SRI-Sarnoff AURORA at TRECVID 2014
Multimedia Event Detection and Recounting

Hui Cheng[†], Jingen Liu[†], Ishani Chakraborty[†], Guang Chen[†], Qiguang Liu[†],
Mohamed Elhoseiny[†], Gary Gan[†], Ajay Divakaran[†], Harpreet S. Sawhney[†],
James Allan[⁕], John Foley[⁕], Mubarak Shah[♠], Afshin Dehghan[♠], Michael Witbrock[♡], Jon Curtis[♡]

[†] SRI-International Sarnoff, Vision Technologies Lab, 201 Washington Road, Princeton NJ 08540
[⁕] University of Massachusetts-Amherst,
[♠] University of Central Florida,
[♡] Cycorp Inc

Abstract

In Multimedia Event Detection 2014 evaluation [20], SRI Aurora team participated in task 000Ex, 010Ex and 100Ex with full system evaluation. Aurora system extracts multi-modality features including motion features, static image feature, and audio features from videos, and represents a video with Bag-of-Word (BOW) and Fisher Vector model. In addition, various high-level concept features have been explored. Other than the action concept features and SIN features, deep learning based semantic features including both DeCaf and Overfeat implementation have been explored. The deep-learning features achieve good performance for MED, but they are not the right features for MER. In particular, we performed further study on semi-supervised Automatic Annotation to expand our action concepts. To distinguish event categories efficiently and effectively, we introduce Linear SVM into our system, as well as the feature-mapping technique to approximate the Histogram Intersection Kernel for BOW video model. All the modalities are fused by an ensemble of classifiers including techniques such as Logistic Regression, SVR, Boosting, and so on. Eventually, we achieve satisfied achieved satisfactory 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.

1 Introduction

The task of Multimedia Event Detection (MED) aims at detecting complex events, such as “dog show”, “wedding ceremony”, “parkour” and so on from open source videos. It is very challenging due to the characteristics of events and videos. The event videos usually cover a great diversity of visual contents including various objects, atomic human actions, physical scene, and audio information. Furthermore, open source videos may own various quality issues such as low-resolution, camera motion, occlusion, and so on. Therefore, representing a video with multiple modalities enhances its discriminative capability of the detection. In addition to the low-level features used in MED13 evaluation, such as DenseTrack, DenseSIFT, HessianAffine, Color SIFT, TCH, and Audio features, two types of deep learning features have been developed. The deep learning features are similar to ObjectBank or Pseudo-Annotation features which were used in MED13 [1]. Although each individual deep feature (corresponding to one detector) does not perform reasonably good, the combination produces very discriminative feature vectors. They are kind of features staying between high-level concept features and low-level features. More details are in Section 2. In Section 3, we focus on describing the action concepts, audio concepts, as well as ASR/OCR concepts. To automatically expand our action concept dictionary, we also developed an Automated Annotation system, which is introduced at the end of this section. As Kernel SVM has relatively low efficiency in speed, we introduced Linear SVM into the Aurora evaluation system. To properly feed BOW video representation into LinearSVM, BOW is mapped onto a higher dimensional space using feature-mapping technique [2]. In addition, a supplementary representation Fisher Vector is also introduced, which actually performs soft feature quantization rather than the hard-quantization in BOW. For more details, please refer to Section 4. MER approach is discussed in Section 5. Finally, Section 6 discusses the fusion strategy, followed by the experimental results on MED Test and Prog Test.

2 Visual and Audio Features

The visual features include static image features, motion features, and deep learning features. Although deep learning features seem possess more semantics than the regular low-level static feature and motion feature, they still behave like low-level features. It is because the individual deep feature (concept detection) is not reasonably meaningful.

2.1 Static Visual 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. SIFT [3]: 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 (HessianAffine) and dense SIFT (D-SIFT). HessianAffine 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.

B. colorSIFT [4] : 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.

C. Transformed Color Histogram [5]: It is a normalized color histogram as describe in [4].

2.2 Dynamic Motion Features

Dynamic features are computed from detected XYT-volumes of a video. These XYT-volumes are sampled by 2D corner point trajectories. They are supposed to capture the motion information of a video. In MED14, we only select dense trajectory feature [6].

Dense Trajectory Feature (DTF): 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.

2.3 Audio Features

A. MFCC Feature: The audio is PCM-formatted with a sample rate of 16kHz. The extracted acoustic features, using HTK[25], are the typical Mel-Frequency Cepstral Coefficients (MFCCs) C0-C19, with delta and double deltas, for a total of 60 dimensions. Each feature frame is computing using a 25 ms window, with 10 ms frame shifts. Short-time Gaussian feature warping using a three-second window is used, and temporal regions containing identical frames are removed.

B. CMU Audio Features: We also adopt another two types of AUD feature (UC, Bauds), as described in [13] and [14].

2.4 Deep Learning Features

Convolutional neural networks (CNN) have recently shown outstanding performance in large image classification tasks such as ImageNET. In [21], [22] and [23] it has shown that features extracted form an already trained network on a large image dataset can be used as a powerful representation for classification/detection on other dataset. In MED, two types of implementations are adopted by our system.

A. DeCaf Implementation: DeCaf is implemented in [24]. The network is trained using ILSVRC-2012 data. The network take a square image and pass it through five successive convolutional layers (C1,C2,…,C5) and three fully connected layers (FC6,FC7 and FC8), as shown in the following figure. We use the output of FC7 layer as the feature representation for each key-frame of a video which is a 4096 dimensional vector. The key-frames are sampled form a video every 2 seconds. The final representation for each video is obtained by taking the average score of each dimension across the key-frames of that video. Thus the dimension of the final representation for each video remains 4096. For classification we used Support Vector Machine with Histogram Intersection Kernel. All the features are L2 normalized.

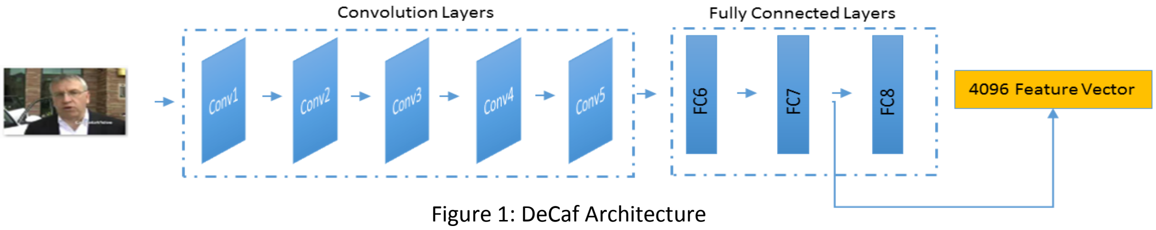
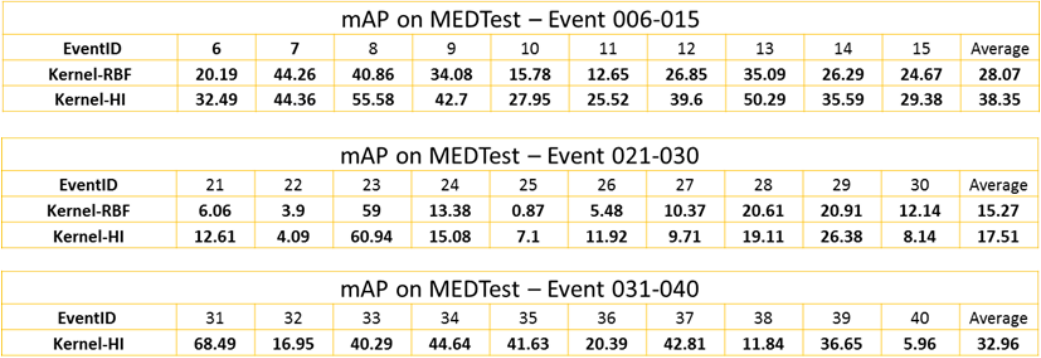


Figure 1: DeCaf Architecture



mAP on MEDTest – Event 006-015

EventID	6	7	8	9	10	11	12	13	14	15	Average
Kernel-RBF	20.19	44.26	40.86	34.08	15.78	12.65	26.85	35.09	26.29	24.67	28.07
Kernel-HI	32.49	44.36	55.58	42.7	27.95	25.52	39.6	50.29	35.59	29.38	38.35

mAP on MEDTest – Event 021-030

EventID	21	22	23	24	25	26	27	28	29	30	Average
Kernel-RBF	6.06	3.9	59	13.38	0.87	5.48	10.37	20.61	20.91	12.14	15.27
Kernel-HI	12.61	4.09	60.94	15.08	7.1	11.92	9.71	19.11	26.38	8.14	17.51

mAP on MEDTest – Event 031-040

EventID	31	32	33	34	35	36	37	38	39	40	Average
Kernel-HI	68.49	16.95	40.29	44.64	41.63	20.39	42.81	11.84	36.65	5.96	32.96

Figure 2: MED results on MEDTest using DeCaf feature

B. Overfeat Implementation: The Overfeat network was one of the best performing models on ImageNet ILSVRC 2013 challenge and the code and models are publicly available. We used Overfeat in MED, both in the form of generic deep features based on aggregate statistics from the whole video as well as in the form of concepts per key-frame for event query definition and recounting. Specifically, we used the features from layer 21 which is the softmax layer and produces 1000 dimensional vector per image, where each dimension corresponds to an object category. Given image keyframes from video, each image is resized into 256 x 256 image size and classified into 1000 image categories, in the form of a [1 x 1000] feature vector. The imagenet labels consist of general to specific categories. For example, there are 200 classes of dogs and 60 classes of cats, but only one type of plier. For feature aggregation, we considered several pooling strategies per video, namely, average pooling, thresholds at 0 and 1 followed by average pooling, and pooling following by binarizing responses. We tuned the performance on Medtest on PS11 categories. The best performance was achieved by threshold at 1.0 + average pooling, at 33.80% mAP.

3 High-Level Concept Features

One of the challenges for event recognition is to bridge the semantic gap between low-level features and high-level events. Concepts are directly connected to the Event Kit Descriptions. In Aurora system, we develop three types of concepts: visual concepts, audio concepts, and ASR/OCR text.

3.1 Visual Concepts

The visual concepts include object, scene and action concepts. The former two are usually defined over a still image, while the latter is defined over a spatial-temporal video volume.

A. Action Concepts:

Actions are typically atomic and localized motion and appearance patterns, which are strongly associated with some specific event. According to our evaluations on MED12 and MED13, action concepts base features are significant for MED, especially with less training examples such as 010Ex and 000Ex tasks. Training action detectors needs a large number of annotated videos. The human manual annotation is very laborious. To release annotators from the time-consuming job, we have been exploring a novel strategy to achieve semi-automatic concept annotation (SAA). A regular process of concept annotation over consumer videos starts with downloading relevant videos of one concept using search queries and then annotators start to annotate the starting and ending period as the positive clips. During this procedure, we noticed that, given a specific well-defined concept, the major parts of the majority of the collected videos are relevant. Having this observation, we developed the SAA system to automatically select relevant video clips for a given concept using PageRank technique, as shown in Figure 3. The assumption of our approach is that the majority of the videos are relevant to the concept.

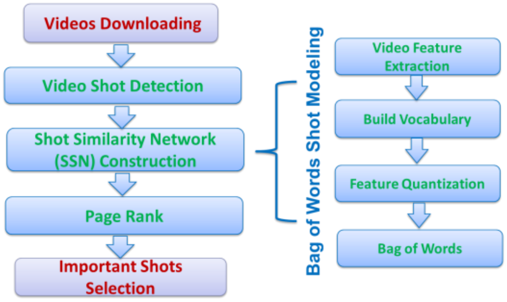
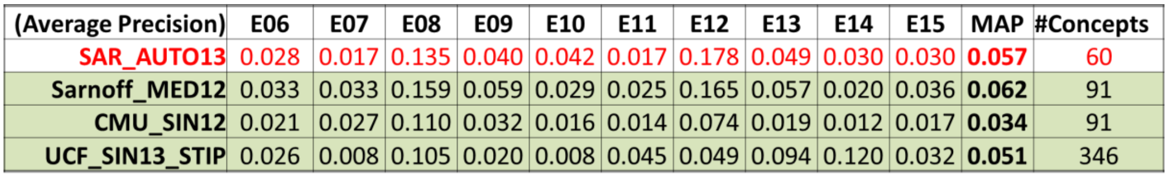


Figure 3: Semi-Supervised Automated Annotation

To evaluate the performance of SAA, we select 60 SAA action concepts, and then apply these detectors onto videos to extract concept features for MED, as discussed in 3.4. This waypoint experiment is conducted on PS11 events, as shown in Table 1, where SAR_AUTO13 is the SAA concepts. We compare it



(Average Precision)	E06	E07	E08	E09	E10	E11	E12	E13	E14	E15	MAP	#Concepts
SAR_AUTO13	0.028	0.017	0.135	0.040	0.042	0.017	0.178	0.049	0.030	0.030	**0.057**	60
Sarnoff_MED12	0.033	0.033	0.159	0.059	0.029	0.025	0.165	0.057	0.020	0.036	**0.062**	91
CMU_SIN12	0.021	0.027	0.110	0.032	0.016	0.014	0.074	0.019	0.012	0.017	**0.034**	91
UCF_SIN13_STIP	0.026	0.008	0.105	0.020	0.008	0.045	0.049	0.094	0.120	0.032	**0.051**	346

Table 1. MED results on PS11 using concept features.

with concepts Sarnoff_MED12 which is human annotated concepts, as well as SIN concepts. As we can see, SAA achieves competitive results to human annotated concepts.

B. Object and Scene Concepts:

The object and scene concepts are covered by TRECVID SIN concepts. TRECVID SIN task defines about 500 concepts which include objects such as TV screen, car, building, and scenes such as mountain, beach, street, office, and so on.

3.2 Audio Concepts

The audio concepts are either taken from CMU or annotated by our team. The Neural Network -based audio concept classification system employs the Parallel Neural Network Trainer TNet [17] technology from Brno University. It has a basic architecture which consists of two hidden layers with 1,000 neurons each and sigmoid activation functions. For the training phase a stochastic gradient descent optimizing cross-entropy loss function was used. The learning rate was updated using the “newbob” algorithm: It's kept fixed at LR=0.002 as long as the single epoch increment in cross-validation frames accuracy is higher than 0.5%. For the subsequent epochs, the learning rate is being halved until the cross-validation increment of the accuracy is inferior to the stopping threshold 0.1%. The NN weights and biases are randomly initialized and updates were performed per blocks of 1024 frames. Short-time Gaussian feature warping using a three-second window is used, and temporal regions containing identical frames are removed.

3.3 ASR/OCR Text Information

We adopted an information retrieval based approach 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. An index is created for ONR/ASR outputs of video clips using the Galago engine. 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.

3.4 Concept Based Event Representation (CBER)

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. 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 $\mathrm{C_{ij}} \propto \mathrm{p}(\mathrm{c_i}|\mathrm{w_j})$. Each $\mathrm{C_{ij}}$ 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 $\mathrm{C_{ij}}$. One option is to select the maximum detection score $\mathrm{C_i^{max}}$ 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 $\mathrm{C^{max}}$ 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 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.

4 Video Visual Representation

As we observed in previous MED evaluation, spatial pooling beyond BOW can further improve MED performance. We used 12 pre-specified Region of Interests (ROI) in MED13, but it significantly increases the number of training and search time. To overcome this disadvantage, we employ two video representation techniques in our MED14 system: Feature Mapping and Fisher Vector. Feature Mapping directly project the BOW features onto a higher dimensional kernel space, where the videos can be linearly separated. As a result, Linear SVM is able to split the hyper-space easily and efficiently. Feature Mapping still works on hard quantized features. Fisher Vector is the soft quantization version which embedding a video into a higher dimensional space. They are complementary to each other.

4.1 Feature Mapping

Feature mapping represents the video as the histogram of “words” corresponding to each feature type computed over the entire video clip. In order to compute BoW descriptors for each feature type, feature specific vocabularies are first learned using k-means clustering of raw features. All the features such as SIFT, DTF, STIP have a vocabulary of 10000 words. Once the features in a video are quantized using the respective vocabularies, a BoW is computed per feature. Event models could be trained using SVM with intersection kernel. While instead of directly training models with kernel SVM, we adopt an efficient and effective kernel approximation algorithm named as “Feature Mapping”(FMAP) to speed up the training and test steps. FMAP maps each dimension of histogram based feature into an infinite feature space, and then sample out a discrete feature map with finite dimension. After FMAP, the original feature is transformed into a higher dimensional mapped feature vector, which could be simply fed into a linear SVM to learn models. Comparing with traditional kernel SVM method, our FMAP framework achieves similar accuracies and around 50 times faster evaluation speed, as shown in table 2.

	BOW+ HI kernel		**Fisher Vector(GMM 256)**		**Feature Mapping**	
~4,100 training data, ~3,200 testing data						
Feature	Feature length: 10,000		Feature length: 163,840		Feature length: 30,000	
	mAP	**Trn/Tst time(sec)**	**mAP**	**Trn/Tst time(sec)**	**mAP**	**Trn/Tst time(sec)**
HOF	0.1862	Trn:4017.43 Tst: 4444.54	**0.2841**	Trn: 110.030 Tst: 163.697	0.1736	Trn:91.094 Tst: 85.058
HOG	0.2993		**0.3158**		0.2939	
MBH	0.3475		**0.3547**		0.3319	
STIP	0.2205		**0.2381**		0.2084	
D-SIFT	0.2720		**0.3222**		0.2769	
HESSIAN AFFINE	**0.2639**		0.1368		0.2492	

Table 2: The performance comparison between Feature Mapping, Fisher Vector and Kernel SVM. The mean Average precision (mAP) is on 3-fold dataset of Event 6-15. The table summarizes the accuracies and evaluation speed of different low-level features.

4.2 Fisher Vector

The second method is based on state-of-the-art image classification algorithm—"Fisher Vector"(FV). FV first trains a Gaussian Mixture Model with 256 components by standard EM algorithm, and then encodes each extracted low-level descriptor with the fisher kernel. After averaging all the fisher-kernel vectors into a single feature vector, a linear SVM is adopted for model generation. Comparing with the first feature mapping method, FV embeds higher-order statistics with better discriminative capacity, and also describes the fine-grained information in a probabilistic way which increases the generality of feature representation. For some events, FV achieves better performance than FMAP.

5 Zero-shot Learning for 000Ex

000Ex task is to conduct event detection without any training examples. The only information available is the event kit which provides the description of the target events. We developed a system which leverages the open knowledge source such as Wikipedia to bridge the gap between the event kit and the CBER models and available OCR/ASR text. As a result, our system is able to achieve good performance using the sequential dependence model [18] given only OCR/ASR information and concept detection results. This model assumes dependencies between neighboring words without modifying order and achieves substantial gains in common text collections. Differing to what we did in MED13, we have new query construction and search approach, as described below.

5.1 Zero-shot Query Construction

In the case when there are no exemplar videos, we created a query that targeted the OCR and ASR text sources as well as the action and object concepts extracted from videos. Consistent with the TRECVID guidelines, the queries were created by a person using an interface and not by automatic processing of the event description as was done in previous years.

To support this process, we created a simple web interface. The interface allowed for the separation of concepts and ASR/OCR terms into separate groupings referred to as "aspects." We experimented with a number of approaches when creating this approach, but ultimately settled on using two aspects called “primary” and “secondary.” The “primary” aspect represented concepts which the searcher felt strongly indicated the specific event, while the “secondary” aspect represented concepts and terms which although related to the event, did not clearly distinguish it from other events. For example, for event 42 (Building a Fire), the searcher selected "Flames" and "Smoke" as primary concepts and "Grilling food" and "thatch" as secondary concepts. "Flames" and "smoke" are strongly indicative of the act of making a fire, while "grilling food" is an action that may occur when people build a fire, and likewise "thatch" is a texture that may be seen amongst the materials used to make a fire.

For each aspect, the searcher could separately select visual concepts (likely to appear in videos), audio concepts (likely to be heard), OCR terms (likely to be as text in the video), and ASR terms (likely to be said in the video). OCR and ASR terms are typed directly into text boxes. (The OCR and ASR concepts were also augmented with a query expansion process that brought in large numbers of potential synonyms or alternate phrasings.) For concepts, the searcher entered a free-text query that ranked related concepts so they could be chosen; the searcher could also enter a concept ID directly if it was known.

As a final step of processing, the system automatically added any concepts that were synonymous with concepts that were already selected. For example, we had two concept detectors called XX, so if a searcher selected one, the other was almost certainly intended, too. When the query was constructed to the satisfaction of the searcher, it was converted into a semantic query for downstream processing.

5.2 Searching by OCR/ASR text

We processed the OCR text and the ASR text as separate Galago indexes. We used the appropriate parts of the semantic query to search the indexes and combined the results. Search was done using the Markov Random Field-based sequential dependence model (Metzler and Croft 2005). The model takes both ordered and unordered phrases into account.

The OCR and ASR rankings were merged. That combined score was then combined with an additional score that used the expanded OCR/ASR terms.

5.3 Searching by Video Concepts

Retrieval models that exploit non-sequential dependencies have shown to be successful in information retrieval (Metzler and Croft 2005) and image retrieval (Feng and Manmatha 2008). In our work, we focus on temporal and spatial relationships between concepts to improve video retrieval effectiveness. Our dependency work uses and MRF-based approach (Metzler and Croft 2005), one of the best performing algorithms in the information retrieval community. We explore the dependency settings shown in the figure below (v is a video frame and c is a concept): (1) full independence, where each concept is considered independently; (2) spatial dependence where the presence of two concepts in the same video frame is treated as important; and (3) temporal dependence, where having concepts occur in consecutive frames is treated as important.

Since the event kit is the only input in the EK0 task, the specific text used is the key to our performance, one of our focuses is to improve the textual descriptions of the events. We replaced the name of the event with a short query. Then, we automatically removed common phrases based upon the Lemur 418 English-word stop-list. Using the name and short description fields, we ran these queries against Wikipedia, adding a field of pseudo-relevance feedback terms.

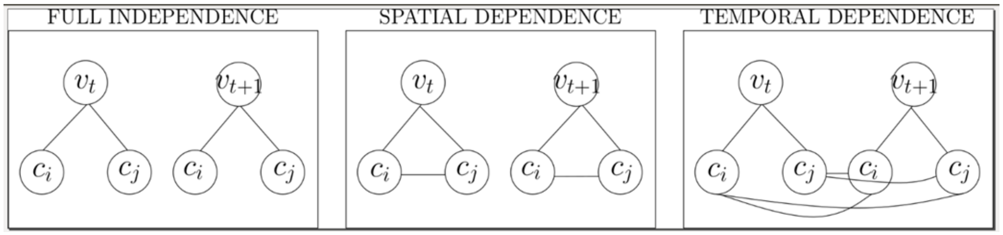


Table 1. Three Dependency Settings

6 Multimedia Event Recounting

The purpose of recounting is to provide data driven evidence to explain the decision made by an event classifier on a new video query. Recounting captures the key entities, actions, and scenes that pertain to that particular event and differentiate it from other events. Hence, it captures the semantic description of the scene that is also discriminative. Our approach to MER strongly integrates event classifier response with recounted evidence. We localize spatio-temporal coordinates of the evidence that contribute most to the final decision and rank them by their importance. To this end, we first use action and image concepts as intermediate semantic representation of a video. Then, we decompose the aggregate event score into contributions from individual concepts.

For concept based event recounting, concepts detectors are applied to video shots and image key frames. The scores are max-pooled statistics of the shot and frame level scores. These feature vectors represent concept responses per video and are used as inputs to train a non-linear SVM classifier. Specifically, we apply the min (histogram intersection) kernel, which maintains additivity and mutual independence of features while learning the weights. This property is exploited to derive a simple closed form solution for individual concept importance.

Concept importance is analyzed at two levels. First, after model training an "event query" is generated, in which concepts are ranked and filtered according to the weights learnt by the event model. During "recounting", the event query is used as a prior and combined with concept scores for the particular video. The details are as follows.

While for a general kernel it is difficult to interpret the SVM weights, we could approximately compute the concept importance from the weighted average of support vector dimensions. Given an additive SVM model, the decision function is represented by Equation 1,

$$h(x) = \sum\nolimits_{SV} \alpha^{sv} \cdot \sum\nolimits_{D} K(x_d^{sv}, x_d) + b \qquad (1)$$

$$h(x) = \sum\nolimits_{D} \sum\nolimits_{SV} h_d(x) + b \qquad (2)$$

where x^{sv} represents one of the support vectors and $\alpha^{sv}$is the signed weight of x^{sv}. K is the kernel function that operates on $x^{sv}$and a particular query instance x. An equivalent form is presented in Equation 2, where the sum over dimensions is swapped to separately compute contributions per dimension in the decision. Here, h_d equals $\alpha^{sv} K(x_d^{sv}, x_d)$ are the individual feature wise decisions.

A model driven ***event query*** is a template that defines a general list of concepts that are relevant to the event. This is computed as the weighted average of the support vectors, as shown in Equation 3. Here, h_d is the importance of the concept at dimension d, computed as the inner product between support vector weights and the corresponding concept scores. The concept scores are ranked according to this importance score and listed in the event query.

$$h_d = \sum_{SV} \alpha^{sv} \cdot x_d^{sv} \qquad (3) \qquad\qquad h_d(x) = \sum_{SV} \alpha^{sv} \cdot min(x_d^{sv}, x_d) \qquad (4)$$

Per video recounting: Given a new video query, the associated concept scores are incorporated to update evidence, as shown in Equation 4. Specifically, the min kernel function operates as a gated filter that allows only those concepts that are weighted highly by all the support vectors as well by the video query.

7 Experiments

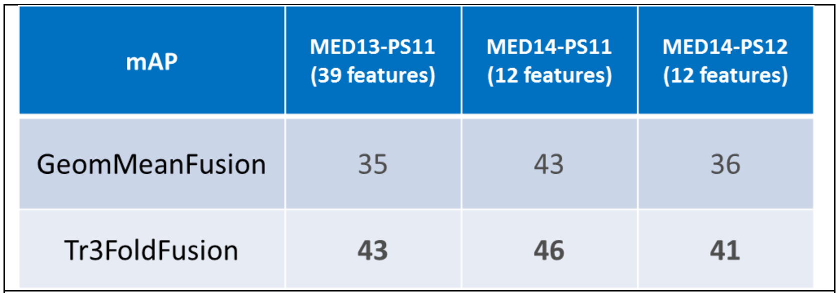
7.1 Training/Testing Methodology

We follow the MED14 evaluation plan [20], and use the exact positive and negative videos specified in the evaluation package to training our event models. All training process adopts the same 5K background videos as the negatives. For the low-level features, with Feature Mapping and Fisher Vector representation, we employ Linear SVM (libLinear [10]) to learn the event detectors.

7.2 Fusion Approach

Classifier fusion is the technique of fusing confidence scores generated by multiple classifiers to make final event decisions. Given enough training data, the task is to partition the training data into folds and train meta-classifiers on individual classifier scores. We perform three-fold fusion for the Ex100 module. Specifically, we partition the 100 examples and background randomly into three folds. Then we train on two folds, and test of the remaining fold iteratively. Finally, the test fold decisions are used for learning meta-classifiers.

We apply logistic regression (L1 and L2 loss), linear SVM, Adaboost and Extremely randomized trees for the ensemble learning. These were selected based on their average performance on the PS11 (events 6-15) dataset. Using any one classifier alone, instead of the ensemble leads to at least 2% loss in mAP scores. This is also corroborated by earlier studies. Overall, we observed an improvement of 3-5% over simple Geometric mean fusion, averaged across 20 events.



mAP	MED13-PS11 (39 features)	MED14-PS11 (12 features)	MED14-PS12 (12 features)
GeomMeanFusion	35	43	36
Tr3FoldFusion	**43**	**46**	**41**

Table 3: Comparing geometric fusion and learning based fusion results

Similar to MER, we also use the event model to analyze importance of individual visual classifiers in the final decision. These can be later used for improving or filtering weak classifiers to reduce computational footprint. Figure 2 shows the contribution of individual classifiers, which we have aggregated broadly into four modalities. For example, for event “parade”, motion and color contribute most to the final classification, while deep features are irrelevant. In contrast, for event “rock climbing”, all the modalities contribute almost equally.

8 Acknowledgement

This work has been supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center contract number D11-PC20066. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes not with-standing any copyright annotation thereon. 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. J. Liu, H. Cheng, et al., SRI-Sarnoff AURORA System at TRECVID 2013 Multimedia Event Detection and Recounting, TRECVID 2013.
2. Vedaldi, A. and Zisserman, A.Efficient Additive Kernels via Explicit Feature Maps, CVPR 2010.
3. D. Lowe, Distinctive image features from scale invariant key-points. IJCV, pp. 91-110. 2004
4. K. E. Sande, T. Gevers, C. G. Snoek, Evaluating color descriptors for object and scene recognition. TPAMI, 2010.
5. G.J. Burghouts and J.M. Geusebroek, Performance Evaluation of Local Color Invariants, CVIU, vol. 113, pp. 48-62, 2009.
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.
13. S. Chaudhuri, B. Raj. Unsupervised Structure Discovery for Semantic Analysis of Audio, Neural Information Processing Systems (NIPS), 2012
14. S. Chaudhuri, B. Raj., Unsupervised Hierarchical Structure Induction For Deeper Semantic Analysis of Audio. IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP) 2013
15. H. Cheng, J. Liu, O. Javed, et al., SRI-AURORA System at TRECVID 2012, Multimedia Event Detection and Recounting, TRECVID 2012 Evaluation Workshop, 2012.
16. UCF SIN Task report
17. K. Vesely, L. Burget, and F. Grezl, Parallel Training of Neural Networks for Speech Recognition, in Proceeding of Interspeech, 2010
18. Metzler, Donald, and W. Bruce Croft. A Markov random field model for term dependencies. In Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 472-479. ACM, 2005.
19. https://sourceforge.net/p/lemur/wiki/RankLib/
20. P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A.F. Smeaton, and G. Quéenot, TRECVID 2014 -- An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics, Proceedings of RECVID 2014, NIST, USA
21. J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, “Decaf: A deep con-

volutional activation feature for generic visual recognition,

22. M. D. Zeiler and R. Fergus, “Visualizing and understand- ing convolutional networks
23. M. Oquab, L. Bottou, I. Laptev, and J. Sivic, “Learning and Transferring Mid-Level Image Representations using Convolutional Neural Networks,” in *Proc. CVPR*, 2014.
24. Y. Jia, “Caffe: An open source convolutional architecture for fast feature embedding,” http://caffe.berkeleyvision. org/, 2013
25. S. Young, “The HTK hidden markov model toolkit: Design and philosophy”, Entropic Cambridge Research Laboratory, Ltd, vol. 2, pp. 2–44, 1994.