Fudan University at TRECVID 2010:

Semantic Indexing

Xiangyang Xue, Hong Lu, Renzhong Wei, Lei Cen, Yao Lu, Yingbin Zheng

Shanghai Key Laboratory of Intelligent Information Processing

School of Computer Science, Fudan University, Shanghai, China

1 Introduction

In this notebook paper we describe our participation in the NIST TRECVID 2010 evaluation. We took part in semantic indexing task of benchmark this year.

For semantic indexing, we submitted 3 automatic runs using only IACC training data:

Fudan.TV10.3: this run is based on visual features of keyframes.

Fudan.TV10.2: this run is based on visual features of keyframes and object detection.

Fudan.TV10.1: this run is based on Fudan.TV10.2 and metadata of video.

2 Semantic Indexing

For high-level feature extraction task, we principally focus on:

(1) Context-based concept fusion based on the initial results of visual features.

(2) Object detection by region of interest (ROI) extraction.

(3) Video metadata extraction for concept detection.

2.1 Visual Features

This year we explore the same visual features as our HLF system last year [1], i.e., global visual features (MPEG-7 descriptors [2]) and local features (SIFT feature [3] and bag-of-visual-words [4]).

We extract six MPEG-7 visual features [1] at global scale for each keyframe of the video shots. The features are: Color Layout Descriptor (CLD, 12 dims), Color Structure Descriptor (CSD 256 dims), Scalable Color Descriptor (SCD, 64 dims), Homogenous Texture (HT, 62 dims), Edge Histogram Descriptor (EHD, 80 dims), and Region Shape (RS, 35 dims).

For each keyframe in the dataset, we extract local features using dense sampling and SIFT descriptor [3]. We use the local feature implementation of [5]. A codebook vocabulary $V= \{v_1, v_2, \ldots, v_n\}$ of SIFT points is constructed through k-means clustering of the local features. In our experiment we choose n=1000. Then the keyframe can be described as a bag of visual words (BoW) [4]. A codebook histogram is obtained for each keyframe with each bin representing a codeword v_i in V. Besides the standard BoW model, we also incorporate spatial information by selecting the 2*2 grid and 1*3 grid to represent the layout in last year’s paper [1].

2.1.1 Context-Based Concept Fusion

In this part, we introduce a framework based on constructing context spaces of concepts to improve the performance of concept detectors [6]. Different from traditional CBCF approach, we present two kinds of such context spaces: explicit context space for modeling the correlation of pairwise concepts, and implicit context space for representing latent themes trained from a set of concepts. The final concept detection scores are then directly fused from explicit and implicit context spaces.

Given a semantic lexicon of *m* concepts $C = \{c_1 , c_2 , ..., c_m \}$ and a video dataset of *n* shots $X = \{ x_1 , x_2 , ..., x_n \}$, concept detection aims to give prediction scores of the concepts to each shot. We define the ground-truth label and prediction score of video shot *k* for concept *i* as y_{ik} and s_{ik}, respectively. And the goal of context-based concept fusion is to generate the refined score s'_{ik} that improves concept detection results.

For explicit context space, we construct the correlation graph based on the annotation of training data. Pearson product moment correlation (*PM*) is applied to model the correlation:

$$PM(c_i, c_j) = \frac{\sum_{k=1}^{n}(y_{ik} - \mu_i)(y_{jk} - \mu_j)}{(n-1)\sigma_i\sigma_j}$$

where $y_{ik} = 1$ indicates shot *k* annotated with concept c_i, otherwise $y_{ik} = 0$; μ_i and σ_i are the mean and standard deviation of $\{ y_{ik} | k = 1, 2, ..., n\}$, respectively. The *PM* value ranges between +1 and -1.

Then we directly enhance detection scores by weighted fusing scores of the most *P* positive and *N* negative-correlated concepts:

$$s_{ik}^{E} = \sum_{j \in C_p} w_{ij} s_{jk} + \sum_{j \in C_n} w_{ij} s_{jk}$$

For implicit context space, our method aims to find some “latent theme” with the ability to model relation of concepts. We use the sparse coding algorithm for constructing implicit context space. Learning sparse codes can be formatted as the following optimization problem:

$$\begin{aligned} \min_{B,f} \quad & \sum_{k=1}^{n} ||s_k - Bf_k||_2^2 + \lambda ||f_k||_1 \\ s.t. \quad & \sum_{i} B_{ij}^2 \leq 1, \ \forall j = 1, 2, ..., n \end{aligned}$$

where *B* is *m* × *d* matrix, column vectors of *B* are the sparse coding basis, *d* is the number of basis, and f_k is the transformed feature vector for shot *k*.

Then SVM with linear kernel is selected as the second-layer learner. The transformed features f_k corresponding to shot x_k are the input of SVM. Each testing shot x_k is given a prediction p_{ik}. The final results of implicit context space are the normalized output of predictions:

$$s_{ik}^{I} = \frac{p_{ik} - \mu_i^p}{\sigma_i^p}$$

Where $\mu^p{}_i$ and $\sigma^p{}_i$ are the mean and standard deviation of { p_{ik} |k = 1, 2, ..., n}, respectively.

To obtain better detection which combines the two context spaces, we apply the average fusion strategy:

$$s'_{ik} = s_{ik} + \alpha \cdot s_{ik}^{E} + \beta \cdot s_{ik}^{I}$$

For more detail information of this method, reader can refer to [6].

2.2 Object Detection by Region of Interest (ROI) Extraction

In this part, we introduce our high level feature extraction method by combining region of interest (ROI) extraction. We aim to improve the performance of detecting certain concepts by the ROI method proposed in [7]. Most of the concepts we are going to detect are animals and vehicles, such as birds, cows, car, etc. We use Harris-Laplace detector [8] and SIFT descriptor [3], and follow the Bag-of-visual-words [4] representation framework. Then we learn discriminative words for each concept. In test images, we give each image a confidence score of each concept using the discriminative words we obtained.

The framework of our proposed contextual model for ROI extraction is shown in Figure 1 [7].

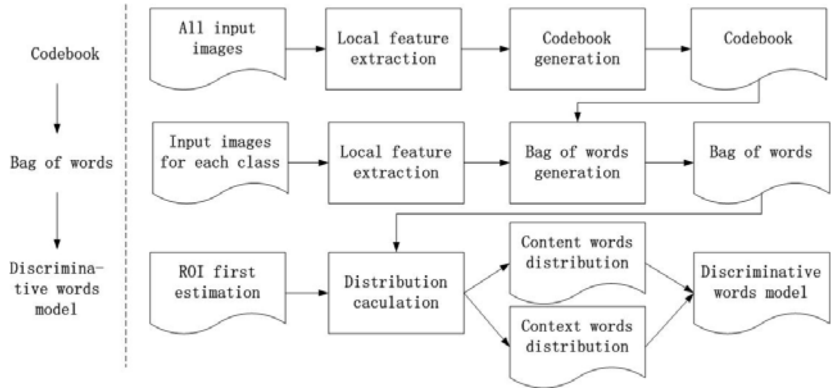

Figure 1. Framework for ROI extraction.

2.2.1 Feature Extraction and Bag-of-Visual-Words Representation

We extract 128 dimensional SIFT feature descriptor on interesting points which are detected by Harris-Laplace point extractor. We select 20,000 images, and randomly select 100,000 feature points from these images. A codebook vocabulary with size 3,000 is generated by *k*-means clustering algorithm. Then each feature point is quantized to assign its descriptor to the nearest cluster. Each image can then be represented as a feature map with the same size as ordinary image, where each pixel indentify the codebook index if there is a feature point, or zero if there is not.

2.2.2 Concept Modeling

To model the concept we are going to detect, first we need to annotate some positive samples with bounding box as train samples. For each concept, we annotate 100 samples with bounding box as training samples. For these samples, we extract visual words histogram both inside bounding box and outside bounding box using Equ. (1-1). In Equ. (1-1), O_i represents the words histogram inside the bounding box which is referred to as content area, C_i represents the words histogram outside the bounding box which is referred to as context area.

$$D(w_i) = \frac{|O_i - C_i|^2}{O_i + C_i} \cdot \mathrm{sign}(O_i - C_i) \qquad (1\text{-}1)$$

We sort the words according to $D(w_i)$ values for each concept. The visual words with high $D(w_i)$ is more likely to appear in object than that appear in context. We choose top K visual words with the highest $D(w_i)$ as discriminative words for the one specific object class as our proposed contextual model. K is set to 64 based on our empirical study.

2.2.3 Testing

For a given image, we first do Bag-of-visual-words representation as that for training images. Assume in the image there exist certain concept, such as car, and we have obtained discriminative words in concept modeling step. By adding all the $D(w_i)$ (which) value together for detected points, we can obtain a final confidence to represent whether the image contains the object or not.

By adding the $D(w_i)$ values (K words which corresponding to specific object class) together for detected points, we can obtain a final confidence to represent whether the image contains the object or not.

2.3 Video Metadata Extraction

In this part, we propose to make use of text information in a straightforward way. We use *TF-IDF* feature to represent each video, and cosine similarity on the feature space to measure the similarity of two shots. The annotations of a video are summarized into a histogram or a distribution across all 130 features. For each test video, 50 most similar video in training set is found via *K*-nearest neighbor method. The annotation distributions of the 50 nearest neighbors are averaged to obtain a distribution, which indicates how possible a shot of each feature may be contained in this test video. It needs to be note that, since the text sources pertain to the video instead of the shot, no specific prediction is made to the shots; the resulted distribution only represents the video.

2.4 Experiments

We use TRECVID 2010 collaborative annotation organized by LIG and LIRIS to train our models. The classifiers of visual features are trained by libSVM package [9]. The kernel for global features is RBF, while that for local features is chi-square kernel [10]. For each video shot in the testing set, we extract 3 keyframes to capture more information. The maximum score of the 3 keyframes is determined as the final score of the shot. For fusion of different features and different models, we apply linear weighted fusion method for its simple and efficiency.

We submitted a total of 3 automatic runs using only IACC training data. The description and infAP of each run are shown in Table 1.

Table 1. Description and infAP of our runs.

Run	infAP	Description
Fudan.TV10.3	0.027	visual features (with CBCF)
Fudan.TV10.2	0.025	visual features (without CBCF) + object detection
Fudan.TV10.1	0.025	visual features (without CBCF) + object detection + video metadata

3 Acknowledgement

This work was supported in part by 973 Program (Project No. 2010CB327900), National High Technology Research and Development Program of China (No. 2009AA01A346), Natural Science Foundation of China (No. 60875003 and 60873178), 211-Project Sponsorship for Young Professors.

References

1. Xue, X., Zheng, Y., Liu, H., Wen, Z., Guo, X., Wei, R., Lu, H., Zhang, Q.: Fudan University at TRECVID 2009:High-Level Feature Extraction and Copy Detection. In TRECVID Workshop (2009)
2. Smith, T.F., Waterman, M.S.: Color and Texture Descriptors. IEEE Transactions on Circuits and Systems for Video Technology, 11(6): 703--715 (2001)
3. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision 60(2): 91--110 (2004).
4. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In International Conference on Computer Vision, vol. 2: 1470--1477 (2003).
5. van de Sande, K. E. A., Gevers, T., Snoek, C. G. M.: Evaluating Color Descriptors for Object and Scene Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 32(9): 1582--1596 (2010)
6. Zheng, Y., Wei, R., Lu, H., Xue, X.: Semantic Video Indexing by Fusing Explicit and Implicit Context Spaces. In ACM International Conference on Multimedia (2010)
7. Wei, R., Lu, H., Zheng, Y., Cen, L., Jin, C., Xue, X., Wu, W.: How Context Helps: A Discriminative Codeword Selection Method for Object Detection. In ICIP, 3905-3908 (2010)
8. Mikolajczyk, K., Schmid, C.: Scale & Affine Invariant Interest Point Detectors. International Journal of Computer Vision 60(1): 63--86 (2004).