CHAPTER 4

Hyperspectral Image Classification by Chebyshev Moment Multifractal Profiles

Dm.V. Uchaev, D.V. Uchaev, V.A. Malinnikov

This chapter introduces Chebyshev moment multifractal profiles (CMMPs) used for hyperspectral image (HSI) classification. It is shown that CMMPs are effective texture descriptors and can be obtained through Chebyshev moments (CMs) for Hölder exponent images. It is also demonstrated that HSI classification based on the use of the proposed CMMPs can include the following steps: extracting the first few principal components (PCs) of HSIs, calculation of CMMPs from the extracted PCs, stacking PC-values and CMMPs into high-dimensional feature vectors, classification of the formed feature vectors by some supervised classifier (e.g., support vector machine (SVM) classifier). In the last part of the book chapter, results of spectral-spatial classification of two test HSIs using the proposed CMMP features are presented and discussed. It has been experimentally established that HSI classification by spectral features and CMMPs exhibits better performance than competitive methods of

Dm.V. Uchaev, D.V. Uchaev, V.A. Malinnikov
Moscow State University of Geodesy and Cartography (MIIGAiK),
4 Gorokhovsky pereulok, 105064 Moscow
Moscow, Russia
e-mail: d-uchaev@yandex.ru

Editor: G.A. Papakostas, *Recent Progress in Image Moments and Moment Invariants*
DOI: 10.15579/gcsr.vol7.ch4, GCSR Vol. 7, pp. 75-99, 2021

spectral-spatial HSI classification. In particular, it is shown that CMMP-based classification can outperform deep learning methods that are widely used for HSI classification in recent years.

4.1 Introduction

HSI classification is a complex problem that attracts the attention of wide scientific community and can be solved using different image classification methods. Currently, to classify HSIs, methods of spectral-spatial classification are usually used, where various spatial features are used in addition to spectral features to improve the accuracy of HSI classification [15, 12]. This is explained by that the integration of spectral and spatial features can substantially improve the discrimination power of classifiers for complex scenes.

Image processing using moments and moment invariants is about five decades old [37]. As noted in [37], the main advantages of image moments and their invariants are the ability to describe uniquely pixel intensity distributions and the robustness to geometrical and non-geometrical transformations of the image's content. The application of image moments and moment invariants to extract spatial (texture) features from images of natural scenes have been considered in many publications [26, 38, 30, 13, 39]. However, these publications do not address the combined use of spectral and moment features for pattern recognition. Some years ago, a few papers were published which demonstrate that moment features can be used in addition to HSI spectral components for spectral-spatial HSI classification. Papers [21, 22] demonstrate that geometric moment (GM) features can be combined with spectral features for classification of HSIs. As follows from the papers, to obtain GM features for HSIs, GMs of orders up to 2 (or up to 3) are computed for first PCs of HSIs within small $w \times w$ windows centered at each image point. After that, absolute deviation of the GMs from their mean value are calculated to map moment image $M_{p,q}$ to the texture feature image $F_{p,q}$ using the following transformation: $F_{p,q}(i,j) = \sum_{r=1}^{w} \sum_{c=1}^{w} |M_{p,q}(r,c) - \bar{M}_{p,q}|$, where $(r,c) \in W_{i,j}$, $W_{i,j}$ is a $w \times w$ window centered in (i,j), and $\bar{M}_{p,q}$ is the mean of $M_{p,q}$. Thus, first-order GMs produce three texture images ($F_{0,0}, F_{0,1}$ and $F_{1,0}$). Similarly, second- and third-order GMs generate six ($F_{0,0}, F_{0,1}, F_{1,0}, F_{1,1}, F_{0,2}$ and $F_{2,0}$) and ten ($F_{0,0}, F_{0,1}, F_{1,0}, F_{1,1}, F_{0,2}, F_{2,0}$, $F_{1,2}, F_{2,1}, F_{0,3}$ and $F_{3,0}$) texture images, respectively. At the finished stage, the texture features are stacked with spectral features and the stacked spectral-spatial features are used for classification. In [24], Mahi et al. have been demonstrated that Zernike moments (ZMs) can be used to describe objects extracted from multispectral images (MIs). In their approach, MIs are first segmented using mean shift clustering, and after that a few ZMs are extracted from each segment. The obtained results have shown good capabilities of ZMs for detection of buildings on MIs. In [19], third order ZMs are extracted from neighborhoods of pixels in hyperspectral data. These features are extracted from all bands of a HSI and are fed into the SVM classifier used for HSI classification. Paper [28] introduces a method for extracting moment features from high-resolution HSIs. This method is based on extracting two-dimensional moments from neighborhoods of pixels. Three types of image moments were considered in [28]: GMs, complex ZMs and Legendre

moments (LMs). The moments are extracted from a few PCs of HSI, and then they are stacked with the original HSI to form joint spatial-spectral feature vectors that are classified by SVM classifier.

Thus, as follows from the papers [22, 21, 24, 19, 28], moment features have a great potential for spectral-spatial classification of HSIs. However, geometric and orthogonal moments generate the large number of features (as demonstrated in [28], the number of moment features can be more than 600). It is known [47] that too many features can lead to the overfitting problem because the model generated during learning becomes complex. Moreover, classification methods with a large set of features are computationally expensive [20]. All of the above-mentioned allow us to conclude that features generated by geometrical and orthogonal moments of spectral components of HSIs are of limited use for the HSI classification.

This chapter introduces CMMPs that can be used for spectral-spatial classification of HSIs. These profiles are combinations of multifractal features (MFs) generated by CMs of Hölder exponent images obtained for a few PCs of HSIs [46]. To construct the multifractal profiles, we use discrete CMs, because these moments have many useful properties: no numerical approximation is needed for evaluating CMs of images; Chebyshev polynomials (CPs) used to construct CMs can be calculated by simple recurrence formulas; the lower order CMs capture low-frequency variations of image intensities, but the higher order CMs capture high-frequency variations.

The book chapter is organized as follows. Section 4.2 gives a general description of CMs of Hölder exponent images used to construct CMMPs. The HSI classification method based on the use of CMMPs is introduced in Section 4.3. Section 4.4 exhibits an extensive experimental study of the proposed CMMP method. Section 4.5 summarizes the main concepts presented in this chapter and also outlines future work.

4.2 CMs of Hölder Exponent Images

In this section, in order to characterize and distinguish textures of HSIs, spatial features are introduced based on the use of CMs for Hölder exponent images. Hölder exponent images are formed by Hölder exponents corresponding to each image point. Hölder exponents measure the degree of local irregularity for an image function and can be calculated by a multifractal measure generated for the image. As shown in [41], if the measure μ formed for the image I is multifractal, then the following fundamental scaling relation must be satisfied for any $(x, y) \in I$ and sufficiently small ε:

$$\mu_\varepsilon(x, y) \sim \varepsilon^{\alpha(x,y)}, \tag{4.1}$$

where $\mu_\varepsilon(x, y)$ is the measure in a box of size ε centered in (x, y); $\alpha(x, y)$ is a Hölder exponent (local dimension) in (x, y). It can be derived from Eq.(4.1) that $\alpha(x, y)$ for each $(x, y) \in I$ is defined as

$$\alpha(x, y) = \lim_{\varepsilon \to 0} \frac{\ln \mu_\varepsilon(x, y)}{\ln \varepsilon}. \tag{4.2}$$

Thus, having a multifractal measure for an image I of size $M \times N$, we can estimate Hölder exponents $\alpha(x, y)$ in all $(x, y) \in I$ and calculate the following moments for all

$p \in \{0,1,...,M-1\}$ and $q \in \{0,1,...,N-1\}$:

$$K_{p,q} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \Phi_{p,q}(x,y)\alpha(x,y), \tag{4.3}$$

where $\Phi_{p,q}(x,y)$ are Chebyshev kernels (CKs).

Orthogonal moments of Hölder exponent images were first considered in [44], where they are associated with two-dimensional multifractal spectra of the images. As shown in [44], in contrast to one-dimensional multifractal spectra (e.g., singularity spectrum, Rényi spectrum [41]), two-dimensional spectra allow to restore Hölder exponents in each image point.

In the book chapter, we will assume that the CKs in Eq.(4.3) are defined as

$$\Phi_{p,q}(x,y) = \tilde{t}_p(x)\tilde{t}_q(y), \tag{4.4}$$

where $\tilde{t}_p(x)$ and $\tilde{t}_q(y)$ are orthonormal discrete CPs obtained by normalization of discrete CPs defined as [31]

$$t_p(x) = p! \sum_{k=0}^{p} (-1)^{p-k} \left(\begin{array}{c} M-1-k \\ p-k \end{array} \right) \left(\begin{array}{c} p+k \\ p \end{array} \right) \left(\begin{array}{c} x \\ k \end{array} \right), \tag{4.5}$$

where $p, x = 0,1,2,...,M-1$. It is assumed that different techniques for normalization of discrete CPs can be used to obtain $\tilde{t}_p(x)$ and $\tilde{t}_q(y)$ (see [29]).

CKs defined by Eq.(4.4) have many properties that are useful for pattern recognition. The most significant properties of the kernels are the following [26, 45]: a) $s = p+q$-order CKs are rotated versions of the same sampling grid, capturing the same pattern of variation in the image at different orientations (the orientation of these kernels (grids) is given by the arrangement of positive and negative pulses); b) grouping s-order CKs results in a specific frequency band; c) combination of CKs from each frequency band has the effect of a bandpass filter. The peak value of this filter moves from the origin ($s = 0$) toward the maximum frequency ($s = 2N-2$) (the whole range of s-values defines a filter bank that entirely covers the frequency plane); d) the lower order CKs capture low-frequency variations of image intensities, but the higher order CKs capture high-frequency variations.

MFs characterizing average values of Hölder exponents

Zero-order $K_{p,q}$ is of particular importance. One can show that there is a relationship between $K_{0,0}$ and the average value of Hölder exponents $\alpha(x,y)$ denoted as $\bar{\alpha}$. In particular, if $\tilde{t}_0(x) = 1/\sqrt{M}$ and $\tilde{t}_0(y) = 1/\sqrt{N}$ then

$$K_{0,0} = \sqrt{MN}\,\bar{\alpha}. \tag{4.6}$$

Indeed, taking into account that $K_{p,q}$ are defined by Eq.(4.3) and $\Phi_{p,q}(x,y)$ are defined by Eq.(4.4), we obtain that

$$K_{0,0} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \tilde{t}_0(x)\tilde{t}_0(y)\,\alpha(x,y). \tag{4.7}$$

Using the above-mentioned $\tilde{t}_0(x)$ and $\tilde{t}_0(y)$, we arrive to Eq.(4.6):

$$K_{0,0} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \frac{1}{\sqrt{MN}} \alpha(x,y) = \sqrt{MN}\bar{\alpha}. \tag{4.8}$$

Therefore, we can say that $K_{0,0}$ provides information about the average value of Hölder exponents.

MFs characterizing the similarity of Hölder exponent images with CKs

In order to provide information about the similarity of $\alpha(x,y)$ with each of CKs, magnitudes of $K_{p,q}$ can be used: magnitudes of $K_{p,q}$ will be higher for those $\alpha(x,y)$-images that oscillate at a similar rate to $\Phi_{p,q}(x,y)$ along both directions. Thus, a description of texture properties of $\alpha(x,y)$-images can be obtained by assessing the dependence of the moment magnitude on the order s, which is related to the frequency content of CKs. Magnitudes of $K_{p,q}$ describing texture properties of $\alpha(x,y)$-images also characterize textures of images described by $\alpha(x,y)$-values. In this regard, to characterize texture properties of images with size of $N \times N$, the Chebyshev multifractal signature $A(s/s_{max})$ can be used, which is defined through a normalized sum of magnitudes for $K_{p,q}$ as [45]

$$A\left(\frac{s}{s_{\max}}\right) = \frac{1}{s_{\max}} \sum_{p+q=s} |K_{p,q}|, \tag{4.9}$$

where $s = 1, 2, ..., s_{max}$ are orders of CKs, $s_{max} = 2N - 2$ is a maximal order of CKs (it is assumed that $A(s/s_{max})$ is not computed for $s = 0$, because term $A(0)$ usually gives much larger values than other terms).

The multifractal signature $A(s/s_{max})$ has many properties useful for pattern recognition. Since a combination of CKs from each frequency band has the effect of a bandpass filter, $A(s/s_{max})$ quantify responses $\alpha(x,y)$ to each of bandpass filters. Thus, values of $A(s/s_{max})$ determine the occurrence of the pattern given by $s = p + q$ in $\alpha(x,y)$-images. The directions at which patterns are observed depend on p and q. If $p < q$, patterns are captured that are oriented at directions in the range $(-\pi/4; \pi/4) \cup (3\pi/4; 5\pi/4)$ rad. For $p > q$, directions of observed patterns are $(\pi/4; 3\pi/4) \cup (5\pi/4; 7\pi/4)$ rad. Finally, in the case when $p = q$, one can capture patterns oriented at $(2k-1)\pi/4$ rad, where k is any integer.

The number of s-order CKs that contribute to $A(s/s_{max})$ is such that one can capture the fluctuating pattern specified by s at any orientation in $\alpha(x,y)$-images: if $1 \le s \le N-1$ then the number of kernels is equal to $s+1$, if $N \le s \le 2N-2$ then the number of kernels is defined as $2N-1-s$. This follows from the interpretation of $A(s/s_{max})$ in the frequency domain. When s increases from 1 to $N-1$, more kernels are needed to capture all possible orientations at which the target pattern may appear (i.e., to cover the frequency plane region that corresponds to s, additional kernels are necessary). If $s \ge N$, the target pattern cannot appear at all possible orientations in $\alpha(x,y)$-images. Therefore, in order to capture its occurrence, fewer kernels are required.

It can also be noted that $A(s/s_{max})$ provide a noticeable dimensional reduction over $K_{p,q}$: for images of size $N \times N$, $A(s/s_{max})$ are vectors of $2N-2$ elements, while $K_{p,q}$ are sets consisting of N^2 coefficients. Moreover, as proved in [45], $A(s/s_{max})$ are invariant to image scaling, and low order $A(s/s_{max})$ are insensitive to high-frequency additive noise of $\alpha(x,y)$-images.

To take into account all values of the multifractal signature $A(s/s_{max})$, the sum of signature values can be used. This sum, denoted further as S_A, coincides with the area of the figure under the signature plot.

The next section of the book chapter demonstrates that $K_{0,0}$-values and S_A-values can be used to form spatial feature vectors that can be used in addition to spectral features for HSI classification.

4.3 Spectral-spatial Classification of HSIs by CMMPs

This section presents multifractal profiles that can be used as spatial features for supervised classification of HSIs. To generate the multifractal profiles, we propose to use MFs based on calculation of CMs of Hölder exponents and denoted as Chebyshev moment multifractal features (CMMFs). Hereinafter, multifractal profiles containing CMMFs will be called CMMPs. Each CMMP is generated by concatenating different CMMFs, calculated for windows with different sizes using window functions of different types (as noted in [42], ideal window functions should be positive, with weights decreasing away from the center, refinable, isotropic in different dimensions and separable).

In this work, we will use the following notation for CMMFs corresponding to any (x,y) from an image: $\mathrm{CMMF}_F^{t,w}(x,y)$, where w is a linear size of an image window centered in (x,y) and used to calculate the CMMF, t is a type of a window function used to calculate Hölder exponents (in this study, only Haar and Gaussian window functions are considered: t corresponds to h and g for Haar and Gaussian functions, respectively), F is a feature type (in this study, it is assumed that $F \in \{K, S\}$, where K corresponds to $K_{0,0}$-features, but S corresponds to S_A-features). In particular, $\mathrm{CMMF}_K^{g,8}(x,y)$ corresponds to the $K_{0,0}$-value calculated for an 8×8 window centered in (x,y) using Gaussian window function. All CMMFs extracted from the ith PC of a HSI and corresponding to $(x,y) \in PC_i$ can be denoted as $\mathrm{CMMP}[PC_i(x,y)]$. Thus, the CMMP for $(x,y) \in$ HSI can be formally defined as

$$CMMP(x,y) = \{\mathrm{CMMP}[PC_1(x,y)], ..., \mathrm{CMMP}[PC_n(x,y)]\}, \tag{4.10}$$

where n is the number of PCs extracted from the HSI.

Figure 4.1 shows the schematic of the proposed CMMP method for HSI classification. As can be seen from Fig.(4.1), the CMMP method includes the following steps: i) extracting the first few PCs of HSIs, ii) calculation of CMMPs from the extracted PCs, iii) stacking PC values and CMMPs into high-dimensional feature vectors, iv) classification of the formed feature vectors by some supervised classifier (e.g., SVM classifier).

It is known that HSIs usually consist of hundreds of spectral bands, which include a large amount of redundant spectral information. In this regard, at the first step of

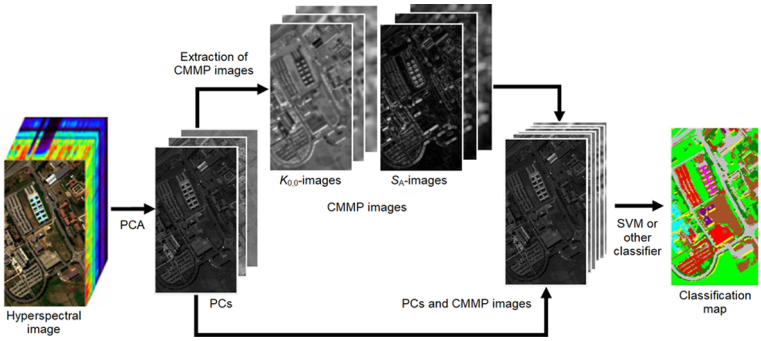


Figure 4.1: Schematic of the proposed CMMP method.

the classification by the CMMP method, PCs of HSIs are extracted using principal component analysis (PCA) [40]. PCA is a widely used linear transformation method providing data dimension reduction. The main concept of PCA is a preservation of most spectral features of HSIs in the PCs of fewer dimensions by a linear transformation, which aims at searching the maximal variance of the data in the projected space.

At the next steps of classification by the CMMP method, CMMPs are obtained for all extracted PCs of the HSI. For this, CMMFs are calculated for the PCs. In particular, to compute $\mathrm{CMMF}_K^{t,w}[PC_i(x,y)]$, $K_{0,0}$ should be calculated for a $w \times w$ window centered in $(x,y) \in PC_i$ using a t-type window function. $K_{0,0}$ can be calculated by Eq.(4.6) through the average value of Hölder exponents for PC_i. To compute the Hölder exponent $\alpha(x_0,y_0)$ in $(x_0,y_0) \in PC_i$, first for several boxes $B_\varepsilon(x_0,y_0)$ of sizes $\varepsilon \times \varepsilon$ centered in $(x_0,\, y_0)$, measures $\mu_\varepsilon(x_0,y_0)$ for $B_\varepsilon(x_0,y_0)$ are calculated as

$$\mu_\varepsilon(x_0,y_0) = \sum_{(x,y)\in B_\varepsilon(x_0,y_0)} \omega\left(\frac{x-x_0}{\varepsilon}, \frac{y-y_0}{\varepsilon}\right) PC_i(x,y), \qquad (4.11)$$

where ω is the t-type window function. After that, the Hölder exponent $\alpha(x_0,y_0)$ is estimated as a slope of a regression line approximating the dependence $\ln \mu_\varepsilon(x_0,y_0) - \ln \varepsilon$ for selected values of ε. To obtain $\mathrm{CMMF}_S^{t,w}[PC_i(x,y)]$, S_A-values should be calculated for a $w \times w$ window centered in $(x,y) \in PC_i$ using a t-type window function. It is assumed that i) S_A are obtained from $K_{p,q}$ estimated for all integer values of p and q up to $w-1$, ii) $K_{p,q}$ are calculated by Eq.(4.3).

CMMPs are obtained by combining the corresponding CMMFs, and then they are stacked with PC-values into high-dimensional feature vectors.

At the final step of the CMMP method, it is performed supervised classification of the formed vectors of spectral and spatial (multifractal) features. Since the obtained feature vectors can include many features (usually the number of features is not extremely much), we propose to use SVM or another classifier with penalization capabilities. The SVM is a supervised classifier that performs classification tasks by constructing hyperplanes in a multidimensional space that separates cases of different class labels. It is proved that SVM is robustness to the variation of data dimensions.

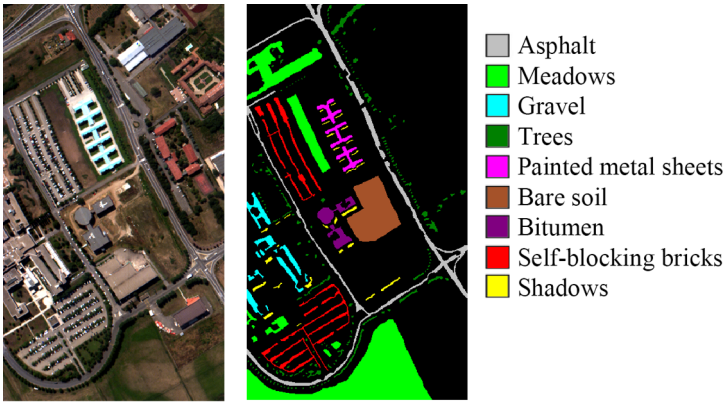


Figure 4.2: Pavia University dataset and corresponding ground truth. (a) Composite image (R-G-B = band 46-27-10); (b) the ground truth image with 9 land-cover classes.

Moreover, it is shown that the SVM classifier has an outstanding performance in the classification of HSIs [27, 43].

The next section of the book chapter exhibits an extensive experimental study of the proposed CMMP method.

4.4 Experimental study

4.4.1 Dataset Description

The experimental analysis was carried out on two HSIs. The first dataset is the Pavia University image, which was acquired by the reflective optics system imaging spectrometer (ROSIS-03) over an urban area surrounding the University of Pavia (Italy). The dataset has 103 spectral bands of size 610×340 pixels. It has a spectral coverage from 0.43 to 0.86 μm and a spatial resolution of 1.3 m. A three-band color composite and the corresponding reference data are presented in Fig.(4.2). As shown in Fig.(4.2b), the reference contains nine classes, most of which are man-made building objects.

The second dataset is the Salinas image, which was acquired by the airborne visible infrared imaging spectrometer (AVIRIS) over the agricultural area of Salinas Valley (California). After discarding 20 water absorption bands, this dataset contains 204 bands of size 512×217, with a spatial resolution of 3.7 m. Figure 4.3 shows the three-band color composite image and the reference of the Salinas dataset. It can be seen that the reference image contains 16 different classes, which represent mostly different types of crops.

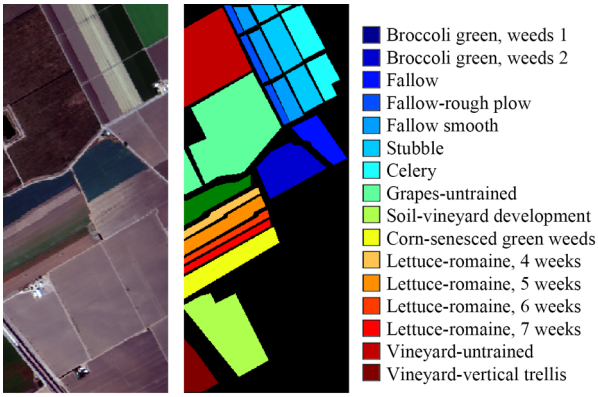


Figure 4.3: Salinas dataset and corresponding ground truth. (a) Composite image (R-G-B = band 35-19-9); (b) the ground truth image with 16 land-cover classes.

4.4.2 Comparison With Other HSI Classification Methods

In this section, the proposed CMMP method is compared with competitive HSI classification methods that use the SVM classifier and the following popular spatial features: locally binary patterns (LBPs) [23], edge-preserving features (EPFs) [17], extended multi-attribute profiles (EMAPs) [32], and GM features [28]. In order to calculate the above-mentioned spatial features, MALTAB-codes were used, taken from [1] (LBP), [5] (EPF) and [33] (EMAP). CMMPs were obtained using own MATLAB application (GMAToolbox) that calculates different multifractal features and classifies HSIs by the features using various classifiers [2].

All spatial features were extracted from PCs of HSIs using PCA. The extracted PCs contain 99% information of the original HSIs (GM features contain 97.5% HSI information). The obtained spatial features were stacked with PC-values into high-dimensional feature vectors, and after that these vectors were classified by SVM classifier. The SVM classification was performed using LIBSVM library [10, 4]. For SVM classification by each of the compared methods, training samples were randomly chosen from the reference data shown in Figs.(4.2)-(4.3). To realize good classification results, radial basis function kernels were used for SVM classification.

Due to the unavailability of codes for GM features, all classification results by these features were taken from papers [28, 25]. As follows from the papers, for HSI classification by GM features, a polynomial kernel $K(\mathbf{x}_1, \mathbf{x}_2) = (\gamma \mathbf{x}_1^{\mathrm{T}} \mathbf{x}_2 + c_0)^d$ was used without tuning its parameters: $\gamma = 1/\text{number of features}$, $c_0 = 0$ and $d = 3$ ($\mathbf{x}_1$ and $\mathbf{x}_2$ are two arbitrary feature vectors, and T denotes vector transpose).

4.4.2.1 LBP

LBP is a powerful local texture descriptor that is constructed by comparing each pixel with its neighboring pixels [23]. The neighboring pixels are from a set of equally spaced

samples over a circle of radius r centered at the center pixel. Radius r determines how far the neighboring pixels can be located away from the center pixel. Along with the selected m neighbors $\{f_i\}_{i=0}^{m-1}$, the LBP code for the center pixel f_c is given by

$$\mathrm{LBP}_{m,r} = \sum_{i=0}^{m-1} U(f_i - f_c)2^i, \tag{4.12}$$

where $U(f_i - f_c) = 1$ if $f_i > f_c$ and $U(f_i - f_c) = 0$ if $f_i \leq f_c$.

As follows from Eq.(4.12), if the gray value of each neighborhood's pixel is more than the gray value of the centered one, this pixel takes the value 1, otherwise it takes the 0 value. Thus, the output of LBP code reflects the texture orientation and smoothness in a local region. The size of the region is a user-defined parameter that affects classification results. After obtaining the LBP code for all pixels, an occurrence histogram is computed over a local patch centered at pixels. The resulted histogram is insensitive to illumination and gray-scale variations, invariant under image translation, computed easily and thus can be used as local descriptor in order to distinguish several image patterns. In order to form the spatial feature vector, all bands of LBP histograms are concatenated.

In our experiments, for LBP classification of HSIs, we used uniform LBPs [35] calculated in windows of size 7×7. A uniform LBP contains at most two 0–1 or 1–0 transitions and effectively reduces the feature vector (the feature vector for each of PCs consists of 59 elements). During the calculation of uniform LBPs, parameters m and r from Eq.(4.12) were assigned values of 8 and 1, respectively.

4.4.2.2 EPF

EPFs were extracted using image fusion and recursive filtering (IFRF) technique [17]. The recursive filter can effectively remove the texture information and can also preserve the strong edge structures. For the IFRF-based feature extraction technique, the parameters δ_s and δ_r of the recursive filter need to be determined. Parameters δ_s and δ_r are spatial and range parameters of EPFs, which determine the degree of smoothness: when δ_r is a relatively large, i.e., $\delta_r = 2$, the filtering output will tend to be extremely smooth, and only little useful information is then preserved; when δ_s tends to infinity, e.g., $\delta_s = 1000$, the recursive filter will not produce unbounded smoothing of the image.

In our experiments, we assumed that $\delta_s = 200$ and $\delta_r = 0.3$. As shown in [18], the corresponding values for δ_s and δ_r can give satisfactory classification results.

4.4.2.3 EMAP

EMAPs are obtained when multiple so-called extended attribute profiles (EAPs) are computed considering different attributes and stacked together in the same data structure [32]. EAPs are analogous to well-known extended morphological profiles (EMP) often used for HSI classification [7]. They are obtained by a concatenation of attributive profiles (APs) computed on each feature extracted from a HSI. The main advantage of using EAPs instead of EMPs is the great flexibility that comes with defining attributes used for modelling spatial features that need to be extracted [32].

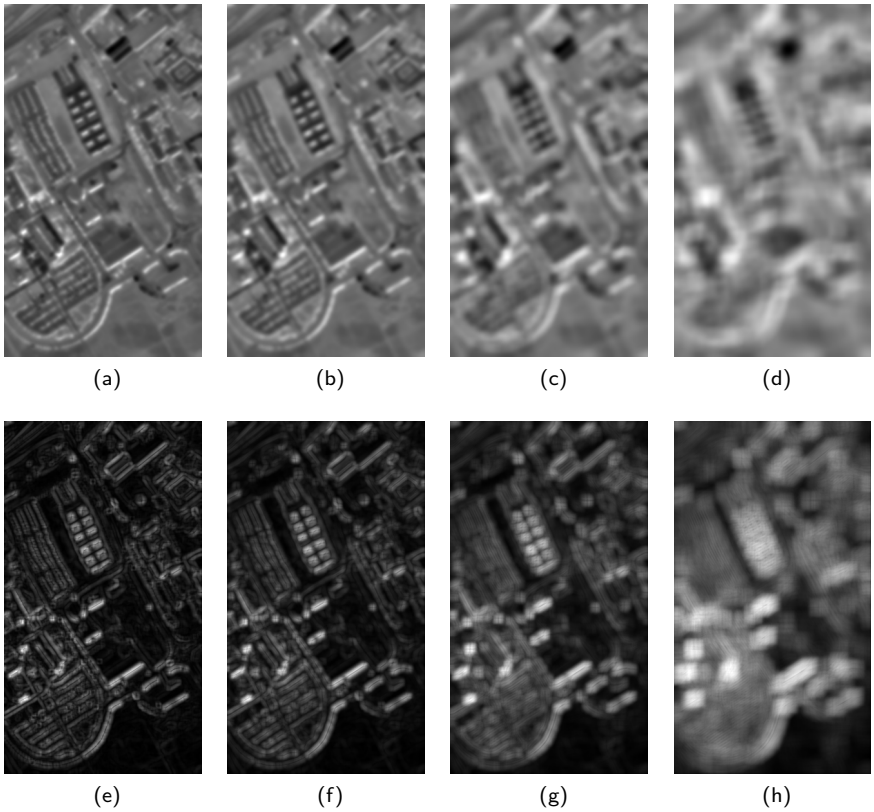
In our experiments, four EAPs were obtained by considering four different attributes for PCs extracted by PCA: i) EAPa, area of image regions; ii) EAPd, length of the diagonal of the box bounding each region (it is a measure of a linear size of regions); iii) EAPi, first moment invariant of Hu, or moment of inertia [16] (it measures the elongation of regions); iv) EAPs, standard deviation of gray-level values of pixels belonging to each region (it measures the homogeneity of intensity values of pixels and gives information related to the spectral contrast of pixels). Four thresholds $\lambda_a, \lambda_d, \lambda_i$ and λ_s were used for constructing each of four EAPs. Each EAP is 36-dimensional, because it is composed of four APs with 9 levels computed on each of PCs. The thresholding values were chosen as those used in [34]: $\lambda_a = \left[\begin{array}{cccc} 100 & 500 & 1000 & 5000 \end{array}\right]$ for EAPa; $\lambda_d = \left[\begin{array}{cccc} 10 & 25 & 50 & 100 \end{array}\right]$ for EAPd; $\lambda_i = \left[\begin{array}{cccc} 0.2 & 0.3 & 0.4 & 0.5 \end{array}\right]$ for EAPi; $\lambda_s = \left[\begin{array}{cccc} 20 & 30 & 40 & 50 \end{array}\right]$ for EAPs.

4.4.2.4 CMMP

In all our experiments, different sets of CMMPs were used to classify the Pavia University and Salinas datasets. In particular, for the Pavia University dataset, CMMF_K^{t,w_1} and CMMF_S^{t,w_2} were extracted from each of four PCs, where $t \in (h, g)$, $w_1 = 4, 8, 16, 32, 64, 128$ and $w_2 = 4, 8, 16, 32$. For the Salinas dataset, only $\mathrm{CMMF}_K^{h,w}$ were extracted, where $w = 4, 8, 16, 32, 64, 128$. Thus, each CMMP for the Pavia University dataset is an 80-dimensional vector, because it is composed of 12 $K_{0,0}$-values and 8 S_A-values computed on each of four PCs. Similarly, each CMMP for the Salinas dataset is a 24-dimensional vector (it is composed of 6 $K_{0,0}$-values computed on each of four PCs). As an example, Fig.(4.4) shows CMMP images for the first PC extracted from the Pavia University dataset.

To compare the CMMP method with all above-mentioned methods, four experiments were performed. In the first experiment, we have evaluated the performance of all compared methods using the following accuracy estimates [14]: class accuracy (CA), overall accuracy (OA), average accuracy (AA), and Kappa coefficient. Here, CA is the percentage of correctly classified pixels for each class, which refers to the user's accuracy; OA is the percentage of correctly classified pixels; AA is the mean of the percentage of correctly classified pixels for each class, the Kappa coefficient gives the percentage of correctly classified pixels corrected by the number of agreements that would be expected purely by chance. In order to obtain all accuracy estimates, 20 samples for each class were randomly chosen from the reference data as training samples and remaining samples were used as the test set (see Tables 4.1-4.2).

For each method (excepting the GM method), classification results were obtained 30 times (for the GM method, 10 experiments were performed), and then means and standard deviations for classification accuracies were estimated (Tables 4.3-4.4). As can be seen from Tables 4.3-4.4, results provided by the proposed CMMP method are better than those obtained by compared classification methods in terms of OA and Kappa coefficients. Moreover, we can see that CAs providing by CMMPs are highest for most of the test dataset classes. In particular, highest CAs are obtained for such classes of the Pavia University dataset as meadows, gravel, trees, painted metal sheets and bare soil. In general, we can say that for training samples with 20 reference pixels, the CMMP method achieves an 1–15% advantage over other methods in terms of



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 4.4: CMMP images for the first PC extracted from the Pavia University dataset: the first row shows $K_{0,0}$-images obtained using windows with sizes of (a) 4×4, (b) 8×8, (c) 16×16, (d) 32×32; the second row shows S_A-images obtained using windows with sizes of (e) 4×4, (f) 8×8, (g) 16×16, (h) 32×32.

Table 4.1: Train-test distribution of samples for the Pavia University dataset.

#	Class	Train	Test
1	Asphalt	20	6621
2	Meadows	20	18639
3	Gravel	20	2089
4	Trees	20	3054
5	Painted metal sheets	20	1335
6	Bare soil	20	5019
7	Bitumen	20	1320
8	Self-blocking bricks	20	3672
9	Shadows	20	937
Total		180	42686

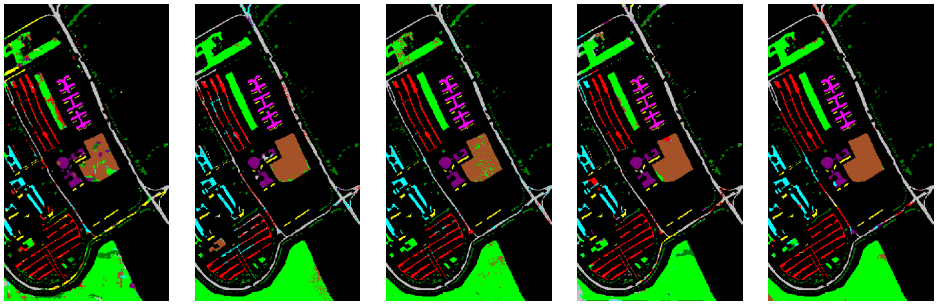
OA. In particular, we can see from Table 4.3 that for the Pavia University dataset, the CMMP method outperforms the LBP method by 15% in OA.

Figures 4.5-4.6 demonstrate classification maps obtained on the Pavia University and Salinas datasets using LBP, IFRF, EMAP, GM and CMMP methods. As can be seen form the figures, the LBP method can lead to oversmoothed classification maps. The reason is the utilization of a slack variable, which can reduce the influence of both noise and spectral changeability but may lead to oversmoothing when the number of training samples is quite limited. The classification results obtained by the EMAP and GM methods are not very satisfactory since some noise-like mislabeled pixels are still visible. The IFRF method is much better in removing "noisy pixels", but it cannot distinguish well between certain classes (for example, asphalt and bricks). In contrast, the CMMP method is good for separating all classes and doesn't produce oversmoothed classification maps.

During the second experiment, the influence of the number of training samples to the performance of different methods was analyzed. Figures 4.7-4.8 show the OA, AA, and Kappa coefficient of different methods as functions of the number of training samples (ranging from 5 to 30). For all compared methods, excepting the GM method, accuracy values were obtained by averaging accuracy estimates, obtained after 30 runs (accuracies for the GM method were obtained by averaging results of 10 runs). As can be seen from Figs.(4.7-4.8), an increase in the training size has a favorable effect on the performance of all classification methods. However, the CMMP method always obtains the highest values of means for OA and Kappa. Moreover, compared with the LBP and GM method, the proposed CMMP method shows obvious improvements, especially when the number of training samples is quite limited. In particular, as follows from Fig.(4.7), for 5 training samples, the proposed CMMP method outperforms the LBP method by 25% in OA for the Pavia University dataset.

Table 4.2: Train-test distribution of samples for the Salinas dataset.

#	Class	Train	Test
1	Broccoli green, weeds 1	20	1989
2	Broccoli green, weeds 2	20	3706
3	Fallow	20	1956
4	Fallow-rough plow	20	1374
5	Fallow smooth	20	2658
6	Stubble	20	3939
7	Celery	20	3559
8	Grapes-untrained	20	11251
9	Soil-vineyard development	20	6183
10	Corn-senesced green weeds	20	3258
11	Lettuce-romaine, 4 weeks	20	1048
12	Lettuce-romaine, 5 weeks	20	1907
13	Lettuce-romaine, 6 weeks	20	896
14	Lettuce-romaine, 7 weeks	20	1050
15	Vineyard-untrained	20	7248
16	Vineyard-vertical trellis	20	1787
Total		320	53809

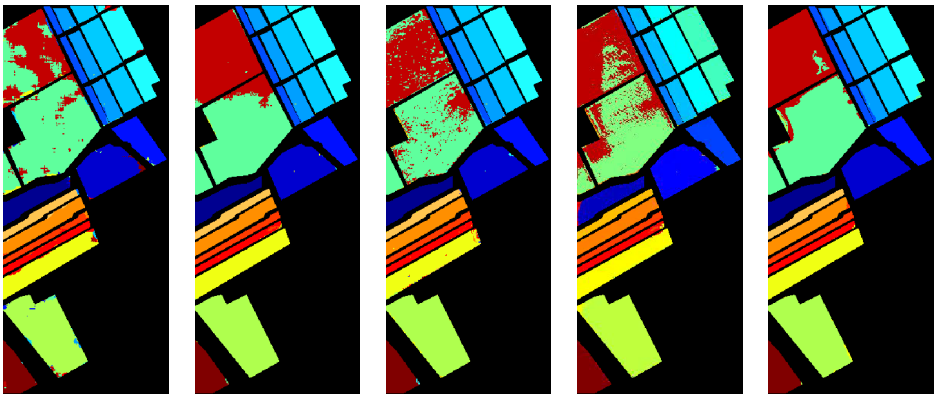


(a) (b) (c) (d) (e)

Figure 4.5: Classification maps obtained by the compared methods on the Pavia University dataset: (a) LBP, (b) IFRF, (c) EMAP, (d) GM, (e) CMMP. The number of training samples is set to 20 pixels of the reference.

Table 4.3: Classification accuracies of the compared methods on the Pavia University dataset. The number of training samples is set to 20 pixels of the reference. In each row, the best result is shown in boldface.

Class	LBP	IFRF	EMAP	CMMP
Asphalt	89.02±6.11	91.00±4.04	**98.79±0.69**	96.97±1.71
Meadows	92.66±2.25	97.49±1.39	98.15±1.12	**99.36±0.58**
Gravel	68.88±6.79	72.25±6.93	79.58±7.69	**85.53±5.22**
Trees	66.84±7.00	91.82±6.74	90.10±8.28	**93.04±4.28**
Painted metal sheets	98.01±1.94	99.46±2.30	99.65±0.53	**99.85±0.19**
Bare soil	64.30±9.27	81.97±8.92	80.13±9.50	**88.46±7.63**
Bitumen	76.89±9.51	71.69±9.38	**99.43±0.69**	88.66±5.99
Self-blocking bricks	74.62±6.65	81.43±5.11	**92.28±3.90**	88.05±3.35
Shadows	46.95±5.69	94.65±3.94	**99.01±1.11**	94.53±3.47
OA(%)	79.23±2.21	89.70±2.41	93.23±2.12	**94.69±1.40**
AA(%)	**93.43±0.75**	86.86±2.29	93.02±1.83	92.72±1.43
k × 100(%)	90.48±0.95	86.57±3.04	91.17±2.68	**93.03±1.80**



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

Figure 4.6: Classification maps obtained by the compared methods on the Salinas dataset: (a) LBP, (b) IFRF, (c) EMAP, (d) GM, (e) CMMP. The number of training samples is set to 20 pixels of the reference.

Table 4.4: Classification accuracies of the compared methods on the Salinas dataset. The number of training samples is set to 20 pixels of the reference. In each row, the best result is shown in boldface.

Class	LBP	IFRF	EMAP	CMMP
Broccoli green, weeds 1	99.93±0.13	99.86±0.34	**100.00±0.00**	99.35±1.44
Broccoli green, weeds 2	98.29±1.76	**99.97±0.04**	99.93±0.07	99.45±1.51
Fallow	93.12±5.09	**99.83±0.07**	97.71±1.18	98.20±2.02
Fallow-rough plow	92.96±2.71	**97.95±0.45**	97.55±0.71	96.75±2.21
Fallow smooth	96.97±2.31	**99.93±0.10**	99.24±0.43	99.46±0.73
Stubble	98.25±1.43	99.97±0.15	99.95±0.21	**99.99±0.01**
Celery	94.10±4.19	99.69±0.40	**99.90±0.31**	99.67±0.71
Grapes-untrained	88.90±3.51	**97.56±2.05**	91.08±3.68	97.11±1.84
Soil-vineyard development	97.89±1.34	**99.87±0.12**	99.40±0.32	99.63±0.39
Corn-senesced green weeds	97.83±2.06	99.49±0.47	95.65±2.55	**99.54±0.35**
Lettuce-romaine, 4 weeks	98.24±2.20	**99.85±0.11**	94.09±2.41	94.15±4.64
Lettuce-romaine, 5 weeks	98.19±1.20	99.62±0.79	97.32±1.26	**99.74±0.52**
Lettuce-romaine, 6 week	91.30±5.17	**98.55±2.10**	96.63±2.25	97.52±2.08
Lettuce-romaine, 7 weeks	84.43±6.56	**98.45±1.29**	97.80±1.77	97.63±2.09
Vineyard-untrained	75.79±4.14	87.96±4.80	75.98±5.97	**93.66±3.87**
Vineyard-vertical trellis	88.70±6.19	98.33±2.38	98.36±1.47	**100.00±0.02**
OA(%)	91.43±0.86	97.41±0.81	93.52±1.11	**97.94±0.61**
AA(%)	93.43±0.75	**98.55±0.31**	96.29±0.45	98.24±0.48
k × 100(%)	90.48±0.95	97.12±0.89	92.79±1.22	**97.70±0.68**

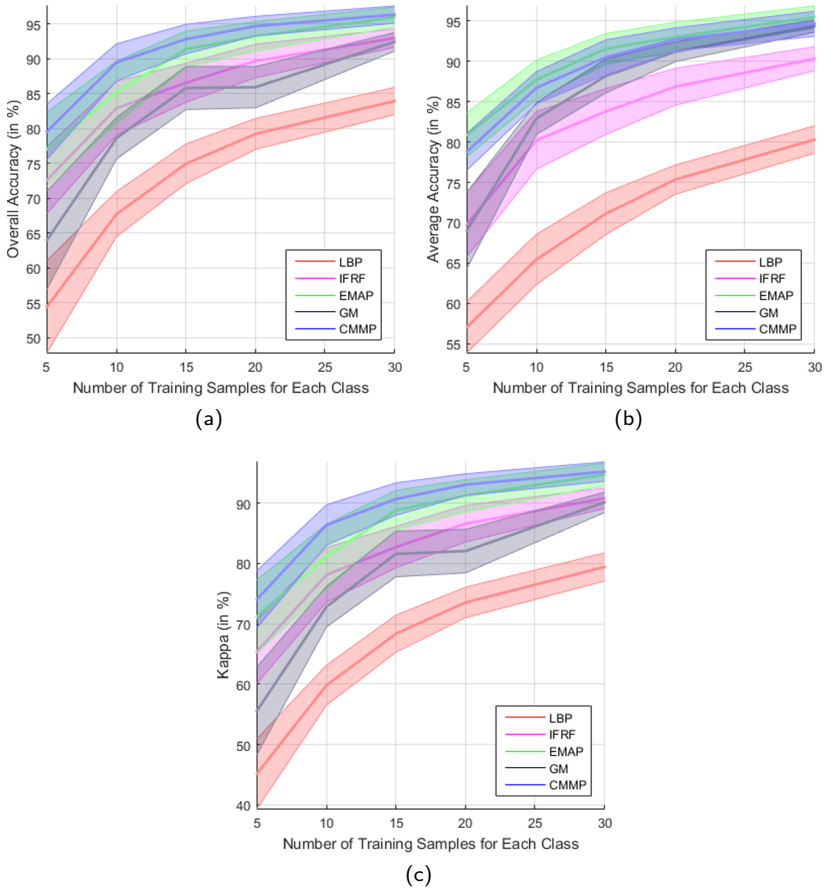


Figure 4.7: Performance of the compared methods on the Pavia University dataset as functions of the number of labeled samples per class (varying from 5 to 30): (a) OA, (b) AA, (c) Kappa coefficient. Shaded areas refer to the standard deviations of accuracies obtained in 30 repeated experiments (for the GM method, standard deviations of accuracies for 10 experiments are shown).

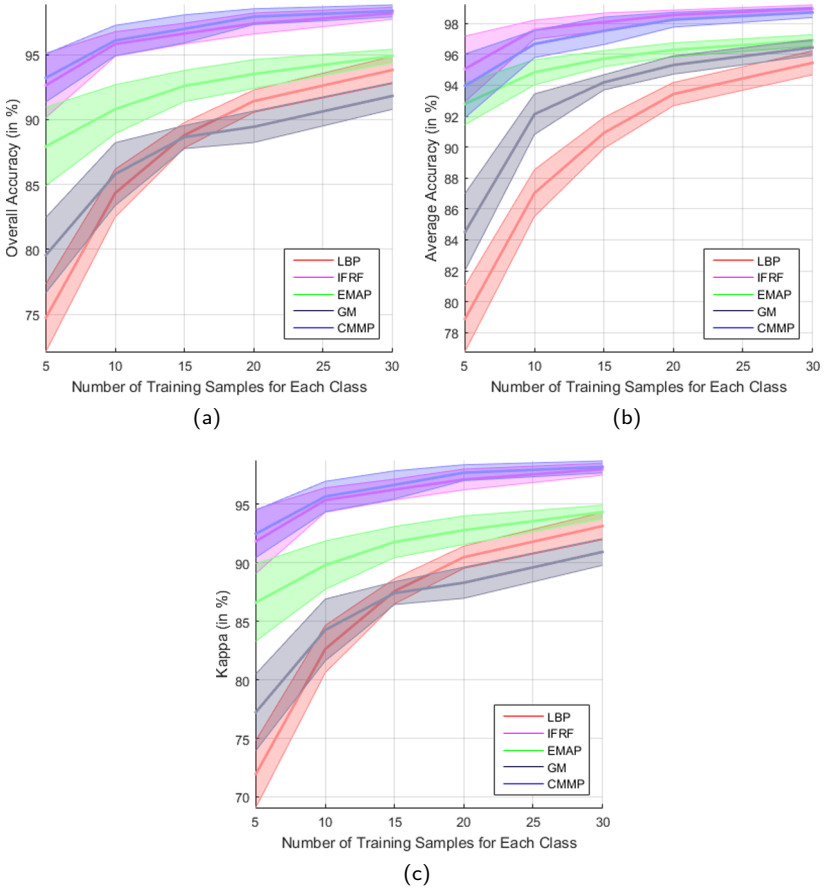


Figure 4.8: Performance of the compared methods on the Salinas dataset as functions of the number of labeled samples per class (varying from 5 to 30): (a) OA, (b) AA, (c) Kappa coefficient. Shaded areas refer to the standard deviations of accuracies obtained in 30 repeated experiments (for the GM method, standard deviations of accuracies for 10 experiments are shown).

Table 4.5: Overall accuracies of the compared methods. In each column, the best result is shown in boldface.

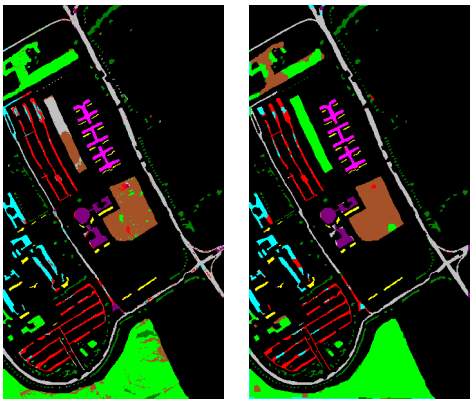
Method	Pavia University		Salinas	
	5%	10%	5%	10%
LBP	99.23±0.15	99.78±0.04	99.83±0.06	99.95±0.02
IFRF	98.94±0.10	99.37±0.12	99.79±0.05	99.87±0.04
EMAP	99.48±0.04	99.64±0.04	99.46±0.08	99.63±0.04
GM	99.46±0.11	99.78±0.04	99.81±0.06	99.94±0.02
ZM	99.60±0.05	99.82±0.03	98.92±0.12	99.84±0.10
LM	99.47±0.10	99.79±0.03	99.65±0.09	99.92±0.03
CMMP	**99.77±0.05**	**99.93±0.04**	**99.86±0.05**	**99.95±0.01**

In the next experiment, 5% and 10% of labeled samples were selected as training samples, and OA estimates were calculated by each of compared methods (Table 4.5). For this experiment, in addition to all above-mentioned methods, LM- and ZM-based methods were used. OA estimates were obtained by averaging results after 10 runs (OA values obtained by GMs, ZMs and LMs were taken from [28]). As can be seen from Table 4.5, all OA estimates are close to each other and they have very small standard deviations. We can say that our method is as good as other methods because it has high values of OA and lowest standard deviations for the Pavia University and Salinas datasets. Moreover, we can see from Table 4.5 that the CMMP method always gives higher values of OA than the EMAP and ZM methods.

In the last experiment, the CMMP method was compared with two deep learning methods. The feasibility of using deep learning methods for HSI classification is explained by the fact that deep features informative for HSIs can be obtained by a convolutional neural network (CNN). However, deep learning methods can perform poorly when the number of training samples is small [11]. In this regard, for comparison with our method, we have chosen two deep learning methods that are adopted for small training samples.

The first of the compared deep learning methods, denoted as CNN-MRF, uses CNN and Markov random field (MRF) to enforce class label smoothness to boost the classification performance. CNN-MRF is an implementation of CNN-AL-MRF [8] without active learning. At the first step of the method, CNN is used to extract spectral-spatial discriminative features from 3-D patches of HSI. At the second step, MRF is used which encourages the neighboring pixels to have the same labels.

The second compared deep learning method, denoted as R-VCANet [36], is based on the use of rolling guidance filter (RGF) and vertex component analysis network (VCANet). VCANet is a modification of PCA network (PCANet) [9]. Similarly to PCANet, VCANet contains four layers: the input, two convolution, and output layers. In VCANet, instead of convolution kernels, the spectra extracted from the HSI by



(a) (b)

Figure 4.9: Classification maps obtained by deep-learning methods on the Pavia University dataset: (a) R-VCANet; (b) CNN-MRF. The number of training samples is set to 20 pixels of the reference.

vertex component analysis (VCA) are used. In R-VCANet, first each band of a HSI is smoothed by RGF to remove the spatial variability and image noise, and after that spectra are extracted from the HSI by VCA. RGF used in R-VCANet is a recently proposed edge-preserving filter that can smooth away small textures while retaining spatial structure information [48]. As shown in [36], R-VCANet is able to achieve a high classification accuracy with much less training samples than traditional deep learning methods.

In order to perform our experiments with deep learning methods, we used codes from [3] (CNN-MRF) and [6] (R-VCANet). For classification by the chosen deep-learning methods, the number of training set was varied from 5 to 30 pixels. Using deep learning methods, classification maps were obtained and classification accuracies were estimated. Figure 4.9 demonstrates classification maps obtained by two compared deep-learning methods when the number of training samples set to 20 pixels of the reference. As can be seen from the figure, the quality of both classification results is lower than the quality of maps obtained by the CMMP method. Compared accuracy plots from Fig.(4.10), we can say that the CMMP method is mostly characterized by higher values of OA, AA and Kappa coefficients. Moreover, in contrast to our method, deep learning methods are time consuming even when the number of training samples is small.

4.5 Summary

In this book chapter, the new CMMP method for HSI classification has been proposed. The proposed method is based on the use of CMs for Hölder exponent images, which are obtained for spectral components of HSIs, and combines spatial information with the spectral components of HSIs. Several experiments were performed to evaluate the

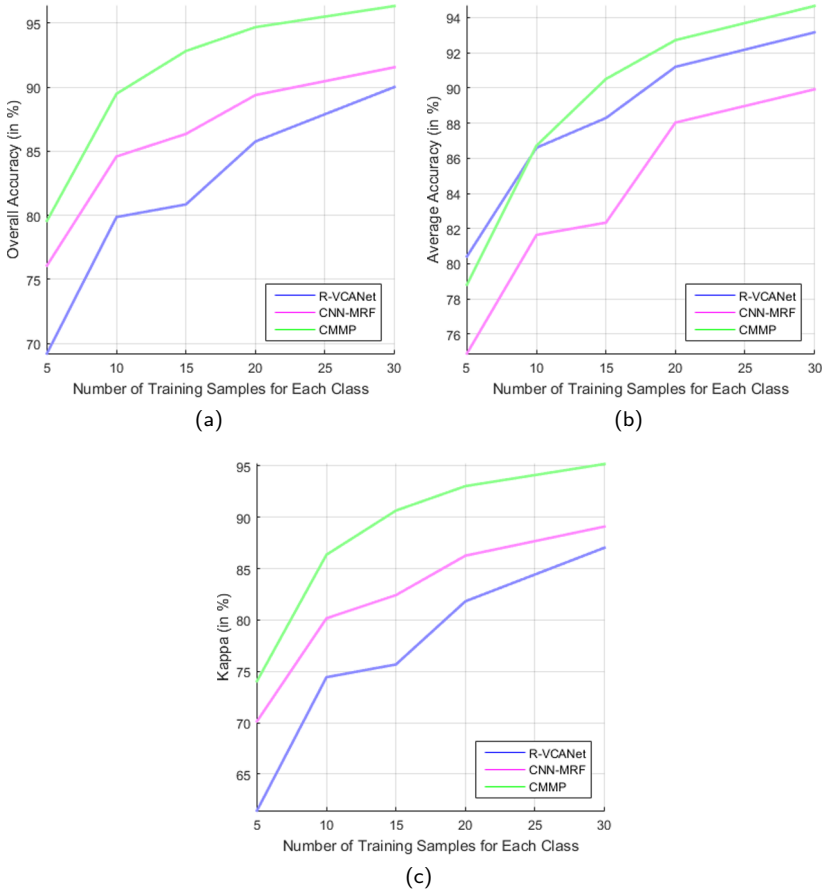


Figure 4.10: Performance of the compared deep learning methods on the Pavia University dataset as functions of the number of labeled samples per class (varying from 5 to 30): (a) OA, (b) AA, (c) Kappa coefficient. Function values are mean values of accuracies obtained in n repeated experiments ($n = 3$ for R-VCANet, $n = 5$ for CNN-MRF, and $n = 30$ for CMMP).

performance of the proposed CMMP method. All experiments have been carried out on two widely used HSIs.

The results of the experiments demonstrate the effectiveness of the proposed CMMP method, which provides better OAs and Kappa statistics than other popular HSI classification methods, including modern deep-learning methods. Moreover, we have been established that the CMMP method has the following advantages: i) for this method, a limited number of labeled samples can be used for training; ii) the number of features used for classification by the CMMPs is not too much; iii) the CMMP method is computationally efficient because it is based on the computation of discrete CMs that can be fastly calculated using recurrence formulas.

Although the classification accuracy provided by the CMMP method depends on the number of features and parameters used to obtain the features, this dependence is not critical. The reason is that there is an optimal set of features for which the CMMP method has satisfactory classification results and outperforms other HSI classification methods in terms of OA and Kappa.

In the future, we plan to conduct a comprehensive study to improve the proposed HSI classification method. In particular, we would like to explore the possibilities of using continuous orthogonal polynomials instead of discrete CPs to generate multifractal profiles. We also plan to replace the SVM classifier by semisupervised classifiers, which are used for small training samples.

Acknowledgement

Research devoted to Chebyshev moment multifractal profiles was carried out within the framework of the state assignment 0708-2020-0001 of the Ministry of Science and Higher Education of the Russian Federation. The study of the proposed HSI classification method has been supported by the Ministry of Education and Science of the Russian Federation (Project No. МК-3477.2019.5). The work devoted to the first and third experiments with HSI classification methods was supported by the Russian Foundation for Basic Research (Project No. 19-05-00330 А). Research devoted to the influence of the number of training samples to the performance of different classification methods (second and fourth experiments) was supported by the Grant for young scientists from MIIGAiK.

References

[1] GitHub - Mr-Yuppie/Indian_elm_lbp_test. https://github.com/Mr-Yuppie/Indian_ELM_LBP_test.

[2] GitHub - UchaevD/GMAToolbox: Software for image processing by generalized multifractal analysis. https://github.com/UchaevD/GMAToolbox.

[3] GitHub - xiangyongcao/CNN-AL-MRF: This is the code of "Hyperspectral Image Classification with Convolutional Neural Network and Active Learning". https://github.com/xiangyongcao/CNN-AL-MRF.

[4] LIBSVM - A Library for Support Vector Machines. https://www.csie.ntu.edu.tw/cjlin/libsvm/.

[5] Xudong Kang's Homepage - Home. http://xudongkang.weebly.com/.
[6] Zhenwei Shi's Homepage. http://levir.buaa.edu.cn/.
[7] J.A. Benediktsson, J.A. Palmason, and J.R. Sveinsson. Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, 43(3):480–491, March 2005.
[8] X. Cao, J. Yao, Z. Xu, and D. Meng. Hyperspectral image classification with convolutional neural network and active learning. *IEEE Transactions on Geoscience and Remote Sensing*, 58(7):4604–4616, July 2020.
[9] H.T. Chan, K. Jia, S. Gao, J. Lu, Z. Zeng, and Y. Ma. PCANet: A simple deep learning baseline for image classification? *IEEE Transactions on Image Processing*, 24(12):5017–5032, December 2015.
[10] C.C. Chang and C.J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, May 2011.
[11] Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2094–2107, June 2014.
[12] N. Falco, J. Xia, X. Kang, S. Li, and J.A. Benediktsson. Supervised classification methods in hyperspectral imaging - recent advances. In Jose Manuel Amigo, editor, *Data Handling in Science and Technology*, volume 32 of *Hyperspectral Imaging*, pages 247–279. Elsevier, January 2020.
[13] J. Flusser, T. Suk, and B. Zitova. *Moments and Moment Invariants in Pattern Recognition*. Wiley, 2009.
[14] G.M. Foody. Status of land cover classification accuracy assessment. *Remote Sensing of Environment*, 80(1):185–201, April 2002.
[15] P. Ghamisi, E. Maggiori, S. Li, R. Souza, Y. Tarablaka, G. Moser, A. De Giorgi, L. Fang, Y. Chen, M. Chi, S. B. Serpico, and J. A. Benediktsson. New frontiers in spectral-spatial hyperspectral image classification: The latest advances based on mathematical morphology, Markov random fields, segmentation, sparse representation, and deep learning. *IEEE Geoscience and Remote Sensing Magazine*, 6(3):10–43, September 2018.
[16] M.K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, February 1962.
[17] X. Kang, S. Li, and J. A. Benediktsson. Feature extraction of hyperspectral images with image fusion and recursive filtering. *IEEE Transactions on Geoscience and Remote Sensing*, 52(6):3742–3752, June 2014.
[18] X. Kang, S. Li, and J. A. Benediktsson. Spectral-spatial hyperspectral image classification with edge-preserving filtering. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5):2666–2677, May 2014.
[19] K. Kavitha, S. Arivazhagan, R. Dhivya Priya, and I. Kanaga Sangeetha. Role of Zernike moments in hyperspectral image classification. *International Journal of Scientific Engineering and Technology*, 2(5):383–387, 2013.
[20] R. Kohavi and G.H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, December 1997.
[21] B. Kumar and O. Dikshit. Texture based hyperspectral image classification. In *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume XL-8, pages 793–798. Copernicus GmbH,

November 2014.

[22] B. Kumar and O. Dikshit. Spectral-spatial classification of hyperspectral imagery based on moment invariants. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(6):2457–2463, June 2015.

[23] W. Li, C. Chen, H. Su, and Q. Du. Local binary patterns and extreme learning machine for hyperspectral imagery classification. *IEEE Transactions on Geoscience and Remote Sensing*, 53(7):3681–3693, July 2015.

[24] H. Mahi, H. Isabaten, and C. Serief. Zernike moments and SVM for shape classification in very high resolution satellite images. *The International Arab Journal of Information Technology*, 11(1):43–51, 2014.

[25] R.N. Marandi and H. Ghassemian. A new feature extraction based on local energy for hyperspectral image. In *2017 Artificial Intelligence and Signal Processing Conference (AISP)*, pages 59–64, October 2017.

[26] J.V. Marcos and G. Cristobal. Texture classification using discrete Tchebichef moments. *JOSA A*, 30(8):1580–1591, August 2013.

[27] F. Melgani and L. Bruzzone. Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8):1778–1790, August 2004.

[28] F. Mirzapour and H. Ghassemian. Moment-based feature extraction from high spatial resolution hyperspectral images. *International Journal of Remote Sensing*, 37(6):1349–1361, March 2016.

[29] R. Mukundan. Some computational aspects of discrete orthonormal moments. *IEEE Transactions on Image Processing*, 13(8):1055–1059, August 2004.

[30] R. Mukundan. Local Tchebichef moments for texture analysis. In G.A. Papakostas, editor, *Moments and Moment Invariants - Theory and Applications*, volume 1 of *Gate to Computer Science and Research*, pages 127–142. Science Gate Publishing, 2014.

[31] R. Mukundan, S. H. Ong, and P.A. Lee. Image analysis by Tchebichef moments. *IEEE Transactions on Image Processing*, 10(9):1357–1364, September 2001.

[32] M. Dalla Mura, J.A. Benediktsson, B. Waske, and L. Bruzzone. Extended profiles with morphological attribute filters for the analysis of hyperspectral data. *International Journal of Remote Sensing*, 31(22):5975–5991, December 2010.

[33] M. Dalla Mura and M. Fauvel. Image analysis of hyperspectral data using mathematical morphology. In *6th Workshop on Hyperstectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS 2014)*, 2014.

[34] M. Dalla Mura, A. Villa, J.A. Benediktsson, J. Chanussot, and L. Bruzzone. Classification of hyperspectral images by using morphological attribute filters and Independent Component Analysis. In *2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS 2010)*, pages 1–4, June 2010.

[35] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, July 2002.

[36] B. Pan, Z. Shi, and X. Xu. R-VCANet: A new deep-learning-based hyperspectral image classification method. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(5):1975–1986, May 2017.

[37] G.A. Papakostas. Over 50 years of image moments and moment invariants. In G.A. Papakostas, editor, *Moments and Moment Invariants - Theory and Applications*, volume 1 of *Gate to Computer Science and Research*, pages 3–32. Science Gate Publishing, 2014.

[38] G.A. Papakostas, E.G. Karakasis, and D.E. Koulouriotis. Novel moment invariants for improved classification performance in computer vision applications. *Pattern Recognition*, 43(1):58–68, January 2010.

[39] G.A. Papakostas, E.G. Karakasis, and D.E. Koulouriotis. Orthogonal image moment invariants: highly discriminative features for pattern recognition applications. In *Cross-Disciplinary Applications of Artificial Intelligence and Pattern Recognition: Advancing Technologies*, pages 34–52. IGI Global, 2013.

[40] S. Prasad and L. M. Bruce. Limitations of principal components analysis for hyperspectral target recognition. *IEEE Geoscience and Remote Sensing Letters*, 5(4):625–629, October 2008.

[41] H. Salat, R. Murcio, and E. Arcaute. Multifractal methodology. *Physica A: Statistical Mechanics and its Applications*, 473:467–487, May 2017.

[42] M. Suhling, M. Arigovindan, P. Hunziker, and M. Unser. Multiresolution moment filters: theory and applications. *IEEE Transactions on Image Processing*, 13(4):484–495, April 2004.

[43] Dm.V. Uchaev. Development of the method of spectral-spatial classification of hyperspectral images based on local multifractal analysis and the support vector machine. *Current Problems in Remote Sensing of the Earth from Space*, 16(1):46–57, 2019.

[44] Dm.V. Uchaev and D.V. Uchaev. Theory and methodology of multifractal interpretation of aerospace images. In *Twelfth International Conference on Machine Vision (ICMV 2019)*, volume 11433, page 114333B. International Society for Optics and Photonics, January 2020.

[45] Dm.V. Uchaev, D.V. Uchaev, and V.A. Malinnikov. Chebyshev multifractal signatures and their use in multifractal interpretation of SAR images of ice-covered sea areas. In *Twelfth International Conference on Machine Vision (ICMV 2019)*, volume 11433, page 1143307. International Society for Optics and Photonics, January 2020.

[46] Dm.V. Uchaev, D.V. Uchaev, and V.A. Malinnikov. Spectral-spatial classification of hyperspectral images based on multifractal features. In *Image and Signal Processing for Remote Sensing XXVI*, volume 11533, page 115330T. International Society for Optics and Photonics, October 2020.

[47] X. Ying. An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022, February 2019.

[48] Q. Zhang, X. Shen, L. Xu, and J. Jia. Rolling guidance filter. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014*, Lecture Notes in Computer Science, pages 815–830, Cham, 2014. Springer International Publishing.