A Novel Bag Generator for Image Database Retrieval With Multi-Instance Learning Techniques

Zhi-Hua Zhou, Min-Ling Zhang and Ke-Jia Chen
National Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China
zhouzh@nju.edu.cn {zml,chenkj}@ai.nju.edu.cn

Abstract

In multi-instance learning, the training examples are bags composed of instances without labels and the task is to predict the labels of unseen bags through analyzing the training bags with known labels. In content-based image retrieval (CBIR), the query is ambiguous because it is hard to ask the user precisely specify what he or she wants. Such kind of ambiguity can be gracefully dealt with by multi-instance learning techniques, and previous research shows that bag generators can significantly influence the performance of a CBIR system. In this paper, a novel bag generator named ImaBag is presented, where the pixels of each image are first clustered based on their color and spatial features and then the clustered blocks are merged and converted into a specific number of instances. Experiments show that ImaBag achieves comparable results to some existing bag generators but is more efficient in retrieving images from databases.

1 Introduction

In investigating the problem of drug activity prediction, Dietterich et al. [1] proposed the notion of *multi-instance learning*, in which the training set is composed of many *bags* each containing many instances. The bags are labeled in the way that if a bag contains at least one positive instance then it is labeled as a positive bag. Otherwise it is labeled as a negative bag. The task is to learn some concept from the training bags for correctly labeling unseen bags. The difficulty of multi-instance learning lies in that unlike standard supervised learning where all the training instances are labeled, the labels of the individual instances are unknown in multi-instance learning. Dietterich et al. [1] showed that learning methods ignoring the characteristics of multi-instance problem could not work well in this scenario.

In CBIR, the query, i.e. the example image posed by the user is actually ambiguous and difficult to be perceived. For instance, suppose a user poses the image shown in Figure. 1 and asks the system to retrieve 'similar' images from the database. This kind of query is rather ambiguous since the query can be regarded as 'lake', 'mountains', 'clouds', 'trees', etc., while it is hard to ask the user precisely specify which one he or she really wants. However, if the query can be processed as an image bag that preserves original semantic meanings of the image, then the ambiguity can be tackled by multi-instance learning techniques. That is, by receiving several positive and negative bags, the system can learn what the user requires with multi-instance learning techniques.

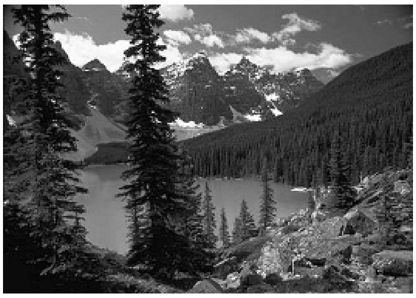

Figure 1: A sample query image

In fact, several multi-instance learning based CBIR systems have been developed [5][6] and achieved nice performance in recent years. It is worth noting that in Maron and Ratan's work [5], several bag generators for transforming images into image bags have been presented and tested, and their results showed that these bag generators significantly affected the performance of the retrieval.

In this paper, a novel image bag generator named *ImaBag* is presented. Experiments show that ImaBag achieves comparable results to some existing generators but is more efficient in retrieving images from the image databases. The rest of this paper is organized as follows. Section 2 briefly introduces the application

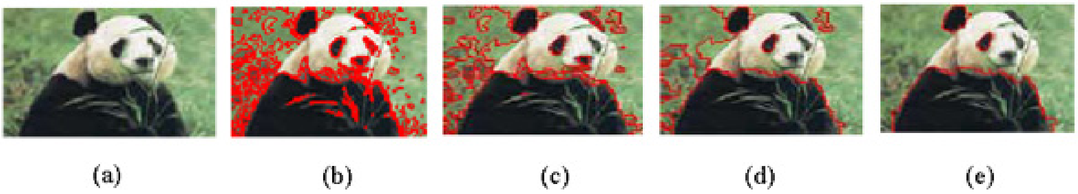

Figure 2: A sample run of ImaBag. The input image is transformed into four disjunctive blocks. The red dots in the pictures show the borders of each remaining region. (a) Original image; (b) After SOM-based clustering; (c) After eliminating isolated pixels; (d) After merging scattered blocks based on size and position; (e) After merging scattered blocks based on color and position.

of multi-instance learning techniques to CBIR. Section 3 presents ImaBag. Section 4 reports the experimental results. Finally, Section 5 concludes and indicates several issues for future work.

2 Multi-Instance Learning

The most famous multi-instance learning algorithm is Diverse Density proposed by Maron and Lozano-Pérez [4]. Maron and Ratan [5] applied Diverse Density to image retrieval by utilizing image color statistics and color distribution patterns. Each image is initially filtered and subsampled to a matrix of 'color blobs'. Then, different bag generators are used to transform various configurations of blobs of each image into feature vectors (instances) of the corresponding image bag. Maron and Ratan [5] showed that simple bag generators are sufficient to work out the particular CBIR problem of natural scene classification. More importantly, they demonstrated that bag generators play a key role in developing a practical multi-instance learning based CBIR system.

Yang and Lozano-Pérez [6] developed and tested another image bag generator. First, each image is divided into many overlapping regions. For each region, the sub-image is filtered and converted into a feature vector and then each image can be represented by a number of feature vectors. A variation of the weighted correlation statistic is used to measure the similarity between feature vectors and Diverse Density is employed in the training phase to learn the target concept together with the relative importance of the features. Yang and Lozano-Pérez [6] illustrated that their system works well on both natural scene and object image databases.

3 ImaBag

Except the multi-instance learning algorithm used for learning target concept and retrieving images, the bag generator employed for transforming images into corresponding image bags is crucial in developing a practical multi-instance learning based CBIR system. In fact, Maron and Ratan [5] indicated that a good bag generator, e.g. one that generates coherent subregions of various sizes of query images that results in a smaller number of instances per bag, might lead to easier learning and precise retrieval.

ImaBag is a novel image bag generator, which is derived from a SOM-based image segmentation technique [2]. At first, the pixels of each image are clustered based on their color and spatial features, where the clustering process is accomplished with a SOM neural network [3]. Then, the clustered blocks are merged into a specific number of regions. Finally, each resulting region is transformed into a 3-dimensional feature vector formed by its mean R, G, B values. It was shown that this segmentation method could transform each image into several image regions that preserve some semantic meanings [2]. It is obvious that through regarding each feature vector corresponding to each region as an instance, the image is easily transformed as a bag for multi-instance learning, which is the way ImaBag goes.

Figure. 2 illustrates a sample run of ImaBag. For an input image such as the one shown in Figure. 2 (a), every pixel is represented by a five-dimensional feature vector where the features are x, y coordinates and R, G, B values of the pixel, respectively. Obviously, the coordinates encode the spatial information of a pixel and the R, G, B values encode its color information. All the feature values have been normalized and then the image can be treated as a collection of feature vectors.

In step 2, every five-dimensional feature vector is fed to a SOM neural network with five input neurons. The output of the SOM neural network is n classes where n is a parameter specified by the user. When the training phase terminates, all the feature vectors are automatically clustered into n different classes and feature vectors that are topologically close are mapped to the same class. At the same time, the input image is divided into a number of regions where the pixels in each region share the same class label. However, pixels belong to the same class are not always connected. As Figure. 2 (b) shows, there may exist a lot of isolated pixels and small blocks. Those small blocks are called *scattered blocks*. Then, the following two steps are performed in order to eliminate the isolated pixels and merge the scattered blocks.

The isolated pixels are eliminated using a gliding window. If an isolated pixel is found in the gliding window, its associated class will be substituted by the most common class occurred in the gliding window. The gliding window moves across the whole image from left to right, and from top to bottom. This process is repeated until there is no isolated pixel remained. For instance, with a 3 × 3 gliding window, Figure. 2 (b) becomes Figure. 2 (c).

The scattered blocks are merged as follows: all the remaining blocks are sorted based on their number of pixels, then the block with the least number of pixel number is identified and merged into its largest neighbor. The process is repeated until there are $2n$ blocks left. Now Figure. 2 (c) becomes Figure. 2 (d).

After that, the mean of R, G, B values of each remaining block is calculated and the distance between each pair of neighboring blocks is computed according to the Euclidean distance of each other's R, G, B mean values. Then the block with the least mean of R, G, B values is identified and merged into its nearest neighboring block. This process is repeated until there are n blocks remained. As shown by Figure. 2 (e), these n blocks are the ultimate result of image segmentation.

Each of the n remaining blocks is transformed into a 3-dimensional feature vector formed by each block's R, G, B mean values. Eventually, the input image is converted into a corresponding image bag consisting of n 3-dimensional feature vectors (instances).

4 Experiments

4.1 Experimental Setup

An image database consisting of 500 images is used in the experiments, which includes 100 images from each of the five image types: *cats&dogs*, *flowers*, *mountains*, *planes*, and *buildings*. A *potential training set* of 100 images is created by randomly choosing 20 images from each of the five image types. The remaining images constitute a *test set* consisting of 400 images, 80 from each of the five image types. In this paper, each image type corresponds to a concept class to be learned. Note that the purpose of the experiments is to compare the performance of different bag generators instead of testing the whole image retrieval system, thus a medium-sized image database with 500 images is sufficient to fulfill the objective. However, in order to get the statistics of the whole retrieval performance, experiments on a larger image database is necessary, which is left for future work.

For each image type, an *initial training set* is created by randomly picking several positive examples of the target concept and several negative examples, all from the potential training set. A bag generator is chosen, and then the concept is learned by Diverse Density [4]. After the concept has been learned, the 400 images in the test set are sorted by the distance from the learned concept [5]. In this paper, two different training schemes are used: *5p5n* that picks 5 positive examples and 5 negative examples to form the initial training set; and *10p10n* that selects 10 positive examples and 10 negative examples for training.Figure. 3 shows a sample run of ImaBag with *5p5n* where the target concept is the type *cats&dogs*.

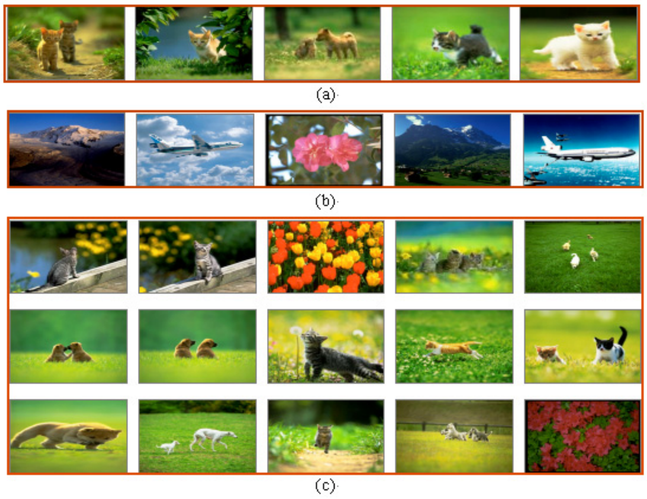

Figure 3: A sample run of ImaBag for retrieving *cats&dogs* with training scheme *5p5n*. (a) User-selected positive examples; (b) User-selected negative examples; (c) Final retrieval from test set (top 15 images).

4.2 Compared Bag Generators

In the experiments, Maron and Ratan [5]'s and Yang and Lozano-Pérez [6]'s bag generators are compared with ImaBag.

Maron and Ratan [5] smoothed each image using a Gaussian filter and subsampled the image to an 8 × 8 matrix of *color blobs* where each blob is a 2 × 2 set of pixels within the 8 × 8 matrix. Then, seven different bag generators are used to transform various configurations of blobs of each image, such as *rows*, *single*

Table 1: Results of training scheme *5p5n*. The number following '±' is standard deviation.

Image type	ImaBag		Maron & Ratan's SBN		Yang & Lozano-Pérez's	
	precision	recall	precision	recall	precision	recall
cats&dogs	.469±.094	.586±.118	.679±.026	.848±.033	.321±.070	.401±.087
flowers	.548±.072	.685±.091	.623±.089	.779±.111	.284±.057	.355±.071
mountains	.421±.077	.526±.096	.469±.073	.586±.091	.357±.050	.446±.062
planes	.321±.138	.401±.173	.394±.113	.493±.141	.244±.052	.305±.065
buildings	.311±.057	.389±.071	.404±.091	.505±.114	.217±.102	.271±.128
average	.414±.088	.517±.110	.514±.078	.642±.098	.285±.066	.356±.083

Table 2: Results of training scheme *10p10n*. The number following '±' is standard deviation.

Image type	ImaBag		Maron & Ratan's SBN		Yang & Lozano-Pérez's	
	precision	recall	precision	recall	precision	recall
cats&dogs	.535±.111	.669±.139	.705±.018	.881±.023	.335±.066	.419±.083
flowers	.590±.100	.738±.125	.728±.066	.910±.082	.279±.044	.349±.056
mountains	.444±.098	.555±.122	.526±.044	.658±.055	.362±.043	.453±.054
planes	.341±.074	.426±.092	.423±.056	.529±.070	.274±.020	.343±.025
buildings	.351±.045	.439±.057	.415±.055	.519±.069	.219±.125	.274±.157
average	.452±.086	.565±.107	.559±.048	.699±.060	.294±.060	.368±.075

blob with neighbors, *two blobs with no neighbors*, etc., into feature vectors (instances) of the corresponding bag. In this paper, the bag generator SBN, i.e. *single blob with neighbors*, is chosen. An SBN is defined as the combination of a single blob with its four neighboring blobs (up, down, left, right). The sub-image is described as a 15-dimensional vector, where the first three attributes represent the mean R, G, B values of the central blob and the remaining twelve attributes correspond to the differences in mean color values between the central blob and other four neighboring blobs respectively. Therefore, each image bag is represented by a collection of nine 15-dimensional feature vectors obtained by using each of the nine blobs not along the border as the central blob.

Yang and Lozano-Pérez [6] transformed color images into gray-scale images at first. Then, they divided each image into many overlapping regions. For each region, the sub-image is filtered and converted into an $h \times h$ matrix and treated as an h^2-dimensional feature vector. The weighted correlation coefficient is used to measure the similarity between feature vectors, and each feature vector is further transformed into an equivalent one for the sake of fitting into Euclidean space. In this paper, each image bag generated in this way is formed by a set of forty 64-dimensional feature vectors obtained by dividing each image into forty overlapping regions and setting h to be 8.

4.3 Results

One way to evaluate image retrieval performance is to measure the *precision* and *recall*. Precision is the ratio of the number of correctly retrieved images to the number of all images retrieved so far. Recall is the ratio of the number of correctly retrieved images to the total number of correct images in the test set. For either of the training schemes and every image type, 10 runs are performed for each of the three different bag generation techniques, i.e. Maron and Ratan's SBN, Yang and Lozano-Pérez's method ($h = 8$), and ImaBag ($n = 4$). Here the precision and recall are calculated on the first 100 images sorted by the learned concept upon each run. For each combination of bag generator and image type, Tables 1 and 2 report the average value of precision and recall of 10 runs, where the number follows '±' is the corresponding standard deviation.

According to Tables 1 and 2, it is obvious that the results of ImaBag are worse than that of Maron and Ratan's SBN, but are much better than that of Yang and Lozano-Pérez's bag generator. One possible reason for the inferior performance of Yang and Lozano-Pérez's generator is that, in this method, all images must be transformed into gray-scale images such that some important information may be discarded. Furthermore, it is clear from Tables 1 and 2 that the performance of all the three generators improves with more example images (*10p10n* is better than *5p5n*).

Note that here each image is converted into a small bag with four 3-dimensional instances by ImaBag, but there are nine 15-dimensional instances and forty 64-dimensional instances in each image bag generated by Maron and Ratan's bag generator and Yang and Lozano-Pérez's bag generator, respectively. Due to the simplicity of the bags generated by ImaBag, the retrieval process runs several times faster with ImaBag than with other two bag generators when Diverse Density is employed in training and retrieving. It is noteworthy that such kind of reduction in running time is very desirable for practical CBIR systems, especially when dealing with a real-world large image database.

5 Conclusion

Multi-instance learning techniques can be incorporated into CBIR systems to deal with the ambiguity existing in the user queries. One of the keys in developing a practical multi-instance learning based CBIR system is to obtain a nice bag generator. In this paper, a bag generator named ImaBag is presented. At first, the image pixels are clustered based on their color and spatial features, where the clustering process is accomplished by a SOM neural network. Then, the clustered blocks are transformed into a specific number of regions by eliminating isolated pixels and merging scattered blocks. Finally, the resulting regions are converted into 3-dimensional numerical instances of the image bag formed by their mean R, G, B values.

It should be noted that the experiments reported in this paper are preliminary because only a small image database is used. Although the experimental results show that ImaBag is promising in helping achieve relatively good retrieval results while using significantly smaller time cost, more experiments on larger image databases are needed for exploring its strength and weakness, which is left for future work.

At present, only the color and spatial properties of an image are utilized in ImaBag. It is obvious that enabling ImaBag employ more features such as texture and structural characteristics is an interesting issue for future work. Another deficiency of ImaBag is that the number of instances contained in each bag is a predefined constant. Therefore, endow ImaBag with the ability of automatically determining the suitable number of instances for each image bag is another interesting issue for future work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under the Grant No. 60105004, the Natural Science Foundation of Jiangsu Province under the Grant No. BK2001406, and the National 973 Fundamental Research Program of China under the Grant No. 2002CB312002.

References

[1] T.G. Dietterich, R.H. Lathrop, and T. Lozano-Pérez, "Solving the multiple-instance problem with axis-parallel rectangles," *Artificial Intelligence*, vol.89, no.1–2, pp.31–71, 1997.

[2] Y. Jiang, K.-J. Chen, and Z.-H. Zhou, "SOM-based image segmentation," in *Lecture Notes in Artificial Intelligence 2639*, G. Wang, Q. Liu, Y. Yao, and A. Skowron, Eds. Berlin: Springer, pp.640–643, 2003.

[3] T. Kohonen, *Self-Organizing Maps*, 2nd edition, Berlin: Springer, 1997.

[4] O. Maron and T. Lozano-Pérez, "A framework for multiple-instance learning," in *Advances in Neural Information Processing Systems 10*, M.I. Jordan, M.J. Kearns, and S.A. Solla, Eds. Cambridge, MA: MIT Press, pp.570–576, 1998.

[5] O. Maron and A.L. Ratan, "Multiple-instance learning for natural scene classification," in *Proceedings of the 15th International Conference on Machine Learning*, Madison, WI, pp.341–349, 1998.

[6] C. Yang and T. Lozano-Pérez, "Image database retrieval with multiple-instance learning techniques," in *Proceedings of the 16th International Conference on Data Engineering*, San Diego, CA, pp.233–243, 2000.