PKU-ICST at TRECVID 2013: Instance Search Task

Yuxin Peng, Xiaohua Zhai, Jian Zhang,

Lei Huang, Nianzu Li, Panpan Tang, Xin Huang and Yunzhen Zhao

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 2013: 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 adopt two basic visual features and four keypoint-based BoW features, and combine them to represent effectively the frame image. In feature matching, multi-bag SVM is adopted since it can make full use of few query examples. Moreover, we conduct keypoint matching algorithm on the top ranked results. It is effective yet efficient since only top ranked results are concerned. In re-ranking stage, we observe that the top ranked videos always contain a few noisy videos. To eliminate such noise, we proposed semi-supervised learning based and query expansion based re-ranking algorithm to refine the top ranked results. In query expansion, we automatically choose top 10 predicted results as positive samples to retrain the model.

1 Overview

In instance search task of TRECVID 2013[8], we participate in all two types: automatic search and interactive search. We submitted 4 runs for the instance search task of TRECVID 2013, including 3 runs for automatic search and 1 run for the interactive search. The evaluation results of our 4 runs are shown in Table 1.

Table 1: Results of our submitted 4 runs on Instance Search task of TRECVID 2013.

Type	ID	MAP	Brief description
Automatic	F_NO_PKU-ICST-MIPL_1	**0.212**	B+K+C+Q+M+T
	F_NO_PKU-ICST-MIPL_3	0.200	B+K+C+ R+M+T
	F_NO_PKU-ICST-MIPL_4	0.198	B+K+C+Q+R+M+T
Interactive	I_NO_PKU-ICST-MIPL_2	**0.245**	B+K+H

In automatic search, our team is ranked 3rd in all 20 teams. In interactive search, our run is ranked 2nd. Table 2 gives the explanation of brief description in Table 1. The framework of our system for instance search task of TRECVID 2013 is shown in Figure 1.

Table 2: Description of our methods.

Abbreviation	Description
B	**B**asic feature
K	**K**eypoint based feature
M	Keypoint **m**atching
T	**T**ext Matching
C	Topic **C**ategory
R	**R**e-ranking based on semi-supervised learning
Q	**Q**uery expansion
H	**H**uman feedback

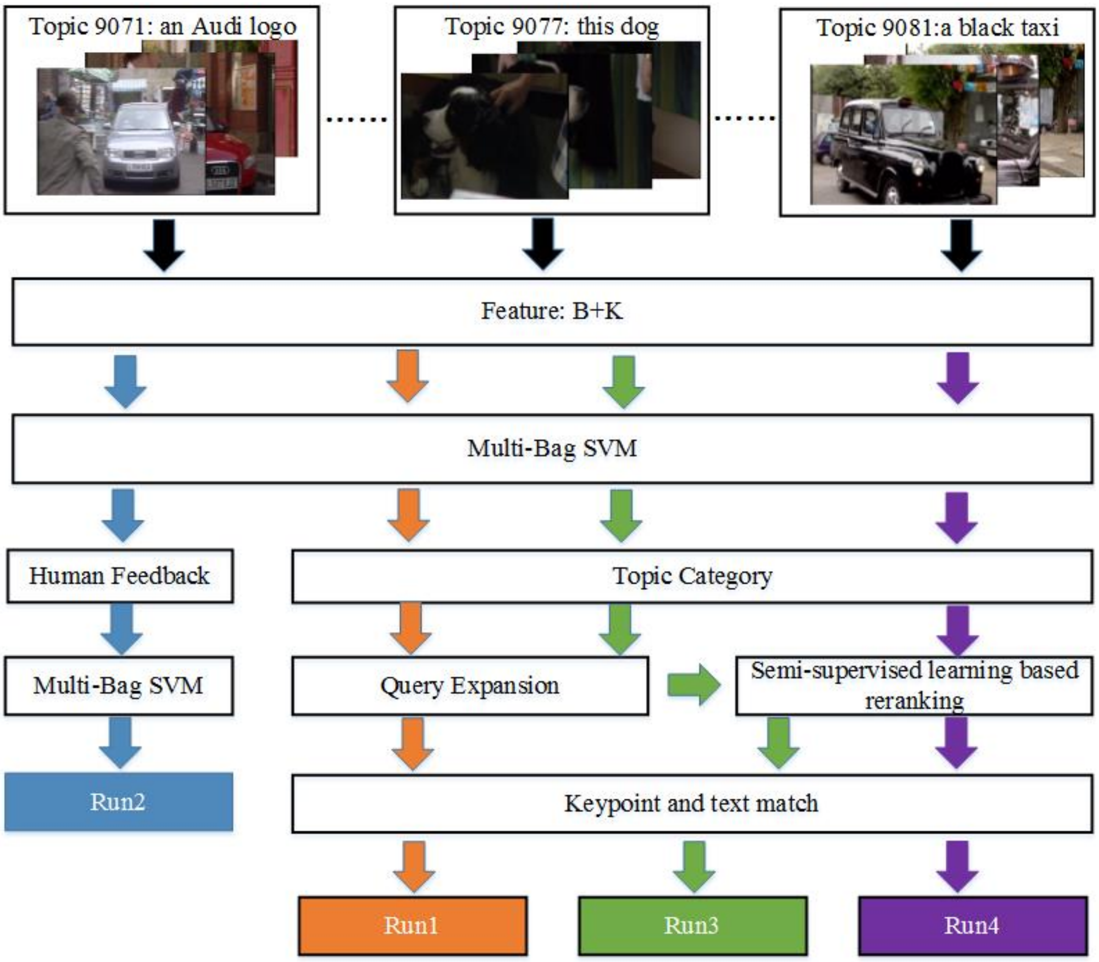

Figure 1: Framework of our instance search approach for the submitted four runs.

2 Feature Representation

We use two kinds of features for the instance search tasks, namely basic visual features and keypoint-based BoW features.

2.1 Basic visual features

We extract two basic visual features namely CMG(Color Moment Grid) and LBP(Local Binary Pattern) 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 edges 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 Keypoint-based BoW features

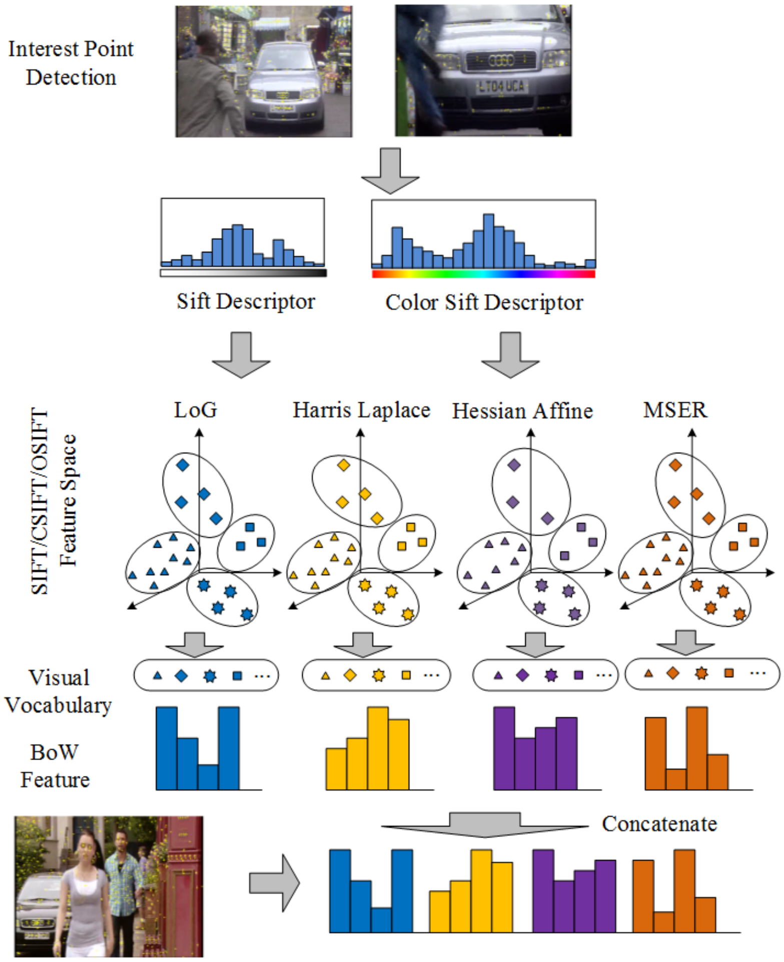

Figure 2: Combination of BoW features based on detectors and descriptors.

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 three descriptors to present 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.

3 Feature Matching

In feature matching, multi-bag SVM is adopted since it can make full use of few 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 concerned.

The query examples are considered as positive samples. Due to the fact that only 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, the algorithm details are presented in Figure 3 and 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.

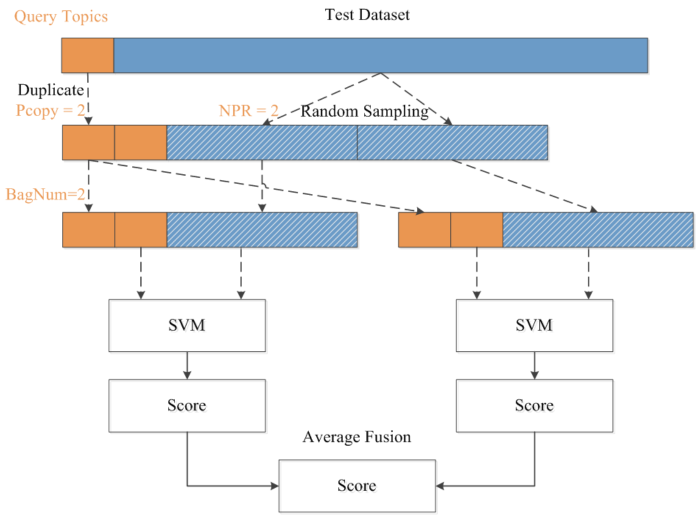

Figure 4: Diagram of MBSVM algorithm, where Pcopy=2, NPR=2 and BagNum=2.

Noticing that the object instances of some topics always appear in the same or similar scenes, we classify these correlated topics into same categories to enrich the training data. More specifically, we classify 6 topics into 3 categories as shown in Table 3.

Table 3: Three topic categories

Category	Topics
1	9076, 9078
2	9091. 9094
3	9090, 9095

We adopt the following steps to calculate $\mathrm{Prob}(kf, T)$, that is, the final probability score that keyframe *kf* belongs to topic *T*: (1) Get the original prediction score $\mathrm{Prob}_{org}(kf, T)$ and $\mathrm{Prob}_{cat}(kf, C)$ with MBSVM algorithm. $\mathrm{Prob}_{cat}(kf, C)$ stands for the probability that keyframe *kf* belongs to the category *C*. (2) We used the categories' information as a filter to get the final prediction score on topic *T*:

$$\mathrm{Prob}(kf, T) = \sqrt{\mathrm{Prob}_{org}(kf, T) \times \mathrm{Prob}_{cate}(kf, C)}$$

4 Re-ranking

In re-ranking stage, we observe that the top ranked videos always contain a few noisy videos. Figure 5 shows an example of query "*Stonehenge*". Most of the top ranked videos are correct and they look similar to each other. To eliminate such noise, we proposed a re-ranking algorithm to refine the top ranked results, which can make full use of the data distribution information. It consists of semi-supervised learning based re-ranking and query expansion based re-ranking. The detail of semi-supervised learning based re-ranking is described in Figure 6.

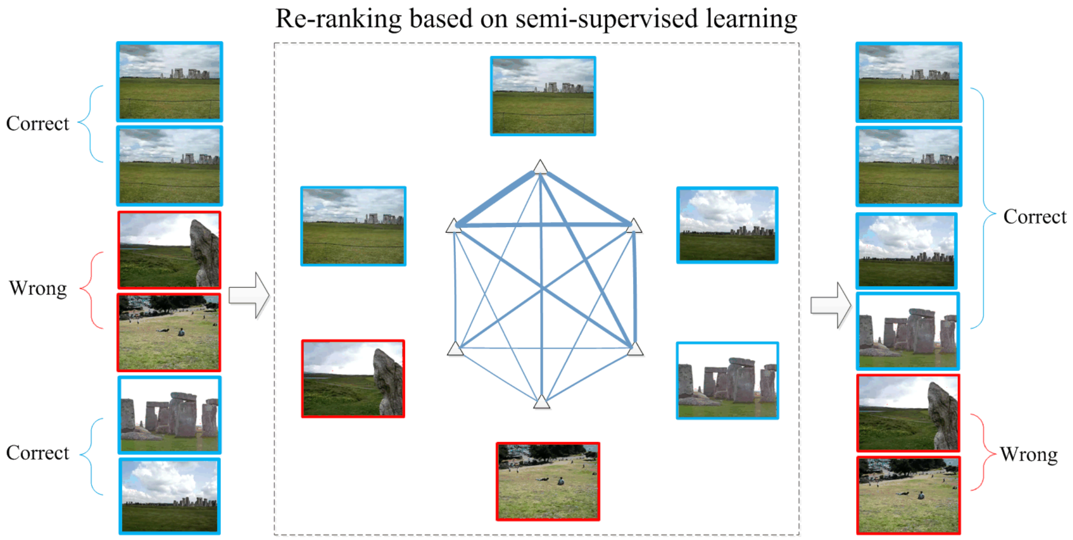

Figure 5: Results of query "*Stonehenge*". The top ranked videos always contain a few noisy videos. Most of the top ranked videos are correct and they look similar to each other. To eliminate such noise, we proposed a re-ranking algorithm based on semi-supervised learning to refine the top ranked results, which can make full use of the data distribution information.

(1) Given the data matrix of 1000 top ranked videos F and L, where F_i stands for the feature vector of a frame image and L_i stands for the video ID of vector F_i, $i \in \{1,2,\ldots,n\}$ where $n > 1000$ means there are n frames from 1000 videos.

(2) Initialize the affinity matrix W with all zeros, and update as following:

$$\mathrm{W_{i,j}} = \frac{F_i \cdot F_j}{|F_i| \cdot |F_j|}, \mathrm{i,j} \in \{1,2,\ldots,\mathrm{n}\}, \mathrm{i} \neq \mathrm{j}. \tag{1}$$

(3) Generate the k-NN graph:

$$\mathrm{W_{i,j}} = \begin{cases} W_{ij}, & F_i \in kNN(F_j); \\ 0, & otherwise. \end{cases} \tag{2}$$

$kNN(F_j)$ stands for the set of k-nearest neighbors of F_j.

(4) Construct the matrix: $S = D^{-1/2} W D^{-1/2}$, where D is a diagonal matrix with its (i,i)-element equal to the sum of the i-th row of W.

(5) Iterate $G_{t+1} = \alpha S G_t + (1-\alpha) Y$ until convergence, where G_t denotes the refined result in t-th round and we set $G_0 = Y$, α is a parameter in the range (0,1). Y is the initial score list of the frames of 1000 top ranked videos, we set the score of each frame the same as its original video.

Figure 6: re-ranking algorithm based on semi-supervised learning.

As for query expansion based re-ranking, we use the top-ranked results to further expand the queries and re-training the ranking model. Our method is described as follows: (1) For each topic, select the top 10 keyframes from different shots with the highest prediction score; (2) Using these keyframes as new positive samples and train MBSVM models for each topic; (3) Get the prediction scores with MBSVM; (4) Fuse the new prediction results with the original results.

Furthermore, this year NIST provides transcripts about the videos, and two topics (the topic 9088 for "*Tamwar*" and the topic 9096 for "*Aunt Sal*") explicitly point out the name of person to search. A person appears in corresponding shot when his name appears in the transcript. We move such shot in front of the ranking list.

5 Interactive Search

In the interactive search, we only adopt SIFT descriptor and two kinds of keypoint detectors: Harris Laplace detector and Hessian Affine detector. Each frame is represented as a 2000-dimension BoW feature vector, combining with 1413- dimension basic feature (CMG and EOH). The detail of interactive search is described as following: Firstly, we retrieve the related 1000 videos by Multi-bag SVM as introduced in Figure 3. Then, we manually annotate about 50 positive or negative results for each topic. According to our observation, we found following three key factors: (1) Positive and negative samples are both helpful, and positive samples provide much more information than negative ones. (2) Positive samples ranked lower are helpful because they provide much information complementary to query examples. (3) Negative samples ranked higher are helpful because they look similar to positive samples and are easily mistaken.

With those new positive and negative samples, we adopted Multi-bag SVM again to re-train models. In this round, we only predict the 1000 top ranked results from last round for efficiency. Finally, we got the interactive search results and return to users.

6 Conclusion

By participating in the instance search task in TRECVID 2013, we have the following conclusions: (1) effective feature is still vital, (2) learning-based similarity measure is a key factor, (3) re-ranking based on semi-supervised learning is helpful, and (4) query expansion can improve the performance of instance search task.

Acknowledgements

This work was supported by National Hi-Tech Research and Development Program (863 Program) of China under Grant 2012AA012503, Ph.D. Programs Foundation of Ministry of Education of China under Grant 20120001110097, and National Natural Science Foundation of China under Grants 61073084 and 61371128. We also thank BBC for providing the EastEnders dataset: Programme material © 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 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in Proceedings of TRECVID 2013. NIST, USA, 2013.