Machine Learning and Data Mining in Pattern Recognition
Volume 7988 of the series Lecture Notes in Computer Science pp 234-245. The final publication is available at Springer via http://dx.doi.org/10.1007/978-3-642-39712-7_18.

Evaluation of Hyperspectral Image Classification using Random Forest and Fukunaga-Koontz Transform

Semih Dinç[1] and Ramazan S. Aygün[2]

Computer Science Department
University of Alabama in Huntsville
Huntsville, AL 35899 USA
[1]sd0016@uah.edu, [2]aygunr@uah.edu
www.cs.uah.edu

Abstract. Since hyperspectral imagery (HSI) (or remotely sensed data) provides more information (or additional bands) than traditional gray level and color images, it can be used to improve the performance of image classification applications. A hyperspectral image presents spectral features (also called spectral signature) of regions in the image as well as spatial features. Feature reduction, selection, and transformation has been a challenging problem for hyperspectral image classification due to the high number of dimensions. In this paper, we firstly use Random Forest (RF) algorithm to select significant features and then apply Kernel Fukunaga Koontz Transform (K-FKT), a non-linear statistical technique, for the classification. We provide our experimental results on AVIRIS hyperspectral image dataset that contains various types of field crops. In our experimental results, we have obtained overall classification accuracy around 84 percent for the classification of 16 types of field crops.

1 Introduction

An ordinary color image has three bands (e.g., red, green, and blue) of the visible light just as human eye can see. So most imaging systems are restricted only a few spectral bands. These few bands or dimensions are usually not enough for the classification of a single pixel. However, hyperspectral imaging sensors divide the visible light spectrum into hundreds of bands for a single pixel as presented in Figure 1.

Spectral features (or bands) are sensitive to the type of material in addition to the color or shapes of objects. Because of this ability, hyperspectral imagery has different applications in many areas such as agriculture, mineralogy, physics, and homeland security. Although it was inconvenient technology in the past, recent advances have simplified the capture and process of hyperspectral images.

On the other hand, hyperspectral imagery (HSI) has some challenges to be overcome. Firstly, some of the spectral bands may have large amount of noise due to water absorption bands and some other environmental effects, since spectral

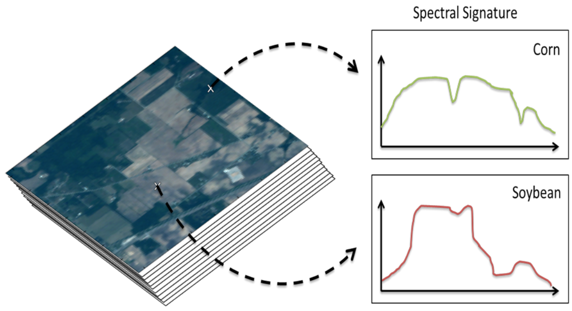

Fig. 1. Example of Hyperspectral Imagery

bands are more sensitive than typical vision sensors. Secondly, image processing and classification takes longer than the analysis of traditional gray scale and color images, since there are many number of features to be analyzed and processed. Therefore, some spectral bands might be noisy and redundant for image content analysis. It is necessary to detect the redundant bands and eliminate them to improve the processing speed and classification accuracy [1].

The research on HSI classification can be categorized based on feature selection and the classification method. The feature selection is sometimes handled manually by using prior information about the spectral bands or using previous experimental results. Various classification techniques including neural networks, support vector machines, bayesian classifier and decision trees have been used. In 2005, Benediktsson et al. [2] proposed a classification method using extended morphological models and neural networks. Banarjee et al. [3] studied on anomaly detection in HSI and used support vector machines for the classification in 2006. In 2007, Borges et al. [4] proposed discriminative class learning using a new Bayesian based HSI segmentation method. In 2008, Alam et al. [5] proposed a Gaussian filter and post processing method for HSI target detection. Du et al. [6] studied on HSI classification based on decision level fusion in 2010. In 2011, Tuia et al. [7] worked on the same topic using multi-scale cluster kernels. Samiappan et al. [8] proposed an SVM based HSI classification study which uses the same dataset used in this paper. Automated feature selection and acquiring high accuracy are still major problems for HSI.

In this study, we evaluate the combination of Random Forest (RF) algorithm and Kernel Fukunaga-Koontz transform (K-FKT). Firstly we performed automated feature elimination using RF algorithm, and then K-FKT is used to classify HSI data. We present our experimental results on the AVIRIS dataset [9] that contains images of 145x145 pixels with 220 spectral bands. Each spectral interval is 10 nm from the range 400 to 2450 nm wavelength. AVIRIS Image covers 2 x 2 mile portion of Northwest Tippecanoe County, Indiana.

This paper is organized as follows. The following section explains feature selection or ranking using Random Forest algorithm. Section 3 presents Kernel-Fukunaga-Koontz Transform in detail including its training and testing stages.

Classification results of different feature sets are presented in Section 4. The last section concludes our paper.

2 Feature Selection using Random Forest

Random Forest (RF) algorithm is applied to select informative features in spectral signatures. RF is a type of an ensemble classifier that employs many different (independent) decision trees. Basically in this method every single decision tree makes a prediction for a data item. The predictions of each decision tree are evaluated to determine the class of the data item. If the majority voting is used, the most voted class is chosen as the class of the data item. This approach as a random forest was first proposed by Leo Breiman and Adele Cutler in 2001 [10]. RF algorithm provides remarkably high accuracy in various studies [11] especially on large data sets. Cumbaa et al. [12] present the performance of RF algorithm on protein crystallization analysis using very a large database.

Random forests (RF) are comprised of decision trees and starts with selecting many bootstrap samples. A bootstrap data set has almost 63 percent of the original observations which are chosen randomly from the original dataset. The other samples which are not in the bootstrap dataset are called as out-of-bag observations. The best split for each tree is selected by using chosen attributes. This process repeats for each branch until our bootstrap grows into a proper well-formed tree. When the leaf nodes have small number of samples to split or no splitting criterion can be found, the decision tree induction ends. A decision tree is constructed for each bootstrap sample. Then each decision tree is employed to classify the out-of-bag observations. The predicted class of an observation is calculated by majority of votes of all decision trees for that observation (see Figure 2).

Generally a statistical classification or a regression method measures feature importance by choosing variables using statistical importance. However, RF approach runs in a completely different way. For each individual decision tree in the forest, there is a misclassification rate for the out-of-bag samples. In order to decide the importance of a specific predictor variable or a feature, the values of the features are randomly ordered for the out-of-bag observations. The algorithm performs prediction and checks for the change of the mean squared error (MSE) of out-of-bag data in which the corresponding variable is reordered and all others are fixed. In this way, a variable can be scored based on the prediction results.

3 Classification using Kernel Fukunaga-Koontz Transform

Kernel Fukunaga-Koontz Transform is applied in order to classify field crops in hyperspectral image for each selected feature set. Classical FKT is a well-known approach [13] [14] [15] [16] for separating two classes, and it operates by

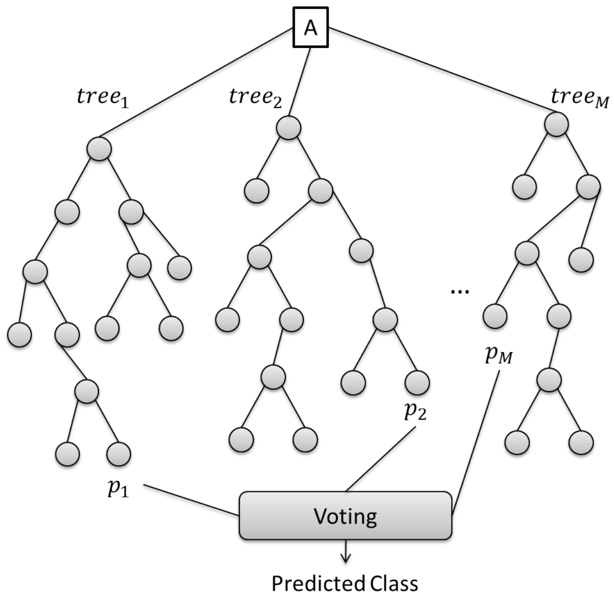

Fig. 2. Voting in Random Forest

transforming data into a new space where both classes share the same eigenvalues and eigenvectors. However, when the data is non-linearly distributed, the classical FKT method is not able to give satisfactory results for the classification. Therefore, Kernel transformation is combined with classical the FKT to classify non-linearly distributed data as if it was linearly distributed. K-FKT algorithm as a supervised classification approach consists of two stages: training and testing.

3.1 Training Stage

K-FKT is a binary classification approach. So we employ one-versus-all method to deal with the multi-class classification problem. The target class is the one which we want to classify, and the background class is the combination of all the other classes. Equations (1) and (2) represent these separated datasets for a sample target class as follows:

$$X = [x_1, x_2, ...x_N] \tag{1}$$

$$Y = [y_1, y_2, ...y_N] \tag{2}$$

where X and Y contain the target training data and the background training data, respectively; and x_i and y_i represent the samples (observations) for the target and background classes (or training signatures), respectively.

When data have a non-linear distribution, we are not able to separate classes with a linear classifier. But we can achieve this limitation with transforming the data into a higher dimensional space. Assume that a virtual mapping function maps the data into higher dimensional space as shown in (3) and (4):

$$\tilde{X} = [\tilde{x}_1, \tilde{x}_2, ...\tilde{x}_N] \tag{3}$$

$$\tilde{Y} = [\tilde{y}_1, \tilde{y}_2, ...\tilde{y}_N] \tag{4}$$

where the symbol '$\sim$' indicates that corresponding sample has been transformed to the higher dimensional kernel space. In that case transformed data may be linearly separable only if the proper mapping function is chosen. However, in most cases, the virtual mapping function does not exist (or we do not need to use) in applications. Instead, a kernel function 5 is employed by following a 'kernel trick' approach [15]. By using kernel trick, we are able to measure the distance between samples directly in higher dimensional kernel space without mapping data into that space.

$$K(x_i, x_j) = exp(\frac{\|x_i - x_j\|^2}{2\sigma^2}) \tag{5}$$

In this study Gaussian type kernel function is selected. This function measures the distances between two samples by using a calibration parameter σ ($0 < \sigma < 1$). In this way the classification process can be achieved in a linear fashion.

According to the FKT, covariance matrices of $\tilde{X}$ and $\tilde{Y}$ must be computed. As shown in (6) and (7), covariance matrices are named as T_0 and C_0 for the target and the background datasets, respectively:

$$T_0 = \tilde{X}\tilde{X}^T \tag{6}$$

$$C_0 = \tilde{Y}\tilde{Y}^T \tag{7}$$

The next step of the FKT algorithm is to sum T_0 and C_0 and to decompose this sum matrix into eigenvalues and eigenvectors. In (8) V represents the eigenvector matrix and D represents the eigenvalue matrix. Diagonal elements of D correspond to eigenvalues of the summation matrix.

$$T_0 + C_0 = VDV^T \tag{8}$$

In this way we are able to construct transformation operator P by using V and D matrices. This new operator is employed to transform data into the eigenspace. Note that this transformation is different from the kernel transformation in (5). Equation (9) shows how to derive the transformation operator:

$$P = VD^{-\frac{1}{2}} \tag{9}$$

Then T_0 and C_0 matrices are transformed into a lower dimensional eigenspace using the transformation operator P. The new T and C matrices are obtained as shown in (10) and (11),

$$T = PT_0P^T \tag{10}$$

$$C = PC_0P^T \tag{11}$$

where T and C represent the transformed target and background matrices, respectively. The sum of the transformed matrices should be equal to the identity matrix as shown in (12) since they are transformed by the same operator and share the same eigenvectors and eigenvalues:

$$I = T + C \tag{12}$$

Equation (12) states that while T contains more valuable information for the target class, C contains more important information for the background class. After obtaining these two matrices, the training stage is completed. These matrices are in the testing stage.

3.2 Testing Stage

In the testing stage, the test vector z must be transformed into the kernel space as performed in the training stage. Due to similar requirements as in the training stage, the 'kernel trick' method must be applied again for the kernel transformation:

$$Z = [K(x_1, z), K(x_2, z), ..., K(x_N, z)] \tag{13}$$

In (13), z is a test sample (spectral signature that we want to classify) on the hyperspectral image, x_i is the i^{th} target training sample in (1), $K(x, y)$ is a kernel function, and Z is the kernel matrix of the corresponding test sample. This matrix is necessary to have the transformed feature vector for the test sample as shown in (14):

$$F_j = \frac{1}{\sqrt{\lambda_j}}\phi_j^T Z \qquad j = 1, 2, ...N \tag{14}$$

where λ and ϕ represent the eigenvalues and eigenvectors of the normalized target matrix $\hat{T}$ in (15):

$$\hat{T} = T_0 - I_{1/N}T_0 - T_0I_{1/N} + I_{1/N}T_0I_{1/N} \tag{15}$$

where $I_{1/N}$ is equal to the division of an identity matrix I_{NxN} by N (I_{NxN}/N).

Then we decompose the eigenvalues and eigenvectors of the normalized target matrix $\hat{T}$ and multiply the feature vector F with the transpose of the eigenvector matrix of $\hat{T}$ as shown in (16):

$$R = \Phi^T F \tag{16}$$

where Φ represents the eigenvectors of $\hat{T}$. The decision result R is obtained by (16). This operation helps us decide the class that a test sample belongs. If the value of R is greater than a threshold value, it belongs to the target class; otherwise, it belongs to the background class.

4 Experimental Results

Dataset and Preprocessing. In this section, we explain the set of experiments which we performed using the AVIRIS Hyperspectral Image datasets called 'Indian Pines' [17]. The Indian Pines scene contains several type of areas. Among these areas, there are agricultural fields, forests, highways, a rail line, and some low density housing. The colored view of the scene is shown in Figure 3. In this dataset, there are 16 different classes which are represented as a ground truth data in Figure 4. The names of the classes and the number of samples are shown in Table 1.

Fig. 3. RGB view of AVIRIS Image

The AVIRIS data (or image) comprises of a $145 \times 145 \times 220$ matrix that corresponds to 220 different bands of images having size of 145×145. We transform the matrix data to a vector form as a 21025×220 matrix. This representation indicates that there are 21025 samples with 220 different features.

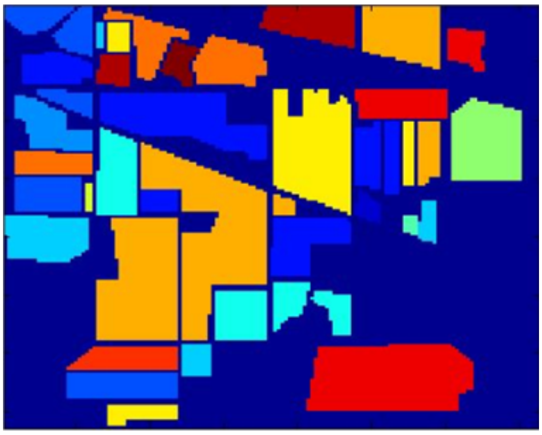

Fig. 4. Ground Truth data of AVIRIS Image

Class Number	Class	Samples
1	Alfalfa	54
2	Corn-notill	1434
3	Corn-mintill	834
4	Corn	234
5	Grass-pasture	497
6	Grass-trees	747
7	Grass-pasture-mowed	26
8	Hay-windrowed	489
9	Oats	20
10	Soybean-notill	968
11	Soybean-mintill	2468
12	Soybean-clean	614
13	Wheat	212
14	Woods	1294
15	Buildings-Grass-Trees-Drives	380
16	Stone-Steel-Towers	95
Total		10366

Table 1: Class Names and Number of Samples

Our goal is to classify the field crops. So firstly, we removed regions that do not correspond to field crops (dark blue areas in Figure 4) from the dataset. Among 21025 different samples, nearly half of them do not have meaning since they do not correspond to field crops. These samples are labeled with class 0 (zero). After this operation, the number of remaining samples is 10336 as presented in Table 1.

We performed a randomization to the permutation of the data in order to have a reliable classification result. After that we split data into two parts and choose 5190 samples for the training set and 5176 samples for the testing set.

Feature Selection Experiments. For the feature selection part, the number of of decision trees that are trained is set to 500. Only 14 features are available as the candidates at each split to increase the independence among decision trees. The square root of the total number of features is usually recommended as the number of candidate features at a node of a decision tree. According to these parameters, Figure 5 represents the importance coefficients, produced by RF, for each feature in the AVIRIS data.

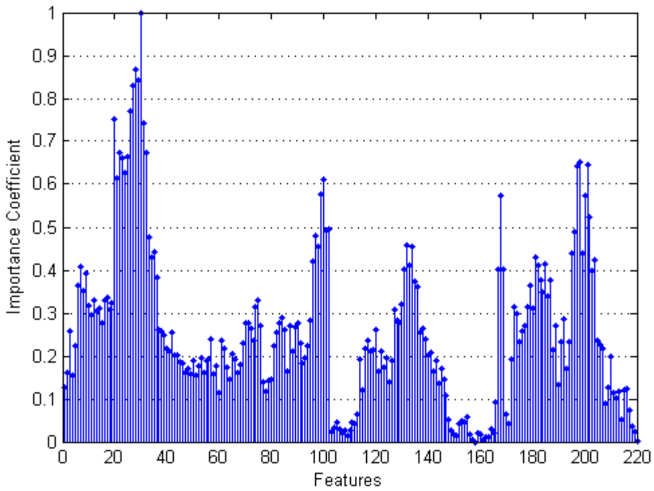

Fig. 5. Importance Coefficients of Features

Classification Performance.We rank the features and select the best N features using importance coefficients in Figure 5 for each experiment. Table 2 lists the precision and recall for each feature set. The results show that reducing the number of features increases the accuracy for most classes. This consequence verifies the presence of redundant features in HSI to be eliminated.

#of features		219	190	182	163	129	98	70	53	43	29	20	12	7	#of samp.
Class 1	prec	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.93	0.90	0.90	0.90	0.87	26
	recall	0.77	0.77	0.77	0.77	0.92	0.96	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	
Class 2	prec	0.76	0.76	0.76	0.76	0.71	**0.65**	0.61	0.60	0.60	0.60	0.60	0.47	0.46	716
	recall	0.89	0.89	0.89	0.90	0.92	0.95	0.97	0.98	0.98	0.99	**1.00**	0.85	0.81	
Class 3	prec	0.88	0.88	0.88	0.88	0.86	**0.82**	0.78	0.77	0.76	0.70	0.64	0.60	0.45	416
	recall	0.73	0.73	0.73	0.73	0.77	0.81	0.84	0.85	0.85	0.85	**0.85**	0.80	0.76	
Class 4	prec	0.95	0.95	0.95	0.95	0.89	**0.70**	0.65	0.64	0.64	0.64	0.64	0.64	0.49	116
	recall	0.93	0.93	0.93	0.93	0.95	0.97	1.00	1.00	1.00	1.00	1.00	**1.00**	0.86	
Class 5	prec	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.08	249
	recall	0.95	0.95	0.95	0.95	0.96	0.98	1.00	1.00	1.00	1.00	1.00	**1.00**	0.08	
Class 6	prec	1.00	1.00	1.00	1.00	1.00	**1.00**	0.97	0.92	0.89	0.43	0.41	0.41	0.36	373
	recall	0.87	0.87	0.87	0.87	0.88	0.90	0.93	0.95	**0.95**	0.76	0.69	0.69	0.57	
Class 7	prec	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.80	0.80	0.80	0.80	0.80	0.71	12
	recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	
Class 8	prec	1.00	1.00	1.00	1.00	1.00	**1.00**	0.98	0.97	0.95	0.92	0.89	0.89	0.85	245
	recall	0.95	0.95	0.95	0.95	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99	**1.00**	
Class 9	prec	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.92	0.92	0.92	10
	recall	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	
Class 10	prec	0.76	0.76	0.76	0.76	0.71	**0.68**	0.61	0.61	0.46	0.46	0.46	0.46	0.46	484
	recall	0.87	0.87	0.87	0.87	0.89	0.93	0.99	**0.99**	0.85	0.84	0.83	0.83	0.80	
Class 11	prec	0.79	0.79	0.79	0.79	0.75	**0.69**	0.66	0.65	0.64	0.45	0.45	0.45	0.45	1234
	recall	0.76	0.76	0.76	0.77	0.81	0.86	0.92	0.93	**0.94**	0.80	0.79	0.79	0.75	
Class 12	prec	0.85	0.85	0.85	0.85	0.84	**0.80**	0.70	0.66	0.63	0.60	0.46	0.46	0.46	306
	recall	0.91	0.91	0.91	0.91	0.92	0.92	0.97	0.98	0.99	**1.00**	0.86	0.86	0.83	
Class 13	prec	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.96	0.86	0.78	0.73	0.68	0.21	106
	recall	0.94	0.94	0.94	0.94	0.95	0.97	0.98	0.99	0.99	1.00	1.00	**1.00**	0.26	
Class 14	prec	0.81	0.81	0.81	0.81	0.79	**0.79**	0.80	0.80	0.80	0.80	0.79	0.79	0.79	646
	recall	0.91	0.91	0.91	0.91	0.93	0.96	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	
Class 15	prec	0.91	0.91	0.91	0.91	0.88	**0.84**	0.73	0.73	0.73	0.73	0.73	0.73	0.44	190
	recall	0.86	0.86	0.86	0.87	0.87	0.87	0.89	0.98	0.93	0.94	0.95	**0.95**	0.79	
Class 16	prec	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.87	0.74	0.47	0.41	47
	recall	0.72	0.72	0.72	0.72	0.72	0.72	0.79	0.83	0.85	0.89	**0.94**	0.89	0.70	
Avg.	prec	0.91	0.91	0.91	0.91	0.90	**0.87**	0.84	0.82	0.79	0.72	0.69	0.66	0.52	5176
	recall	0.87	0.88	0.88	0.88	0.90	0.93	0.95	**0.96**	0.95	0.94	0.93	0.92	0.77	

Table 2: Precision and Recall Classification Results of 16 Classes with respect to number of features

For overall evaluation of experimental results, Table 3 is generated with a single accuracy value for each case. This accuracy is computed as the ratio of the number of 'True Positive' and 'True Negative' samples to the number of all samples.

At the bottom of the table, the weighted accuracy is presented in which the ratio of the number of test samples in a class to the total number of test samples is considered as the weight of a class. Table 3 points out that to select best 98 features offers the best performance for classes.

#of features	219	190	182	163	129	98	70	53	43	29	20	12	7	#of samp.
Class 1	0.93	0.93	0.93	0.93	0.94	0.94	0.98	0.98	0.98	**1.00**	0.98	0.98	0.98	26
Class 2	0.72	0.72	0.72	0.72	0.74	0.75	0.75	0.75	**0.75**	0.74	0.74	0.74	0.74	716
Class 3	0.80	0.80	0.80	0.80	0.80	**0.80**	0.79	0.77	0.76	0.76	0.74	0.73	0.72	416
Class 4	0.89	0.89	0.89	0.89	0.92	**0.90**	0.87	0.80	0.79	0.71	0.68	0.68	0.51	116
Class 5	0.84	0.84	0.84	0.85	0.86	0.88	0.90	0.92	0.92	**0.92**	0.90	0.87	0.86	249
Class 6	0.91	0.91	0.91	0.91	0.91	0.91	0.94	**0.94**	0.93	0.90	0.84	0.70	0.63	373
Class 7	0.96	0.96	0.96	0.96	0.96	1.00	1.00	1.00	1.00	1.00	**1.00**	0.96	0.96	12
Class 8	0.93	0.93	0.93	0.93	0.94	0.96	0.97	0.98	**0.98**	0.97	0.96	0.95	0.95	245
Class 9	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	**1.00**	0.97	0.94	0.94	10
Class 10	0.83	0.84	0.84	0.84	0.85	**0.87**	0.81	0.77	0.73	0.69	0.68	0.67	0.66	484
Class 11	0.83	0.83	0.83	0.84	0.83	**0.82**	0.81	0.80	0.80	0.79	0.77	0.77	0.77	1234
Class 12	0.78	0.78	0.78	0.78	0.78	**0.77**	0.74	0.73	0.73	0.72	0.68	0.65	0.64	306
Class 13	0.94	0.94	0.94	0.95	0.95	0.97	0.98	**0.98**	0.97	0.96	0.97	0.95	0.94	106
Class 14	0.84	0.84	0.84	0.84	0.84	0.84	0.87	0.88	0.88	0.88	**0.88**	0.87	0.87	646
Class 15	0.84	0.84	0.84	0.84	0.86	**0.88**	0.84	0.83	0.82	0.82	0.81	0.79	0.79	190
Class 16	0.81	0.81	0.81	0.81	0.82	0.84	0.84	0.87	0.89	**0.91**	0.89	0.87	0.86	47
Weig. Acc.	0.83	0.83	0.83	0.83	0.84	**0.84**	0.83	0.82	0.82	0.81	0.79	0.77	0.76	5176

Table 3: Overall Accuracy Results

However, the performance of some classes such as "Class 2" and "Class 12" does not exceed 80% accuracy. In order to clarify this lower performance spectral signatures are examined in more detail and we realized that these two classes does not have a specific distinguishing behaviour like other classes. In other words, there are samples which are very similar to the other class signatures. That is why our classification approach is not able to predict these problematic samples. We have focused on using only spectral features in this study but employing also spatial features (e.g, neighbourhood information) of HSI may improve the results.

Comparison. Samiappan et al. [8] classifies the same dataset using support vector machines with non-uniform feature selection. They divide the spectral bands into regions in order to obtain the best feature set combination. Finally they employ radial-based SVM (Support Vector Machine) classifier to classify regions. Their results presented 75% of overall accuracy by using the 100 of 220 features on the AVIRIS dataset. In this study we select the best 98 features of 220. Our results point out a remarkable contribution and exceeds 75% accuracy by reaching 84% overall accuracy.

In a different study [18], same 'Indian Pine' dataset is used for the same classification problem. According to that study, their overall classification accuracy is 87% and slightly higher than our results. Although there is a 3% difference, the way of selecting training and testing samples may be different because the authors do not mention how they divided their data into training and testing. They also used higher number of training samples and lower number of test samples than our study.

Another important issue is that optimal number of features may be different for classes. So we do not have to use all 98 features for all classes. For example, choosing the best 29 features for classes 1, 5, 9 and 16 produce the highest accuracy. Similarly classes 7 and 14 give the best results using best 20 features. Therefore, if our goal is to classify a specific class we can consider the optimal feature set for that particular class. Otherwise, we can use 98 features (bold values in Table 3) for all classes with 84% overall accuracy.

5 Conclusion

In this study we evaluated a combination of two powerful methods for the HSI classification problem. We have used Random Forest algorithm to select the important features. Then we used Kernel Fukunaga-Koontz Transform to apply binary classification. Experimental results show that reducing the number of features using RF algorithm increases the performance up to some limit for majority of classes. Moreover we have obtained promising weighted classification accuracy around 84%.

References

1. Binol, H., Bal, A., Dinc, S.: Classification on hyperspectral images using enhanced covariance descriptor. In: Signal Processing and Communications Applications Conference (SIU), 2012 20th. (2012) 1–4
2. Benediktsson, J.A., Palmason, J.A., Member, S., Sveinsson, J.R., Member, S.: Sveinsson, classification of hyperspectral data from urban areas based on extended morphological profiles. IEEE Transactions on Geoscience and Remote Sensing **43** (2005) 480–491
3. Banerjee, A., Burlina, P., Diehl, C.: A support vector method for anomaly detection in hyperspectral imagery. IEEE T. Geoscience and Remote Sensing (2006) 2282–2291
4. Borges, J.S., Bioucas-Dias, J.M., Maral, A.R.S.: Bayesian hyperspectral image segmentation with discriminative class learning. In: IbPRIA (1)'07. (2007) 22–29
5. Alam, M.S., Islam, M.N., Bal, A., Karim, M.A.: Hyperspectral target detection using gaussian filter and post-processing. Optics and Lasers in Engineering **46**(11) (2008) 817 – 822
6. Du, P., Zhang, W., Xia, J.: Hyperspectral remote sensing image classification based on decision level fusion. Chin. Opt. Lett. **9**(3) (Mar 2011) 031002
7. Tuia, D., Camps-Valls, G.: Urban Image Classification With Semisupervised Multiscale Cluster Kernels. Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of (2011) 1
8. Samiappan, S., Prasad, S., Bruce, L.M.: Automated hyperspectral imagery analysis via support vector machines based multi-classifier system with non-uniform random feature selection. In: IGARSS, IEEE (2011) 3915–3918

9. : Aviris hyperspectral data @ONLINE. http://aviris.jpl.nasa.gov/html/data.html (2012)
10. Breiman, L.: Random forests. Mach. Learn. **45**(1) (October 2001) 5–32
11. Verikas, A., Gelzinis, A., Bacauskiene, M.: Mining data with random forests: A survey and results of new tests. Pattern Recognition **44**(2) (2011) 330 – 349
12. Cumbaa, C., Jurisica, I.: Protein crystallization analysis on the world community grid. Journal of Structural and Functional Genomics **11** (2010) 61–69 10.1007/s10969-009-9076-9.
13. Fukunaga, K., Koontz, W.L.G.: Application of the karhunen-loéve expansion to feature selection and ordering. IEEE Trans. Comput. **19**(4) (April 1970) 311–318
14. Ochilov, S., Alam, M.S., Bal, A.: Fukunaga-koontz transform based dimensionality reduction for hyperspectral imagery. Proceedings-SPIE The International Society For Optical Engineering **6233** (2006) 62332A–62332A–8
15. Li, Y.H., Savvides, M.: Kernel fukunaga-koontz transform subspaces for enhanced face recognition. 2012 IEEE Conference on Computer Vision and Pattern Recognition **0** (2007) 1–8
16. Dinc, S., Bal, A.: A statistical approach for multiclass target detection. Procedia Computer Science **6**(0) (2011) 225 – 230 Complex adaptive sysytems.
17. : Hyperspectral remote sensing scenes, aviris sensor (indian pines) @ONLINE. http://www.ehu.es/ccwintco/index.php/Sensores-hiperespectrales (2013)
18. Bagan, H., Yasuoka, Y., Endo, T., Wang, X., Feng, Z.: Classification of airborne hyperspectral data based on the average learning subspace method. Geoscience and Remote Sensing Letters, IEEE **5**(3) (2008) 368–372