequency support of the low-frequency region, representative horizontal cone and representative vertical cone are shown in Figure 2.

In practice, the above continuous shearlet needs to be discretized in digital image processing. Let us consider a single-band digital image $\mathbf{I} \in \mathbb{R}^{X\times Y}$ mapped into a two-dimensional grid$\{(\frac{x}{X}, \frac{y}{Y})|1 \leq x \leq X, 1 \leq y \leq Y\}$; therefore, the related parameters, such as a, s, and t, can be computed as:

$$\begin{aligned} a_j &:= 2^{-2j}, \qquad j = 0, 1, ..., j_0 - 1 \\ s_{j,k} &:= k2^{-j}, \qquad -2^j \leq k \leq 2^j \\ t_{x,y} &:= (\frac{x}{X}, \frac{y}{Y}), \quad 1 \leq x \leq X, 1 \leq y \leq Y \end{aligned} \tag{7}$$

where j_0 is the number of scales. By substituting these discretized parameters (7) into equations (3), (5) and (6), three shearlets in the frequency domain can be rewritten as (the fixed factor $a^{\frac{3}{4}}$ is ignored):

$$\begin{aligned} \hat{\psi}^h_{j,k,x,y}(\omega) &= e^{-2\pi i<\omega,\binom{x/X}{y/Y}>}\hat{\psi}_1(2^{-2j}\omega_1)\hat{\psi}_2(2^j\frac{\omega_2}{\omega_1} + k) \\ \hat{\psi}^v_{j,k,x,y}(\omega) &= e^{-2\pi i<\omega,\binom{x/X}{y/Y}>}\hat{\psi}_1(2^{-2j}\omega_2)\hat{\psi}_2(2^j\frac{\omega_1}{\omega_2} + k) \\ \hat{\phi}_{x,y}(\omega) &= e^{-2\pi i<\omega,\binom{x/X}{y/Y}>}\hat{\phi}(\omega) \end{aligned} \tag{8}$$

where $\omega_1 = -\lfloor\frac{M}{2}\rfloor, ..., \lceil\frac{M}{2}\rceil - 1$ and $\omega_2 = -\lfloor\frac{N}{2}\rfloor, ..., \lceil\frac{N}{2}\rceil - 1$. To avoid overly complicated math-

ematical formulas and theoretical derivations, we directly give the final expression of the shearlet transform:

$$\begin{aligned}\mathcal{SH}(\mathbf{I}) =&\text{ifft2}(\hat{\phi}(\omega_1,\omega_2)\hat{\mathbf{I}}(\omega_1,\omega_2)) \\ &+\text{ifft2}(\hat{\psi}(2^{-2j}\omega_1, 2^{-2j}k\omega_1+2^{-j}\omega_2)\hat{\mathbf{I}}(\omega_1,\omega_2)) \\ &+\text{ifft2}(\hat{\psi}(2^{-2j}\omega_2, 2^{-2j}k\omega_2+2^{-j}\omega_1)\hat{\mathbf{I}}(\omega_1,\omega_2))\end{aligned} \tag{9}$$

where ifft2 represents two-dimensional inverse Fourier transform, and $\hat{\mathbf{I}}(\omega_1,\omega_2)$ is the response of $\mathbf{I}$ in the frequency domain. In this equation, the support domains of the first, second, and third parts are the low-frequency region, horizontal cone, and vertical cone, respectively. Additional details of shearlet construction and derivation can be found in [84].

2.2 Multi-channel SNIC

SNIC represent the simple non-iterative clustering superpixel algorithm, which has both low computational complexity and good segmentation results [67]. Through parameter control in SNIC, the number of superpixel and the weight of spectral-spatial information can be set manually. In this arcticle, the multi-channel SNIC is adopted, which is more suitable for multi-channel hyperspectral image segmentation. Specifically, SNIC starts from the initialization of the centroid and adds the elements into a priority queue. Next, when an element is taken from the priority queue, the surrounding pixels are marked and added to the queue. At the same time, the coordinates of the centroid are updated accordingly. This process will continue until the queue is empty. The comparison criterion of the priority queue is the distance between the elements i and centroid j, which is defined as follows:

$$d_{i,j} = \sqrt{\gamma_1 \left\|\mathbf{r}_i - \mathbf{r}_j\right\|^2 + \gamma_2 \left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2} \tag{10}$$

where $\mathbf{r}$ and $\mathbf{x}$ represent the spectral vector and space coordinates, respectively, and γ_1 and γ_2 are their corresponding weights.

2.3 Superpixel-Guided KPCA

As a generalization of principal components analysis (PCA), KPCA maps the input data into a high-dimensional or Hilbert space by a mapping function and can well reflect the complex structures in the corresponding high-dimensional space [85, 86]. However, in the sample selection strategy of original KPCA, random sampling and conducting all samples (there may be a million levels) strategies usually cause feature degradation and computational explosion. Therefore, a superpixel-based KPCA scheme by taking advantages of superpixel homogeneity is applied for dimension reduction [87].

Specifically, in terms of a raw HSI $\mathbf{R} \in \mathbb{R}^{X\times Y\times B}$ (X, Y and B are respectively the spatial and spectral dimensions), SNIC is applied (here the number of superpixels is simply as $X+Y$) and a superpixel map is obtained. Then, the mean vector of each region is calculated and forms the input sample set $\mathbf{r} = (\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{X+Y})$. Subsequently, the mapping function Φ converts the input low-dimensional sample data into a high-dimensional feature $\Phi(\mathbf{r}) = (\Phi(\mathbf{r}_1), \Phi(\mathbf{r}_2), ..., \Phi(\mathbf{r}_{X+Y}))$.

Let us consider the covariance matrix:

$$\overline{\mathbf{C}} = \frac{1}{X+Y}\sum_{i=1}^{X+Y}\Phi(\mathbf{r}_i)\Phi(\mathbf{r}_i)^T = \frac{1}{X+Y}\Phi(\mathbf{r})\Phi(\mathbf{r})^T \tag{11}$$

Therefore characteristic equation can be denoted as:

$$\overline{\mathbf{C}}\boldsymbol{\beta} = \boldsymbol{\lambda}\boldsymbol{\beta} \tag{12}$$

where $\boldsymbol{\lambda} = diag(\lambda_1, \lambda_2, ..., \lambda_{X+Y})$ is a diagonal matrix composed of eigenvalues arranged from large to small, and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_{X+Y})$ is an $(X+Y)\times(X+Y)$ matrix composed of corresponding eigenvectors. For convenience of calculation, an ingenious substitution is made for $\boldsymbol{\beta}$:

$$\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_{X+Y}) = (\Phi(\mathbf{r})\boldsymbol{\alpha}_1, \Phi(\mathbf{r})\boldsymbol{\alpha}_2, ..., \Phi(\mathbf{r})\boldsymbol{\alpha}_{X+Y}) = \Phi(\mathbf{r})\boldsymbol{\alpha} \tag{13}$$

where $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_{X+Y})$ is the$(X+Y)\times(X+Y)$ coefficients matrix, which is used for explaining the relationship between $\boldsymbol{\beta}$ and $\Phi(\mathbf{r})$. Next, through simultaneously multiplying a matrix $\Phi(\mathbf{r})^T$ by the left-hand side of the equation (12), we can obtain:

$$\Phi(\mathbf{r})^T\overline{\mathbf{C}}\boldsymbol{\beta} = \frac{1}{n}\Phi(\mathbf{r})^T\Phi(\mathbf{r})\Phi(\mathbf{r})^T\Phi(\mathbf{r})\boldsymbol{\alpha} = \frac{1}{X+Y}\mathbf{K}^2\boldsymbol{\alpha} \tag{14}$$

$$\Phi(\mathbf{r})^T\boldsymbol{\lambda}\boldsymbol{\beta} = \boldsymbol{\lambda}\Phi(\mathbf{r})^T\Phi(\mathbf{r})\boldsymbol{\alpha} = \boldsymbol{\lambda}\mathbf{K}\boldsymbol{\alpha} \tag{15}$$

where $\mathbf{K} = \Phi(\mathbf{r})^T\Phi(\mathbf{r})$ is known as the kernel function. For equations (14) and (15), $\boldsymbol{\lambda}\mathbf{K}\boldsymbol{\alpha} = \frac{1}{X+Y}\mathbf{K}^2\boldsymbol{\alpha}$ can be optimized into $(X+Y)\boldsymbol{\lambda}\boldsymbol{\alpha} = \mathbf{K}\boldsymbol{\alpha}$, which is regarded as a new characteristic equation. Finally, for each spectral vector $\mathbf{R}_i$ of $\mathbf{R}$, the dimensionality reduction process can be expressed as:

$$\Phi(\mathbf{R}_i)^T\bar{\boldsymbol{\beta}} = \Phi(\mathbf{R}_i)^T(\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, ..., \boldsymbol{\beta}_K) = \Phi(\mathbf{R}_i)^T\Phi(\mathbf{r})(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_K) \tag{16}$$

where K represents the reserved dimension.

3 Shearlet-based Structure-Aware Filtering

We now consider the proposed shearlet-based structure-aware filtering design. Briefly, in order to take full advantage of the structural information of objects, the following idea is complied: when a certain point approaches the edge, the size of the filter will shrink, while when it is at the center, the size of the filter will enlarge. Our ShearSAF approach to obtain this adaptive-size filter involves the following four steps: preprocessing, shearlet-based feature extraction, gradual region merging, and structure-aware filter designing. Table 1 summarizes some important mathematical symbols used in this paper for additional clarification.

Table 1: Definitions of the mathematical symbols used in the paper.

Symbol	Meaning
$\mathbf{R}$, $\mathbf{H}$, $\mathbf{L}$	raw HSIs, KPCA-reduced HSIs and LiDAR
X, Y	spatial dimension of HSIs and LiDAR
B, K	spectral and KPCA-reserved dimension of HSIs
$\mathbf{H}^E$, $\mathbf{H}^A$	texture and area features of HSIs
$\mathbf{L}^E$, $\mathbf{L}^A$	texture and area features of LiDAR
$G^S_{m,n}$, $G^E_{m,n}$, $G^A_{m,n}$	the spectral, texture and area distance
$D_{m,n}$, $C_{m,n}$	regional dissimilarity and merging cost
$\mathbf{S}_N$	initial over-segmentation map
$\mathbf{M}$	final merging map
$\mathbf{D}$, $\mathbf{S}$	distance map and scale map
$\mathbf{F}$	structure-aware filters
$\mathbf{Z}$, $\mathbf{C}$	classification feature and classification map

3.1 Preprocessing

This part mainly includes two aspects: superpixel-guided KPCA for HSI dimension reduction and multi-channel SNIC for superpixel oversegmentation.

3.1.1 Superpixel-Guided KPCA for HSI Dimension Reduction

For high-dimensional HSI data with complex structures, KPCA has superior capabilities for dimensionality reduction. In our superpixel-guided KPCA, the radial basis function (RBF) kernel ($\mathbf{K}_{i,j} = \Phi(\mathbf{r}_i)^T\Phi(\mathbf{r}_j) = \exp(-\frac{1}{2}||\mathbf{r}_i - \mathbf{r}_j||^2)$) is adopted and 99.5% energy is maintained in the principal components. Afterward, the information-focused hyperspectral data $\mathbf{H} \in \mathbb{R}^{X\times Y\times K}$ is attained.

3.1.2 Multi-channel SNIC for Superpixel Oversegmentation

SNIC is an emerging superpixel algorithm containing both low computational complexity and good segmentation results. The multi-channel SNIC is applied on raw HSI data $\mathbf{R}$ and an initial oversegmentation map $\mathbf{S}_N$ can be obtained, in which the homogeneity of each superpixel can be largely ensured. Instead of directly provide the number of superpixels, the number of pixels inside each superpixel is set as N_p (the value of this parameter will be discussed in the experimental section), and weight parameters γ_1 and γ_2 are set as $1/B$ and 0.5 as default, respectively.

3.2 Shearlet-based Feature Extraction

The shearlet is a tight framework with clear mathematical meaning that provides directional scale decomposition. In the high-frequency part, it can effectively obtain texture information, and is thus used for edge detection and corner detection [56]. Alternatively, in the low-frequency part, it can effectively obtain area information, and is thus used for denoising [88].

Let us start with the single-band LiDAR data $\mathbf{L} \in \mathbb{R}^{X\times Y}$. The number of scale j_0 in equation (7) is set as 3 by default in our shearlet transform, while the construction of $\hat{\psi}_1$ (i.e., the meyer wavelet function) and $\hat{\psi}_2$ (i.e., the bump function) are the same as [84].

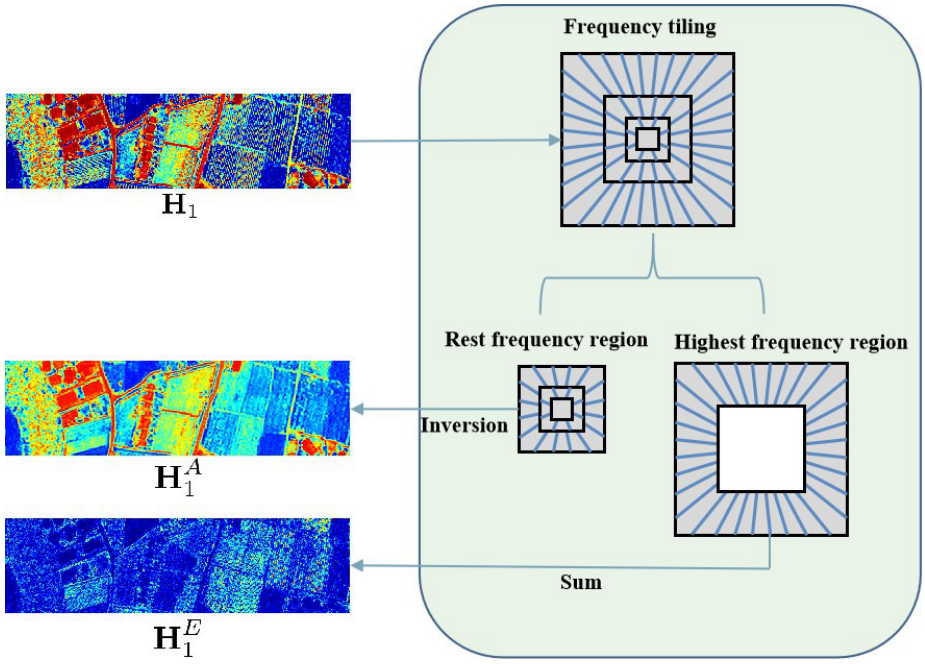


Figure 3: The process of obtaining texture and region description by shearlet, where the input data is the first component of KPCA-reduced HSI data $\mathbf{H}_1$.

In the shearlet compact frame, when the scale is 0, 1, and 2 respectively, there are 4, 8 and 16 support cones with different directions (including the horizontal cone and vertical cone). Among them, the 16 highest frequency part is related to the texture information, while the remaining parts can well characterize the area information; therefore, the shearlet-based frequency features are divided into two parts as follows:

$$\mathcal{SH}(\mathbf{L}) = \mathcal{SH}^H(\mathbf{L}) + \mathcal{SH}^R(\mathbf{L}) \tag{17}$$

where $\mathcal{SH}^H(\mathbf{L})$ and $\mathcal{SH}^R(\mathbf{L})$ respectively represents the highest frequency and the rest frequency information of the LiDAR data $\mathbf{L}$.

Furthermore, for the highest frequency parts $\mathcal{SH}^H(\mathbf{L})$, i.e., $j = 3$, the sum of coefficients in 16 directions is used as the measure of texture feature $\mathbf{L}^E$. For the other remaining 13 frequency parts $\mathcal{SH}^R(\mathbf{L})$, including the low-frequency region and remaining high-frequency cones with $j = 1$ and $j = 2$, the inversion of the shearlet transform is applied to acquire the area information $\mathbf{L}^A$. They can be computed as follows:

$$\mathbf{L}^A = inv(\mathcal{SH}^R(\mathbf{L})); \quad \mathbf{L}^E = |\sum^{16} \mathcal{SH}^H(\mathbf{L})| \tag{18}$$

where inv is the inversion operator, and $|\cdot|$ is the absolute value operator.

Correspondingly, for the KPCA-reduced HSI data $\mathbf{H} = (\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_K)$, each component performs the above frequency separation process, and then the results are concatenated. Therefore, the

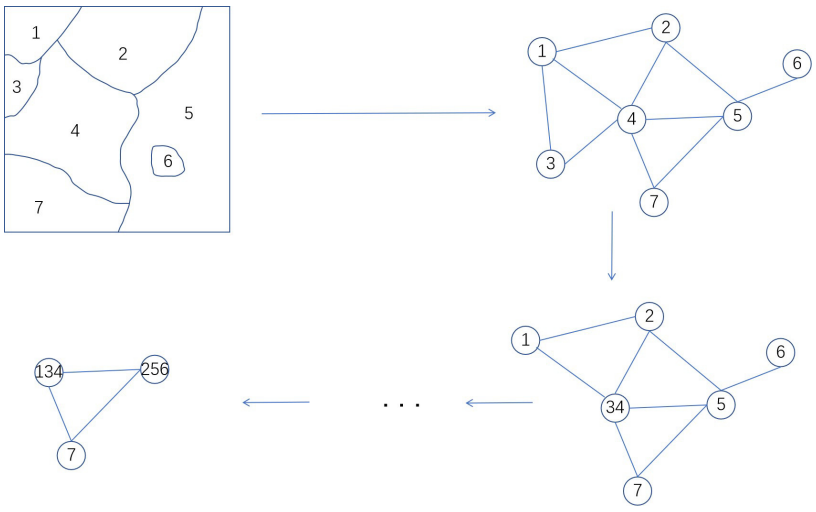


Figure 4: The gradual region merging procedure.

texture information $\mathbf{H}^E$ and area information $\mathbf{H}^A$ for HSI data can be express as:

$$\mathbf{H}^E = (\mathbf{H}_1^E, \mathbf{H}_2^E, ..., \mathbf{H}_K^E); \quad \mathbf{H}^A = (\mathbf{H}_1^A, \mathbf{H}_2^A, ..., \mathbf{H}_K^A) \tag{19}$$

where $\mathbf{H}_i^E$ and $\mathbf{H}_i^A$ contains the texture and area information of the ith component $\mathbf{H}_i$, respectively. The detailed procedure of shearlet-based feature extraction is displayed in Figure 3.

3.3 Gradual Region Merging

The previous steps provide an over-segmentation map ($\mathbf{S}_N$) and three different types of description of objects, including spectral information ($\mathbf{H}$ and $\mathbf{L}$), area information ($\mathbf{H}^A$ and $\mathbf{L}^A$) and texture information ($\mathbf{H}^E$ and $\mathbf{L}^E$). Apparently, it is advantageous to investigate the three features in an unified framework to guide the fusion process of the oversegmentation map $\mathbf{S}_N$.

Specifically, the over-segmentation map $\mathbf{S}_N$ is mapped onto an undirected graph. Each superpixel is regarded as a node and there exists edge only when two superpixels are adjacent. In order to make the description of the proposed progressive region merging process more clear and intuitive, it is divided into two parts: merging cost definition and region merging procedure. Figure 4 illustrates the gradual region merging procedure.

3.3.1 Merging Cost Definition

It can be easily found that the merging cost between two adjacent regions is not only related to the size of the region and the length of shared edges, but also related to the similarity among the three different kinds of features, including spectral, area and texture features. Suppose there are two adjacent regions m and n in the over-segmentation map $\mathbf{S}_N$, the distance between the two regions

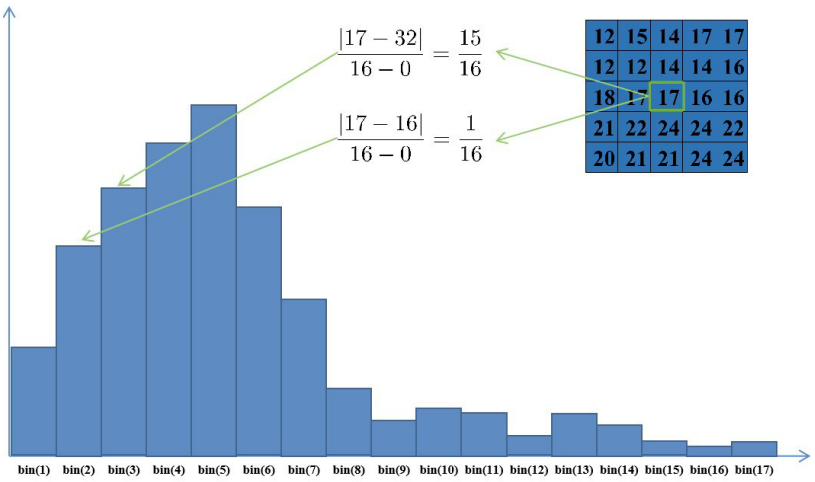


Figure 5: The process of bin-based statistics.

in the spectral domain is calculated by the mean gaps and can be defined as follows:

$$G^S_{m,n} = \sqrt{\frac{1}{K}\sum_{i=1}^{K}|\overline{\mathbf{H}}_i(m) - \overline{\mathbf{H}}_i(n)|^2} + |\overline{\mathbf{L}}(m) - \overline{\mathbf{L}}(n)| \tag{20}$$

where $\overline{\mathbf{H}}_i(m)$ and $\overline{\mathbf{L}}(m)$ represents the mean value of region m in the ith band of KPCA-reduced HSI data $\mathbf{H}$ and LiDAR data $\mathbf{L}$ respectively.

On the other hand, for the area information $(\mathbf{H}^A, \mathbf{L}^A)$ and texture information $(\mathbf{H}^E, \mathbf{L}^E)$, it is necessary to adopt statistical manner to measure the region distance since all the area and the texture features are extracted in the frequency domain. Taking the LiDAR texture information $\mathbf{L}^E$ as an example, it is firstly normalized into the interval [0, 256] for convenience. Then, we select r interval endpoints ($x_1 < x_2 < ... < x_r$, including 0 and 256) to divide the whole interval into $r-1$ parts with the same length. At the same time, these r endpoints are considered as r bins in the histogram and its value in region m is denoted as follows:

$$\begin{aligned} \underset{i=1,2,...,r}{bin(i)} &= \sum_{x_p \in \mathbf{L}^T(m)} (1 - \frac{|x_p - x_i|}{x_2 - x_1}) F(\frac{|x_p - x_i|}{x_2 - x_1}); \\ F(x) &= \begin{cases} 1 & if \quad 0 \le x \le 1; \\ 0 & other; \end{cases} \end{aligned} \tag{21}$$

where $\mathbf{L}^E(m)$ represents the LiDAR texture information value in region m. Thus, the frequency histogram in region m is calculating by:

$$f_i = \frac{bin(i)}{bin(1) + bin(2) + ... + bin(r)} \quad i = 1, 2, ..., r \tag{22}$$

After obtaining the frequency distribution in each region, the G-statistic distance measurement

is applied for two adjacent regions m and n:

$$\begin{aligned} G_{m,n} = &\sum_{m,n}\sum_{i=1}^{r} f_i \log f_i + (\sum_{m,n}\sum_{i=1}^{r} f_i)\log(\sum_{m,n}\sum_{i=1}^{r} f_i) \\ &- \sum_{m,n}(\sum_{i=1}^{r} f_i)\log(\sum_{i=1}^{r} f_i) - \sum_{i=1}^{r}(\sum_{m,n} f_i)\log(\sum_{m,n} f_i) \end{aligned} \tag{23}$$

Thus, the LiDAR texture distance can be expressed as $G_{m,n}^{\mathbf{L}^E}$. Similarly, the statistical distance of LiDAR area feature $\mathbf{L}^A$, denoted as $G_{m,n}^{\mathbf{L}^A}$, can be computed in the same way. Concerning $\mathbf{H}^E$ and $\mathbf{H}^A$ extracted from the hyperspectral feature $\mathbf{H}$, the above statistical calculation procedure is applied on each band, and $G_{m,n}^{\mathbf{H}_1^E}$, $G_{m,n}^{\mathbf{H}_2^E}$, ... , $G_{m,n}^{\mathbf{H}_K^E}$ and $G_{m,n}^{\mathbf{H}_1^A}$, $G_{m,n}^{\mathbf{H}_2^A}$, ..., $G_{m,n}^{\mathbf{H}_K^A}$ can be obtained correspondingly.

Since the information contained in each spectral band of $\mathbf{H}$ and LiDAR $\mathbf{L}$ is inconsistent, it is necessary to fuse these distance measures in a weighted manner, which is computed based on the homogeneity of each segmentation area [89]. Specifically, if the segmentation area has good homogeneity, it should has high weight. Conversely, if the segmented area is heterogeneous, the weight value should be small. In our framework, a locally adaptive approach is implemented.

$$\begin{aligned} G_{m,n}^{A} &= \sum_{j=1}^{K} max(f_i^{\mathbf{H}_j^A}) * G_{m,n}^{\mathbf{H}_j^A} + max(f_i^{\mathbf{L}^A}) * G_{m,n}^{\mathbf{L}^A} \\ G_{m,n}^{E} &= \sum_{j=1}^{K} max(f_i^{\mathbf{H}_j^E}) * G_{m,n}^{\mathbf{H}_j^E} + max(f_i^{\mathbf{L}^E}) * G_{m,n}^{\mathbf{L}^E} \end{aligned} \tag{24}$$

where $max(f_i)$ is the maximal frequency in corresponding band, while $G_{m,n}^{A}$ and $G_{m,n}^{E}$ represents the area distance and texture distance respectively.

As indicated so far, the dissimilarity of two adjacent region can be defined by:

$$D_{m,n} = G_{m,n}^{A} + G_{m,n}^{E} + \delta G_{m,n}^{S} \tag{25}$$

where δ is the balance factor addressing that the spectral distance and statistical distance are at the same order of magnitude. In our experiment, δ is set as 0.001 and the interval endpoints r is set as 17.

As mentioned before, the merging cost of region m and n is not only related to the dissimilarity $D_{m,n}$, but also related to the size of the region and the length of shared edges. The smaller the size of the region and the larger the shared boundary between the two regions, the easier the two regions merge together. Based on this point of view, the merging cost $C_{m,n}$ of region m and n is defined as:

$$C_{m,n} = \frac{1}{L_{m,n}} \frac{S_m S_n}{S_m + S_n} D_{m,n} \tag{26}$$

where $L_{m,n}$ is the length of shared boundary of region m and n, while S_m and S_n represents the number of pixels in region m and n, respectively.

3.3.2 Region Merging Procedure

A progressive region merging technique is introduced to effectively alleviate the conflict between over-segmentation and under-segmentation of superpixels and largely guarantee the homogeneity of the final merging map. Over-segmented superpixels ensure the homogeneity of each region, while region merging that gradually combines two adjacent similar regions does not introduce an under-segmentation problem with the help of shearlet extracted features.

Specifically, for the initial over-segmentation map $\mathbf{S}_N$, a data structure is utilized to record each pair of adjacent nodes with their merging cost, and a priority queue (denoted as Q) is built to store all these structure. Based on the queue, the structure with the smallest cost is chosen and the corresponding two regions (called m and n for similarity) are obtained as well. Subsequently, all structures related to m and n in the priority queue are removed. Through adding all points in region n into region m, some new structures are created to record the reconstructed region m and its neighborhood, which are then put into the priority queue. This progressive region merging procedure is carried out until the number of regions reaches a predefined value N. At last, the over-segmentation map $\mathbf{S}_N$ is gradually transformed into a final merging map $\mathbf{M}$.

3.4 Structure-Aware Filter Designing

For a point close to the edge, the surrounding labels are more likely to be different for object classifications, indicating that the neighboring spatial relationship should have less consideration. However, when it is located in the center of a local region, the surrounding objects tend to be the same, thus the neighboring spatial relationship should be paid more attention. This perspective motivates us to design an adaptive structure-aware filter whose kernel size changes with the distance from a point to the edge.

In fact, it is difficult to obtain accurate edges between objects in HSIs due to the inherent low-spatial resolution of remote sensing images. Fortunately, through applying the well-designed shearlet-based gradual region merging scheme on the SNIC over-segmentation map $\mathbf{S}_N$, a final merging map $\mathbf{M}$ with lower space constraint conflicts is thus achieved, in which the homogeneity of local regions is largely ensured. Meanwhile, the junctions between regions are regarded as edges. In particular, for each point p in $\mathbf{M}$, its region boundary must be a continuous closed curve, which means the number of edge points is limited. Therefore, all spatial distances between this point and its region boundary can be calculated. The smallest value is selected to form the distance map $\mathbf{D}$. This process can be expressed by the following formula:

$$\mathbf{D}_p = \underset{p' \in \mathcal{L}}{min}\{\sqrt{(p_x - p'_x)^2 + (p_y - p'_y)^2}\}, p \in \mathbf{P} \tag{27}$$

where $\mathbf{P}$ is the spatial position matrix of $\mathbf{M}$, p' represents a point of region boundary $\mathcal{L}$, (p_x, p_y) and (p'_x, p'_y) represents the two-dimensional spatial coordinates of point p and p' respectively.

However, the direction from different points p to their nearest boundary point is not fixed, implying that directly using $\mathbf{D}_p$ as the filtering size may cause the filter to be oversized and introduce some disturbing information of other ground objects. As we know, the diagonal of the square is longer

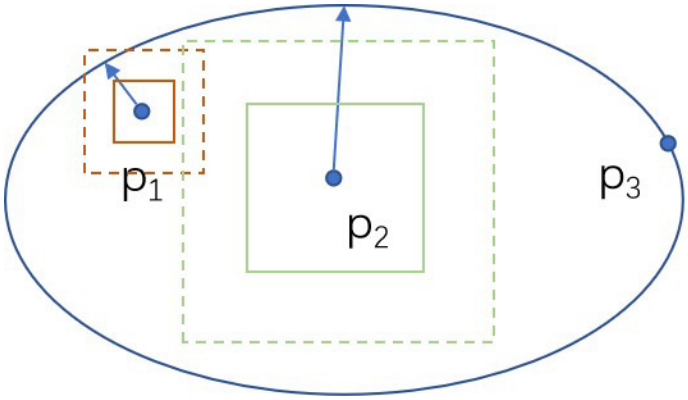


Figure 6: The process of filter size determination.

than the other inner straight lines. In other words, as long as the diagonal length of the adaptive-size filter is less than $\mathbf{D}_p$, the filter centered by point p will not exceed the boundary. Therefore, we convert the distance map $\mathbf{D}$ into the so-called scale map $\mathbf{S}$:

$$\mathbf{S}_p = 2 \times \lfloor (\frac{\mathbf{D}_p}{\sqrt{2}}) \rfloor + 1, p \in \mathbf{P} \tag{28}$$

In addition, when the point p is at the center of the region, an overly large filter size may contain more outside-region points, which could degrade the feature representation ability. Thus, a threshold-truncated method is introduced:

$$\mathbf{S}_p = \begin{cases} \mathbf{S}_p & \mathbf{S}_p > T \\ T & \mathbf{S}_p \leq T \end{cases} \tag{29}$$

In our experiments, the threshold T is simply set as 55.

Figure 6 illustrates the three circumstances of filter size determination procedure. Concretely, the dotted frame centered on $p1$ is the filter with $2 \times \lfloor \mathbf{D}_{p1} \rfloor + 1$, while the solid frame centered on $p1$ is the filter with $\mathbf{S}_{p1}$. For the point $p2$, the dotted frame and solid frame represent the filters without a threshold process and with threshold process, respectively. Clearly, the dotted frames of $p1$ and $p2$ are more precise for filter size than those two solid frames. Besides, for the region edge point such as $p3$, the filter size is only $1 * 1$, which obeys our filter size calculation process as well.

A final note is that all the points in the scale map $\mathbf{S}$ are assigned an odd value ranging from 1 to 55, indicating the filter size with each pixel. For each spatial pixel p, the corresponding structure-aware filter $\mathbf{F}_p \in \mathbb{R}^{\mathbf{S}_p \times \mathbf{S}_p}$ is formulated as:

$$\mathbf{F}_p = \frac{1}{\mathbf{S}_p \times \mathbf{S}_p} \mathbf{A}[\mathbf{S}_p, \mathbf{S}_p], p \in \mathbf{P} \tag{30}$$

where $\mathbf{A}$ represents a matrix that all elements values are 1. Obviously, $\mathbf{F}_p$ can be considered as a mean filter with adaptive size for each spatial pixel, which can be visually seen in Figure 1. Hence, the obtained adaptive-size filter achieves structure-aware based on the geometric position of the

Algorithm 1 ShearSAF for HSI and LiDAR feature extraction and classification

INPUT: raw HSI data $\mathbf{R} \in \mathbb{R}^{X \times Y \times B}$, LiDAR data $\mathbf{L} \in \mathbb{R}^{X \times Y}$;
OUTPUT: the classification map $\mathbf{C} \in \mathbb{R}^{X \times Y}$;
BEGIN
$N_p = 50$, $j_0 = 3$, $\delta = 0.001$, $r = 17$, $T = 55$;
using SNIC on $\mathbf{R}$ to obtain the over-segmentation superpixel map $\mathbf{S}_N$ with $X \times Y / N_p$ regions;
using KPCA on $\mathbf{R}$ to obtain the information-focused HSI data $\mathbf{H} \in \mathbb{R}^{X \times Y \times K}$;
using equation (18) and (19) to obtain the texture and area information $\mathbf{H}^E$, $\mathbf{L}^E$, $\mathbf{H}^A$ and $\mathbf{L}^A$;
using equation (26) to calculate the merging cost C;
conducting progressive region merging procedure on $\mathbf{S}_N$ and obtain the final merging map $\mathbf{M} \in \mathbb{R}^{X \times Y}$, where the number of regions N in $\mathbf{M}$ is calculated by equation (32);
using equation (27) to obtain distance map $\mathbf{D}$;
using equations (28) and (29) to obtain scale map $\mathbf{S}$;
using equation (30) to obtain structure-aware filter $\mathbf{F}_p$ for each spatial pixel p;
using equation (31) to obtain convolution feature $\mathbf{Z} \in \mathbb{R}^{X \times Y \times (K+1)}$ from $\mathbf{H}$ and $\mathbf{L}$;
using RF classifier on $\mathbf{Z}$ to achieve classification map $\mathbf{C}$;
END

convolution center. This flexible filter can well protect the difference of different objects on the edge, while reducing the abnormal points in the center area.

3.5 Feature Extraction and Classification

Since the edges in the final merging map $\mathbf{M}$ may not be accurate edges, classification errors can occur more frequently near the edge. Hence, the formulated structure-aware filter $\mathbf{F}$ is solely used for feature extraction rather than regularization of classification results. Taking the LiDAR data $\mathbf{L}$ as an example, the filtering process on each spatial pixel p can be expressed as follows:

$$\mathbf{Z}_{\mathbf{L}_p} = \mathbf{F}_p \otimes \mathbf{L}_p, p \in \mathbf{P} \tag{31}$$

where $\otimes$ is the convolution operator. After applying the convolution procedure on each pixel in $\mathbf{P}$, the feature $\mathbf{Z}_\mathbf{L} \in \mathbb{R}^{X \times Y}$ can be extracted. Similarly, through applying the convolution procedure on each band of $\mathbf{H}$, the corresponding feature cube $\mathbf{Z}_\mathbf{H} \in \mathbb{R}^{X \times Y \times K}$ can be obtained. By concatenating the both features $\mathbf{Z}_\mathbf{L}$ and $\mathbf{Z}_\mathbf{H}$ along the spectral direction, the final feature $\mathbf{Z} \in \mathbb{R}^{X \times Y \times (K+1)}$ can be thus achieved.

During classification, random forest (RF) classifier is chosen, which can not only achieve high classification accuracy but also possess fast computation speed. Meanwhile, RF has advantages for anti-overfitting and anti-noise. Notably, RF is consists of two steps: randomly selecting repeatable training subsets and building multiple decision trees, which involves bagging sampling techniques. In the experiments, the default subspace of RF is the floor of the logarithmic value of the features, and the number of trees in the forest is set as 500. Finally, by employing the RF classifier on the extracted feature $\mathbf{Z}$, the classification map $\mathbf{C}$ can be thus obtained. At last, the pseudocode of the proposed ShearSAF approach for HSI and LiDAR feature extraction and classification is outlined in Algorithm 1.

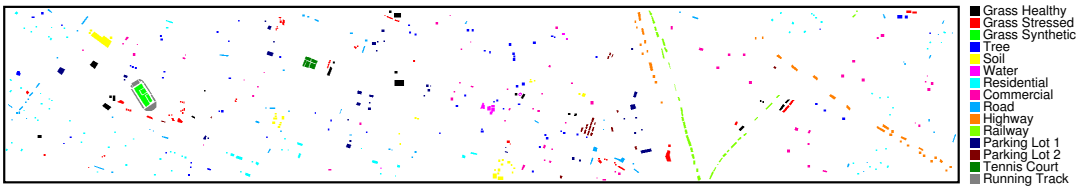


Figure 7: Ground-truth map of the Houston dataset (fifteen land-cover classes).

The computational complexity of our proposed ShearSAF can be divided into three parts. Firstly, the complexity of SNIC and SNIC-guided KPCA are $O(XY)$ and $O(XY + (X + Y)^2)$ respectively, while computational complexity of shearlet transform is $O(XYKlog(XY))$. Secondly, since the number of adjacent nodes for a region is limited, the computational complexity of priority queue is $O(\frac{XY}{N_p}log(\frac{XY}{N_p}))$ ($N_p = 50$ in our experiments). Thus the complexity of the region merging process is $O((\frac{XY}{N_p} - N)\frac{XY}{N_p}log(\frac{XY}{N_p}))$. Finally, the computational complexity of the convolution process and RF classification is $O(XYK)$ and $O(XYlog(K))$ respectively.

4 Experimental Data and Ablation Analysis

In this section, three real HSI and LiDAR datasets in diverse areas are used to evaluate the effectiveness of the proposed ShearSAF framework. Firstly, the three HSI and LiDAR datasets are presented. Secondly, the parameters contained in ShearSAF are analyzed. Thirdly, two ablation experiments are carried out to validate the advantage of the well-designed structure-aware filtering scheme and the superiority of the proposed ShearSAF method over other related filters.

4.1 Datasets

1). Houston dataset: The first dataset is captured over the University of Houston campus [19], in which the Houston HSI contains 144 spectral bands ranging from 380 to 1050nm. Each band contains 349 $\times$ 1905 pixels with 2.5m of spatial resolution. Meanwhile, the corresponding LiDAR data has the same spatial size with the height information of surface materials. Fifteen land-cover classes and 15,029 labeled samples are given in the ground-truth image, as shown in Table 2 and Figure 7.

2). Trento dataset: The second dataset is collected over the south of Trento, Italy, consisting of 63 spectral bands that range from 400 to 980nm [90]. Each band is 600 $\times$ 166 pixels with a spatial resolution of 1m. Likewise, the LiDAR data only has one band of the same spatial size. The six land-cover classes and 30,414 labeled pixels are listed in Table 3 and Figure 8.

3). MUUFL Gulfport Dataset: The third dataset was collected over the Gulf Park Campus of the University of Southern Mississippi [91, 92]. The spatial size of both HSI and LiDAR data is 325 $\times$ 220 with a spatial resolution of 1m. After removing eight noisy bands from the original 72 bands of the HSI data, 64 spectral bands are employed in the experiment. The details are given in Table 4 and Figure 9.

Table 2: Land-cover classes in the Houston dataset.

Class	Land-cover Type	No. of Samples
C1	Healthy Grass	1251
C2	Stressed Grass	1254
C3	Synthetic Grass	697
C4	Trees	1244
C5	Soil	1242
C6	Water	325
C7	Residential	1268
C8	Commercial	1244
C9	Roads	1252
C10	Highways	1227
C11	Railways	1235
C12	Parking Lot 1	1233
C13	Parking Lot 2	469
C14	Tennis Court	428
C15	Running Track	660
	Total	15,029

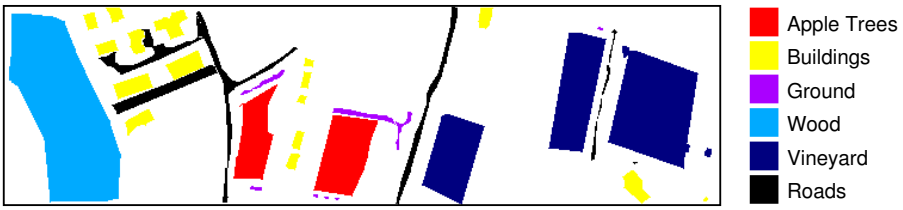


Figure 8: Ground-truth map of the Trento dataset (six land-cover classes).

4.2 Parameter Setting

In our proposed ShearSAF framework, there are several parameters that should be carefully specified. Concerning the scale parameter (j_0) for Shearlet transform, it is set as 3 according to their original paper. With respect to the weight parameters for SNIC, γ_1 and γ_2, they respectively corresponds to the spectral- and spatial-dimension, and thus are set as $\frac{1}{B}$ and 0.5. Meanwhile, the number of internal endpoints r is set as 17 to facilitate the subsequent G-statistic distance computation. For the dimension K in KPCA, the corresponding number should guarantee 99.5% energy is reserved.

In fact, there are two parameters in the gradual region merging procedure that are necessary to be determined: the initial number of pixels inside superpixel block N_p in the oversegmentation map $\mathbf{S}_N$ and the number of regions N in the final merging map $\mathbf{M}$. In fact, it is difficult to obtain the final number of homogeneous regions N as a fixed value for different datasets because of the impacts of object distributions, spatial complexity and so on. Here, we propose a heuristic way to calculate

Table 3: Land-cover classes in the Trento dataset.

Class	Land-cover Type	No. of Samples
C1	Apple Trees	4034
C2	Buildings	2903
C3	Ground	479
C4	Wood	9123
C5	Vineyard	10501
C6	Roads	3374
	Total	30,414

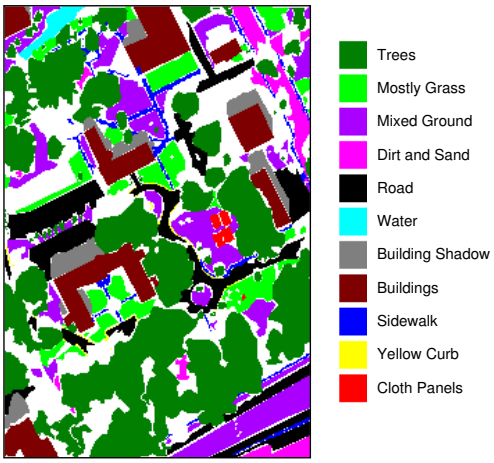


Figure 9: Ground-truth map of the MUUFL Gulfport dataset (eleven land-cover classes).

N, which contains the class number (C), spatial complexity (σ) and space size (X and Y).

$$N = \lfloor \frac{\sigma CXY}{10(X+Y)} \rfloor \tag{32}$$

where $\lfloor \cdot \rfloor$ is the floor operator. σ is defined as follows: the Sobel operator is adopted on the three normalized principal components of HSI and normalized LiDAR to calculate their gradients, and then the sum of absolute values divided by 10^5 is used as the spatial complexity. By this heuristic method, the σ is 7.63, 1.44, and 1.25 and the N is 3375, 112, and 180 for Houston, Trento and MUUFL Gulfport datasets, respectively.

To prove the effectiveness of our strategy, we conduct a series of experiments to track the process of gradual region merging and record the overall accuracy (OA, which is computed by dividing the correctly predicted samples with the number of testing ones) varying with different N_p and N. Figure 10 shows that the OA varies with the parameter N_p and N for the Houston, Trento and MUUFL Gulfport dataset. Here the parameter N_p ranges from 20 to 100 with the steps of 10, and then the parameter N ranges from 50 to 600 with the steps of 50 for the Trento dataset and MUUFL Gulfport dataset, while it ranges from 1000 to 6000 with steps of 500 for the Houston dataset. As far as the small sample set scenario is concerned, only 3, 5, 10 and 15 samples per class are randomly chosen

Table 4: Land-cover classes in the MUUFL Gulfport dataset.

Class	Land-cover Type	No. of Samples
C1	Trees	23,246
C2	Mostly Grass	4270
C3	Mixed Ground	6882
C4	Dirt and Sand	1826
C5	Roads	6687
C6	Water	466
C7	Building Shadows	2233
C8	Buildings	6240
C9	Sidewalks	1385
C10	Yellow Curbs	183
C11	Cloth Panels	269
	Total	53,687

Table 5: Parameter setting in the proposed ShearSAF approach.

Parameter	Value
dimension K in KPCA	99.5% energy reserved
weight parameter γ_1, γ_2 in SNIC	$1/B$ and 0.5 respectively
initial number of pixel inside superpixel N_p	50
shearlet scale parameter j_0	3
number of interval endpoints r in equation (21)	17
number of region in the final merging map N	heuristically decided by equation (32)

from the labeled set, and the remaining labeled samples are used for testing. Each experiment is executed 20 times to obtain the mean value. It can be seen from Figure 10 that the OA is better when N are respectively 3500, 100 and 200 for the three data sets, which are close to the values that are heuristically calculated by equation (32).

Two more observations can be found from Figure 10. First, the OA increases first and then decreases with decreasing N. This is reasonable because the adjacent regions with similar objects are merged to improve feature performance at the beginning, while two adjacent areas with different objects are merged after N reaches the critical value, leading to a decline in classification performance. The second is that OA has a slight increase with the decrease in N_p in the three datasets. In fact, the parameter N_p is used to ensure the homogeneity of each oversegmented region so that too many pixels inside the superpixel region would decrease the homogeneity. In the experiment, N_p is set as 50 for the three datasets, which not only keeps the region homogeneity but also promotes the calculation speed in region merging. To be more clear, Figure 11 illustrates the result of the gradual region merging procedure on the three data sets. It can be easily observed the structure information of various materials can be well represented.

At last, the parameter setting in the proposed ShearSAF approach are summarized in Table 5. Apparently, all the parameters included can be either preset and kept unchanged for different experimental datasets (such as N_p, j_0 and r) or heuristically computed (such as N and K), hence the robustness and generalization ability of ShearSAF can be guaranteed, which is a distinct advantage of the proposed ShearSAF approach.

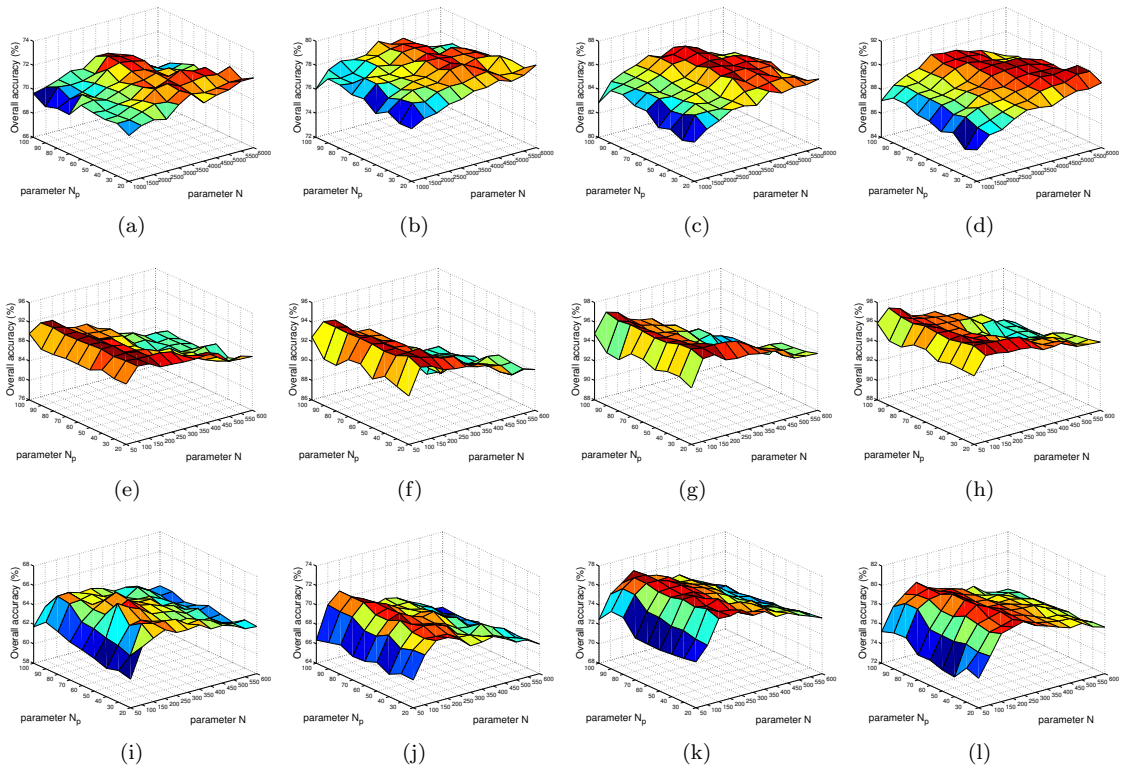
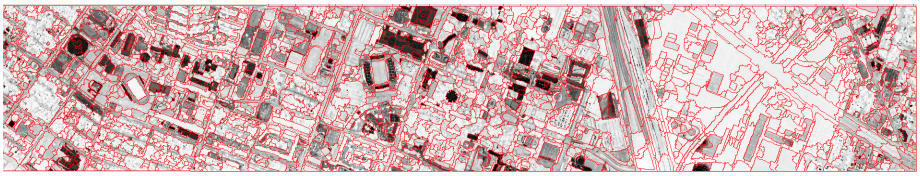


Figure 10: Overall accuracy vs the initial number of pixels inside superpixel (N_p) and the number of region in the final merging map (N) with different numbers of training samples per class on the Houston dataset (first row), Trento dataset (second row) and MUUFL Gulfport dataset (third row). The number of training samples per class is 3 (first column), 5 (second column), 10 (third column), and 15 (last column).

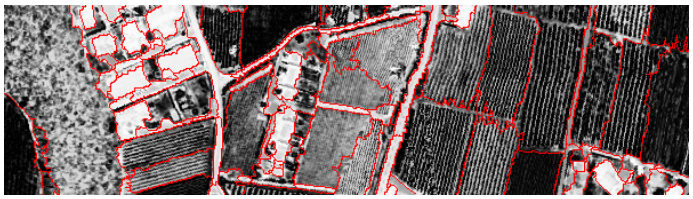
4.3 Ablation Analysis

In this part, two ablation experiments are carried out to validate the effectiveness and superiority of the proposed structure-aware filtering scheme. On the one hand, our ShearSAF is compared with fixed-size mean filters, whose kernel sizes range from 1 to 55 with a step of 2. That is, the features are obtained by convolving the KPCA-reduced HSI $\mathbf{H}$ and LiDAR $\mathbf{L}$ with the mean filter that has the fixed spatial size all the time, and the RF classifier is then employed. Similarly, the experiment is executed 20 times due to the small training sample scenario, and the OA of the mean filters with different size on the Houston, Trento and MUUFL Gulfport datasets is illustrated in Figure 12. It should be mentioned that the four curves from the bottom to the top (blue, red, green and black) indicate the performance of the mean filters with fixed-size under the conditions of 3, 5, 10 and 15 training samples per class as the training set, respectively, and correspondingly, the horizontal dotted lines from the bottom to the top (blue, red, green and black) represent the performance of ShearSAF with the same training set, respectively.

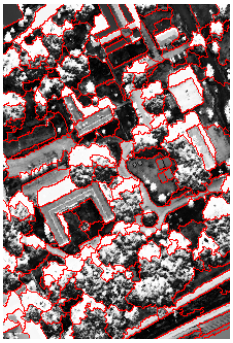
It can be easily observed from Figure 12 that the OA of the curve rises when the filter size is relatively small. Analytically, the ability to filter noise and abnormal points is improved as the kernel



(a) Houston



(b) Trento



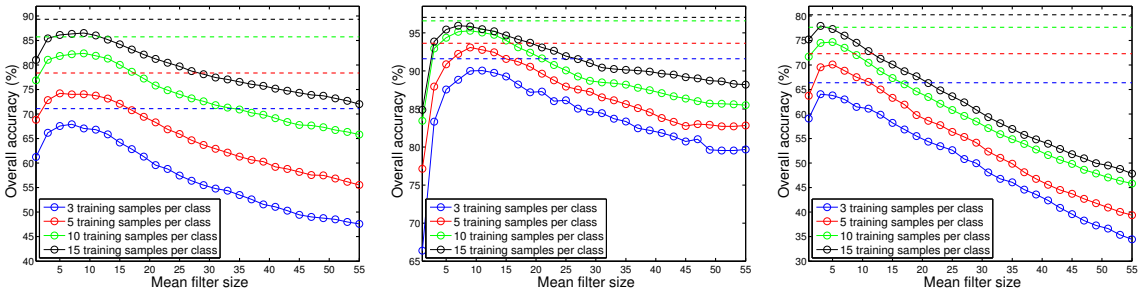
(c) MUUFL Gulfport

Figure 11: The result of the gradual region merging procedure.

size increases for considering more neighborhood relations. Then, the OA drops when the filter size increases continuously. This is because the continuous increase in the filter size will damage the feature performance at the junctions of objects. Moreover, it can be clearly seen that our ShearSAF approach always shows the best performance, implying that our structure-aware filter design does protect the edges and filter the noise in the center region. Besides, it is worth mentioning that the kernel size of the optimal filter is inconsistent for different data sets. For the three real data sets concerned here, as illustrated in Figure 12, the optimal filter size is 9, 7 and 3 for Houston, Trento and MUUFL Gulfport, respectively, and thus the filter size is hard to be determined in advance in practice. Alternatively, our structure-aware filter design can automatically adjust the filter size according to the well-designed scale map and achieve higher accuracy, indicating the advantage and feasibility of the proposed ShearSAF approach.

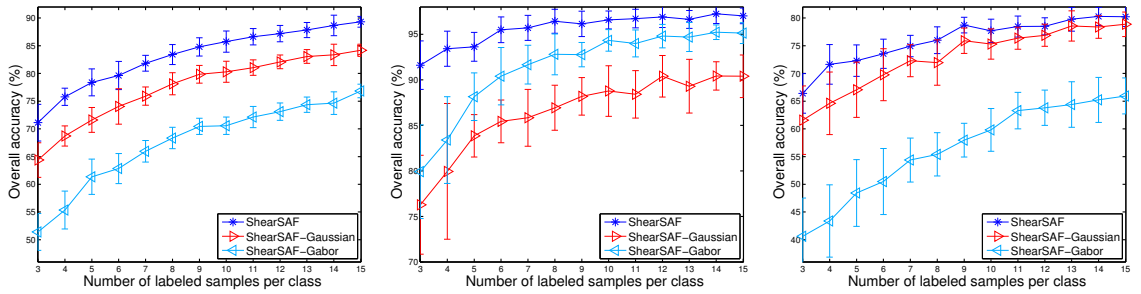
Alternatively, our structure-aware design with other filters is also examined, as illustrated in Figure 13. Here both the Gaussian and Gabor filters are taken into consideration.

Specifically, the ShearSAF-Guassian means that two-dimensional (2D) Gaussian with structure-aware size is applied on the stacked HSI and LiDAR data. In other words, we obtain the scale map in the same way as the ShearSAF, and each point in the obtained scale map represents the corresponding Gaussian filter size. Then the structure-aware Gaussian filters are convolved with the stacked HSI and LiDAR data to achieve the related features. At last, the RF classifier is utilized for classification. Similarly, a series of 2D Gabor filters (four scales and six orientations) with adaptive spatial size is applied on the stacked HSI and LiDAR data for feature extraction, called ShearSAF-Gabor. It can be seen from Figure 13 that ShearSAF-Gabor performs better than ShearSAF-Gaussian on the Trento dataset, while the opposite situation can be observed in the Houston and MUUFL Gulfport datasets. This is reasonable since the spatial distribution of objects



(a) (b) (c)

Figure 12: Overall accuracy as functions of the fixed-size mean filter on the a) Houston, b) Trento and c) MUUFL Gulfport datasets. The dotted lines with different colors represent the classification accuracy values of the corresponding training size by our ShearSAF.



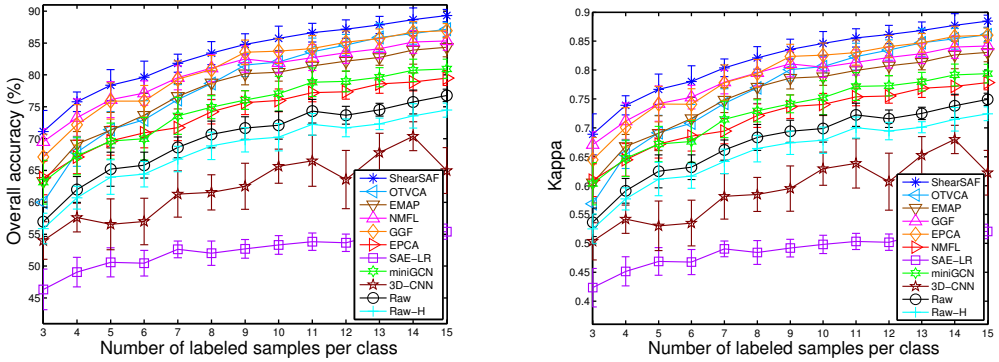
(a) (b) (c)

Figure 13: Overall accuracy of various spatial filters as functions of the number of labeled samples per class on the a) Houston, b) Trento and c) MUUFL Gulfport datasets.

in the Trento dataset (as shown in Figure 8) is more regular than the rest two datasets (as shown in Figure 7 and 9), the features obtained by the 2D Gabor filters with various orientations and scales can be more specific than those extracted by the Gaussian filter. Furthermore, the proposed ShearSAF constantly achieves the best results on the three datasets all the time, validating the importance and suitability of the simplest mean filter for our ShearSAF approach.

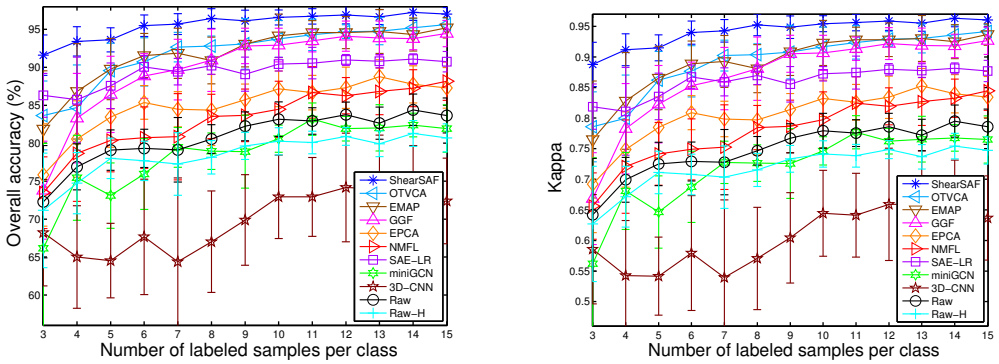
5 Experimental Results

In this section, a number of state-of-the-art feature extraction and fusion algorithms are incorporated to compare with the proposed ShearSAF approach. Firstly, two simplest methods, including the RF classifier on the raw HSI data (named as Raw-H) and on the concatenation of both HSI and LiDAR data (named as Raw), are used as the benchmark. Secondly, three deep learning-based methods, 3D-CNN (3-D convolutional neural network [35], a classic deep learning-based method that can simultaneously capture spatial-spectral joint information), miniGCN (mini-batch graph convolutional network [93], an emerging deep learning-based method that allows to train large-scale GCNs in a mini-batch fashion) and SAE-LR (stacked auto-encoder with logistic regression [94], an



(a) (b)

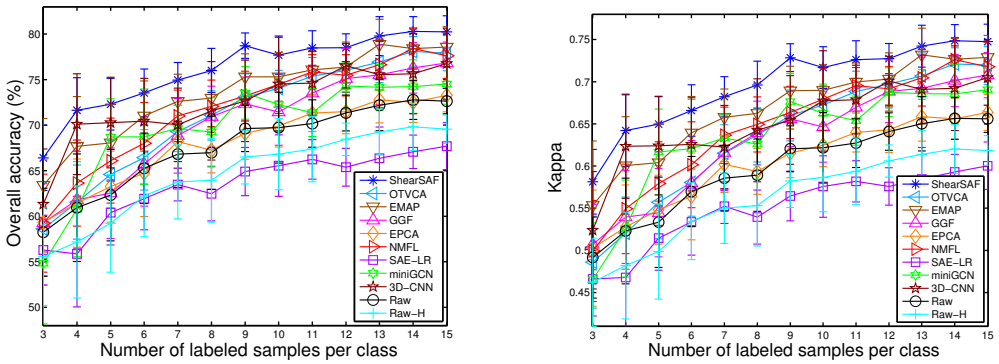
Figure 14: Houston dataset: (a) Overall accuracy and (b) Kappa as functions of the number of labeled samples per class for training.



(a) (b)

Figure 15: Trento dataset: (a) Overall accuracy and (b) Kappa as functions of the number of labeled samples per class for training.

auto-encoder-based deep learning method that can preserve abstract and invariant information in deeper features), are taken into consideration for HSI and LiDAR data classification. Thirdly, five widely-used feature extraction and fusion algorithms, that is, NMFL (nonlinear multiple feature learning based classification [95] that explores different types of available features in a collaborative and flexible way), EMAP (extended morphological attribute profile [96]), GGF (generalized graph-based fusion [23]), EPCA(a novel ensemble classifier [97]) and OTVCA (orthogonal total variation component analysis [98] that can get the best low rank representation and show strong anti-noise ability), are also conducted on both HSI and LiDAR data. For the classification issue, 3 to 15 samples per class are randomly selected from the labeled dataset to form the training set, while the rest are used for the testing set. At the same time, each experiment is run twenty times in order to reduce the effects of random factors. Both the mean values and standard deviations are reported. Except the OA measure, the kappa coefficient (κ), which reflects the impact of classes, is also adopted to evaluate the classification performance.



(a) (b)

Figure 16: MUUFL Gulfport dataset: (a) Overall accuracy and (b) Kappa as functions of the number of labeled samples per class for training.

Figures 14-16 show the OA and κ of the eleven compared methods including the Raw-H, Raw, 3D-CNN, miniGCN, SAE-LR, NMFL, EPCA, GGF, EMAP, OTVCA and our ShearSAF when the training set ranges from 3 to 15. It should be noted that, the OA obtained by a single LiDAR data is much smaller than that of other methods; thus, it has not been added for comparison. Generally, the classification performs better as the number of training sample grows for the three datasets. Compared to the Raw method, the performance of the Raw-H that just uses HSI data shows lower classification accuracies, confirming that the supplement of LiDAR information can improve the performance of HSI classification. Specifically, HSI data provides abundant spectral information for distinguishing materials with different physical properties, while LiDAR provides shape and height information that can be used to distinguish different targets of the same material. For the rest compared methods, the proposed ShearSAF outputs the highest results all the time, which is reasonable since the designed structure-aware filters can reduce its size to avoid interclass interference at the near edge and introduce more neighborhood information to reduce the environmental impact at the region center.

In addition, it should be noted that deep-learning based methods behaved badly for three dataset for limited training samples. Specifically, SAE-LR gives the worst performance on the Houston and MUUFL Gulfport dataset, while 3D-CNN performed the worst on the Trento dataset and the second worst on the Houston dataset. As for miniGCN, it is also lower than the traditional classification method in most cases. Analytically, the deep learning-based methods usually need a great quantity of training samples to constantly modify the magnanimous parameters in the process of training model. But the small sample set in the experiments significantly limits the performance of deep learning-based methods. Meanwhile, the training process of deep learning methods requires considerable time consumption as well.

Furthermore, when there are only five training samples per class, the classification performances, including each class accuracy, OA and κ of the eleven methods, have been summarized in Tables 6-8 for the Houston, Trento and MUUFL Gulfport datasets, respectively. It can be seen that ShearSAF outputs the best performance in most cases, which favors the superiority of our ShearSAF method.

Table 6: Classification performance using Raw-H, Raw, 3D-CNN, miniGCN, SAE-LR, NMFL, EPCA, GGF, EMAP, OTVCA and ShearSAF for the Houston dataset with five labeled samples per class as the training set.

Class	RAW-H	RAW	3D-CNN	miniGCN	SAE-LR	NMFL	EPCA	GGF	EMAP	OTVCA	ShearSAF
C1	86.49	86.81	77.37	91.52	68.78	79.98	86.76	79.20	86.18	73.49	**91.97**
C2	72.20	72.62	72.07	76.09	50.45	80.47	78.90	78.54	69.67	70.37	**82.66**
C3	93.67	94.78	91.53	97.27	94.33	**100.00**	**100.00**	99.91	99.89	98.44	98.16
C4	91.48	92.32	59.31	89.70	90.43	78.17	90.81	92.85	79.76	73.47	**95.61**
C5	78.61	78.86	92.61	**98.18**	72.20	91.86	88.29	79.10	83.21	90.93	89.67
C6	85.23	85.43	60.21	85.69	34.49	85.72	90.80	88.65	89.02	86.12	**91.25**
C7	56.28	61.06	30.32	70.82	36.17	76.96	75.71	**83.64**	70.12	55.22	70.81
C8	36.13	38.50	49.11	45.07	60.84	47.90	**62.19**	52.70	42.49	46.19	47.63
C9	64.72	65.43	30.73	48.84	65.79	47.45	71.87	72.20	65.41	54.39	**73.78**
C10	41.32	41.58	46.54	45.16	9.38	61.74	63.13	66.15	61.56	65.93	**67.85**
C11	50.94	52.39	47.65	49.00	26.56	49.77	**84.10**	72.15	68.49	75.87	74.91
C12	32.24	34.30	29.78	56.56	17.38	45.91	49.42	**59.25**	46.13	53.21	56.36
C13	21.80	23.34	26.48	32.05	26.63	43.04	58.73	48.62	58.66	77.88	**86.11**
C14	93.52	94.07	72.20	78.25	59.42	96.61	98.75	96.30	98.04	98.96	**99.24**
C15	86.63	86.70	86.06	99.42	37.09	98.68	99.75	98.30	98.58	98.30	**99.98**
OA	63.96	65.20	56.57	69.62	50.60	69.66	75.90	76.12	71.32	71.42	**78.38**
Kappa	0.61	0.62	0.53	0.67	0.47	0.67	0.74	0.74	0.69	0.69	**0.77**

Table 7: Classification performance using Raw-H, Raw, 3D-CNN, miniGCN, SAE-LR, NMFL, EPCA, GGF, EMAP, OTVCA and ShearSAF for the Trento dataset with five labeled samples per class as the training set.

Class	RAW-H	RAW	3D-CNN	miniGCN	SAE-LR	NMFL	EPCA	GGF	EMAP	OTVCA	ShearSAF
C1	80.10	79.99	80.21	66.94	83.49	65.62	87.49	95.32	91.13	94.07	**95.82**
C2	70.52	74.02	65.58	61.76	79.91	80.81	90.75	84.77	86.26	67.09	**92.43**
C3	96.00	96.25	88.66	86.62	72.51	84.80	82.09	96.42	**98.09**	86.94	89.95
C4	92.68	93.24	71.96	83.87	98.77	96.38	98.01	91.78	96.92	97.98	**98.77**
C5	72.14	72.58	53.36	67.59	89.66	69.26	69.45	82.11	85.15	91.28	**98.23**
C6	56.73	60.87	55.60	76.50	62.84	**79.45**	76.28	73.47	79.01	66.70	68.70
OA	78.01	79.09	64.53	73.10	87.56	80.39	83.45	86.43	89.76	89.57	**93.62**
Kappa	0.71	0.73	0.54	0.65	0.84	0.74	0.78	0.82	0.87	0.86	**0.91**

In more detail, considering the C5 class (Vineyard) of the Trento dataset, it can be found from the ground-truth map (Figure 8) that the spatial distribution of C5 is very regular, and ShearSAF effectively filters the noise in the area and protects the edges; thus, the performance increases from 72.58% for the RAW method to 98.23% for our approach, as illustrated in Table 7. Alternatively, concerning the C10 class (Yellow Curbs) of the MUUFL Gulfport dataset in Table 8, it has scattering in the scene and is even hard to be seen in Figure 9. Although our method is not optimal in the C10 class, this structure-aware filter does work for reducing its own size and keeping the target information from being interfered by neighboring objects. To illustrate, the groundtruth and the complete classification maps for all the three datasets of the eleven compared methods are shown in Figures 17-19. It can be easily observed that our ShearSAF approach is outstanding compared to the others, demonstrating the effectiveness of the proposed method.

Finally, when there are five training samples per class, the computation time is given in Table 9, which was recorded by a workstation with a 24-core Intel processor at 2.20 GHz with 128 GB RAM. As expect, the deep learning-based method (3D-CNN, miniGCN and SAE-LR) takes more times than the others because model training and parameter optimization require considerable time. It

Table 8: Classification performance using Raw-H, Raw, 3D-CNN, miniGCN, SAE-LR, NMFL, EPCA, GGF, EMAP, OTVCA and ShearSAF for the MUUFL Gulfport dataset with five labeled samples per class as the training set.

Class	RAW-H	RAW	3D-CNN	miniGCN	SAE-LR	NMFL	EPCA	GGF	EMAP	OTVCA	ShearSAF
C1	67.06	70.45	83.15	79.98	68.09	69.77	63.27	61.10	70.13	75.05	**83.52**
C2	70.85	71.44	68.84	**90.58**	58.11	67.42	70.54	72.06	73.26	53.32	72.54
C3	42.47	42.95	37.43	38.60	30.16	50.18	36.47	52.52	40.68	32.49	**58.26**
C4	47.33	47.38	61.81	59.96	61.73	68.54	46.71	49.72	61.84	63.78	**65.74**
C5	69.63	72.68	66.42	81.13	57.12	73.75	76.04	79.67	80.58	54.02	**82.70**
C6	84.24	86.15	61.03	94.41	65.21	98.40	97.99	96.78	96.75	**99.87**	87.55
C7	62.38	68.86	71.62	76.19	55.73	72.29	82.60	76.64	77.69	74.99	**89.25**
C8	32.23	39.17	74.54	78.41	**79.28**	59.27	73.59	52.81	70.90	66.80	64.08
C9	42.46	43.83	38.06	40.08	29.70	45.47	39.57	45.62	39.87	34.40	**48.61**
C10	61.07	61.64	49.21	76.40	40.82	**84.78**	53.33	61.45	59.73	69.40	61.72
C11	90.37	90.65	61.23	72.30	42.86	74.31	93.09	88.90	94.33	**96.49**	89.96
OA	59.23	62.32	70.28	68.69	60.40	66.11	63.25	62.53	68.05	64.60	**72.30**
Kappa	0.50	0.53	0.62	0.62	0.51	0.58	0.55	0.54	0.60	0.56	**0.65**

Table 9: The running time (in seconds) for different algorithms when five samples per class are used for training.

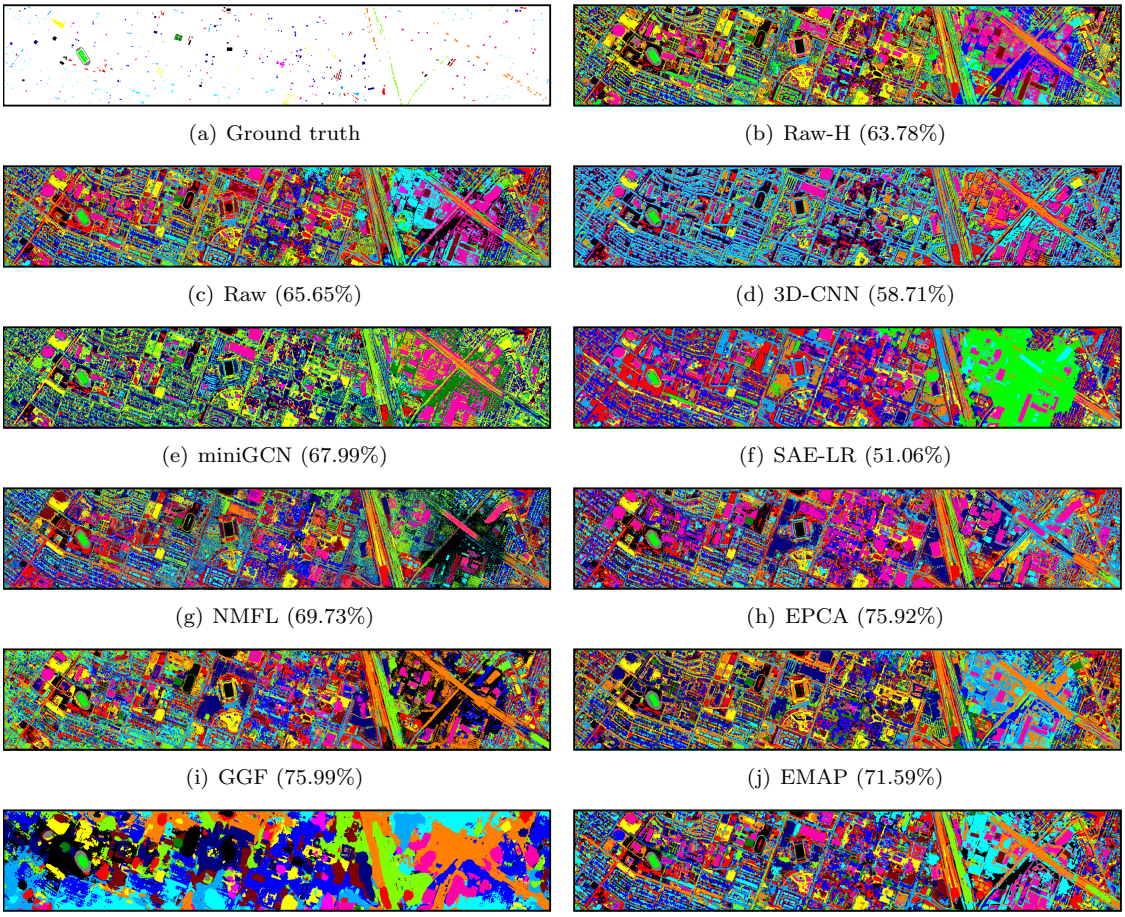
Methods	3D-CNN	miniGCN	SAE-LR	NMFL	EPCA	GGF	EMAP	OTVCA	ShearSAF
Houston	2576.78	9105.33	7576.49	681.47	896.98	621.54	482.40	1092.23	462.73
Trento	324.52	1628.74	1348.92	79.75	95.28	70.10	46.56	181.33	43.04
MUUFL Gulfport	296.60	1261.83	1120.77	65.76	77.74	56.05	36.76	144.31	34.58

can be observed that the time cost of our ShearSAF method is less than that of the other methods, which is mainly due to the irrelevance of the structure-aware feature extraction procedure with the training set. That is to say, the feature extraction procedure in ShearSAF method is executed only once, while RF classifier has low computational cost, therefore the proposed ShearSAF method is computationally efficient is applicable for remote sensing image with large spatial size, which proves the superiority of our method once again.

6 Conclusions

In this paper, a newly-designed shearlet-based structure-aware filtering approach has been proposed for HSI and LiDAR feature extraction. Specifically, the shearlet transform is implemented on the KPCA-reduced HSI and LiDAR data for area and texture feature extraction. Then, the spectral, area and texture features are used to guide the gradual region merging procedure, which converts the initial over-segmentation map into a final merging map, and the spatial structure of objects can be well characterized. By calculating the edge distance in the final merging map, the scale map can be acquired, which is utilized to adaptively select the filter size for convolution. Finally, the RF classifier is used for classification.

In summary, the most important contribution of this article involves the design of the structure-aware filtering design. In this process, we innovatively proposed a shearlet-based area and texture feature representation that could effectively measure the distance between two adjacent areas. At the same time, the structure-aware filter is constructed in an elegant manner to ensure that the



(a) Ground truth (b) Raw-H (63.78%)

(c) Raw (65.65%) (d) 3D-CNN (58.71%)

(e) miniGCN (67.99%) (f) SAE-LR (51.06%)

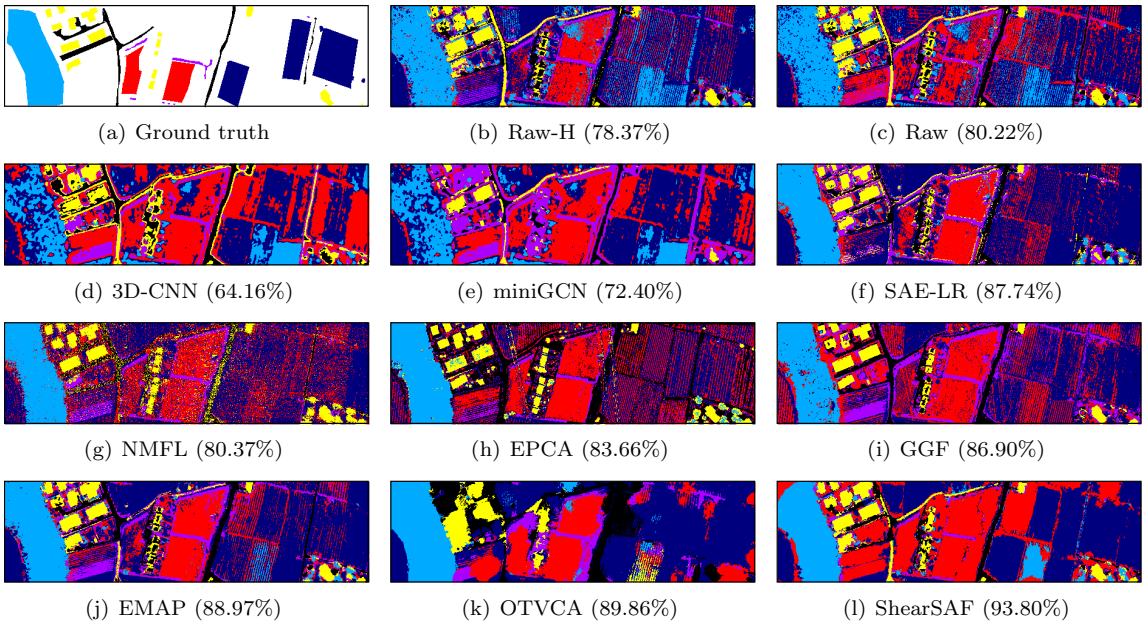
(g) NMFL (69.73%) (h) EPCA (75.92%)

(i) GGF (75.99%) (j) EMAP (71.59%)

(k) OTVCA (71.53%) (l) ShearSAF (78.79%)

Figure 17: Houston dataset: a) Ground truth map, the rest are classification maps obtained by b) Raw-H, c) Raw, d) 3D-CNN, e) miniGCN, f) SAE-LR, g) NMFL, h) EPCA, i) GGF, j) EMAP, k) OTVCA and l) ShearSAF when the number of training samples is five per class (the percentage in the brackets is the corresponding accuracy).

pixel near the edge could has a small-size kernel to protect the information from being disturbed by nearby objects, while the point at the center of the area could has a larger kernel size to filter noise and abnormal points. Two ablation experiments with various fixed-size mean filters and other adaptive-size filters (Gaussian and Gabor) demonstrate the effectiveness of the proposed ShearSAF method. Meanwhile, comparison with several state-of-the-art methods (3D-CNN, miniGCN, SAE-LR, NMFL, EPCA, GGF, EMAP and OTVCA) constantly show the superiority of the proposed ShearSAF approach. At last, we still want to emphasize that the structure-aware filtering design presented here can be further embedded with other kinds of feature. For instance, manifold learning-based methods, such as LLE (locally linear embedding) and ISOMAP (isometric mapping), can be used for dimension reduction and feature extraction. Furthermore, structure-aware filter design pattern can be integrated with other central-based filters (including Gaussian and Median filter) to



(a) Ground truth (b) Raw-H (78.37%) (c) Raw (80.22%)

(d) 3D-CNN (64.16%) (e) miniGCN (72.40%) (f) SAE-LR (87.74%)

(g) NMFL (80.37%) (h) EPCA (83.66%) (i) GGF (86.90%)

(j) EMAP (88.97%) (k) OTVCA (89.86%) (l) ShearSAF (93.80%)

Figure 18: Trento dataset: a) Ground truth map, the rest are classification maps obtained by b) Raw-H, c) Raw, d) 3D-CNN, e) miniGCN, f) SAE-LR, g) NMFL, h) EPCA, i) GGF, j) EMAP, k) OTVCA and l) ShearSAF when the number of training samples is five per class (the percentage in the brackets is the corresponding accuracy).

extract discriminative feature and improve the robustness of the whole framework. All these aspects are worthy of more attention.

References

[1] J. Richards, *Remote sensing digital image analysis: an introduction*. Springer, 2013.

[2] G. Camps-Valls, D. Tuia, L. Gómez-Chova, S. Jiménez, and J. Malo, *Remote Sensing Image Processing*. Morgan & Claypool, Dec. 2011.

[3] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot, "Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches," *J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 2, pp. 354–379, 2012.

[4] J. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot, "Hyperspectral remote sensing data analysis and future challenges," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 2, pp. 6–36, Jun. 2013.

[5] M. Khodadadzadeh, J. Li, S. Prasad, and A. Plaza, "Fusion of hyperspectral and lidar remote sensing data using multiple feature learning," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2971–2983, 2015.

[6] M. Kishore and S. Kulkarni, "Approches and challenges in classification for hyperspectral data: A review," in *Proc. Int. Conf. Electr. Electron. Optim. Techn. (ICEEOT)*, Mar. 2016, pp. 3418–3421.

[7] S. Jia, Z. Zhu, L. Shen, and Q. Li, "A two-stage feature selection framework for hyperspectral image classification using few labeled samples," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 7, no. 4, pp. 1023–1035, Apr. 2014.

[8] Y. Zhou, J. Peng, and C. Chen, "Dimension reduction using spatial and spectral regularized local discriminant embedding for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 2, pp. 1082–1095, Feb. 2015.

[9] P. Hartzell, C. Glennie, and S. Khan, "Terrestrial hyperspectral image shadow restoration through lidar fusion," *Remote Sens.*, vol. 9, no. 5, 2017.

[10] S. Sun and C. Salvaggio, "Aerial 3d building detection and modeling from airborne lidar point clouds," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 6, no. 3, pp. 1440–1449, 2013.

[11] C. Paris and L. Bruzzone, "A three-dimensional model-based approach to the estimation of the tree top height by fusing low-density lidar data and very high resolution optical images," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 1, pp. 467–480, 2015.

[12] P. Ghamisi and B. Höfle, "Lidar data classification using extinction profiles and a composite kernel support vector machine," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 659–663, 2017.

[13] J. Rau, J. Jhan, and Y. Hsu, "Analysis of oblique aerial images for land cover and point cloud classification in an urban environment," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1304–1319, 2015.

[14] M. Khodadadzadeh, J. Li, S. Prasad, and A. Plaza, "Fusion of hyperspectral and lidar remote sensing data using multiple feature learning," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 8, no. 6, pp. 2971–2983, Jun. 2015.

[15] M. Soleimanzadeh, A. Karami, and P. Scheunders, "Fusion of hyperspectral and lidar images using non-subsampled shearlet transform," in *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2018, pp. 8873–8876.

[16] M. Zhang, P. Ghamisi, and W. Li, "Classification of hyperspectral and lidar data using extinction profiles with feature fusion," *Remote Sens. Lett.*, vol. 8, no. 10, pp. 957–966, Oct. 2017.

[17] B. Bigdeli and P. Pahlavani, "A dempster shafer-based fuzzy multisensor fusion system using airborne lidar and hyperspectral imagery," *Int. J. Remote Sens.*, pp. 7718–7737, Jun. 2018.

[18] C. Ge, Q. Du, W. Li, Y. Li, and W. Sun, "Hyperspectral and lidar data classification using kernel collaborative representation based residual fusion," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 12, no. 6, pp. 1963–1973, 2019.

[19] C. Debes, A. Merentitis, R. Heremans, J. Hahn, N. Frangiadakis, T. Kasteren, W. Liao, R. Bellens, A. Pizurica, S. Gautama, W. Philips, S. Prasad, Q. Du, and F. Pacifici, "Hyperspectral and LiDAR data fusion: Outcome of the 2013 GRSS data fusion contest," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2405–2418, Jun. 2014.

[20] Z. Zhong, B. Fan, K. Ding, H. Li, S. Xiang, and C. Pan, "Efficient multiple feature fusion with hashing for hyperspectral imagery classification: A comparative study," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4461–4478, Aug. 2016.

[21] B. Rasti, P. Ghamisi, and R. Gloaguen, "Hyperspectral and lidar fusion using extinction profiles and total variation component analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3997–4007, 2017.

[22] W. Chen, X. Dai, B. Pan, and T. Huang, "A novel discriminant criterion based on feature fusion strategy for face recognition," *Neurocomputing*, vol. 159, pp. 67–77, Jul. 2015.

[23] W. Liao, A. Pizurica, R. Bellens, S. Gautama, and W. Philips, "Generalized graph-based fusion of hyperspectral and lidar data using morphological features," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 3, pp. 552–556, 2015.

[24] Z. Ye, S. Prasad, W. Li, J. Fowler, and M. He, "Classification based on 3-d dwt and decision fusion for hyperspectral image analysis," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 1, pp. 173–177, 2014.

[25] K. Schindler, "An overview and comparison of smooth labeling methods for land-cover classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4534–4545, 2012.

[26] W. Liao, R. Bellens, A. Pizurica, S. Gautama, and W. Philips, "Combining feature fusion and decision fusion for classification of hyperspectral and lidar data," in *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2014, pp. 1241–1244.

[27] R. Luo, W. Liao, H. Zhang, Y. Pi, and W. Philips, "Classification of cloudy hyperspectral image and lidar data based on feature fusion and decision fusion," in *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2016, pp. 2518–2521.

[28] R. Luo, W. Liao, H. Zhang, L. Zhang, P. Scheunders, Y. Pi, and W. Philips, "Fusion of hyperspectral and lidar data for classification of cloud-shadow mixed remote sensed scene," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 8, pp. 3768–3781, 2017.

[29] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, 2014.

[30] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, "Deep supervised learning for hyperspectral data classification through convolutional neural networks," in *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2015, pp. 4959–4962.

[31] M. Zhang, W. Li, and Q. Du, "Diverse region-based cnn for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2623–2634, 2018.

[32] A. Ben Hamida, A. Benoit, P. Lambert, and C. Ben Amar, "3-d deep learning approach for remote sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4420–4434, 2018.

[33] L. Mou, P. Ghamisi, and X. X. Zhu, "Deep recurrent neural networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3639–3655, 2017.

[34] A. Ma, A. Filippi, Z. Wang, and Z. Yin, "Hyperspectral image classification using similarity measurements-based deep recurrent neural networks," *Remote Sens.*, vol. 11, no. 2, 2019.

[35] Y. Li, H. Zhang, and Q. Shen, "Spectral-spatial classification of hyperspectral imagery with 3d convolutional neural network," *Remote Sens.*, vol. 9, no. 1, 2017.

[36] P. Ghamisi, B. Höfle, and X. X. Zhu, "Hyperspectral and lidar data fusion using extinction profiles and deep convolutional neural network," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 6, pp. 3011–3024, 2017.

[37] Q. Cao, Y. Zhong, A. Ma, and L. Zhang, "Urban land use/land cover classification based on feature fusion fusing hyperspectral image and lidar data," in *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2018, pp. 8869–8872.

[38] S. Yu, S. Jia, and C. Xu, "Convolutional neural networks for hyperspectral image classification," *Neurocomputing*, vol. 219, pp. 88–98, Jan. 2017.

[39] B. Liu, X. Yu, P. Zhang, X. Tan, A. Yu, and Z. Xue, "A semi-supervised convolutional neural network for hyperspectral image classification," *Remote Sens. Lett.*, vol. 8, no. 9, pp. 839–848, 2017.

[40] H. Wu and S. Prasad, "Semi-supervised deep learning using pseudo labels for hyperspectral image classification," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1259–1270, 2018.

[41] B. Pan, Z. Shi, and X. Xu, "R-vcanet: A new deep-learning-based hyperspectral image classification method," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 10, no. 5, pp. 1975–1986, 2017.

[42] B. Pan, Z. Shi, and X. Xu, "Mugnet: Deep learning for hyperspectral image classification using limited samples," *ISPRS J. Photogramm. Remote Sens.*, vol. 145, pp. 108–119, 2018.

[43] K. Guo, D. Labate, W.-Q. Lim, G. Weiss, and E. Wilson, "Wavelets with composite dilations and their mra properties," *Appl. Comput. Harmon. Anal.*, vol. 20, pp. 202–236, Mar. 2006.

[44] G. Easley, D. Labate, and W.-Q. Lim, "Sparse directional image representations using the discrete shearlet transform," *Appl. Comput. Harmon. Anal.*, vol. 25, pp. 25–46, Aug. 2008.

[45] S. Jia, L. Shen, J. Zhu, and Q. Li, "A 3-d gabor phase-based coding and matching framework for hyperspectral imagery classification," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1176–1188, Apr. 2018.

[46] S. Jia, Z. Lin, B. Deng, J. Zhu, and Q. Li, "Cascade superpixel regularized gabor feature fusion for hyperspectral image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 5, pp. 1638–1652, May 2020.

[47] E. Candès, L. Demanet, D. Donoho, and L. Ying, "Fast discrete curvelet transforms," *SIAM Journal on Multiscale Modeling and Simulation*, vol. 5, Sep. 2006.

[48] J. Ma and G. Plonka, "A review of curvelets and recent applications," *IEEE Signal Process. Mag.*, vol. 27, Apr. 2011.

[49] A. L. Da Cunha, J. Zhou, and M. N. Do, "The nonsubsampled contourlet transform: Theory, design, and applications," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 3089–3101, 2006.

[50] W. Lim, "The discrete shearlet transform: A new directional transform and compactly supported shearlet frames," *IEEE Trans. Image Process.*, vol. 19, no. 5, pp. 1166–1180, 2010.

[51] G. R. Easley, D. Labate, and W. Lim, "Optimally sparse image representations using shearlets," in *Fortieth Asilomar Conference on Signals, Systems and Computers*, 2006, pp. 974–978.

[52] P. S. Negi and D. Labate, "3-d discrete shearlet transform and video processing," *IEEE Trans. Image Process.*, vol. 21, no. 6, pp. 2944–2954, 2012.

[53] W. Lim, "Nonseparable shearlet transform," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 2056–2065, 2013.

[54] S. Yi, D. Labate, G. R. Easley, and H. Krim, "A shearlet approach to edge analysis and detection," *IEEE Trans. Image Process.*, vol. 18, no. 5, pp. 929–941, 2009.

[55] K. Guo, D. Labate, and W.-Q. Lim, "Edge analysis and identification using the continuous shearlet transform," *Appl. Comput. Harmon. Anal.*, vol. 27, pp. 24–46, Jul. 2009.

[56] M. A. Duval-Poo, F. Odone, and E. De Vito, "Edges and corners with shearlets," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3768–3780, 2015.

[57] G. R. Easley, D. Labate, and F. Colonna, "Shearlet-based total variation diffusion for denoising," *IEEE Trans. Image Process.*, vol. 18, no. 2, pp. 260–268, 2009.

[58] S. Häuser and G. Steidl, "Convex multiclass segmentation with shearlet regularization," *International Journal of Computer Mathematics - IJCM*, vol. 90, Dec.12 2011.

[59] Y. Li, L. Po, C. Cheung, X. Xu, L. Feng, F. Yuan, and K. Cheung, "No-reference video quality assessment with 3d shearlet transform and convolutional neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 6, pp. 1044–1057, 2016.

[60] M. Zaouali, S. Bouzidi, and E. Zagrouba, "3-d shearlet transform based feature extraction for improved joint sparse representation hsi classification," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 11, no. 4, pp. 1306–1314, 2018.

[61] H. Rezaei, A. Karami, and P. Scheunders, "Hyperspectral and multispectral image fusion based on spectral matching in the shearlet domain," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGRASS)*, 2018, pp. 8070–8073.

[62] A. Moore, S. Prince, J. Warrell, U. Mohammed, and G. Jones, "Superpixel lattices," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2008, pp. 1–8.

[63] W. Wang, D. Xiang, Y. Ban, J. Zhang, and J. Wan, "Superpixel segmentation of polarimetric sar images based on integrated distance measure and entropy rate method," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 10, no. 9, pp. 4045–4058, 2017.

[64] F. Meng, H. Li, Q. Wu, B. Luo, C. Huang, and K. N. Ngan, "Globally measuring the similarity of superpixels by binary edge maps for superpixel clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 4, pp. 906–919, 2018.

[65] S. Patel and B. Kadhiwala, "Comparative analysis of cluster based superpixel segmentation techniques," in *2nd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2018, pp. 1454–1459.

[66] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, 2012.

[67] R. Achanta and S. Süsstrunk, "Superpixels and polygons using simple non-iterative clustering," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4895–4904.

[68] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, 2000.

[69] M. Liu, O. Tuzel, S. Ramalingam, and R. Chellappa, "Entropy rate superpixel segmentation," in *IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2011, pp. 2097–2104.

[70] Z. Hu, Q. Zou, and Q. Li, "Watershed superpixel," in *IEEE Int. Conf. Image Process. (ICIP)*, 2015, pp. 349–353.

[71] N. Zhang and L. Zhang, "SSGD: Superpixels using the shortest gradient distance," in *IEEE Int. Conf. Image Process. (ICIP)*, 2017, pp. 3869–3873.

[72] Y. Guo, L. Jiao, S. Wang, S. Wang, F. Liu, and W. Hua, "Fuzzy superpixels for polarimetric sar images classification," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 5, pp. 2846–2860, 2018.

[73] C. Wu, J. Zheng, Z. Feng, H. Zhang, L. Zhang, J. Cao, and H. Yan, "Fuzzy SLIC: Fuzzy simple linear iterative clustering," *IEEE Trans. Circuits Syst. Video Technol.*, Aug. 2020.

[74] S. Jia, X. Deng, J. Zhu, M. Xu, J. Zhou, and X. Jia, "Collaborative representation-based multiscale superpixel fusion for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 10, pp. 7770–7784, 2019.

[75] Q. Leng, H. Yang, J. Jiang, and Q. Tian, "Adaptive multiscale segmentations for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 8, pp. 5847–5860, 2020.

[76] L. Shen and S. Jia, "Three-dimensional Gabor wavelets for pixel-based hyperspectral imagery classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 12, pp. 5039–5046, Dec. 2011.

[77] F. Mirzapour and H. Ghassemian, "Multiscale gaussian derivative functions for hyperspectral image feature extraction," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 4, pp. 525–529, 2016.

[78] Y. Teng, Y. Zhang, Y. Chen, and C. Ti, "Adaptive morphological filtering method for structural fusion restoration of hyperspectral images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 2, pp. 655–667, 2016.

[79] S. Wu, J. Zhang, C. Shi, and W. Li, "Multiscale spectral-spatial hyperspectral image classification with adaptive filtering," in *IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, 2018, pp. 2591–2594.

[80] Z. Sun, Z. Zhang, Y. Chen, S. Liu, and Y. Song, "Frost filtering algorithm of SAR images with adaptive windowing and adaptive tuning factor," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 6, pp. 1097–1101, 2020.

[81] Y. Yang, W. Wan, S. Huang, F. Yuan, S. Yang, and Y. Que, "Remote sensing image fusion based on adaptive ihs and multiscale guided filter," *IEEE Access*, vol. 4, pp. 4573–4582, 2016.

[82] C. Kadam and S. B. Borse, "An improved image denoising using spatial adaptive mask filter for medical images," in *International Conference on Computing, Communication, Control and Automation (ICCUBEA)*, 2017, pp. 1–5.

[83] D. Labate, W.-Q. Lim, G. Kutyniok, and G. Weiss, "Sparse multidimensional representation using shearlets," *Wavelets XI*, vol. 5914, Aug. 2005.

[84] S. Huser and G. Steidl. (2014) Fast finite shearlet transform. [Online]. Available: https://arxiv.org/abs/1202.1773

[85] M. Fauvel, J. Chanussot, and J. Atli-Benediktsson, "Kernel principal component analysis for the classification of hyperspectral remote sensing data over urban areas," *EURASIP J. Adv. Signal Process.*, pp. 1–14, 2009.

[86] H. Halim, S. Isa, and S. Mulyono, "Comparative analysis of pca and kpca on paddy growth stages classification," in *2016 IEEE Region 10 Symposium (TENSYMP)*, May. 2016, pp. 167–172.

[87] S. Jia, Z. Zhan, M. Zhang, M. Xu, Q. Huang, J. Zhou, and X. Jia, "Multiple feature-based superpixel-level decision fusion for hyperspectral and lidar data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 2, pp. 1437–1452, 2020.

[88] A. Karami, R. Heylen, and P. Scheunders, "Band-specific shearlet-based hyperspectral image noise reduction," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 9, pp. 5054–5066, 2015.

[89] Z. Hu, Z. Wu, Q. Zhang, Q. Fan, and J. Xu, "A spatially-constrained color–texture model for hierarchical vhr image segmentation," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 1, pp. 120–124, 2013.

[90] M. Zhang, W. Li, Q. Du, L. Gao, and B. Zhang, "Feature extraction for classification of hyperspectral and lidar data using patch-to-patch cnn," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 100–111, 2020.

[91] P. Gader, A. Zare, R. Close, J. Aitken, and G. Tuell, "Muufl gulfport hyperspectral and lidar airborne data set," *University of Florida, Gainesville*, 2013.

[92] X. Du and A. Zare, "Technical report: Scene label ground truth map for muufl gulfport data set," *University of Florida, Gainesville*, 2017.

[93] D. Hong, L. Gao, J. Yao, B. Zhang, A. Plaza, and J. Chanussot, "Graph convolutional networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, pp. 1–13, 2020.

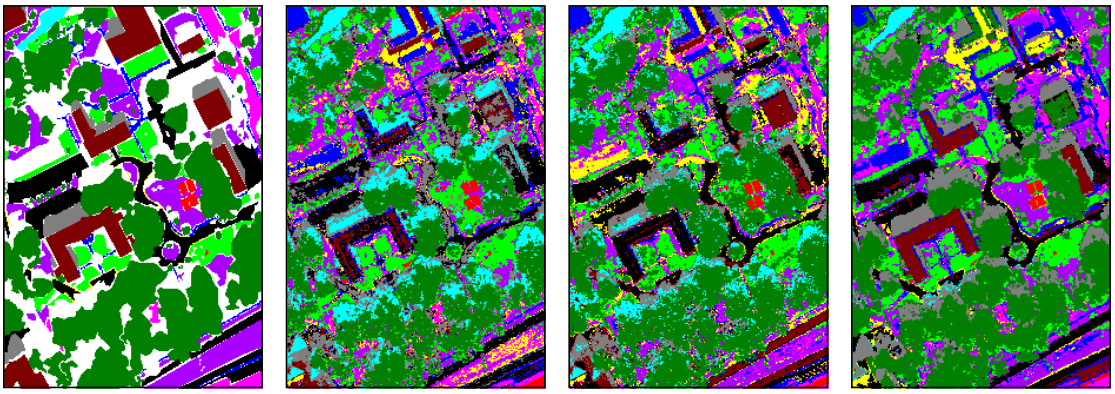
[94] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, "Deep learning-based classification of hyperspectral data," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, 2014.

[95] J. Li, X. Huang, P. Gamba, J. M. Bioucas-Dias, L. Zhang, J. A. Benediktsson, and A. Plaza, "Multiple feature learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 3, pp. 1592–1606, 2015.

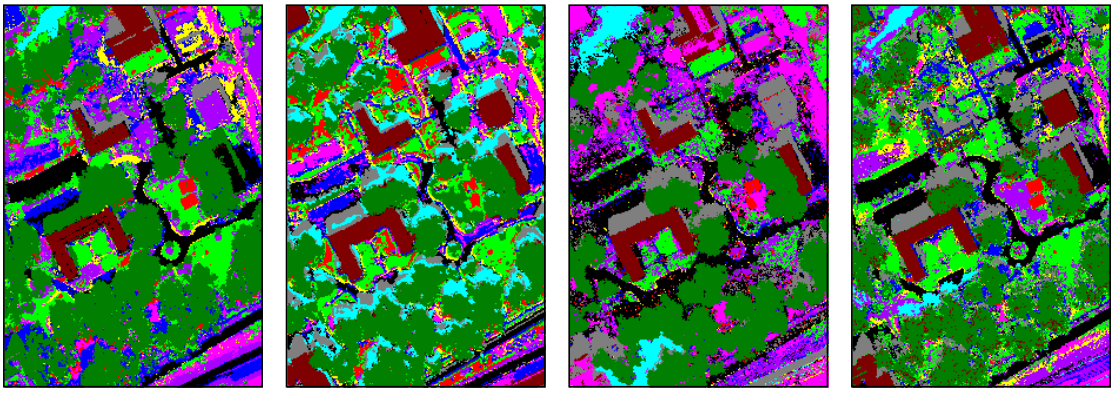
[96] M. Dalla-Mura, J. Atli-Benediktsson, B. Waske, and L. Bruzzone, "Extended profiles with morphological attribute filters for the analysis of hyperspectral data," *Int. J. Remote Sens.*, vol. 31, no. 22, pp. 5975–5991, Dec. 2010.

[97] J. Xia, N. Yokoya, and A. Iwasaki, "Fusion of hyperspectral and lidar data with a novel ensemble classifier," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 6, pp. 957–961, 2018.

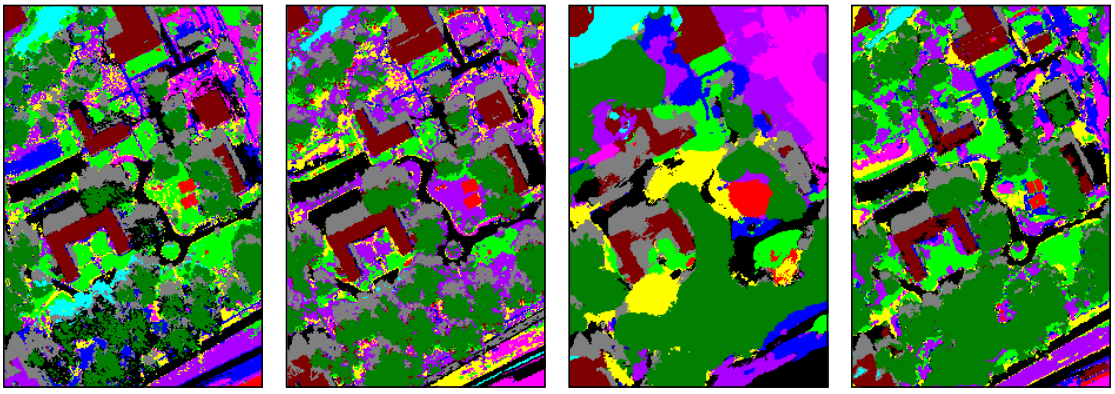
[98] B. Rasti, D. Hong, R. Hang, P. Ghamisi, X. Kang, J. Chanussot, and J. A. Benediktsson, "Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox," *IEEE Geosci. Remote Sens. Mag.*, vol. 8, no. 4, pp. 60–88, 2020.



(a) Ground truth (b) Raw-H (59.12%) (c) Raw (62.34%) (d) 3D-CNN (69.23%)



(e) miniGCN (69.01%) (f) SAE-LR (60.28%) (g) NMFL (66.75%) (h) EPCA (63.75%)



(i) GGF (62.51%) (j) EMAP (68.70%) (k) OTVCA (64.45%) (l) ShearSAF (72.98%)

Figure 19: MUUFL Gulfport dataset: a) Ground truth map, the rest are classification maps obtained by b) Raw-H, c) Raw, d) 3D-CNN, e) miniGCN, f) SAE-LR, g) NMFL, h) EPCA, i) GGF, j) EMAP, k) OTVCA and l) ShearSAF when the number of training samples is five per class (the percentage in the brackets is the corresponding accuracy).