Pattern Recognition 47 (2014) 2940–2951

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Extended Spectral Regression for efficient scene recognition

Liyuan Li*, Weixun Goh, Joo Hwee Lim, Sinno Jialin Pan

Institute for Infocomm Research, Singapore 138632, Singapore

ARTICLE INFO

Article history:
Received 5 October 2012
Received in revised form
1 October 2013
Accepted 18 March 2014
Available online 27 March 2014

Keywords:
Spectral Regression
Subspace learning
Machine learning
Image classification
Scene recognition
Computer vision

ABSTRACT

This paper proposes a novel method based on Spectral Regression (SR) for efficient scene recognition. First, a new SR approach, called Extended Spectral Regression (ESR), is proposed to perform manifold learning on a huge number of data samples. Then, an efficient Bag-of-Words (BOW) based method is developed which employs ESR to encapsulate local visual features with their semantic, spatial, scale, and orientation information for scene recognition. In many applications, such as image classification and multimedia analysis, there are a huge number of low-level feature samples in a training set. It prohibits direct application of SR to perform manifold learning on such dataset. In ESR, we first group the samples into tiny clusters, and then devise an approach to reduce the size of the similarity matrix for graph learning. In this way, the subspace learning on graph Laplacian for a vast dataset is computationally feasible on a personal computer. In the ESR-based scene recognition, we first propose an enhanced low-level feature representation which combines the scale, orientation, spatial position, and local appearance of a local feature. Then, ESR is applied to embed enhanced low-level image features. The ESR-based feature embedding not only generates a low dimension feature representation but also integrates various aspects of low-level features into the compact representation. The bag-of-words is then generated from the embedded features for image classification. The comparative experiments on open benchmark datasets for scene recognition demonstrate that the proposed method outperforms baseline approaches. It is suitable for real-time applications on mobile platforms, *e.g.* tablets and smart phones.

1. Introduction

With the proliferation of mobile computing devices such as smart phones and tablets, there is an increasing demand for real-time vision algorithms for scene recognition and image classification on these platforms. Potential applications include wearable egocentric systems [1], mobile robots [2], and tourist information access [3]. With limited memory and computational resources, efficient high performance algorithms will be core technologies for these emerging fields.

In the past decade, scene classification has attracted much attention from researchers in computer vision and pattern recognition. Recent progress has shown that approaches based on bag-of-words can achieve impressive performance [4–10]. In BOW-based methods, the first step is clustering the local visual features into small groups as codewords, *i.e.* the bag-of-words. Each image is then represented as a histogram of the bag-of-words. The next step is applying a learning model on the histogram representation for classification. The conventional BOW-based approach is simple and effective, but its performance is not satisfactory on challenging datasets. Most recent efforts focus on the extension of BOW-based representation for improving performance on a few challenging benchmark datasets. The successful approaches include Spatial Pyramid Matching (SPM) to exploit spatial information [5], various coding and pooling techniques to improve discriminative power [11], and advanced classifiers for accurate classification [12]. The introduction of these techniques has significantly improved performance on the benchmark datasets. However, it also comes at the expense of great increases in memory requirements and computational costs.

Techniques of data dimensionality reduction are frequently used to improve the efficiency of complicated algorithms based on high-dimensional features. By reducing the dimensionality of features, both memory size and computation time required by the algorithm can be reduced greatly. Recently, Spectral Regression (SR), a general framework based on graph learning for dimensionality reduction, has achieved better performance than the conventional approaches of PCA (Principal Component Analysis) and LDA (Linear Discriminant Analysis) in many applications [13,14]. These recent progresses motivate us to apply SR to improve the efficiency of scene recognition. To apply spectral regression to a real-world problem, one has to solve the eigen-problem on an $m \times m$ matrix, where m is the number of samples. It becomes prohibitive to directly apply it to embed local image features, *e.g.* dense SIFT

* Corresponding author. Tel.: +65 64082509; fax: +65 67761378.
E-mail address: lyli@i2r.a-star.edu.sg (L. Li).

http://dx.doi.org/10.1016/j.patcog.2014.03.012
0031-3203/

descriptors, for image classification since there are typically over millions of dense local features extracted from all the training images. Solving such an eigen-problem on a personal computer is still infeasible now since more than 1TB memory space is required.

In this paper, a novel approach to apply SR to encapsulate low-level image features for efficient scene recognition is proposed. First, a new SR approach, called Extended Spectral Regression (ESR), is proposed particularly for feature dimensionality reduction on a vast dataset which contains a huge number of data samples. Based on ESR, we propose a novel method to map the low-level image features into an embedded manifold subspace for efficient BOW-based image classification. We propose an enhanced low-level feature representation which combines the local appearance descriptor and its spatial, scale and orientation information into a concise representation. ESR is then applied to encapsulate the enhanced low-level features into a low dimensional manifold subspace. The Bag-of-Words is then generated on the embedded manifold subspace for image classification. With the powerful manifold learning, it is possible to pull related local features of the same class closer and push the local features from different image classes apart in the manifold subspace. Hence, the BOW generated on the manifold subspace will have better descriptive power for image representation.

Our method not only generates lower dimension visual words but also integrates various aspects of the low-level feature into the compact representation. Therefore, we can obtain an effective image representation of much lower dimension while achieving better results compared with PCA and SPM. We have evaluated our method on two challenging datasets for scene recognition, namely Scene-15 and UIUC-Sports. We also implemented our method with OpenCV for easy deployment and tested on a new indoor scene dataset. The memory requirement and computational cost indicate that our technique is suitable for deployment on portable computing devices.

1.1. Related works

To achieve efficient image classification for real-time tasks, it is desirable to reduce the dimension of image features, or visual words. This speeds up the computation of histogram representation and reduces the memory requirements for visual vocabulary. PCA has been applied on SIFT (Scale-Invariant Feature Transform) and HOG (Histogram of Oriented Gradients) features. In some papers [15,16], almost no loss of performance has been observed when PCA is used, while in other papers [17], it is found that PCA degrades performance considerably. In this paper, we propose to use SR to reduce the dimension of low-level image features for efficient scene recognition since recent papers have shown the superiority of SR over PCA and LDA in many applications [13,14]. As mentioned above, SR cannot be applied directly to the dataset of low-level image features since there may be over millions of local features extracted from all training images. Hence, we propose the ESR for this purpose.

Since its introduction [18,19], the bag-of-words model has become the most effective representation for image classification. Subsequent research has focused on extensions of BOW representation to achieve improved performance on challenging benchmark datasets. The progresses are achieved along two general directions, namely (a) enhanced encoding and (b) spatial layout representation.

In usual BOW methods, simple hard histogram encoding is used where a local feature is assigned to the nearest visual word in the vocabulary. On the bag-of-words generated by k-means, enhanced encoding techniques have been proposed. In [16], Gemert et al. proposed to replace hard quantization by soft quantization, or kernel codebook encoding. Using soft assignment, a local feature may be assigned to a few closer visual words in the dictionary with weights within [0,1] according to its distance to the words. Sparse coding approaches are further proposed to improve the soft assignment [11,20,21]. Sparse coding uses a linear combination of a small number of visual words to approximate the local feature, and the coefficients of projecting the feature down to the local linear subspace spanned by the set of visual words are pooled in the histogram. An appealing encoding approach, Fisher encoding, has been proposed for image classification recently [15,22]. With a dictionary (BOW) of K words, Fisher encoding captures the average first and second order differences between the local features and the visual words on a learned GMM (Gaussian Mixture Models) model for the codewords. Hence, it leads to an extended image representation of $K(2D+1)$ dimensions where D represents the feature dimension. Improvements over state-of-the-arts have been obtained through these enhanced encoding techniques, in particular the Fisher encoding. However, image representation is greatly extended and extra computation is required for feature encoding.

In the basic BOW framework, the image representation is a frequency histogram of quantized local features, where the spatial layout of the local features is completely ignored. Clearly, spatial information of low-level features is useful since the compositions of particular visual objects and context regions typically share common spatial layout properties. Various approaches to encode spatial information in BOW representation have been explored [10,23,24]. Among them, the most effective way is to extend the basic BOW representation by using Spatial Pyramid Matching (SPM) [5]. SPM partitions the image into increasingly finer cells, up to 3 layers, and concatenates the BOW histograms of the cells. For a 3-layer pyramid, the image representation is extended to $\sum_{i=0}^{2} 4^i K = 21K$, where K is the dictionary size. The SPM strategy is used in most state-of-the-art approaches [11,15,17,20,21,25–27]. More recently, Krapac et al. [28] proposed to extend the BOW representation by using Fisher kernel to encode the spatial layout of visual words, which is represented by learned spatial MoG (Mixture of Gaussians, *i.e.* GMM) models. It can reduce the image representation from 64,500 dimensions by SPM to 13,300 dimensions and obtain comparable results. In our method, we propose a concise low-level feature representation which includes the spatial, scale, orientation, and appearance information of the local feature. Such enhanced descriptor is then mapped into a compact manifold subspace learned by ESR. Hence, there is no need to extend the BOW representation to encode the spatial information of local features.

It is worth to note that some sample selection methods, such as Editing and Condensing algorithms [29], also generate a compact sample set for classification. These methods aim at selecting a sufficiently small set of samples from the whole training set by removing outliers. The compact sample set can reduce the computational burden of classification. However, as reported in [30], discarding any features, even the most non-informative features will result in the deterioration of image classification performance. Different from these methods, our method embeds all the local image features into a compact and effective manifold subspace for efficient image classification.

1.2. Contributions

Our method is illustrated by Fig. 1, where the gray blocks indicate the novel parts. It contains two stages, *i.e.* Training Stage and Testing Stage. There are three steps in the Training Stage, as illustrated by the three columns of blocks from the left to the right in the figure. In the first step, we first cluster the huge number of enhanced low-level local features from all training

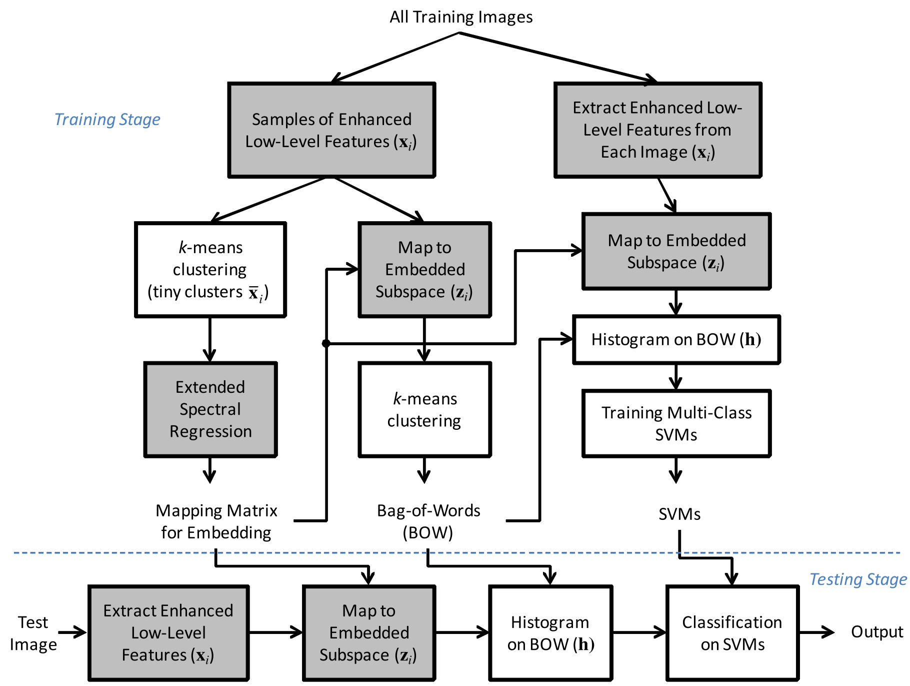

Fig. 1. The block diagram of the proposed method.

images into tiny clusters using k-means clustering, and then apply ESR on the tiny clusters to learn a Mapping Matrix which embeds the low-level local features into a compact manifold subspace. We derive a reduced similarity matrix for graph learning on the tiny clusters. Since the samples within each tiny cluster may have different semantic labels, we propose a novel approach to compute the weights for the similarity matrix which combines the local distance and semantic labels of the features within each cluster. In this way, SR on a vast dataset can be performed on a personal computer. In the second step, we first map all the low-level features into the manifold subspace, and then apply k-means clustering to learn a dictionary, or Bag-of-Words, on the embedded visual features. In the last step, for each training image, the low-level features are extracted and mapped into the manifold subspace, and then a histogram of the embedded features on the Bag-of-Words is generated. Multiple SVM classifiers are learned using these histogram representations and image labels. In the end of Training Stage, we obtain a Mapping Matrix for feature embedding, a Bag-of-Words for global-level image representation, and multiple SVMs for image classification. In the Testing Stage, for each incoming new image, the enhanced low-level features are extracted first, they are mapped into the compact manifold subspace by the Mapping Matrix, a histogram of these embedded features on the Bag-of-Words is then generated and fed to the multiple SVMs for classification.

The two important contributions of our paper are summarized as below:

(1) A new Spectral Regression approach, Extended Spectral Regression (ESR), for SR-based manifold learning on a large dataset.
(2) A novel method based on ESR for efficient scene recognition which uses ESR to effectively embed low-level image features for BOW-based image classification.

The rest of this paper is organized as follows. Section 2 presents the proposed method, including a brief description of Spectral Regression, the novel Extended Spectral Regression and its application for efficient scene recognition. The experimental results are reported in Section 3. Finally, Section 4 concludes this paper.

2. The method

2.1. Background: spectral regression

Spectral Regression is a powerful computational approach for manifold learning, clustering, and dimensionality reduction [13,31]. Given m samples $\{\mathbf{x}_i\}_{i=1}^m \subset \mathcal{R}^q$ ($\mathbf{x}_i$ is a feature vector representing the sample point), using spectral regression, the sample points can be mapped into an embedded manifold subspace represented as $\{\mathbf{z}_i\}_{i=1}^m \subset \mathcal{R}^d$, $d \ll m$ and $d < q$, where the statistical or geometric properties of the dataset are preserved and the discriminant ability for classification can be enhanced [31].

The mapping is learned in a general graph embedding framework. From the training samples, one can construct a weighted graph G with m nodes. Let W be an $m \times m$ symmetric similarity matrix with W_{ij} representing the weight of the edge connecting the ith and jth nodes in G, and $\mathbf{y} = [y_1, y_2, \ldots, y_m]^T$ be the map from the graph to the real line. Consider the problem of mapping the graph G to a line so that strongly connected points (*i.e.* similar or related points) stay as close together as possible. The optimal $\mathbf{y}$ is obtained by minimizing

$$\sum_{i,j}(y_i - y_j)^2 W_{ij} = 2\mathbf{y}^T L \mathbf{y}, \tag{1}$$

where $L = D - W$ and D is a diagonal matrix with $D_{ii} = \sum_j W_{ji}$. Formally, this can be expressed as

$$\mathbf{y}^* = \arg \min_{\mathbf{y}^T D \mathbf{y} = 1} \mathbf{y}^T L \mathbf{y} = \arg \min \frac{\mathbf{y}^T L \mathbf{y}}{\mathbf{y}^T D \mathbf{y}} = \arg \max \frac{\mathbf{y}^T W \mathbf{y}}{\mathbf{y}^T D \mathbf{y}}. \tag{2}$$

This is equivalent to the maximum eigenvectors of eigen-problem

$$W\mathbf{y} = \lambda D \mathbf{y}. \tag{3}$$

Solving this problem, one can obtain d eigenvectors $\{\mathbf{y}_k\}$ corresponding to the d largest eigenvalues. According to other works on spectral learning [31], here, we skip the top eigenvector $\mathbf{y}_0$ since it corresponds to the eigenvector of the minimum eigenvalue 0 for the corresponding minimum eigen-problem $L\mathbf{y}=\lambda' D\mathbf{y}$.

For mapping all samples including new samples, a linear function $y_i=f(\mathbf{x}_i)=\mathbf{a}^T\mathbf{x}_i$ is chosen in [13]. This implicates an assumption that the linear function passing the center of embedded feature space, which may not be suitable for complex local image features. According to the standard approach in linear regression, we augment the feature vector $\mathbf{x}_i$ with the constant 1, that is $\mathbf{x}_i'=[\mathbf{x}_i^T,1]^T$. Then, the linear mapping function becomes $y_i=f(\mathbf{x}_i)=\mathbf{a}^T\mathbf{x}_i'$ with $\mathbf{a}\in\mathcal{R}^{q+1}$. The mapping vectors $\{\mathbf{a}_k\}_{k=1}^d$ can be obtained as the solution of the regularized least square problem

$$\mathbf{a}_k=\arg\min_{\mathbf{a}}\left(\sum_{i=1}^{m}(\mathbf{a}^T\mathbf{x}_i'-y_i^k)^2+\alpha\|\mathbf{a}\|^2\right), \tag{4}$$

where $k=1,\ldots,d$ and y_i^k is the ith element of $\mathbf{y}_k$. Expressing (4) as a linear system, $\mathbf{a}_k$ is the solution of

$$(XX^T+\alpha I)\mathbf{a}_k=X\mathbf{y}_k \tag{5}$$

where I is a $(q+1)\times(q+1)$ identity matrix. Let $A=[\mathbf{a}_1,\mathbf{a}_2,\ldots,\mathbf{a}_d]$ be the learned $(q+1)\times d$ transformation matrix, then new input sample $\mathbf{x}$ is embedded into the d dimensional manifold subspace by $\mathbf{x}\rightarrow\mathbf{z}=A^T\mathbf{x}'$. The linear function $y_i=f(\mathbf{x}_i)$ helps to generalize the embedding so as to predict embedding for out-of-sample examples without to retrain the embedding model, while out-of-sample problem is a typical problem for spectral embedding methods [32].

Different from PCA, SR maps the local image features into a manifold subspace where the distance measure could be greatly different from that on original feature space. Hence, with a suitable clustering of related features, it is possible to obtain better discriminative ability on the embedded manifold subspace than that on the original feature space.

2.2. Extended spectral regression

Suppose we have m training samples. To apply spectral regression to learn feature embedding, one has to solve the eigen-problem on an $m\times m$ matrix W, where $W_{ij}=d(\mathbf{x}_i,\mathbf{x}_j)$ describes the similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$. Existing methods applies spectral regression to learn the embedded manifold of global image representations, where the whole image is a sample represented by a high-dimensional feature vector $\mathbf{x}$, such as a histogram on bag-of-words. In this way, performing spectral regression on all training samples is feasible since in most cases there are only hundreds to thousands of images in training sets. In this paper, we investigate to apply spectral regression to learn the embedded manifold of low-level local features in images, such as SIFT or LBP (Local Binary Pattern) descriptors [33]. Typically, one can extract hundreds or thousands of low-level local features from an image. As an example, for an image of 320×240 pixels, one can obtain 1064 dense SIFT features on scale of 16×16 pixels [5]. If the training set consists of 1000 images, over one million dense SIFT local features are generated from all the training images (*i.e.* $m>1,000,000$). It becomes prohibitive to directly apply spectral regression to embed local image features from all training images. Solving such an eigen-problem on a personal computer is still infeasible now due to the limitation of memory space and computational cost.

Let $(\mathbf{x}_i,c_i)$ be a low-level local feature extracted from an image of class c_i, where $c_i=1,\ldots,L$ is the label of image category. If dense SIFTs are employed as local features, a sample $\mathbf{x}_i$ represents a feature vector of 128 dimensions ($q=128$), and over one thousand sample features are extracted from an image of 320×240 pixels [5]. We first group the low-level feature samples $\{\mathbf{x}_i\}_{i=1}^m$ from all training images into tiny clusters $\{\overline{\mathbf{x}}_k\}_{k=1}^n$ by using k-means clustering, where n can be selected from 3000–6000 (based on the available memory size of the computer) while m may be larger than 1,000,000. The low-level features are assigned to the clusters and the distortion of the features within one tiny cluster is small since n is quite large. In one experiment on the Scene-15 dataset [5], we group 1,440,284 normalized dense SIFT features from 1500 images of 15 classes into 3000 clusters. The smallest cluster only contains 1 samples, and the largest one contains up to 9872 samples. The samples in one tiny cluster may come from images of one class or all 15 classes.

To apply spectral regression to learn a better manifold subspace of low-level image features, we propose an approach to reduce the matrix $W_{m\times m}$ to matrix $\overline{W}_{n\times n}$ by merging the rows and columns belonging to the same tiny cluster. To be easy to understand, let us first to see how to merge two rows and two columns of the samples belonging to the same cluster. This is illustrated in Fig. 2. The upper-left figure shows the matrix $W_{m\times m}$ for all samples $\{\mathbf{x}_i\}_{i=1}^m$, where $W_{ij}=d(\mathbf{x}_i,\mathbf{x}_j)$. Suppose the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same tiny cluster $\overline{\mathbf{x}}_k$ and the sample $\mathbf{x}_p$ belongs to a different tiny cluster $\overline{\mathbf{x}}_l$. When we replace $\mathbf{x}_i$ and $\mathbf{x}_j$ with $\overline{\mathbf{x}}_k$ and $\mathbf{x}_p$ with $\overline{\mathbf{x}}_l$, the matrix becomes the upper-right version in Fig. 2, where W_{ii}, W_{ij}, W_{ji}, and W_{jj} turn into $W_{kk}=d(\overline{\mathbf{x}}_k,\overline{\mathbf{x}}_k)$ since $\mathbf{x}_i$ and $\mathbf{x}_j$ are replaced by $\overline{\mathbf{x}}_k$ in $d(\cdot,\cdot)$, and W_{ip}, W_{jp}, W_{pi}, and W_{pj} become $d(\overline{\mathbf{x}}_k,\overline{\mathbf{x}}_l)=W_{kl}$. Now, the ith and jth rows become identical and redundant, as well as the ith and jth columns. We can merge the two rows and two columns belonging to the same tiny cluster $\overline{\mathbf{x}}_k$ and the matrix $W_{m\times m}$ is reduced as $\overline{W}_{(m-1)\times(m-1)}$, as shown in the lower-left in Fig. 2, where the weight $\overline{W}_{kl}$ is the sum of the corresponding weights from the merged rows and columns in the similarity matrix $W_{m\times m}$ defined on the tiny clusters.

Extending the above procedure to the merger of all related rows and columns, one can obtain the reduced similarity matrix $\overline{W}_{n\times n}$. Suppose that n_k and n_l are the numbers of samples belonging to the clusters $\overline{\mathbf{x}}_k$ and $\overline{\mathbf{x}}_l$, respectively. By merging n_k rows of samples belonging to the cluster $\overline{\mathbf{x}}_k$ and n_l columns of samples belonging to the cluster $\overline{\mathbf{x}}_l$ in the original similarity matrix $W_{m\times m}$, one can obtain the weight $\overline{W}_{kl}$ in $\overline{W}_{n\times n}$ as

$$\overline{W}_{kl}=n_k n_l W_{kl},\quad k,l=1,\ldots,n, \tag{6}$$

where W_{kl} is defined on $d(\overline{\mathbf{x}}_k,\overline{\mathbf{x}}_l)$.

Obviously, one tiny cluster may contain low-level feature samples from images of different classes. According to related researches [31], to learn an effective mapping for feature embedding for classification, the similarity measure for samples from the same class should be defined differently with that for samples from different classes. Considering this factor, we can modify (6) as

$$\overline{W}_{kl}=\sum_{c_k=1}^{L}n_k^{c_k}\sum_{c_l=1}^{L}n_l^{c_l}W_{kl}(c_k,c_l)=\sum_{c_k,c_l=1}^{L}n_k^{c_k}n_l^{c_l}W_{kl}(c_k,c_l), \tag{7}$$

where $n_k^{c_k}$ is the number of low-level feature samples within the tiny cluster $\overline{\mathbf{x}}_k$ and coming from images of category c_k (*i.e.*, $\sum_{c_k=1}^{L}n_k^{c_k}=n_k$), and $W_{kl}(c_k,c_l)$ is defined on $d(\overline{\mathbf{x}}_k,\overline{\mathbf{x}}_l)$ as well as the class labels c_k and c_l, which will be described below.

In existing methods of spectral analysis, there are two ways to calculate the similarity weight $W_{ij}(c_i,c_j)$ [13,31,34]. The first one completely depends on the semantic labels of the samples, that is $W_{ij}(c_i,c_j)=1$ for $c_i=c_j$ and 0 otherwise. The second is heat kernel which depends only on the distance between two samples, whether $c_i=c_j$ or not. The first one is not applicable for our case since a tiny cluster may contain hundreds of low-level local features extracted from images of different classes, and the low-level local features from images of the same class may come from different parts in the images for different objects. The second is not

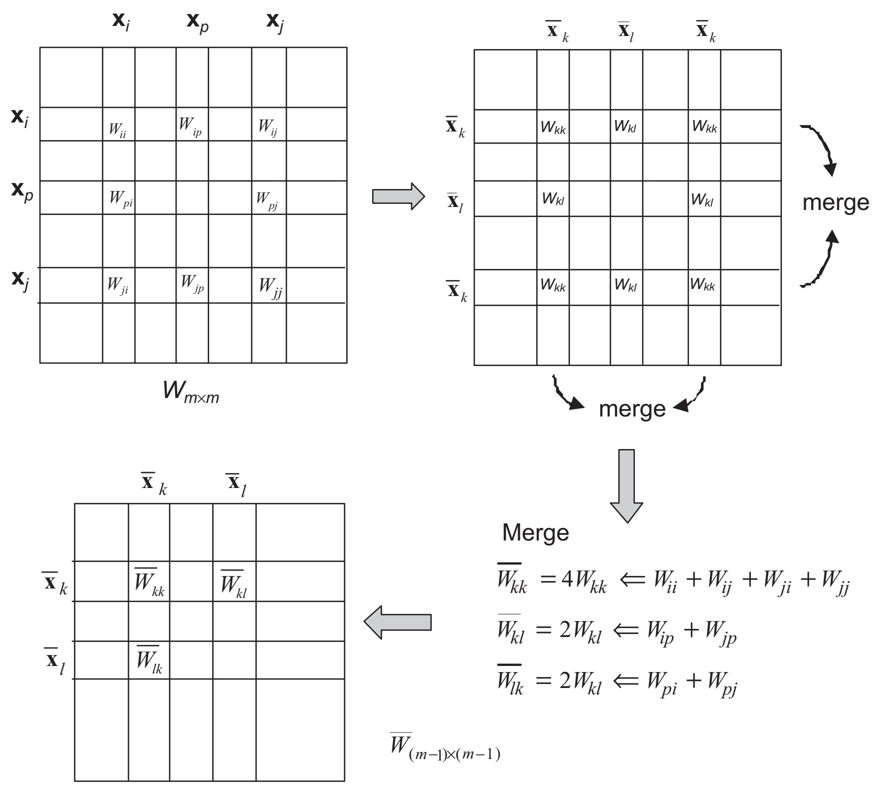

Fig. 2. The illustration of merging the rows and columns belonging to the same tiny cluster in the similarity matrix $W_{m\times m}$. The upper-left figure shows the matrix $W_{m\times m}$ for all samples $\{\mathbf{x}_i\}_{i=1}^m$, where $W_{ij} = d(\mathbf{x}_i, \mathbf{x}_j)$. Suppose the samples $\mathbf{x}_i$ and $\mathbf{x}_j$ belong to the same tiny cluster $\overline{\mathbf{x}}_k$ and the sample $\mathbf{x}_p$ belongs to a different tiny cluster $\overline{\mathbf{x}}_l$. When we replace $\mathbf{x}_i$ and $\mathbf{x}_j$ with $\overline{\mathbf{x}}_k$ and $\mathbf{x}_p$ with $\overline{\mathbf{x}}_l$, the matrix becomes the upper-right version, where W_{ii}, W_{ij}, W_{ji}, and W_{jj} turn into $W_{kk} = d(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_k)$, and W_{ip}, W_{jp}, W_{pi}, and W_{pj} become $d(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l) = W_{kl}$. Now, the ith and jth rows become identical and redundant, as well as the ith and jth columns. We can merge the two rows and two columns belonging to the same tiny cluster $\overline{\mathbf{x}}_k$ and the matrix $W_{m\times m}$ is reduced as $\overline{W}_{(m-1)\times(m-1)}$, as shown in the lower-left.

suitable for our task since, for classification purpose, we would like the similarity to be smooth and large for local features from the images of the same classes so that it can adapt to visual variance caused by lighting conditions, viewing angles, *etc.*, but small for local features from images of different classes. In the embedded manifold feature space, we wish to pull similar features from the images of the same class as close as possible but push features from images of different classes apart even though they may be quite similar. Hence, we propose to combine the semantic information and distance measure to compute the similarity weight between tiny clusters. The weight is defined as

$$W_{kl}(c_k, c_l) = \begin{cases} C_S d_S(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l) & \text{if } c_k = c_l, \\ C_D d_D(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l) & \text{if } c_k \neq c_l, \end{cases} \quad c_k, c_l = 1, \ldots, L, \tag{8}$$

where subscripts 'S' indicates 'same class' and 'D' indicates 'different classes', $C_S \gg C_D$, and $d_S(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l)$ is smoother than $d_D(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l)$. In this paper, to be adaptive to varying class numbers for different multi-class classification problems, $C_S = 1$ and $C_D = \min[0.2, 1/(L-1)]$ are chosen. Here, simply setting C_D to be 0 is not acceptable since the similar low-level image features could be shared by images of different classes, *e.g.*, the corners of windows may come from a building in an image of a street or a window in an image of a bedroom. The feature similarity measures $d_S(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l)$ and $d_D(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l)$ can be defined on feature's characteristics in different applications.

2.3. ESR-based efficient scene recognition

A low-level local image feature, such as a SIFT or HOG descriptor, is a coded representation of local visual pattern. In existing methods for scene recognition and image classification, the bag-of-words (visual dictionary) is directly built upon the local features. However, other aspects of a low-level local feature, such as the position, scale, and orientation, also provide important information of the image. In recent approaches, such information is represented separately from visual words, which results in combinatorial (or exponential) extension of image representation. If one can encode such information into the low-level feature representation, the visual words will have enhanced expressive power. It becomes possible to express the information on 'where' and 'what' a local feature appears in an image with a visual word. As an example, such a visual word can express the concept of 'a large right corner on the upper-left region' in the image. In addition, applying powerful subspace learning on such enhanced low-level features, one can obtain a compact representation of visual words which combine various aspects of low-level features. This will result in efficient high-performance scene recognition. For this purpose, we propose an enhanced low-level local feature to integrate the information of spatial position, scale, orientation, and local appearance into a concise feature vector.

Let $\mathbf{e}_i$ be a low-level local feature descriptor (*e.g.* SIFT) extracted at point (x_i, y_i) with scale S_i and orientation O_i from an image, and the size of the image be $M \times N$ pixels. The scale measure can be normalized as $s_i = S_i/(\max(M, N)/2)$ since some open source software (*e.g.* OpenCV) can extract the SIFT feature of half image size. The orientation measure can be normalized as $o_i = (O_i/180)*\pi$. A set of 9 spatial reference points in the image is used to encode the spatial measurement of a local feature, as shown in Fig. 3. The spatial reference points (blue dots in Fig. 3) are 3×3 grid points evenly distributed in the image with offsets $M/4$ and $N/4$ to each other and the image boundaries on the horizontal and vertical directions, respectively. The spatial reference points can be

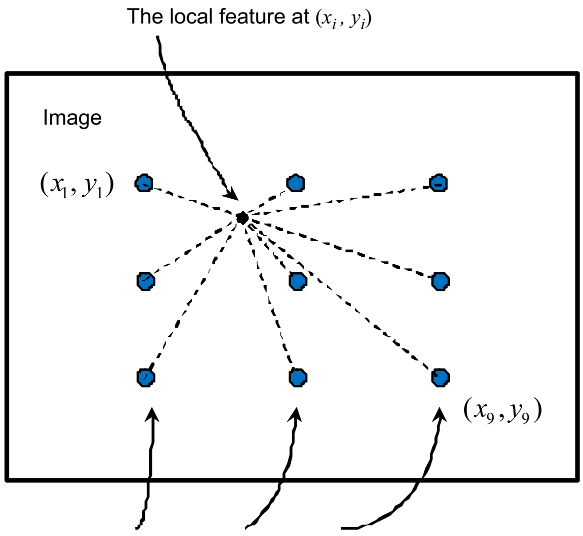

Fig. 3. The illustration of encoding the spatial information of a low-level local feature. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

expressed as $(x_k, y_k), k = 1, \cdots, 9$. Then, the spatial information of the low-level local feature at (x_i, y_i) (black dot in Fig. 3) can be encoded according to its distances to the 9 spatial reference points. It can be expressed as a vector $\mathbf{r}_i = [r_1, \ldots, r_9]^T$, where

$$r_k = \exp\left[-\left(\frac{(x_i - x_k)^2}{\sigma_x^2} + \frac{(y_i - y_k)^2}{\sigma_y^2}\right)\right] \tag{9}$$

with $\sigma_x = M/2$ and $\sigma_y = N/2$ since the scale of a SIFT feature could be half the size of the image. Compared with the widely used spatial pyramid strategy, one advantage of our approach for encoding the spatial information of a local feature is that it avoids the hard divisions of image regions. It provides a soft and concise coding of the spatial information. Combining the aspects of spatial position, scale, orientation, and visual appearance of a local visual feature, an enhanced low-level local feature can be obtained as $\mathbf{x}_i = [s_i, o_i, \mathbf{r}_i^T, \mathbf{e}_i^T]^T$, where, in this paper, $\mathbf{e}_i$ denotes a dense SIFT or a salient SIFT feature.

Obviously, combining various aspects of a local visual feature into an enhanced low-level feature can increase the expressive power of the visual words. However, simply generating the bag-of-words directly on the enhanced low-level features may result in the requirement of a great number of words for the dictionary due to the effect of combinatorial explosion of involved information. In addition, the simple and direct method might not be able to well attribute the relative importance of different feature aspects since their scales and lengths in the enhanced low-level features are different, *e.g.*, the length of $\mathbf{e}_i$ is much larger than s_i and o_i. Hence, we propose to use ESR to embed the enhanced low-level features for generating compact visual words (*i.e.* low-dimension features) and an effective bag-of-words of limited size (*i.e.* small-size dictionary) for scene recognition.

To apply ESR to the training dataset of enhanced low-level local features from all training images, we have to define the way to compute the similarities $d_S(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l)$ and $d_D(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l)$ in (8) since the enhanced low-level feature combines various aspects of a local feature. Let $\overline{\mathbf{x}}_k = (\overline{s}_k, \overline{o}_k, \overline{\mathbf{r}}_k^T, \overline{\mathbf{e}}_k^T)$ represent a tiny cluster of the enhanced low-level features. The measurements of feature's various aspects can be assumed as independent with each other. Then, the similarity measures between two tiny clusters of enhanced low-level features are defined as

$$\begin{cases} d_S(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l) = d_S(\overline{\mathbf{e}}_k, \overline{\mathbf{e}}_l) d_S(\overline{\mathbf{r}}_k, \overline{\mathbf{r}}_l) d_S(\overline{s}_k, \overline{s}_l) d_S(\overline{o}_k, \overline{o}_l), \\ d_D(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l) = d_D(\overline{\mathbf{e}}_k, \overline{\mathbf{e}}_l) d_D(\overline{\mathbf{r}}_k, \overline{\mathbf{r}}_l) d_D(\overline{s}_k, \overline{s}_l) d_D(\overline{o}_k, \overline{o}_l), \end{cases} \tag{10}$$

where the similarity measures for feature's each aspect are defined as

$$\begin{cases} d_S(\overline{\mathbf{e}}_k, \overline{\mathbf{e}}_l) = \exp\left(-\dfrac{\|\overline{\mathbf{e}}_k - \overline{\mathbf{e}}_l\|^2}{(\rho\sigma_e)^2}\right) \\ d_S(\overline{\mathbf{r}}_k, \overline{\mathbf{r}}_l) = \exp\left(-\dfrac{\|\overline{\mathbf{r}}_k - \overline{\mathbf{r}}_l\|^2}{(\rho\sigma_r)^2}\right) \\ d_S(\overline{s}_k, \overline{s}_l) = \exp\left(-\dfrac{(\overline{s}_k - \overline{s}_l)^2}{(\rho\sigma_s)^2}\right) \\ d_S(\overline{o}_k, \overline{o}_l) = \exp\left(-\dfrac{(\overline{o}_k - \overline{o}_l)^2}{(\rho\sigma_o)^2}\right) \end{cases}, \quad \begin{cases} d_D(\overline{\mathbf{e}}_k, \overline{\mathbf{e}}_l) = \exp\left(-\dfrac{\|\overline{\mathbf{e}}_k - \overline{\mathbf{e}}_l\|^2}{\sigma_e^2}\right) \\ d_D(\overline{\mathbf{r}}_k, \overline{\mathbf{r}}_l) = \exp\left(-\dfrac{\|\overline{\mathbf{r}}_k - \overline{\mathbf{r}}_l\|^2}{\sigma_r^2}\right) \\ d_D(\overline{s}_k, \overline{s}_l) = \exp\left(-\dfrac{(\overline{s}_k - \overline{s}_l)^2}{\sigma_s^2}\right) \\ d_D(\overline{o}_k, \overline{o}_l) = \exp\left(-\dfrac{(\overline{o}_k - \overline{o}_l)^2}{\sigma_o^2}\right) \end{cases}. \tag{11}$$

In (11), $\rho \geq 2$, which means the bandwidths of Gaussian kernels for the same classes are larger than those for different classes, *i.e.*, the kernel functions for the same classes are smoother and wider than those for different classes. This is based on the following considerations. For the similar local features from images of the same scene class, there is a high chance that they come from the similar objects in the images. As an example, in the images of the street scene, the local features from the wheels of vehicles are similar. However, due to the variations of view angles, scales, and lighting conditions, the local features from similar objects in the images of the same scene class might vary from image to image. To be able to learn such variations for the local features from similar objects, it is better to choose large bandwidths for wider kernel functions. On the other hand, for the local features from images of different scene classes, there is a high chance that they come from different objects since, in many cases, the visual objects in images of different scenes are different. For examples, we can observe high buildings in images of city, while in the images of countryside, what we often observe are the wild fields, farms, and forests. Of course, there are local features from similar objects in images of different scenes, *e.g.*, the corner features from windows in images of both street and bedroom. Hence, for different classes, it is better to select small bandwidths for tighter peaked Gaussian kernels. The benefits of this choice are twofold. First, it can cluster similar low-level features from images of the same classes as close as possible even though there are quite large variations, so that it enhances the robustness to image variations caused by lighting conditions, viewing angles, and variance of similar objects in the same class of scenes. On the other hand, it can separate the low-level features from images of different classes far apart even though the difference is small, which can enhance the discriminative power for image classification. In addition, computing the weight using (7), (8), (10), and (11) is also helpful for clustering shared features of a few classes in the embedded feature subspace. Suppose the local features of clusters $\overline{\mathbf{x}}_k$ and $\overline{\mathbf{x}}_l$ are similar and shared for classes c_1 and c_2, and they rarely appear in the images of the other classes. Then, $d_S(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l)$ is large, and $n_k^{c_1}$, $n_k^{c_2}$, $n_l^{c_1}$ and $n_l^{c_2}$ are large, while the numbers for the rest classes are very small. According to (7) and (8), $\overline{W}_{kl}$ can be expressed as

$$\overline{W}_{kl} = \overline{W}_{Large} + \overline{W}_{Small}, \tag{12}$$

where

$$\begin{aligned} \overline{W}_{Large} &= C_S(n_k^{c_1} n_l^{c_1} + n_k^{c_2} n_l^{c_2}) d_S(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l) + C_D(n_k^{c_1} n_l^{c_2} + n_k^{c_2} n_l^{c_1}) d_D(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l) \\ \overline{W}_{Small} &= \sum_{c_k, c_l} n_k^{c_k} n_l^{c_l} W_{k,l}(c_k, c_l) \quad \text{with } c_k, c_l = 1, \ldots, L \text{ and } c_k, c_l \neq c_1, c_2 \end{aligned} \tag{13}$$

since $\overline{W}_{Large}$ is quite large, local features of clusters $\overline{\mathbf{x}}_k$ and $\overline{\mathbf{x}}_l$ would be mapped to close positions in the manifold subspace. This would result in a compact and effective bag-of-words for robust multi-class classification.

When applying Extended Spectral Regression on the dataset $\{\overline{\mathbf{x}}_k\}_{k=1}^n$ with $\overline{W}_{kl}$, one can obtain the mapping matrix A^T which can be used to map an enhanced low-level feature $\mathbf{x} \subset \mathcal{R}^q$ to the embedded manifold subspace $\mathbf{z} \subset \mathcal{R}^d$ since the distortion of the extended low-level features within a tiny cluster is small. Image classification can be performed under the conventional BOW framework on the embedded features, due to the effectiveness of BOW representation for image classification and the simplicity of it for real-time tasks. First, a bag-of-words on the embedded features $\{\mathbf{z}_i\}_{i=1}^m$ is generated by using k-means clustering. The size of the bag-of-words, or the number of visual words in the dictionary, can be denoted as K. Then, the histogram on the BOW for each image is employed as image representation for image classification. The histogram can be represented as a vector $\mathbf{h}$ of K dimensions. A SVM is trained for each image category with the 1-vs-rest rule, and a test image is assigned the label of the classifier with the highest response.

There are two benefits of applying ESR to BOW-based scene recognition. First, with the bag-of-words on the embedded subspace, the dimension of the visual word has been reduced to about half the size of the original low-level feature representation (*i.e.* $d \approx q/2$), which reduces the memory space for visual words and speeds up the computation for generating histogram representation of an image. In this paper, the enhanced low-level feature on SIFT is employed as the original low-level feature, which has 139 dimensions. Second, since the spatial and scale information of local features has been embedded in the learned subspace for visual words, there is no need to extend the BOW representation for image representation using SPM. For a dictionary (bag-of-words) of K words, our method uses a histogram of K dimensions to represent an image, while the image representation using SPM has $21K$ dimensions. For the similar performance, the image representation of our method is less than 1/10 of that on SPM. This means our image representation is much more effective than SPM. Therefore, it is possible to perform real-time scene recognition in mobile computing platform with limited computing resources.

3. Experimental results

We evaluated the effectiveness of our method on two challenging datasets for scene recognition: Scene-15 [5] and UIUC-Sports [35]. Since we aim at easy implementation with off-the-shelf open sources for real-world applications, we also implemented a version using OpenCV and evaluated on an indoor scene dataset. The performance of Extended Spectral Regression with respect to various parameters is also evaluated. The details are presented in the rest of this section.

Low-level features: To be fair for comparison, densely sampled grayscale SIFT features are used for Scene-15 and UIUC-Sports datasets as in [5]. The dimension of SIFT descriptor is 128, and the dimension of enhanced low-level local feature on SIFT is 139. The dense SIFT features are sampled at 6 scales of 16, 24, 32, 48, 64, 80 pixels for patch sizes and the offsets of grids are the half sizes of scales. From an image of 320×240 pixels, totally, 1882 local features are extracted. In practice, the multiple scale SIFTs capture two clusters of visual appearance features. The features of the first three scales capture small local visual patterns, such as corners, while the features of the second three scales capture some distinctive objects in the scene, *e.g.* a window in a building or a car on the street. Thanks to the integral image technique [36], the extra cost to compute the multi-scale SIFTs over single scale SIFTs is very low. In the OpenCV implementation, salient SIFT keypoints are extracted using OpenCV functions. Standard k-mean algorithm is used to generate BOW from the embedded features from all training images, which clusters the embedded features into K clusters and the centers of the clusters are used as the visual words of the dictionary (bag-of-words). Typically, the size of the bag-of-words (K) can be selected from 400 to 3000. Simple hard quantization is used to generate histogram on BOW.

Parameters: The parameters used for all the experiments are the same. The empirically chosen parameters are: α in (4) is 0.6; σ_e, σ_r, and σ_s in (11) are 0.05, 0.64, and 0.09, respectively. σ_o is not used except for the last testing on an indoor dataset with salient SIFTs from OpenCV. The enhanced low-level local features ($\mathbf{x}_i$) of 139 dimensions are reduced as embedded features ($\mathbf{z}_i$) of 70 dimensions in the experiments. Since spectral regression is a powerful machine learning approach, it is not sensitive to the selection of parameters. In a quite large range of each parameter, the variations of the final performance of scene recognition are within 1%. The evaluation of ESR on parameters is presented in the end of this section.

Classifiers: Both linear (Linear) and histogram intersection kernel (HI) SVM classifiers from LibSVM are used in the evaluation on Scene-15 and UIUC-Sports datasets. The linear SVM is efficient and the computational cost is almost constant once the size of BOW is fixed. Intersection kernel SVM can improve the classification accuracy, but it requires to save the image representations of all training images and computes the distance on kernels. Both memory requirements and computational costs are very high. Hence, the linear SVM is much more efficient for real-time tasks on mobile platforms. In the OpenCV implementation, the linear SVM from SVM-Light is employed for its efficiency.

Baselines: First, PCA is used as a baseline method to compare the benefit of data dimensionality reduction for BOW-based scene recognition. Second, to evaluate the effectiveness of embedding various aspects of low-level feature for BOW-based scene recognition, we use 1-level SPM (SPM-1) as a baseline approach which depends only on dense SIFT features and does not involve any spatial information. Third, to evaluate the effectiveness of exploiting spatial information of local features for scene recognition, we use 3-level SPM (SPM-3) as baseline. For evaluation on the implementation on an open source, the BOW method based on salient SIFTs extracted by OpenCV is used as a baseline. The baseline results on the first two datasets are obtained either from the literature or by running the source code from the author.

3.1. Scene-15

This dataset [5] contains 4485 images of 15 scene categories, including indoor and outdoor scenes. Each category has 200–400 gray-level images, and average image size is 300×250 pixels. Following the experimental setup for this dataset [5], 100 images per class are randomly selected as training samples and the rest are used for testing, and the experiment is repeated 10 times. In each experiment, the number of multi-scale dense SIFT features extracted from all 1500 training images is over 2,658,000, *i.e.* $m > 2,658,000$. They are first clustered into $n=3000$ tiny clusters to perform ESR. The average classification accuracy and standard deviation on 15 categories are reported. The test is performed on different size of the bag-of-words, *i.e.* $K=400$, 800, 1600, 2400 and 3600, respectively.

The performances of our method and comparisons with the three baseline approaches on linear and intersection kernel SVM classifiers are shown in Fig. 4, where the plot shows the classification accuracies as a function of the dimension of the image representation. First, let us examine the effectiveness on feature

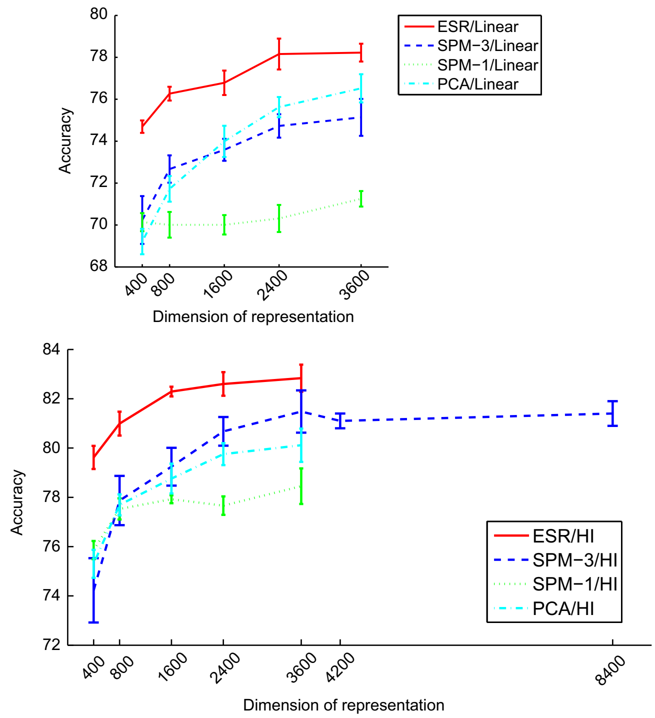

Fig. 4. The performance of our method and comparisons with baseline approaches on the 15-Scenes dataset. Upper: the performance plots on Linear SVM classifiers (Linear); lower: the performance plots on Intersection Kernel SVM classifiers (HI).

dimensionality reduction. Comparing the plots of our method (ESR) with those of PCA on both Linear and HI SVMs, one can observe that ESR clearly outperforms PCA. This result indicates that ESR is more effective on feature dimensionality reduction for scene recognition. Second, let us compare our method with SPM-1. It can be observed that with the same vocabulary size K, our method outperforms SPM-1 by more than 6% when Linear SVM is used and 4% when HI SVM is used. This means that the visual words on the embedded low-level features have better representation power than that on the original SIFTs even though its dimension ($d=70$) is only slightly more than half the size of the original SIFT descriptor (128). Third, compared with SPM-3, one can notice that, for the image representations of the same dimensions, our method is superior to the baseline for over 2%. Our method achieves the comparable performance (81%) with the image representation of 800 dimensions, while SPM-3 requires 8400 (21×400, over 10×800) dimensions to achieve 81.4% accuracy. When using 3600 words, our method achieves 83.0% accuracy, better than the best performance of SPM-3 (81.4%) with 8400 dimensions. This indicates that our embedding approach ESR is much more effective than SPM to integrate spatial information of low-level features.

Discussions: Scene-15 dataset has been widely used to test image classification methods. As mentioned in Section 1.1, recent researches focused on extensions of BOW representation to achieve improved performance, or focused on post-stages after the generation of bag-of-words. Our method focuses on pre-stage processing before the generation of bag-of-words, as illustrated in Fig. 1. We aim at mapping the raw low-level features into a compact and enhanced manifold subspace so that the BOW representation would be more effective for image classification. Direct comparison with state-of-the-arts may not be suitable. As examples, in [11], by introducing sparse coding on macro-features, the performance on Scene-15 is improved from 80% to 85.6% with the image representation being extended from 8400 to 43,008; and by using Fisher encoding, accuracy is improved from 81.4% to

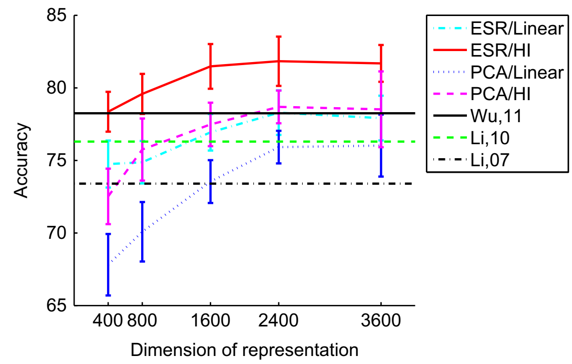

Fig. 5. The performance of our method (ESR) and comparisons with PCA and state-of-the-arts on the Sports Event dataset.

88.1% with the dimension of image representation being extended to 64,500 [22,28]. In [37], the method just on SIFT features achieves 80.1% accuracy, and when 4 types of features are introduced, it achieves 86.6%. Our method achieves 83.0% with 3600 dimensions of image representation when only SIFT features are employed. If we only consider the feature coding directly on SIFT features, our method is superior to soft coding (76.7%) [16] and sparse coding (80.3%) [20]. In principle, the post-stage processes can be applied to our BOW representation for further improvements, but with the expenses of great extension of image representation and extra computation for feature encoding. It might not be applicable for real-time systems on mobile platforms.

3.2. UIUC-sports

This is a dataset of 8 complex sports event categories (*bocce, croquet, polo, rowing, snowboarding, badminton, sailing, and rock-climbing*) [35]. Each category has 137–250 images. Following the setup in [35], for each class, 70 images are randomly sampled for training and 60 images are used for testing. All images are resized to the resolution of about 320×240 pixels. For this dataset, totally $m = 1,053,920$ multi-scale dense SIFT features are extracted from all 560 training images for ESR learning. Again, they are first clustered into $n=3000$ tiny clusters to perform ESR. The test is performed on vocabulary (bag-of-words) of size $K=400$, 800, 1600, 2400, and 3600, respectively.

In Fig. 5, we show the classification accuracies for different vocabulary sizes with both linear and intersection kernel SVM classifiers, where the three horizontal lines represent the performances of state-of-the-arts. Again, when compared with PCA, one can observe the clear increases of performances, *i.e.* over 4% increases in average, on both Linear and HI SVMs. In addition, one can notice that the performance of ESR/Linear is very close to PCA/HI. Using intersection kernel SVM, we obtain a clear margin of the performance superiority over the state-of-the-arts. The best performance of $82.1 \pm 0.7\%$ is obtained with the dimension $K=2400$, superior to 73.4% in [35], 76.3% in [25], and 78.25% in [9].

3.3. Indoor scenes with OpenCV SIFTs

One of the most significant benefits of ESR-based scene recognition is its efficiency for real-time tasks. To evaluate our method for real-time tasks on mobile platforms, a new dataset on indoor navigation using mobile platform with small training samples was built. The images are randomly extracted from videos captured by a mobile platform when it was moving in a large office building. During data collection, the camera was looking around as human beings observing a new place. The videos were captured in two different days. Training samples were extracted from one day and testing samples were extracted from the other day. The scenes in the building are classified into five categories for context interpretation by the robot. They are *lift-lobby*, *corridor*, *cubicle*, *meeting-room*, *and pantry*. The difficulties of this dataset can be highlighted as (a) Lighting conditions for training and testing images are different; (b) many views in the testing set are not observed in the training set; (c) there are many uninformative images in both training and testing sets; (d) limited training samples for real-world application. A few example images of the 5 scenes are displayed in Fig. 6 with one row for each category, where the left-side images are more informative and the right-side images are less informative since both training and testing images are randomly selected from the videos.

For each category, we randomly selected 60 images for training and 120 images for testing, where the testing images are much more than the training images compared with some public datasets [35]. When using $K=800$ words, the baseline performance is 75.1% and the accuracy rate of our method is 84.1%. We implemented our algorithm using Visual C++ 2005 on a Samsung Series 7 Slate handheld tablet with Core i5 processor (1.86 GHz and 4 GB on OS64bits) and OpenCV 2.3.1 is employed. For the image size of 320×240 pixels, the average time for our method is 245 ms, among them, the time for SIFT detection by OpenCV is 203 ms. Hence, it can achieve scene recognition at about 4 fps, which is fast enough for real-time indoor navigation for robots and humans. It can be further speeded up by improving the implementation of SIFT detection for mobile devices.

3.4. Evaluations on ESR

Comparison with SR: To perform SR on a large dataset, we use a cluster center to represent a tiny cluster of samples, and merge the identical rows and columns of the same cluster center in the original similarity matrix W to reduce its size. Since a tiny cluster may contain many similar samples from different classes, in the reduced similarity matrix $\overline{W}$, the weight $\overline{W}_{kl}$ is a weighted sum of elements from the original similarity matrix W. Then, a question arises: how close ESR approximates the original SR. Due to the great number of low-level features from the datasets used above, it is prohibited to perform SR on the such datasets. We perform the comparisons on synthetic datasets.

To give a comprehensive visualization of the performance, existing researches in the related field usually use synthetic data of low-dimensions (*e.g.* 3D to 2D, like the famous Swiss roll [38]) to illustrate the performance. In this paper, random 3D dataset of two classes are generated for each test. The samples of the first class are generated by 3 symmetrical Gaussians centered at [2,0,0], [−2,0,0] and [0,−2,0] at *XYZ* space with variances 0.7, 0.7 and 0.8, respectively, for all 3 dimensions, and the samples of the second class are created by another 3 Gaussians centered at [0,0,0], [0,1.5,0] and [0,3.5,0] with variances of 0.3, 0.5 and 1.0, respectively. 250 samples are generated for each Gaussian cluster, hence, totally 1500 samples are generated for each test. The projections of the test datasets on the *XY* plane are shown in the top row in Fig. 7, where the red cross '$\times$' indicates the samples of the first class and the blue circle '$\circ$' indicates the samples of the second class.

For comparison, same implementations of SR and ESR, except that ESR is based on tiny clusters, are tested. The weight of the similarity matrix W for SR is computed as

$$W_{ij} = \begin{cases} C_S d_S(\mathbf{x}_i, \mathbf{x}_j) \\ C_D d_D(\mathbf{x}_i, \mathbf{x}_j) \end{cases} = \begin{cases} C_S \exp\left(\dfrac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{(\rho\sigma)^2}\right) \\ C_D \exp\left(\dfrac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma^2}\right) \end{cases}, \tag{14}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are two 3D samples. For ESR, the weight W_{kl} (8) in the reduced similarity matrix $\overline{W}$ (7) is implemented as

$$W_{kl}(c_k, c_l) = \begin{cases} C_S d_S(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l) \\ C_D d_D(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}_l) \end{cases} = \begin{cases} C_S \exp\left(\dfrac{\|\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_l\|^2}{(\rho\sigma)^2}\right) & \text{if } c_k = c_l, \\ C_D \exp\left(\dfrac{\|\overline{\mathbf{x}}_k - \overline{\mathbf{x}}_l\|^2}{\sigma^2}\right) & \text{if } c_k \neq c_l, \end{cases} \tag{15}$$

where $\overline{\mathbf{x}}_k$ and $\overline{\mathbf{x}}_l$ represent the centers of two tiny clusters. We use both SR and ESR to embed the original 3D sample data into 2D manifold spaces.

Fig. 7 shows the comparisons of ESR with SR on different σ values while the other parameters kept as constants ($C_S=1.0$, $C_D=0.2$ and $\rho=3$). From the left column to the right column in Fig. 7, the results are generated with σ being 0.06, 0.08, 0.12, and

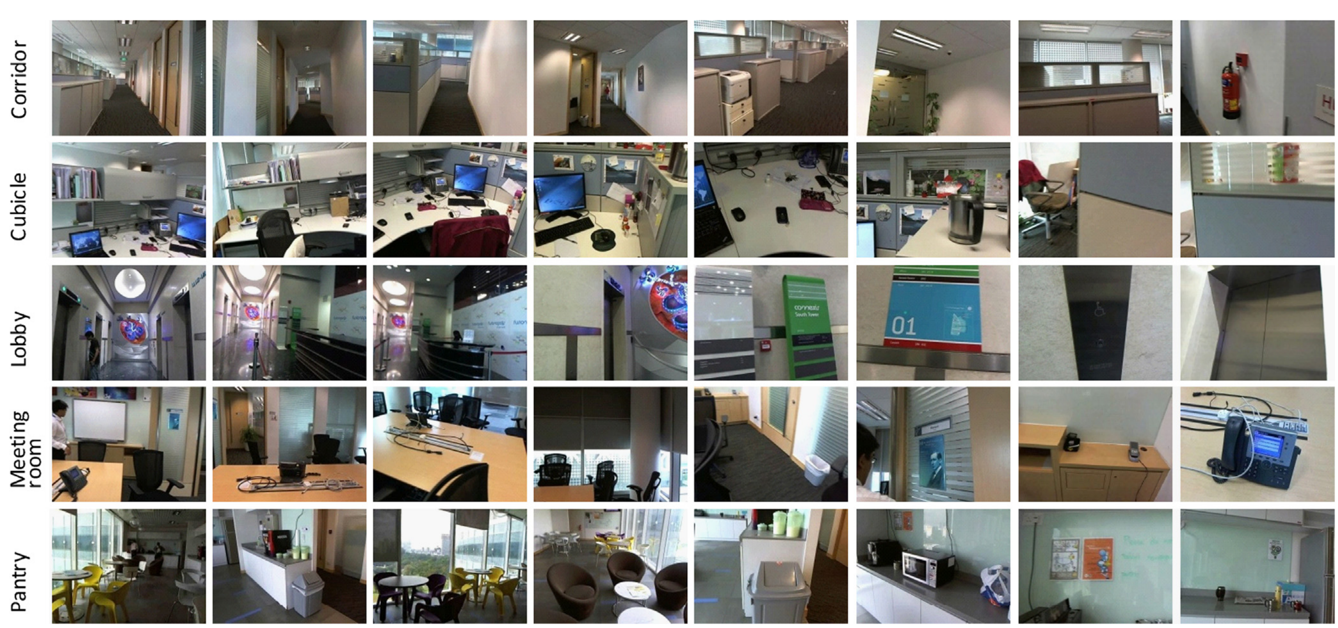

Fig. 6. Example images of indoor scenes captured in an office building.

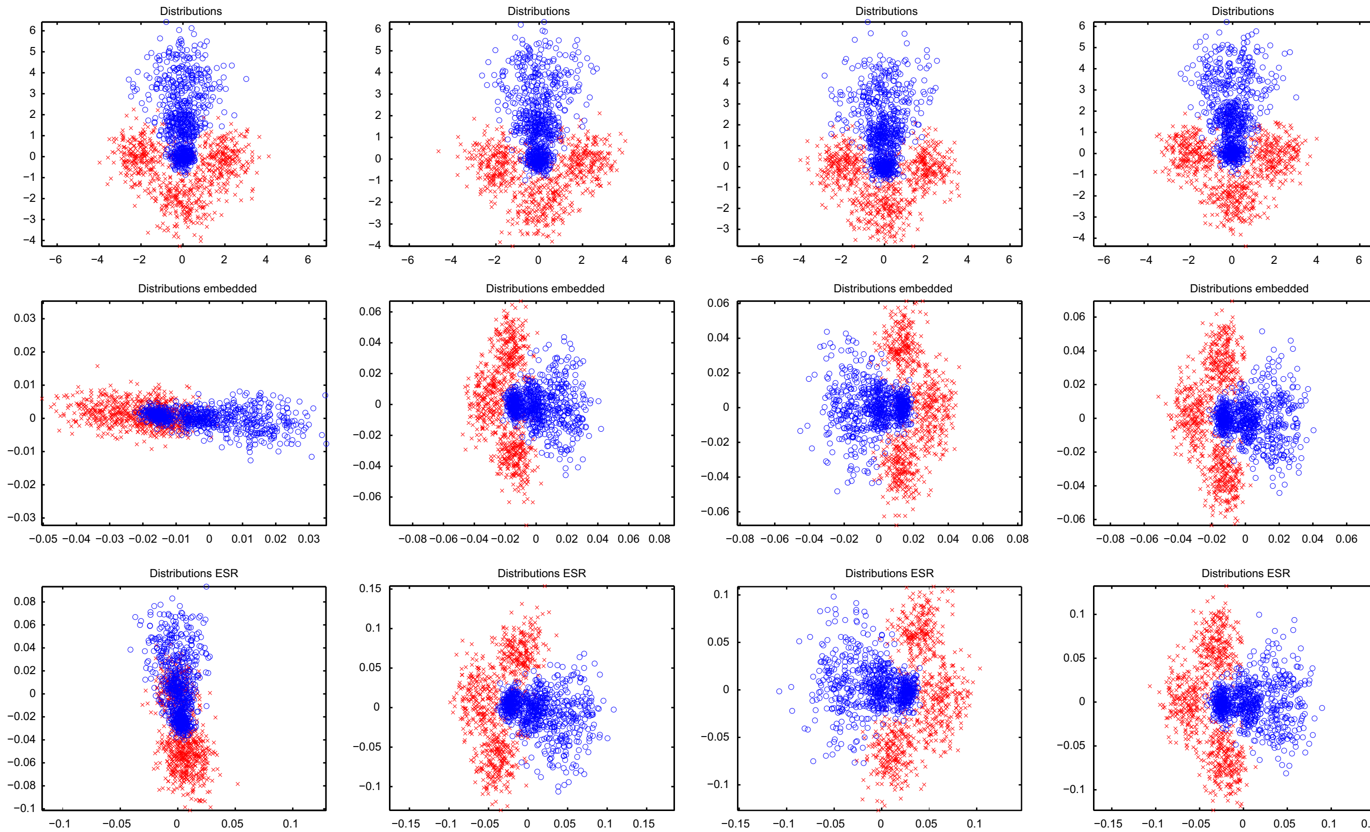

Fig. 7. Comparisons of SR and ESR on synthetic datasets. Top row: distributions of raw datasets; middle row: distributions of data embedded by SR; bottom row: distributions of data embedded by ESR. From left to right, the σ values used in SR and ESR are 0.06, 0.08, 0.12, and 0.14, respectively. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

0.14, respectively. First, let us examine the effectiveness of data embedding by SR. Comparing the figures of the second row with those of the first row, one can find that, when the value of σ is suitable (*e.g.* $\sigma \geq 0.08$), the geometric properties of the dataset are preserved in the low-dimensional space. Meanwhile, the distribution along the y-dimension in the original space is compressed, which results in a more compact distribution, and, on the other hand, the distribution along the x-direction is extended, which may result in an enhanced discrimination since the clusters of blue points (second class) are surrounded by the red points (first class). However, if the value of σ is too small (*e.g.* $\sigma \leq 0.06$), the correlations between close samples are too tight that might result in over-compression, as shown by the left figure in the middle row, which seems to indicate that SR tries to embed the 3D distribution of the dataset into a 1D manifold subspace. This may also indicate that, if one wants to map the 3D dataset into a 1D manifold subspace, it is better to select a small σ value. Then, let us compare the figures in the two lower rows, we can find that ESR generates the similar distributions of the embedded datasets. Moreover, if we examine the details more closely, we can find that the datasets generated by ESR are slightly sparser than those by SR, especially along the regions overlapped by the two classes, which might be helpful to separate the data of the two classes better. These are visual observations. We try to verify these observations from a measure. Let $R = d^d_{np}/d^s_{np}$ be the ratio of the average of the distances between two nearest points of different classes to the average of the distances of two nearest points of the same class. From 20 randomly selected cases similar to the two right columns in Fig. 7, we obtain the mean and variance of R for SR as 130.75 and 16.9 and those for ESR as 132.32 and 15.13, while the corresponding values for the source data are 28.77 and 1.96. It can be seen that the performances of SR and ESR are very close. It is noteworthy that this measure might not be a formal one to compare the data scatter since the scales of the distributions mapped by SR and ESR are greatly different. If the data distribution is sparser, both d^d_{np} and d^s_{np} are larger, but R value may be reduced.

For this case of synthetic datasets, it is found that the performance of ESR is stable and close to SR on a large range of parameter values. First, if $\rho = 1$, which means that the same bandwidth is used for Gaussian kernels employed for samples of both the same class and different classes, both SR and ESR cannot work since the computation for eigen-analysis fails. If ρ is small, the value of σ should be large (*e.g.* $\sigma > 0.2$), otherwise, it may result in over-compression by both SR and ESR. Once $\rho \geq 3$, both SR and ESR become stable with $\sigma = 0.08$. Too large of ρ might cause over-smoothing of the dataset in the embedded subspaces. If C_D is set as the same as C_S (*e.g.* $C_D = C_S = 1$), both SR and ESR result in over-compression and the data of two classes are merged together. For this synthetic case of two class datasets, when C_D is chosen from 0.8 to 0.01, both SR and ESR produce the results close to those displayed in Fig. 7 with $\rho = 3$ and $\sigma = 0.09$. The number of tiny clusters depends on how wide the raw dataset scatters in the original data space. In this test, the original samples are generated by 6 Gaussians. We find that when more than 200 tiny clusters are employed (*i.e.* $n \geq 200$), the performance of ESR is stable and close to SR. We also evaluate the performance of SR and ESR on the parameter α. With fixed parameters $C_S = 1$, $C_D = 0.2$, $\rho = 3$, $\sigma = 0.09$, and $n = 300$, when α varies from 0.01 to 50, the performance of SR and ESR is stable and close to the results shown in the right three columns in Fig. 7.

Performance on parameters: To evaluate the ESR performance with respect to parameters on real datasets, we perform the experiments on the Scene-15 dataset. First, we fix a default

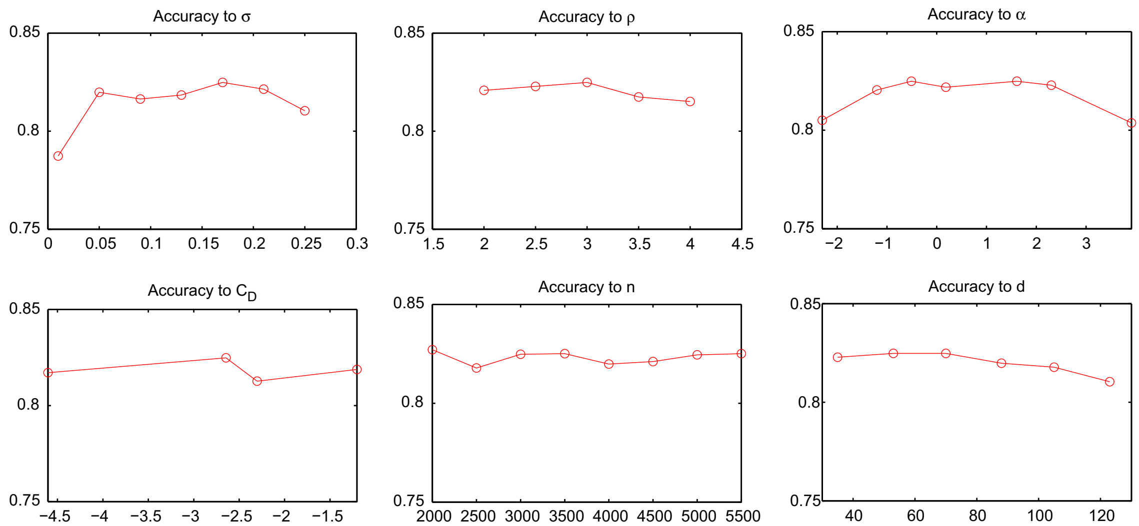

Fig. 8. Evaluation of our method's performance with respect to parameters on the Scene-15 dataset. From the upper-left figure to the lower-right figure are the curves of accuracy rates on parameters σ, ρ, α, C_D, n, and d, respectively.

parameter set as $d=70$, $\rho=3$, $(\sigma_e, \sigma_r, \sigma_s) = (0.17, 0.967, 0.136)$, $\alpha=0.6$, $C_D = 1/(L-1) = 0.071$, $n=3000$, and $K=1600$. Then, each time, we change the value of one parameter and obtain the curve of recognition accuracy. The curves of accuracy rates with respect to σ, ρ, α, C_D, n and d are shown in Fig. 8, where, in all plots, the range of vertical axis which represents the accuracy rate is from 75% to 85%. The upper-left figure shows the accuracy rate on σ. It can be seen that the accuracy rates are close and stable when σ varies from 0.05 to 0.21, and there is a slight drop when $\sigma < 0.05$ or $\sigma > 0.21$. The upper-middle curve shows the performance on ρ. When ρ varies from 2 to 4, the obtained accuracy rate changes between 82.08% and 82.48%. The upper-right figure shows the obtained curve when α changes from 0.1 to 50, where the horizontal axis represents log α. This plot indicates that when $0.3 \le \alpha \le 10$, the performance of ESR changes less and is over 82%. The lower-left figure shows the curve of the performance on C_D, where the horizontal axis also represents log C_D. The tested C_D values are 0.01, 0.071, 0.1, and 0.3, and the performance changes between 81.27% and 82.48%. The evaluation on the number of tiny clusters (n) is presented by the lower-middle figure. It shows that no significant difference of performance is observed when n is selected from 2000 to 5500. The accuracy rate is within [81.78%, 82.71%]. The last figure shows the change of accuracy rate with respect to the dimension of the embedded features (d). While less changes of performance are observed, it is interesting to find that when the dimension is lower ($d \le 70$), the performance is slightly better, which means ESR is an effective subspace learning approach. All these figures in Fig. 8 indicate that ESR is very robust to the parameters. In a quite large range for each parameter, the changes of performance are within 0.5%.

4. Conclusions

We proposed a new spectral regression approach, called Extended Spectral Regression (ESR), for subspace learning on a large dataset containing a huge number of data samples. We first cluster the huge number of samples into a large number of tiny clusters. Then, we derive a reduced similarity matrix on the tiny clusters for all data samples, and propose a new way to compute the similarity weights for the edges between the tiny clusters, where, each tiny cluster may contain hundreds of data samples with different semantic labels. Evaluation on synthetic datasets shown that ESR performs similar to SR with a much smaller similarity matrix.

We apply ESR to embed the low-level image features to form an effective bag-of-words representation for efficient scene recognition. To be able to effectively involve various aspects of low-level features in the subspace manifold learning, we first proposed an enhanced low-level feature representation which includes scale, orientation, spatial position, and visual appearance of a local feature. The similarity measure based on the enhanced low-level feature representation for computing the edge weights in ESR learning is designed. The ESR is then applied to learn an effective embedding which maps the enhanced low-level features into an optimized subspace manifold. The bag-of-words (dictionary) is then generated from the embedded features for scene recognition. The experiment results show that ESR is more effective than PCA in data dimensionality reduction for scene recognition, and more effective than SPM in combining spatial information of local features for scene recognition.

There are two advantages of the proposed method for scene recognition. First, it generates a compact representation of visual words (about half size of original features) which can speedup the computation of histogram representation on bag-of-words. Second, it generates a much low dimension image representation on bag-of-words which integrates various aspects of low-level features, such as spatial information, for image classification. These lead to an efficient approach for real-time scene recognition on a portable mobile platform with limited memory and computational resources. The performance of scene recognition can be future improved by combining more local features (*e.g.* colors) using ESR, or introducing post-stage processes after the generation of bag-of-words, *e.g.,* the Fisher Vector encoding.

Conflict of interest

None declared.

References

[1] A. Flint, I. Reid, D. Murray, Learning Texton Models for Real-Time Scene Context, CVPR Workshop on Egocentric Vision, 2009.
[2] A. Pronobis, B. Caputo, P. Jensfelt, H.I. Christensen, A realistic benchmark for visual indoor place recognition, Robot. Auton. Syst. 58 (1) (2010) 81–96.

[3] J.H. Lim, Y. Li, Y. You, J.P. Chevallet, Scene Recognition with Camera Phones for Tourist Information Access, ICME, 2007.
[4] F.-F. Li, P. Perona, A Bayesian Hierarchical Model for Learning Natural Scene Categories, CVPR, 2005.
[5] S. Lazebnik, C. Schmid, J. Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, CVPR, 2006.
[6] A. Bosch, A. Zisserman, X. Munoz, Scene classification using a hybrid generative/discriminative approach, IEEE Trans. Pattern Anal. Mach. Intell. 30 (4) (2008) 712–727.
[7] C. Wang, D. Blei, L. Fei-Fei, Simultaneous Image Classification and Annotation, CVPR, 2009.
[8] N. Xie, H. Ling, W. Hu, X. Zhang, Use Bin-Ratio Information for Category and Scene Classification, CVPR, 2010.
[9] J. Wu, J.M. Rehg, CENTRIST: a visual descriptor for scene categorization, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1489–1501.
[10] J. Liu, C. Zhang, Q. Tian, C. Xu, H. Lu, S. Ma, One Step Beyond Bags of Features: Visual Categorization Using Components, ICIP, 2011.
[11] Y-L. Boureau, F. Bach, Y. LeCun, Learning Mid-Level Features For Recognition, CVPR, 2010.
[12] M. Pandey, S. Lazebnik, Scene Recognition and Weakly Supervised Object Localization with Deformable Part-based Models, ICCV, 2011.
[13] D. Cai, X.F. He, J.W. Han, Spectral Regression: A Unified Subspace Learning Framework for Content-Based Image Retrieval, ACM MultiMedia, Germany, 2007.
[14] D. Cai, X.F. He, J.W. Han, Spectral Regression for Efficient Regularized Subspace Learning, ICCV, 2007.
[15] F. Perronnin, J. Sánchez, T. Mensink, Improving the Fisher Kernel for Large-Scale Image Classification, ECCV, 2010.
[16] J. Gemert, C. Veenman, A. Smeulders, J. Geusebroek, Visual word ambiguity, IEEE Trans. Pattern Anal. Mach. Intell. 32 (7) (2010) 1271–1283.
[17] K. Chatfield, V. Lemtexpitsky, A. Vedaldi, A. Zisserman, The Devil is in the Details: An Evaluation of Recent Feature Encoding Methods, BMVC, 2011.
[18] J. Sivic, A. Zisserman, Video Google: A Text Retrieval Approach to Object Matching in Videos, ICCV, 2003.
[19] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual Categorization with Bags of Keypoints, ECCV International Workshop on Statistical Learning in Computer Vision, 2004.
[20] J. Yang, K. Yu, Y. Gong, T. Huang, Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification, CVPR, 2009.
[21] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-Constrained Linear Coding for Image Classification, CVPR, 2010.
[22] F. Perronnin, C. Dance, Fisher Kernels on Visual Vocabularies for Image Categorization, CVPR, 2007.
[23] A. Bosch, A. Zisserman, X. Munoz, Image Classification using Random Forests and Ferns, ICCV, 2007.
[24] N. Morioka, S. Satoh, Building Compact Local Pairwise Codebook with Joint Feature Space Clustering, ECCV, 2010.
[25] L.-J. Li, H. Su, E.P. Xing, F.-F. Li, Object Bank: A High-Level Image Representation for Scene Classification and Semantic Feature Sparsification, NIPS, 2010.
[26] J. Xiao, J. Hays, K.A. Ehinger, A. Oliva, A. Torralba, SUN Database: Large-Scale Scene Recognition from Abbey to Zoo, CVPR, 2010.
[27] Y. Su, F. Jurie, Visual Word Disambiguation by Semantic Contexts, ICCV, 2011.
[28] J. Krapac, J. Verbeek, F. Jurie, Modeling Spatial Layout with Fisher Vectors for Image Categorization, ICCV, 2011.
[29] J.M. Sotoca, R.A. Mollineda, J.S. Sánchez, Problem Difficulty Analysis for Enhanced Application of Editing and Condensing, Pattern Recognition: Progress, Directions, and Applications, 2006.
[30] F. Jurie, B. Triggs, Creating Efficient Codebooks for Visual Recognition, ICCV, 2005.
[31] S.D. Kamvar, D. Klein, C.D. Manning, Spectral Learning, IJCAI, 2003.
[32] Y. Bengio, O. Delalleau, N. Le Roux, J. Paiement, P. Vincent, M. Ouimet, Spectral dimensionality reduction, Stud. Fuzziness Soft Comput. 207 (2006) 519–550.
[33] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[34] M. Belkin, P. Niyogi, Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering, Advances in Neural Information Processing Systems, 2001.
[35] L-J. Li, F.-F. Li, What, Where and Who? Classifying Events by Scene and Object Recognition, ICCV, 2007.
[36] P. Viola, M. Jones, Robust real-time object detection, IJCV 57 (2) (2001) 137–154.
[37] T. Harada, H. Nakayama, Y. Kuniyoshi, Improving Local Descriptors by Embedding Global and Local Spatial Information, ECCV, 2010.
[38] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, Science 290 (2000) 2323–2326.

Liyuan Li received BE and ME degrees from Southeast University, China in 1985 and 1988, and PhD degree from Nanyang Technological University, Singapore in 2001. During 1988 to 1999, he was on the faculty of Southeast University. In 2001, he joined the Institute for Infocomm Research, Singapore, where he is currently a Senior Scientist.

Weixun Goh received the B.A. degree in Computer Science from Cornell University, USA. He is currently a doctoral student in the Computer Vision & Robotics group at the Department of Engineering, University of Cambridge. His main research interests include computer vision, machine learning, and robotics.

Joo-Hwee Lim received B.Sc. (Hons I) and M.Sc. degrees in Computing from NUS, Singapore and Ph.D. degree in Computer Science & Engineering from UNSW, Australia. He is the Head, Visual Computing Department, at the Institute for Infocomm Research, A*STAR, Singapore. He has published more than 190 international refereed journal and conference papers and co-authored 18 patents (awarded and pending) in computer vision and pattern recognition.

Sinno Jialin Pan received the BS and MS degrees in applied mathematics from Sun Yatsen University in China in 2003 and 2005, respectively, and the PhD degree in computer science and engineering from the Hong Kong University of Science and Technology in 2010. He is currently a research scientist at the Institute for Infocomm Research, Singapore. His main research interests include transfer learning, semi-supervised learning, and their applications in pervasive computing and information extraction.