UNIVERSITY OF ENGINEERING & TECHNOLOGY, LAHORE THE UNIVERSITY OF SHEFFIELD AT TRECVID 2015: INSTANCE SEARCH

Maira Alvi†, Muhammad Usman Ghani Khan†, Yoshihiko Gotoh,Mehroz Sadiq†,Mubeen Aslam†*

† Department of Computer Science & Engineering , UET Lahore
*The University of Sheffield, UK

ABSTRACT

This paper describes our contribution for Instance Search(INS) tasks to TRECVID 2015. For instance search task we proposed four approaches, (i) combining hsvSIFT features with GMM matching rank list, (ii) SIFT features with Bhatacharya distance for similarity measurement, (iii) Combination of Colour SIFT descriptor with LUCENE,Terrier matching algorithm,(iv) feature vector is combination of HOG(Histogram of Oriented Gradients) features alone, while for matching we used euclidean distance.

Index Terms— video retrieval, instance search, video indexing

1. INTRODUCTION

TRECVID is a series of workshop focussed towards annotation, classification, summarization and retrieval of multimedia data [1]. The INS task is a pilot task introduced in TRECVID 2010 campaign. Yearly, different testing video and query images are released to the participants for the INS task. In TRECVID 2011, the testing data was produced form the rushes collection. They automatically decomposed each video in the dataset into short and equally length clips with different names from the original video file. There were a total number of 20,982 test video clips and 25 image test queries. Some image transformations were also applied to random test clips. The task includes recurring queries with people, location and objects in the rushes.

In TRECVID 2012, there were 30 topics and more than 70000 short clips as testing data collected from the Flicker. The main objectives from participant was to explore the task definition and the evaluation issues. This year number of topics remained same, where 26 topics are objects and 4 are related to humans [2]. Dataset consisted of 464 hours of the BBC soap opera EastEnders which was available in MPEG-4 format.

In TRECVID 2014 testing video data was produced from BBC East Ender dataset collection. There were 243 test video clips and 29 image test queries. This year approximately 244 video files and 29 search topic were to evaluate.

2. INSTANCE SEARCH TASK

For Instance Search task we submitted four runs. Following sections present detailed discussion of these runs.

2.1. Run 1: hsvSIFT Features with GMM Matching

2.1.1. Framework Overview

The overview of proposed framework is shown Figure2. The initial step is to segment the video into frames. Video to frame conversion was achieved by using ffmpeg library at one frame per second. From these frames shots and then key frames were extracted, we used only one key frame for further feature extraction and matching process. Features were extracted using hsvSift from *VLFeat toolbox* [3]. These features were matched by Gaussian Mixture Models (GMM), using means, covariances and priors. Then these results were normalized and rank list was generated.

2.1.2. Segment stage

The video this year seems like a movie or TV play with voice and coherent plot. For some cases, it is not easy to detect the boundary, especially for the scene with non abrupt change. In order to catch each boundary changing on the video content, here, we adopt the difference calculation between successive frames, if it is greater than a constant value(which was achieved after too many manual tests) it is taken as a shot. These shots are further compared and threshold is calculated using various features, if sucessive shots diference increase threshold value it is known as key frame. In the end the middle frame of these key frames is taken as master frame. Features of video are extracted from this single key frame, using hsvSIFT algorithim. We used this algorithim from *VLFeat toolbox* [3]. These features act as key points of an image thus reducing much size, which make it efficent for matching of thousands of images.

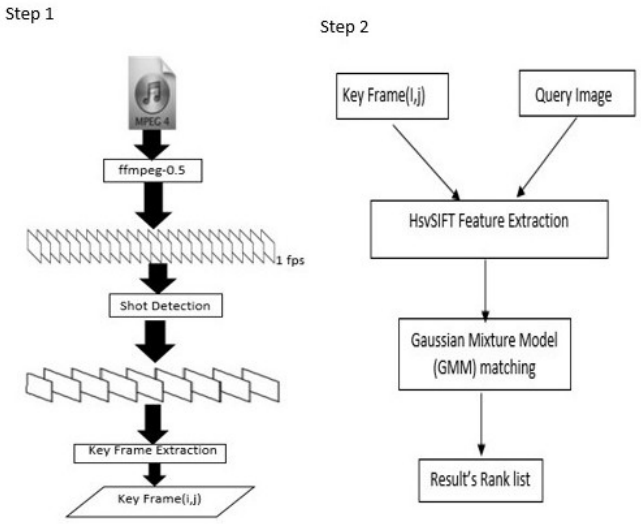

Fig. 1. Overview of Framework in Run 1

2.1.3. Matching stage

In this stage, we combined means, covariances and prior of Gaussian Mixture Model. Matching algorithim uses built in API from *VLFeat toolbox* [3]. Later their cross corealtion was calculated. rank list was generated using maximum of cross co-related results.

Evaluation results for this run are presented in figure 2

2.2. Run 2: SIFT features with Bhatacharya distance for similarity measurement

2.2.1. Offline Indexing

Similar to the first run, one frame per second are extracted from every video clips and used to compute PHOW descriptors. We also used the SIFT code available from the *VLFeat toolbox* [3]. The descriptors are computed from 4×4 cells and with 8 bins for histogram of oriented gradients (HOG).

2.2.2. Online Indexing

The framework of online searching is presented in part of Figure 3. Given the image set of topic, we extracted the SIFT features of query image. Then the feature vector consists of PHOW descriptors are computed for test images. For the search, each SIFT keypoint in the query topic is matched to its corresponding descriptors in the video clip database as proposed in [4].For similarity measurement, distances between each topic and every video clip is computed using the Bhattacharyya matching as following:

$$d_{Bhattacharyya}\left(I_1, I_2\right) = \sqrt{1 - \sum_i \frac{\sqrt{I_1\left(i\right) \cdot I_2\left(i\right)}}{\sum_i I_1\left(i\right) \cdot \sum_i I_2\left(i\right)}} \quad (1)$$

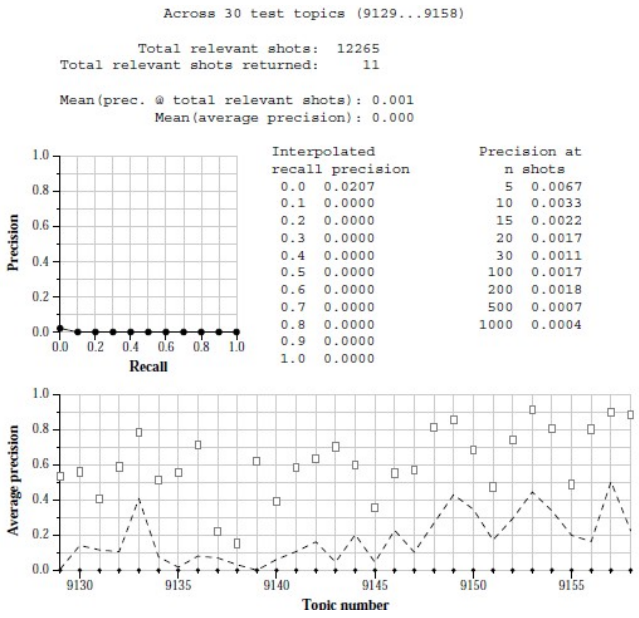

Fig. 2. Performance for Run 1 of instance search task

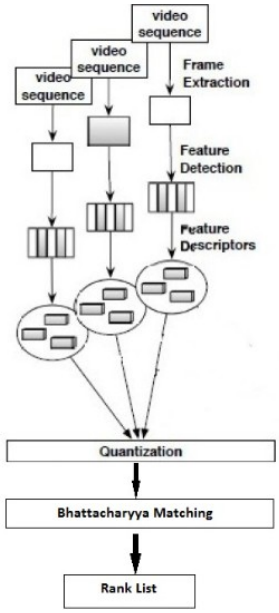

Fig. 3. Framework for run 2

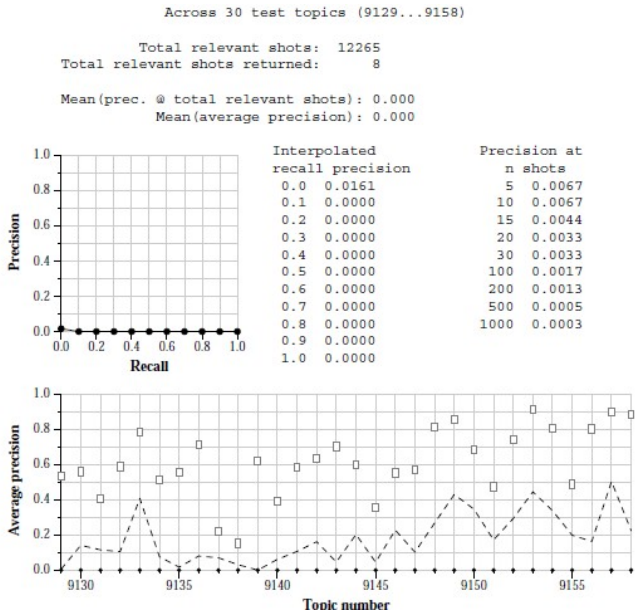

Fig. 4. Performance for Run 2 of instance search task

where I_1 and I_2 are the query topic and the video clip key frame image. The distances are sorted and the first 1000 lowest scores are returned as good matches.Evaluation results for this run are presented in figure 4.

2.3. Run 3: IR based Approach

An IR-based framework is proposed to efficiently retrieve candidate images from large source collections. The source collection is indexed off line. The testing image is split into smaller queries. The index is queried against each query from the testing image to retrieve a set of potential source video segments. The top N images are selected for each testing image and the results of multiple queries merged using a score-based fusion approach [5] to generate a ranked list of source videos. The top K images in the ranked list generated by CombSUM are marked as potential candidate images.

Figure 5 shows the proposed process for retrieving candidate images using an IR-based approach. The source collection is indexed with an IR system (an offline step). The candidate retrieval process can be divided into four main steps: (1) pre-processing, (2) query formulation, (3) retrieval and (4) result merging. These steps are described as follows:

1. **Pre-processing:** This is the step for feature generation. Similar to the first two runs, for each of the suspicious document, Colour SIFT features are calculated and histograms of those features are generated. These histograms are considered as sentences of any text document.

2. **Query Formulation:** Sentences from the suspicious document are used to make a query. The length of a query can vary from a single sentence to all the sentences appearing in a document, i.e. the entire image. A long query is likely to perform well in situations when large portions of image are similar. On the other hand, small portions of similar images are likely to be effectively detected by a short query. Therefore, the choice of query length is important to get good results.

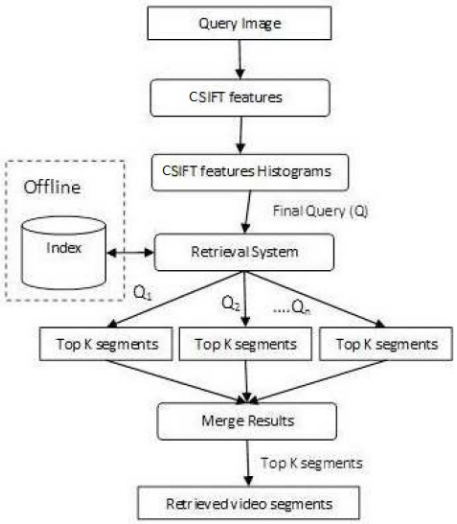

Fig. 5. IR Based Approach

3. **Retrieval:** Terms are weighted using the *tf.idf* weighting scheme. Each query is used to retrieve relevant source documents from the source collection.

4. **Result Merging:** The top N source documents from the result sets returned against multiple queries are merged to generate a final ranked list of source documents. In a list of source documents retrieved from a query, document(s) at the top of the list are likely to be the similar videos. In addition, portions of text from a single source document can be reused at different places in the same video segment. Therefore, selecting only the top N documents for each query in the result merging process is likely to lead to the original source document(s) appearing at the top of the final ranked list of the documents.

   A standard data fusion approach called CombSUM method [5] is used to generate the final ranked list of documents by combining the similarity scores of source documents retrieved against multiple queries. In the CombSUM method, the final similarity score, $S_{finalscore}$, is obtained by adding the similarity scores of source documents obtained from each query q:

$$S_{finalscore} = \sum_{q=1}^{N_q} S_q\,(d) \qquad (2)$$

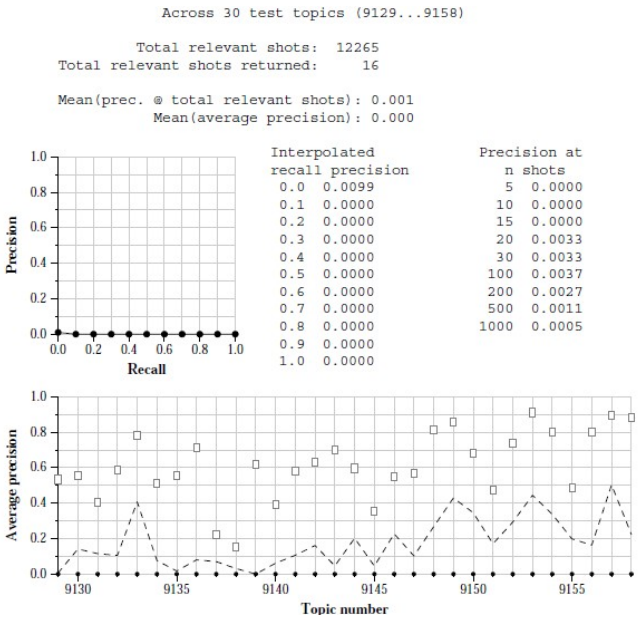

Fig. 6. Performance of the run 3 for instance search task

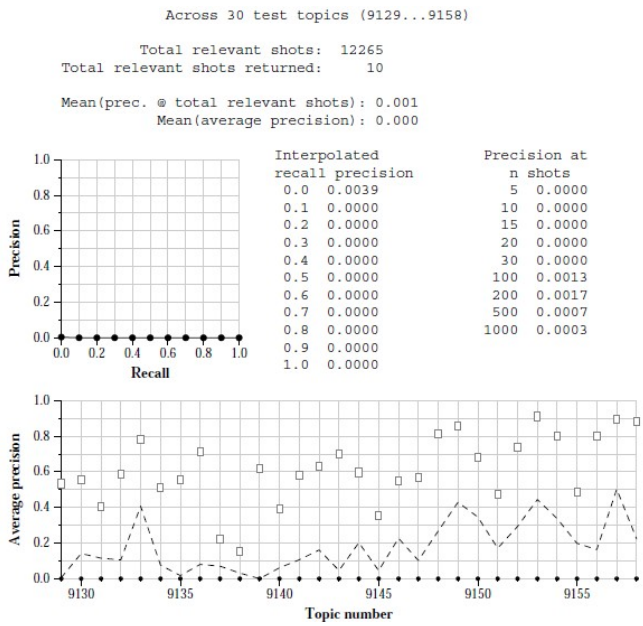

Fig. 7. Performance of the run 4 for instance search task

where N_q is the total number of queries to be combined and $S_q\,(d)$ is the similarity score of a source document d for a query q.

The top K documents in the ranked list generated by the CombSUM method are marked as potential candidate source documents.

2.3.1. Implementation

Two popular and freely available Information Retrieval systems are used to implement the proposed IR-based framework: (1) Terrier [6] and (2) Lucene [7]. In both Terrier and Lucene, terms are weighted using the *tf.idf* weighting scheme. In Terrier, documents against a query term are matched using the TAAT (Term-At-A-Time) approach. Using this approach, each query term is matched against all posting lists to compute the similarity score. In Lucene, the similarity score between query and document vectors is computed using the cosine similarity measure. The performance of this run is presented in Figure 6.

2.4. Run 4: HOG (Histogram of Oriented Gradient) Features Descriptor and Euclidean Distance Matching

Histograms of Oriented Gradient technique was used for finalizing this run. For extraction of these features, VLFeat library developed by Oxford University team was used [3].Distance between query image and test image features for matching purpose uses Euclidean distance as shown in equation.Finally, the highest scores are used as rank in the final result.The performance of this run is presented in Figure 7.

3. CONCLUSION

In this paper we presented our experiments performed in the TRECVID 2015 instance search tasks. This participation rewarded us an experience in our researches and in finding new ideas and directions in the domain of object-based video retrieval.

4. REFERENCES

[1] Alan F. Smeaton, Paul Over, and Wessel Kraaij, "Evaluation campaigns and trecvid," in *MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval*, New York, NY, USA, 2006, pp. 321–330, ACM Press.

[2] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Wessel Kraaij, Alan F. Smeaton, Georges Quenot, and Roeland Ordelman, "Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2015*. NIST, USA, 2015.

[3] Andrea Vedaldi and Brian Fulkerson, "Vlfeat: an open and portable library of computer vision algorithms," in *Proceedings of the international conference on Multimedia*, New York, NY, USA, 2010, pp. 1469–1472, ACM.

[4] David G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, pp. 91–110, 2004.

[5] E. Fox and J. Shaw, "Combination of multiple searches," *NIST SPECIAL PUBLICATION SP*, pp. 243–243, 1994.

[6] I. Ounis, G. Amati, Plachouras V., B. He, C. Macdonald, and Johnson, "Terrier Information Retrieval Platform," in *Proceedings of the 27th European Conference on Information Retrieval*. 2005, pp. 517–519, Springer.

[7] E. Hatcher, O. Gospodnetic, and M. McCandless, *Lucene in Action*, Manning Publications, 2004.