Object Classification via Feature Fusion Based Marginalized Kernels

Xiao Bai, Chuntian Liu, Peng Ren, Jun Zhou *Senior Member, IEEE*, Huijie Zhao, and Yun Su

***Abstract*—Various types of features can be extracted from very high resolution remote sensing images for object classification. It has been widely acknowledged that the classification performance can benefit from proper feature fusion. In this paper, we propose a softmax regression based feature fusion method by learning distinct weights for different features. Our fusion method enables the estimation of object-to-class similarity measures and the conditional probabilities that each object belongs to different classes. Moreover, we introduce an approximate method for calculating the class-to-class similarities between different classes. Finally, the obtained fusion and similarity information are integrated into a marginalized kernel to build an SVM classifier. The advantages of our method are validated on Quickbird imagery.**

***Index Terms*—Feature fusion, remote sensing image, object classification, kernel.**

I. Introduction

Very high resolution (VHR) remote sensing images have been used for land cover and object classification for several decades. Both pixel-based and object-based approaches have been investigated for this purpose. Pixel-based classification methods identify the class of each pixel individually. Object-based methods, on the other hand, aim at classifying regions generated from image segmentation. To this end, object features such as shape characteristics and neighborhood relationships have been exploited for the categorization purpose.

One key step in the object-based classification is feature extraction for object encoding. VHR remote sensing images have demonstrated unique advantages in feature extraction because their high spatial resolution can provide a variety of detailed information (*i.e.* shape, spectral and texture) of objects. To effectively use these features for classification, some approaches have been proposed to combine different types of features. One solution is encoding multiple features into a low-dimensional vector. For example, Jimenez *et al.* [1] introduced an unsupervised pixel homogeneity enhancement method to integrate the spectral and spatial information in a local neighborhood. Van Coillie *et al.* [2] built an image classifier by exploiting genetic algorithms for feature selection. Combining multiscale features is another effective solution for object classification. Huang *et al.* [3] proposed a wavelet method for multiscale spectral and spatial information fusion.

On the other hand, some methods aim to learn the feature relevance in classification tasks. Zhang *et al.* [4] introduced a path alignment framework to linearly combine multiple features and to obtain a unified low-dimensional representation of these features. Tuia *et al.* [5] proposed a kernel-based framework to learn relevant weight in different features, resulting in an optimized linear combination of kernels. Nevertheless, most aforementioned methods have not considered the dependency between features and classes. Therefore, it is not guaranteed that the selected features are the most relevant for a specific classification task.

A partial solution to address this limitation is the logistic regression based feature fusion (LRFF) [6], which learns the feature weights with respect to the binary classes by a logistic regression model. However, this model is not feasible when object classes are mutually exclusive or when the number of classes is larger than two. Moreover, the kernel representation in this model ignores the relationships between different classes and is unable to capture feature similarities accurately. To tackle these problems, we propose a feature fusion method based on softmax regression and develop a new marginalized kernel which generalizes the models presented in [7]. This method learns the weights of each component in a feature vector and use this information to calculate the object-to-class similarity. It measures the margin between images and the learned class hyperplanes. The dependency between objects and classes can be thus effectively explored. Moreover, we develop an approximate method for calculating the similarity between different classes, which we call the class-to-class similarity. All these obtained information are then integrated to build the proposed novel marginalized kernel.

The contributions of this paper are three-fold. Firstly, we use the softmax regression, rather than logistic regression, to model the probabilities of each sample object belonging to the object classes. Secondly, to characterize the relationships between different classes, we develop an approximate method for calculating the class-to-class similarity, which is also incorporated into the marginalized kernel. Finally, we present the proposed new marginalized kernel and show its effectiveness

This work was supported in part by NSFC under Grants 61370123, 61105002, Shandong Outstanding Young Scientist Fund under Grant BS2013DX006, the Open Project Program of the National Laboratory of Pattern Recognition (NLPR), Australian Research Councils DECRA Projects funding scheme (project ID DE120102948), and High Resolution earth observation system major project for youth innovation support project.

X. Bai and C. Liu are with School of Computer Science and Engineering, Beihang University, Beijing 100191, China (e-mail: lcntn3@gmail.com; baixiao.buaa@gmail.com).

P. Ren is with College of Information and Control Engineering, China University of Petroleum (Huadong), Qingdao 266580, China (e-mail: pengren@upc.edu.cn).

J. Zhou is with School of Information and Communication Technology, Griffith University, Nathan, QLD 4111, Australia (e-mail:jun.zhou@griffith.edu.au).

H. Zhao is with School of Instrumentation Science and Optoelectronics Engineering, Beihang University, Beijing 100191, China (e-mail: huijiezhao@buaa.edu.cn).

Y. Su is with Beijing Institute of Space Mechanics and Electricity, Beijing 100094, China (e-mail: syun@126.com).

in object classification by fusing shape, spectral and textural features.

II. Feature Fusion with Marginalized Kernels

In this section, we give brief introduction on how to encode object samples into concatenated feature vectors. Based on the coding strategy, we develop a novel feature fusion method using a softmax regression model. Then we describe how to approximate the similarity between different classes. Finally, we use the feature fusion output and the class-to-class similarity to construct a new marginalized kernel for SVM classifiers. A summary of the proposed method is shown in Figure 1.

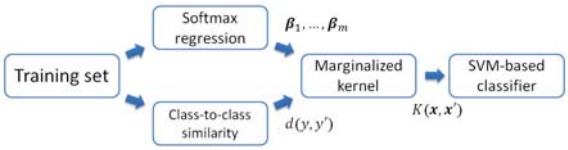

Fig. 1. Key steps of the proposed method.

A. Object Representation

The first step of the proposed method is to encode an object sample into a feature vector that will be used for feature fusion. To this end, the traditional Bag-of-Visual Words (BoW) model [8] is adopted. Suppose that the number of distinct types of features extracted for characterizing an object is k, and each type of feature is clustered into $d_i (1 \leq i \leq k)$ distinctive codewords. The BoW model calculates a d_i-dimensional vector for each type of feature, with each entry of the vector corresponds to the frequency that a codeword appears in the object sample. It then generates a feature vector $\boldsymbol{x}$ of dimensionality $d = \sum_{i=1}^{k} d_i$ for each object sample by combining all features through concatenation. An example of object representation is shown in Fig. 2. Here each component in the feature vector corresponds to one codeword. A training dataset that contains n object samples from m class are denoted by $\{(\boldsymbol{x}^{(i)}, y^{(i)}), i = 1 \ldots n\}$, where $\boldsymbol{x}^{(i)}$ and $y^{(i)} \in \{1, \ldots, m\}$ are the feature vector and the class label of the object sample i, respectively.

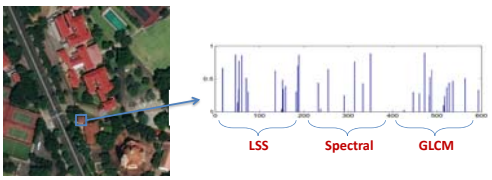

Fig. 2. An example of object representation with three types of features concatenated. Local self-similarity (LSS) and Gray-level co-occurrence matrices (GLCM) stand for shape and texture features respectively.

The BoW model assigns an equal weight to each codeword. In practice, the effectiveness of different types of features and different codewords varies for different classification tasks. Therefore, it is desirable that each type of features are weighted distinctively with respect to the object class. If one type of feature contributes significantly to a class in object characterization, it is expected that large weights should be assigned to the feature and its codewords with respect to this class. In this paper, we refer to the process of weight assignment to feature components as feature fusion.

B. Feature Fusion Using Softmax Regression

To achieve the goal of feature fusion, we investigate how to learn the feature weights with respect to different classes. Different from [6], which used logistic regression to handle the binary classification problem, we use the softmax regression [9] as the learning method. In this context, we straightforwardly address the multi-class problem and do not need to partition the training set into two classes to feed the one-vs-rest classification system. Moreover, softmax regression is capable for addressing the relationship between objects and classes when the classes are mutually exclusive.

Softmax regression models the posteriori probabilities of the feature components in the feature space via a regression function. Here the 1-of-m coding scheme is adopted to represent the class labels. Let $\boldsymbol{t}^{(i)} = \{\boldsymbol{t} | \boldsymbol{t} \in \{0,1\}^m, \|\boldsymbol{t}\|_1 = 1\}$ be the label vector for $\boldsymbol{x}^{(i)}$ with $t_j^{(i)} = 1$ if $\boldsymbol{x}$ belongs to class j and $t_j^{(i)} = 0$ otherwise (t and y are both class labels but in different forms). Suppose $\boldsymbol{\beta} = \{(\boldsymbol{\beta}_i)\}_{i=1}^m$ is a $(d \times m)$-dimensional weight vector, which is the target of learning. The probability of $\boldsymbol{x}$ belonging to class i can be written as

$$p(t_i = 1 | \boldsymbol{x}, \boldsymbol{\beta}) = \frac{\exp\{{\boldsymbol{\beta}_i}^{\mathrm{T}} \boldsymbol{x}\}}{\sum_{j=1}^{m} \exp\{{\boldsymbol{\beta}_j}^{\mathrm{T}} \boldsymbol{x})\}} \tag{1}$$

The term $\boldsymbol{\beta}_j^{\mathrm{T}} \boldsymbol{x}$ reflects the object-to-class similarity between the object sample $\boldsymbol{x}$ and the class j, which has been proved to be effective in the nearest-neighbor based image classification [10]. Particularly, the relation in (1) is a logistic regression model when $m = 2$. For $m > 2$, it leads to softmax regression, or multinomial logistic regression. The conditional probabilities formulated in (1) satisfies the normalization rule

$$\sum_{i=1}^{m} p(t_i = 1 | \boldsymbol{x}, \boldsymbol{\beta}) = 1 \tag{2}$$

Assuming the training set $\left\{(\boldsymbol{x}^{(i)}, \boldsymbol{t}^{(i)})\right\}_{i=1}^n$ are i.i.d. (independently and identically distributed) with respect to $\boldsymbol{t}$ given $\boldsymbol{x}$ and $\boldsymbol{\beta}$, the log-likelihood function can be formulated as

$$\begin{aligned} \ell(\boldsymbol{\beta}) &= \sum_{j=1}^{n} \log p(\boldsymbol{t}^{(j)} | \boldsymbol{x}^{(j)}, \boldsymbol{\beta}) \\ &= \sum_{j=1}^{n} \left\{ \sum_{i=1}^{m} y_i^{(j)} {\boldsymbol{\beta}_i}^{\mathrm{T}} \boldsymbol{x}^{(j)} - \log \sum_{i=1}^{m} \exp\left({\boldsymbol{\beta}_i}^{\mathrm{T}} \boldsymbol{x}^{(j)}\right) \right\} \end{aligned} \tag{3}$$

The optimal value of $\boldsymbol{\beta}$ is obtained by maximizing a posteriori estimation

$$\hat{\boldsymbol{\beta}} = \arg\max_{\boldsymbol{\beta}} \{\ell(\boldsymbol{\beta}) + \log p(\boldsymbol{\beta})\} \tag{4}$$

where $p(\boldsymbol{\beta})$ is the posterior distribution of the parameter $\boldsymbol{\beta}$. In our framework, we formulate it with ℓ_1 norm regularization

$$p(\boldsymbol{\beta}) \propto \exp(-\lambda\|\boldsymbol{\beta}\|_1) \tag{5}$$

where λ is a regularization parameter controlling the bias-variance trade-off. The ℓ_1-regularization scheme not only avoids overfitting but also results in sparsity in the weight vector $\boldsymbol{\beta}$, which is effective in limiting the number of feature components (or visual words) involved in classification. Projection L1 method is used to optimize this model [11].

C. Class-to-Class Similarity

The similarities between different classes are the prior information that can be computed from the training set. We develop an approximate method for calculating class-to-class similarity that helps to measure the similarity of features. To this end, we calculate group average between two classes, which is the classical strategy used in agglomerative algorithms to compute the similarity between different clusters. Suppose $\mathcal{C}_1$ and $\mathcal{C}_2$ represent two sets containing the objects in class y_1 and y_2, and $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ represent two objects belonging to $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. Based on the BoW model, each component of the feature vector $\boldsymbol{x}$ represents the corresponding visual codeword occurrences. To define the similarity between objects, we convert codeword occurrences to the *tf-idf* weights

$$w_{ij} = \frac{c_{ij}}{\sum_j c_{ij}} \log \frac{n}{\sum_i c_{ij}} \tag{6}$$

where w_{ij} is the weight of the j-th component in the feature vector of object i, c_{ij} is the number of occurrences of the codeword j for the object i, and n is the total number of objects. The term $\sum_i c_{ij}$ denotes the number of object samples associated with the codeword j in the training dataset. Each object can be represented by a vector consisting of the normalized *tf-idf* weights $\boldsymbol{w}_i = [w_{i1}, w_{i2}, \ldots, w_{ik}]$, where k is the length of codebook. The similarity between objects is calculated using cosine metric $s(\boldsymbol{w}_1, \boldsymbol{w}_2) = \boldsymbol{w}_1 \cdot \boldsymbol{w}_2 / \|\boldsymbol{w}_1\|_2 \|\boldsymbol{w}_2\|_2$, and $s(\mathcal{C}_1, \mathcal{C}_2)$ is defined to be the average similarity between the object samples from the two different classes

$$s(\mathcal{C}_1, \mathcal{C}_2) = \text{mean}_{(\boldsymbol{w}_1, \boldsymbol{w}_2) \in \mathcal{C}_1 \times \mathcal{C}_2} s(\boldsymbol{w}_1, \boldsymbol{w}_2) \tag{7}$$

where $s(\boldsymbol{w}_1, \boldsymbol{w}_2)$ is the object similarity. The class-to-class similarity $S(y, y')$ for classes y and y' is defined as follows

$$S(y, y') = \begin{cases} 1 & \text{if } y = y'; \\ s(\mathcal{C}_1, \mathcal{C}_2) & \text{otherwise.} \end{cases} \tag{8}$$

D. New Marginalized Kernel

Given the training samples $(\boldsymbol{x}, y)$ and $(\boldsymbol{x}', y')$, the marginalized kernel is defined as follows

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_y \sum_{y'} p(y|\boldsymbol{x}) p(y'|\boldsymbol{x}') K_z(z, z'). \tag{9}$$

Here $p(y|\boldsymbol{x})$ and $p(y'|\boldsymbol{x}')$ are the conditional probabilities that $\boldsymbol{x}$ and $\boldsymbol{x}'$ belong to classes y and y' respectively, which can be obtained from the softmax regression (1). The term $K_z(z, z')$ is a joint kernel over the samples $z = (\boldsymbol{x}, y)$ and $z' = (\boldsymbol{x}', y')$ with class labels y and y' considered to be hidden variables.

We present a new marginalized kernel that exploits the output of softmax regression and class-to-class similarity. We formulate $K_z(z, z')$ in terms of the feature weights and class-to-class similarity as follows

$$K_z(z, z') = S(y, y') \times \boldsymbol{\beta}_y^{\mathrm{T}} \boldsymbol{x} \times \boldsymbol{\beta}_{y'}^{\mathrm{T}} \boldsymbol{x}'. \tag{10}$$

If the similarity of classes y and y' is high, their marginal hyperplanes in the feature space will be close. Then two objects locating on the same side of one hyperplane lead to a positive product $\boldsymbol{\beta}_y^{\mathrm{T}} \boldsymbol{x} \times \boldsymbol{\beta}_{y'}^{\mathrm{T}} \boldsymbol{x}'$, and the kernel $K(\boldsymbol{x}, \boldsymbol{x}')$ results in a high similarity accordingly. In [6], the authors only considered the case when $y = y'$. This simplification reduces the performance of the joint kernel because it ignores the relationships between different classes. To overcome this drawback, we incorporate the approximate class-to-class similarity into the marginalized kernel as described in (10). By substituting (10) into (9), we have the new kernel as follows

$$K(\boldsymbol{x}, \boldsymbol{x}') = \sum_y \sum_{y'} p(y|\boldsymbol{x}) \times p(y'|\boldsymbol{x}') \times S(y, y') \times \boldsymbol{\beta}_y^{\mathrm{T}} \boldsymbol{x} \times \boldsymbol{\beta}_{y'}^{\mathrm{T}} \boldsymbol{x}' \tag{11}$$

According to Mercer's theorem, one kernel function is valid if it can be represented as a inner product of two feature vectors. In our framework, we map each feature $\boldsymbol{x}$ to $\boldsymbol{\beta}^{\mathrm{T}} \boldsymbol{x}$. The term $\boldsymbol{\beta}_y^{\mathrm{T}} \boldsymbol{x} \times \boldsymbol{\beta}_{y'}^{\mathrm{T}} \boldsymbol{x}'$ can be thought of as the inner product $\langle \boldsymbol{\beta}_y^{\mathrm{T}} \boldsymbol{x}, \boldsymbol{\beta}_{y'}^{\mathrm{T}} \boldsymbol{x}' \rangle$. Subject to the closed property under addition and multiplication of the positive semidefinite (PSD) kernels, our new kernel $K(\boldsymbol{x}, \boldsymbol{x}')$ is also a valid kernel.

Finally, we have the decision function for support vector machines formulated as follows

$$H(\boldsymbol{x}') = \text{sgn}\left(\sum_{i=1}^{n} y_i \alpha_i K(\boldsymbol{x}, \boldsymbol{x}') + b \right) \tag{12}$$

where α_i is the Lagrange multiplier. The kernel function $K(\boldsymbol{x}, \boldsymbol{x}')$ is formulated in terms of (11), rather than the linear or Gaussian kernel commonly used in the research literature.

III. Experiments

A. Experimental Setup

We test our method on a 4000×4000 high-resolution satellite image taken in Chengdu, China, in 2009, which consists of three bands (RGB) with 0.6-m resolution. Five classes of objects are identified based on visual appearance and spatial proximity. In total, 5009 objects of lands, lawns, residential areas (RA), roads and trees are collected after image segmentation by a commercial software eCognition (the shape parameter value is set to 0.5, the parameter scale is set to 50, and the other parameter values are set as default values). We randomly split the image data into a training set and a testing set for ten times and then calculate the average classification accuracy as the final result. Table I shows the size of the training and testing sets for each object class.

To evaluate the performance of the proposed method, we choose the support vector machine (SVM) classifiers with two types of kernels (linear and RBF) and the LRFF method [6] as the baseline.

TABLE I
NUMBER OF TRAINING AND TESTING SAMPLES FOR EACH OBJECT CLASS

CLASS	TRAINING	TEST
C1 - Land	300	452
C2 - Lawn	413	620
C3 - RA	485	728
C4 - Road	479	719
C5 - Tree	325	488
TOTAL	2002	3007

B. Feature Extraction

Four types of features are used in the experiment, which characterize the properties of the segmented regions in three aspects: shape, spectral and texture. For the shape information, we use the scale invariant feature transform (SIFT) [12] and local self-similarity (LSS) [13]. In each segmented region, features are extracted in evenly sampled grid with 10 pixels apart in both horizontal and vertical directions. The size of the patch is randomly sampled between a scale of 10 to 30 pixels. Moreover, we use the mean and standard deviations of three spectral bands (*i.e.*, RGB) as the spectral feature. Gray-level co-occurrence matrices (GLCM) [14] is used to represent the texture feature.

C. Classification Results

In the experiments, we use the following feature combinations: 1) SIFT+Spectral; 2) SIFT+Spectral+GLCM; 3) SIFT+LSS+Spectral+GLCM. The number of codewords are set to be 250, 200, 200 and 200 for SIFT, Spectral, LSS and GLCM, respectively. SVM classifiers with different kernels are employed to perform classification. Table II shows the classification results To analyze the results, we compute two statistical measures, i.e., overall accuracy (OA) and average accuracy (AA). Fig. 3 illustrates a sample subimages segmented by using the software eCognition, and its classification results by using our method with four features.

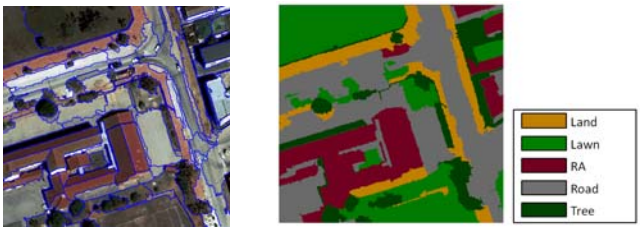

Fig. 3. (a) Segmented subimage; (b) Classification result.

The classification results with different kernel methods can be seen in Table II. An intuitive observation is that more features lead to better classification results. This is natural because distinct features have captured different characteristics of the objects in the image. The LRFF [6] and our methods have shown significantly better performance than the SVM methods with linear and nonlinear kernels. Such results validate the advantage of feature fusion and the proposed marginalized kernel. In particular, the proposed method with four features has yielded the best results among all the combinations of features and classifiers. The best overall accuracy of our approach is 94.39%, with 8.29%, 4.26%, 1.48% improvement over those from SVM (linear), SVM (RBF) and LRFF methods, respectively. This has demonstrated the advantage of our softmax regression based model in measuring feature similarity in multiple feature space.

D. Influence of Size of Training Set

Fig. 4 shows the changes of the overall accuracy as a function of the percentage of training samples. The performance of the two kernel methods both increase as the number of training samples increases. However, our method maintains an advantage of 3% to 6% over the LRFF method. These results show that, by using softmax regression model and class-to-class similarity, more effective kernel machines can be constructed for encoding the relationships of the observed data. Moreover, to evaluate the significance of the proposed method, the classification results are then processed by the resampled paired t test, where the number of resampled times is 10. All the tests are performed using 2-sides tests with the confidence level 0.05. In Fig. 5, the p-values are less than 0.05 in most cases, indicating that the performances of our method and LRFF are significantly different.

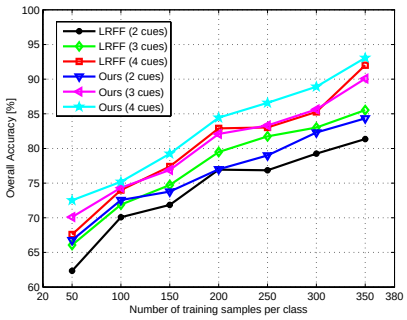

Fig. 4. Classification accuracy using LRFF and our proposed with 2, 3 and 4 features (cues). Detailed feature combination can be seen in Table II.

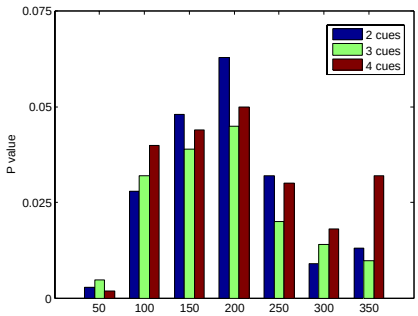

Fig. 5. P-value for each statistical test. The three adjacent bars for each test present the p-values of three types of feature combinations.

E. Robustness against Outliers

In the BoW model, each patch is assigned to the closest codeword to obtain the object representation. However, patches

TABLE II
OVERALL ACCURACY(OA[%]) AND AVERAGE ACCURACY(AA[%]) FOR DIFFERENT KERNEL CLASSIFIERS: SVM WITH LINEAR KERNEL (SVM-LINEAR), SVM WITH RBF KERNEL (SVM-RBF), SVM WITH LR-BASED MARGINALIZED KERNEL (LRFF), AND THE PROPOSED METHOD (OURS)

METHOD	FEATURES	CATEGORIES					RESULTS	
		C1	C2	C3	C4	C5	OA[%]	AA[%]
SVM-linear	SIFT+Spectral	74.19	74.08	77.43	78.04	83.70	77.42	77.49
	SIFT+Spectral+GLCM	81.00	82.97	84.91	82.43	85.03	83.35	83.27
	SIFT+LSS+Spectral+GLCM	84.08	85.23	88.65	84.82	87.18	86.10	85.99
SVM-RBF	SIFT+Spectral	79.89	78.43	81.32	79.97	85.78	80.91	81.08
	SIFT+Spectral+GLCM	85.09	83.86	87.82	84.25	87.54	85.69	85.71
	SIFT+LSS+Spectral+GLCM	89.67	87.88	92.08	89.78	91.04	90.13	90.09
LRFF	SIFT+Spectral	81.09	83.78	84.83	83.09	87.61	84.09	84.08
	SIFT+Spectral+GLCM	87.02	87.96	89.59	87.73	90.08	88.50	88.48
	SIFT+LSS+Spectral+GLCM	91.06	92.39	93.82	93.42	93.19	92.91	92.78
Ours	SIFT+Spectral	84.77	83.25	86.54	84.04	88.06	85.24	85.33
	SIFT+Spectral+GLCM	89.14	91.06	92.42	90.23	91.75	90.01	90.92
	SIFT+LSS+Spectral+GLCM	**92.86**	**93.42**	**95.40**	**94.84**	**94.87**	**94.39**	**94.28**

in an object shall not be assigned to any suitable codeword if the object is an outlier which does not belong to any sample classes. To reduce the influence of outliers, we employ the thresholding strategy described in [8]. In the training stage, for each cluster, we set the threshold as the maximum distance between its center and the patches in it. If the distance between one patch and its closest codeword is smaller than the chosen threshold, the patch will be assigned to this codeword. Otherwise, we assign the patch to a single virtual codeword. In this way, the virtual codeword occurs more frequently in the outliers than others, which is helpful to detect the outliers. To evaluate the robustness of our method against outliers, we include 50 objects from undefined classes in the classification experiments. The confusion matrix are shown in Table III, in which 45 objects are rejected correctly and only 5 objects are misclassified. The overall accuracy is 94.21%, which is promising and comparable to the experimental results in Table II. This validates the robustness of the proposed kernel against outliers.

TABLE III
CONFUSION MATRIX OF OBJECT CLASSIFICATION WITH 50 OUTLIERS ADDED.

	Predicted					
Actual	Land	Lawn	RA	Road	Tree	Outliers
Land	419	8	2	15	6	2
Lawn	6	579	5	11	18	1
RA	4	6	694	12	11	1
Road	13	10	11	681	4	0
Tree	3	9	7	2	462	5
Noise	1	1	0	0	3	45

Overall accuracy = 94.21%

IV. CONCLUSION

In this paper, we have introduced a softmax regression-based feature fusion method for object classification. This method takes into account the information about object-to-class similarities and the conditional probabilities that one object sample belongs to different classes. Moreover, an approximate method has been developed for measuring the similarity between different classes, and the information between features and classes could thus be better modeled. All the obtained information have been integrated for the development of a novel marginalized kernel. Experimental evaluations show that the proposed method has outperformed the baseline SVM and LRFF methods.

REFERENCES

[1] L. Jimenez, J. Rivera-Medina, E. Rodriguez-Diaz, E. Arzuaga-Cruz, and M. Ramirez-Velez, "Integration of spatial and spectral information by means of unsupervised extraction and classification for homogenous objects applied to multispectral and hyperspectral data," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 43, no. 4, pp. 844–851, 2005.

[2] F. Van Coillie, L. P. Verbeke, and R. R. De Wulf, "Feature selection by genetic algorithms in object-based classification of IKONOS imagery for forest mapping in Flanders, Belgium," *Remote Sensing of Environment*, vol. 110, no. 4, pp. 476–487, 2007.

[3] L. Z. X. Huang and P. Li, "A multiscale feature fusion approach for classification of very high resolution satellite imagery based on wavelet transform," *International Journal of Remote Sensing*, vol. 29, no. 20, pp. 5923–5941, 2008.

[4] L. Zhang, L. Zhang, D. Tao, and X. Huang, "On combining multiple features for hyperspectral remote sensing image classification," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 50, no. 3, pp. 879–893, 2012.

[5] D. Tuia, G. Camps-Valls, G. Matasci, and M. Kanevski, "Learning relevant image features with multiple-kernel classification," *IEEE Trans. on Geosci. and Remote Sens.*, vol. 48, no. 10, pp. 3780–3791, 2010.

[6] B. Fernando, E. Fromont, D. Muselet, and M. Sebban, "Discriminative feature fusion for image classification," in *CVPR*, 2012.

[7] K. Tsuda, T. Kin, and K. Asai, "Marginalized kernels for biological sequences," *Bioinformatics*, vol. 18, no. suppl 1, pp. S268–S275, 2002.

[8] S. Xu, T. Fang, D. Li, and S. Wang, "Object classification of aerial images with bag-of-visual words," *IEEE Geosci. and Remote Sens. Letters*, vol. 7, pp. 366–370, 2010.

[9] G. C. Cawley, N. L. Talbot, and M. Girolami, "Sparse multinomial logistic regression via bayesian l1 regularisation," 2007.

[10] E. O.Boiman and M.Irani, "In defense of nearest-neighbor based image classification," in *CVPR*, 2008.

[11] M. Schmidt, G. Fung, and R. Rosales, "Fast optimization methods for l1 regularization: A comparative study and two new approaches," in *ECML*, 2007.

[12] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91–110, 2004.

[13] H. Zhang, X. Bai, H. Zhang, H. Zhao, J. Zhou, J. Cheng, and H. Lu, "Hierarchical remote sensing image analysis via graph laplacian energy," *IEEE Geosci. and Remote Sens. Letters*, vol. 10, no. 2, pp. 396–400, 2013.

[14] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, no. 6, pp. 610–621, 1973.