PKU-ICST at TRECVID 2014: Instance Search Task

Yuxin Peng, Jian Zhang, Panpan Tang,

Lei Huang, Xin Huang and Xiangteng He

Institute of Computer Science and Technology,

Peking University, Beijing 100871, China.

pengyuxin@pku.edu.cn

Abstract

We participate in all two types of instance search task in TRECVID 2014: automatic search and interactive search. This paper presents our approaches and results. In this task, we mainly focus on exploring the effective feature representation, feature matching and re-ranking algorithm. In feature representation, we extract three types of features: (1) Two basic visual features CMG and EOH, (2) Bag-Of-Words feature representation based on K-means, (3) Bag-Of-Words feature representation based on AKM (Approximate K-means). We combine them to represent the frame image effectively. In feature matching, multi-bag SVM is adopted since it can make full use of all query examples. Moreover, we conduct keypoint matching algorithm on the top ranked results. It is effective yet efficient since only top ranked results are considered. In re-ranking stage, we also incorporate transcripts into our framework to explore context information. Official evaluations show that our team is ranked 1st on interactive search and 4th on automatic search.

1 Overview

In instance search task of TRECVID 2014[8], we participate in all two types: automatic search and interactive search. We submitted 5 runs for the instance search task of TRECVID 2014, including 4 runs for automatic search and 1 run for the interactive search. The evaluation results of our 5 runs are shown in Table 1.

Table 1: Results of our submitted 5 runs on Instance Search task of TRECVID2014.

Type	ID	MAP	Brief description
Automatic	F_D_PKU-ICST_1	0.2108	K2+M+R
	F_D_PKU_ICST_2	**0.2321**	B+K1+K2+M+R
	F_E_PKU-ICST_4	0.1882	B+K1+M+R
	F_D_PKU-ICST_5	0.1810	B+K1+M+R
Interactive	I_D_PKU-ICST_3	**0.3169**	K2+R+H

In automatic search, our team is ranked 4th in all 22 teams. In interactive search, our team is ranked 1st. Table 2 gives the explanation of brief description in Table 1. The framework of our system for instance search task of TRECVID 2014 is shown in Figure 1. Among the 4 automatic search runs, the difference between Run4 and Run5 is that Run4 uses the video examples provided this year (defined as example set E), and Run2 is the late fusion of Run1 and Run5.

Table 2: Description of our methods.

Abbreviation	Description
B	**B**asic feature
K1	**K**eypoint based feature based on K-means, 1,000D
K2	**K**eypoint based feature based on AKM, 1,000,000D
M	Keypoint **m**atching
R	**R**e-ranking based on text matching
H	**H**uman feedback

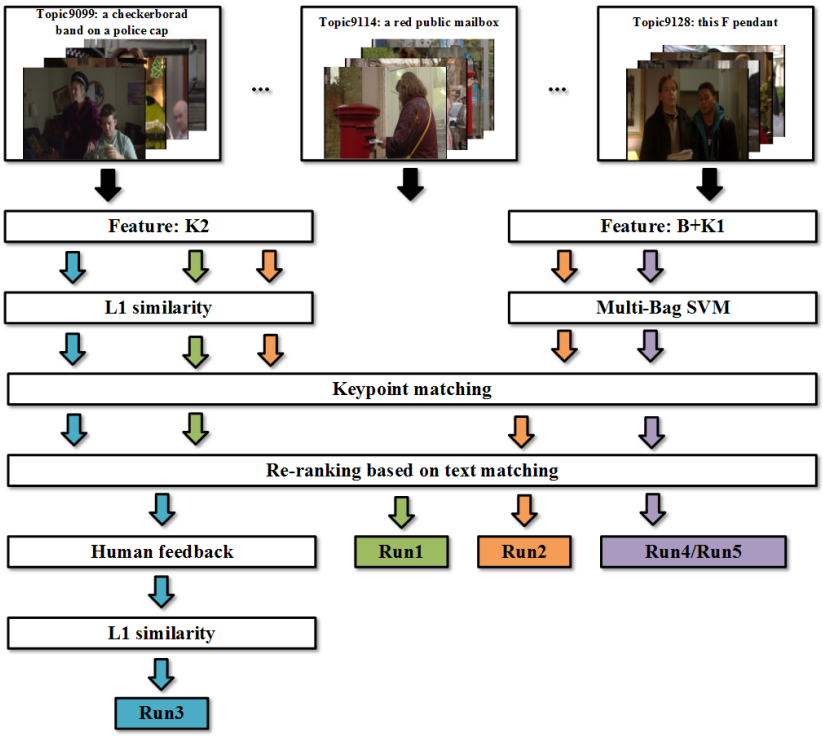

Figure 1: Framework of our instance search approach for the submitted five runs.

2 Feature Representation

We use three kinds of features for the instance search tasks, namely basic visual features, BoW features based on K-means and BoW features based on AKM.

2.1 Basic visual features

We extract two basic visual features namely CMG (Color Moment Grid) and EOH (Edge Orientation Histogram) from each keyframe image. The details of these visual features are given as follows:

(1) **CMG** (756-d): the image is divided into sub-images by 1x1, 3x3, 5x5 and 7x7 grid in the CIE-Lab color space. The color moments of the 1st, 2nd and 3rd order are extracted from these sub-images in each channel.

(2) **EOH** (657-d): we use sobel operator to detect the edges, then the image is divided into sub-images by 3x3 grid from each sub-image, the edge directions are evenly quantified into 72 bins, and we use another bin to collect pixels where edge strength is zero, thus we construct a histogram with 73 bins.

2.2 BoW features based on K-means

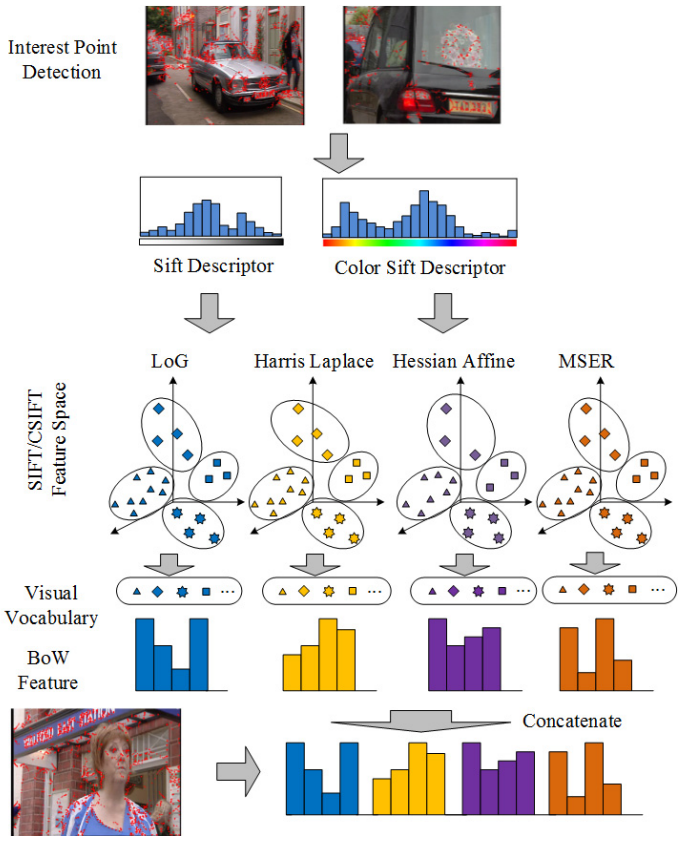

Figure 2: Combination of BoW features based on detectors and descriptors.

We explore the keypoint-based BoW (Bag-of-Word) features to represent each keyframe image. In our method, the extraction of keypoint-basd BoW features includes three steps:

(1) Detect keypoints using four detectors from the images, and use two descriptors to represent the regions of those keypoints.

(2) Use K-means algorithm to cluster the keypoints into 1000 clusters, and form a visual vocabulary with the cluster centroids.

(3) Adopt soft-weighting[5] method to assign keypoints to multiple nearest visual words (centroids), where the word weights are determined by keypoint-to-word similarity. The normalized histogram of visual words forms a BoW feature vector.

In step (1), we adopt four complementary detectors to detect the keypoints from images: Laplace of Gaussian (LoG)[1], Harris Laplace[2], Hessian Affine[3], and MSER [4]. For each detector, we use following two descriptors to generate two Bow features: 128-dimension SIFT descriptor[1] and 192-dimension ColorSIFT descriptor[7]. As shown in Figure 2, for each combination of detector and descriptor, a 1000-dimension feature vector is generated separately. Different BoW

features and basic features are concatenated to form the final feature in different runs, as described in Table 1.

2.3 BoW features based on AKM

The extraction of AKM based BoW features also includes three steps: First, we extract SIFT and color SIFT using the same methods in 2.2. Second, we use approximate K-means algorithm to generate a one-million codebook[9]. Third, we quantize each frame into a BoW vector, and further utilize average pooling to aggregate BoW vectors of all frames in each shot. Thus each shot can be represented by a one-million dimension BoW vector. Therefore we get 4×2=8 BoW features for each video shot and each query topic.

3 Feature Matching

In feature matching, we adopted two kinds of methods for different features. For the AKM based BoW features, we use L1 distance to calculate the similarity between the query and each shot. We can get 8 ranking scores using 8 BoW features, the final ranking list is generated by fusing the 8 ranking scores. For the basic visual features and K-means based BoW features, multi-bag SVM is adopted since it can make full use of all query examples. Moreover, we conduct keypoint matching algorithm on the top ranked results. It is very effective yet efficient since only top ranked results are considered.

The query examples are considered as positive samples. Due to the fact that only a few shots are relevant with the topics in the test data set, we adopt the random sampling of test data as negative examples. A problem of learning-based method is that there are too few positive samples and too many negative samples. In our approach, we use MBSVM algorithm to handle this imbalanced problem, and the algorithm details are presented in Figure 3 and the diagram is shown in Figure 4.

(1) Over-sample the positive samples: Duplicate the positive sample set P for $(PCopy - 1)$ times and get a new set of positive samples P' with $PCopy \times PN$ samples, where *PN* is the number of positive samples in P before over-sampling.

(2) Under-sample the negative samples: Randomly select $NPR \times PCopy \times PN$ negative samples, and combine them with the over-sampled positive sample set P' to form a bag. That is to say, in each bag, the number of negative samples is *NPR* times as the number of positive samples, where *NPR (negative-to-positive-ratio)* is a parameter to control the degree of data imbalance in each bag. A model is trained by *LibSVM* for each bag, where *RKF* kernel is used with default parameters.

(3) Repeat the above step (2) for *BagNum* times, where *BagNum* is a parameter specifying the number of bags. Then for each shot in the test data set, the *BagNum* prediction scores given by different models are averaged to form the final result. Notice that the negative samples in each bag are selected without repetition, that is, the negative samples are totally different in these bags. This ensures that we can make full use of the most of negative samples.

Figure 3: our algorithm for learning-based retrieval.

Totally, there are three important parameters in MBSVM algorithm: *PCopy*, *NPR* and *BagNum*. Experiments show that *PCopy=100*, *NPR=5* and *BagNum=5* can achieve good performance in both the accuracy and efficiency, while *PCopy* needs to be set according to the number of frames

extracted from each video clip in the query examples.

We use keypoint matching method based on SIFT descriptor to further improve the performance. Since keypoint matching is time consuming, we only conduct keypoint matching algorithm on the 1000 top ranked videos, which is effective yet efficient.

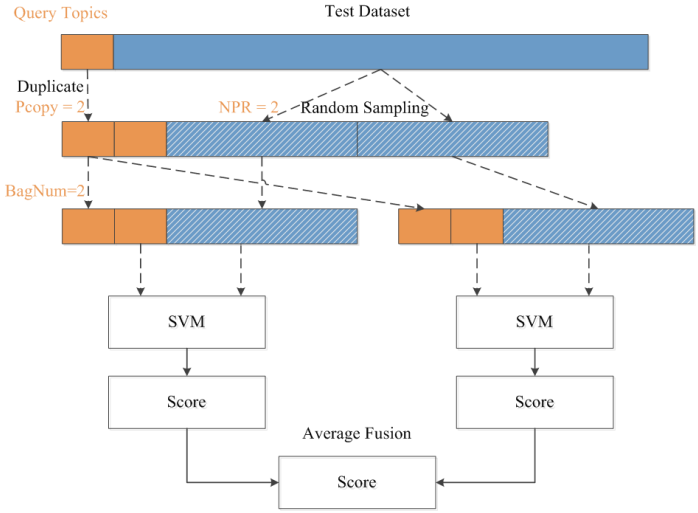

Figure 4: Diagram of MBSVM algorithm, where Pcopy=2, NPR=2 and BagNum=2.

4 Re-ranking

In re-ranking stage, NIST provides transcripts about the videos this year, and the topic 9105 (“this dog, Wellard”) explicitly points out the name of the instance to search. An instance appears in corresponding shot when its name appears in the transcript. We move such shot to the front of the ranking list.

5 Interactive Search

The detail of interactive search is described as follows: First, we retrieve the related 1000 video shots by using our methods of Run1 in Figure 1. Second, users will manually annotate the returned results. The negative samples will not appear in the final ranking list, and the positive samples will be regarded as extending queries. Third, we retrieve again using the extending positive samples to get extending scores. The final ranking list is the late fusion of extending scores and original scores. In our interactive system, we encourage users to select positive samples from different shots, for such samples can provide rich and diverse information. The artificial feedback significantly boosts the final accuracy, compared with fully automatic results.

6 Conclusion

By participating in the instance search task in TRECVID 2014, we have the following conclusions: (1) Effective feature is still vital, (2) High dimension BoW feature representation with simple distance-based similarity measure can outperform low dimension feature with

learning-based similarity measure, and (3) Keypoint matching is very helpful.

Acknowledgements

This work was supported by National Hi-Tech Research and Development Program (863 Program) of China under Grants 2014AA015102 and 2012AA012503, National Natural Science Foundation of China under Grant 61371128, and Ph.D. Programs Foundation of Ministry of Education of China under Grant 20120001110097. We also thank BBC for providing the EastEnders dataset: Programme material copyrighted by BBC.

References

[1] D. G. Lowe, “Distinctive image features from scale-invariant keypoints”, *International Journal of Computer Vision(IJCV)*, vol. 60, no.2, pp.91-110, 2004.

[2] C.G.M. Snoek, K.E.A. van de Sande, O. de Rooij, et al., “The MediaMill TRECVID 2008 Semantic Video Search Engine”, *Proceedings of the 6th TRECVID Workshop*, 2008.

[3] K. Mikolajczyk, and C. Schmid, “Scale and affine invariant interest point detectors”, *International Journal of Computer Vision(IJCV)*, vol. 60, no. 1, pp. 63-86, 2004.

[4] J. Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions”, *British Machine Vision Conference(BMVC)*, pp.384-393, 2002.

[5] Y.-G. Jiang, C.-W. Ngo, and J. Yang, “Towards optimal bag-of-features for object categorization and semantic video retrieval”, *ACM international conference on Image and video retrieval(CIVR),* pp.494-501, 2007.

[6] Koen E. A. van de Sande, Theo Gevers, and Cees G. M. Snoek, “Evaluating Color Descriptors for Object and Scene Recognition”, *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, vol. 32, no.9, pp. 1582-1596, 2010.

[7] K. Mikolajczyk and C. Schmid, “A performance evaluation of local descriptors”, *IEEE Transactions on Pattern Analysis and Machine Intelligence(TPAMI)*, vol. 27, no.10, pp. 1615-1630, 2004.

[8] P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quenot, “TRECVID 2014 -- An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics,” in Proceedings of TRECVID 2014. NIST, USA, 2014.

[9] Philbin, James, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman, "Object retrieval with large vocabularies and fast spatial matching", *IEEE International Conference on Computer Vision and Pattern Recognition(CVPR),* pp.1-8, 2007.