Scene Classification Using Spatial and Color Features

Peilong Zeng, Zhixin Li, Canlong Zhang

Scene Classification Using Spatial and Color Features

Peilong Zeng, Zhixin Li*, Canlong Zhang

College of Computer Science and Information Technology
Guangxi Normal University, Guilin, 541004, China
* Corresponding author.

zplhang@126.com, lizx@gxnu.edu.cn, Zhangcl@gxnu.edu.cn

Abstract. With the increment of images in modern time, scene classification becomes more significant and harder to be settled. Many models have been proposed to classify scene images such as Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Analysis (PLSA) and Latent Dirichlet Allocation (LDA). In this paper, we propose an improved method, which combines spatial and color features and bases on PLSA model. When calculating and quantizing spatial features, chain code is used in the process of feature extraction. At the same time, color features are extracted in every block region. The PLSA model is applied in the scene classification. Finally, the experiment results between PLSA and other models are compared. The results show that our method is better than many other state-of-the-art scene classification methods.

Keywords: Probabilistic Latent Semantic Analysis; scene classification; spatial feature; chain code; KNN-SVM classifier

1 Introduction

In recent years, image understanding and classification has been frequently researched and widely applied to all kinds of practical systems [9]. As an important issue of image classification, scene classification aims to label an image among a set of semantic categories (such as mountains and tall buildings) automatically [3]. For example, images in Fig. 1 can be classified into the category of "coast".

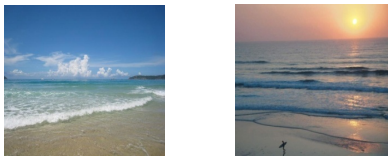


Fig. 1. The coast scene images with different illumination

It differs from the conventional object classification. Scene classification is an extremely challenging and difficult task because of the ambiguity, rotation, variability and the wide change of illumination and scale conditions of the scene images even if for the same scene category [5]. The images in Fig. 1 also show that a category may

include many coast images with different illumination. What's more, a scene is generally composed of several entities and the entities are often organized in an unpredictable layout [12]. It is difficult to define a set of properties that would include all its possible manifestations and extract effective common features to classify the images to the same category.

As was mentioned in [4], there were two basic strategies in the literature about scene classification. The first strategy uses low-level features such as global color, texture histograms and the power spectrum. It is normally used to classify small number of scene categories (city versus open country, etc.) [9]. The second one uses an intermediate representation before classifying scenes [10,11] and it has been applied to cases where there are a larger number of scene categories [4]. Biederman [1] showed that humans can recognize scenes by considering them as a whole one, without recognizing individual objects. Oliva and Torralba [11] proposed a low dimensional representation of scenes, based on global properties such as naturalness and openness.

A lot of efforts have been made to solve the problem in greater generality, through design of techniques capable of classifying relatively large number of scene categories in the last few years [2, 13]. In this paper, we calculate spatial features via chain code method instead of Hough transform in [9] first and then combine them with color features, and then use PLSA to classify scene images.

The main contributions of our paper lie in two aspects. One is the application of chain code method to the spatial features' extraction. We use the chain code to describe the shape and spatial information of a scene image. The combination of spatial and color features improve the performance of scene classification. The other is that we use PLSA for learning and SVM and KNN classifiers for classifying. An improved classifier KNN-SVM [17], the hybrid of KNN and SVM classifier, is used. The experiment results prove the good effect of the classifier.

The next section briefly describes the PLSA model. Then, in Section 3, we describe the classification method by applying PLSA model and KNN-SVM classifier to images. Section 4 describes the spatial and color features used to form the visual vocabulary as well as the feature extraction. The details of experiments and results are provided in section 5. Conclusion is drawn in the 6th section.

2 PLSA model

Probabilistic Latent Semantic Analysis (PLSA) was proposed by Hofmann in 1999 to solve the problem of ambiguity between words [6]. It is a generative model from the statistical literature [7] which is researched a lot and generates many variations.

In text analysis, it is used to discover topics in a document. Here in scene classification, we treat images as documents and discover categories as topics (such as mountain and road). The model is applied to images by using visual words which is formed by vector quantizing color, shape, texture and SIFT features [4].

A collection of scene images $D=d_1, \ldots, d_N$ with words from a visual vocabulary $W=w_1, \ldots, w_V$ are given. The data in a $V \times N$ co-occurrence table are defined as $N_{ij}=n(w_i, d_j)$, where $n(w_i, d_j)$ denotes how often the word w_i occurred in an image d_j. In PLSA

a latent variable model for co-occurrence data associates an unobserved class variable $z \in Z=z_1, ..., z_Z$ with each observation.

The algorithm of PLSA is as follows:

1. Select a document d_i with probability $P(d_i)$;
2. Pick a latent class z_k with probability $P(z_k|d_i)$;
3. Generate a word w_j with probability $P(w_j|z_k)$;

As a result, an observation pair (d_i, w_j) is obtained, and z_k is discarded. (d, w) is a joint probability model over $N \times V$ which is defined by the mixture:

$$P(d,w) = \sum_{z \in Z} P(d,w,z) = P(d) \sum_{z \in Z} P(w \mid z) P(z \mid d) \tag{1}$$

And then from $P(d,w)=P(d)P(w|d)$, we get

$$P(w \mid d) = \sum_{z \in Z} P(w \mid z) P(z \mid d) \tag{2}$$

$P(w|z)$ is the topic specific distribution and each image is modeled as a mixture of topics $P(z|d)$.The graphical model is represented in Fig. 2.

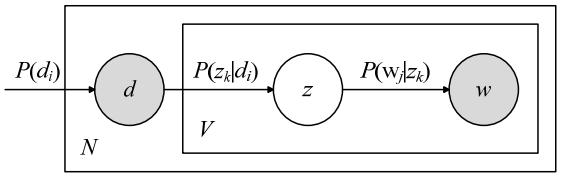


Fig. 2. Graphical model of PLSA

3 KNN-SVM classification

The process of the training mainly includes two steps: In the first step, the probabilistic distributions of topics $P(w|z)$ are learned from the training images. $P(w|z)$ and $P(z|d_{train})$ are determined by applying the PLSA model to the whole set of training images. A Z-vector $P(z|d_{train})$ represents each training images where Z is the number of learned topics. In the second step, KNN-SVM classifier is trained by using the vector $P(z|d_{train})$ of each training image and the class label. In this stage, we propose a hybrid classifier of KNN and SVM to improve the classifier performance.

When classifying the unseen test images, the specific coefficients $P(z|d_{test})$ are computed and then they are used to classify the test image using the KNN-SVM classifier. The unseen image is projected onto the simplex using the $P(w|z)$ learned during the training process. The mixing coefficients $P(z|d_{test})$ are computed such that the divergence between the distribution and $P(w|d_{test})$ is minimized. The EM algorithm is run in similar manner to achieve the result. Now only the coefficients $P(z|d_{test})$ are updated in each M-step with the learned $P(w|z)$ kept unchanged. The test image is then classified by the KNN-SVM classifier. Fig. 3 shows graphically the process for both training and testing.

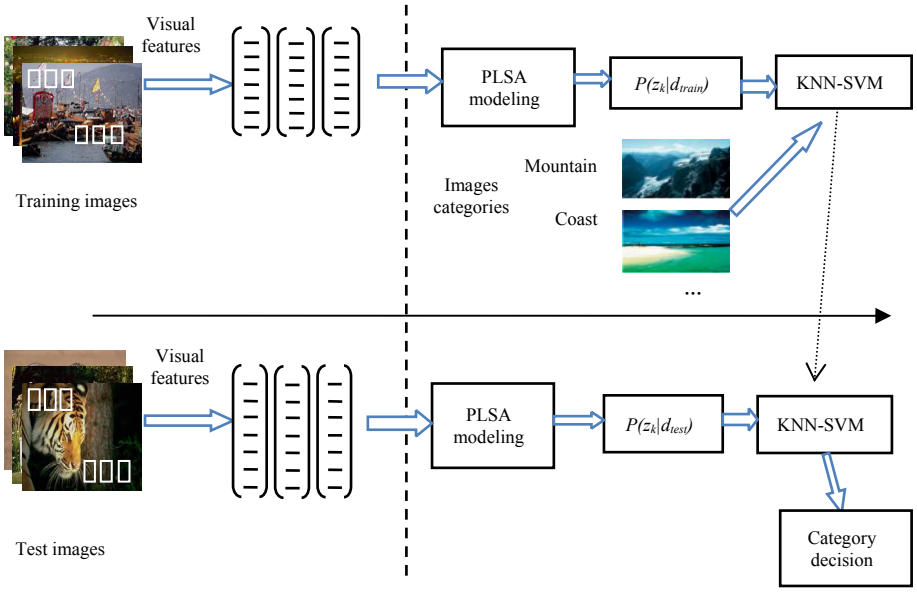


Fig. 3. Overview of visual vocabulary formation, modeling and classification

In detail, the KNN selects K nearest neighbors of the new images in the training dataset using Euclidean distance, and then the test image is classified according to the category label which is fit most in the K nearest neighbors; while for SVM classifier, an exponential kernel *exp-αd* is used, where *d* is the Euclidean distance between the vectors and *α* is the learned weight of the training image example [15]. The perspectives of the KNN-SVM classifier are: The algorithm behaves as KNN classifier and it can easily deal with huge multiclass of the scene images when *K* is small, while it becomes a SVM model when *K* is large [17]. No matter the image dataset is huge or small, the classifier can perform very well.

The algorithm is as follows:

─ Use a crude and simple distance function to find a collection of *Ko* neighbors.

─ Compute the accurate distance on the *Ko* sample images and select the *K* nearest neighbors.

─ Compute (or read from cache if the query repeat) the pairwise distance of the *K* neighbors and the query.

─ Convert the pairwise distance matrix into a kernel matrix.

─ Apply SVM to the kernel matrix to classify the test image and return the result.

4 Spatial and color features extraction

4.1 Spatial feature extraction by chain code

Chain code is a compact way to represent the contour of an object for shape coding and recognition. The chain code histogram (CCH) [8] and minimize sum statistical

direction code (MSSDC) [16] are two methods of all and their advantages are translation and scale invariant. However, they don't take the direction and spatial information into consideration. In the paper, we adopt an advanced method, chain code spatial distribution entropy (CCSDE) [14], to calculate the spatial features because it takes full advantage of the statistical feature, the distribution and the relativity of the chain code sequence [14].

The chain code is defined as follows: for one pixel on the boundary of an image or object, it has n neighbors, numbered from *0* to *n-1*, which are called direction codes. There are 4-direction and 8-direction chain code illustrated in Fig. 4.

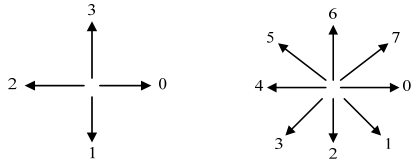


Fig. 4. 4-direction and 8-direction chain code

CCH defines $h_i=n_i/N$, where n_i is the number of chain code with *i*-direction, and N is the number of code links. CCSDE defines I as a contour, $I(x, y)$ as the direction at *the point* (x, y). $A_i = \{(x, y)|(x, y) \in I, I(x, y) = i, 0 \le i \le n-1\}$ denotes the chain codes set with *i* direction. $|A_i|$ is the number of chain codes in set A_i and $C_i\ (x_i, y_i)$ as the centroid with

$$x_i = \sum_{(x,y)\in A_i} x / |A_i| \tag{3}$$

$$y_i = \sum_{(x,y)\in A_i} y / |A_i| \tag{4}$$

Let r_i be the maximum distance from C_i to *i*-direction chain code:

$$r_i = \max_{(x,y)\in A_i} (\sqrt{(x-x_i)^2 + (y-y_i)^2}) \tag{5}$$

With C_i as center and jr_i/M as radius, we draw M concentric circles. Let $|A_{ij}|$ be the count of the chain codes with *i* direction inside *j*th circle. After normalizing, CCSDE is denoted as

$$SE_i(m_i) = -\sum_{j=1}^{M} m_{ij} \log_2(m_{ij}) \tag{6}$$

Combined with CCH, the new feature vector is given as

$$\langle (h_0, SE_0), \cdots, (h_i, SE_i), \cdots, (h_{n-1}, SE_{n-1}) \rangle \tag{7}$$

If we compare the similarity of two contours c_1 and c_2, the similarity can defined as

$$S(c_1, c_2) = \sum_{i=0}^{n-1} \min(h_i^{(c_1)}, h_i^{(c_2)}) \times \frac{\min(SE_i^{(c_1)}, SE_i^{(c_2)})}{\max(SE_i^{(c_1)}, SE_i^{(c_2)})} \tag{8}$$

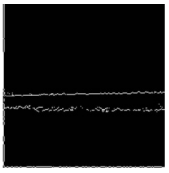
4.2 Color feature extraction

The color features are vital to a scene image but is not sufficient to distinguish similar scenes. To present semantic description better, we combine color feature with the spatial feature. There are a large number of different scene images which have similar structural and spatial features and they are easily classified into the same category using the spatial features only. For instance, Fig. 5 shows two distinctly different scene images of coast and open country but they have similar edge detection images. They both have 3 levels from top to bottom: the former has sky, sea and sands, while the latter has sky, flowers and grassland. The main difference between the two scene images is the color of sands and grassland, instead of the structure and spatial features.

(a) Image of coast

(b) Edge detection image of (a)

(c) Image of open country



(d) Edge detection image of (c)

Fig. 5. Similar edge detection images of different scene images

Computer system can display over 2^{24} colors with 24 bits. We take advantage of this capability to define color data directly as RGB values and eliminate the step of mapping numerical values to locations in a colormap. We define an m-by-n-by-3 matrix to specify truecolor for the scene image. To reduce the number of colors in an image, some functions can be used to approximate the colors in the original image.

4.3 Features combination

In our model, the visual words contain two types: RGB color and spatial description. If the image can be easily classified in a category using only color or spatial feature when training the images, the classification is easy done. On the contrary, we take advantage of both color and spatial feature to classify.

When using color features, the image is split into block regions. The RGB color features are extracted in every block of each image first and then global RGB color features are formed as visual words. They are vectors of RGB values. After the spatial and color features are extracted, the K-means algorithm is applied to cluster these features. Each cluster center corresponds to a visual word and a visual vocabulary includes K visual words. A set of visual words make up an image, which is similar to that the text is consist of words.

5 Experiments and results

According to the previous description, in our model, the scene image dataset divides into two parts. One of them is used to extract spatial and color features and train the parameters for the PLSA model, and another is used to test the model we proposed and do experiments to compare with other models. The experiments and the results display as follows.

5.1 Datasets and parameters setting

In this paper, we choose Oliva and Torralba (OT) and Fei-Fei and Perona (FP) as the image datasets to conduct our experiment.

The OT dataset has 2688 images and is divided into 8 categories: four nature scenes (328 forest, 374 mountain, 360 coasts and 410 open country) and four artifact ones (308 inside cities, 260 highway, 292 streets and 356 tall buildings).The FP dataset has more than 2688 images but it is only available in gray scale. It includes 13 categories: 8 OT categories plus 174 bedrooms, 151 kitchens, 241 residences, 216 offices and 289 living rooms.

When comparing with [9], we use the OT dataset. What's more, we also compare our experiment results with other models based on both datasets.

The values of the latent semantic variables (Z), the number of visual vocabulary (V) in PLSA models and the number of neighbors (K) in KNN classifier are especially important. So, firstly, we choose 100 random scene images from the training set to find the optimal parameters, and the rest of the training images are used to calculate the visual vocabulary and topics of PLSA. However, for the other parameters (the number of block regions etc.), we assign them by referring to the experience or other similar experiments.

5.2 Experiments and classification results

As to the classification result, the performance is measured by the average of precision and recall. The precision is defined as the number of images correctly classified in the topic category divides the total number of images in the same category. While the recall is defined as the number of images correctly classified in their categories divides the number of images which are relative to the topic.

We use a group of different values of K, V and Z to experiment and analysis the effect and the optimal value. To achieve objective variations, we repeat the experiments 5 times with selecting different random training images and test sets. The mean variation curves are displayed below in Fig.6.

According to the figures below, we can observe that we get good performance above *75* percent when the number of visual words $V=700$, the number of latent topics $Z=30$ and the number of neighbors $K=10$.

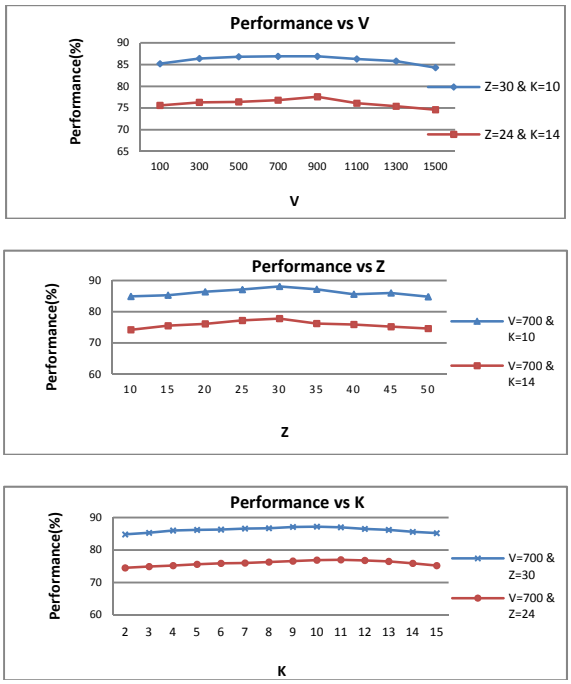


Fig. 6. Performance under variation of different V, Z and K

We apply our methods to the experiment and use the OT dataset, the same one as [9]. The experiment results are comparing in Table 1. The categories of forest, highway, street and tall building have the best result in Ken Shimazaki's experiment. The average is 0.549. However, our proposed model has a better result and almost all 8 categories have a higher mean recall and precision rate.

Table 1. Performance comparison between our model and [9]

Categories	**Coast**	**Forest**	**Mountain**	**Open country**
Mean Recall and Precision	0.516	0.697	0.354	0.409
Our model	0.632	0.799	0.532	0.527
Categories	**Highway**	**Inside city**	**Street**	**Tall building**
Mean Recall and Precision	0.670	0.380	0.766	0.656
Our model	0.783	0.502	0.860	0.837

The experiments using color and spatial features respectively and a hybrid of both are done when extracting image features. We use the KNN-SVM classifier we propose. Table 2 displays the performance for both OT and FP dataset by using color and spatial features. After that, the experiments of different classifiers (KNN, SVM and KNN-SVM) are conducted using the method of extracting both color and spatial features we propose when classifying the scene images. Table 3 shows that our

classifier is improved the classification performance on OT dataset as well as FP dataset.

Table 2. Comparison among color, spatial and both features

# Categories	Color	Spatial	Both (Our model)
8 OT dataset	0.742	0.683	0.830
13 FP dataset	0.736	0.622	0.815

Table 3. Comparison among KNN, SVM and KNN-SVM classifiers

# Categories	KNN	SVM	KNN-SVM (Our model)
8 OT dataset	0.853	0.864	0.871
13 FP dataset	0.724	0.738	0.743

Other researches on scene classification such as LDA [13] and modified PLSA (based on SIFT features) model [2] were proposed. We compare the average performance of our model with them and the results are showed in Table 4.

Table 4. Comparison among PLSA (SIFT), LDA and our model

# training images	128	256	512	1024	1344
PLSA (SIFT)	0.682	0.753	0.795	0.846	0.859
LDA	0.772	0.783	0.794	0.810	0.816
Our model	0.690	0.764	0.798	0.853	0.859

6 Conclusion

We have proposed an improved method to classify scene images using PLSA model. When extracting the features of the scene image, the color and spatial features are combined to obtain full data of the image. The spatial features are calculated using the chain code and color features are represented in RGB space. Then we use the PLSA to model and train some images to get the proper parameters of the model. At last, the other images in the dataset are used to test our proposed model.

According to our experiments, the results certify that our approach is excellent for scene classification. After the comparison with other models, higher mean precision and recall of our method is achieved, although they are not the best. What's more, the scalability and robustness of our model are also excellent. Our future works are improving the method we proposed to enhance the efficiency and conducting more experiments with different parameters to increase our method's performance.

7 Acknowledge

This work is supported by the National Natural Science Foundation of China (Nos. 61165009, 61262005, 61363035, 61365009), the National Basic Research Program of

China (No. 2012CB326403), the Guangxi Natural Science Foundation (Nos. 2012GXNSFAA053219, 2013GXNSFAA019345) and the "Bagui Scholar" Project Special Funds.

8 References

1. I. Biederman. Aspects and extension of a theory of human image understanding. *Computational processes in human vision: an inter-disciplinary perspective*, New Jersey, 1988.
2. A. Bosch, A. Zisserman, X. Munoz. Scene classification via pLSA. In: *Proc. European Conf. Computer Vision (ECCV)*, Vol. 4, pp. 517-530, 2006.
3. A. Bosch, X. Muñoz, R. Martí. Which is the best way to organize/classify images by content?. *Image and vision computing*, 25(6), 778-791, 2007
4. A. Bosch, A. Zisserman, X. Muoz. Scene classification using a hybrid generative/discriminative approach. *Pattern Analysis and Machine Intelligence, IEEE Trans.*, 30(4), 712-727, 2008
5. Shizhi Chen, YingLi Tian. Evaluating effectiveness of latent dirichlet allocation model for scene classification. In: *Wireless and Optical Communications Conference (WOCC), 2011 20th Annual, IEEE*, pp. 1-6, 2011.
6. T. Hofmann. Probabilistic latent semantic indexing. In: *Proc. 22nd annual Int'l. ACM SIGIR Conf. on Research and development in information retrieval*, pp. 50-57, 1999.
7. T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine learning*, 42(1-2), 177-196, 2001
8. J. Iivarinen and A. Visa. Shape recognition of irregular objects. *In: Intelligent Robots and Computer Vision XV: Algorithms, Techniques, Active Vision, and Materials Handling, SPIE*, pp. 25-32,1996.
9. K. Shimazaki, T. Nagao. Scene classification using color and structure-based features. *Sixth International Workshop on Computational Intelligence and Applications (IWCIA), IEEE*, pp. 211-216, 2013.
10. Fei-Fei Li, P. Perona. 2005. A bayesian hierarchical model for learning natural scene categories. In: *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, Vol. 2, pp. 524–531, 2005.
11. A. Oliva, A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001.
12. P. Quelhas, F. Monay, J. M. Odobez, et al. Modeling scenes with local descriptors and latent aspects. *ICCV*, Vol. 1, pp. 883-890, 2005.
13. N. Rasiwasia, N. Vasconcelos. Latent Dirichlet Allocation Models for Image Classification. *IEEE Trans. PAMI*, 35(11): 2665-2679, 2013.
14. Junding Sun, Heli Xu. Contour-Shape recognition and retrival based on chain code. In: *Proc. Computational Intelligence and Security, CIS'09*, Vol. 1, pp. 349-352, 2009.
15. B. Schölkopf, A. J. Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* MIT press. 2002.
16. X.L.Wang and K.L. Xie. A novel direction chain code-based image retrieval. *In Proceedings of the Fourth International Conference on Computer and Information Technology (CIT'04)*, pp. 190-193, 2004.
17. H. Zhang, A. C. Berg, M. Maire, J. Malik. SVM-KNN: Discriminative nearest neighbor classification for visual category recognition. In: *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference* , Vol. 2, pp. 2126-2136, 2006.