WHU-NERCMS at TRECVID2015:Instance Search Task

Lei Yao, Mang Ye, Dongjing Liu, Rui Shao, Tao Liu, Jun Liu, Zheng Wang, Chao Liang*

National Engineering Research Center for Multimedia Software, School of Computer, Wuhan University, Wuhan, 430072, China
{cliang@whu.edu.cn}

Abstract. This paper introduces our work at the automatic instance search task of TRECVID 2015. The purpose of this task is to search specific targets in a large-scale video database. The key problems this year we are concerned about includes: 1. How to improve traditional BoW models; 2. How to improve retrieval precision with cross-mode information as an auxiliary to traditional visual features; 3. How to optimize the initial retrieval results. Correspondingly, our work is divided into three parts: First part is object retrieval with visual features based on BoW model. In this part, Bow model is augmented by Query Adaptive Similarity Measure [1]. Second part is object retrieval with textual information. We adopt caption information and plot information of the series *EastEnders* in its official websites [2]. Third part includes several fusion and optimization strategies adopted to improve the initial results. The main improvement in the strategies is Query Expansion With Adjacent Shots, which aims to retrieve more precisely in the adjacent shots of initial top-k results.

1 Overview

In TRECVID 2015 [3], we participate in the automatic instance search task (INS). Table 1 gives the explanation of brief description in Table 2. The evaluation results of our 4 runs are shown in Table 2. Mean Average Precision (MAP) is the evaluation index [4]. The framework of our system for instance search task of TRECVID 2015 is shown in Figure 1.

2 Object Rertieval Based On BoW model

In this part, visual features based on BoW model is utilized to conduct an object search in the video database. The whole procedure of retrieval contains keyframe extraction, SIFT feature extraction, codebook training, BOW feature generation and query adaptive similarity measure. The detailed descriptions of the five parts are as follows.

* Corresponding author.

Table 1: **Description of our methods.**

Abbreviation	Method
B	**B**oW model
T	**T**extual Retrieval
O	Query Specific **O**ptimization
A	Query Expansion With **A**djacent Shots

Table 2: **Results of our submitted 4 runs on Instance Search task of TRECVID 2015.**

ID	MAP	Brief description
F_NO_NERCMS_1	0.052	B
F_NO_NERCMS_2	0.282	B + T
F_NO_NERCMS_3	0.318	B + T + O
F_NO_NERCMS_4	0.367	B + T + O + A

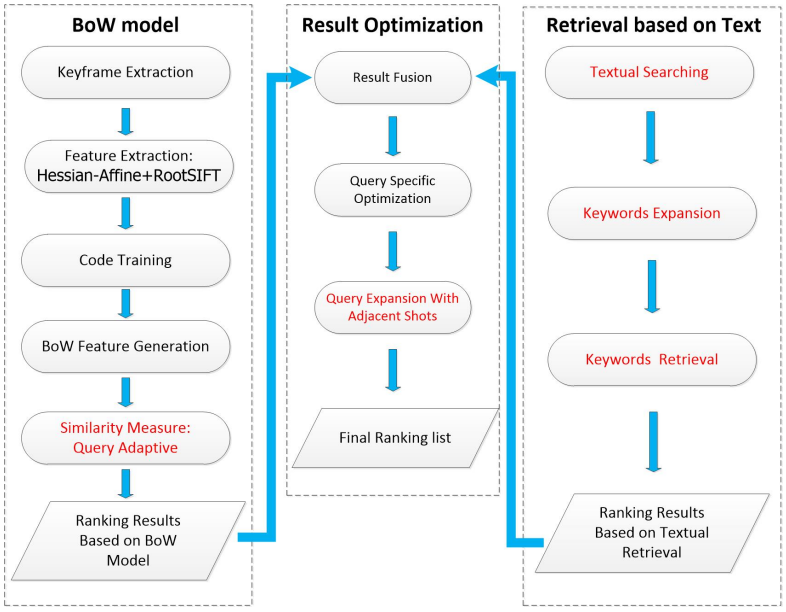

Fig. 1: **The framework of our team**

2.1 Keyframe Extraction

This section presents our keyframe extraction. Our purpose is to extract several keyframes in shots to represent them. In 2014, the middle frame of shot which includes frames less than 50 is extracted as the keyframe. When a shot includes 50 frames or more, we select 2 frames at the location of 1/3 and 2/3 of the keyframes. However, there are two main drawbacks in the method last year. Firstly, the amount of selected keyframes is not large enough so as to miss a lot of target frames. Secondly, there are many frames which are short of information in the initial selected keyframes. We call them vacant keyframes, which reduce the efficiency of the retrieval. Therefore, this year, first of all, we extract one frame in every 15 frames with the same intervals as a candidate keyframe. Then, information entropy is utilized to filter the vacant candidate keyframes. In John Zachary method [5], the information entropy function of a frame is

$$E(H) = -\sum_{i=0}^{n} h_i log_2(h_i) \tag{1}$$

Where n = 255 and grey level histogram h_i denotes the probability of grey level i appearing in the whole grey levels of a frame. When the information entropy of a candidate keyframe is less than 0.01 J, it will be judged as a vacant one and be filtered.

2.2 SIFT Feature Extraction

In feature extraction, we employed SIFT features to express a keyframe, since the local features based on scale-invariant key points have already been shown to be effective in object retrievals. Last year, we adopted SparseSIFT features [6]. Instead of taking features from every part of the image uniformly, SparseSIFT only returns features which are strong and distinct. This year, we replace SparseSIFT with hessian-affine SIFT features [7] because combination of Hessian-affine feature finder and SIFT features is most robust to viewpoint change. Considering the balance and calculation convenience, we can take no more than 1000 hessian-affine SIFT features per frame. Finally, we extract square roots of hessian-affine SIFT features to get the Root-SIFT features.

2.3 Codebook Training

As we know, codebook training is important to instance search. And many early research works show that the retrieval precision can benefit from larger size of visual codebook. We adopted Approximate K-Means (AKM) [8] algorithm to train the codebook. AKM uses randomized KD trees to perform approximate nearest neighbor search, which makes it possible to train large codebook in reasonable time.

Firstly, considering the hardware configuration and the requirement of time complexity, a subset consisting of 50 million features is selected randomly from

about 5 billion Root-SIFT features as training data. Then, it takes 7 circles to accomplish clustering these data, and every circle costs about 6 hours. Finally, we get a 1 million dimensional codebook.

2.4 BoW Feature Generation

With the trained codebook, we can quantize each 128 dimensional hessian-affine SIFT features of keyframes into one of the codes ranging from 1 to 1000000. There are two quantization strategies: hard assignment method and soft assignment method [9]. The hard assignment method requires each SIFT points quantized into only one word, while the soft assignment method allows each SIFT points to be quantized into several codes in codebook. For keyframes, there are a large number of SIFT points existing in non-region-of-interest. So soft assignment will increase the interference caused by these SIFT points in the retrieval. As a result, we adopt hard assignment method to quantize their hessian-affine SIFT features. And the experiments also prove that we can acquire better results with hard assignment method in stead of soft.

For query images, the BoW feature generation process is shown in Fig 4. Firstly, we select pictures which can reflect the features of the topics from different angles, scales and illumination sufficiently from the video. Secondly, we delimit the region-of-interest (ROI) in the selected query images manually. This method is simple and fast but can highlight the key information and filter the interference information effectively. The experiment proves that this method is much better than the stare model used last year. Thirdly, different from keyframes, after the second steps above, most of SIFT points of query images are existing in the ROI. So we make use of soft assignment method to quantize hessian-affine SIFT features of query images. In our method, each SIFT point is quantized into 3 different words in codebook. Finally, max pooling technique [10]is utilized to fuse features of query images from different sizes and angels selected in first step, with which we can get a more complete features vector.

2.5 Query Adaptive Similarity Measure

After getting the BoW features of both query images and keyframes, we need to find a effective metric to measure the similarity between them. Last year, we firstly convert the features vectors to binaryzation vectors. Then, we calculate the inner product to measure the similarity between query and gallery images. This method lose a great deal of features information in images. General image retrieval methods often extract SIFT features of the whole image. But Instance Search Task just require us to retrieve the specific targets from the ROI of query images. So for the keyframes, only the SIFT points clustered in the same code with the SIFT points extracted from query images can regarded as useful features for the similarity measure. For this reason, we adopt Query Adaptive Similarity Measure [1] this year. For every query image, the BoW features vector is $Q = (q_1, q_2, ..., q_N) \in R^N$. The counterpart of gallery image is $G = (g_1, g_2, ..., g_N) \in$

Fig. 2: **BoW Feature Generation**. Vectors f_k, k$\in$ (1, n), denotes BoW features of expansive query images respectively, n is the amount of the expansive query images, vector F denotes the final BoW features vector representing a topic.

R^N, the N is 1 million which denotes the amount of the codes in codebook. In this method, firstly, we get a query adaptive feature vector of gallery image, which is denoted as $G^{'} = (g_1^{'}, g_2^{'}, ..., g_N^{'}) \in R^N$. Its component can be described as

$$g_i^{'} = \begin{cases} g_i & q_i \neq 0 \\ 0 & q_i = 0 \end{cases}$$

where integer variable i $\in$ (1, N). Finally, dissimilarity between query image Q and gallery image G can be denoted as

$$S(Q, G) = \| \frac{Q}{\|Q\|} - \frac{G^{'}}{\|G^{'}\|} \| \tag{2}$$

where the symbol $\|*\|$ denotes the model of variable * .

3 Object Retrieval Based On Textual Information

This year, we take advantage of caption information for the first time. Firstly, we select the keywords for every topic. Then, we use textual retrieval to obtain target frames. It is clear that textual retrieval and BoW model retrieval can supplement with each other.

3.1 Keywords Mining

For the first part, we search keywords of every topic. This part consists of two steps, which are selecting initial keywords and expanding keywords. The framework of our method is shown in Figure 3. To begin with, after watching the

Video 0 and browsing the official website of the series, we select the initial keywords representing topics. Then, we make use of web crawler to obtain all the information describing plots of *EastEnders* in the official website. More specifically, Pan Gu Segment [11], a library that can segment Chinese and English words from sentence, is used to classify the crawled information as adjectives, adverbs, interjections, nouns and verbs. We intend to maintain nouns and verbs, and screen out all the adjectives, adverbs and interjections which lack useful information. After this, we input both initial keywords and maintained information into word2vec [12], from which we can acquire words relevant to initial keywords. This tool can transform words into vectors so as to replace textual semantic similarity with similarity calculation between vectors. Therefore, we obtain the expansive keywords of topics. Some examples of expanding keywords is shown in Figure 4.

Fig. 3: **The framework of our method**

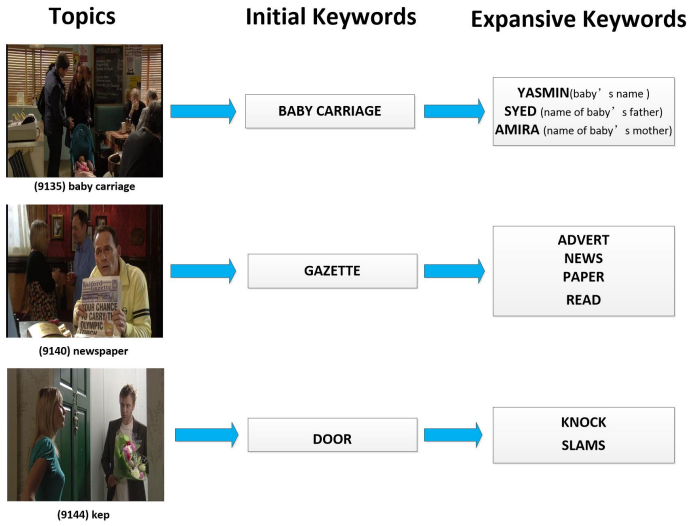

Fig. 4: **The three examples of expanding keywords**

3.2 Textual Retrieval

For the second part, we make use of simple string matching of keywords mentioned above in caption files. Keywords are retrieved one by one. Therefore, we can obtain several result collections. For example, about the topic *newspaper*, the amounts of the textual retrieval results appearing in the collection of the keyword *Gazette* are 125, and *advert* are 55, and *news* are 816, and *paper* are 259, and *read* are 647 respectively. After that, we suppose that the results appearing in all collections twice or more times have a higher reliability, which can be ranked forward in result list based on textual retrieval.

4 Result Optimization

After above steps, we fuse results based on BoW model and textual retrieval, then we get the initial results without any optimization. However, a lot of prior knowledge can be beneficial to optimize the initial results in practice.

4.1 Query Specific Optimization

Color Classifier Color filter is discussed in this subsection. The goal of color filter is to eliminate the images do not contain the target instance color obviously. The basic idea of our color filter method can be summarized as follows: firstly, extracting the h, s, v component of image in the HSV color model, and get the main color histogram of target Ht. It is much more precise than the method last year, in which only h component is extracted. Then, getting the area of image within the scope of color by doing reverse projection for each image through Ht; Thirdly, calculating the scope area $Area$. At last, if the $Area < threshold$, filter out the image.

In our implementation, considering the efficiency, the above process is decomposed into offline and online processes. Offline part statistics the area of each color value for every pictures under query; online part aims to determine the color range, and calculate the area for each color values of query images based on the offline part. When changing the target image or the parameters of threshold in the above final step, the advantage of these processes will be highlighted. The result of offline part process can be used repeatedly, we need only to do online part again. Furthermore, the offline part is time-consuming in entire process, while online part can be finished in a few seconds.

Face Filter In this subsection, we introduce the face filter we adopted for optimization. When the goal of topic is a person or always appears with persons, this method is adopted to filter the images which do not include persons. The face filter use the Viola-Jones face detect algorithm [13], which extract the integral images to calculate the Haar-like features efficiently. The Viola-Jones algorithm uses Adabosst leaning algorithm to select features and train the classifiers. And the cascade classifier is applied to promote efficiency.

In our implementation, firstly, we use the Viola-Jones face detect algorithm to detect all the keyframes. When face is detected, label 1 to the keyframe. Otherwise, label 0 to it. Then we get a vector valued by 0 and 1. Secondly, we select the topic of which target is person or always appears with persons manually. At last, we can optimize the results of person relevant topics using the vector achieved at first step.

4.2 Query Expansion With Adjacent Shots

This year, we take advantage of an optimization method called query expansion with adjacent shots. This method aims to find out the missing objects caused by keyframes extraction of shot. Basing on the prior knowledge—objects usually occur in several adjacent shots, firstly, we regard top N as right shots from initial results. The value of N depends on different topics. Secondly, we expand several adjacent shots according to characteristics of different objects. At last, our similarity measurement algorithm searches objects in these adjacent shots frame by frame. This algorithm conducts SIFT matching between query and gallery images directly, which is more precise but more time-consuming than similarity measurement algorithm based on BoW model. Then, the amounts of matched SIFT points are regarded as the evaluation index of similarity. The more matched SIFT points, the more similar two images are.

5 Results And Analysis

The four results of our INS system are shown in Fig 5. And the Fig 6 shows the results compare with other teams. The dot represents our best run score, the line represents median score and the best score is represented by box. From the Fig 6, we found that there are 8 topics having the best average precision.

Compared to our work at TRECVID 2014 [14], we have following conclusions:

- In comparison with BoW model last year, we perform a feature fusion on query images as a pretreatment. In the experiment, we make use of iteration to acquire object images from different scales, angles and definitions. After that, we extract their features and make a feature fusion. Apparently, the result of our experiment shows that the strategy of iterative feature fusion is considerably valid;
- In terms of similarity measure, with the experiment, we prove that method of query adaptive is much more effective than dot multiplication after binaryzation which we used last year;
- Apart from retrieval with images information, we also utilize target keywords and caption information to search objects. This strategy of textual retrieval is significantly simple and fast, especially for animals(topic 9139, 9141, 9142) and person(topic 9129, 9143), which can detects nonrigid objects, fixing the disadvantages of SIFT features;

– Color classifier is efficient for the targets with pure and bright color, such as blue baby carriage (topic 9135) and yellow VW beetle (topic 9136), because the color histogram of these targets are discriminative;
– Adjacent shots query expansion is a useful optimization method. Basing on the prior knowledge—targets usually occur in several adjacent shots, firstly, we manually screen out right shots from initial results. Secondly, we expand several adjacent shots according to different features of object images. At last, we search targets with the use of user-friendly interface. The fourth result we submit includes adjacent shots query expansion, whose MAP is 0.366, which is higher than the third result without this method by 0.05.

After analysing our results and comparing to other participants [15–17], we get some suggestions and experiences to guide future work:

– Adopt DPM training method. We can firstly learn it with a small-scale database, and then take a test. Since this method can filter the background information in a keyframe effectively but need a large amount of data for training of different topics;
– Extract deep learning features based on CNN learned from ImageNet to represent different instances as for global features;
– Adopt scene filter to optimize initial results, since many targets always appear in a specific scene. So we can optimize the results by filter the frames which do not include a kind of scene.

Acknowledgement. Our work use programme material copyrighted by B-BC. Thank for the great support to our work by prfessor Ruimin Hu and professor Jun Chen.

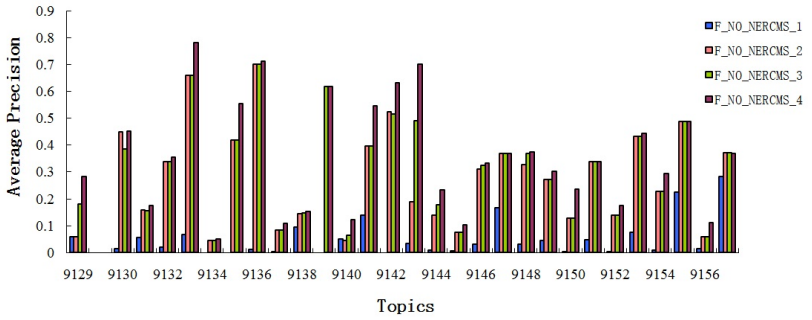

Fig. 5: **Our NERCMSs results**

References

[1] Cai-Zhi Zhu, Herve Jegou, Shinichi Satoh. Query-adaptive asymmetrical dissimilarities for visual object retrieval. In ICCV. (2013)

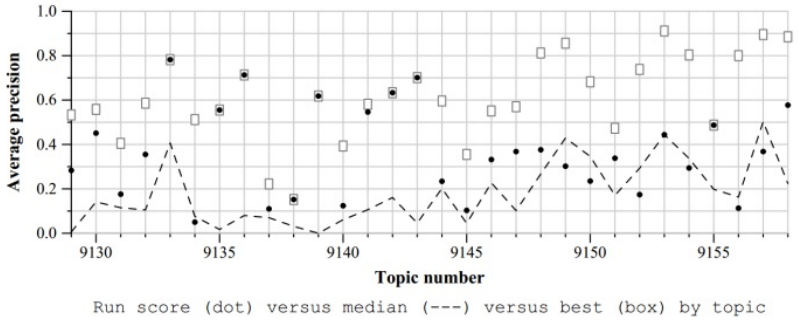

Fig. 6: **The results compare with other teams**

[2] http://www.bbc.co.uk/programmes/b006m86d
[3] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Wessel Kraaij, Alan F. Smeaton, Georges Quenot, Roeland Ordelman. TRECVID 2015 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. In: Proceedings of TRECVID 2015. (2015)
[4] Alan F. Smeaton, Paul Over, and Wessel Kraaij. Evaluation campaigns and trecvid. In: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval(ACM). 321-330 (2006).
[5] John Z M. An information theoretic approach to content based image retrieval [D]. Louisiana State University and Agricuhural and Mechanical College. (2000)
[6] Hammad Naeem, Maria Minhas, Jameel Ahmed. A Comparative Study about Object Classification Based On Global and Local Features [J]. IJCSI International Journal of Computer Science Issues. (2013)
[7] Pierre Moreels, Pietro Perona. Evaluation of Features Detectors and Descriptors based on 3D Objects. In: International Journal of Computer Vision. (2007)
[8] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In: Computer Vision and Pattern Recognition (CVPR). (2007)
[9] Arandjelovic R, Zisserman A. Three things everyone should know to improve object retrieval. In: Computer Vision and Pattern Recognition (CVPR). (2012)
[10] Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In: Neural Information Processing Systems (nips). (2012)
[11] http://pangusegment.codeplex.com/
[12] https://code.google.com/p/word2vec/
[13] Paul Viola, Michael J. Jones.: Robust Real-Time Face Detection[J]. In: International Journal of Computer Vision(IJCV). (2004)
[14] Mang Ye, Bingyue Huang, Lei Yao, Jian Qin, Jian Guan, Xiao Wang, Bo Luo, Zheng Wang, Dongjing Liu, Zhuosheng Zhang, Su Mao, Chao Liang. WHU-NERCMS at TRECVID2014: Instance Search Task. In: Participant Notebook Paper of TRECVID. (2014)
[15] Duy-Dinh Le, Vinh-Tiep Nguyen, Cai-Zhi Zhu, Duc M. Nguyen, Thanh Duc Ngo, Siriwat Kasamwattanarote, Poullot Sebastien, Minh-Triet Tran, Duc A. Duong, and Shinichi Satoh. NII at TRECVID 2014 Instance Search Task. In: Participant Notebook Paper of TRECVID. (2014)
[16] Zhicheng Zhao, Wenhui Jiang, Qi Chen, Jinlong Zhao, Yuhui Huang, Xiang Zhao, Lanbo Li, Yanyun Zhao, Fei Su, Anni Cai. BUPT-MCPRL at TRECVID 2014. In: Participant Notebook Paper of TRECVID. (2014)
[17] Cai-Zhi Zhu, Yinqiang Zheng, Ichiro Ide, Shinichi Satoh, Kazuya Takeda. Nagoya University at TRECVID 2014: the Instance Search Task. In: Participant Notebook Paper of TRECVID. (2014)