National Institute of Informatics, Japan at TRECVID 2014

Duy-Dinh Le [1], Sang Phan [1], Vinh-Tiep Nguyen [2], Cai-Zhi Zhu [3],
Duc M. Nguyen [4], Thanh Duc Ngo [5], Siriwat Kasamwattanarote [1],
Poullot Sebastien [1], Minh-Triet Tran [2], Duc A. Duong [5], and Shin'ichi Satoh [1]

[1] National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo, Japan 101-8430
[2] University of Science, VNU-HCMC
227 Nguyen Van Cu, Dst 5, Ho Chi Minh City, Vietnam
[3] Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Japan 464-8601
[4] University of Engineering and Technology, VNU
144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
[5] University of Information Technology, VNU-HCMC
Quarter 6, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Abstract. National Institute of Informatics (NII) participated in two tasks: Instance Search (INS) and Multimedia Event Detection (MED). In the first part, we describe our Instance Search system that achieved best performance in this year's evaluation. In the second part, we describe our Multimedia Event Detection system along with some technical improvements for MED.

NII at TRECVID 2014
Instance Search Task

Duy-Dinh Le [1], Vinh-Tiep Nguyen [2], Cai-Zhi Zhu [3],
Duc M. Nguyen [4], Thanh Duc Ngo [5], Siriwat Kasamwattanarote [1],
Poullot Sebastien [1], Minh-Triet Tran [2], Duc A. Duong [5], and Shin'ichi Satoh [1]

[1] National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo, Japan 101-8430
[2] University of Science, VNU-HCMC
227 Nguyen Van Cu, Dst 5, Ho Chi Minh City, Vietnam
[3] Graduate School of Information Science, Nagoya University
Furo-cho, Chikusa-ku, Nagoya, Japan 464-8601
[4] University of Engineering and Technology, VNU
144 Xuan Thuy, Cau Giay, Hanoi, Vietnam
[5] University of Information Technology, VNU-HCMC
Quarter 6, Linh Trung Ward, Thu Duc District, Ho Chi Minh City, Vietnam

Abstract. We report the two Visual Instance Search Systems in TRECVID 2014. The first system is based on our system last year and is improved by using a new spatial consistency enforcement method. Firstly, top10K shots are returned by the baseline BoW system. Next, the DPM-based object localizer trained on query image regions is used to apply on these top10K shots to detect possible locations of the target object. In addition, RANSAC is also used to check spatial consistency between matches of the query images and representative keyframes of the shots. Finally, the final score of these shots is computed by fusing the scores of BoW, DPM, and RANSAC. The key idea of our proposal is to design a flexible scoring scheme to prioritize shots whose matched regions by RANSAC and DPM are more consistent. Since RANSAC is point-based and DPM is region-based spatial consistency verification, they are complementary each other. As a result, many false positives are removed, leading to better precision. Meanwhile, BoW-based top10K guarantees reasonable recall. Using this new spatial consistency enforcement method, the performance is improved 128.8%.
The second system is based on the state of the art BoW model and use video for query expansion. The experiments show minor improvement compared to the baseline.

1 Visual Instance Search Framework Using DPM-based Object Localizer for Enhancing Spatial Consistency Enforcement

Our Instance Search system is designed following the guideline provided by TRECVID [1]. For our this year system, we propose a new method of post processing to deal with texture-less and feature-less query objects. Most state-of-the-art systems last year using Bag-of-Word (BOW) model [2] combined with RootSIFT [3], an extended version of SIFT[4]. The main contribution of our team last year is to propose a new query adaptive asymmetric dissimilarity to handle problem of query-image relation: the query object is mostly included in the database image, while the converse is not necessarily true. We also combined many feature detectors and descriptors but the system improve performance not very much, as special for texture-less object such as logos, small objects. Because we use different features with the same model to represent image, they are not complementary together. This year, we propose a new region of interest based reranking method using BOW model and object detection algorithm.

Specifically, our the BOW baseline has the following settings. For the local feature extraction, we use hessian-affine detector [5] and RootSIFT descriptor[3]. These features are sampled and clustered to a one million-sized codebook. For fast clustering and quantization, we use approximate k-mean and fast approximate nearest neighbours library[6][7]. Each frame is represented by a high dimensional BOW vector using tf-idf weighting scheme. Each shot is represented by aggregating all BOW vector of frames using average pooling technique. Note that, to reduce the effects of quantization error, we use soft-assignment technique for each feature [8]. In our experiment, there are two ways of quantization: soft-assignment on database and query (soft-soft), soft-assignment on query image only (hard-soft). Our experiment shows that, hard-soft assignment are comparable to soft-soft assignment on both database and query image. However, quantization on query side saves a lot of memory and computational cost when comparing to database side. For fast retrieval, we build inverted index for all of shots in the database. To compare a query BOW vector with a shot BOW vector, we use asymmetric dissimilarity proposed last year [9]. The rank list of this baseline will be used for region of interest based reranking later.

BOW model is a non-structured model, therefore, we need a geometric verification post processing to rerank final list of previous step. However, both BOW and geometric verification only work well when queries and shot frames actually have much enough shared visual words. To solve this problem, we propose a new post processing method using object detection based method. Base on rank list of the baseline, we use Deformable Parts Model (DPM) object detection algorithm [10] to find bounding box containing query object. DPM works very well on textureless objects because it based on shape structure of query object while BOW only works with much texture objects. That is the reason why these two methods are complementary together. We apply DPM algorithm on rank list from previous BOW step. Each DPM and BOW has a rank list with scores. In a naive way, we simply compute average on normalized similarity scores of these methods. This combination improve the performance significantly although we only one feature detector and one descriptor. However, to take into account both bounding box of DPM and shared visual word of DPM, we propose a new similarity score formula:

$$S_{new} = (1 + N_d)^2 + (1 + N_f g - N_d) log_2 N_{bg}(w_1.S_{BOW} + w_2.S_{DPM})$$

where,

$S_{new}, S_{BOW}, S_{DPM}$: score of new score, score of Bag of Word model and DPM respectively

N_d: number of shared words of foreground inside bounding box

N_fg: number of shared word of foreground

N_bg: number of shared word of background

w_1: weight of BOW score

w_2: weight of DPM score

Figure 1 illustrates the idea of our proposed score. The red lines are outlier pair matching which will be discarded using geometric verification algorithm such as RANSAC. is number of green lines, is number of both green and blue lines, is number of light blue lines. In our experiment we set parameters and .

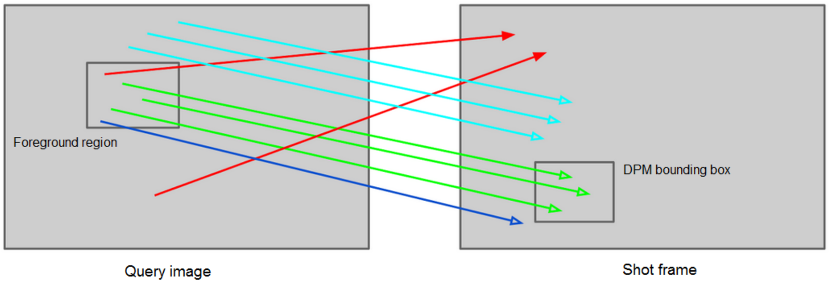


Fig. 1. Reranking using shared word matching and DPM bounding box combination

For our performance at TRECVID INS 2014, we got three first positions when using the same technique as illustrated above. Run $F\_D\_NII\_2$ is using hard-soft assignment baseline while $F\_D\_NII\_1$ uses soft-soft assignment. We also use the best config which archives first position at TRECVID INS 2013 (the system combining 3 detectors, 2 descriptors) with our new reranking method (run $F\_D\_NII\_3$). The performance Table 1 shows that, using one detector, one descriptor with hard-soft assignment baseline archives the best in mAP.

Table 1. Our TRECVID INS 2014 result.

Runs	mAP(%)	Position
F_D_NII_2	32.46	1st
F_D_NII_1	32.43	2nd
F_D_NII_3	32.19	3rd

Figure 2 is our performance of run $F\_D\_NII\_2$ when comparing to the median and best score of each query. Figure 3 illustrates an example of our algo-

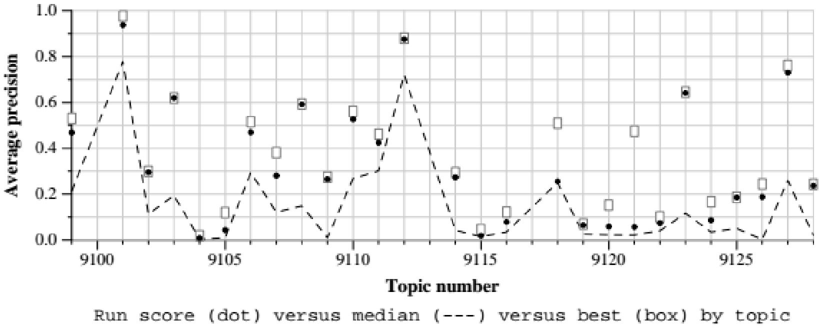


Fig. 2. Performance of run *F_D_NII_2* using hard assignment on database and soft assignment on query image.

rithm. The first row is visualization of Mercedes DPM trained model.The second row shows an example of true matching but wrong object. In this case, even having a good asymmetric dissimilarity, the BOW similarity score still recognize the chair as the given Mercedes logo because of many inlier shared words. The last row shows that, DPM bounding box (yellow rectangle) supports for the foreground shared words to be relevant to query object. These visual words in the bounding box will boost the similarity score higher than others.

2 Visual Instance Search Framework Using Video for Query Expansion

The image and video retrieval system is based on several state-of-the-art approaches. However, we selected an off-line-on-line architecture that exists in [9] and [11] (Fig. 4). This system will run in a client-server fashion in the near future.

The off-line part consists in constructing the bag-of-word (BoW) database. First the well-known local feature SIFT are extracted at the Hessian Affine detector locations, as was done in [7], on regularly sampled frames from the videos (5fps). The codebook is computed by a classic K-means on randomly sampled descriptors, the codebook size is set to 1 million words. The assignment to the visual word is performed with an approximate search in the FLANN way [12, 13] to construct the bag-of-word signature for each frame. Then an average pooling is performed in order to produce a shot signature. Plus, to reduce the effect of burstiness, the power-law normalization is applied along with the tf-idf on word frequency. Finally, for the fast retrieval (online part), the BoW are indexed in an inverted index.

The online part corresponds to the processing of the query topics. As done off-line, a BoW representation is computed for each query image. This BoW is to be searched in the inverted index database.

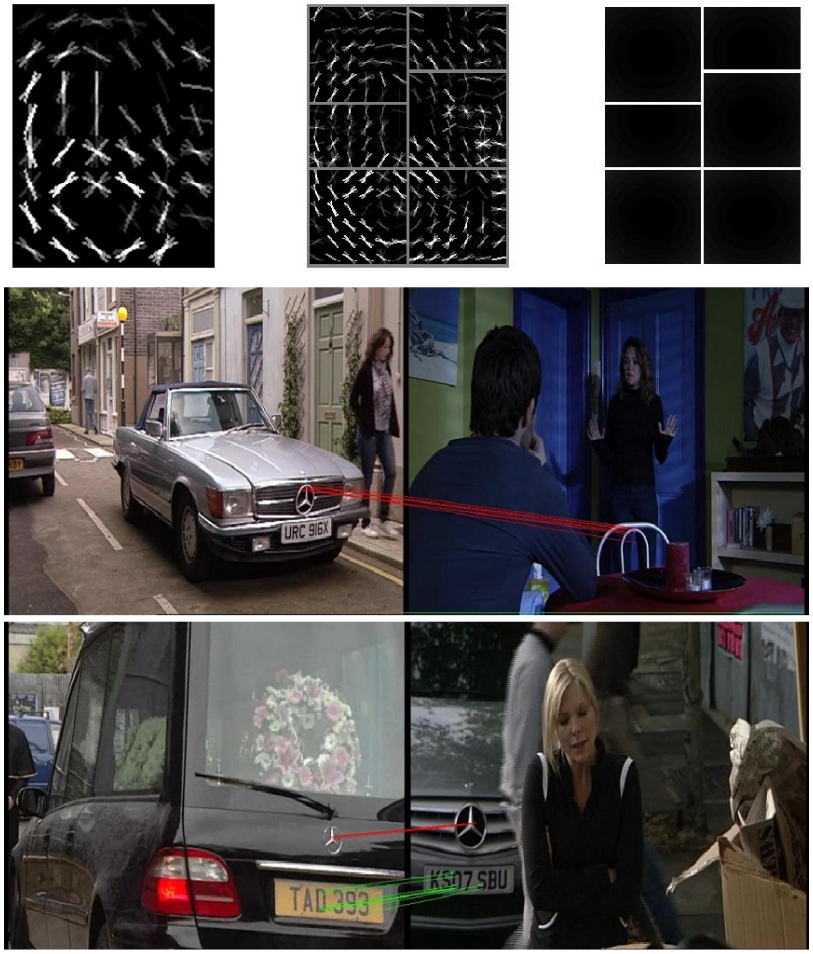


Fig. 3. First row: DPM model of Mercedes logo. Second row: shared word matching after geometric verification. Third row: Reranking combining BOW model and DPM object detection algorithm. (Programme material copyrighted by BBC.)

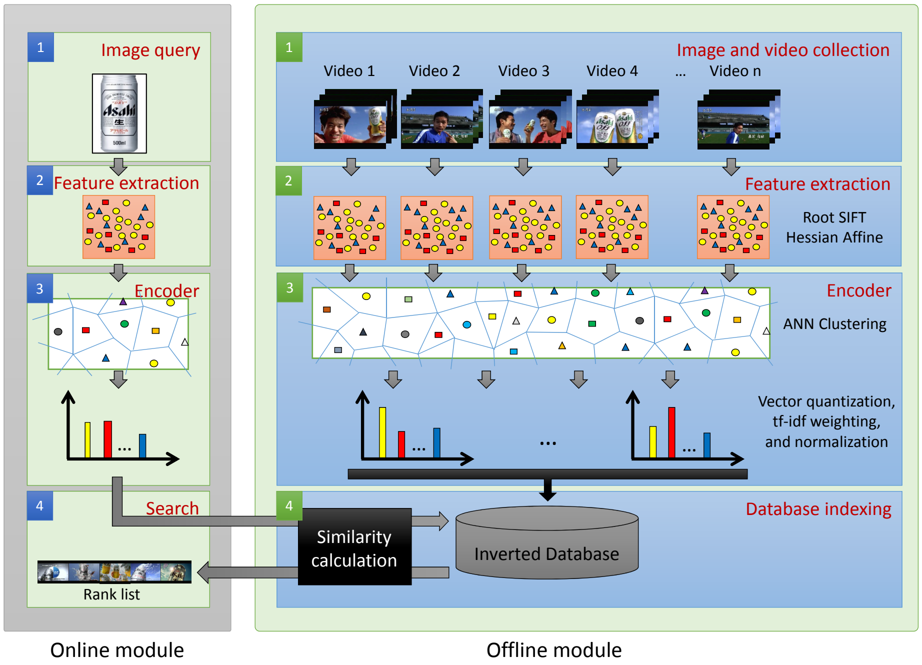


Fig. 4. A system architecture of our core retrieval with off-line and on-line module.

2.1 Query Expansion Using Video

We apply a video expansion technique for each query frames provided in run D (4 images) as follows.

First all frames from all query video are extracted using ffmpeg. We noticed that the video are interlaced therefore some adapted filter were used (yadif), however the quality of the frames is very low when there is some movement (typical streaks appear, making the local descriptors very noisy).

Each query image is compared to the list of frames extracted from its corresponding video in order to find the copy one (undergoing the quality degradation). For robustness, the distance between frames is composed of a pixel-to-pixel intensity distance and a RGB histogram distance.

Form the original frame, an extension is made backward and forward, basically 5 frames on each side. If the extension is not possible on one side (shot boundary, fast moves, etc) the extension will be longer on the other side.

Finally on the central frame, SURF detector is applied and SURF descriptors are extracted. Only the ones belonging to the mask from the image query are kept. These descriptors are tracked backward then forward: compute the SURF in an enlarged mask version (a bounding box) on the next frame, perform the descriptor matching and filter with RANSAC. The homography found between the two sets of points is used to project the current bounding box in the next frame. This process is done iteratively until the end of the frame expansion. In case of small objects the original bounding box is stretched a bit in order to capture more local descriptors.

The expanded frames with their masks (bounding boxes) are simply provided to our framework as new instances of the objects.

2.2 Experiments

We re-implemented the baseline shown in figure 4 with pure C++ for the high-speed purpose. Our experiments mostly run on 40 cores of Xeon 2.4 GHz CPU. We extracted 7.8 million keyframes in 470,000 shots, in total by using only one SIFT feature, we then have 9.8 billion keypoints.

The offline feature extraction run at 40 fps, overall 3 days were necessary for the feature extraction, 3 days for clustering, 6 day for ANN then 10 hours to construct the 57 GBs database (BoW + inverted index).

The online process, with the 100% cache hit, can retrieve over 30 topics in less than a second (excluding feature extraction). However some disk accesses must be done due to the memory limitation, consequently it takes between five seconds and one minute for one topic. The speed depends on how much cache hits for that topic, and how much cache misses must be load from the disk.

In this year TRECVID 2014 [1], we submitted one run for the four images query task (D4), three runs for the video query task (E1,E2,E3), and one run is the late-fusion result of our E3 run with our D1 run. We run our re-implemented baseline with 4 parameters on both 4 images (run D) and our video expansion

Table 2. Our performance with description

RunID	Type	Description (only different)	MAP
F_D_NII_4	4 images	Stopword-1K, latefusion	20.00
F_E_NII_1	video	Stopword-1K, latefusion	20.50
F_E_NII_2	video	Stopword-1K, earlyfusion	1.20
F_E_NII_3	video	Stopword-10K, latefusion	20.60
F_E_NII_4	video	Stopword-10K, latefusion, NIID1-latefusion	**30.10**

technique on a video query (run E) with the acceptable performance even we use only one SIFT feature.

For our submission, we fixed some parameters: sifthesaff (Hessian Affine detector with SIFT descriptor), 1M-word (1 million codebook), powerlaw-bgonly (power-law normalization is applied only on the background region), fg-bg-weight-0.9 (the respective weights for foreground and background descriptors are 0.9 and 0.1). The rest of the parameters for each run are indicated in Table 2, and the performances evaluated by [14] were shown in figure 5 and 6.

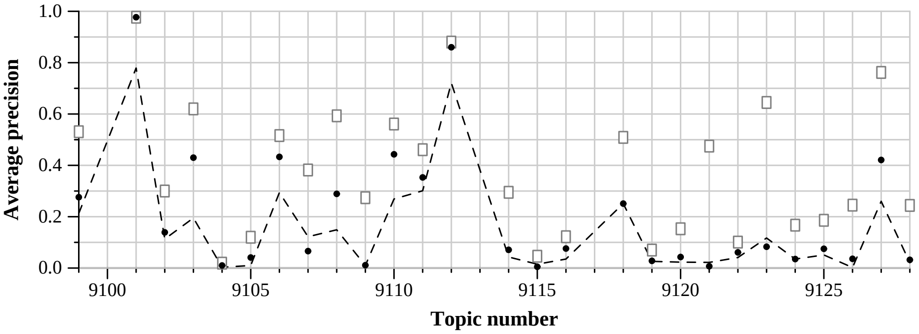


Fig. 5. Our performance (dot) vs. median vs. best for each topic.

We got a final MAP above 20%, the variation depends on these parameters: late fusion or early fusion, and how many stop word. After investigation on INS2013 dataset, the stop word was set to 10k, and late fusion was preferred for each image query. Also thanks to our novel video expansion, we got a small improvement of the performance compared to the four images framework. Nevertheless, as mentioned before, we think that not interlaced video queries would have given even better results.

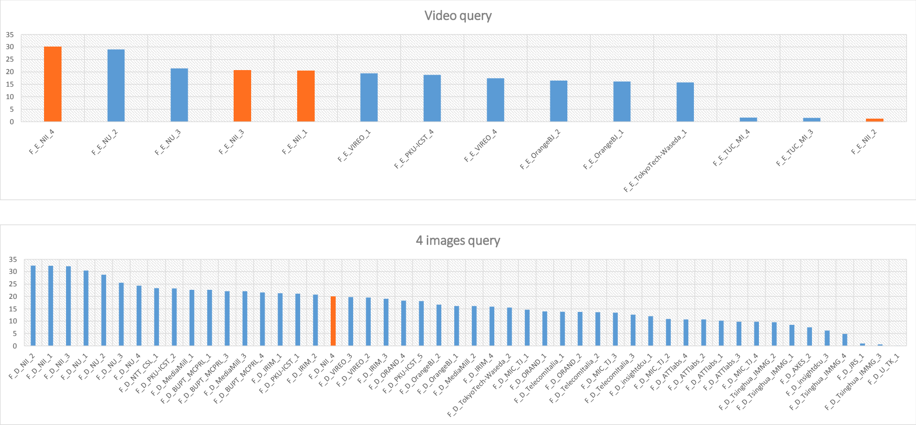


Fig. 6. Our performance (orange) comparing with total runs (blue).

References

1. Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Wessel Kraaij, Alan F. Smeaton, and Georges Queenot, "Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2014*. NIST, USA, 2014.
2. Josef Sivic and Andrew Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003.
3. R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *CVPR*, 2012.
4. David G. Lowe, "Distinctive image features from scale-invariant keypoints," 2004.
5. K. Mikolajczyk and C. Schmid, "An affine invariant interest point detector," in *Proceedings of the 7th European Conference on Computer Vision-Part I*, 2002, ECCV '02.
6. "Fast clustering library," http://www.robots.ox.ac.uk/ vgg/software/fastanncluster/.
7. J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
8. J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008.
9. C.-Z. Zhu, H. Jegou, and S. Satoh, "Query-adaptive asymmetrical dissimilarities for visual object retrieval," in *ICCV*, 2013.
10. P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008.
11. Cai-Zhi Zhu and Shin'ichi Satoh, "Large vocabulary quantization for searching instances from videos," in *ICMR*, 2012.
12. Marius Muja and David G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration.," in *VISAPP*, 2009.

13. Michal Perdoch, Ondrej Chum, and Jiri Matas, "Efficient representation of local geometry for large scale object retrieval.," in *CVPR*, 2009.
14. 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.

NII at TRECVID 2014
Multimedia Event Detection

Sang Phan, Duy-Dinh Le, Shin'ichi Satoh
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430

Abstract

We report our Multimedia Event Detection (MED) system in TRECVID 2014. Our system consists of following parts: (1) pre-processing, (2) feature extraction, (3) feature representation and (4) event detection. We use both audio and visual features with Fisher vector encoding. In the evaluation, we compare the technical improvements of using motion, image and audio features. We also evaluated different ways to use related videos. We submitted our MED system in the full evaluation for both EK100 and EK10 setting.

1 Data Resources

Currently, we only use data provided by TRECVID [1] in our evaluation. Following the TRECVID 2014's guidelines [2], we use events in the Event Kit for training event detection models. Background videos are used to train the visual and audio codebooks. We haven't use the Research Collection yet. In order to test the performance of our system, we use the KINDREDTEST 13 and KINDREDTEST 14 dataset.

2 MED Framework

The MED framework is shown in Figure 1. Basically, it consists of following steps: preprocessing, feature extraction, feature representation and event classification.

2.1 Preprocessing

At first, all videos are normalized to around 320x240. We fix the width dimension to 320 and change the height so that the aspect ratios are kept. The frame rates are also kept, however, for those videos that have frame rate larger than 50 fps, we use the standard one instead, i.e. 25 fps. The audio channels are

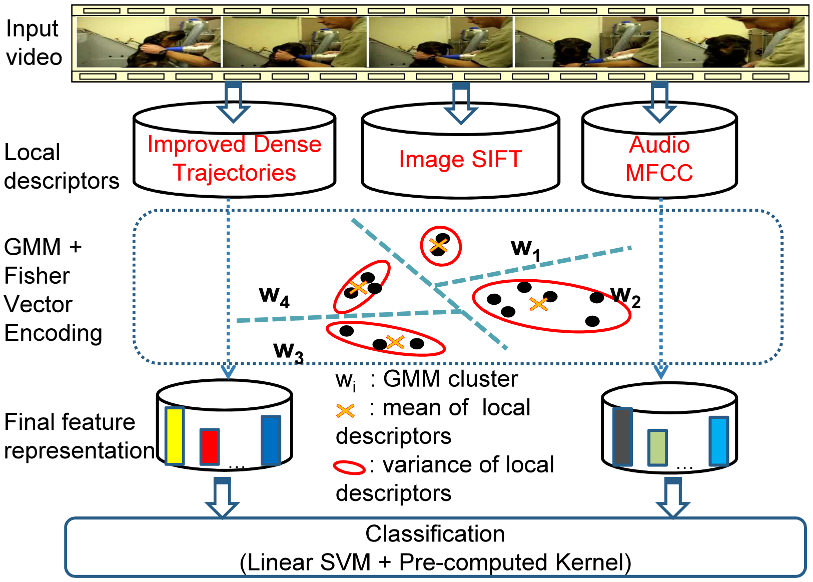

Input video
Local descriptors
Improved Dense Trajectories
Image SIFT
Audio MFCC
GMM + Fisher Vector Encoding
w1
w2
w3
w4
Final feature representation
wi : GMM cluster
× : mean of local descriptors
⬭: variance of local descriptors
Classification
(Linear SVM + Pre-computed Kernel)

Figure 1: Our MED framework

removed from resized videos to save disk space. After that, we extract one representative keyframe from resized videos at every two seconds and audio feature from the original videos.

2.2 Feature Extraction

We use feature from different modalities to model multimedia events: still image features, motion features and audio features. We use the standard SIFT with Hessian Laplace detector for extracting still image feature. For motion feature, we use Improved Dense Trajectories with the combination Hitogram of Oriented Gradient, Histogram of Optical Flow (HOGHOF) and Motion Boundary Histogram (MBH) descriptors. We use the library provided online by the authors, and keep all the default parameters. We use the popular MFCC for extracting audio feature. We choose a length of 25ms for audio segments and a step size of 10ms. The 13d MFCCs along with each first and second derivatives are used for representing each audio segment.

2.3 Feature Representation

Fisher vector representation is a newly developed feature encoding technique where the mean and variance of local descriptors that belong to each cluster are also calculated. Therefore, Fisher vector encodes more information than the bag-of-words feature encoding. Following the standard implementation of Fisher vector, we use the codebook size of 256 clusters which are generated using the Gaussian Mixture Model (GMM). We further improve the expressiveness of Fisher vector by applying PCA for reducing feature dimension, i.e 80-d for SIFT and 128-d for MBH.

2.4 Event Detection

We use the popular Support Vector Machine (SVM) for training event detectors. All the positive videos are considered as positive samples. Background videos are used as negative samples. Miss videos or related videos can be used as negative or positive samples. It can be also excluded from training data. We will discuss how to use related videos in the next section.

LibSVM with linear kernel is incorporated to our system for learning steps because it is a standard implementation for SVM. Furthermore, we also utilized the pre-computed kernel technique to reduce the training time. This technique is especially useful when the number of events are large.

2.5 Fusion Methods

In order to combine features from different modalities, we use the late fusion approach. All features are combined with equal weights.

2.6 Threshold Learning

There are two thresholds that we need to determine for each event: (1) A confidence score value of the system-supplied threshold, and (2) A rank value of the system-supplied threshold. We learnt these two thresholds as follows. At first, we divide the event kit into two parts randomly. The first part contains 80% of event videos which are used for training. The second part contains the remaining videos in the event kit. After training the event detector, we use the second part to determine the thresholds. The score threshold is the mean score of all testing videos and the rank threshold is the mean rank of all testing videos.

3 Improvements over MED'13 System

3.1 For Motion Features

We tried the improved version of Dense Trajectories motion feature [3]. To describe trajectories, we choose to use both HOGHOF and MBH descriptors, which have been proved to be effective for MED by AXES team [4]. In order to combine these two descriptors, we train two independent GMM codebooks. After that Fisher vector is used to encode feature from each descriptor independently. The resulting feature representation at video level of each descriptor is normalized by power normalization and L2 normalization. Finally these two feature vectors are concatenated to form the final motion feature representation of each video.

3.2 For Image Features

We applied two technical improvements on the image feature. At first, a new way of video level feature representation is used to pool feature from its keyframe-based representation. In the last year's system, we aggregated local descriptors from all sampled frames in video without explicitly calculating keyframe-based features. For this year's system, Fisher vector is encoded for each sampled frame and normalized using power and L2 normalization. Features from these sampled frames are averaged to form the video level representation.

The second technical improvement is using RootSIFT features [5]. We have applied RootSIFT with different implementation of SIFT features such as the one use in [6], VLFeat [7], and Color Descriptor [8]. Finally we chose to use VLFeat because it achieved the best performance in our evaluation framework.

3.3 For Audio Features

We investigated several ways to extract MFCC features from audio channel. These MFCC libraries are used in our evaluation: VoiceBox audio toolkit [9], Yaafe audio library [10] and the RASTA-PLP library [11]. We found that the RASTA-PLP implementation achieved slightly better performance than others. Moreover, we did not observe significant improvement when changing parameters such as window length and step between successive windows. So we kept using the default setting in the RASTA-PLP implementation.

3.4 How to Use Related Videos?

Related videos or miss videos are videos that contains some related information to a particular event. However, these videos are not considered as event videos because of missing some key evidences. There are three straight forward approaches to use related videos at the learning step: (1) Related videos as positive videos (RP), (2) Related videos as negative videos (RN) and (3) Without using related videos (NR).

4 Contribution of New Components

We evaluated the contribution of new components on the KINDREDTEST 13 dataset. We always observe the same trends on both the KINDREDTEST and MEDTEST dataset, where the testing event videos are shared. So we only test our system on the KINDREDTEST dataset. All results are reported in terms of Mean Average Precision (MAP). Here we only report the over all performance, which is averaged from all events.

4.1 Improvement of Motion Features

Performance comparison of Dense trajectories and Improve trajectories are shown in Table 1. The improved version of Dense Trajectories has better performance, which confirms the effectiveness of removing camera motion when detecting motion feature in video. We even obtain much better result when combining HOGHOF descriptor with MBH.

Table 1: Performance comparison of different motion feature configurations.

MED13 System	MED14 System	
Dense Trajectories (MBH)	Improved Dense Trajectories (MBH)	Improved Dense Trajectories (HOGHOF + MBH)
28.33	35.07	40.77

4.2 Improvement of Image Features

Performance comparison of image features are shown in Table 2. The new aggregation is slightly better than the previous one. However, we can obtain more improvement when applying RootSIFT feature.

Table 2: Performance comparison of different image feature configurations.

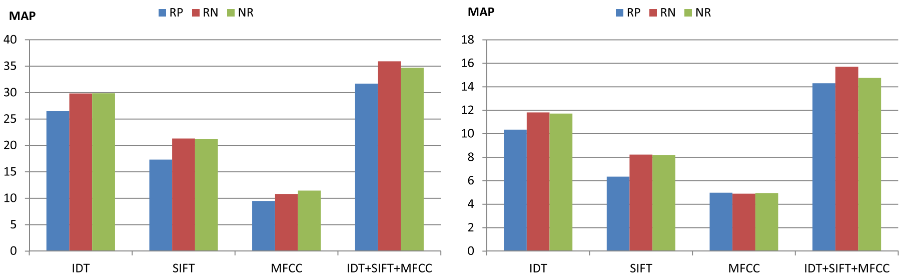
MED13 System	MED14 System	
SIFT	SIFT (New aggregation)	SIFT (New aggregation + RootSIFT)
23.41	24.24	27.02

4.3 How to Use Related Videos?

We evaluated three ways of using related videos as mentioned in Section 3.4. The results are shown in Fig. 2. We found that using related videos as negative training samples always achieves better performance than using related videos as positive ones. The performance difference between using related videos as negative and without using related videos is minor. However, if we combine multiple features, we can always obtain better results than other methods for both EK100 and EK10 settings.

5 Submitted Systems

After evaluating the technical improvements on the KINDREDTEST dataset, we chose the best configuration of each feature and incorporated it in our final system. (1) For motion features, unfortunately, we could not finish running the



(a) On EK100 setting

(b) On EK10 setting

Figure 2: Comparison of different ways to use related videos: Related as Positives (RP), Related as Negatives (RN) and No Related (NR).

best configuration, so we use the same configuration as previous year because it took less computational time; (2) for image features, we use the new configuration, i.e. new aggregation method combined with RootSIFT feature; (3) we use the best configuration as mentioned in Section 3.3. These features are highly complementary as shown on Fig. 2, so we used the late fusion technique to combine these features in our final submission. Finally, for using the related videos, we fixed our system to use them as negative training samples for both EK10 and EK100 settings. We participated in the full evaluation set containing around 200K videos. We used the same system for both Pre-specified (PS) and Adhoc (AH) tasks.

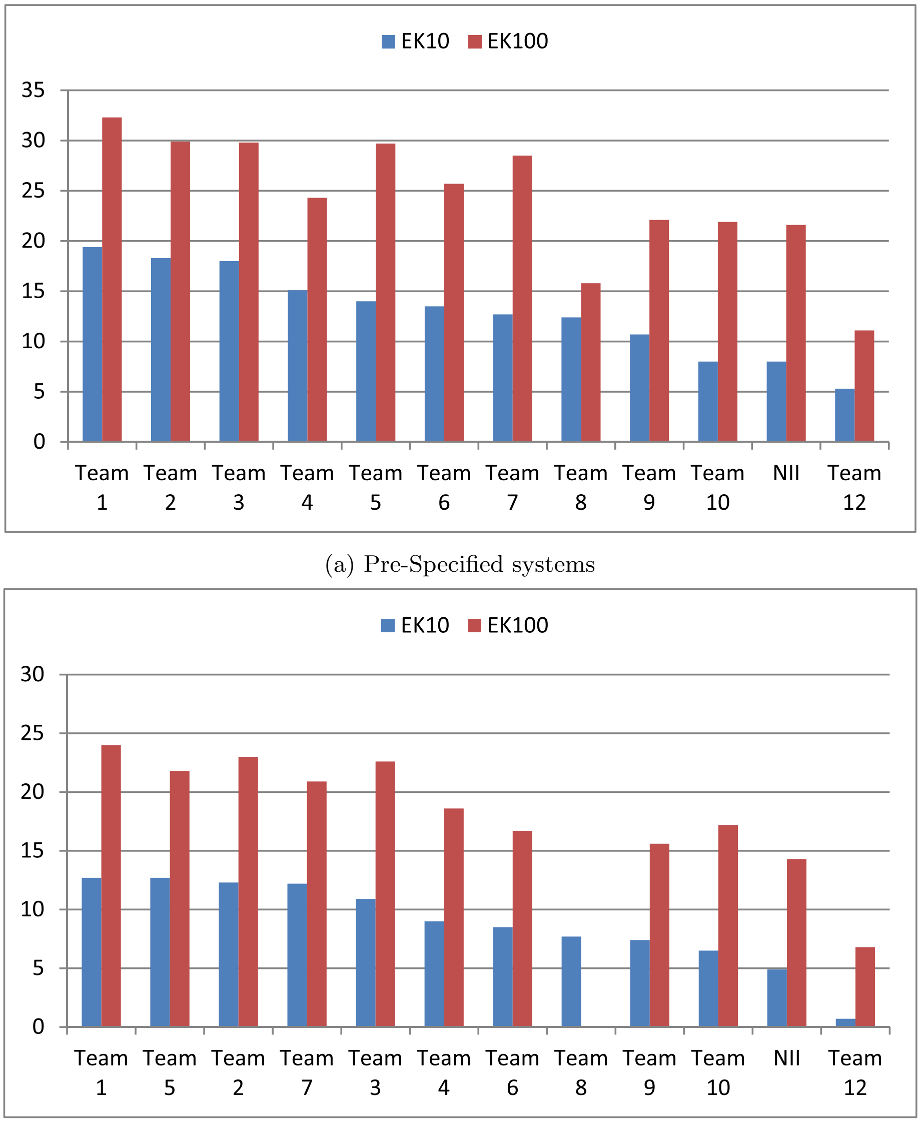
6 Result and Conclusion

Results of our MED system on the full evaluation set is shown in Fig. 3. Comparing with other systems, we are ranked 11th out of 12 teams in the EK10 setting and ranked 10th in the EK100 setting. This observation is same for both PS and AH tasks.

Compared to top MED systems, our system is significantly worse in the EK10 setting. For example, our performance are 67% and 41% relatively to the best MED system in the EK100 and EK10 respectively. We have learnt that top performance system have incorporated semantic concept detection, which can be more helpful when number of training videos are limited. This might be the reason for the significant drop on the performance of our EK10 system.

References

[1] Stephanie Strassel, Amanda Morris, Jonathan Fiscus, Christopher Caruso, Haejoong Lee, Paul Over, James Fiumara, Barbara Shaw, Brian Antonishek, and Martial Michel, "Creating havic: Heterogeneous audio visual



(a) Pre-Specified systems

(b) Ad-Hoc Systems

Figure 3: Comparison of our MED system with others on the full evaluation set for both PS and AH tasks. Results are sorted in the descending order of performance on the EK10 setting.

internet collection," in *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Uur Doan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, Eds., Istanbul, Turkey, may 2012, European Language Resources Association (ELRA).

[2] Paul Over, George Awad, Martial Michel, Jonathan Fiscus, Greg Sanders, Wessel Kraaij, Alan F. Smeaton, and Georges Quenot, "Trecvid 2014 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2014*. NIST, USA, 2014.

[3] Heng Wang and Cordelia Schmid, "Action recognition with improved trajectories," in *IEEE International Conference on Computer Vision*, Sydney, Australia, 2013.

[4] Robin Aly, Relja Arandjelovic, Ken Chatfield, Matthijs Douze, Basura Fernando, Zaid Harchaoui, Kevin McGuinness, Noel E O'Connor, Dan Oneata, Omkar M Parkhi, et al., "The axes submissions at trecvid 2013," 2013.

[5] Relja Arandjelovic and Andrew Zisserman, "Three things everyone should know to improve object retrieval," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2911–2918.

[6] Krystian Mikolajczyk and Cordelia Schmid, "A performance evaluation of local descriptors," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 27, no. 10, pp. 1615–1630, 2005.

[7] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," 2008.

[8] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, 2010.

[9] "Voicebox: Speech processing toolbox for matlab," http://www.ee.ic.ac.uk/hp/staff/dmb/voicebox/voicebox.html.

[10] Benoit Mathieu, Slim Essid, Thomas Fillon, Jacques Prado, and Gaël Richard, "Yaafe, an easy to use and efficient audio feature extraction software.," .

[11] Daniel P. W. Ellis, "PLP and RASTA (and MFCC, and inversion) in Matlab," 2005, online web resource.