International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol.9, No.10, (2016), pp.305-314
http://dx.doi.org/10.14257/ijsip.2016.9.10.29

Multi-Feature Learning via Hierarchical Match Kernel for Image Classification

Fenxia Wu

School of Computer Science, Xianyang Normal University, Xianyang, Shaanxi, China, 712000
wufenxia@126.com

Abstract

Image classification is an important task in computer vision. The methods based on spatial information generally employ some low-level features for image classification, such as gray scale, color, texture and location. It is difficult for vision system to understand and the single feature is too limited to obtain correct classification results. In this paper, an algorithm based on multi-kernel feature learning is proposed and used for image classification. First, the kernel function is used to produce a kernel descriptor, which aggregates the pixel attributes into patch-level features; Then, through the multi-kernel learning, these descriptors are further aggregated to obtain hierarchical multi-feature descriptors; Finally, the label of each image is given by the fusion strategy of on multi-classifiers, which effectively utilizes the advantages of multi-kernel learning and takes the complementary among the classifiers into account. The experimental results show that the proposed method is efficient in promoting the classification results.

Keywords: *image classification; multi-kernel learning; hierarchical kernel; multi-classifiers fusion*

1. Introduction

Image classification assigns a label to a specific image according to different characteristics from the image. Automatic image classification can help people to organize and manage the massive digital images, which brings great convenience to human life. Therefore, image classification has become an important subject in the computer vision field.

Image classification consists of feature extraction, feature fusion, feature learning and classification. The feature extraction of the image is a basic and important step, which is a prerequisite for classification. Features of a specific image are the differences between the other images. Usually, the image visual features are divided into global and local features. Global features refer to the original feature attributes such as color, texture and shape, which cannot express completely the accurate image information. So, it leads to a decrease in discriminant analysis and low classification accuracy. Compared with the global features, local features includes brightness, scale, shift and rotation invariance, but the image description ability of local features is not strong, and it ignores the color information of the image. It is difficult for a single feature to classify the images correctly, because of the rich information in the image. Many research results suggest that [1]: multi-kernel learning can effectively solve the above problems. The multi-kernel learning based methods can select the corresponding kernel functions and then combine them according to the different features of the image. It is better than the single kernel function to describe the image due to the combination of multi-kernel. The nonlinear mapping characteristics of kernel functions are preserved in multi-kernel, which integrate the advantages of different kernel functions. It can be seen that the multi-kernel method has

ISSN: 2005-4254 IJSIP

shown excellent properties and produced high classification accuracy in image classification [2].

Feature extraction and classifier design also have significant influences on image classification. With the improvement of classification accuracy, many researchers proposed lots of efficient classification algorithms, such as the minimum distance method, Bayesian method, K-mean, neural network, maximum likelihood method and support vector machine, etc. Although these algorithms have their own advantages, there is no one algorithm better than other algorithms for all the classification problems. Each classifier has its own limitations. Therefore, multi-classifier fusion method is proposed. The so-called multi-classifier fusion is combining complementary advantages of the different trained multi-classifier to get better results than the single classifier.

This paper proposes an image classification method based on hierarchical match multi-kernel feature learning and multi-classifier fusion. First, gradient, color and shape kernel descriptors are calculated by the kernel function from the images. Second, multi-kernel learning algorithm is used to obtain the hierarchical multi-kernel descriptor. Finally, the multiple classifiers are fused to obtain final labels of images.

2. Hierarchical Match Multi-Kernel

Because the single feature is difficult to obtain accurate classification results, combination of multi-feature, such as color, texture, shape and other low-level visual features, will be helpful for the improvement of the classification accuracy. That is to say image classification will be more accurate for more obtained information [3]. However, it is key to effectively combine image features. Although the kernel based method has shown its superiority in image classification, each kernel function can only describe one feature of the image. For Multi-kernel learning, it integrates linearly different kernels in different appropriate proportions to produce a multi-kernel function for classification training, which replaces the original kernel function. Beside, multi-kernel learning method can be used to measure the importance of different image features [4].

2.1. Histograms of Oriented Gradient (HOG)

Histograms of Oriented Gradient (HOG) is proposed by N. Dalal and B. Triggs in 2005 [5], which can effectively characterize the local differential information of image and is not susceptible to noise. The distribution of the local gradient or edge direction can represent the local appearance and shape of the object, in which the exact location of the gradient need not to know. In HOG, the sample image s segmented into a number of small patches which consists of 4 adjacent units. Each unit is size of 8*8 pixels. $\left[-\pi/2,\pi/2\right]$ is divided into nine intervals and histogram features of all the pixels in all directions are computed to histogram features of each unit and each patch. These histogram features of patches are cascaded to produce the histogram features of the image. The horizontal and vertical gradients of the pixel in the input image are respectively:

$$\begin{aligned} I_x\left(x,y\right) &= I(x+1,y)-I(x-1,y) \\ I_y\left(x,y\right) &= I(x,y+1)-I(x,y-1) \end{aligned} \tag{1}$$

By the gradient operator [-1, 0, 1], the gradient value and direction of pixel $\left(x,y\right)$ are:

$$\begin{aligned} m(x,y) &= [(I_x(x,y))^2+(I_y(x,y))^2]^{1/2} \\ \theta(x,y) &= \arctan(\frac{I_y(x,y)}{I_x(x,y)}) \end{aligned} \tag{2}$$

2.2. Local Binary Pattern (LBP)

Local Binary Pattern (LBP) [6] is used to describe the image texture feature. In the window with size 3*3, the center pixel value is considered as a threshold to compare with the adjacent pixel gray value. If the surrounding pixel value is greater than the threshold, the pixel is labeled as 1; otherwise 0. The binary column vector of the window is obtained. Then, the histogram of each cell is calculated and normalized. Finally, the histogram of each cell is connected as a feature vector, which is LBP texture feature vector of the image. LBP has some remarkable properties, such as, rotation invariance and gray scale invariance, and can effectively employ the global features of the image [7]. LBP is robust to the gray level changes caused by the illumination changes. Besides, it is easy to compute, which makes it possible for real-time image analysis.

3. Feature Extraction via Hierarchical Multi-Kernel

Kernel learning algorithm is an efficient machine learning algorithm developed in recent years, which can deal with the high dimensional feature space problem with low computational cost. The kernel function not only defines the similarity between the two samples, but also considers the regularization. Kernel function is transformed into kernel matrix to facilitate the subsequent use of multi-kernel learning, after the extraction of direction histograms, color feature and LBP.

3.1. Kernel Descriptor

We denote the location of pixel z in the image patch. P and Q are the image patches in different image sets. $\theta(z)$ and $m(z)$ denote the gradient direction and gradient value of pixel z, respectively. c_z is the color of pixel z. $b(z)$ is a binary column vector, which is the result of binarization in the local window. Then the following descriptors are used [5] for feature extraction.

Gradient kernel K_{grad} is:

$$K_{grad}(P,Q)=\sum_{z\in P}\sum_{z'\in Q}\tilde{m}(z)\tilde{m}(z')k_o(\tilde{\theta}(z),\tilde{\theta}(z'))k_p(z,z') \tag{3}$$

where normalized linear kernel $k_m(z)=\tilde{m}(z)\tilde{m}(z')$ is the same as HOG, which computes the contribution of each pixel by gradient values. Gaussian kernel $k_p(z,z')=\exp\left(-\gamma_p\left\|z-z'\right\|^2\right)$ is used to measure the relationship of spatial location of pixels. $k_o(\tilde{\theta}(z),\tilde{\theta}(z'))=\exp(-\gamma_o\|\tilde{\theta}(z)-\tilde{\theta}(z')\|^2)$ calculates the similarity of gradient directions of pixels. The normalized gradient vector, $\tilde{\theta}(z)=[\sin(\theta(z))\cos(\theta(z))]$, is considered when computing k_o.

Color kernel K_{col} is:

$$K_{col}(P,Q)=\sum_{z\in P}\sum_{z\in Q'}k_c(c_z,c_{z'})k_p(z,z') \tag{4}$$

where $k_c(c_z,c_{z'})=\exp(-\gamma_c\|c_z-c_{z'}\|^2)$ is utilized to calculate the similarity among pixel values.

Shape kernel K_{shape} is :

$$K_{shape}(P,Q)=\sum_{z\in P}\sum_{z^{'}\in Q}\tilde{s}(z)\tilde{s}(z^{'})k_b(b(z),b(z^{'}))k_p(z,z^{'}) \tag{5}$$

where $\tilde{s}(z)=s(z)\Big/\sqrt{\sum_{z\in P}s(z)^2+\varepsilon_s}$ and $s(z)$ is the pixel standard deviation in the window with size 3*3. ε_s is a very small constant. The normalized linear kernel $k_s(z)=\tilde{s}(z)\tilde{s}(z^{'})$ is used to compute the contribution of the local binary model. The shape similarity is measured by Gauss kernel $k_b(b(z),b(z^{'}))=\exp(-\gamma_b\,\|b(z)-b(z^{'})\|^2)$ on the local binary model.

Orientation histograms, color features and LBP represent the image from different perspectives and levels. Gradient kernel can capture the image variations. Image appearance can be described by color kernel and the local shape is characterized by shape kernel. However, they ignore the spatial information of the image, which can be obtained by adding the spatial pyramid structure [8] into the kernel function:

$$\begin{aligned}T_{P-col}&=\sum_{l=0}^{L-1}\sum_{t=1}^{2^l}2^{-l}K_{col}(P_{(l,t)},Q_{(l,t)})\\T_{P-grad}&=\sum_{l=0}^{L-1}\sum_{t=1}^{2^l}2^{-l}K_{grad}(P_{(l,t)},Q_{(l,t)})\\T_{P-shape}&=\sum_{l=0}^{L-1}\sum_{t=1}^{2^l}2^{-l}K_{shape}(P_{(l,t)},Q_{(l,t)})\end{aligned} \tag{6}$$

Where L is the number of layers of pyramid. 2^l is the number of spatial units in the jth layer. $P_{(l,t)}$ is the kernel descriptor locating in the spatial unit (l,t).

3.2. Hierarchical Multi-Kernel

We can not only use kernel learning in image patches to get the kernel descriptors, also can apply kernel learning on these kernel descriptors to obtain the hierarchical kernel descriptors. Hierarchical kernel descriptors can aggregate kernel descriptors of adjacent image patches to higher level features by multi-kernel learning methods. The structure of these kernel aggregating image patch features is similar with that of the kernel that aggregates pixel features. The multi-kernel function is constructed by different kernel descriptors with different weights, which replaces the original single kernel to enhance the applicability of the decision function and improve the kernel learning effect. As a feature space projection, multi-kernel function can be used to extract the effective features, which alleviates the sensitivity of the kernel method to parameters [9].

Weighted multi-kernel [10] integrates gradient, color and shape information by different weights, which is defined as follows:

$$K(\bar{P},\bar{Q})=dK_{grad}(\bar{P},\bar{Q})+rK_{col}(\bar{P},\bar{Q})+tK_{shape}(\bar{P},\bar{Q}) \tag{7}$$

where d, r and t are the weights of gradient, color and shape features, respectively. d, r and t vary within the interval [0, 1]. Then, gradient, color and shape kernel are written as:

$$
\begin{aligned}
K_{grad}(\bar{P},\bar{Q}) &= \sum_{A\in\bar{P}}\sum_{A^{'}\in\bar{Q}} \tilde{G}_A \tilde{G}_{A^{'}} k_o(T_A^{P-grad}, T_{A^{'}}^{P-grad}) k_s(S_A, S_{A^{'}}) \\
K_{col}(\bar{P},\bar{Q}) &= \sum_{A\in\bar{P}}\sum_{A^{'}\in\bar{Q}} \tilde{C}_A \tilde{C}_{A^{'}} k_c(T_A^{P-col}, T_{A^{'}}^{P-col}) k_s(S_A, S_{A^{'}}) \\
K_{shape}(\bar{P},\bar{Q}) &= \sum_{A\in\bar{P}}\sum_{A^{'}\in\bar{Q}} \tilde{H}_A \tilde{H}_{A^{'}} k_b(T_A^{P-shape}, T_{A^{'}}^{P-shape}) k_s(S_A, S_{A^{'}})
\end{aligned}
\tag{8}
$$

$$
\begin{aligned}
\tilde{G}_A &= G_A \Big/ \sqrt{\sum_{A\in\bar{P}} G_A^2 + \varepsilon_h} \\
\tilde{C}_A &= C_A \Big/ \sqrt{\sum_{A\in\bar{P}} C_A^2 + \varepsilon_h} \\
\tilde{H}_A &= H_A \Big/ \sqrt{\sum_{A\in\bar{P}} H_A^2 + \varepsilon_h}
\end{aligned}
\tag{9}
$$

where A and $A^{'}$ are the image patches. $\bar{P}$ and $\bar{Q}$ denote the image patch sets. ε_h is a small constant. G_A, C_A and H_A are the means of gradient, color and shape descriptors. T_A^{P-grad}, T_A^{P-col} and $T_A^{P-shape}$ are the descriptors which contain the spatial information embedded by spatial pyramid structure. Gaussian kernel $k_s(S_A, S_{A^{'}}) = \exp(-\gamma_s \| S_A - S_{A^{'}} \|^2)$ measures the spatial relationship between two image patches. S_A is the center of the image patch A.

4. Multi-Classifier Fusion

According to the performance of different classifiers, multi-classifier fusion considers all outputs of all classifiers. Multi-classifier fusion generally consists of three steps [11]. First, some classifiers are trained on the same training set for learning the sub-model. Then, the sample is assigned a label from different sub-model. Finally, through a combination of sub-models, the final category are obtained by label fusion. For classifier fusion, the outputs of different classifiers are fused to obtain a final label. The type of output of classifier is a key factor, which is directly related to the design and selection of the fusion method. Common output types include [12]: abstract level, sort level and metric level. For abstract level, the output is only a class label. In sort level, the output is the ordered possibility belonging to specific label, and metric level outputs a series of measures. Voting is used for the abstract and sort level. Voting principle is that sub-classifier predicts the label of a sample and votes for the label. Then, the majority of the votes of the class is the final label of the sample, which is the simplest classifier fusion method [13]. But, the voting principle does not take into account the different performance of the sub-classifier. Therefore, the voting rules cannot naturally reflect the performance of the accurate classifiers [14]. Thus, an enhanced version, soft voting, is proposed, whose flowchart is shown in Figure 1.

There are two principles for soft voting. The first is that the majority decision is better than a single decision. The other is the good classifier is better than the worse classifier. In this paper, three sub-classifiers, KNN, linear SVM and Bayesian classifier, are selected. Research [15] shows that results of SVM classifier is better than other classifiers based on maximum likelihood. Therefore, SVM determines the final prediction labels, when the classification results of the three classifiers are not consistent. Finally, if results of the three sub-classifiers classification are p, the final prediction results of samples are p; the final results of samples will be p according to the principle of majority, if the results of two sub-classifiers are p; the final prediction result depends on the results of SVM, if three sub-classifiers classification results are not the same. Compared with the traditional majority voting, soft voting combines label information and confidence coefficients.

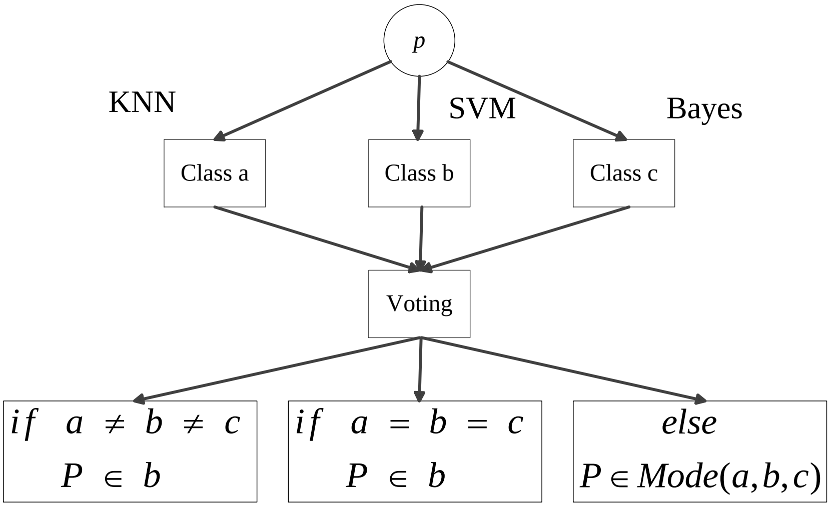

p
KNN
SVM
Bayes
Class a
Class b
Class c
Voting
$if \ a \neq b \neq c$ $P \in b$
$if \ a = b = c$ $P \in b$
$else$ $P \in Mode(a,b,c)$

Figure 1. Flowchart of Soft Voting

5. Experimental Results

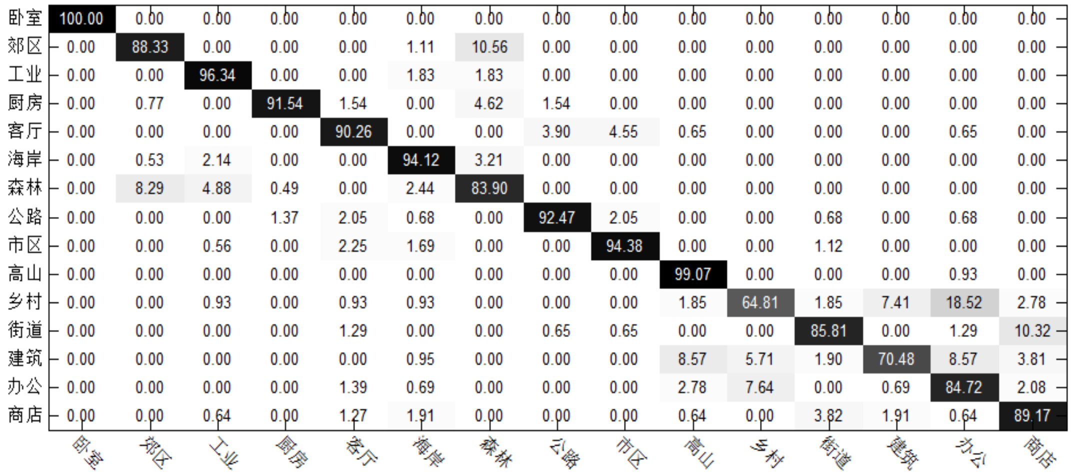
To verify the effectiveness of the proposed in this paper, two experiments are conducted on two image sets, Scene15 and Corel10. The overall accuracy and accuracy of single class are computed. Besides, three classical methods, ScSPM [16], EMK [5] and KSRSPM [17] are selected for comparison.

The parameter settings in [18] are used, where the gradient operator [-1, 0, 1] is employed to extract the gradient features of each image. Kernel descriptor is extracted from the image patch with size 8*8. Pyramid matching kernel utilizes the radial basis function, and three level Pyramid structure describes the geometry characteristics of the spatial information, where the weights of three layers are set as 0.001, 0.01, 0.1, respectively. For each data set, the mean accuracy of 5 independent classification trials are computed for comparison and training images are selected randomly. Three sub-classifiers, KNN, linear SVM and Bayesian classifier, are used in classifiers fusion.

5.1. Datasets

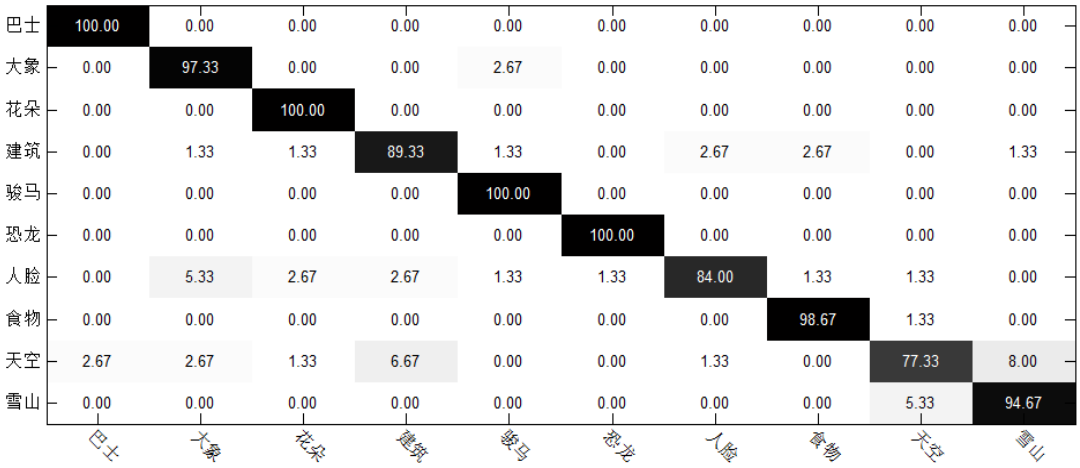
Scene15 data set contains 15 categories with total 4485 images. For fair comparison, 25% images are randomly selected in each category as the training set, which is the same as the setting of [12] and the rest images are regarded as a test set. Figure 2, reports the confusion matrix, where the classification accuracy and the false classification are given. In the confusion matrix, the diagonal values are the correct classification accuracy for the each class.

Corel10 contains 10 categories, such as bus, dinosaur, crowd, beach, construction, flowers, etc. The total number of images is 1000. 25 images randomly selected is used as training sets, and the rest of the images are test sets. Figure 3, shows the confusion matrix of the data set.



	卧室	郊区	工业	厨房	客厅	海岸	森林	公路	市区	高山	乡村	街道	建筑	办公	商店
卧室	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
郊区	0.00	88.33	0.00	0.00	0.00	1.11	10.56	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
工业	0.00	0.00	96.34	0.00	0.00	1.83	1.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
厨房	0.00	0.77	0.00	91.54	1.54	0.00	4.62	1.54	0.00	0.00	0.00	0.00	0.00	0.00	0.00
客厅	0.00	0.00	0.00	0.00	90.26	0.00	0.00	3.90	4.55	0.65	0.00	0.00	0.00	0.65	0.00
海岸	0.00	0.53	2.14	0.00	0.00	94.12	3.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
森林	0.00	8.29	4.88	0.49	0.00	2.44	83.90	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
公路	0.00	0.00	0.00	1.37	2.05	0.68	0.00	92.47	2.05	0.00	0.00	0.68	0.00	0.68	0.00
市区	0.00	0.00	0.56	0.00	2.25	1.69	0.00	0.00	94.38	0.00	0.00	1.12	0.00	0.00	0.00
高山	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	99.07	0.00	0.00	0.00	0.93	0.00
乡村	0.00	0.00	0.93	0.00	0.93	0.93	0.00	0.00	0.00	1.85	64.81	1.85	7.41	18.52	2.78
街道	0.00	0.00	0.00	0.00	1.29	0.00	0.00	0.65	0.65	0.00	0.00	85.81	0.00	1.29	10.32
建筑	0.00	0.00	0.00	0.00	0.00	0.95	0.00	0.00	0.00	8.57	5.71	1.90	70.48	8.57	3.81
办公	0.00	0.00	0.00	0.00	1.39	0.69	0.00	0.00	0.00	2.78	7.64	0.00	0.69	84.72	2.08
商店	0.00	0.00	0.64	0.00	1.27	1.91	0.00	0.00	0.00	0.64	0.00	3.82	1.91	0.64	89.17

Figure 2. Confusion Matrix on the Scene15 Dataset (%)



	巴士	大象	花朵	建筑	骏马	恐龙	人脸	食物	天空	雪山
巴士	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
大象	0.00	97.33	0.00	0.00	2.67	0.00	0.00	0.00	0.00	0.00
花朵	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
建筑	0.00	1.33	1.33	89.33	1.33	0.00	2.67	2.67	0.00	1.33
骏马	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
恐龙	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
人脸	0.00	5.33	2.67	2.67	1.33	1.33	84.00	1.33	1.33	0.00
食物	0.00	0.00	0.00	0.00	0.00	0.00	0.00	98.67	1.33	0.00
天空	2.67	2.67	1.33	6.67	0.00	0.00	1.33	0.00	77.33	8.00
雪山	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	5.33	94.67

Figure 3. Confusion Matrix on the Corel 10 Dataset (%)

5.2. Classification Results

The experimental results of ScSPM [9], EMK [10], KSRSPM [17] and the proposed method on the two data sets are shown in Table 1. We can see that the classification accuracy of the proposed method is improved compared to the other three methods.

Table 1. Average Classification Rates of 4 Different Methods on 2 Datasets

Methods	Scene15	Corel10
ScSPM	79.53	84.30
EMK	77.89	79.90
KSRSPM	83.12	88.40
Proposed method	88.58	94.13

The proposed method behaves well in classification due to the use of hierarchical matching kernel, which combines the low-level features and better captures image semantic information. Moreover, the multi-classifier fusion can effectively improve the classification performance for the imbalanced data.

6. Conclusions

In this paper, hierarchical multi-kernel is proposed by using image features. First of all, kernel descriptors are obtained by projecting the image features into a new feature space, which takes into account the location information of the kernel descriptors in the image. Then, the multi-kernel learning algorithm is used on kernel descriptors to better capture the image semantic information. In the application of computer vision, the combination of multi-features is better than a single feature, which can achieve better results. Besides, the complexity of classification is reduced through the use of multiple classifier fusion, which effectively improves the classification performance of the proposed method. Finally, experiments on the Corel10 and Scene15 databases show that the proposed method can achieve good classification results.

Acknowledgments

This work is funded by the Scientific Research Projects of Xianyang Normal University (No. 13XSYK058), the Science Basic Research Program in Shannxi Province of China(No. 16JK1823)

References

[1] Y. R. Yeh, T. C. Lin, Y. Y. Chung and Y. C. F. Wang, "A Novel Multiple Kernel Learning Framework for Heterogeneous Feature Fusion and Variable Selection", Multimedia, vol. 14, no. 3, **(2012)**.

[2] W. J. Lee, S. Verzakov and R. P. W. Duin, "Kernel Combination Versus Classifier Combination", International Workshop on Multiple Classifier Systems; Czech Republic, **(2007)** May 23-25.

[3] M. Sebban, D. Muselet, E. Fromont and B. Fernando, "Discriminative feature fusion for image classification", Computer Vision and Pattern Recognition, vol. 157, no. 10, **(2012)**.

[4] S. S. Keerthi and C. J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel", Neural computation, vol. 7, no. 15, **(2003)**.

[5] N. Dalal, B. Triggs and C. Schmid, "Histogram of Oriented Gradient Object Detection Toolkit. INRIA, **(2005)**.

[6] T. Ojala and I. Harwood, "A Comparative Study of Texture Measures with Classification Based on Feature Distributions", Pattern Recognition, vol. 29, no. 1, **(1996)**.

[7] T. Ojala and M. PietikaÈinena, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns", Pattern Analysis and Machine Intelligence, vol. 24, no. 7, **(2007)**.

[8] S. Lazebnik, C. Schmid and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories", IEEE Computer Society Conference on Computer Vision & Pattern Recognition, **(2006)**.

[9] G. C. Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification", IEEE Transactions on Geoscience and Remote Sensing Letters, vol. 3, no. 1, **(2006)**.

[10] J. Yang, Y. Tian, L. Y. Duan, T. Huang and W. Gao, "Group-sensitive multiple kernel learning for object recognition", IEEE Transactions on Image Processing, vol. 21, no. 5, **(2012)**.

[11] W. B. Langdon, S. J. Barrett and B. F. Buxton, "Combining Decision Trees and Neural Networks for Drug Discovery", Lecture Notes in Computer Science, vol. 2278, **(2002)**.

[12] Y. C. Tzeng, S. H. Chiu, D. Chen and K. S. Chen, "Multisource remote sensing images classification/ data fusion using a multiple classifiers systemweighted by a neural decision maker", Geoscience and Remote Sensing Symposium, **(2007)**.

[13] C. G. Li and G. F. Shao, "Combination multiclassifier for object-oriented classification of forest cover", Journal of Nanjing Forestry University, vol. 34, **(2010)**.

[14] L. Lam and C. Y. Suen, "Application of majority voting to pattern recognition:An analysis of its behavior and performance", IEEE Trans. Syst., Man, Cybern, vol. 27, no. 5, **(1997)**.

[15] S. Fukuda and H. Hirosawa, "Polarimetric SAR image classification using support vector machine", IEICE Transactions on Electronics, vol. 12, **(2001)**.

[16] J. C. Yang, K. Yu, Y. H. Gong and T. Huang, "Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification", IEEE Conference on Computer Vision and Pattern Recognition, **(2009)**.

[17] S. H. Gao, W. H. Tang and L. T. Chai, "Kernel Sparse Representation for Image Classification and Face Recognition", Proc. of the 11th European Conference on Computer Vision, vol. 6314, **(2010)**.

[18] Y. Huang, Z. Wu, L. Wang and T. Tan, "Feature Coding in Image Classification: A Comprehensive Study", IEEE Transactions on Pattern Analysis and Machine Intelligence, no. 3, vol. 36, **(2014)**.

Author

Fenxia Wu, **s**he received the Master degree in computer applications technology (2007) from EAST CHINA NORMAL University. Now she is a lecturer at School of Computer Science, Xianyang Normal University. She is current research interests include image processing.