SRI-Sarnoff AURORA System at TRECVID 2012
Multimedia Event Detection and Recounting

Hui Cheng[†], Jingen Liu[†], Saad Ali[†], Omar Javed[†], Qian Yu[†], Amir Tamrakar[†], Ajay Divakaran[†], Harpreet S. Sawhney[†], R. Manmatha[⁕], James Allan[⁕], Alex Hauptmann[♦], Mubarak Shah[♠], Subhabrata Bhattacharya[♠], Afshin Dehghan[♠], Gerald Friedland[✧], Benjamin Martinez Elizalde[✧], Trevor Darrell[✧], Michael Witbrock[♡], Jon Curtis[♡]

[†] SRI-International Sarnoff, Vision Technologies Lab, 201 Washington Road, Princeton NJ 08540
[⁕] University of Massachusetts-Amherst
[♦] Carnegie Mellon University,
[♠] University of Central Florida,
[✧] International Computer Science Institute – Berkeley,
[♡] Cycorp Inc.

Abstract

In this paper, we describe the evaluation results for TRECVID 2012 Multimedia Event Detection (MED) and Multimedia Event Recounting (MER) tasks as a part of SRI-Sarnoff AURORA system that is developed under the IARPA ALDDIN program. In AURORA system, we incorporated various low-level features that capture color, appearance, motion, and audio information in videos. Based on these low-level features, we developed Fixed-Pattern and Object-Orientated spatial feature pooling, which result in significant performance improvement to our system. In addition, we collected more than 1800 concepts and designed a set of concept pooling approaches to build the Concept Based Event Representation (CBER, i.e., high-level features). We submitted six runs exploring various fusions of low-level features, high-level features, and ASR/OCR features for MED task. All runs achieve satisfactory results. In particular, two EK10Ex runs for both pre-specified events (PS-Events) and ad-hoc events (AH-Events) obtain relatively better results. In MER task, we developed an approach to provide a breakdown of the evidences of why the MED decision has been made by exploring the SVM-based event detector. Furthermore, we designed evidence specific verification and detection to reduce uncertainty and improve key evidence discovery. Our MER evaluation results for MER-to-Event are very good.

1 Introduction

Retrieving complex events from a huge number of open source videos is very challenging due to the characteristics of events and videos “in the wild”. For example, even a very specific event usually covers a great diversity of contents involving various objects, atomic human actions, scenes, and audio information. On the other hand, open source videos are unconstrained which are typically recorded under uncontrolled conditions with large variations in camera motion, illumination, object appearance and scale, as well as viewpoint. Therefore, to capture different aspects of an event, we developed various low-level features and concept features (high-level features) in the AURORA system, and explored different fusion strategies to inquire discriminative information from all aspects of an event. Moreover, the MER component enables the AURORA system to disclose the evidences of MED decision made by the system.

2 Multimedia Event Detection

The AURORA system incorporates two types of features: low-level features and high-level features. Low-level features are designed to acquire the first-hand characteristics of an event, such as the involved object appearance, color and motion information, and scene structure. These low-level features are quantized into visual-words, which is used to model an event as a Bag of Visual Words (BOW). We treat this BOW as an average feature pooling over the whole frame. However, a specific event typically has its own Region of Interests that produce most informative evidence of this event. Hence, we propose a new strategy for spatial pooling of the low-level features, which result in an event model capturing spatial information. However, the low-level feature based event model training usually needs a large number of training examples for better model generalization. This is because of the diversity of visual/audio contents. To achieve better model generalization with less training examples, we developed over 1,800 visual concepts, from which we derived various concept features. The concept features also enable us to do better MER. We describe all the details in the following sections.

2.1 Low-Level Visual Features

We developed a variety of low-level features to capture various aspects of an event, such as scene, object, action, and so on. There features are extracted either from sample frames (static features), or spatio-temporal windows of frames (i.e., XYT-volumes, dynamic features) of a video.

2.1.1 Static Features

Static features are computed from sampled frames (i.e., one sample every second). They are assumed to provide object or scene appearance information of an event. Following static features are extracted:

A. GIST [1]: This feature was proposed to capture the structure/shape of real world scenes. Basically it is a holistic statistical signature of an image, representing the scene with Spatial Envelope consisting of a set of perceptual dimensions (e.g., naturalness, openness, roughness, expansion, and ruggedness). It is a fast approach to coarsely capture the event scene structure. We quantize the GIST feature of each frame, and represent the video as a bag of quantized gist features.

B. SIFT [2]: SIFT feature is a widely used feature descriptor for image matching and classification. The 128 dimensional SIFT descriptor is rotation invariant, which captures the local texture structure of an image. We extracted two types of SIFT features: sparse SIFT (S-SIFT) and dense SIFT (D-SIFT). S-SIFT is computed around an interest point detected by corner detector, and D-SIFT is computed for dense sampled image patches. The former one is used to describe informative patches of an object, while the latter is good to capture local patch distribution over a scene.

C. **colorSIFT** [3] : This feature is an extension of SIFT. Instead of computing SIFT based on intensity gradient, colorSIFT detects interest points and create descriptors on color gradients. It actually contains 3 128 dimensional vector with first one from intensity gradient and the other two from color gradient. As a result, it is able to capture both intensity and color information.

D. Transformed Color Histogram [4]: It is a normalized color histogram as describe in [4].

2.1.2 Dynamic Motion Features

Dynamic features are computed from detected XYT-volumes of a video. These XYT-volumes are sampled by detecting spatio-temporal interesting points or 2D corner point trajectories. They are supposed to capture the motion information of a video. But with the design of various descriptors, they are able to capture the appearance information too. The following dynamic features have been extracted.

A. STIP [5]: The Space-Time Interest Points (STIP) detects 3D interest points in the spatio-temporal domain, which is the extension of 2D Harris corner detector. It assumes the detected points have the most intensive motions in a video. STIP generate a descriptor on the intensity gradient of frames (HOG) and on the optical flow space (HOF). The final descriptor encodes both HOG and HOF feature description.

B. Dense Trajectory Feature (DTF) [6]: Rather than detecting interest point in XYT space, DTF detects 2D corner points and tracks them in a short time period. The 2D corners are usually associated with objects in a video. By analyzing the velocity or shape of trajectories, we are able to select trajectories with strong enough motions to represent the characteristics of a video. The corners are tracked by KLT tracking. From these trajectories, various features/descriptors can be extracted, such as shape, velocity. The AURORA adopts two types of descriptors: HOG (histogram of orientated gradient) and MBH (Motion Boundary Histogram). HOG captures the static appearance information along the trajectory, while MBH captures the motion information along the trajectory.

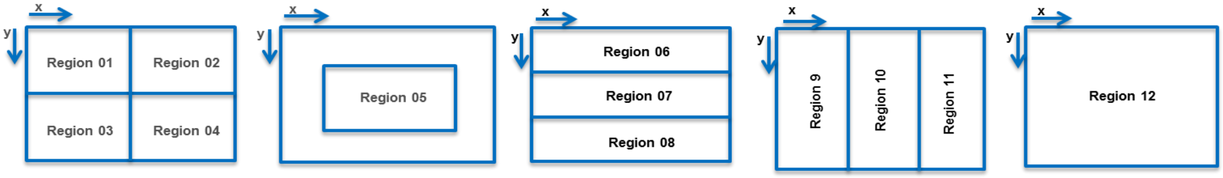


Figure 1. Fixed-Pattern based spatial feature pooling.

C. MoSIFT [11]: Motion SIFT (MoSIFT) extends the 2D SIFT descriptor to the temporal dimension. Unlike SIFT, it combines both local appearance and motion information to detect interest points. The motion information is obtained by computing optical flow.

2.1.3 Low-Level Feature Based Event Representation

The above set of features is computed either on single frames or on spatio-temporal windows of frames (XYT-cubes) throughout a given video clip. The event represented by a video clip is represented as an aggregate feature as the histogram of “words” corresponding to each feature type computed over the entire video clip. This is popularly known as a “Bag-of-Words” (BoW) representation. In order to compute BoW descriptors for each feature type, feature specific vocabularies are first learned using k-means clustering of raw features. Other than GIST feature having 1000 words, the other features such as SIFT, DTF, STIP have a vocabulary of of 10000 words. Once the features in a video are quantized using the respective vocabularies, a BoW is computed per feature. Event models are trained using SVM [10] with intersection kernel.

2.1.4 Spatial Feature Pooling

Bag of Features (BoF) based visual representation has been successfully applied to our AURORA system and produced good results for MED 11. This success is due to the fact that the statistics information of the bag of local features in terms of histogram of visual-words captures the major cues of events to some extent. However, one obvious disadvantage of BoF is that it ignores the spatial or temporal distribution of the features, which might be discriminative for some events. For example, the motion features usually concentrate on the central regions of a video for “sewing project”, while “flash a mop” produces motion from the entire frame.

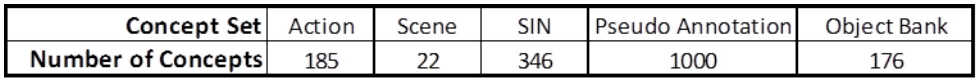
Fixed-Pattern Based Feature Pooling: The basic idea is similar to Spatial Pyramid Match, which constructs a pyramid structure for two images, and the matches happening to the fine level will contribute more to the final match score. Instead of having a strict pyramid structure, we pre-define 12 Region of Interests (ROI) including the full frame to pool features as shown in Fig. 1. Since we adopt the average pooling, each ROI is represented by an *m*-dimensional vector where *m* is the vocabulary size (e.g., 10,000 in our system for most features). We can concatenate the *m*-dimensional vectors of all ROIs to build an event model. However, this results in a high dimensional vector with N*m bins (where N is the number of ROIs, e.g., 12), which makes the training infeasible. Instead, we treat each ROI individually as an information provider, and train a detector for each of them. The video level decision is made by aggregating the evidence provided by the detectors of all ROIs.

2.2 High-Level Visual Features

One of the challenges for event recognition is to bridge the semantic gap between low-level features and high-level events. There are a variety of reasons to represent events in terms of semantic concept features. Concepts are directly connected to the Event Kit Descriptions. Thanks to the semantic meaning of concepts, the concept-based event representation (CBER) [7] potentially has better generalization capability, which is significantly important for event recognition, especially when only a few training examples are available. In addition, CBER enables the system to integrate multi-modality information such as human knowledge and Internet resources for event detection. What is more, it offers a natural schema for multimedia event recounting.

2.2.1 Concept Detectors

In MED12, we collected two types of concepts: data-relevant concepts and data-irrelevant concepts. Data-relevant concepts, which are annotated from the event kit, are directly related to the pre-specified events. To contrast, the data-irrelevant concepts are provided by a third party, such as ImageNet scenes and objects, and TRECVID SIN concepts. They may contain both event related and unrelated concepts. Table 1 lists the number of concepts in each of our concept set.



Concept Set	Action	Scene	SIN	Pseudo Annotation	Object Bank
Number of Concepts	185	22	346	1000	176

Table 1: Concept Sets Used in MED12

Action Concepts: Actions are typically atomic and localized motion and appearance patterns, which are strongly associated with some specific event. Our action concepts cover general actions such as “person walking”, “person running”, “person climbing”, as well as event specific actions such as “standing on top of bike”, and “running next to a dog”. We employ well-established techniques to build our concept detectors. In particular dynamic features (i.e., STIP [5] and Dense Trajectory Based features [6]), and the bag-of-word representations [12] defined over codebooks of these features are used to represent action concepts. Binary SVM classifiers with Histogram Intersection kernel are used for concept classification.

TRECVID SIN concept detectors are provided by CMU, which are trained on images. Pseudo Annotation is detailed in subsection 2.2.3.

2.2.2 Concept Based Event Modeling

Given a video x, a concept detector φ_i can return a confidence value c_i. In practice, however, it is not wise to feed a long length video into a detector and get a single detection confidence for the entire video, because concept detectors are trained on single frames or short video segments. In this section, we discuss how to derive various semantic features from videos over the semantic space.

Our method uses the atomic concept detectors as filters that are applied to a given XYT segment of a video clip to capture the similarity of content to the given concept. So as a first step towards representing a video clip with concepts, each concept detector is applied to each XYT window in a video to obtain an K*W matrix C of scores, where $C_{ij} \propto p(c_i | w_j)$. Each C_{ij} is the detection confidence of concept i applied to window j.

Given the raw detection scores of concepts over the full video, the event depicted in the clip can be represented using a number of features derived from C_{ij}. One option is to select the maximum detection score C_i^{max} over all sliding windows as the detection confidence of concept detector φ_i. As a result, we are able to obtain a K-dimensional vector C^{max} to represent a video. Meanwhile, we have embedded a video into the concept space defined above. What is more, based on the K-dimensional semantic space, we also explore the following four representations:

MAX pooling: for each concept detector, only the maximum detecting score over all sliding windows is pooled to show the probability of concept given a video.

Max-Avg-Std (MAS): Other than the maximum detecting score, we believe other information of the concept distribution over a video, such as average and standard deviation, is also discriminative for an event. Hence, for each concept detector, the maximum, average, and standard deviation values over all sliding windows are selected to form MAS feature.

Bag of Concepts (BOC): Akin to the bag of words descriptors used for visual word like features, a bag of concepts features measures the frequency of occurrence of each concept over the whole video clip. To compute this histogram feature, the SVM output is binarized to represent the presence or absence of each concept in each window.

Co-occurrence Matrix (CoMat): A histogram of pairwise co-occurrences can be used to represent the pairwise presence of concepts.

Max Outer Product (MaxCoMat): Since concepts represent semantic content in a video, the max value of each concept across the whole video represents the confidence in the presence of a concept in a video. The outer product of the vector of max values of each of the concepts represents both the strength of the presence of each concept (diagonal values) as well as the strength of co-occurrence of pairwise concepts (off-diagonal values):

Short of capturing the temporal structure of an event, the above event representations derived from concept detectors capture the first and second order distributional content in a video. In the event recognition phase, a classifier (e.g., SVM classifier) can be trained on each type of event representation, the recognition scores are fused to make the final decision.

2.2.3 Pseudo Annotation Concept for Event Modeling

Concept detectors are classifiers and often fire on multiple concepts or objects in the image. For example, a dog detector may respond to many other things besides dogs. Instead of finding concepts one may want to look at the distribution of concepts in an image or video. This distribution may be a better description of an image or video since the distribution may be consistent for videos of the same event while the individual concepts may be poorly detected. We propose a technique called pseudo-annotations for event detection. Pseudo-annotations are detected as follow. First, a multi-class SVM detector is trained for a 1000 arbitrary concepts using concepts and their images from ImageNet. The concepts are then detected in a set of subsampled frames in each video. Instead of detecting all 1000 concepts in a frame we only detect the top k (k is usually 10 or 20 and is estimated using training). These top k binary concepts are called pseudo-annotations. Each video is described by a histogram of pseudo-annotations by pooling the pseudo-annotation histograms for the frames in a video. For each event a set of training videos is used to train a SVM model using the pseudo-annotation histograms. Test videos are classified as belonging to an event by running a SVM classifier using the corresponding model.

2.2.4 ASR/OCR Information

An information retrieval based approach is adopted to retrieve the videos based on OCR/ASR. The event kit is used to automatically construct the query. All fields in the event kit are used for ASR query while the audio field is dropped in the OCR query. A sequential dependence model is used for retrieval both OCR and ASR. The model takes both ordered and unordered phrases into account. Terms are weighted based on event kit fields. The weighting is set manually. In order to fuse OCR/ASR results with low-level and high level features, an expected-precision is computed. Since many videos do not have OCR/ASR data, a video-level fusion is carried out; where a low OCR/ASR retrieval score does not affect the feature based retrieval score, while a very high OCR/ASR retrieval score significantly increases the final score.

3 Multimedia Event Recounting

An event is typically a complex activity occurring at a specific place and time. On the other hand, a video may contain a lot of other irrelevant information as well. Thus, for each recognized event occurrence in a video clip, the goal of recounting is to describes the spatial and temporal details of the occurrence. The recounting includes key observations regarding the scene, people, objects, and activities pertaining to the event occurrence. Such recounting provides user a semantic description that is useful to perform further analyses. As the concept features that we are using by definition contain semantic information, it has the advantages for recounting purpose compared with low-level features.

As our event classification is based on Support Vector Machines (SVMs), we present an approach to perform the recounting in the context of SVMs. Given the feature vector $x \in R^n$ where n is the feature dimension, the SVM decision function $h(x)$ can be represented as follows,

$$h(x) = \sum_{l=1}^{m} \alpha_l K(x, x_l) + b \ (1),$$

where x_l is one of *m* support vectors, ie. $l = 1, \dots, m$. $K(x, x_l)$ is the kernel value between x and x_l. α_l is the signed weight of x_l and b is the bias. If the kernel functions have the following form, $K(x, z) = \sum_{i=1}^{n} f(x_i, z_i)$, where f is the function and x_i and z_i are the ith feature value of x and z. For example, intersection kernel satisfies such a form where $f_{INT} = \min(x, z)$. Linear kernel also follows this form. Now the decision function can be rewritten as follows,

$$h(x) = \sum_{i=1}^{n} \sum_{l=1}^{m} \alpha_l f(x_i, z_i^l) + b,$$

where $z_i^l$is the ith feature value of lth support vector. Supporse $h_i(x) = \sum_{l=1}^{m} \alpha_l f(x_i, z_i^l)$, we can decompose the decision value of $h(x)$ as

$$h(x) = \sum_{i=1}^{n} h_i(x) + b,$$

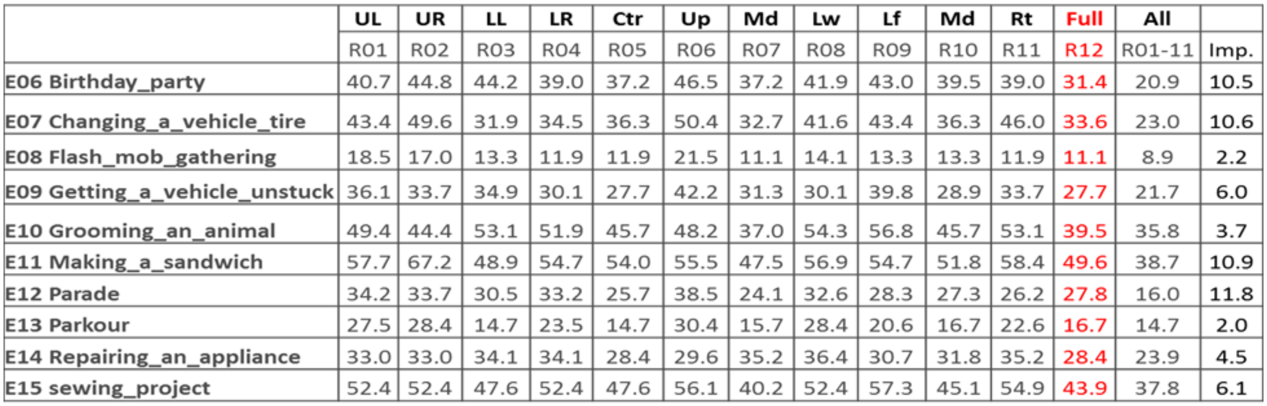
where $h_i(x)$ encodes how encodes how much ith feature contributes towards the final decision value. For our event recounting application, as each feature has semantic information, we are able to retrieve the important evidences by sorting $h_i(x)$. We have shown our recounting approach in the context of SVMs. In fact, the approach can be applied to any additive classifiers as in Eq 1, which cover a wide spectrum of classification approaches.

4 Experiments

4.1 Training/Testing Methodology

We adopt the Support Vector Machine (SVM) as our basic classifiers and use intersection kernel for all histogram-based features and RBF (Radial Basis Function) kernel for concept-based features. Other SVM parameters are default values. We apply L1 normalization to histogram-based features. Event videos are used as positive samples and all non-event videos are used as negative samples to train a binary classifier for each event independently. Each classifier outputs a probability of detection as a score. LibSVM [10] is used as the SVM solver.

Standard Training/Testing Evaluation Folds: For MED11 PS-Events, we use EC+DEVT as training data, and test on DEVO dataset. As for the MED PS-Events and AH-Events, we separate the Event Collection (EC) and DEVT data into a set of standard evaluation folds (e.g., three folds) to avoid the difference caused by dataset. All of our experiments, from exploring good features to fusing features from different modalities, are based on the standard evaluation folds.



	UL	UR	LL	LR	Ctr	Up	Md	Lw	Lf	Md	Rt	Full	All	
	R01	R02	R03	R04	R05	R06	R07	R08	R09	R10	R11	R12	R01-11	Imp.
E06 Birthday_party	40.7	44.8	44.2	39.0	37.2	46.5	37.2	41.9	43.0	39.5	39.0	31.4	20.9	10.5
E07 Changing_a_vehicle_tire	43.4	49.6	31.9	34.5	36.3	50.4	32.7	41.6	43.4	36.3	46.0	33.6	23.0	10.6
E08 Flash_mob_gathering	18.5	17.0	13.3	11.9	11.9	21.5	11.1	14.1	13.3	13.3	11.9	11.1	8.9	2.2
E09 Getting_a_vehicle_unstuck	36.1	33.7	34.9	30.1	27.7	42.2	31.3	30.1	39.8	28.9	33.7	27.7	21.7	6.0
E10 Grooming_an_animal	49.4	44.4	53.1	51.9	45.7	48.2	37.0	54.3	56.8	45.7	53.1	39.5	35.8	3.7
E11 Making_a_sandwich	57.7	67.2	48.9	54.7	54.0	55.5	47.5	56.9	54.7	51.8	58.4	49.6	38.7	10.9
E12 Parade	34.2	33.7	30.5	33.2	25.7	38.5	24.1	32.6	28.3	27.3	26.2	27.8	16.0	11.8
E13 Parkour	27.5	28.4	14.7	23.5	14.7	30.4	15.7	28.4	20.6	16.7	22.6	16.7	14.7	2.0
E14 Repairing_an_appliance	33.0	33.0	34.1	34.1	28.4	29.6	35.2	36.4	30.7	31.8	35.2	28.4	23.9	4.5
E15 sewing_project	52.4	52.4	47.6	52.4	47.6	56.1	40.2	52.4	57.3	45.1	54.9	43.9	37.8	6.1

Figure 2. Miss Detection of different ROI based event detectors at 6% False Alarm. This evaluation use EC+DEVT for training, and test on dEVO data.

4.2 Feature-Based Experiments

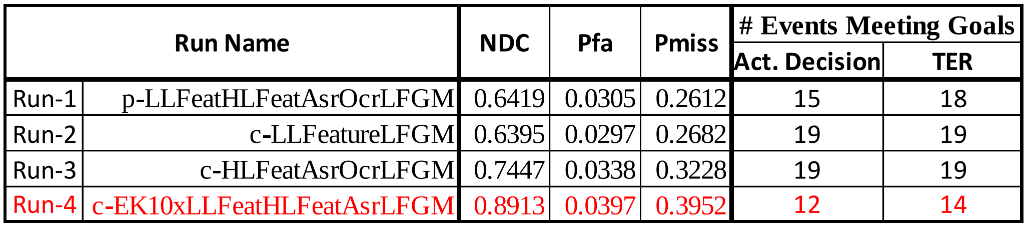
Experiments were performed to explore the advantages of fixed-pattern feature pooling over various static and dynamic features. We exactly follow the MED11 evaluation process using EC and DEVT for training and testing on DEVO. Figure 2 shows the performance (Miss Detection at 6% False Alarm) of each ROI based event detectors over 10 MED PS-Events. Only DTF-HOG low-level features are used in this evaluation. As we can see, although “Full” region is generally perform better, other regions may achieve better MD for some event, e.g., the “Ctr” performs better than “Full” for event “Parade”, and “Md” better than “Full” for event “Parkour”. This observation demonstrates the complementarity of the ROIs. What is more, the fusion of R01-11 consistently works better than “Full” ROI (e.g., vary from 2% to 11%). We repeat this evaluation for all other features, and obtain similar conclusion. However, when we fuse all low-level feature together, the pooling only improves 1.1% to 6.2% (average about 2%). We conjecture the fusion of low-level features leave a very small space for pooling to improve the overall performance.

4.3 MED12 Results and Discussion

All the computations reported in this notebook were performed on the SRI-Sarnoff AURORA system. This system comprises of a number of servers with web interfaces for managing experiments run over a distributed computational pipeline, annotating training data and just browsing the datasets. The computational pipeline currently consists of about 350 AMD Opteron nodes with 5GB RAM per node as well as a number of nVidia Tesla M2050 GPUs and is based on the Apache UIMA (Unstructured Information Management Architecture) which is essentially a highly configurable filter graph like architecture that allows for process distribution across multiple nodes.

In MED12, we have two training modes: EKFull mode trained with all available positive examples and EK10Ex mode trained with 10 positive examples. Both training modes do not have constraint on the negative training examples. There are 25 events to evaluate including 20 pre-specified events (PS-Events) and 5 ad-hoc events (AH-Events).

For PS-Events, we submitted one primary run and three contrastive runs as follows:



Run Name		NDC	Pfa	Pmiss	# Events Meeting Goals	
					Act. Decision	TER
Run-1	p-LLFeatHLFeatAsrOcrLFGM	0.6419	0.0305	0.2612	15	18
Run-2	c-LLFeatureLFGM	0.6395	0.0297	0.2682	19	19
Run-3	c-HLFeatAsrOcrLFGM	0.7447	0.0338	0.3228	19	19
Run-4	c-EK10xLLFeatHLFeatAsrLFGM	0.8913	0.0397	0.3952	12	14

Figure 3. MED 12 evaluation results for PS-Events. Pfa is false alarm rate, and Pmiss is the miss detection rate. The last two columns list the number of events make the year goal of ALADDIN. Run-1 to Run-3 are EKFull runs, while Run-4 is an EK10Ex run, which uses 10 positive training examples. This year goal for PS-Events is 50%MD@4%FA.

- **Run-1 *p-LLFeatHLFeatAsr OcrLFGM*** (EKFull mode): this is the primary run which combines all low-level features, high-level features, and ASR/OCR features with Geometric Mean (GM) fusion.
- **Run-2 *c-LLFeatureLFGM*** (EKFull mode): it is a pure low-level features based run with GM fusion.
- **Run-3 c-*HLFeatAsrOcrLFGM*** (EKFull mode): it is a pure semantic features based run with GM fusion.
- **Run-4 *c-EK10xLLFeatHLFeatAsrLFGM*** (EK10Ex mode): this run is similar to the primary run but under different training modes.

For AH-Events, we have two runs: one primary run trained with EKFull mode and one EK10Ex run:

- **Run-5 *p-LLFeatHL FeatAsrOcrLFGM*** (EKFull mode): it is the correspondence run of Run-1 for AH-Events.
- **Run-6 c-EK10xLLFeatHLFeatOcrAsrLFGM** (EK10Ex mode): this is the corresponding run of Run-4 for AH-Events.

Run Name		NDC	Pfa	Pmiss	# Events Meeting Goals	
					Act. Decision	TER
Run-5	**p-LLFeatHLFeatAsrOcrLFGM_1**	0.6411	0.0274	0.2992	5	5
Run-6	**c-EK10xLLFeatHLFeatOcrAsrLFGM**	1.0097	0.0432	0.4704	5	5

Figure 3. MED 12 evaluation results for PS-Events. Pfa is false alarm rate, and Pmiss is the miss detection rate. The last two columns list the number of events make the year goal of ALADDIN. Run-5 is an EKFull run, while Run-6 is an EK10Ex run, which only uses 10 positive training examples. This year goal for AH-Events is 75%MD@%6FA.

	By Events					Overall	By Data Set	
	E22	E26	E27	E28	E30		Eval_data	Prog_Test
MER-to-Event	100.00%	96.30%	100.00%	100.00%	96.30%	98.52%	98.33%	98.89%
MER-to-Clip	72.22%	37.04%	37.04%	29.63%	50.00%	45.19%	40.56%	54.44%
Combination	83.33%	60.74%	62.22%	57.78%	68.52%	66.52%	63.67%	72.22%

Figure 4. Multimedia Event Recouting (MER 12) evaluation results.

According to the evaluation, all runs achieve satisfactory results. In particular, both EK10Ex runs (i.e., Run-4 and Run-5) obtain very good results. We conjecture that this is due to a large number of concepts used in our system making our learned event model having better generalization capability when less training examples are used.

4.4 MER12 Results

In MER12, we submitted results for MER-to-Event and MER-to-clip on both Evaluation Set and Progress Set. Our MER system obtains 98.52% accuracy for MER-to-Event task. More details for MER shown in Figure 4. We did very well at "MER-to-Event" task. This is due to our MER system is built on the Concept Based Event Representation [7]. Since we do not have specific concepts for "MER-to-Clip" task, it did not perform very well. Please note that our approach works almost equally well on both Eval_data (evaluation dataset) and Prog_Test (progress test data). Therefore, our MER system can be generalized to various data sets.

References

1. A. Torralba and A. Oliva. Modeling the shape of the scene: a holistic representation of the spatial envelope. IJCV, 2001.
2. D. Lowe, Distinctive image features from scale invariant key-points. IJCV, pp. 91-110. 2004
3. K. E. Sande, T. Gevers, C. G. Snoek, Evaluating color descriptors for object and scene recognition. TPAMI, 2010.
4. G.J. Burghouts and J.M. Geusebroek, Performance Evaluation of Local Color Invariants, CVIU, vol. 113, pp. 48-62, 2009.
5. I. Laptev and T. Lindeberg. Space-time interest points. ICCV, pages 432 – 439, 2003.
6. H. Wang, A. Klser, C. Schmid, and C. L. Liu. Action recognition by dense trajectories. CVPR, 2011.
7. J. Liu, Y. Qian, et al., Video event recognition using concept attributes, WACV, 2013.
8. Y. Qian, J. Liu, et al., Multimedia event recounting with concept base representation, ACM MM 2012.
9. A. Tamrakar, S. Ali, Q. Yu, J. Liu, O. Javed, A. Divakaran, H. Cheng, and H. Sawhney, Evaluation of low-level features and their combinations for complex event detection in open source videos, CVPR 2012.
10. C.-C Chang and C.-J. Lin LIBSVM : a library for support vector machines. ACM T-IST, pp. 1-27.2011
11. M. Chen and A. Hauptmann MoSIFT: Reocgnizing Human Actions in Surveillance Videos. CMU-CS-09-161, 2009.
12. J. Liu, J. Luo, and M. Shah, Recognizing realistic human actions from videos "in the wild". CVPR 2009.