Informedia@TRECVID 2013

Zhen-Zhong Lan, Lu Jiang, Shoou-I Yu, Chenqiang Gao, Shourabh Rawat, Yang Cai, Shicheng Xu, Haoquan Shen, Xuanchong Li, Yipei Wang, Waito Sze, Yan Yan, Zhigang Ma, Nicolas Ballas, Deyu Meng, Wei Tong, Yi Yang, Susanne Burger, Florian Metze, Rita Singh, Bhiksha Raj, Richard Stern, Teruko Mitamura, Eric Nyberg, and Alexander Hauptmann

Carnegie Mellon University

In the first part of this three-part report we describe our system and novel approaches used in the TRECVID 2013 Multimedia Event Detection (MED) and Multimedia Event Recounting (MER) tasks. A separate section of the report (SIN) details methods and results for the Semantic Indexing task. The final section (SED) describes our approaches and results on the Surveillance Event Detection task.

Informedia E-Lamp @ TRECVID 2013
Multimedia Event Detection and Recounting (MED and MER)

Zhen-Zhong Lan, Lu Jiang, Shoou-I Yu, Shourabh Rawat, Yang Cai, Chenqiang Gao, Shicheng Xu, Haoquan Shen, Xuanchong Li, Yipei Wang, Waito Sze, Yan Yan, Zhigang Ma, Wei Tong, Yi Yang, Susanne Burger, Florian Metze, Rita Singh, Bhiksha Raj, Richard Stern, Teruko Mitamura, Eric Nyberg, and Alexander Hauptmann

Carnegie Mellon University

CMU Informedia @TREVID 2013
Semantic Indexing (SIN)

Lu Jiang, Shicheng Xu, Zheng-Zhong Lan, Waito Sze, Alexander Hauptmann
Carnegie Mellon University

CMU Informedia @TREVID 2013:
Surveillance Event Detection (SED)

Chenqiang Gao, Yang Cai, Haoquan Shen, Wei Tong,

Yi Yang, Nicolas Ballas, Deyu Meng, Yan Yan, Alex Hauptmann

Carnegie Mellon University

Informedia E-Lamp @ TRECVID 2013

Multimedia Event Detection and Recounting

Zhen-Zhong Lan, Lu Jiang, Shoou-I Yu, Shourabh Rawat, Yang Cai, Chenqiang Gao, Shicheng Xu, Haoquan Shen, Xuanchong Li, Yipei Wang, Wei Tong, Yi Yang, Waito Sze, Susanne Burger, Florian Metze, Rita Singh, Bhiksha Raj, Richard Stern, Teruko Mitamura, Eric Nyberg, and Alexander Hauptmann

Carnegie Mellon University
5000 Forbes Ave., Pittsburgh PA, 15213

Abstract

We report on our system used in the TRECVID 2013 Multimedia Event Detection (MED) and Multimedia Event Recounting (MER) tasks. For MED, it consists of four main steps: extracting features, representing features, training detectors and fusion. In the feature extraction part, we extract more than 10 low-level, high-level, and text features. Those features are then represented in three different ways which are spatial bag-of words, Gaussian Mixture Model Super Vectors (GMM) and Fisher Vectors. In the detector training and fusion, two classifiers and weighted double fusion method are employed. The official evaluation results show that our MED full systems achieve the best scores on Ah-Hoc EK10 and EK0, our audio systems achieve the best scores in EK100 and EK10 for both Pre-specified and Ad-Hoc tasks. Our MER system utilizes a subset of features and detection results from the MED system from which the recounting is generated.

1. MED System

1.1 System Overall

In this section, we give an overview of our MED systems. As shown in Figure 1, there are four key steps in our system. In step one, we perform feature extraction on visual, textual and audio modality. In step two, three different representations are used to aggregate the interesting point features into video level features. In step three, we calculate the kernel matrix and apply early fusion. Also in step three, classifiers are trained to perform the classification. In step four, the outputs of different classifiers are combined by using late fusion strategies.

Table 1 summarizes the features used in our system. Among those features, SIFT, CSIFT, TCH, Object Bank and DCNN are extracted from key-frames using the methods as described in [6, 11, 12]. Other features are extracted from videos following the procedures described in [11, 12].

Given raw features extracted from key-frames and videos, visual low-level features are represented by spatial bag-of-words, GMMs [1] and Fisher Vectors [11]; MFCC is represented by bag-of words. Average pooling [5, 6] is used to aggregate key-frame based features into video level features.

In this year's submissions, for all the tasks, we train both SVM and kernel ridge regression (KRR) classifiers [6, 11, 12] using two-fold cross-validation to choose the parameters. Average z-score fusion is used to fuse outputs different representations and classifiers for each feature. Ensemble weighted double fusion is used to combine the results of different features.

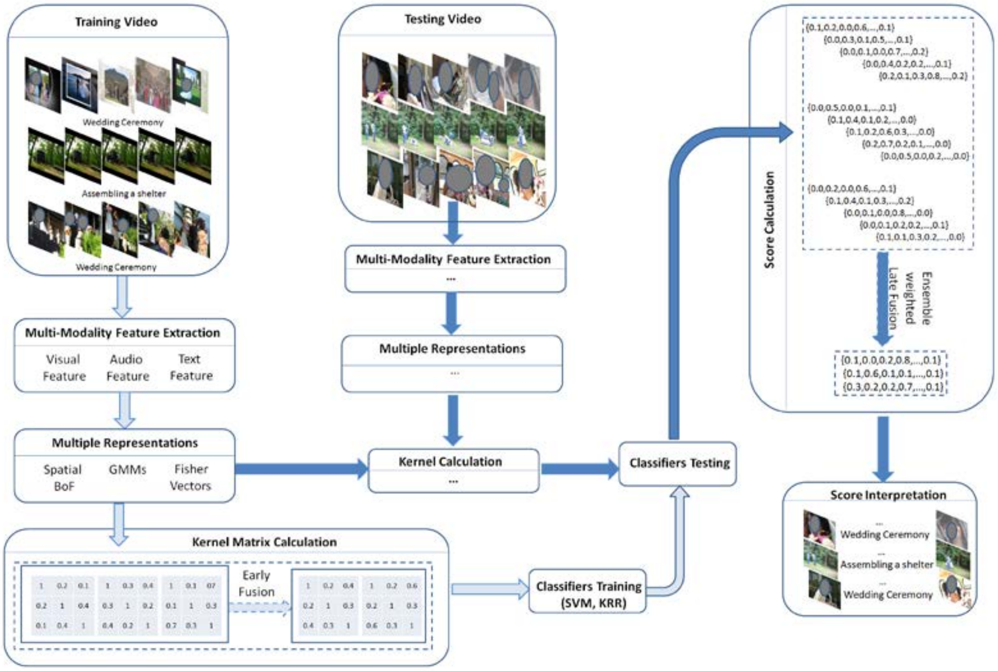


Figure 1, Informedia MED system illustration

Table 1, Features used for MED'13 system

	Visual Features	Audio Features
Low-level Features	SIFT [10] Color SIFT (CSIFT) [10] Motion SIFT (MoSIFT) [3] Transformed Color Histogram (TCH) [10] STIP [15] Dense Trajectory [14]	MFCC Acoustic Unit Descriptors (AUDs)[2] Large-scale pooling (LSF) Sub-band autocorrelation (SPBCA) [17] Log Mel sparse coding (LMEL) UC.8k
High-level Features	Semantic Indexing Concepts (SIN)[9] Object Bank (1000 concpets) [8] Deep Convolutional Neural Networks (DCNN) (1000 concepts) [4]	
Text Features	Optical Character Recognition (OCR)	Automatic Speech Recognition (ASR)

1.2 Improvements over MED'12 System

Besides using the features and machine learning methods we used last year [11], this year we introduce several new features, representations and fusion methods for MED.

1.2.1 New Features

LMEL -- Log-Mel-based Sparse Coding Features

These features are learned by training a single layer sparse coder in an unsupervised manner over LMEL Features. The features are trained over 7 frame stacked LMEL features (210 dimensions) as the input layer and a 60 dimensional output layer with a sparsity factor of 0.02.

Once the features are learned, encoding and classification is performed using bag of audio words approach similar to the one used in the case of MFCC.

LSF -- Large-Scale Pooling Features

These features are useful for capturing sounds that have certain short-term temporal characteristics. To compute them, we extract a number of low-level descriptors, such as MFCC, PLP, LPC, Pitch, Loudness, Chroma, Formants, LSP, Signal Energy, Spectral Flux as well as their functionals, such as Means, Extremes, Moments, Peaks, Percentiles, Onsets, Zero-Crossing, etc.

In our implementation, a set of 6500 features is being extracted over 2 second windows in half second step. Feature Selection is performed using an Information Gain criterion followed by Principal Component Analysis and whitening to reduce dimensionality to 100. These features are used both in the K-Means framework and for trainining semantic ("noiseme") concept detectors, as used for audio segmentation.

SBPCA -- Sub-Band Autocorrelation Features

Feature extracted as described in [17].

DCNN— Deep Convolutional Neural Networks (DCNN) trained on ImageNet Challenge 2012

The Features are trained with ImageNet challenge data which contains 1.2 millions images and 1000 concepts. Given the trained models, we score each keyframe by how likely each concept exists and sum up scores for keyframes to get video level scores.

O*bject Bank -- Object Bank trained on ImageNet Challenge 2012*

We trained 1000 concepts based on ImageNet challenge 2012 dataset with bounding boxes. For each concept there are 400~1,300 positive samples, while for the negative samples we chose randomly N images from other concepts (about 540,000 images), where N is up to 4 times the number of positive samples and its maximum is 2000. For example, if there 600 positive samples, we will select randomly 2000 negative samples. We use Deformable Part Model code [8] to train our concepts with default parameter settings without part training.

1.2.2 New Representation

Besides using spatial bag-of-words and GMM from last year, this year we also used Fisher Vectors as described in our SED reports to represent low-level visual features.

1.2.3 Ensemble Weighted Double Fusion

This year, we still used double fusion [5, 6], however, instead of simple average fusion, we learned weights for late fusion. Given the early fusion results and single features outputs, we applied different learning strategies to fuse the outputs. Table 2 shows the results of different fusion strategies, most of them were first developed and evaluated for the 2013 MED submission. We have 10 different fusion weight learning strategies. The first six are different versions of regression and classification methods; the seventh considers the correlation among features by doing ranking correlation analysis. Features that are highly correlated with other features will be assigned lower weights. The eighth one uses the single feature performance to rank the features. Features with high single feature performance will be given high weights. The ninth one combines the two rankings together. The last one uses leave-one-out performance to rank the features, which gives very stable performance compared to other methods. Our baseline (average fusion) is given in the eleventh row. In the results, in which we applied the fusion methods to Pre-specified EK100 and Ad-Hoc EK10, we can see that when we have 100 training examples, almost all methods are better than average fusion except the weighted correlation methods. However, if we only have 10 training examples, regression and SVM classifiers cannot learn weights very well and tend to perform worse than simple average fusion. If we combined the 10 weighted fusion outputs and average fusion outputs together again using average fusion, we get consistently better results, which is shown in the twelfth row. This result is consistent with what we found last year in our 'double fusion' work, which showed that combining multiple fusion approaches yields better performance. In our submission, we use the combination of all these different fusion methods.

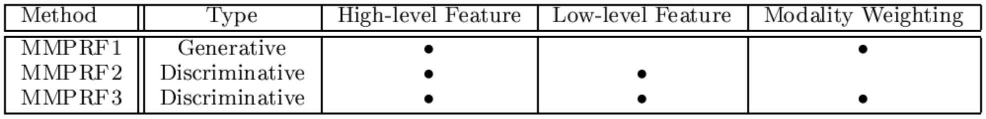
Table 2, Comparing performances of different weight learning strategies

ID	Feature	MAP on Pre-specified EK100 (MEDTEST)	MAP on Ad-Hoc EK10 (Internal Test)
1	L2 Logistic Regression	0.3885	0.2569
2	L1 Logistic Regression	0.3811	0.2482
3	L2 norm SVM	0.3699	0.2154
4	L1 norm SVM	0.3722	0.2141
5	Linear Regression	0.3895	0.2422
6	L2 Linear Regression	0.3893	0.2439
7	Weighted (W.) Correlation (C.)	0.3524	0.2829
8	W. Single Feature (F.) MAP	0.3788	0.2653
9	W. Corr. + Single Feat. MAP	0.3798	0.2659
10	W. Leave-one-out	0.3856	0.2804
11	Average Fusion	0.3621	0.2581
12	Combined	0.3911	0.2869

1.2.4 MultiModal Pseudo Relevance Feedback

In this year's submission, we used a novel method called MultiModal Pseudo Relevance Feedback (MMPRF). It is the most important component in the EK0 scenario which doubles the MAP of our baseline for both pre-specified and Ad-Hoc events. Besides, it also boosts by an absolute 3% our EK10 full system for Ad-Hoc events on our development dataset.
We propose three variants of MMPRF. MMPRF1 is based on the relevance model and the basic idea is to issue a query using the most relevant words found in the top ranked videos and feed retrieved rank list back to the previous result in order to improve the performance. MMPRF2 and MMPRF3 search the pseudo label set that maximizes the likelihood of all modalities. Then a joint model is trained on the pseudo label set using high-level features as well as low-level features. MMPRF2 and MMPRF3 differs in the maximum likelihood estimation where MMPRF2 treats each modality equally whereas MMPRF3 weights each modality with respect to a prior or the query likelihood. Table 3 presents the characteristics of three variants of MMPRF. As we see, both MMPRF2 and MMPRF3 are able to leverage low-level features and MMPRF3 further incorporates the modality weighting. We used MMPRF3 in our final submission. Due to the lack of space, we cannot present detailed algorithms in this report.

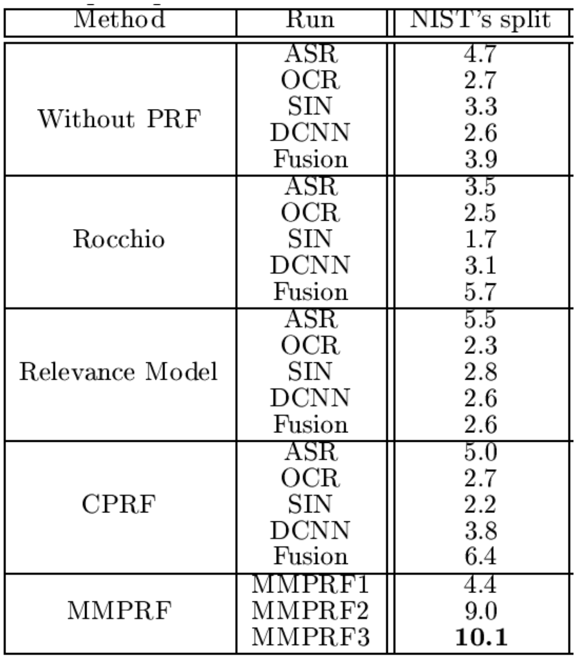
Table 3, Summary of characteristics of MMPRF



Method	Type	High-level Feature	Low-level Feature	Modality Weighting
MMPRF1	Generative	●		●
MMPRF2	Discriminative	●	●	
MMPRF3	Discriminative	●	●	●

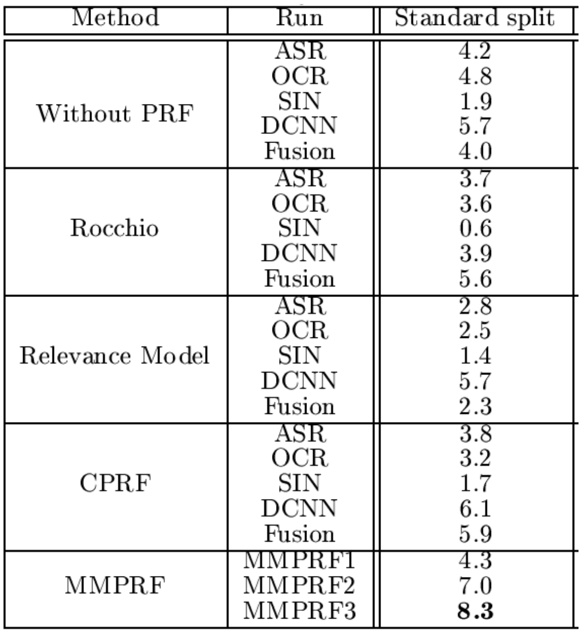
We compare MMPRF methods with four baseline methods. The first baseline is the plain retrieval result without Pseudo Relevance Feedback (PRF). The second baseline is classical Rocchio PRF, where the vector space model with TF-IDF weighting is used. The third one is relevance model and the forth one is Classification-based PRF (CPRF) [16]. For Pre-specified events we used the training/test split provided by NIST and for Ad-Hoc events we used our internal split. The results are summarized in Table 3 and Table 4, where the best result is highlighted. As we see, both MMPRF2 and MMPRF3 significantly outperform the basic retrieval method without PRF. In addition, MMPRF2 and MMPRF3 are also significantly better than the baseline methods on both datasets. The event-level comparison of the baseline methods can be found in Figure 2 and Figure 3.

Table 4, MAP (in Percentage) comparison with baseline methods on Pre-specified tasks



Method	Run	NIST's split
Without PRF	ASR	4.7
	OCR	2.7
	SIN	3.3
	DCNN	2.6
	Fusion	3.9
Rocchio	ASR	3.5
	OCR	2.5
	SIN	1.7
	DCNN	3.1
	Fusion	5.7
Relevance Model	ASR	5.5
	OCR	2.3
	SIN	2.8
	DCNN	2.6
	Fusion	2.6
CPRF	ASR	5.0
	OCR	2.7
	SIN	2.2
	DCNN	3.8
	Fusion	6.4
MMPRF	MMPRF1	4.4
	MMPRF2	9.0
	MMPRF3	**10.1**

Table 5, MAP (in Percentage) comparison with baseline methods Ad-hoc tasks



Method	Run	Standard split
Without PRF	ASR	4.2
	OCR	4.8
	SIN	1.9
	DCNN	5.7
	Fusion	4.0
Rocchio	ASR	3.7
	OCR	3.6
	SIN	0.6
	DCNN	3.9
	Fusion	5.6
Relevance Model	ASR	2.8
	OCR	2.5
	SIN	1.4
	DCNN	5.7
	Fusion	2.3
CPRF	ASR	3.8
	OCR	3.2
	SIN	1.7
	DCNN	6.1
	Fusion	5.9
MMPRF	MMPRF1	4.3
	MMPRF2	7.0
	MMPRF3	**8.3**

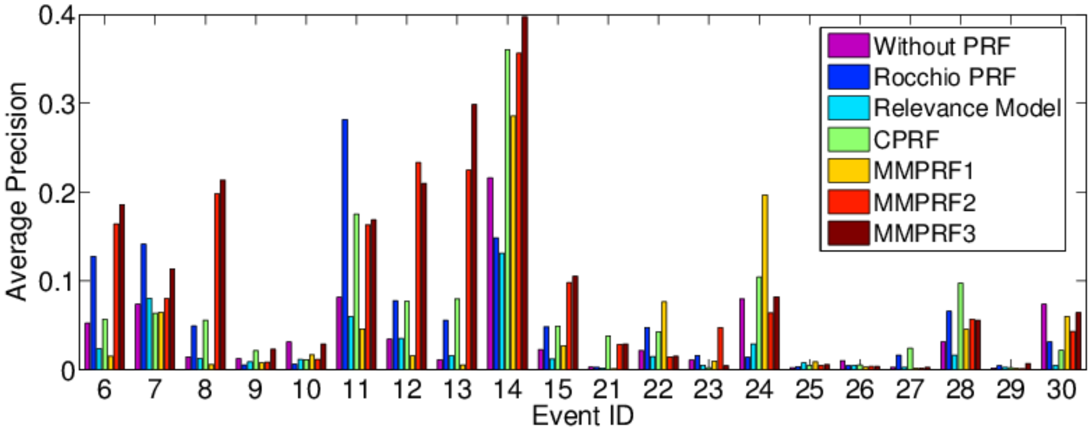


Figure 2, The event-level AP comparison on Pre-specified events

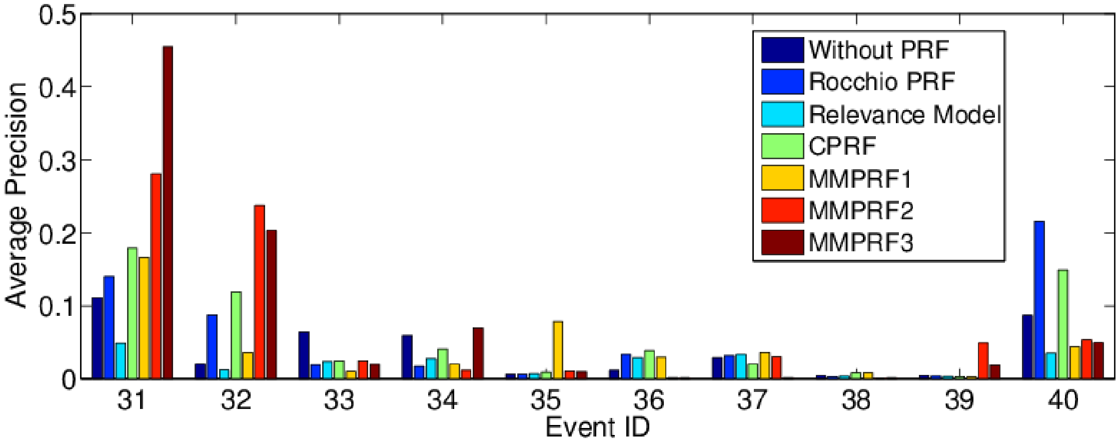


Figure 3, The Event-level AP comparison on Ad-Hoc events

1.2.5 Threshold Learning

We have studied the evaluation metric Minimum Acceptable Recall, R0. The definition of R0 is: $R_0(q) = Recall(T_q) – 12.5$ x $rank(T_q)/V$, where $Recall(T_q)$ and are $rank(T_q)$ the recall and rank at the threshold T for event q, and V is the total number of videos in the search set. According to the definition, the threshold plays important role in getting a higher R0 score. Basically, we expect higher recall when the rank is lower. Intuitively, the threshold actually reflects the point which separates the positive testing data and the negative testing data. In theory, the prediction scores for positive testing data and those for negative testing data should have two different distributions. One possible way is to find the threshold that differentiates the two distributions. For this purpose, we referred to the popular approach in image segmentation which is using maximum entropy theory. With maximum entropy theory, we assume that the prediction scores which are larger than the selected threshold should result in the maximum entropy among all the data points. We adopted the simplest and most common approach that uses histogram-based estimation in which the entropy probability density is represented as a histogram. Specifically, the histogram approach uses the idea that the differential entropy,

$$H(X) = -\int f(x) \log f(x) dx$$

can be approximated by producing a histogram of the observations, and then finding the discrete entropy

$$H(X) = -\sum_{i=1}^{n} f(x_i)\log\left(\frac{f(x_i)}{w(x_i)}\right)$$

of that histogram (which is itself a maximum-likelihood estimate of the discretized frequency distribution), where w is the width of the i-th bin. Table 6 shows our R0 results for MEDTEST

Table 6, R0 for MEDTEST using maximum entropy scheme

	EK10	EK100
FullSys	0.2910	0.5005
ASRSys	0.0469	0.1301
AudioSys	0.1139	0.1900
OCRSys	0.0897	0.0047
VisualSys	0.2465	0.4475

1.3 Contribution of New Components

In this section, we took Pre-specified EK100 and Ad-Hoc EK10 to examine the individual feature's performance and the contributions of each new feature, representation and method by ablation studies. MEDTEST is used to evaluate the Pre-specified task. For the Ad-Hoc evaluation, we used an internal set in which we randomly split the data in half; one half as training data and the other as testing data. These evaluations were performed after the submission results were announced to help explain what worked.

Figure 1 shows the single feature performance in MAPs. From Figure 4, we can see that DCNN and Trajectory are the best two individual features in Pre-specified EK100 task and DCNN perform significantly better than other features in Ad-Hoc EK10 task.

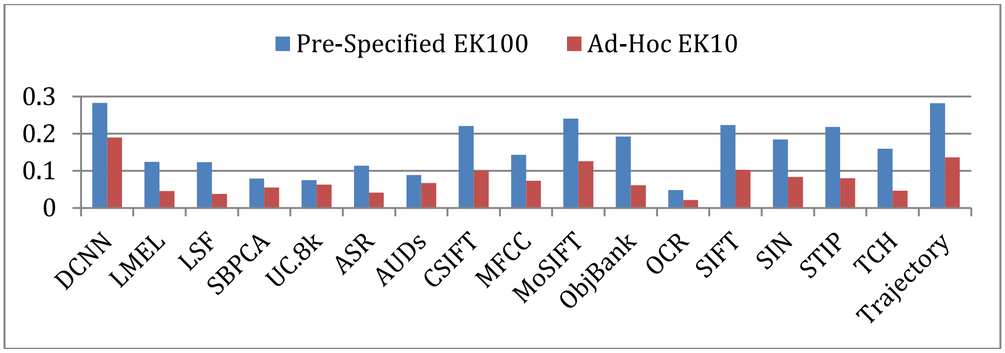


Figure 4, Single feature MAPs for Pre-specified EK100 and Ad-Hoc EK10

Table 7 lists the MAPs of baseline (with all strategies) and the MAPs of leaving out each new strategy. From Table 5, we can see that semantic features (DCNN and Object Bank) contribute significantly, especially in Ad-Hoc task. It would be interesting to see how the performance changes with the number and the accuracy of the concepts. New low-level features such as LSF and SBPCA do not have significant impact. Fisher Vectors do not help much. Weight learning also has about 4 percent better MAP in Pre-specified EK100 and 3 percent improvement in Ad-Hoc EK10. We also estimate the upper-bound of the fusion by learning weights on our internal test data, which are 0.4035 and 0.2930 for Pre-specified EK100 and Ad-Hoc EK10 respectively, which shows that our weight learning strategy is quite close to the upper-bound when we take the distribution difference between training and testing data into consideration.

Table 7, Ablation studies of new strategies, MAP is used for evaluation

	Pre-Specified EK100	Ad-Hoc EK10
Baseline	0.3839	0.2709
-Object Bank	0.3817	0.2690
-DCNN	0.3651	0.2229
-Fisher Vectors	0.3795	0.2641
-Weight Learning	0.3437	0.2475
-LSF	0.3829	0.2705
-SBPCA	0.3848	0.2712

1.4 MED’13 Submission

In this section, we describe the methods used and the time required to generate the submissions. The methods used for generating the Pre-specified and Ad-Hoc submissions are the same. However due to time constraints some methods were not used during the Pre-specified submission and only used in the Ad-Hoc submission. More details will be given in the following paragraphs.

Full System
Weighted double fusion is used for the full system. Early fusion fuses feature vectors of SIFT, Color SIFT, TCH, Motion SIFT, STIP, Trajectories, MFCC, Object Bank, SIN, DCNN and OCR. For the Pre-specified run, DCNN was not added to early fusion due to time constraints. While fusing results from individual features and early fusion, we performed weighted late fusion. Weights were learnt from output generated during cross-validation. Weights were not learnt for Pre-specified EK10 run due to time constraints.
For the Ad-Hoc EK10 runs, we used MMPRF3 for two iterations. In the first iteration, we take the 10 highest ranked videos in the predicted testing set and add them to the training set. In the second iteration, we take the top 30 ranked videos and add them into the testing set. For Pre-specified EK10, we did not run PRF due to time constraints.

Visual System
Early fusion fuses SIFT, Color SIFT, TCH, Motion SIFT, STIP, Trajectories, Object Bank, SIN and DCNN. For the Pre-specified run, DCNN was not added to early fusion due to time constraints. Weighted late fusion was also performed. Weights were not learnt for Pre-specified EK10 run due to time constraints.

Audio System
For both EK100 and EK10, we use all the audio features except ASR. Given the outputs of SVM and KRR classifiers, we first perform z-score normalization and then average late fusion.

ASR System
We use the same late fusion methods as the audio system to fuse the outputs of SVM and KRR classifiers.

OCR System
Same method was adopted as the ASR system.

In Figure 5 and 6, we compare our full system runs with other teams in both Pre-specified and Ad-Hoc tasks using the MAP criterion. From the Figure 5, we can see in the Pre-specified task, we are third in EK100 and fourth in EK10 and EK0. As seen in Figure 6, in the Ad-Hoc task, we are the second and almost as good as the first in EK100 and significantly outperform other teams in EK10 and EK0.

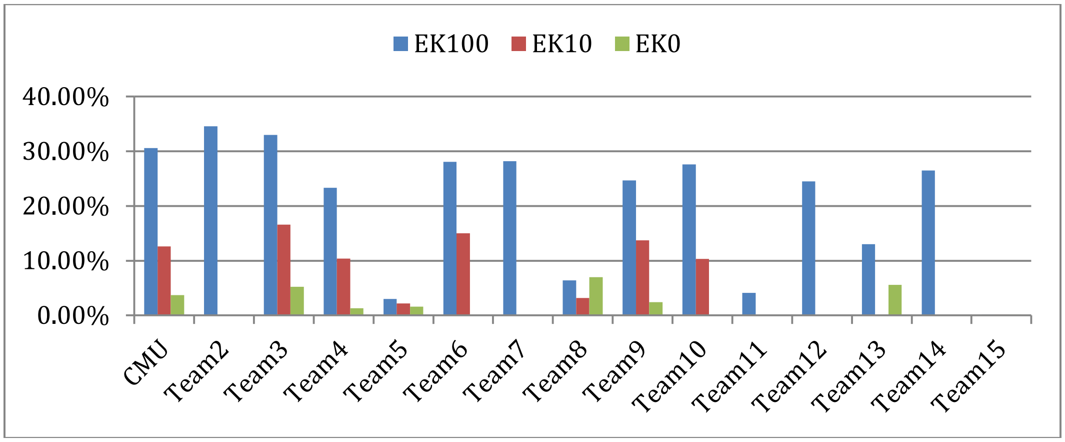


Figure 5, CMU MED Pre-specified full system performance compared to other teams.

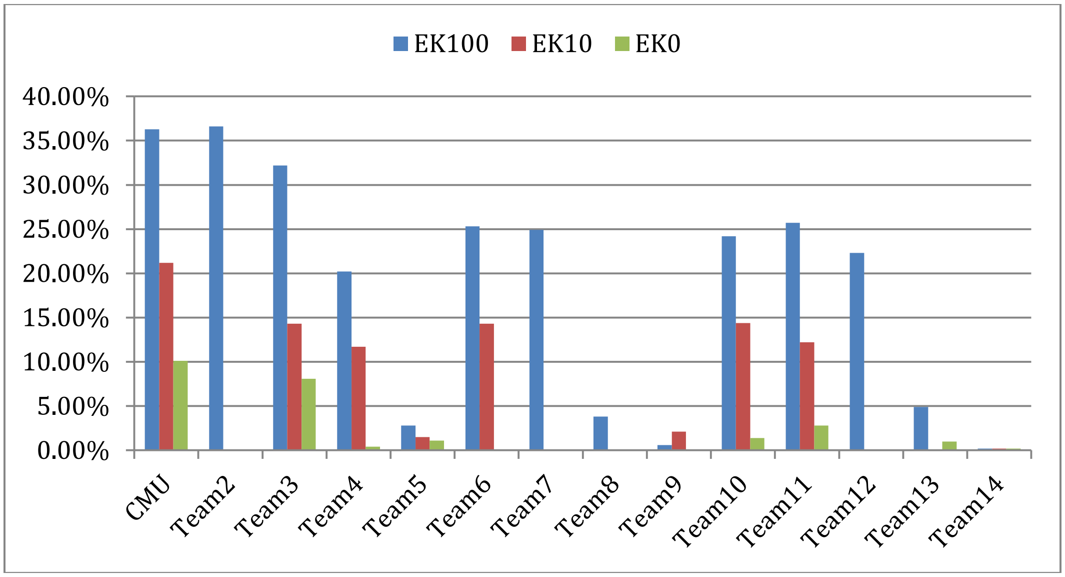


Figure 6, CMU MED Ad-Hoc full system performance compared to other teams.

Table 8 shows the timing information for the submission. Since we used on average 300 cores for the submission, the numbers are in the unit of "300 core hours". For example, for the EK100 full run, if we had 300 cores, it will take 2018.2 hours to complete all the required feature extraction, event agent training and prediction to generate the EK100 full run submission.

Table 8 Compute time information for MED’13 submission

Runs		Feature Extraction			Event Agent		Total
		Event Examples	Event Background	PROGTEST	Generation	Execution	
EK100	Full	1.9	92.6	1817.4	23.5	82.7	2018.2
	Visual	1.7	82.5	1619.4	16	56.2	1775.9
	Audio	0	1.3	25.9	5.7	19.9	52.8
	ASR	0.1	4.4	86	0.9	3.3	94.8
	OCR	0.1	4.4	86	0.9	3.3	94.8
EK10	Full	0.2	92.6	1817.4	5.2	35.4	1950.9
	Visual	0.2	82.5	1619.4	3.5	24.1	1729.7
	Audio	0	1.3	25.9	1.3	8.5	37.0
	ASR	0	4.4	86	0.2	1.4	92.1
	OCR	0	4.4	86	0.2	1.4	92.1
EK0	Full	0	0	1051.5	0	8	1059.5
	Visual	0	0	868	0	0.5	868.5
	Audio	0	0	11.5	0	0	11.5
	ASR	0	0	86	0	0	86.1
	OCR	0	0	86	0	0	86.1

2. MER System

The E-Lamp MER system uses a similar approach to last year’s submission, adapted to the new interfaces. In this new interface, no relationships were computed, but only observations were output. Different models were trained for the EK100 and EK10 conditions. For EK0, simple mappings between event kits and text or concept names were used instead of mappings observed in video exemplars.

2.1 Features

We included the following aspects in our MER submission:

- Event-Relevant and Video-Distinctive Visual Concepts
- Event-Relevant Keyframe Image Concepts
- ASR Transcripts
- Optical Character Recognition Output (Transcripts)
- Audio Concepts (Noisemes)

2.2 Visual and Audio Concepts

We use the histogram of each video semantic class aggregated over the whole video clip. To use the visual concepts, we first generated a bipartite graph matching of concepts with the MED events. The process flow is shown in Figure 7.

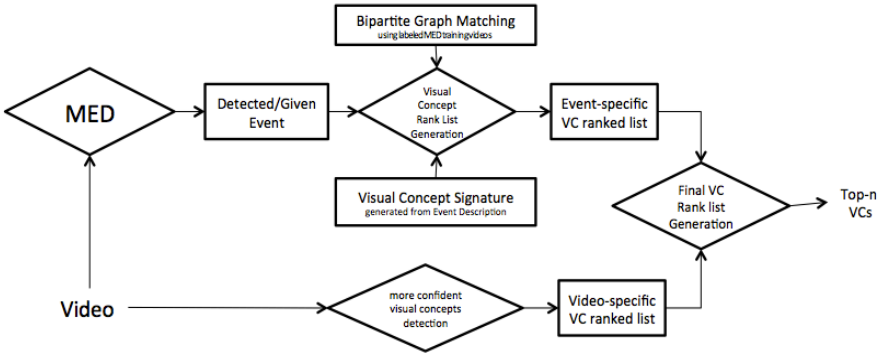

Bipartite Graph Matching
using labeled MED training videos
MED
Detected/Given Event
Visual Concept Rank List Generation
Event-specific VC ranked list
Visual Concept Signature
generated from Event Description
Final VC Rank list Generation
Top-n VCs
Video
more confident visual concepts detection
Video-specific VC ranked list

Figure 7: Flow chart of visual and audio concepts processing

2.3 ASR and OCR Transcripts

Automatic speech recognition and OCR transcripts that indicate "linguistic (audio)" information in the video. (e.g. "okay", "hello", "she didn't" etc.). We use TF-IDF according to the word-level ASR confidence to calculate the relevant of each ASR word result to the event kit. We then rank the ASR Transcripts according to their relevance to the event. For OCR, no confidences were available, so they were all set to an equal value.

2.4 Integration

The information was integrated in a way similar to last year's submission. Approximations to confidence (when available from the features) and importance (as given by TF-IDF or bipartite graph matching) were computed, and used to rank multiple candidates. Cut-offs (for max number to display, confidence, and importance) were used to restrict the maximum number of entries to display to suitable values.

3. Acknowledgments

This work has been supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. The U.S. government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

[1] Campbell, W., & Sturim, D. (2006). Support vector machines using GMM supervectors for speaker verification. *IEEE Signal Processing Letters* .

[2] Chaudhuri, S., Harvilla, M., & Raj, B. (2011). Unsupervised Learning of Acoustic Unit Descriptors for Audio Content Representation and Classification. *Interspeech.*

[3] Chen, M., & Hauptmann, A. (2009). MoSIFT: Reocgnizing Human Actions in Surveillance Videos. Carnegie Mellon University. Carnegie Mellon University.

[4] Krizhevsky, Alex, Ilya Sutskever, and Geoff Hinton. "Imagenet classification with deep convolutional neural networks." Advances in Neural Information Processing Systems 25. 2012.

[5] Lan, Z., Bao, L., Yu, S.-I., Liu, W., & Hauptmann, A. G. (2012). Double Fusion for Multimedia Event Detection. MMM.

[6] Lan, Z. Z., Bao, L., Yu, S. I., Liu, W., & Hauptmann, A. G. (2013). Multimedia classification and event detection using double fusion. Multimedia Tools and Applications, 1-15.

[7] Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. CVPR.

[8] Li, L.-J., Su, H., Xing, E., & Fei-Fei, L. (2010). Object Bank: A High-Level Image Representation for Scene Classification and Semantic Feature Sparsification. NIPS.

[9] Over, P., Awad, G., Michel, M., Fiscus, J., Sanders, G., Shaw, B., et al. (2012). TRECVID 2012 -- An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics. Proceedings of TRECVID 2012.

[10] Sande, K. E., Gevers, T., & Snoek, C. G. (2010). Evaluating color descriptors for object and scene recognition. TPAMI .

[11] Yu, S. I., Xu, Z., Ding, D., Sze, W., Vicente, F., Lan, Z., ... & Hauptmann, A. (2012). Informedia e-lamp@ TRECVID2012: Multimedia event detection and recounting med and mer. In NIST TRECVID Workshop.

[12] Tong, W., Yang, Y., Jiang, L., Yu, S. I., Lan, Z., Ma, Z., ... & Hauptmann, A. G. (2013). E-LAMP: integration of innovative ideas for multimedia event detection.Machine Vision and Applications, 1-11.

[13] Viitaniemil, V., & Laaksonen, J. (2009). Spatial Extensions to Bag of Visual Words. CIVR.

[14] Wang, H., Klaser, A., Schmid, C., & Liu, C.-L. (2011). Action recognition by dense trajectories. CVPR.

[15] Wang, H., Ullah, M. M., Klaser, A., Laptev, I., & Schmid, C. (2009). Evaluation of local spatio-temporal features for action recognition. BMVC.

[16] Yan, Rong, Alexander G. Hauptmann, and Rong Jin. "Negative pseudo-relevance feedback in content-based video retrieval." Proceedings of the eleventh ACM international conference on Multimedia. ACM, 2003.

[17] http://labrosa.ee.columbia.edu/projects/calcSBPCA/

CMU Informedia @TREVID 2013
Semantic Indexing (SIN)

Lu Jiang, Shicheng Xu, Zheng-Zhong Lan, Waito Sze, Alexander Hauptmann
Carnegie Mellon University

1. Introduction

For this year's submission we used the following features:

- SIFT harrislaplace
- SIFT densesampling
- Color SIFT harrislaplace
- Color SIFT densesampling
- Motion SIFT (MoSIFT)
- Deep Convolutional Neural Networks (DCNN).
- Metadata

For the DCNN feature we trained 1000 concepts detectors on ImageNet dataset [1] and use the detectors output as features for in the SIN training. The metadata of a video includes its title, uploader and description information extracted from XML file. Compared with 2012's submission, this year we used the SIN 2013's full label set. The cascade SVM classifiers on Hadoop are adopted as our classifier to accelerate the SIN extraction [4].

2. Submitted Runs

We submitted 4 runs for the main task.

- CMU_Maggie: Our Safe run using all features except DCNN and Metadata.
- CMU_Bart: This run adds DCNN features to CMU_Maggie.
- CMU_Homer: This run is based on CMU_Bart with two modifications. First junk key-frames are detected and removed. Second, the confidence score of correlated concepts are propagated using the last year's algorithm[5].
- CMU_Lisa: This run is based on CMU_Homer and it further fuses the score of uploader model [2] using the collective classification presented in [3].

We submitted 3 runs for the concept pair task. Our general idea is as follows: training individual concept detectors and then enhancing the prediction of pair concept using the related concept detectors.

- CMU_ Todd_and_Rod: This run is the required baseline run where we average the scores of two detectors.
- CMU_Sherri_and_Terri: This run employs the average fusion for the related concepts.
- CMU_Itchy_and_Scratchy: This run fuses the score of related concepts based on their accuracy in the development set.

3 . Experimental Results

In this section we summarize our results. Table 1 shows our results of the main task. The results suggest the DCNN feature is helpful in boosting the performance, though not significantly. Junk keyframes removal and concept propagation used in CMU_Homer manage to improve AP by 1%. The uploader model with collaborative classification fusion improves the performance by another 0.8%.

Table 1. Our final results of the main task.

RUN NAME	INF AP
CMU_Maggie	0.2293
CMU_Bart	0.2353
CMU_Homer	0.2452
CMU_Lisa	0.2537

Table 2 shows our results for the concept pair task. As we see, the related concept enhanement fails to improve the performance of our baseline.

Table 2. The final results of our pair concept detection run

RUN NAME	INF AP
CMU_ Todd_and_Rod	0.1421
CMU_Sherri_and_Terri	0.1161
CMU_Itchy_and_Scratchy	0.1117

4. Acknowledgments

This work was partially supported by Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11PC20068. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government. The work was supported in part by the U. S. Army Research Office (W911NF-13-1-0277). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ARO.

References

[1] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." CVPR 2009

[2] Niaz, Usman, and Bernard Merialdo. "Improving video concept detection using uploader model." Multimedia and Expo (ICME),

[3] Lu Jiang, Alexander Hauptmann, Guang Xiang. Leveraging High-level and Low-level Features for Multimedia Event Detection. In ACM Multimedia, pp. 449-458. Nara, Japan

[4] Cascade SVM on Hadoop: https://code.google.com/p/cascadesvm/

[5] Braunstein, A.; Mézard, R.; Zecchina, R. (2005). "Survey propagation: An algorithm for satisfiability". Random Structures & Algorithms 27 (2): 201–226.

CMU@TRECVID 2013:
Surveillance Event Detection

Chenqiang Gao, Yang Cai, Haoquan Shen, Wei Tong,
Yi Yang, Nicolas Ballas, Deyu Meng, Yan Yan, Alex Hauptmann
Carnegie Mellon University

1 Introduction

We present a generic event detection system and a key-pose-based interactive event detection system evaluated in the Surveillance Event Detection (SED) task of TRECVID 2013. The generic event detection system is designed for six events: “Embrace”, “Objectput”, “PeopleMeet”, “PeopleSplitUp”, “PersonRuns” and “Pointing”. This system consists of two parts: the retrospective part and the interactive part and its difference to our previous version [1] is that we add the STIP feature [2]. For the event “CellToEar”, we design a new key-pose-based interactive system based on a good division of labor between people and computers.

2 Generic Event Detection System

2.1 Fisher Vector Encoding for Retrospective Event Detection

We used MoSIFT and STIP feature. A GMM model with 128 Gaussians is learned to model the distribution of the our features. Sliding window detection is performed. Then each sliding window is represented as a Fisher Vector. Models are learnt using Linear SVM for each of six events: “Embrace”, “Objectput”, “PeopleMeet”, “PeopleSplitUp”, “PersonRuns” and “Pointing”. The final decision is thresholded by the MinDCR value at the training set. Finally, average late fusion is used to combine the two features results.

2.1.1 Fisher Vector Encoding

Fisher Vector Encoding utilizes a Gaussian mixture model (GMM) $U_\lambda(x) = \sum_{k=1}^{K} \pi_k u_k(x)$ trained on local features of a large image set using Maximum Likelihood (ML) estimation. The parameters of the trained GMM are denoted as $\lambda = \{\pi_k, \mu_k, \Sigma_k, k = 1, \cdots, K\}$, where $\{\pi, \mu, \Sigma\}$ are the prior probability, mean vector and diagonal covariance matrix of Gaussian mixture respectively.

2.1.2 Multiscale detection and Non-maximum suppression

Ideally, we need to search over different scales and different step size to locate the exact event in the video sequences. However, it is impractical for our current sliding window framework. For example, the maximum length of PersonRuns event in the Dev dataset is 1000 frames while the minimal length is 10 frames – such diversity of event duration makes the computation cost too high for us can afford. Instead of using exhaustive search, we select three scales which are closest to the average duration of each event and select the scale with highest score.

2.2 Interactive Event Detection

We attempted to address two central problems of an interactive surveillance event detection system: (1) detection results visualization and (2) user feedback utilization. Because of the limited time available for interaction, the system design was driven by efficiency considerations from both these two perspectives. Please refer to [1] for details.

3 Key-Pose-Based Interactive Event Detection system

3.1 Motivation

Among all seven events in SED, the “CellToEar” event is the hardest to detect. This year, we have designed a specific method, namely key-pose-based interactive method, for this event.

Fig. 1 shows sequential images of a representative “CellToEar” instances. According to the TRECVid 2009 Event Annotation rules, the frame 1 is the start time (“When the subject starts to move the phone to his/her head”) and the frame 6 is the end time (“When the phone reaches the head”), and the frame after frame 7 is not considered as the “CellToEar” event. From Fig.1, it can be

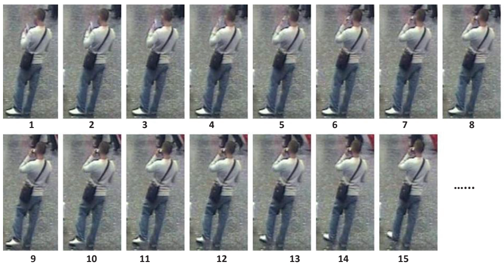


Figure 1: An instance of the "CellToear" event.

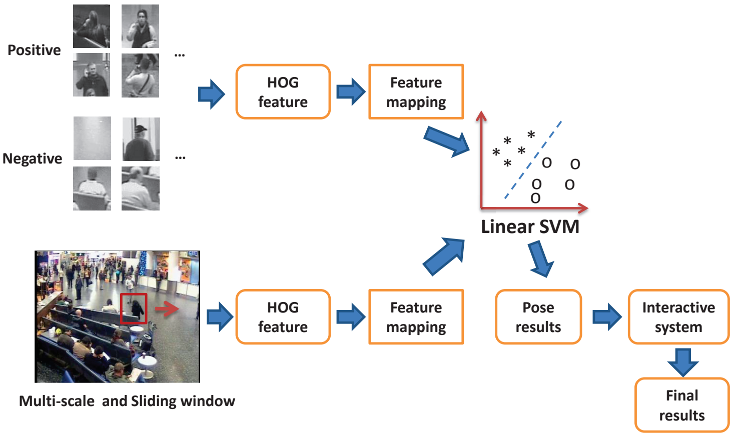


Figure 2: Key pose based interactive event detection system architecture.

easily observed that following the "CellToEar" event, there is usually a stable state when a person talks on the phone (We call this state as "a phoning stage" throughout this report).

The duration period of a "CellToEar" event is very short. Thus in the complex scene with heavy clutter, it is generally hard to capture this motion information by using the traditional descriptors, for example, STIP [4]. During the phoning stage, a person always keeps a stable pose of holding a cell phone on his/her head, which can be easily observed in Fig. 1(We call this pose as "a phoning pose" throughout this report).

Based on the above observation, we convert the "CellToEar" event detection to the problem of detecting phoning poses. Compared to the conventional strategies, our mechanism makes the problem easier. In addition, since the phoning pose can be located in a 2D image patch, we can readily adopt off-the-shelf multi-scale and sliding window techniques to search these poses in images with complex background. After we get the phoning pose locations, an user can then look back in time to find the occurrence of a "CellToEar" event using the interactive system.

3.2 System Framework

As demonstrated in Fig. **??**, our "CellToEar" interactive detection system includes two main components: (1) automatic key pose detection and (2) user interactive search.

3.2.1 Automatic Key Pose Detection

In order to automatically detect phoning poses in videos, we train a specific model for each scene since four scenes are very different (We did not process the videos from camera 4 because there are almost no "CellToEar" event). The main steps are: 1) a linear SVM model; 2) the multi-scale and sliding window technology using for detecting phoning poses in images.

In the first step, we manually annotated around 290 positives and randomly sample 22,155 negatives from each scene in development data. In order to avoid large variation of poses, we just annotated the upper body of a person with the phoning pose. In this way, most of positive samples are similar even though they are from different scenes. For example, the poses from both sitting persons and standing persons are similar to each other, which can be easily seen in Fig.**??**. Another benefit of the positive samples generated like this is that we can use all positives (totally 1,448 positives) from all 4 scenes while only utilize the negative ones from one scene of four to train a specific model for a specific scene. The model so trained inclines to be more stable against the scene.

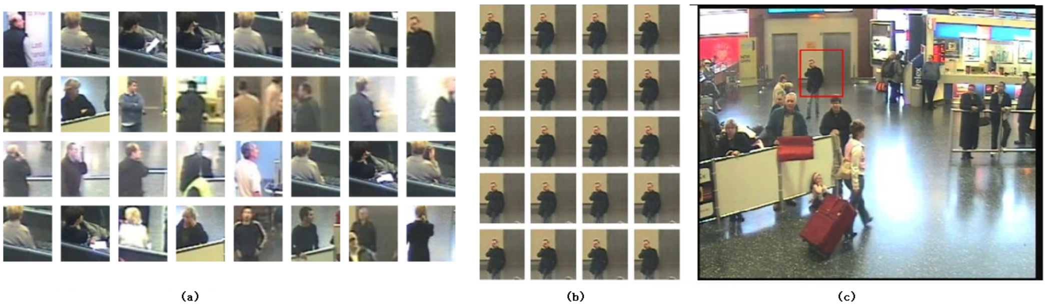


Figure 3: Three visualization parts in user interactive interface for “CellToEar”. (a) Sorted pose detection results, (b) the traced sub-images for a fixing local area, (c) the full image with a rough bounding box annotation.

The histogram of gradient (HOG) feature is employed to describe the phoning pose image for two reasons. The first one is that a HOG descriptor is suitable for describing 2D non-rigid objects such as human. The second one is that the HOG feature can be computed efficiently in sliding window detection. Before training the SVM model, we applied the explicit feature kernel maps [3] to the HOG features. This helps enhance the computational efficiency in both training and testing.

In the second step, we divide the scene into several portions and in each one different scale ranges are adopted according to the annotation data information. Furthermore, we applied the motion mask acquired from background substraction to further reduce computation. This can also help reduce the background clutter.

Since the phoning poses usually last a relatively long time, the used frame stride is set as 12 for saving computation time. After we get preliminary detection results, we sort all the obtained results from the same scene based on SVM classification scores. Finally, we aggregate all sorted results from different scenes to get the final results, as shown in Fig. 3(a), according to the proportion information of scenes which is estimated by using development data.

3.2.2 User Interactive Search

Our user interactive interface contains three main visualization windows: 1) The first one shows the final sorted detection results, as shown in Fig.3(a); 2) The second one shows the sub-images traced in a fixing local area, as shown in Fig.3(b); 3) The third shows the full image of the scene, as shown in Fig.3(c). Firstly the user needs to quickly find the phoning pose in the first window and click it by using a mouse. After that, the interface system will automatically update the corresponding contents of the second and third windows. For example, if the user clicks the last pose in the first row of Fig. 3(a), the corresponding contents will be shown in Fig.3. Then by using the keyboard, the user can trace back the video to find time interval in which people move a cellphone to the ear or the face. Actually many “CellToEar” events may happen outside of the scene, and thus we can stop tracing if the person with phoning pose leaves the scene.

4 Experimental Results

Except for the “CellToEar” event, we follow the pipeline and the experimental setting of last year [1].

4.1 Evaluation of Retrospective Event Detection

We show our primary run results using the MoSIFT and STIP features (*CMU13_FV*) on retrospective task in Table 1 compared with the results of CMU MoSIFT feature of the last year, CMU Bag-of-Words of 2012 (*CMU11_BoW*) and the other teams’ best primary run results of 2012 (*Others12_Best*). Here, both *CMU13_FV* and *CMU11_BoW* use Fisher Vector encoding. Please note that the test video of 2012 is a subset of the last year’s. It is shown that our *CMU13_FV* is better than *CMU12_FV* and *CMU11_BoW*. Except for the “CellToEar” event, the actual DCRs of the other six events are less then 1.0.

4.2 Evaluation of Interactive Event Detection

Table 2 shows the actual DCR comparisons of 2013 retrospective result and interactive event detection results of 2012 and 2013. It can be seen from this table that the performance of five of six events are better than the last year for the interactive event detection since we add a new feature and adopt the late fusion strategy this year.

Table 1: The actual DCR and minimum DCR comparisons of primary runs among *CMU13_FV* *CMU12_FV*, *Others12_Best* and *CMU11_BoW*.

	CMU13_FV		*CMU12_FV*		*Others12_Best*		*CMU11_BoW*	
	ActDCR	MinDCR	ActDCR	MinDCR	ActDCR	MinDCR	ActDCR	MinDCR
CellToEar	**1.0000**	1.0000	1.0007	1.0003	1.0040	0.9814	1.0365	1.0003
Embrace	0.8357	0.8338	**0.8000**	0.7794	0.8247	0.8240	0.8840	0.8658
ObjectPut	**0.9981**	0.9975	1.0040	0.9994	0.9983	0.9983	1.0171	1.0003
PeopleMeet	**0.9474**	0.9450	1.0361	0.9490	0.9799	0.9777	1.0100	0.9724
PeopleSplitUp	0.8947	0.8879	**0.8433**	0.7882	0.9843	0.9787	1.0217	1.0003
PersonRuns	**0.7708**	0.7646	0.8346	0.7872	0.9702	0.9623	0.8924	0.8370
Pointing	0.9959	0.9892	1.0175	0.9921	**0.9813**	0.9770	1.5186	1.0001

Table 2: The actual DCR comparisons of the 2013 retrospective result, and the 2012 and 2013 interactive event detection results.

	CMU13_inter	*CMU12_inter*	*CMU13_retro*
CellToEar	**0.9057**	1.0090	1.0000
Embrace	**0.6540**	0.6696	0.8357
ObjectPut	**0.9889**	1.0064	0.9981
PeopleMeet	**0.8813**	0.9786	0.9474
PeopleSplitUp	0.8549	**0.8177**	0.8947
PersonRuns	**0.5850**	0.6445	0.7708
Pointing	**0.9851**	0.9854	0.9959

For the “CellToEar” event, the actual DCR of this year improved to 0.9057 from 1.009 last year. As shown in Fig.3, a user can quickly localize the person who is on phone by using our specific interactive interface. Focusing on the local location, the user can trace back frames of the video to find the occurrence duration of a “CellToEar” event by using keyboard operations. Since our system has the function of playing video frames with different speeds, this searching process is very fast. However, we have to spend much time in tracing persons until they disappear from the scene, because many “CellToEar” events occur outside the scene. This can be alleviated by using some tracking techniques in the future. Furthermore, currently we just use the HOG feature and future work will use different low-level features and classifiers further to improve the accuracy of the phoning pose detection.

5 Acknowledgments

This material is based in part upon work supported by the National Science Foundation under Grant No. IIS-1251187.

References

[1] Y. Cai, Q. Chen, L. Brown, A. Datta, Q. Fan, R. Feris, S. Yan, A. Hauptmann, and S. Pankanti. Cmu-ibm-nus@ trecvid 2012: Surveillance event detection. 2012.

[2] I. Laptev. On space-time interest points. *International Journal of Computer Vision*, 64(2-3):107–123, 2005/09/01 2005.

[3] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(3):480–492, 2012.

[4] T. YingLi, C. Liangliang, L. Zicheng, and Z. Zhengyou. Hierarchical filtered motion for action recognition in crowded videos. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(3):313–323, 2012.