VIREO @ TRECVID 2014:
Instance Search and Semantic Indexing

Wei Zhang, Hao Zhang, Ting Yao, Yijie Lu, Jingjing Chen, Chong-Wah Ngo
Video Retrieval Group (VIREO), City University of Hong Kong
http://vireo.cs.cityu.edu.hk

Abstract

This paper summarizes the following two tasks participated by VIREO group: *instance search* and *semantic indexing*. We will present our approaches and analyze the results obtained in TRECVID 2014 benchmark evaluation [1].

Instance Search (INS):

We submitted seven runs derived from the following three systems (1) the baseline: our last year's best system; (2) the normalization: the method refining the normalization terms for both query and reference images; (3) the video agumented query: the original image query is augmented with the video example.

- F_A_VIREO_7: Baseline using the first image example only. Our baseline system is based on the Bag-of-Words (BoW) model [2], augmented with Hamming Embeding [3], spatial verification via Delaunay Triangulation [4] and context weighting via "Stare" model [5].
- F_B_VIREO_6: Baseline using the first two image examples only.
- F_C_VIREO_5: Baseline using the first three image examples only.
- F_D_VIREO_2: Baseline using all the four image examples.
- F_D_VIREO_3: Baseline + normalization method, using all the four image examples.
- F_E_VIREO_4: Baseline + video augmented query, using all the four image examples as well as the video examples where the query images are extracted.
- F_E_VIREO_1: Late fusion of the results from all our systems, including the baseline, normalization and video augmented query. This run also queries with all the four images and video examples.

Semantic Indexing (SIN):

This year, we experimented various features including the visual, motion and audio features in concept training. Specifically, state-of-the-art motion feature: Improved Trajectories [6] and aduio features: MFCC, LPC, LSF, OBSI [7] are involved in this year's benchmark evaluation. We submitted two runs to test these newly added features:

- 2B_M_D_VIREO.14_1: Late fusion of the detection scores using visual features.
- 2B_M_D_VIREO.14_2: Late fusion of the detection scores using visual, motion and audio features.

1 Instance Search

This year's dataset consists of 243 TV episodes from "BBC EastEnders", where we uniformly sampled a total number of 4.5 million frames (two frames per second). SIFT features [8] and BoW retrieval model [2] are adopted throughout our runs.

1.1 Methods

1.1.1 Baseline System

The baseline system is the same as our last year's best run, where Hamming Embedding [3], Delaunay Triangulation [4] and Stare model [5] are adopted. For more details, please refer to our last year's report [9]. The first four runs (VIREO_7, 6, 5 and 2) are all based on this baseline and only differ in the number of image examples used for querying.

1.1.2 Normalization Method

In BoW model, images are usually ranked based on the Cosine similarity: $\text{sim}(Q, R_i) = \frac{Q \cdot R_i}{|Q||R_i|}$, where Q and R_i are the BoW vectors for the query and i^{th} reference image respectively, and $|\cdot|$ indicates the vector norm. After matching with the inverted index, only a subset of local features are actually matched between Q and R_i. We further decompose the vectors as: $Q = Q^{in} + Q^{out}$, and $R_i = R_i^{in} + R_i^{out}$, where Q^{in} and R_i^{in} indicate the Bow vectors corresponding to the matched local features between Q and R_i, respectively. Accordingly, the similarity can be rewriten as:

$$\text{sim}(Q, R_i) = \frac{(Q^{in} + Q^{out}) \cdot (R_i^{in} + R_i^{out})}{|Q^{in} + Q^{out}||R_i^{in} + R_i^{out}|} = \frac{Q^{in} \cdot R_i^{in}}{|Q^{in} + Q^{out}||R_i^{in} + R_i^{out}|}, \quad (1)$$

where the cross terms in the numerator go to zero. Since the numerator is fixed given Q and R_i, larger normalization term in the denominator corresponds to heavier punishment to the similarity score. In the following, we will identify the issues in these normalization terms, and propose our solutions.

Query normalization. In Eq. 1, the query normalization term $\text{norm}(Q) = |Q^{in} + Q^{out}|$ stays as a constant for all reference images, and thus is ignored in most cases. Essentially, the reference images covering more visual words on the query should be favored. In the context of INS, where only a few visual words are extracted from the small instance, any single missed words on the querying object matters a lot. Note that this statement is not true for the reference image, since the missed visual words on R_i are mostly on the background. Motivated by this observation, we modify $\text{norm}(Q)$ as follows to normalize the query adaptively according to each reference image R_i:

$$\text{norm}(Q) = \sqrt{Q^{in} + \lambda Q^{out}}, \ \lambda > 1. \quad (2)$$

With this new query norm, we are more rigorous for misses on the query-side than those on the reference-side.

Reference normalization. For INS, instances covering only a small image area are quite common. Figure 1 shows a small querying instance Q and two reference frames R_1 and R_2. Obviously, the reference normalization term $\text{norm}(R_i) = |R_i^{in} + R_i^{out}|$ *over-norms* the score for R_1 by including the dense visual words on the background. With this normalization, relevant images with a small target are ranked much lower.

To address this "over-norm" problem, we modify $\text{norm}(R_i)$ to only include the visual words on a small subimage covering the matched words R_i^{in}. In particular, the reference norm, $\text{norm}(R_i)$, is online calculated only on the green box (Figure 1), which is estimated as the minimum rectangular region

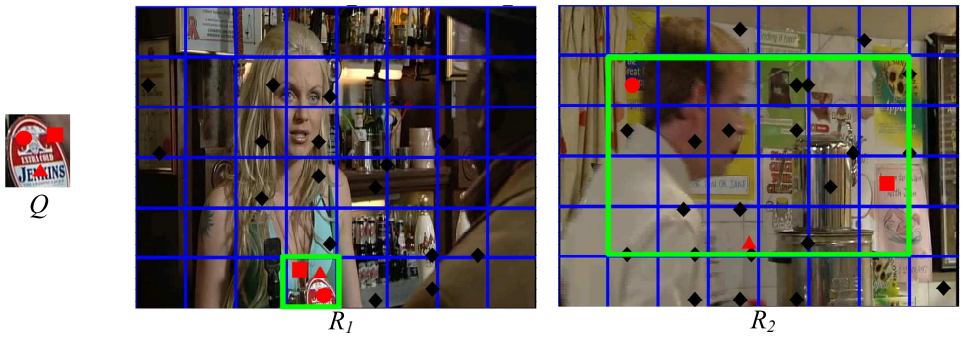

Figure 1: An illustration for the "over-norm" problem and our solution. Given a query Q, R_1 is a relevant reference image with the instance covering a small area, and R_2 corresponds to an irrelavant reference image with some hits for the visual words on Q. Red shapes indicate the corresponding visual words between the query and reference images, i.e., Q^{in} and R_i^{in}, while black diamonds represent the visual words that remain unmatched, i.e., R_i^{out}.

covering all the matched visual words on the reference image, i.e., R_i^{in}. Due to the large amount of possible rectangles on the reference image, we first pre-partition the reference image into small regions, as the blue grids in Figure 1, and then index the normalization term for each grid with the Integral Image technique. By doing so, the extra normalization from unrelated background features is excluded, significantly boosting the chance for small instance retrieval. On the other hand for irrelevant images (e.g., R_2), the matched visual words R_i^{in} are mostly spreaded all over the image, rather than being confined inside a small region. Therefore, the reduction of the reference norm for irrelevant images is quite limited, and do not hurt too much on the overall performance.

1.1.3 Video Augmented Query

This year, the video examples, from which the image queries are extracted, are made available. We also developed a system: video augmented query, to test whether this extra information can improve the performance. Our strategy is to augment the original query image with the video example, in a similar manner as *Query Expansion* [10]. However, the query video contains many irrelevant frames without the querying target, as well as large number of irrelevant features on the background. To augment the image query, these noises must be eliminated as much as possible. To do so, we first locate the timestamp of the query image inside the video, and then track the instance over time for more features to expand. Specifically, (1) the timestamp is estimated by matching the query image to the video frames. Though simple, this method works very well in practice; (2) to collect more information for the query instance, we adopt the object tracking approach STC (Spatio-Temporal Context) [11] to track the instance. Finally, all the features from the tracked instance, together with the original image example are pooled as a single BoW vector for retrieval.

1.2 INS Result Analysis

Overall performance. Figure 2 shows the performance of all runs for this year's INS task, in which our runs are highlighted in red. In general, the baseline runs (F_A_VIREO_7, F_B_VIREO_6, F_C_VIREO_5 and F_D_VIREO_2) are consistent with our last year's result, reaching a mAP of 0.20 by querying with all the four image examples. Our best run was given by F_D_VIREO_3, where the normalization and baseline results are fused. This shows that the results by our normalization and the baseline are complementary

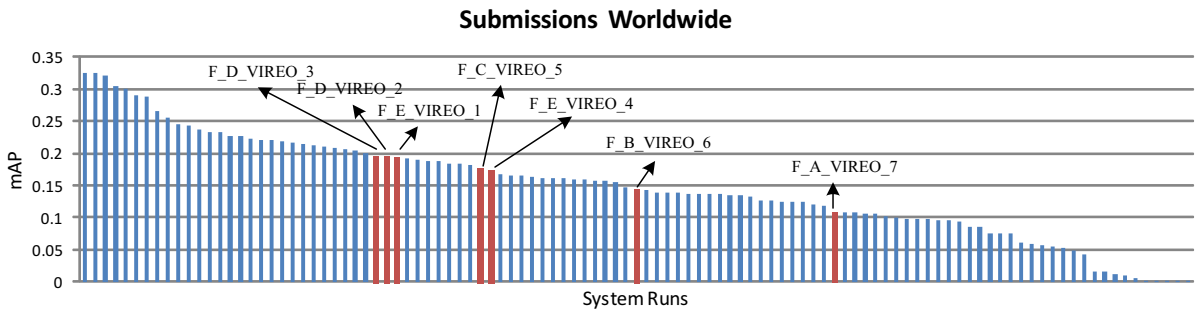

Figure 2: Mean Average Precision of all INS automatic runs submitted to TRECVID INS 2014.

Table 1: Performance of our internal runs TV14 INS task.

	VIREO_2	norm method	VIREO_3	VIREO_2 w/o DT	VIREO_2 w/o DT and Stare
mAP	0.1952	0.1933	0.1966	0.1577	0.1437

to each other. However, augmenting the query with additional video example (F_E_VIREO_4) did not improve the performance. This is due to the low quality issue observed in the provided video example.

Baseline. As expected, the baseline method, which applies Delaunay Triangulation (DT) for topological spatial verification and "Stare" for context weighting, performs consistently over years. Table 1 shows our internal runs without Delaunay Triangulation and Stare. As observed, removing Delaunay Triangulation from our baseline decreases the performance drastically from 0.1952 to 0.1577, and further removing Stare gives even worse result.

Normalization method. As indicated in Table 1, our normalization method alone did not give better performance compared to the baseline. However, fusing the results of normalization and baseline actually improved the performance (F_E_VIREO_3). We attribute this to the novel results returned by our normalization method, but missed by the baseline. Figure 3 shows several example frames that are ranked within the top-100 results by our normalization method, but missed by the baseline. Note that most of these instances are with small sizes and cluttered background, which indicates the importance of proper normalization for small instances retrieval.

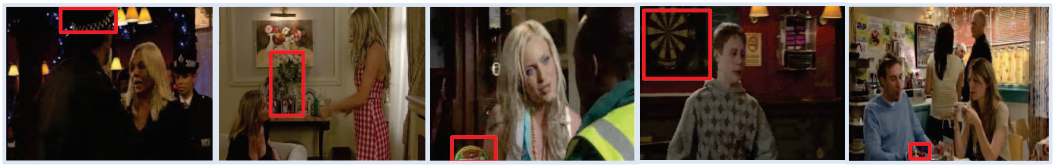

Figure 3: Example frames that are ranked within the top-100 list of our normalization method, but not the baseline. The target instances are marked with red boxes.

Video augmented query. By referring to Figure 4, augmenting the image query with the extra video example only improves a few topics, and hurts the overall performance. This is mainly due to the low visual quality of the video examples compared to the query images. Figure 5 shows the visual quality comparison between the image and video examples on two topics. For topics with high quality video examples, such as "washing machine" (left), our video augmented query improves the performance by a large margin. However, for topics with low quality videos, such as "Mustang grill logo" (right), the performance is much worse. With low quality video example, our method suffers from the problem known as *model drifting*, where the query drifts away from the actual searching target due to the noises.

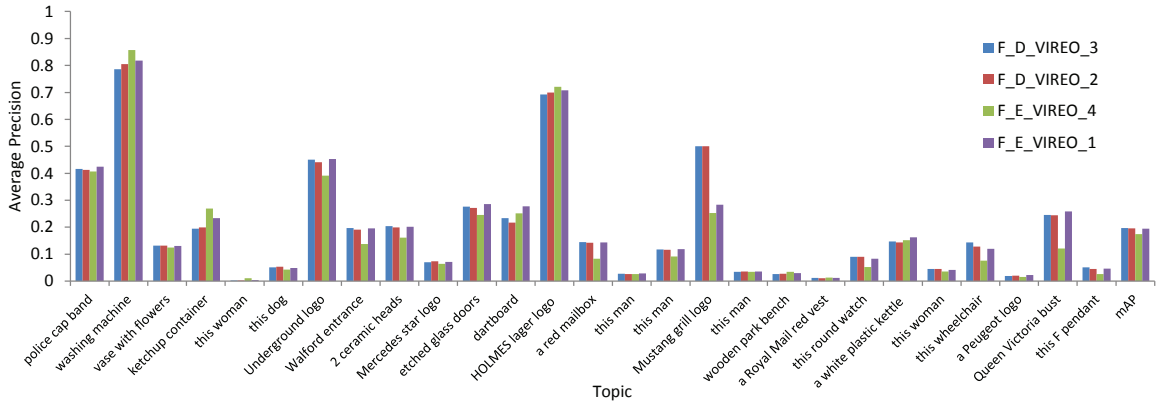

Figure 4: Detailed performance for our runs with types D and E.

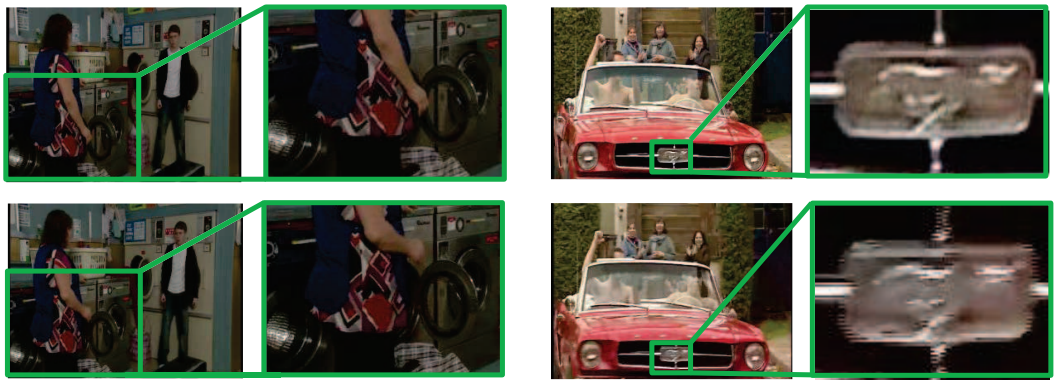

Figure 5: Visual quality comparison between the image and video examples on two topics: washing machine (left), Mustang grill logo (right). Top row: image query examples. Bottom row: the corresponding frame from the video query example.

In general, our experiments show that using the extra video information gives inferior performance, which seems to be a paradox. However, this phenomenon appears to be a common problem for every team. By investigating the results from other teams, systems with type D are better than E for almost every individual group, which confirms our observation on the low visual quality issue.

2 Semantic Indexing

For this year's SIN task, we evaluated various features extracted from different views of multimedia information, e.g., motion and audio. Besides the visual features extracted from static video frames, state-of-the-art motion feature: Improved Trajectories [6] and audio features: MFCC, LPC, LSF, OBSI [7] extracted from video segments are also evaluated in this year's submission.

2.1 Features

By nature, features extracted from different multimedia modalities reveal different views for a video, and are potentially complementary to each other. Thus, combining multimodal features is likely to improve the performance eventually. In this year's submission, besides the commonly adopted visual features, we

include motion and audio features extracted on video segments for concept training.

Visual feature. Besides the visual features adopted last year [9], we also include a few more for further improvement:

- PSS: Pyramid Self-Similarity encoded as Bag-of-Words (BoW) feature.
- PHOW: pyramid histogram of visual words feature, encoded as Fisher Vector.
- PHOG: Pyramid Histogram of Oriented Gradients as a global feature.
- GIST: Glabal feature depicting the context of a scene.
- DCNN: Deep Convolutional Neural Networks feature [12] from multiple layers. The DCNN architecture can be denoted as "Image-C48-P-N-C128-P-N-C192-C192-C128-P-F4096-F4096-F1000", which contains five convolutional layers (C), and three fully-connected layers (F). Max-pooling layers (P, following the 1st, 2nd and 5th layers) and local contrast normalization layers (N, following the 1st and 2nd P layers) are also included. We used the network learned in [12] directly due to limited time and resource. The final representation for each frame is the concatenation of the neuronal responses from layer 5 (C128), layer 6 (F4096), layer 7 (F4096), and layer 8 (F1000).

Motion feature. Improved Trajectory [6] encoded with Fisher Vector is extracted as motion feature. For each trajectory tracked in a video segment, trajectory feature, Histogram of Oriented Gradients (HOG), Histogram of Flow (HOF), and Motion Boundary Histogram (MBH) are separately extracted and then concatenated. Then, PCA is applied on this vector before Fisher Vector encoding.

Audio feature. Line Spectral Frequency (LSF), Octave Band Signal Intensity (OBSI), Linear Prediction Coefficients (LPC), Mel-Frequency Cepstral Coefficients (MFCC) are considered as the audio features. These features are first quantized into BoW vectors, and then encoded with Fisher Vector.

2.2 Concept Training

In our runs, the concept detector is first trained with the aformentioned features separately, and then the responses from different features are late fused as the final score for video ranking.

2.3 SIN Results and Analysis

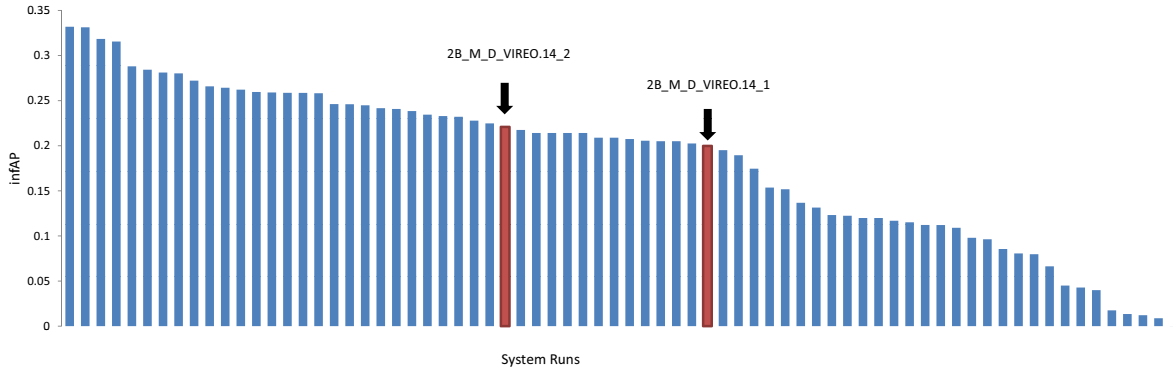

Figure 6: Mean infAP of all 75 SIN full version runs in TRECVID SIN 2014.

Figure 6 shows the mean infAPs of all 75 runs for this year, where our two submitted runs are marked in red. The first run (2B_M_D_VIREO.14_1) is only based on visual features, while our second run (2B_M_D_VIREO.14_2) takes visual, motion and audio features jointly. As shown, the second run

improves the overall performance significantly from 0.200 to 0.221. The improvement is even clear in Figure 7, which further details the performance on individual concept. By featuring the motion and audio information, our second run shows clear advantage over the first run, especially for concepts highly related to motion and audio, e.g., "Cheering", "Running" and "Instrumental_musician". On the other hand, those concepts, on which the second run descreases the performance, are mostly depicting static objects, e.g., "Flags" and "Lakes". Our experiment confirms that the motion and audio features can provide complementary information to visual featues.

To better understand the performance gain, we also internally evaluate the contribution of each individual feature on last year's concepts. As expected, the DCNN feature gives the best performance by outperforming other hand-crafted features. This observation also coincides with that in [13].

To highlight the progress made over years, we re-run our system on last year's testing set (IACC.2.A). Encouraingly, a big performance jump from 0.153 to 0.263 is observed, which implies the power of our newly added features this year.

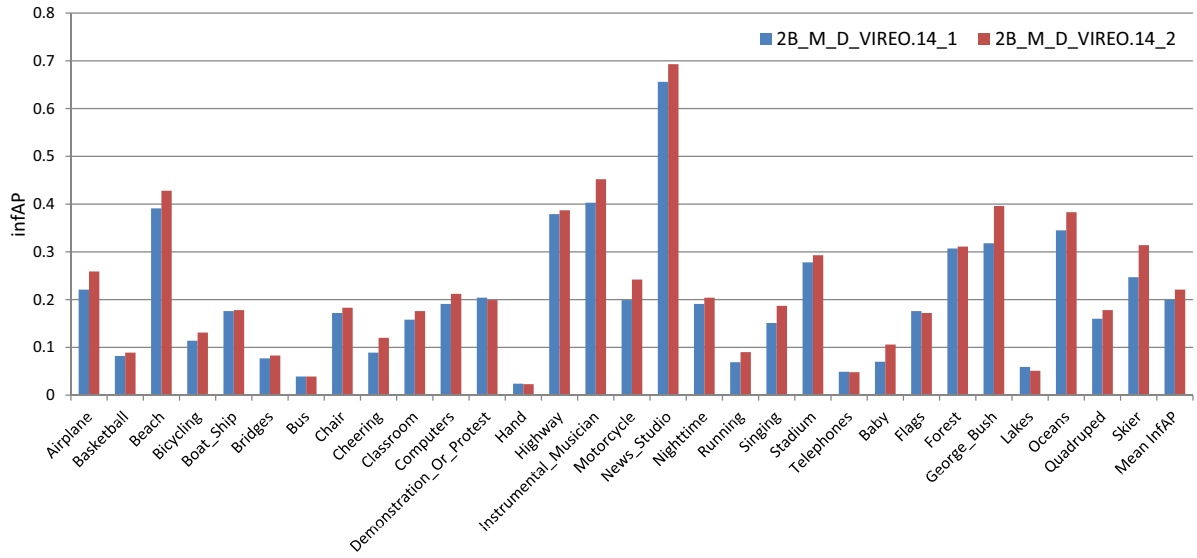

Figure 7: Per-concept performance of our submitted systems.

3 Summary

For INS, we experimented three aspects for object retrieval: baseline, normalization method, and video augmented query. Based on the experiments and analysis, we summarize as follows: (1) the baseline shows consistent results as last year, where Delaunay Triangulation and Context Modeling contribute significantly for our performance; (2) the new normalization method is able to retrieve some novel instances that are missed in baseline. (3) the video augmented query is not helpful in most cases because of the low visual quality.

For SIN, we mainly tested various features for concept detection, including the visual, motion and audio features. Our findings can be summarized as follows: (1) our newly added motion and audio features are complementary to the visual features, boosting the performance significantly when combined with visual features; (2) in terms of single feature, DCNN outperforms other hand-crafted features.

Acknowledgment

The work described in this paper was supported by three grants from the Research Grants Council of the Hong Kong Special Administrative Region, China (CityU 118812), the National Natural Science Foundation of China under Grant 61272290, and National Hi-Tech Research and Development Program (863 Program) of China under Grant 2014AA015102. We also thank BBC for providing the EastEnders videos: Programme material © BBC.

References

[1] 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.

[2] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *ICCV*, vol. 2, Oct. 2003, pp. 1470–1477.

[3] H. Jégou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *ECCV*, 2008.

[4] W. Zhang, L. Pang, and C.-W. Ngo, "Snap-and-ask: Answering multimodal question by naming visual instance," in *ACM international conference on Multimedia*, 2012.

[5] W. Zhang and C.-W. Ngo, "Searching visual instances with topology checking and context modeling," in *International conference on multimedia retrieval*, 2013, pp. 57–64.

[6] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *International Conference on Computer Vision*, 2013.

[7] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "YAAFE, an easy to use and efficient audio feature extraction software," *The International Society for Music Inforamtion Retrieval*, 2010.

[8] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, no. 2, pp. 91–110, 2004.

[9] C.-W. Ngo, F. Wang, W. Zhang, C.-C. Tan, Z. H. Sun, S.-A. Zhu, and T. Yao, "Vireo/ecnu @ trecvid 2013: A video dance of detection, recounting and search with motion relativity and concept learning from wild," in *NIST TRECVID Workshop*, 2013.

[10] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *IEEE International Conference on Computer Vision*, 2007.

[11] K. Zhang, L. Zhang, M.-H. Yang, and D. Zhang, "Fast tracking via spatio-temporal context learning," in *ECCV*, 2014.

[12] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutionalneural networks," in *Advances in Neural Information Processing System*, 2012.

[13] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," *arXiv preprint arXiv:1310.1531*, 2013.