THE UNIVERSITY OF SHEFFIELD, HARBIN ENGINEERING UNIVERSITY AND UNIVERSITY OF ENGINEERING & TECHNOLOGY, LAHORE AT TRECVID 2013: INSTANCE SEARCH & SEMANTIC INDEXING

Muhammad Usman Ghani Khan[†], *Khawar Bashir*[†], *Abad Ali Shah*[†], *Lei Zhang*[‡],
Yoshihiko Gotoh[♣], Pervaiz Iqbal Khan[†] and Mehwish Amiruddin[†]

[†] Department of Computer Science & Engineering , UET Lahore
[‡] Harbin Engineering University, PRC,
[♣] The University of Sheffield, UK

ABSTRACT

This paper describes our contribution for Semantic Indexing(SI) and Instance Search(IS) tasks to TRECVID 2013. For instance search task we propose three approaches, (i) combing HOG descriptors and SIFT Features with window matching algorithm, (ii) SIFT features with Bhatacharya distance for similarity measurement, (iii) IR based approach using SIFT features alone. For semantic indexing task, we present four approaches (i) feature vector is combination of SIFT features alone, while for matching we used inverted index (ii) using histograms of SURF features as feature vectors and Bhatacharya distance for similarity detection (iii) affine invariant SIFT features as feature vectors and (iv) maximally stable extreme region feature vectors.

Index Terms— video retrieval, instance search task, 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.

2. INSTANCE SEARCH TASK

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

2.1. Run 1: HOG descriptors and SIFT Features

2.1.1. Framework Overview

The whole framework is shown in Figure 1. The first step is to segment the video into pieces, since in this year, the video is given as an original form, and some of them last close to 2 hours. Then for each segment, the key frame is extracted and the further searching is based on key frame only for our calculation ability. For each segment, only one key frame is extracted. During the searching stage, since we want to combine advantages of both global descriptor and local descriptor, we compute the HOG and SIFT from local view and compute the LBP form global aspect. Then we normalized the score for each feature and fusion them together.

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 LBP and GIST global feature to represent the content of each frame. Then the distance between adjacent frames are computed and based on these distances, we can find a maximum value. Treat the s% of the maximum distance as the threshold, the segment points can be found in the whole video. s is selected by experiments. This year, the segment label is given in a list. So in our segmentation stage, we relax s selection to make sure the non-segments will be combined together. So our segments number is much more than the

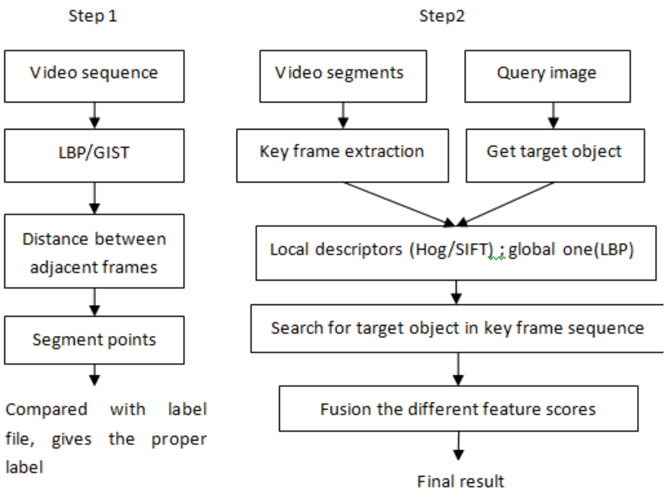

Step 1
Step2
Video sequence
LBP/GIST
Distance between adjacent frames
Segment points
Compared with label file, gives the proper label
Video segments
Query image
Key frame extraction
Get target object
Local descriptors (Hog/SIFT) ; global one(LBP)
Search for target object in key frame sequence
Fusion the different feature scores
Final result

Fig. 1. Framework of searching in first run

labeled one. Compared with the given label list, we re-label our segments, and maybe several continuous segments belong to the same label. Furthermore, we extract the middle frame as the key frame in our segment list, and each segment is with only one key frame for the calculation ability.

2.1.3. *Matching stage*

In this stage, we combined local descriptor and global one together, since for some query image, the target object is too small to get enough information by local descriptor.

LBP feature which is proved to be efficient to represent the content of frame in segment stage is adopted as global feature, while SIFT and HOG are adopted as local descriptors. For both key frame sequence and the query image after background subtraction, LBP, HOG and SIFT are extracted. Since the size of each frame in key frame sequence is as 768*576, being different from target object, a window matching algorithm is applied here. The algorithm is shown in Figure 2.

Evaluation results for this run are presented in figure 3

2.2. Run 2: Baseline run with SIFT only

2.2.1. *Offline Indexing*

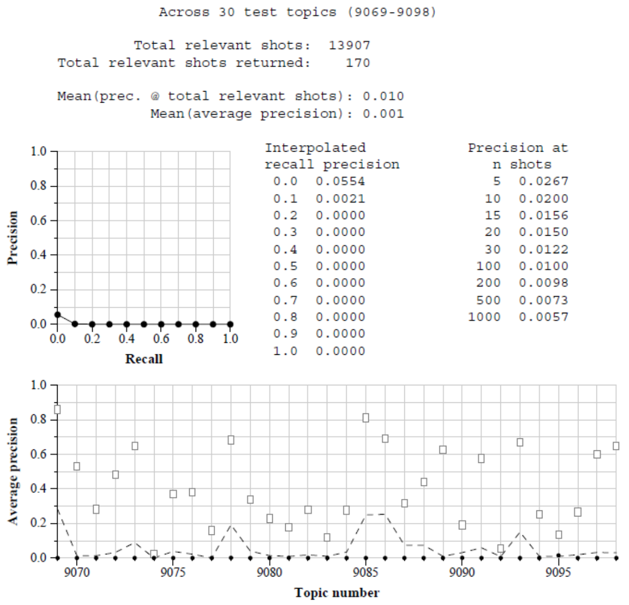
Similar to the first run, one frame per second are extracted from every video clips and used to compute PHOW descriptors. We also used the SIFT code available from the *VLFeat toolbox* [3]. The descriptors are computed from 4 × 4 cells and with 8 bins for histogram of oriented gradients (HOG).

2.2.2. *Online Indexing*

The framework of online searching is presented in part of Figure 4. Given the image set of topic, we extracted the Region of

```
Algorithm 1: window matching algorithm
    For each target object (after background subtraction)
        Compute the SIFT, HOG and LBP features
      For each key frame
        Starting from left-up pixel in key frame
        While the search region in key frame
            Get a rectangle region with the same size with target object
            Compute the SIFT, HOG and LBP features
            For local features
            Compute vector to vector distance
            Sum them together as region-to-target distance
            End
            For global features
            Compute the Euclidean distance
            End
            Move the region for 1/4 length and 1/4 width of target object
        End
        For each feature, keep the minimum distance when every region is
        coved in the key frame
        Normalization the distance into [0-1] range
        The multiplication of distance under each feature as the final score
    End
End
```

Fig. 2. Window matching algorithm



```
              Across 30 test topics (9069-9098)

            Total relevant shots:  13907
   Total relevant shots returned:    170

   Mean(prec. @ total relevant shots): 0.010
              Mean(average precision): 0.001

                  Interpolated              Precision at
                  recall precision             n shots
                   0.0  0.0554               5  0.0267
                   0.1  0.0021              10  0.0200
                   0.2  0.0000              15  0.0156
                   0.3  0.0000              20  0.0150
                   0.4  0.0000              30  0.0122
                   0.5  0.0000             100  0.0100
                   0.6  0.0000             200  0.0098
                   0.7  0.0000             500  0.0073
                   0.8  0.0000            1000  0.0057
                   0.9  0.0000
                   1.0  0.0000
```

Fig. 3. Performance for run 1 of instance serach task

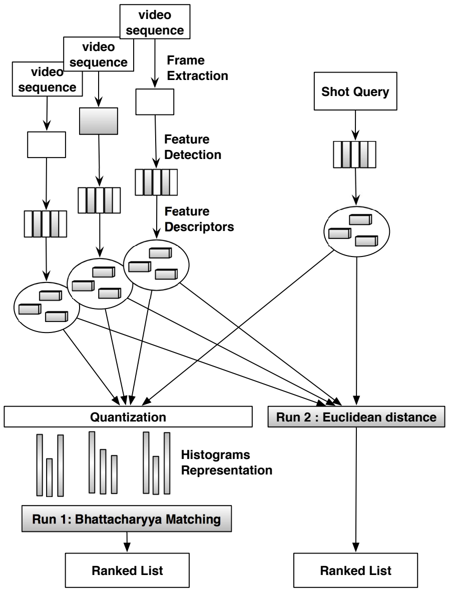
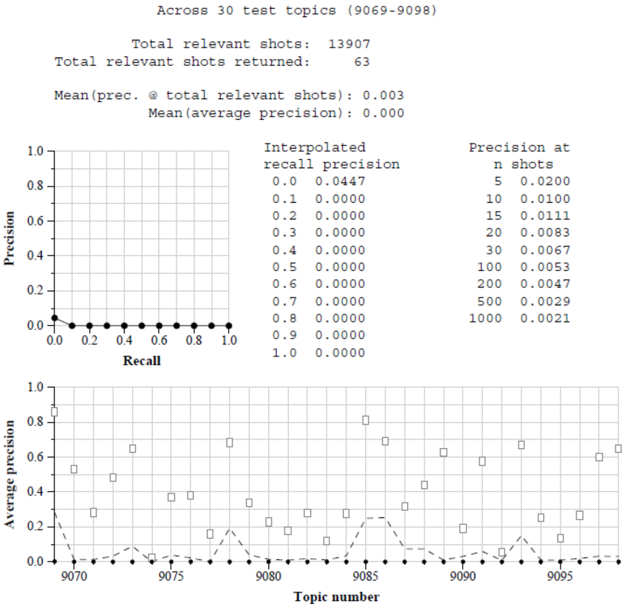


Fig. 4. Framework for online searching for run 2



```
              Across 30 test topics (9069-9098)

          Total relevant shots:  13907
 Total relevant shots returned:     63

 Mean(prec. @ total relevant shots): 0.003
            Mean(average precision): 0.000

                    Interpolated          Precision at
                    recall precision        n shots
                     0.0  0.0447             5  0.0200
                     0.1  0.0000            10  0.0100
                     0.2  0.0000            15  0.0111
                     0.3  0.0000            20  0.0083
                     0.4  0.0000            30  0.0067
                     0.5  0.0000           100  0.0053
                     0.6  0.0000           200  0.0047
                     0.7  0.0000           500  0.0029
                     0.8  0.0000          1000  0.0021
                     0.9  0.0000
                     1.0  0.0000
```

Fig. 5. Performance for run 2 of instance serach task

Interest (ROI) using the related mask. Then the feature vector consists of PHOW descriptors are computed. 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]. The computed scores based on the squared Euclidean distance between the query topic descriptor and the closest descriptor in the video database. Finally, the highest scores are used as rank in the final result. Evaluation results for this run are presented in figure 5

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 6 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, 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.

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

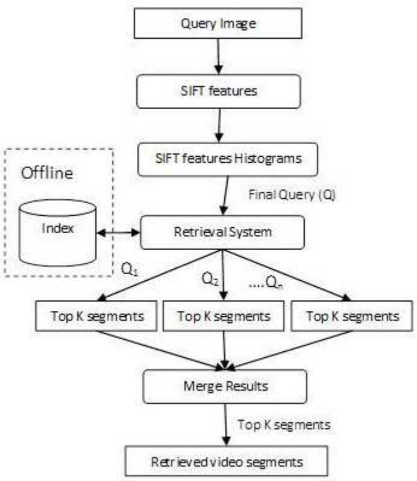


Fig. 6. Process of candidate document retrieval

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:

$$S_{finalscore} = \sum_{q=1}^{N_q} S_q\left(d\right) \tag{1}$$

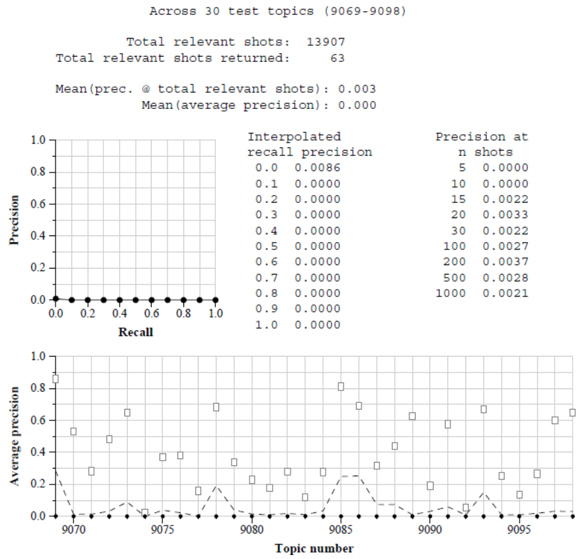
where N_q is the total number of queries to be combined and $S_q\left(d\right)$ is the similarity score of a source document d for a query q.

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 7.

3. SEMANTIC INDEXING TASK



```
                Across 30 test topics (9069-9098)

              Total relevant shots:  13907
     Total relevant shots returned:     63

     Mean(prec. @ total relevant shots): 0.003
                Mean(average precision): 0.000

                        Interpolated            Precision at
                        recall precision          n shots
                         0.0  0.0086                5  0.0000
                         0.1  0.0000               10  0.0000
                         0.2  0.0000               15  0.0022
                         0.3  0.0000               20  0.0033
                         0.4  0.0000               30  0.0022
                         0.5  0.0000              100  0.0027
                         0.6  0.0000              200  0.0037
                         0.7  0.0000              500  0.0028
                         0.8  0.0000             1000  0.0021
                         0.9  0.0000
                         1.0  0.0000
```

Fig. 7. Performance of the run R3 for instance search task

Following four runs were submitted for SI task. Previously, we have been participating in High Level Features extraction task [8]. Experience in that task helped us to devise the runs for this task.

3.1. Run 1: PHOW descriptors and bag-of-word approach

3.1.1. *Offline Indexing*

We extracted one frame per second from every video clips. Then we densely computed the PHOW descriptors on a regular grid across the image and vector quantised them into visual words. The codebook size is set to 500. We used the SIFT code available from the *VLFeat toolbox* [3]. The frequency of each visual word is then recorded in a histogram for each tile of a spatial tiling. The final feature vector for the image is a concatenation of these histograms.

3.1.2. *Online Indexing*

The framework of online searching is presented in part of Figure 4. Given the image set of topic, we extracted the Region of Interest (ROI) using the related mask. Then the feature vector consists of SIFT features computed on a regular grid across the image. Finally, the extracted SIFT features are projected to the vocabulary tree. One histogram is then generated as final representation for each topic. For similarity measurement, distances between each topic and every video clip is computed using the Bhattacharyya matching as following:

$$d_{Bhattacharyya}\left(H_1, H_2\right) = \sqrt{1 - \sum_i \frac{\sqrt{H_1\left(i\right) \cdot H_2\left(i\right)}}{\sum_i H_1\left(i\right) \cdot \sum_i H_2\left(i\right)}} \tag{2}$$

where H_1 and H_2 are the query topic and the video clip histograms. The distances are sorted and the first 1000 lowest scores are returned as good matches.

3.2. Run 2: SURF based run

3.2.1. 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 $4 \times\ 4$ cells and with 8 bins for histogram of oriented gradients (HOG).

3.2.2. Online Indexing

The framework of online searching is presented in part of Figure 4. Given the image set of topic, we extracted the Region of Interest (ROI) using the related mask. Then the feature vector consists of PHOW descriptors are computed. 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]. The computed scores based on the squared Euclidean distance between the query topic descriptor and the closest descriptor in the video database. Finally, the highest scores are used as rank in the final result.

3.3. Run 3: Affine SIFT based Run

3.3.1. Pre-processing

There are two steps for pre-processing. One is for testing video. In order to reduce the data size, for one video, only four frames from start, middle and end position are selected to represent the content of this video. Furthermore, this four frames are composed into one frame by zooming the size of each frame to proper level. The other step is for queries. The mask image is adopted to remove the background of each query. By shrinking the image size, four to five images of one query are also represented by one big image.

3.3.2. Feature extraction and distance matching

The framework of searching is presented in Figure 8. We adopted the Affine-SIFT code available from [9] and extracted ASIFT feature for every testing frame and query image. Then we matched testing frame and query image by the fully affine invariant image comparison method [10].

3.4. Run 4: MSER based run

Maximally Stable Extreme regions technique was used for finalizing this run. For extraction of these features, VLFeat library developed by Oxford University team was used [3].

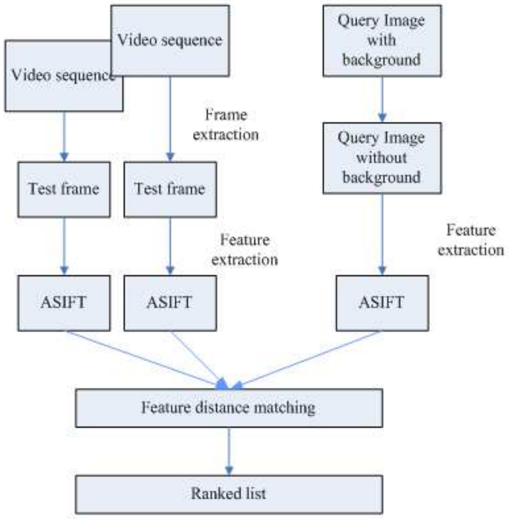


Fig. 8. Matching framework

4. CONCLUSION

In this paper we presented our experiments performed in the TRECVID 2013 instance search and semantic indexing 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.

5. 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, Greg Sanders, Wessel Kraaij, Alan F. Smeaton, and Georges Quenot, "Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2013*. NIST, USA, 2013.

[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.

[8] Alan F. Smeaton, Paul Over, and Wessel Kraaij, "High-Level Feature Detection from Video in TRECVid: a 5-Year Retrospective of Achievements," in *Multimedia Content Analysis, Theory and Applications*, Ajay Divakaran, Ed., pp. 151–174. Springer Verlag, Berlin, 2009.

[9] Guoshen Yu, Jean-Michel Morel, "ASIFT: An Algorithm for Fully Affine Invariant Comparison," *Image Processing On Line*, 2011, http://dx.doi.org/10.5201/ipol.2011.my-asift.

[10] Guoshen Yu and J.-M. Morel, "A fully affine invariant image comparison method," in *Acoustics, Speech and Signal Processing, ICASSP, IEEE International Conference on*, 2009, pp. 1597 –1600.