Feature Annotation for Visual Concept Detection in ImageCLEF 2008

Jingtian Jiang, Xiaoguang Rui, Nenghai Yu
MOE-Microsoft Key Laboratory of Multimedia Computing and Communication
Department of EEIS, University of Science and Technology of China

silyt@mail.ustc.edu.cn, davidrui@gmail.com, ynh@ustc.edu.cn

Abstract

This paper shows our work on CLEF 2008. Our group joined the Visual Concept Detection Task of ImageCLEF 2008 this year. We submitted one run (run id: HJ_FA) for the evaluation. In the run, we applied a method called “Feature Annotation” to detect visual concept for the predefined concepts and we want to know how this information help in solving the photographic retrieval task. The applied method selected high level features for each concept from both local and global features, based on which the visual concepts are detected. The applied method consists of three procedures. First, feature extraction in which both local and global features are extracted from images. Then, a clustering algorithm is applied to “annotate the features”. In this procedure, the features are affiliated with their corresponding concepts. Finally, we applied KNN algorithm to classify tests images according to the training images with the annotated features. The experiments were performed on the given training and test data on the 17 concepts. The paper concludes with an analysis of our results. Finally we identify the weaknesses in our approach and ways in which the algorithm could be optimized and improved.

Categories and Subject Descriptors:

H.3 [Information Storage and Retrieval]: H.3.1 Content Analysis and Indexing; H.3.2 Information Storage; H.3.3 Information Search and Retrieval; E.1 [Data Structures].

General Terms

Measurement, Performance, Experimentation

Keywords

Image retrieval, image classification

1. Introduction

This paper presents an approach for visual object detection using “feature annotation” which can integrate visual features with semantic concepts. We evaluated the method on the ImageCLEF Photo IAPR TC-12 photographic collection.

This paper is organized as follows: section 2 discusses the dataset characteristics. Sections 3 to 5 present our method including feature extraction, feature annotation and model training. Results are presented in section 6 and finally section 7 presents the conclusions.

2. ImageCLEF Data

ImageCLEF dataset for Visual Concept Detection includes 1,800 training images and 1,000 test images. Although the data count is not very large, it can test our approach for image retrieval task. The images vary in quality, levels of noise, and illustrate several concepts, actions or events. And they often contain several concepts.

3. Feature Extraction

As we know, there are many types of features that can present an image. And they perform variably in different tasks. After surveying on many features, we selected SIFT [1] and Color [2] feature, because of their good performance and simplicity.

SIFT is a transformation, which transforms an image into a set of features with scale invariance. It first detects the characteristic points in the scale space, and fixes the points' positions and the scale of the positions. Then for each point, it makes use of the gradient of the point's neighborhood pixels to compute the point's main direction, thereby achieve the scale invariance and direction invariance. Finally, it constructs the characteristic descriptor for each point in order to match the characteristic points for different images. Then a set of features are extracted from the image.

The color feature is simpler relatively. But there is a place worth the whistle in the color feature extraction. Unlike the common method, we first segment an image into 256 little images, and then extract color features from the 256 small images instead of the original images. The reason is as follows. An image usually contains several concepts other than one, so we can imagine that only one feature from the original image may not represent well all the concepts in it. Therefore we part the image into smaller images and extract much more features, and in this way the different features may represent different concepts in an image.

Therefore, we extract the SIFT feature which has 128 dimensions for each key points, and an image has about 300 features (128 dimensions); the color feature which has 64 dimensions with an image 256 features (64 dimensions).

4. Feature Annotation

The common visual features can only represent low-level information, leaving alone the high-level semantics of the images. We propose a simple method to assign semantics to visual features, which is called "Feature Annotation". "Feature Annotation" aims to annotate different region features with semantic concepts. For example, for SIFT features, we want to know which concept the features of each key point represent. It has two steps: the first step aims to annotate features in an image, and the other step aims to annotate features in images in one class. It is described as follows.

Step I:

Generally, after extracting visual features, there are much more features than concepts in an image. As we discover in the experiment, there are about 5 concepts in an image, while there are hundreds of features in one image. So there should be a variety of features corresponding to one concept. On intuition, the features of an image representing the same concept would get together while features representing different concepts would scatter. Also there may be some noisy features that don’t belong to any cluster, because different concepts may have similar features.

In such a case, we adopt a clustering algorithm – DBSCAN[3]. DBSCAN clusters a set of features based density into one or more classes and noise, so it fits this situation well. By using this algorithm, we can cluster the features of one image into several classes. And it is best of all that the number of generated classes is equal to that of concepts in the image. Thus in our experiment, the algorithm can modify its parameters by itself in order to generate a right number of classes. In addition, the features of one image gathered representing the same concept is replaced with their cancroids. In this way, the several hundred of features are transformed into several features of one image, and we can intuit that a feature of an image stands for a concept in the image.

Step II:

On the same intuition in the first step, the features of different images representing the same concept would be gathered. As a feature of an image stands for a concept in the image, there is a fact that, out of all the features of the training images, the number of features representing the same concept is equal to the number of images containing the concept corresponding with the features. It’s a useful clue.

We use the DBSCAN algorithm again. However, this time the clustering should be performed on each concept while the clustering is performed on each image in the first step. For each concept, the input features are all the centric features of images that contain this concept. And the output is a unique class and noise features. Based on the fact, the number of features of the unique class should be equal to that of images used for the input. Then we can conclude that the features of that class just represent the concept. So we annotate these features with that concept.

When the clustering is performed on all the 17 concepts, we have combined the features with their corresponding concept. So finally, for each concept, there are a variety of features combined to it, and the number is equal to the number of images containing this concept.

Note that the SIFT feature and color feature are processed in this step respectively. Thus at last, we have two set of features. One is the SIFT feature, the other is the color feature. They are all “annotated”.

5. Model Training

There are so many machine learning algorithms to train a model for classifying concepts, just like Bagging, Logic Regression, and SVM, and so on. In our work, we select K-Nearest-Neighbors algorithm (KNN)[4]. There are two reasons for this selection. First, the algorithm is easy to implement. Second, this algorithm supports incremental learning. This character is important that we can retrain the model easily when we have new training images. Just for our test experiment, SVM may lead to better performance, and that may be our future work.

For KNN algorithm, the annotated feature is just the model. For a test image, SIFT feature and Color feature are extracted from it first, then the features are clustered using DBSCAN algorithm, and we get two set of features for the test image. After that, these two set of features are classified by KNN algorithm, so we can get which concepts the features belong to. Finally, we assign the concepts to the test image.

6. Results & Discussion

After training, we get about 9 thousand of annotated features, about five times of the training images, which has 1827 images. Then we annotate the test images using them. The number of the test images is about 1000.

In our method, the parameters of the two algorithms are important. Their values should be selected seriously and accurately. In the experiment, we use validation data to tune the system. Ultimately, we set minPts of DBSCAN to 5, while its Eps could be adjusted by itself, and K of KNN algorithm is set to 25. Throughout the whole experiment, the similarity of features is measured with the Euclidean distance. We have experimented with other distances, but their performance is not good than Euclidean distance.

By testing on the 1000 images, its average EER value is 45.07 and average AUC value is 19.96. So its performance is not good, and some concepts did not be detected completely. In such a state of affairs, our method has a trend that, the more common a concept is, the easier it to be assigned to an image. We think there are two reasons for that. First, the number of features of concepts varies. In our experiment, there are about 1600 images contain the concept “outdoor” while the number of images containing the concept “indoor” is about 300. That leads the system to prefer the common concept to the uncommon. Second, the KNN algorithm is to be blamed. The KNN algorithm assigns the class which has most features out of the K nearest features to the new feature. So, the value of K is important, especially for the case that different classes has different amount of features. In an extreme case, if K is greater than the number of features of one class, then this class will never be selected. Thus, maybe KNN is not a proper classification algorithm here. Additionally, our KNN cannot output probabilistic values, so it may fail when evaluating by ROC.

We also evaluate the result by other evaluation criterion such as the Precision and Recall, the result is just better. The precision reaches 75% while the recall 67% over all concepts. But for each concept, there is a similar problem as above. The common concepts get higher recall much more easily, and the uncommon are prone to get lower recall. The precision is just the reverse that the common concepts’ precision is a little lower than that of the uncommon. The reasons have been discussed as above.

7. Conclusions

In this paper, we proposed a simple supervised method to detect concepts from an image. It makes use of the SIFT feature and Color feature. The DBSCAN algorithm is applied to “annotate” features of training images, while the KNN algorithm to classify the annotated features of the test images. Then the test images are assigned to the concepts which their features belong to. The Euclidean distance is applied throughout the whole experiment. However, unfortunately, because of its simplicity, its performance is not good. We also discussed the reasons, and we believe that, with some right modification, the performance could be improved.

Acknowledgments

The research is supported in part by National Natural Science Foundation of China (60672056) and USTC Postgraduate Innovation Foundation.

Reference

[1] Lowe, D. G., Object recognition from local scale-invariant features, Proceedings of International

Conference on Computer Vision, 1999, pp. 1150-1157.

[2] Gevers T. and Aldershoff F., Color feature detection and classification by learning, In Proceedings IEEE International Conference on Image Processing (ICIP), 2005.

[3] Ester M., Kriegel H.-P., Sander J., Xu X., Density-Connected Sets and their Application for Trend Detection in Spatial Databases, Proceedings of International Conference on Knowledge Discovery and Data Mining(KDD ‘97), Newport Beach, CA, AAAI Press, 1997, pp. 10-15.

[4] http://people.revoledu.com/kardi/tutorial/KNN/index.html