SALIENCY AWARE LOCALITY-PRESERVING CODING FOR IMAGE CLASSIFICATION

Quan Fang[1,2], *Jitao Sang*[1,2], *Changsheng Xu*[1,2]

[1]National Lab of Pattern Recognition, Institute of Automation, CAS, Beijing 100190, China
[2]China-Singapore Institute of Digital Media, Singapore, 119615, Singapore
{qfang, jtsang, csxu}@nlpr.ia.ac.cn

ABSTRACT

The Bag-of-Features (BOF) model is widely used for image classification. Most BOF models incorporate a step of maximum pooling to generate the raw image representation, where salient atoms with maximum response are reserved for final representation. However, recent locality-preserving coding schemes do not account for the saliency characteristic during the process of generating the raw image representations. In this paper, we propose a saliency aware locality-preserving coding scheme by explicitly considering saliency into the dictionary creation and feature coding stages. The novel coding scheme guarantees strong response in the pooling operation and thus contributes to a discriminative image representation. Experiments on three benchmark datasets validate the effectiveness of the proposed method.

Index Terms— Locality-preserving, Saliency, Image Representation and Classification

1. INTRODUCTION

Image classification, including object and scene classification, is a fundamental problem in computer vision. The Bag-of-Features (BoF) approach [1] represents an image as a compact histogram of visual word occurrences over a pre-learned dictionary which defines a collection of "visual words". Combined with powerful classifiers such as SVM, the BoF method and its extensions have achieved state-of-the-art performance on several famous image categorization datasets like Caltech101 [2], UIUC-Sport dataset [3] and 15-Scenes [4].

Typically, BoF method for image classification involves four stages: feature extraction, dictionary creation, feature coding and feature pooling. In the feature extraction step, image descriptors such as SIFT [5] or HOG [6] are extracted by interest detectors or dense sampling. The Dictionary which could be predefined or learned over all descriptors is designed for further encoding. Feature coding is designed to encode the response of feature descriptors over the dictionary into raw representations. Much recent work has devoted to utilizing the **locality** constraint to generate the raw image representation and shown impressive performance [7, 8]. Laplacian sparse coding [8] uses a similarity matrix to describe the locality for input features. Wang et al.[7] proposed the locality-constrained linear coding (LLC) to project each descriptor into the local coordinates formed by its k nearest neighbors. Locality-preserving image representation methods hold several attractive properties such as the stableness of the coding algorithm, local smooth sparsity, etc. Feature pooling summarizes the distribution of the codes by some well-chosen aggregation statistic. Pooling features over a local neighborhood will obtain invariance to small transformations and constitute the image final representation. The pooling operation is typically a sum, an average, a max rule. Extensive work indicates that max-pooling that chooses the largest coefficient for a visual word can lead to better classification performance [9, 7, 10]. According to the popular maximum feature pooling operation, only the maximum response on each dictionary atom is preserved while the lower ones are discarded. This is called **saliency** characteristic [11] between features and atoms, which has been employed for descriptive and discriminative image representation. The saliency for dictionary means that if a dictionary atom is much closer to a descriptor than other atoms, it will obtain a very strong response and contributes to the descriptive power of final representation.

However, the existing locality-preserving coding schemes do not account for the saliency characteristic of the pooling stage. In this paper, we consider saliency into the dictionary creation and feature coding stages and propose a saliency-aware locality-preserving coding scheme. Firstly in the dictionary creation stage, we take into account the local geometric and learn a locality-preserving dictionary for saliency pooling. Specifically, to learn a locality-preserving dictionary, it determines the location of the dictionary atom in the input feature space by analyzing the approximated tangent plane of the atom and its neighbors distribution. In the feature coding stage, we adaptively select the coding bases according to the local density distribution, instead of fixedly selecting the number of bases [7]. Fixedly selecting coding bases may lead to weak responses in low-density areas while poor reconstruction results in high-density areas. By adaptively selecting the coding bases depending on the local density distribution, it can obtain appropriate size of coding bases for input features and guarantee deriving large response. Therefore,

978-0-7695-4711-4/12 $26.00 © 2012 IEEE
DOI 10.1109/ICME.2012.164

this adaptively coding scheme can make the representation descriptive and thus boost the classification performance.

Therefore, the contributions of this paper can be summarized as three-fold:

1. We propose a novel locality-preserving coding scheme by explicitly considering saliency pooling operation. It is simple to implement with high computational efficiency. Experimental results on three benchmark datasets validate the effectiveness of the method.

2. Local geometrical structure is exploited in the dictionary creation stage. We introduce a locality-preserving dictionary creation algorithm to guarantee salient response in the pooling stage.

3. For feature coding, local density distribution is considered for adaptively selecting the coding bases. The sparse coefficients are obtained by using the $Epanechnikov$ quadratic kernel in an assignment fashion with respect to the saliency characteristic.

2. SALIENCY AWARE LOCALITY-PRESERVING CODING

In this section, we present our saliency aware locality-preserving coding method including locality-preserving dictionary learning algorithm and adaptively locality-constrained coding scheme for image representation and classification.

Let X be a set of n-dimensional local descriptors extracted from images, i.e.$X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] = [X_1, X_2, \ldots, X_c] \in \mathbb{R}^{n \times N}$, where X_i is sub-set of the training samples from class i and c is the total number of classes. Learning a structured dictionary $D = [\mathbf{d}_1, \mathbf{d}_2, \ldots, \mathbf{d}_K] \in \mathbb{R}^{n \times K}$ with K entries, where each $\mathbf{d}_i$ represents a basis vector (i.e. visual word) in the dictionary. Let $S = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_N] \in \mathbb{R}^{K \times N}$ be the sparse coefficient matrix, where each column is a sparse representation for the corresponding local feature in X.

2.1. Locality-preserving Dictionary Learning

Since in the locality-constrained coding stage, each local descriptor leads to a representation using K nearest basis vectors selected from the dictionary. After encoding all descriptors, each dictionary atom obtains multiple responses. Due to the maximum pooling operation, the maximum response is preserved while the other low responses are discarded. Intuitively, if a basis vector is much closer to a local feature than other vectors, it should have a relatively stronger response. This saliency characteristic [11] can enhance the descriptive power for image representation. For dictionary, if a dictionary atom is surrounded by a local dense group of local descriptors, it could obtain very strong response and thus benefits the maximum pooling operation. Therefore, the dictionary atoms should be distributed in the local dense regions in the input feature space regarding the saliency characteristic. This constraint makes the dictionary locality-preserving.

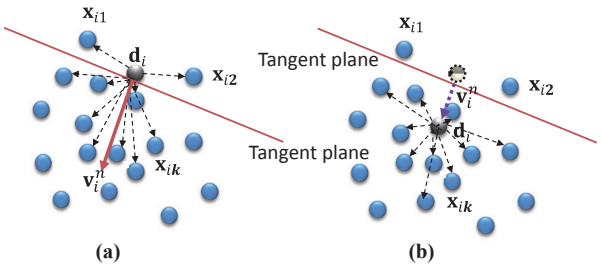

Fig. 1. Dictionary learning by exploring the local geometrical structure around dictionary code $\mathbf{d}_i$. Derivation of normal vectors $\mathbf{v}_i^n$ of the tangent plane for $\mathbf{d}_i$. Short dash arrows are the normalized vectors from $\mathbf{d}_i$ to its k-nearest neighbors formed by the feature descriptors.

Given a set of local features $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ and denote K as the size of the dictionary D, we aim to learn a locality-preserving dictionary by exploring the local geometrical structure and statistical information between local features and dictionary atoms.

We initialize the dictionary via k-means. We update each atom $\mathbf{d}_i$ of D individually. Each atom $\mathbf{d}_i$ should be surrounded with a set of local features. If a point of the dictionary lies on the surface of the local set of feature descriptors, then most of the feature points come from the dense side. Therefore, we can update the dictionary based on the determination of the distribution of nearest neighbor feature points for each dictionary atom. This process is illustrated in Figure 1.

For a dictionary atom $\mathbf{d}_i$, find its k-nearest neighbors $(\mathbf{x}_{ij}, j = 1, \ldots, k)$ from the local features. A vector v_{ij} from $\mathbf{d}_i$ to each of its kNNs can be represented as:

$$\mathbf{v}_{ij} = \mathbf{x}_{ij} - \mathbf{d}_i, \forall j = 1, \ldots, k \tag{1}$$

Normalize each $\mathbf{v}_{ij}$ as a unit vector $\mathbf{v}'_{ij}$ and the normal vector of the tangent plane at $\mathbf{d}_i$ is approximated as

$$\mathbf{v}_i^n = \sum_{j=1}^{k} \mathbf{v}'_{ij} \tag{2}$$

The relative location between $\mathbf{d}_i$ and its nearest neighbors can be determined by computing the angle between the normal vector of the tangent plane at $\mathbf{d}_i$ and $\mathbf{v}_{ij}$. If the angle θ_{ij} between $\mathbf{v}_i^n$ and $\mathbf{v}_{ij}$ is within 0 and $\pi/2$, then the $\mathbf{x}_{ij}$ is on the positive side of the tangent plane. Formally, we check the sign of the dot product :

$$\theta_{ij}^n = \mathbf{v}_{ij}^T \cdot \mathbf{v}_i^n \tag{3}$$

If $\theta_{ij}^n > 0$, then $\mathbf{x}_{ij}$ is on the positive side of the tangent plane at $\mathbf{d}_i$. If $\mathbf{d}_i$ lies on the surface of its local patch formed by its

nearest feature points, then all or most of its nearest neighbors are on one side of the tangent plane. Therefore, the relative location of $\mathbf{d}_i$ can be determined by counting the number of neighbors with $\theta_{ij} \geq 0$, written as

$$p_i = \frac{1}{k} \sum_{j=1}^{k} (\theta_{ij} \geq 0) \tag{4}$$

We set a threshold γ applied to p_i and determine whether the $\mathbf{d}_i$ lies on the surface of the local patch. More specifically, if $p_i \geq 1 - \gamma$, $\mathbf{d}_i$ is updated along the direction of the normal vector of the tangent plane.

$$\mathbf{d}_i^{t+1} = \mathbf{d}_i^t + \tau \mathbf{v}_i^n \tag{5}$$

where t is the iteration number and τ is the step size of the updating. If $p_i < 1-\gamma$, $\mathbf{d}_i$ lies in the relatively dense region of its local patch and is not updated. Through this procedure, we can learn a locality-preserving dictionary for feature coding in the next section. The algorithm of dictionary learning is summarized in Algorithm 1.

Algorithm 1 DICTIONARY LEARNING

Input: A data set of N data points i.e. local feature descriptors $X = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$, the parameter τ, γ.

Output: the learned dictionary D

1: **Initialization**: Initialize D via k-means.
2: **Update the matrix** D.
Compute $\mathbf{d}_i, i = 1, 2, \ldots, K$ individually by solving Eq.(1-5)
3: **Check the optimality conditions step:**
Return ro **step 2** until convergence or maximum iteration number achieves.

2.2. Adaptive Coding Bases Selecting

Given a learned dictionary D and an input local feature descriptor $\mathbf{x}$, our goal is to select the k nearest neighbor basis vectors as the coding bases for $\mathbf{x}$ such that $\mathbf{x}$ has a sparse representation regarding the locality. Previous work LLC [7] projects each descriptor on the space formed by its k nearest neighbors (k is fixed and small, e.g. $k = 5$). The density distribution for feature descriptors and visual codes is different locally. If a descriptor is located in a sparse neighborhood, fixedly selecting coding bases may lead to weak responses for atoms. For a descriptor in a dense neighborhood, fixedly selecting insufficient coding bases may lead to inappropriate responses for atoms and not match the locality. Therefore, when a feature descriptor is in a dense neighborhood, the number of its nearest coding bases k should be large. When a feature descriptor is surrounded by a sparse group of neighborhoods, the k should be small. This assumption based on the saliency constraint means that it can make the dictionary atoms close

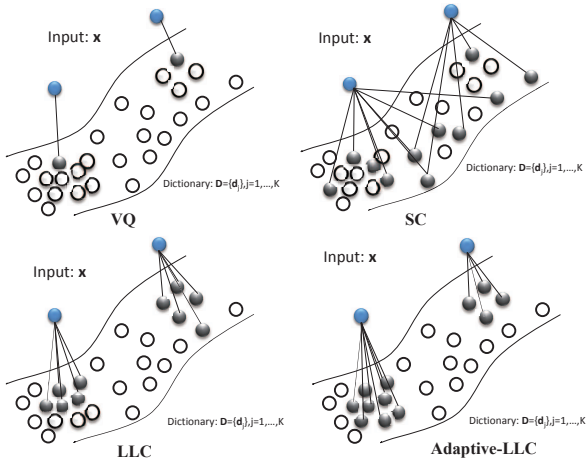

Fig. 2. Comparison between vector quantization(VQ), sparse coding(SC), standard locality-constrained linear coding (LLC, k is fixed), Adaptive-LLC(k is adaptive). The selected reconstruction bases for representation is highlighted in black.

to the descriptors and obtain strong responses in dense or sparse neighbors. Therefore, we should choose the k nearest coding bases for a feature descriptor locally and adaptively with respect to the saliency characteristic. Four kinds of coding bases selecting schemes are illustrated in Figure 2.

Selecting adaptive coding bases for a feature descriptor $\mathbf{x}$ can be achieved by utilizing different criteria. Here the bases are chosen based on the density distribution of the dictionary atoms. The density field created by a single data point can be described by a kernel of the form $\mathbf{K}(\cdot)$. Placing a kernel on each point, the estimated density at $\mathbf{x}$ is given by

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \mathbf{K}(\mathbf{x} - \mathbf{d_i}), \quad \mathbf{d}_i \in NN(\mathbf{x}) \tag{6}$$

where $NN(\mathbf{x})$ denotes a set of nearest neighbors of $\mathbf{x}$ and its cardinal size is N. We choose this coarsely. When the condition:$k \to \infty$ and $k/N \to 0$ as $N \to \infty$ is satisfied, the probability of error of k-NN asymptotically approaches the Bayes error [12]. Based on this condition, we set $N = 5 \ln K$, where K is the dictionary size. Here we use a Gaussian kernel and Eq.(6) can be rewritten as:

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{(2\pi h^2)^{\frac{1}{2}}} \exp\left(-\frac{\| \mathbf{x} - \mathbf{d}_i \|^2}{2h^2}\right), \quad \mathbf{d}_i \in NN(\mathbf{x}) \tag{7}$$

This density estimator contains only a single parameter, the kernel width h, which can be estimated effectively from the training data via cross-validation or,applying a self-tune technique to adaptively assign h. In this work, we define h as the average of k distance values from its k nearest neighbors.

After obtaining the estimated density probability at $\mathbf{x}$, we can select k nearest neighbors of $\mathbf{x}$ forming its local coordinate system according to the formula below based on the logistic function. $k = \left\lfloor \frac{E}{1+e^{-\beta p(x)}} + F \right\rfloor$, where β, E, F are scalar parameters. Apart from the constraints on k above, we set k vary in a fixed and small range. That is, k should satisfy the constraint: $K_{min} \leq k \leq K_{max}$, where K_{min} and K_{max} are the minimum and maximum number of nearest neighbors of $\mathbf{x}$ respectively. E, F can be computed using K_{min} and K_{max}.

2.3. Sparse Representation Learning

After obtaining the adaptive coding bases for the input local feature $\mathbf{x}$, we employ the assignment based coding utilizing the saliency characteristic to obtain the sparse coefficients.

For the assignment based method, we aim to seek a vector to $\mathbf{s}_i$ that measures the potential relationships between input local feature $\mathbf{x_i}$ and k selected basis vectors. Denote $s_{ji}, j = 1, \ldots, k$ as the response coefficient representing the weight between $\mathbf{x}_i$ and $\mathbf{d}_j, j = 1, \ldots, k$ in D_i, we can adopt different weighted methods to assign weights smoothly such as kernel regression with respect to the saliency characteristic.

$$s_{ij} = \frac{K(\cdot)}{\sum_{j=1}^{k} K(\cdot)} \tag{8}$$

where $K(\cdot)$ denotes the kernel and can be any appropriate form. In this paper, we choose the $Epanechnikov$ quadratic kernel.

$$K(\cdot) = \begin{cases} \frac{3}{4}(1-t^2) & if|t| \leq 1 \\ 0 & otherwise. \end{cases} \tag{9}$$

where $t = \frac{\|\mathbf{x}-\mathbf{d}_i\|}{\lambda}$, λ determines the size of the local region. $\lambda(\mathbf{x}) = |\mathbf{x} - \mathbf{d}_n|, \mathbf{d}_n$ is the k nearest neighbor of $\mathbf{x}$.

2.4. Computational Analysis

Our coding method has a computational complexity of $O(K + k)$, where K is dictionary size and k is number of nearest neighbors. Other coding methods, ScSPM [9], LScSPM [8], LLC [7] have computational complexity of $O(K^2), O(K^2), O(K + k^2)$ respectively. Our assignment based coding scheme is very simple to implement and thus excellently holds the computational advantage over other optimization based coding methods.

3. EXPERIMENTS AND RESULTS

3.1. Dataset and Experiment Setup

We evaluate the performance of the proposed locality-preserving feature learning method for image classification on three datasets: 15 scene dataset [4], the UIUC-Sport dataset [3], Caltech101 [2]. We compared our results with several existing state-of-the-art methods. Following common setting, we use 128 dimensional SIFT [5] features which are densely extracted from images on a grid with step size of 6 pixels and patch size of 16 pixels. All the images are pre-processed into gray scale and resized to keep the maximum size of height and width no more than 300 pixels. The SIFT descriptors are normalized with unit ℓ_2-norm. The dictionary is generated by our Alg1 and its size is fixed as 2048. The coding method is used to obtain the sparse coefficients. In the coding process, we use the sum normalization to normalize the obtained coefficients. To incorporate spatial information, the linear SPM [4] with three levels of $1 \times 1, 2 \times 2, 4 \times 4$ is adopted. Max-pooling and ℓ_2 normalization are adopted to generate the final image representation. Lib-linear SVM[13] is used for classification wherein the penalty coefficient is set to 10.

In our dictionary learning algorithm, the three parameters are (1): the step size for updating τ is set to 0.001 for UIUC-Sport, 15-Scene dataset , and Caltech 101 according to the good performance based on the observation. (2): the threshold γ is set to 0.9 empirically. (3):the nearest neighbor size of dictionary atom $k = 5\log_{10} N$ based on the condition that the probability of error of k-NN asymptotically approaches the Bayes error if it satisfies the condition: $k \to \infty$ and $k/N \to 0$ as $N \to \infty$ [12]. In the coding stage, assignment based coding method is adopted to generate the sparse representation. The $Epanechnikov$ quadratic kernel is used. The parameters K_{min},K_{max} for k in coding stage are set as 3 and 15 respectively referred to LLC [7] and β is set to 1.

3.2. Experimental Results and Analysis

UIUC-Sport. We first conduct the comparison on the UIUC-Sport [3] data set which contains 8 classes and 1792 images. These 8 categories are *rowing, badminton, polo, bocce, snow boarding, croquet, sailing, and rock climbing* and the image number of each category ranges from 137 to 250. We randomly select 70 images from each class for training and 60 test images from per class. We repeat this process for 10 rounds. We compared our result with three existing algorithms: linear SPM using sparse codes [9], Histogram Intersection Kernel [14], Laplacian sparse coding [8]. Classification accuracy is compared in Table 1. As shown, our proposed method improved performance compared with the state-of-the-art methods, which demonstrates the effectiveness of our method.

Scene-15. The 15-Scenes [4] dataset contains 4485 images distributing in 15 categories, with number of images each category ranging from 200 to 400. The images categories vary from indoor scenes like living room and kitchen to outdoor places like street and industrial. To compare with others' work, we randomly choose 100 images per class for training and the remaining is reserved for test. This process is repeated for 10 rounds. The results compared with four methods including nonlinear kernel SPM [4], linear SPM using s-

Table 1. Comparison of classification rate(%) on UIUC-Sport.

Algorithm	Classification Accuracy
ScSPM [9]	82.74±1.46
HIK+OCSVM [14]	83.54±1.13
LScSPM [8]	85.31±0.51
Ours	86±1.59

parse codes [9], Kernel codebooks [15], Laplacian sparse coding [8] are shown in Table 2. In this experiment, our method can achieve comparable performance while maintaining clear computational advantage compared with the optimization involved coding method. Note that LScSPM [8] outperforms our method. The possible reason is that scene images contain more heavy textures in single patch, LScSPM employs smooth constraints into coding process and similar patches will be encoded into similar sparse codes, which benefits the classification performance. To serve for saliency, we focus on the relations between dictionary atoms and local features but not explicitly consider the relations between local features and our method has computational advantage.

Table 2. Comparison of classification rate(%) on 15-scenes.

Algorithm	Classification Accuracy
KSPM [4]	81.40±0.50
ScSPM [9]	80.28±0.93
KC [15]	76.67±0.39
LScSPM [8]	89.75±0.50
Ours	82.55±0.01

Caltech-101. The Caltech-101 dataset [2] contains 9144 images in 101 classes with high variance in shape. The number of images per category varies from 31 to 800. Following the common experiment setup for Caltech-101, we use 30 images per class for training and leave the remaining for test. We repeat this process for 5 rounds. The results are compared with three existing algorithms including nonlinear kernel SPM [4], linear SPM using sparse codes [9] and LLC [7] in Table 3. As can be seen, our method can achieve comparable results while holds the computational efficiency compared with other optimization based coding methods.

3.3. Discussion

To comprehensively understand the effectiveness of the proposed method, we further analyze its performance with respect to dictionary learning, the adaptiveness for selecting bases. Here we report the results using UIUC-Sport, but the similar performance can be also applied to other datasets.

Table 3. Comparison of classification rate(%) on Caltech-101.

Algorithm	Classification Accuracy
KSPM [4]	64.40±0.80
ScSPM [9]	73.2±0.54
LLC [7]	73.44±-
Ours	73.96± 0.0038

First, we compare the classification accuracy using dictionary generated by K-means clustering algorithm and by our proposed locality-preserving dictionary leaning Alg1. Two dictionary generation strategies are: k-means clustering (KM) and our learning method utilizing locality constraint (DL). Two coding schemes are: fixedly coding method (LLC) and adaptively coding method (ALC). Four kinds of experimental setup (KM+LLC,DL+LLC,KM+ALC,DL+LLC) are compared. The results are plotted in Figure 4. As shown, the learned dictionary and the adaptively coding method improve the classification accuracy by 0.4∼1 percent over the method using K-means and fixedly coding scheme. This validates the effectiveness of the dictionary leaning algorithm and adaptively coding scheme under locality constraint.

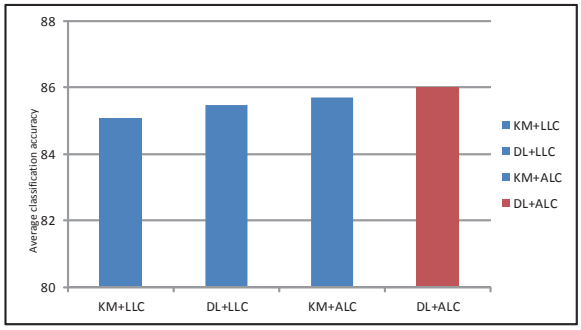

Fig. 3. Performance comparison of the methods with different codebook generation and coding schemes on UIUC-Sport Data Set(%). Assignment based coding with the $Epanechnikov$ quadratic kernel is used in all coding process to obtain the coefficients.

Second, we evaluate the effectiveness of adaptively selecting the size of coding bases for feature vectors by comparing with fixedly selecting the bases size for features coding. Figure 5 presents the performance using 2,5,10,40 neighbors and adaptively selecting neighbors respectively. As can be seen, a relatively small and proper number of neighbors leads to good classification accuracy. Adaptively selecting the neighbors can achieve better classification performance compared with the fixed, which indicates adaptive feature coding regarding the local structure can make the representation more

descriptive.

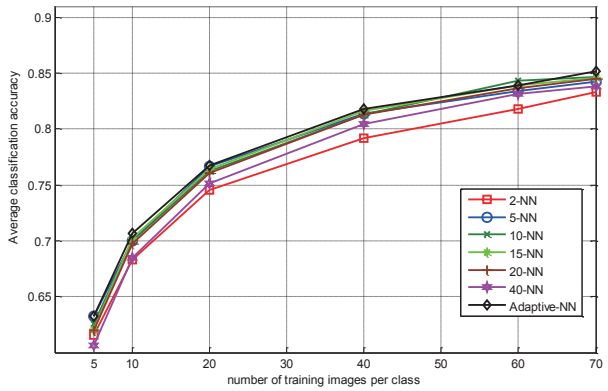

Fig. 4. Performance comparison under fixedly selecting different neighbors and adaptively selecting neighbors on UIUC-Sport Data Set.

4. CONCLUSIONS

In this paper, we propose a saliency aware locality-preserving coding method for image classification by exploring the local geometrical structure and statistical information between local features and dictionary atoms. Experimental results demonstrate that our method achieves or outperforms the state-of-the-art performance on several benchmark datasets. Meanwhile our dictionary learning procedure is simple to implement and the coding process involves no optimization, hence our method can maintain high computational efficiency, which validate the effectiveness of our method especially in the context of large-scale image classification task.

5. ACKNOWLEDGMENT

This work was supported in part by National Program on Key Basic Research Project (973 Program, Project No. 2012CB316304) and the National Natural Science Foundation of China (Grant No. 90920303, 61003161).

6. REFERENCES

[1] Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003, pp. 1470–1477.

[2] L. Fei-Fei, R. Fergus, and P. Perona, "Learning generative visual models from few training examples," in *Workshop on Generative-Model Based Vision, IEEE Proc. CVPR*, 2004.

[3] Li jia Li, "What, where and who? classifying event by scene and object recognition," in *In IEEE International Conference on Computer Vision*, 2007.

[4] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *CVPR (2)*, 2006, pp. 2169–2178.

[5] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[6] Navneet Dalal and Bill Triggs, "Histograms of oriented gradients for human detection," in *CVPR (1)*, 2005, pp. 886–893.

[7] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas S. Huang, and Yihong Gong, "Locality-constrained linear coding for image classification," in *CVPR*, 2010, pp. 3360–3367.

[8] Shenghua Gao, Ivor Wai-Hung Tsang, Liang-Tien Chia, and Peilin Zhao, "Local features are not lonely - laplacian sparse coding for image classification.," 2010.

[9] Jianchao Yang, Kai Yu, Yihong Gong, and Thomas S. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *CVPR*, 2009, pp. 1794–1801.

[10] Lingqiao Liu, Lei Wang, and Xinwang Liu, "In defense of soft-assignment coding," in *ICCV*, 2011, pp. 2486–2493.

[11] Yongzhen Huang, Kaiqi Huang, Yinan Yu, and Tieniu Tan, "Salient coding for image classification," in *CVPR*, 2011, pp. 1753–1760.

[12] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification*, Wiley, 2001.

[13] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[14] Jianxin Wu and James M. Rehg, "Beyond the euclidean distance: Creating effective visual codebooks using the histogram intersection kernel," in *ICCV*, 2009, pp. 630–637.

[15] Jan C. van Gemert, Jan-Mark Geusebroek, Cor J. Veenman, and Arnold W. M. Smeulders, "Kernel codebooks for scene categorization," in *ECCV 2008, PART III. LNCS*. 2008, pp. 696–709, Springer.