A New Method of Image Classification Based on Weighted Center Symmetric Local Ternary Pattern Feature

Mingming Huang Zhichun Mu* Hui Zeng

School of Automation and Electrical Engineering, University of Science and Technology Beijing,
Beijing, 100083, China
mmhuang1205@163.com
*mu@ies.ustb.edu.cn
hzeng@163.com

Received 7 September 2014; Revised 26 September 2014; Accepted 31 March 2015

Abstract. Texture information is critical to the accuracy of image classification systems. In this paper, we propose a novel descriptor called weighted center symmetric local ternary pattern (WCS-LTP), constructed by using the CS-LTP variance of the local region as an adaptive weight to adjust the contribution of the CS-LTP code in histogram calculation. Then, based on the proposed WCS-LTP descriptor, we introduce a new local WCS-LTP feature extraction approach. Compared with conventional local CS-LTP feature, our proposed WCS-LTP feature, which exploits the complementary information of local spatial pattern and local contrast, can better characterize the image local texture. Finally, WCS-LTP feature based sparse coding spatial pyramid matching (ScSPM) representation classification is proposed for image classification. Extensive experimental results demonstrate that the effectiveness of our proposed WCS-LTP feature based ScSPM representation classification algorithm.

Keywords: WCS-LTP feature, sparse coding spatial pyramid matching, image classification

1 Introduction

Image classification, which annotates an image with one or multiple labels corresponding to different semantic classes, is a highly useful yet still challenging task in the computer vision. It has attracted an increasing amount of attention over the past few decades as a result of its wide use in many applications such as human-computer interaction [1], video surveillance [2], robot path planning [3], and so on.

In most studies, there are two main steps in an image classification system [4].The first step is to extract visual image features for an effective representation of the image. And the second step is to classify the new image with a good classifier. For good classification, features should be descriptive and discriminative, and on the other hand, invariant to different transformations and robust enough to allow intra-class variation. In recent years, much effort has been invested in developing features that yield good classification and the focus in extracting features for classification has shifted from global features describing the object as a whole, to local features.

Famous contributions include SIFT (Scale Invariant Feature Transform) [5], PCA-SIFT [6], SURF (Speeded-up Robust Features) [7], HOG (Histogram of Oriented Gradient) [8], LBP (Local Binary Pattern) [9], CS-LBP (Center Symmetric Local Binary Pattern) [10], LTP (Local Ternary Pattern) [11], CS-LTP (Center Symmetric Local Ternary Pattern) [12] and so on. Among them, the SIFT descriptor, proposed over a decade ago, is currently among the best quality descriptors for image classification. It has shown great success in object recognition and detection because it is invariant to a variety of possible image transformations, such as scale, rotation, blur, illumination and viewpoint changes. The basic idea is to detect the extreme points in Difference-of-Gaussian (DoG) scale space, filter these extreme points to find the stable feature points known as interest points, and finally assign orientation and generate descriptor for interest points by vectors. Inspired by the high discriminative power and robustness of SIFT, many researchers have developed varieties of local descriptors following the way of SIFT. The PCA-SIFT descriptor is an extension of the SIFT descriptor, which applies PCA (Principal Component Analysis) to reduce the dimensionality of the SIFT descriptor vector from 128 to 36. The SURF descriptor also relies on local gradient histograms and speeds up the gradient computations using integral images, while almost preserving the quality of SIFT. Varieties of existing texture operators have been used for describing interest regions so far. The LBP operator, considered as one of the most popular texture features, has performed very well in various computer vision problems such as background subtraction, face recognition and texture classification. However, the LBP operator tends to produce a rather long histogram and is not that robust to flat image areas. To address these problems, Heikkila et al. [10] combined the strength of the SIFT descriptor and LBP texture operator to form the CS-LBP (Center Symmetric Local Binary Pattern) descriptor, which is reported to have better performance than SIFT, especially for matching image pairs with illumination changes. The LTP (Local Ternary Pattern) operator [11], which extends LBP to 3-valued codes, is more discriminant and less sensitive to noise in uniform regions. Therefore, the LTP descriptor has strong discriminative ability for describing texture structure. Unfortunately, the dimensionality of the LTP histogram is extremely high. To address this problem, Gupta et al. presented the CS-LTP descriptor [12] which generalized the CS-LBP descriptor with a ternary coding style.

For local features, the Bag-of-Visual-Words (BoV) model [13], which has been very popular, is used in image classification. The BoV method represents an image as an orderless collection of local features and its descriptive ability is severely limited due to discarding the spatial information of features. By overcoming this problem, one popular extension of the BoV method, called the spatial pyramid matching (SPM) [14], is proposed and has been shown to be effective for image classification. The SPM partitions an image into several segments in different scales, then computes the BoV histogram within each segment and concatenates all the histograms to form a high dimension vector representation of the image. For the purpose of reducing the training complexity and improving the scalability, sparse coding spatial pyramid matching (ScSPM) method [15] taking into account some aspects of the spatial layout of the image is proposed, which contribute to improving classification performance. Csurka [16] proposed BoV-based method for image classification. The proposed method was based on BoV model, where a set of SIFT features is first extracted and then an image is represented by the BoV frequency histogram of SIFT features for image classification. Wang et al [8] developed a new method of image classification by using the Histogram of Oriented Gradient (HOG) features which is computed on a dense grid of uniformly spaced cells. In addition, Akata et al [17] applied Principal Component Analysis (PCA) to reduce the dimensionality of the SIFT descriptor from 128 to 64 for image classification.

In modern days, the images on the website or computers generally contain complex background. Although local features have been proven to be very effective in image classification, the accuracy of classification is often limited by the presence of uninformative local features typically extracted from background [18]. The SIFT feature is able to capture local edge or shape of an object with distributions of intensity gradients. For an image with simple background, the SIFT feature is capable of accurately representing the foreground object without noise interference. However, the SIFT feature will perform poorly when the image contains complex background because a portion of extracted features may come from the noisy background. On the contrary, the CS-LTP descriptor [12], which does not take into account shape information, can not only filter out background noise through local ternary patterns but also capture the texture information of images. In fact, texture information is critical to the accuracy of image classification systems. However, the CS-LTP descriptor does not involve the information variance of the local region because it is obtained by building a histogram in which no matter what the CS-LTP variance of the local region, each CS-LTP pattern is assigned the same weight 1. Hence, the CS-LTP descriptor could not effectively characterize texture information of images to some extent due to the fact that the information variance of local regions is closely related to the texture feature. It is worth noting that effective local feature extraction approaches, which could better characterize the image local texture, are still needed to be investigated for image classification.

This paper investigates an effective algorithm based on sparse coding spatial pyramid matching representation of WCS-LTP feature for image classification. Our feature extraction scheme is first to construct a novel descriptor called weighted center symmetric local ternary pattern (WCS-LTP), which uses the CS-LTP variance of the local region as an adaptive weight to adjust the contribution of the CS-LTP code in histogram calculation. Then, based our proposed descriptor, we introduce a new local WCS-LTP feature extraction approach. Compared with conventional local CS-LTP feature, our proposed WCS-LTP feature which exploits the complementary information of local spatial pattern and local contrast, can better characterize the image local texture. By using the proposed local features, WCS-LTP feature based sparse coding spatial pyramid matching (ScSPM) representation classification algorithm is proposed for image classification. The proposed algorithm treats the WCS-LTP features from all the training samples as the dictionary of our ScSPM representation, and then the test image is represented a sparse vector using SPM strategy. Extensive experimental results show that the proposed classification paradigm achieves much better classification performance.

The rest of this paper is organized as follows. In Section 2, we describe our proposed WCS-LTP feature extraction method. Section 3 presents details of the proposed WCS-LTP feature based ScSPM representation classification algorithm. Experiments results are discussed in Section 4. Finally, we conclude this paper in Section 5.

2 WCS-LTP Feature

Before presenting in detail our proposed WCS-LTP feature, we briefly review of LTP and CS-LTP that form the basis for our work.

2.1 LTP and CS-LTP

The LTP operator extends LBP to 3-valued codes, in which the gray values in a zone of width T around the center pixel are set to one, ones above this are set to two and ones below it to zero, as illustrated in Eq. (1). Formally, the LTP operator takes the form as

$$LTP_{R,N}(x,y)=\sum_{i=0}^{N-1} s(n_i-n_c)3^i, s(x)=\begin{cases}2, x\ge T\\ 1, -T<x<T\\ 0, x\le -T\end{cases}. \tag{1}$$

where n_c denotes the gray value of the center pixel of a local neighborhood, n_i corresponds to the value of N neighboring pixels equally located on a circle of radius R, and T is a user-specified threshold. Obviously, the LTP operator produces 3^N distinct values, resulting in 3^N-dimensional histogram. Fig. 1 presents an example of calculating the CS-LTP code with eight neighboring pixels.

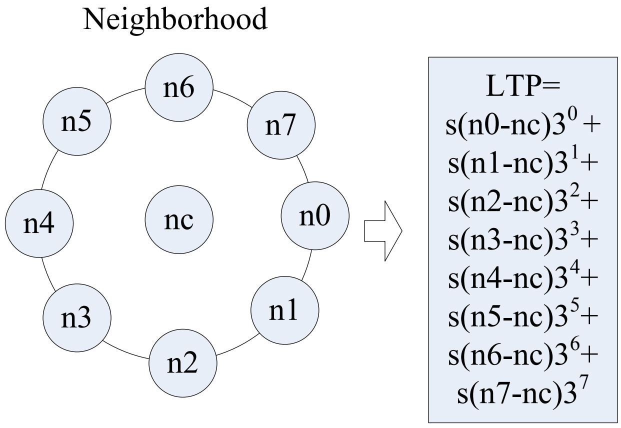

Fig. 1. Calculation of the LTP operator with eight neighboring pixels

To reduce the histogram size of LTP, the CS-LTP operator [12], which only compares the intensities of central symmetric neighboring sample points, is a powerful texture operator which characterizes the spatial structure of the local image texture. Formally, the CS-LTP operator is represented as

$$CS-LTP_{R,N}(x,y)=\sum_{i=0}^{N/2-1} s\left(n_i-n_{i+N/2}\right)3^i, s(x)=\begin{cases}2, x\ge T\\ 1, -T<x<T\\ 0, x\le -T\end{cases}. \tag{2}$$

where n_i and $n_{i+N/2}$ represent the gray value of center-symmetric pairs of pixels of N equally spaced pixels on a circle of radius R and T is a user-specified threshold. Fig. 2 shows an example of calculating the CS-LTP code with eight neighboring pixels.

Suppose the image patch is $W \times H$. In general, the CS-LTP histogram can be computed as

$$H(k)=\sum_{u=1}^{W}\sum_{v=1}^{H} f\left(CS-LTP(u,v),k\right), f(x,y)=\begin{cases}1, x=y\\ 0, x\ne y\end{cases}. \tag{3}$$

where $k\in[0,K]$, K is the maximal CS-LTP pattern value.

2.2 WCS-LTP Feature

For varieties of existing texture operators, the final feature will be obtained by building a histogram based on the code for each pixel within the image such as LBP, CS-LBP and LTP. Calculation of the CS-LTP histogram does not involve the information variance as shown in Eq. (2). That is to say, no matter what the CS-LTP variance of the local region, each CS-LTP pattern is assigned the same weight 1 for the histogram calculation. Actually, the variance is closely related to the texture feature. Generally, the high frequency texture regions will have higher variance and they contribute more to the discrimination of texture images [19].

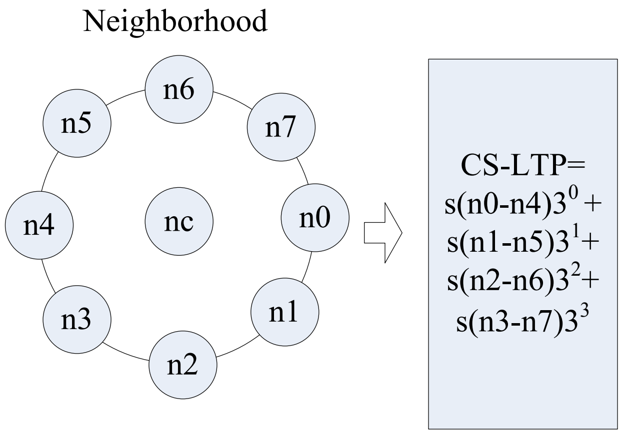

Fig. 2. Calculation of the CS-LTP operator with eight neighboring pixels

In this paper, we propose a new method of construction the WCS-LTP descriptor, in which the CS-LTP variance of the local region is used as an adaptive weight to adjust the contribution of the CS-LTP code in histogram calculation. Our proposed weighted histogram is computed as

$$H(k)=\sum_{u=1}^{W}\sum_{v=1}^{H} f\left(CS-LTP(u,v),k\right), f\left(x,y\right)=\begin{cases}\left|n_i-n_{i+N/2}\right|, x=y \\ 0, \qquad x\neq y\end{cases}. \tag{4}$$

In comparison with conventional CS-LTP descriptor, our proposed WCS-LTP descriptor is constructed by using the CS-LTP variance of the local region as an adaptive weight to adjust the contribution of the CS-LTP code in histogram calculation. It can better characterize the image local texture. In this paper, the WCS-LTP descriptor related parameters, *N*, *R*, *T* are fixed as 2, 8, 0.01.

Based on the proposed WCS-LTP descriptor, our proposed WCS-LTP feature is built. First, image patches, $p_i, i=1,2,...,n$, with size of 16 × 16 pixels are densely sampled from an image on a grid with stepsize 6 pixels and the image is reprocessed into gray scale. Then the WCS-LTP descriptors of each image patch p_i are generated. Finally, WCS-LTP features of an image are obtained. Compared with conventional CS-LTP feature, the proposed WCS-LTP feature, which exploits the complementary information of local spatial pattern and local contrast [20], can better characterize the image local texture. The whole process of the WCS-LTP feature extraction of an image is presented in Algorithm 1 below.

Algorithm 1 WCS-LTP features extraction of an image

Input: an image I(z)=I(x, y).

Output: WCS-LTP features of the image $I(x, y)$.

Step 1. Reprocess the image into gray scale and densely sample image patches with size of 16 × 16 pixels on a grid with stepsize 6 pixels.

Step 2. Generate the WCS-LTP descriptors of each patch $p_i, i=1,2,...,n$, denoted as WCS-LTP$_i$.

Step 3. Define all WCS-LTP$_i$ of the image I(x,y) as WCS-LTP features.

---

3 WCS-LTP Feature based ScSPM Representation Classification

ScSPM proposed by Yang [15] has shown its effectiveness in image representation [21]. Standard ScSPM framework consists in four key parts: (1) local features, (2) codebook representation, (3) sparse coding of local features and (4) Spatial Pyramid Matching model. The details of the four key concepts are as follows.

Local features. The essential aspect of the ScSPM model is to extract a set of local features, such as SIFT descriptors in an image.

Codebook representation. The codebook is a way which images can be represented as a set of local features. A codebook is learned offline in a training phase. Let **X** be a set of local features in a *D* dimensional feature space, i.e. $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_M]^{\mathrm{T}}$, where *M* is the total number of local features. Let $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_K]^{\mathrm{T}}$ be the codebook which needs to be learned and $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_M]^{\mathrm{T}}$ be the set of sparse reconstruct coefficients, where *K* is the number of visual words in the codebook. Then codebook learning can be formulated as follows:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V}} = \sum_{m=1}^{M} \left\| \mathbf{x}_m - \mathbf{u}_m \mathbf{V} \right\| + \lambda \left| \mathbf{u}_m \right| \\ &\textit{subject to } \left\| \mathbf{v}_k \right\| \le 1, \forall k = 1,2,...K \end{aligned} \quad . \tag{5}$$

where λ is a regularization parameter and normally the codebook **V** is an overcomplete basis set, i.e. $K > D$. A unit L_2-norm constraint on $\mathbf{v}_k$ is typically applied to avoid trivial solutions.

Sparse coding of local features. After obtaining the codebook, each local feature is quantized to one “visual word” by sparse coding. In a nutshell, every local feature is to be assigned a small number of visual words.

Spatial Pyramid Matching model. The SPM partitions an image into several segments in different scales, then computes the histogram within each segment and concatenates all the histograms to form a high dimension vector representation of the image.

Inspired by ScSPM, we firstly extracted WCS-LTP features from every training sample. Now **X** is a set of WCS-LTP features in a *D* dimensional feature space. Then the codebook **V** needs to be learned. We set the sparsity of the sparse codes $\lambda = 0.1$ in our paper.

The Eq. 5 is not convex for **U** and **V** simultaneously, but it is convex for **U** when **V** is fixed and it is also convex for **V** when **U** is fixed. Consequently, our way to solve Eq. 5 is to solve it iteratively by alternatingly optimizing over **U** or **V** while fixing the other. Fixing **V**, the optimization can be solved by optimizing over each coefficient $\mathbf{u}_m$ individually:

$$\min_{\mathbf{u}_m} = \sum_{m=1}^{M} \left\| \mathbf{x}_m - \mathbf{u}_m \mathbf{V} \right\| + \lambda \left| \mathbf{u}_m \right| . \tag{6}$$

Fixing **U**, the problem converts to a least square problem with quadratic constraints:

$$\begin{aligned} &\min_{\mathbf{V}} = \sum_{m=1}^{M} \left\| X - \mathbf{U}\mathbf{V} \right\|_F^2 \\ &\textit{subject to } \left\| \mathbf{v}_k \right\| \le 1, \forall k = 1,2,...K \end{aligned} \quad . \tag{7}$$

In fact, ScSPM image representation has a training phase and a coding phase. At first, a set of WCS-LTP features from a random collection of image patches is used to solve Eq. 5 with respect to **U** and **V**, where **V** is retained as the codebook; In the coding phase, for all features of each image are obtained by optimizing Eq. 5 with respect to U only. Finally all sparse reconstruct sparse coefficients of an image are concatenated to a high dimension sparse vector using SPM strategy.

In this article, we use 50000 WCS-LTP features extracted from random patches to train the dictionary, by iterating the steps of Eq. 6 and Eq. 7. When we get the dictionary in this off-line training, we can do on-line sparse coding efficiently as in Eq. 6 on each feature of an image. We use Multi-class linear SVM [15] for image classification. The proposed WCS-LTP feature based ScSPM representation classification approach is shown in Fig. 3. As illustrated in Fig. 3, the WCS-LTP features of training images are extracted by our proposed feature extraction approach. Then a ScSPM model is trained on these features. And the obtained image representation will work with linear SVM trained using the one-against-all rule: a classifier is learned to separate each category from the rest. Given a testing image, it is classified into the category with the maximum SVM output decision value.

4 Experiments and Results

As a local feature description method, our proposed WCS-LTP feature can tend to extract more precise texture information. Therefore, we propose to use WCS-LTP features extracted from random patches of all the training images to train the codebook of our proposed WCS-LTP feature based ScSPM representation classification model.

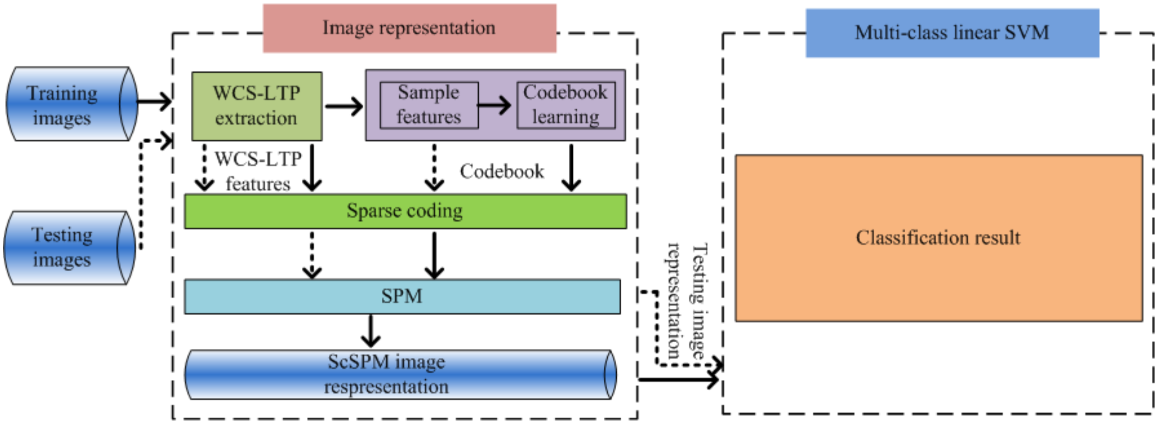

Fig. 3. The workflow of our proposed method. The training process is along the solid arrow while testing process is along the dotted arrow

In this Section, we will investigate our proposed WCS-LTP feature based ScSPM representation classification algorithm for image classification. Extensive experiments are carried out on SIMPLIcity dataset, Caltech 101 dataset and 15-Scene dataset to validate the claims of the previous sections.

4.1 Image Classification on SIMPLIcity Database

SIMPLIcity database [22] is used in the experiment to evaluate the performance of our proposed algorithm for image classification. This database is a subset of COREL image database, which contains totally 1000 images equally divided into 10 different categories: African people, beach, building, bus, elephant, flower, food, horse, dinosaur, and mountain. Some example images are shown in Fig. 4.

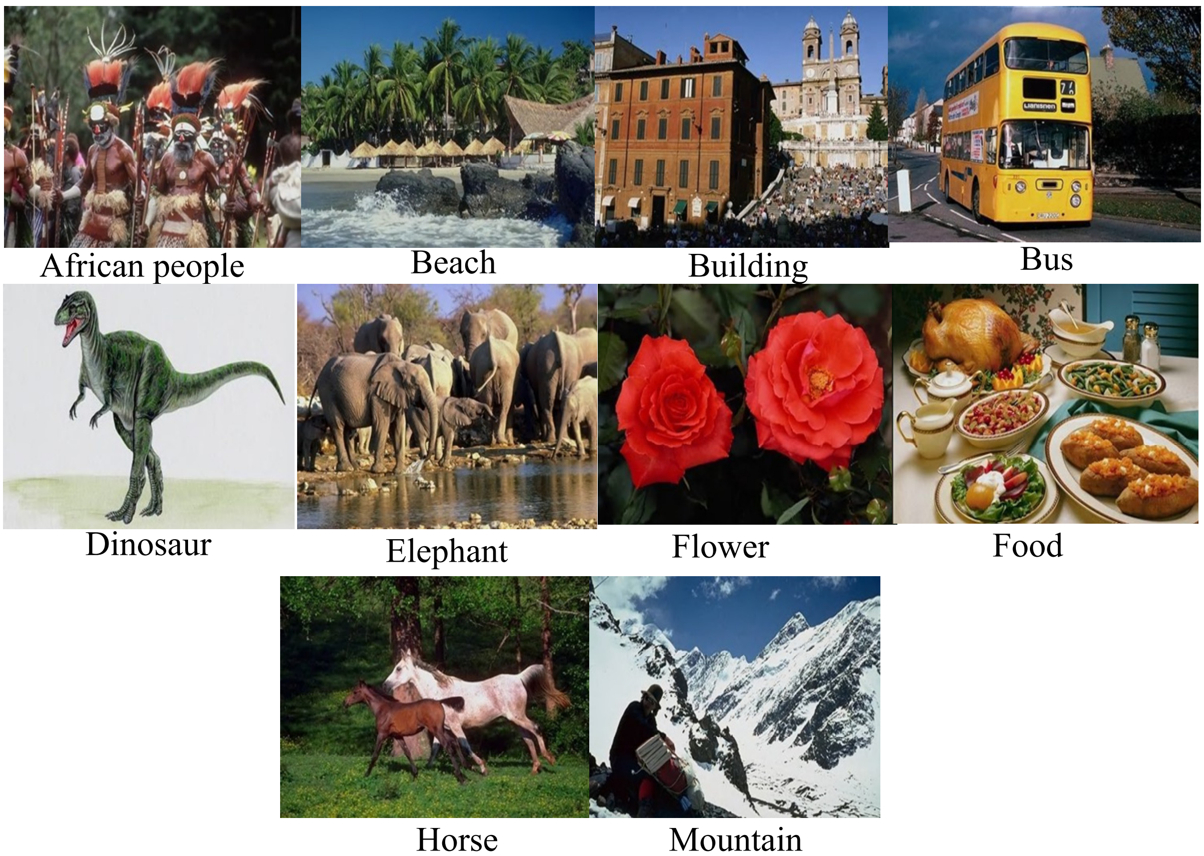

Fig. 4. Examples images of SIMPLIcity dataset

We randomly choose 30 images per category for training and the other for test. For all compared method, the codebook size is fixed as 256. The average of per-class recognition rates is computed based on the percentage of all test images classified correctly. And the mean of the average of per-class recognition rates is reported. We compare our approach with conventional CS-LTP features and SIFT features. The experiments results are shown in Tab. 1. From Tab. 1, we observe that our proposed WCS-LTP feature achieves the performance of 85.0%, which is better than SIFT and CS-LTP.

Table 1. Classification results on the SIMPLIcity dataset

Accuracy (%)	WCS-LTP	SIFT	CS-LTP
People	66.43	64.26	65.71
Beach	64.29	70.71	58.57
Building	81.43	71.43	70.0
Bus	100.0	100.0	100.0
Elephant	82.86	79.86	82.86
Flower	94.29	80.71	95.71
Food	87.14	68.0	87.14
Horse	95.71	93.57	98.57
Dinosaur	100.0	100.0	100.0
Mountain	77.86	72.86	60.0
Mean	**85.0**	**80.14**	**81.86**

4.2 Image classification on Caltech 101 database

We also evaluate the performance of our proposed algorithm on Caltech101 dataset [23] commonly used to evaluate the image classification. This dataset holds 9144 images in 101 categories including flowers, animals, vehicles, etc., with high shape variability. The number of images per class varies from 31 to 800. Most images are medium resolution, i.e. about 300×300 pixels. Caltech-101 is probably the most diverse object dataset available today, though it is not without shortcomings. In other words, most images feature relatively little clutter, and the objects are centered and occupy most of the images. Some example images are shown in Fig. 5.

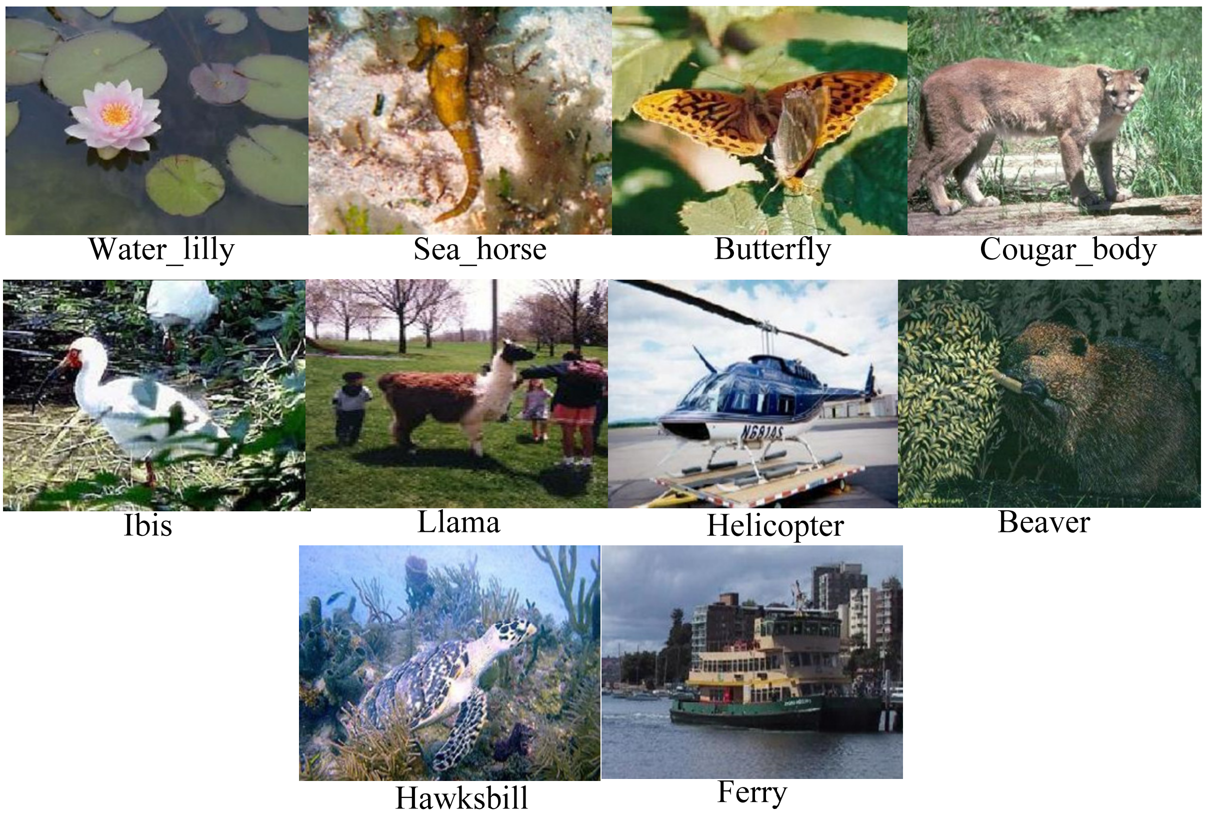

Fig. 5. Examples images of Caltech 101 dataset

In fact, if the number of the training samples is required to be sufficiently large, the classification algorithm could more accurately determine the identity of the test sample. Therefore, different numbers of training samples will affect the performance of our proposed algorithm. In this experiment, we randomly train on 15 and 30 images per category and test on the rest. And the codebook size is fixed as 1024. In order to get reliable results, this experimental process is repeated by 5 times with different training and testing images which are random selected. And the average of per-class recognition rates for each run is recorded. We compare our approach with conventional CS-LTP features. Detailed comparison results are shown in Tab. 2 in which our final results are reported by the mean and standard deviation of the average of per-class recognition rates. From the results described in Tab. 2, we can see that our proposed WCS-LTP feature outperforms the CS-LTP feature. When the number of images per category used to train is 30, the average accuracy of our proposed WCS-LTP outperforms CS-LTP by more than 4 percent; while the standard deviation of our proposed WCS-LTP is somewhat higher than that of CS-LTP. Even when the number of images per category used to train is 15, our proposed WCS-LTP can still achieve an average accuracy of 58.74%, outperforming CS-LTP by more than 6 percent; and the standard deviation of our proposed WCS-LTP is the same as that of CS-LTP.

When the codebook size is fixed as 1024, Tab. 2 has shown the performance of our proposed WCS-LTP. However, the codebook size plays an important role in our proposed WCS-LTP feature based ScSPM representation classification algorithm. Intuitively, the WCS-LTP feature may lose discriminative ability when the codebook size is too small; and the feature from the same class of images will not match when the codebook size is too large. Therefore, we investigate the effects of codebook size on our proposed algorithm. Inspired by Ref. 15, we tried three sizes: 256, 512 and 1024. And the final result is reported as the mean of the average of per-class recognition rates. The experimental results are illustrated in Tab. 3. For all the cases, our proposed WCS-LTP based ScSPM representation classification outperforms conventional CS-LTP based ScSPM representation classification. The performance of WCS-LTP based ScSPM representation classification increases as the codebook size grows further.

Table 2. Classification rate (%) on the Caltech 101 dataset

Algorithm	15 training	30 training
WCS-LTP	58.74 ± 0.004	66.32 ± 0.014
CS-LTP	53.58 ± 0.004	61.63 ± 0.009

Table 3. The effects of codebook size on WCS-LTP and CS-LTP respectively on Caltech 101 dataset

	Codebook size	256	512	1024
30 training	WCS-LTP	63.17	63.47	66.32
	CS-LTP	59.12	60.22	61.63
15 training	WCS-LTP	53.96	57.99	58.74
	CS-LTP	51.04	53.05	53.58

4.3 Image classification on15-Scene database

To further evaluate our proposed algorithm, we also tried our algorithm on 15-Scene dataset [24]. This dataset contains totally 4485 images falling into 15 categories, with the number of images each category from 200 to 400. The 15 categories vary from living room and kitchen to street and industrial. Some example images are shown in Fig. 6. Following the same experiment procedure of Yang et al. [15], we randomly choose 100 images per class for training and use the left for testing. And the codebook size is fixed as 1024. The average of per-class recognition rates is computed based on the percentage of all test images classified correctly. And the mean of the average of per-class recognition rates is reported. The experiments results are shown in Tab. 4. From Tab. 4, we observe that our proposed WCS-LTP feature based ScSPM representation classification algorithm achieves the performance of 80.52%, which is better than CS-LTP based ScSPM representation classification algorithm. Furthermore, additional contrast measures are added to the pattern histogram by using of WCS-LTP and this usually produces significantly better results than using CS-LTP. However, our proposed WCS-LTP is sensitive to illumination change. As can be seen in Tab. 4, the classification performance of WCS-LTP is worse than CS-LTP for some categories greatly affected by illumination variation. For example, the performance of WCS-LTP is 5% worse on MITcoast and MITstreet.

Fig. 6. Examples images of 15-Scene dataset

Table 4. Classification rate (%) comparision on 15-Scene dataset

Accuracy (%)	WCS-LTP	CS-LTP
CALsuburb	98.58	100.0
MITcoast	81.54	86.92
MITforest	93.86	93.86
MIThighway	88.13	83.75
MITinsidecity	80.52	81.25
MITmountain	86.50	74.45
MITopencountry	76.13	67.10
MITstreet	86.46	90.10
MITtallbuilding	82.81	85.16
PARoffice	96.52	96.52
bedroom	68.10	62.93
industrial	61.14	58.29
kitchen	66.36	71.82
livingroom	63.49	64.55
store	77.67	79.53
Mean	**80.52**	**79.75**

In Section 4, we evaluate the performance of our proposed algorithm on SIMPLIcity dataset, Caltech 101 dataset and 15-Scene dataset. First, we compare our proposed WCS-LTP feature with SIFT and CS-LTP on SIMPLIcity dataset. Results show that the average of per-class recognition rates of WCS-LTP is greater than that of SIFT and CS-LTP. For Caltech 101 dataset, our proposed WCS-LTP performs better than CS-LTP. Meantime we investigate the effects of codebook size on our proposed algorithm. And results show that our proposed WCS-LTP based ScSPM representation classification outperforms conventional CS-LTP based ScSPM representation classification in different codebook sizes. Finally, we also evaluate the performance of our proposed algorithm on 15-Scene dataset. The performance of our proposed WCS-LTP based ScSPM representation classification is better than the CS-LTP based ScSPM representation classification. In a word, the experiments in three different datasets - SIMPLIcity dataset, Caltech 101 dataset and 15-Scene dataset- show the effectiveness of the proposed WCS-LTP.

4 Conclusions

In this paper, WCS-LTP feature based sparse coding spatial pyramid matching representation classification algorithm is proposed. The proposed WCS-LTP feature employs the CS-LTP variance of the local region as an adaptive weight to adjust the contribution of the CS-LTP code in histogram calculation. Compared with conventional CS-LTP feature, the proposed WCS-LTP feature, which exploits the complementary information of local spatial pattern and local contrast, can better characterize the image local texture. Experimental results on SIMPLIcity, Caltech 101, and 15-Scene database demonstrate the effectiveness of our proposed WCS-LTP feature based sparse coding spatial pyramid matching representation classification algorithm. In our future work, we are willing to extend our proposed WCS-LTP feature which can not only extract more precise texture information but also capture the shape information of images.

Acknowledgement

This article is supported by the National Natural Science Foundation of China (Grant No. 61170116, No. 61375010, No. 61300075 and No. 61472031), Beijing Higher Education Young Elite Teacher Project (Grant No. YETP0375) and the Fundamental Research Funds for the Central Universities under the Grant No. FRF-TP-14-120A2. The authors would like to express their sincere appreciation to the anonymous reviewers for their insightful comments, which greatly helped them to improve the quality of the paper.

References

[1] V. I. Pavlovic, R. Sharma, T. S. Huang, "Visual Interpretation of Hand Gestures for Human Computer Interaction: A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 19, No. 7, pp. 677-695, 1997.

[2] K.Q. Huang, D.C. Tao, Y. Yuan, X. L. Li, T. N. Tan, "Biologically inspired features for scene classification in video surveillance," *IEEE Transactions on Syst. Man Cybern. Part B: Cybern*, Vol. 41, No. 1, pp. 307-313, 2011.

[3] C. Siagian, L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 2, pp. 300-312, 2007.

[4] J. Sánchez, F. Perronnin, "High-dimensional signature compression for large-scale image classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1665-1672, 2011.

[5] Z. Akata, Z. Harchaoui, C. Schmid, "Good practice in large-scale learning for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 36, No. 3, pp. 507-520, 2014.

[6] D. G. Lowe, "Distinctive image features from scale invariant keypoints," *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 91-110, 2004.

[7] H. Bay, A. Ess, T. Tuytelaars, L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding*, Vol. 110, No. 3, pp. 346-359, 2008.

[8] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, "Locality-constrained linear coding for image classification," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3360-3367, 2010.

[9] M. Heikkila, M. Pietikainen, "A texture-based method for modeling the background and detecting moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 4, pp. 657-662, 2006.

[10] T. Ojala, M. Pietikainen, T. Maenpaa, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 971-987, 2002.

[11] X. Tan, B. Triggs, “Enhanced local texture feature sets for face recognition under difficult lighting conditions,” *Analysis and Modeling of Faces and Gestures*, Springer Berlin Heidelberg, pp. 168-182, 2007.

[12] R. Gupta, H. Patil, A. Mittal, “Robust order-based methods for feature description,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 334-341, 2010.

[13] J. Sivic, A. Zisserman, “Video Google: A text retrieval approach to object matching in videos,” in *Proceedings of IEEE International Conference on Computer Vision*, pp. 1470-1477, 2003.

[14] S. Lazebnik, C. Schmid, J. Ponce, “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” in *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 2169-2178, 2006.

[15] J. Yang, K. Yu, Y. Gong, T. S. Huang, “Linear spatial pyramid matching using sparse coding for image classification,” in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1794-1801, 2009.

[16] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, “Visual categorization with bags of keypoints,” in *Proceedings of Workshop on Statistical Learning in Computer Vision*, *ECCV*, Vol. 1 No. 1-22, pp. 1-2, 2004.

[17] Z. Akata, Z. Harchaoui, C. Schmid, “Good practice in large-scale learning for image classification, ” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 36, No. 3, pp. 507-520, 2014.

[18] Y. Fan, S. Qin, “A new method of image classification based on local appearance and context information,” *Neurocomputing*, Vol. 119 pp. 33-40, 2013.

[19] B. S. Manjunath, W. Y. Ma, “Texture features for browsing and retrieval of image data,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, pp. 837-842, 1996.

[20] Ojala T, Pietikainen M, Maenpaa T, “Multiresolution gray-scale and rotation invariant texture classification with local binary patterns,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7, pp. 971-987, 2002.

[21] J. Wang, J. Yang, K. Yu, T. Huang, Y. Gong, “Locality-constrained linear coding for image classification,” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3360-3367, 2010.

[22] J. Z. Wang, J. Li, G. Wiederhold, “Simplicity: Semantics-sensitive integrated matching for picture Libraries,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, 947-963, 2001.

[23] F.-F. Li, R. Fergus, P. Perona, “Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories,” *Computer Vision and Image Understanding*, Vol. 106, No. 1, pp. 59-70, 2007.

[24] J. Z. Wang, J. Li, G. Wiederhold, “Simplicity: Semantics-sensitive integrated matching for picture Libraries,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, 947-963, 2001.