Informedia@TRECVID 2014

MED and MER

Shoou-I Yu, Lu Jiang, Zhongwen Xu, Zhenzhong Lan, Shicheng Xu, Xiaojun Chang, Xuanchong Li, Zexi Mao, Chuang Gan, Yajie Miao, Xingzhong Du, Yang Cai, Lara Martin, Nikolas Wolfe, Anurag Kumar, Huan Li, Ming Lin, Zhigang Ma, Yi Yang, Deyu Meng, Shiguang Shan, Pinar Duygulu Sahin, Susanne Burger, Florian Metze, Rita Singh, Bhiksha Raj, Teruko Mitamura, Richard Stern and Alexander Hauptmann

Carnegie Mellon University
5000 Forbes Ave., Pittsburgh,15213

We report on our system used in the TRECVID 2014 Multimedia Event Detection (MED) and Multimedia Event Recounting (MER) tasks. On the MED task, the CMU team achieved leading performance in the Semantic Query (SQ), 000Ex, 010Ex and 100Ex settings. Furthermore, SQ and 000Ex runs are significantly better than the submissions from the other teams. We attribute the good performance to 4 main components: 1) large-scale semantic concept detectors trained on video shots for SQ/000Ex systems, 2) better features such as improved trajectories and deep learning features for 010Ex/100Ex systems, 3) a novel Multistage Hybrid Late Fusion method for 010Ex/100Ex systems and 4) improved reranking methods for Pseudo Relevance Feedback for 000Ex/010Ex systems. On the MER task, our system utilizes a subset of features and detection results from the MED system from which the recounting is then generated. Recounting evidence is presented by selecting the most likely concepts detected in the salient shots of a video. Salient shots are detected by searching for shots which have high response when predicted by the video level event detector.

Semantic Indexing

Lu Jiang, Xiaojun Chang, Zexi Mao, Anil Armagan, Zhengzhong Lan, Xuanchong Li, Shoou-I Yu, Yi Yang, Deyu Meng, Pinar Duygulu-Sahin, Alexander Hauptmann

Carnegie Mellon University

We report on our system used in the TRECVID 2014 Semantic Indexing (SIN) task. We highlight the following new components: 1) self-paced learning pipeline for concept training, 2) dense trajectory with fisher vector encoding, 3) multi-modal pseudo relevance feedback for final results reranking and 4) deep convolutional neural networks directly trained on SIN keyframes. With the help of the above components, we were ranked in the top 3 among all type A runs (using only TRECVID IACC training data).

Surveillance Event Detection

Xingzhong Du[3], Yang Cai[1], Yicheng Zhao[2], Huan Li[1], Yi Yang[3] and Alexander Hauptmann[1]

Carnegie Mellon University[1], Beijing Institute of Technology[2], The University of Queensland[3]

We present a generic event detection system for the SED task of TRECVID 2014. It consists of two parts: the retrospective system and the interactive system. The retrospective system uses STIP, MoSIFT and Improved Dense Trajectories as low level features, and uses Fisher Vector encoding to represent shots generated by sliding window approach. Linear SVM is used to perform event detection. To improve performance, we applied several spatial schemas to generate the fisher vectors in our experiments. For the interactive system, we applied a general visualization scheme for all the events and a temporal locality based search method for user feedback utilization. Among primary runs of all teams, our retrospective system ranked 1st for 3 of 7 events in terms of actual DCR.

Informedia@TRECVID 2014

MED and MER

Shoou-I Yu, Lu Jiang, Zhongwen Xu, Zhenzhong Lan, Shicheng Xu, Xiaojun Chang, Xuanchong Li, Zexi Mao, Chuang Gan, Yajie Miao, Xingzhong Du, Yang Cai, Lara Martin, Nikolas Wolfe, Anurag Kumar, Huan Li, Ming Lin, Zhigang Ma, Yi Yang, Deyu Meng, Shiguang Shan, Pinar Duygulu Sahin, Susanne Burger, Florian Metze, Rita Singh, Bhiksha Raj, Teruko Mitamura, Richard Stern and Alexander Hauptmann
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh,15213

Abstract

We report on our system used in the TRECVID 2014 Multimedia Event Detection (MED) and Multimedia Event Recounting (MER) tasks. On the MED task, the CMU team achieved leading performance in the Semantic Query (SQ), 000Ex, 010Ex and 100Ex settings. Furthermore, SQ and 000Ex runs are significantly better than the submissions from the other teams. We attribute the good performance to 4 main components: 1) large-scale semantic concept detectors trained on video shots for SQ/000Ex systems, 2) better features such as improved trajectories and deep learning features for 010Ex/100Ex systems, 3) a novel Multistage Hybrid Late Fusion method for 010Ex/100Ex systems and 4) improved reranking methods for Pseudo Relevance Feedback for 000Ex/010Ex systems. On the MER task, our system utilizes a subset of features and detection results from the MED system from which the recounting is then generated. Recounting evidence is presented by selecting the most likely concepts detected in the salient shots of a video. Salient shots are detected by searching for shots which have high response when predicted by the video level event detector.

1. MED System

On the MED task, the CMU team has enhanced the MED 2013 [1] system in multiple directions, and these improvements have enabled the system to achieve leading performance in the SQ (Semantic Query), 000Ex, 010Ex and 100Ex settings. Furthermore, our system is very efficient in that it can complete Event Query Generation (EQG) in 16 minutes and Event Search (ES) over 200,000 videos in less than 5 minutes on a single workstation. The main improvements are highlighted below:

1. Large-scale semantic concept detectors (for SQ/000Ex systems): Our large-scale semantic video concept detectors, which is 10 times larger than the vocabulary from last year, enabled us to outperform other systems significantly on the SQ and 000Ex settings. The detector training is established based on self-paced learning theory [2] [3] [4].
2. CMU improved dense trajectories [5] (for 010Ex/100Ex systems): We enhanced improved trajectories [6] by encoding spatial and time information to model spatial information and temporal invariance.
3. ImageNet deep learning features (for 010Ex/100Ex systems): We have derived 15 different low-level deep learning features [7] from ImageNet [8], and these features have shown to be one of the best low-level features in MED.

4. Multistage Hybrid Late Fusion (for 010Ex/100Ex systems): We designed a multiple stage fusion method to fuse single feature predictions and early fusion predictions in a unified framework. At each stage we generate a different rank list based on different loss functions. These ranked lists are fused together at the final stage to ensure the robustness of the fusion results.
5. MMPRF/SPaR (for 000Ex/010Ex systems): Our novel reranking methods [4], provided consistent improvements on both the 000Ex and 010Ex runs for both the pre-specified and ad-hoc events. This contribution is evident because the reranking method is the only difference between our noPRF runs and PRF runs.
6. Efficient pipeline with linear classifiers and product quantization (PQ) (for 010Ex/100Ex and 000Ex PRF systems): As a first step towards an interactive system, we streamlined our system by employing linear classifiers and Product Quantization (PQ) [9], thus allowing us to perform search over 200,000 videos on 47 features in less than 5 minutes.

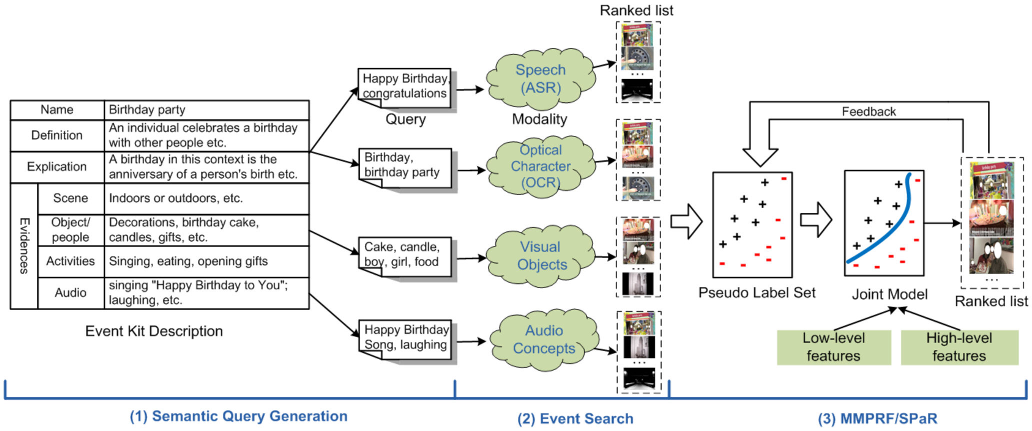
In the following sections, we will first give a quick overview of our system. Then, we will go into the details of the new components we developed this year.

1.1 System Overview

There are 4 tasks in MED this year: SQ, 000Ex, 010Ex and 100Ex. We designed two different pipelines for SQ/000Ex and 010Ex/100Ex respectively. The system for SQ/000Ex is very different from the 010Ex/100Ex system because the former system does not utilize any video training data. In the following section, we will describe our SQ/000Ex system and our 010Ex/100Ex system.

1.1.1 SQ/000Ex system

SQ/000Ex system takes the event-kit description as the input, and outputs a ranked list of relevant videos. It is an interesting task because it mostly resembles a real-world video search scenario, where users typically search videos by using query words than by providing example videos. According to [10], it consists of three major components, namely Semantic Query Generation (SQG), Event Search and Pseudo-Relevance Feedback (PRF), as shown in Figure 1.



Name		Birthday party
Definition		An individual celebrates a birthday with other people etc.
Explication		A birthday in this context is the anniversary of a person's birth etc.
Evidences	Scene	Indoors or outdoors, etc.
	Object/ people	Decorations, birthday cake, candles, gifts, etc.
	Activities	Singing, eating, opening gifts
	Audio	singing "Happy Birthday to You"; laughing, etc.

Figure 1: The framework of the SQ/000Ex system **[10]**.

Semantic Query Generation component translates the event kit description into a set of multimodal system queries that can be processed by the system. There are two challenges in this step. First, since the semantic vocabulary is usually limited, how to address the out-of-vocabulary issue in the event-kit description. Second given a query word, how to determine its modality as well as the weight associated with that modality. For example, the query "cake and candles" tends to be assigned to visual modality whereas the query "happy birthday" to ASR or OCR. For the first challenge, we use WordNet similarity [11], Point-wise Mutual Information on Wikipedia, and word2vec [11] [12] to generate a preliminary mapping that maps the event-kit description to the concepts in our vocabulary. Then it is then examined by human experts to figure out the final system query. The second challenge is tackled by prior knowledge provided by human experts. Indeed, this process is rather ad-hoc and premature as humans are in the loop and play an important role. Automatic SQG component is still not well understood, and thus worth of our further research effort.

Event Search component retrieves multiple ranked lists for a given system query. Our system incorporates various retrieval methods such as Vector Space Model, tf-idf, BM25, language model [13], etc. We found that different retrieval algorithms are good at different modalities. For examples, for ASR/OCR, the language model performs the best whereas for the visual concepts, the attribute retrieval model designed by our team obtains the best performance. An interesting observation that challenges our preconception is that for fixed vocabulary, the different yield by different retrieval methods can be significant. For examples, the relative difference for tf-idf model and language model is around 67% for the same set of ASR features. Surprisingly, a better retrieval model on worse features actually outperforms a worse retrieval model on better features. This observation suggests the role of retrieval models in SQ/000Ex system may be underestimated. After retrieving the ranked lists for all modalities, we apply a normalized fusion to fuse different ranked lists according to the weights specified in SQG.

PRF component refines the retrieved ranked lists by reranking videos. Our system incorporates MMPRF [10] and SPaR [4] to conduct the reranking, in which MMPRF is used to assign the starting values, and SPaR is used as the core reranking algorithm. The reranking is inspired by the self-paced learning proposed in [4] that the model is trained iteratively as opposed to simultaneously. Our methods are able to leverage high-level and low-level features which generally lead to an increased performance [14]. The high-level features used are ASR, OCR, and semantic visual concepts. The low-level features include DCNN, improved trajectories and MFCC features. We did not run PRF for SQ and 100Ex runs. For SQ run it is because our SQ run is essentially the same as our 0Ex run. For 100Ex it is because the improvement on the validation set is less significant.

	Visual Features	Audio Features
Low-level features	1. SIFT (BoW, FV) [15] 2. Color SIFT (CSIFT) (BoW, FV) [15] 3. Motion SIFT (MoSIFT) (BoW, FV) [16] 4. Transformed Color Histogram (TCH) (BoW, FV) [15] 5. STIP (BoW, FV) [17] **6. CMU Improved Dense Trajectory** (BoW, FV) [5]	1. MFCC (BoW, FV) 2. Acoustic Unit Descriptors (AUDs) (BoW) [18] 3. Large-scale pooling (LSF) (BoW) 4. Log Mel sparse coding (LMEL) (BoW) 5. UC.8k (BoW)
High-level features	**1. Semantic Indexing Concepts (SIN)** [19] 2. UCF101 [20] **3. YFCC** [21] 4. **Deep Convolutional Neural Networks (DCNN)** [7]	1. Acoustic Scene Analysis **2. Emotions** [22]
Text Features	1. Optical Character Recognition	**1. Automatic Speech Recognition**

Table 1: Features used in our system. Bolded features are new or enhanced features compared to last year's system. BoW: bag-of-words representation. FV: Fisher Vector representation.

1.1.2 010Ex/100Ex system

The MED pipeline for 010Ex and 100Ex consists of low-level feature extraction, feature representation, high-level feature extraction, model training and fusion, which are detailed as follows.

1. To encompass all aspects of a video, we extracted a wide variety of low-level features from the visual, audio and textual modality. Table 1 summarizes the features used in our system. The features marked in bold are the new features or features we have improved on, and the rest are features used in last year's system [1]. A total of 47 different feature representations are used in our system.
2. Low-level features are represented with the spatial bag-of-words [23] or Fisher Vector [24] representation.
3. High-level features such as Semantic Indexing concepts are extracted based on the low-level features. Deep Convolutional Neural Networks features are also computed on the extracted keyframes.
4. Single-feature linear SVM and linear regression models are trained. Also, early fusion is performed and their models computed. A total of 47 SVMs, 47 linear regressions, and 6 early fusion linear SVMs were computed during the EQG phase for 010Ex and 100Ex. 6 early fusion models consist of different combinations of features, which include combining all MFCCs, all audio features, all improved trajectories variants, and 3 different early fusion combinations of DCNNs.
5. The trained models are fused with our proposed Multistage Hybrid Late Fusion method, which fuses both late fusion and early fusion predictions [25]. R0 threshold is computed using the same method from last year [1].

1.1.3 System Performance

Figure 2 and Figure 3 summarizes the MAP performance of our system in different settings for pre-specified and adhoc events. Our system achieves leading performance in each setting. The SQ and 000Ex runs are significantly better than the other systems, which we attribute to the increased semantic concept vocabulary. The performance improvement over other systems in the 010Ex and 100Ex is smaller but consistent, and we attribute this improvement to better features and fusion methods. Finally, our reranking methods provide yet more performance gain for the 000Ex and 010Ex settings. We detail the sources of improvements in the following sections.

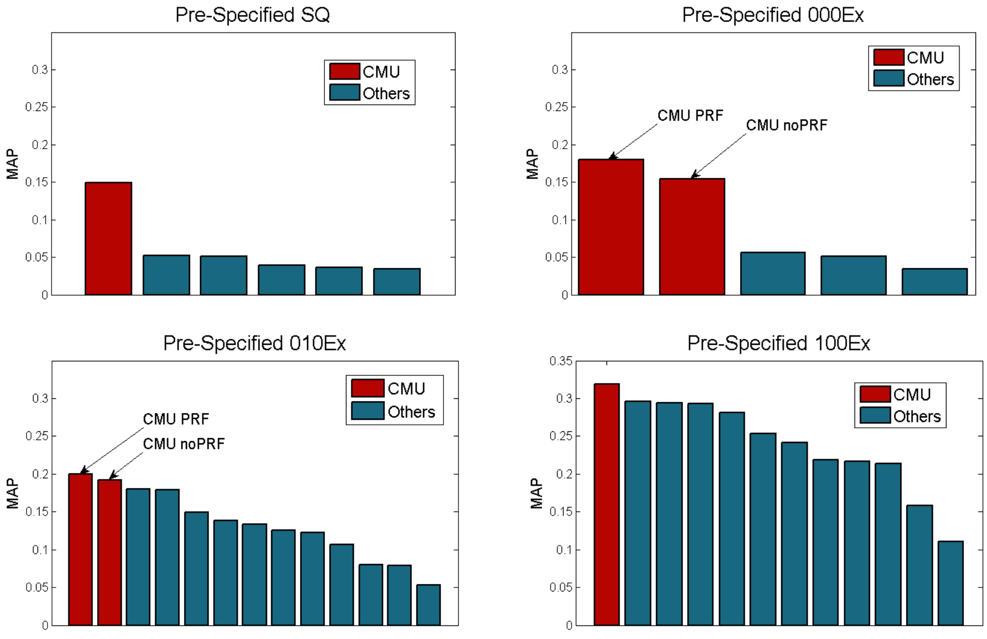


Figure 2: MAP performance on MED14-Eval Full in different settings for pre-specified events

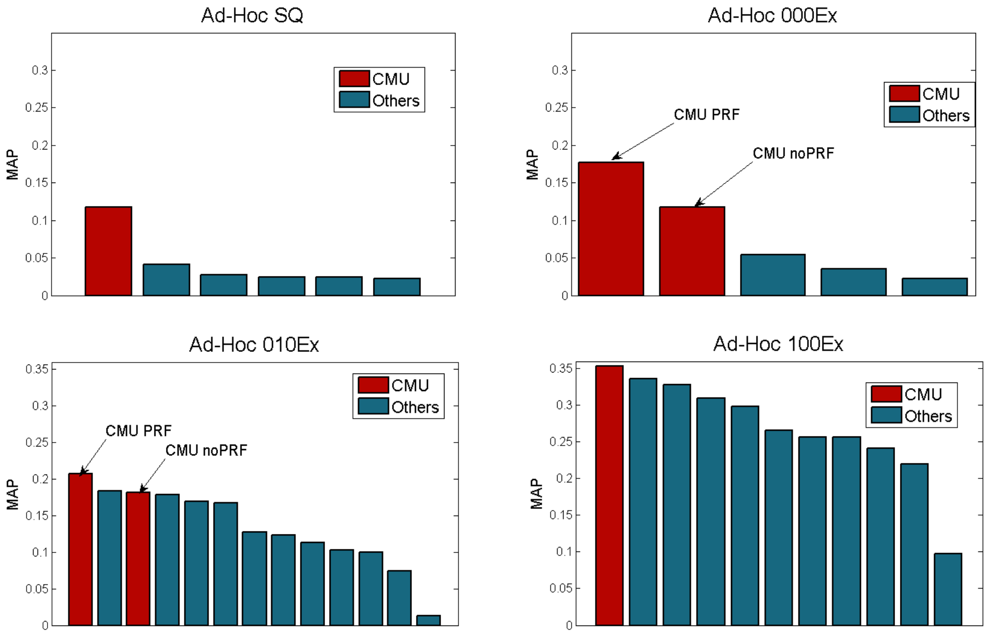


Figure 3: MAP performance on MED14-Eval Full in different settings for ad-hoc events

1.2 Improved Features

1.2.1 Large-scale Shot-based Semantic Concept

The shot-based semantic concepts are directly trained on video shots beyond still images for the following two reasons: 1) the shot-based concepts are of minimum domain difference; 2) it allows for action detection. The domain difference on the MED data is significant and thus detectors trained on still images usually not work well.

The shot-based semantic concept detectors are trained by our pipeline designed at Carnegie Mellon University based on our previous study on CascadeSVM and new study on self-paced learning [3] [2]. Our system includes more than 3,000 shot-based concept detectors which are trained over around 2.7 million shots using the standard improved dense trajectory [6]. It was 346 detectors over 0.2 million trained on SIFT/CSIFT/MoSIFT in the last year. The detectors are generic including people, scenes, activities, sports, and fine-grained actions described in [26]. The detectors are trained on several datasets including Semantic Indexing [19], YFCC100M [21], MEDResearch, etc. Some of the detectors are downloaded from the Internet, including Google Sports [27]. The notable increased quantity and quality of our detectors significantly attribute to the improvement of our SQ/000Ex system.

Training large-scale concept detectors on big data is very challenging. It is impossible without our effort in theoretical and practical studies. Regarding the theoretical progress, we explore the self-paced learning theory, which provides theoretically justification for the concept training. Self -paced learning is inspired by the learning process of humans and animals [2] [28], in which the samples are not learned randomly but organized in a meaningful order which illustrates from easy to gradually more complex ones. We advance the theory in two directions: augmenting the learning schemes [4] and learning from easy and diverse samples [3]. The two studies offer a theoretical foundation for our detector training system. We recommend to read [4] [3] for the details of our approach. We are still studying to implement the training paradigm on Cloud [29].

As for practical progress, we optimize our pipeline for high-dimensional features (around one-hundred-thousand dimensional dense vector). Specifically, we utilize large shared-memory machines to store the kernel matrices, e.g. 512GB, in size in memory to achieve 8 times speedup in training. This enabled us to efficiently train more than 3,000 concept detectors over 2.7 million shots by self-paced learning [3]. We use around 768 cores in Pittsburgh Computing Center to train for about 5 weeks, which roughly breaks down to two parts: low-level feature extraction for 3 weeks and concept training for 2 week. For testing, we convert our models to linear models to achieve around 1,000 times speedup in prediction. For example, it used to take about 60 days on 1,000 cores to extract semantic concepts for PROGTEST collection in 2012 but now it only takes 24 hours on 32-cores desktop.

In summary, our theoretical and practical progresses allows for developing sharp tools for large-scale concepts training on big data. Suppose we have 500 concepts over 0.5 million shots. Optimistically speaking, we can finish the training within 48 hours on 512 cores, including the raw feature extraction. After getting the models, the prediction for a shot/video only takes 0.125s on a single core with 16GB memory.

1.2.2 CMU Improved Dense Trajectories

CMU Improved Dense Trajectory [5], also known as multi-skip feature stacking (MIFS), improves the original Improved Dense Trajectory [6] in two ways: first, it achieves temporal scale-invariance by extracting features from videos with different frame rates, which are generated by skipping frame at certain intervals. Different from what has been described in [6], we use the combination of level 0, 2 and 5 to balance the speed and performance. Second, we encode spatial and location information into Fisher vector representation by attaching spatial (*x, y*) and temporal (*t*) location to the raw features. By using above two modifications, we can improve MAP on MEDTEST14 by about 2%, absolutely. For details, please consult [5].

1.2.3 Features from DCNN Models Trained on ImageNet

We extract a total of 15 different DCNN features. The models are all trained on ImageNet. 3 models are trained on the whole Imagenet dataset which contains around 14 million labeled images. The structure of the network is as described in [30]. We took the networks at the stage of epoch 5, 6 and 7 and generate features for MED key-frames using the first fully connected layer and probability layer. For generating video features from image features, we use both maximum pooling and average pooling for probability layer and only average pooling for fully connected layer. This procedure results in 9 DCNN-Imagenet representations for each video. Another 5 models were trained from training images of ImageNet ILSVRC 2012 dataset with 1.28 million images and 1,000 classes. The training process was tuned on the ImageNet ILSVRC 2012 validation set with 50 thousand images. Two models were trained with six convolutional layers, two models were trained with smaller filters, and one was trained with larger number of filters. A multi-view representation was used for one of the models. The network structure is as described in [31]. Except for different structures among models, the models with the same structures differ in initialization. These models result in another 6 different feature representations. More details and also some further improvements after the submission are described in [32].

1.2.4 Kaldi ASR

Our ASR system is based on Kaldi [33], an open-source speech recognition toolkit. We build the HMM/GMM acoustic model with speaker adaptive training. The models are trained from instructional video data [26]. Our trigram language model is pruned aggressively to speed up decoding. When applied on the evaluation data, we first utilize Janus [34] to segment out speech segments, which is subsequently given to the Kaldi system to generate the best hypothesis for each utterance. Two passes of decoding are performed with an overall real-time factor of 8.

1.2.5 Emotions

In addition to other audio-semantic features which we have used in the past, such as noisemes, we have trained random-tree models on the IEMOCAP [22] database for emotion classification. Our models take acoustic features extracted from OpenSmile [35] and classify each 2s frame with 100ms overlap as an angry, sad, happy, or neutral emotion. The most common label is then used for the entire video’s “emotion”.

1.3 Multistage Hybrid Late Fusion Method

We propose a new learning based late fusion algorithm, named the “Multistage Hybrid Late Fusion”. The key idea of our method is to model the fusion process as a multiple stage generative process. At each stage, we design a specific algorithm to extract the information we need. The methods used in the multiple stage fusion include dimension reduction, clustering, and stochastic optimization. After the multistage information extraction, we perform hybrid fusion where we simultaneously exploit many fusion strategies to learn multiple fusion weights. Subsequently, the results of the multiple strategies are averaged to get the final output.

1.4 Self-Paced Reranking

Our PRF system is implemented according to SPaR detailed in [4]. SPaR represents a general method of addressing multimodal pseudo relevance feedback for SQ/000Ex video search. As opposed to utilizing all samples to learn a model simultaneously, the proposed model is learned gradually from easy to more complex samples. In the context of the reranking problem, the easy samples are the top-ranked videos that have smaller loss. As the name “self-paced” suggests, in every iteration, SPaR examines the “easiness” of each sample based on what it has already learned, and adaptively determines their weights to be used in the subsequent iterations.

The mixture weighting/scheme self-paced function is used, since we empirically found it outperforms the binary self-paced function on the validation set. The mixture self-paced function assigns 1.0 weight to top 5 videos and a weight from 0.2 to 1 for the videos ranked between top 6 to top 15 (i.e. 0.2 for the top 15 video), according to its loss. Since the starting values can significantly affect final performance, we did not use random starting values but the reasonable starting values generated by MMPRF [10]. The off-the-shell linear regression model is used to train the reranking model. The high-level features used are ASR, OCR, and semantic visual concepts. The low-level features are DCNN, improved trajectories and MFCC features. We did not run PRF for SQ since our 000Ex and SQ runs are very similar. The final run is the average fusion of the original ranked list and the reranked list to leverage high-level and low-level features, which, according to [14], usually yields better performance. To be prudent, the number of iteration is no more than 2 in our final submissions. For more details, please refer to [10] and [4].

The contribution of our reranking methods is evident because the reranking method is the only difference between our noPRF runs and PRF runs. According to the MAP on MED14Eval Full (200K videos), our reranking method boosts the MAP of 000Ex system by a relative 16.8% for pre-specified events and a relative 51.2% for ad-hoc events. Besides, it also boosts the 010Ex system by a relative 4.2% for pre-specified events, and a relative 13.7% for ad-hoc events. This observation is consistent with the ones reported in [10] and [4]. Note that the ad-hoc queries are very challenging because the query is unknown to the system beforehand, and after getting the query it has to finish the process within an hour. As we see, our reranking methods still manage to yield significant improvement on ad-hoc events.

It is interesting that our 000Ex system for ad-hoc events actually outperforms 010Ex systems of most of other teams. This year, the difference between the best 000Ex with PRF (17.7%) and the best 010Ex noPRF (18.2%) is marginal. In last year, however, this difference is huge, and the best 000Ex system is 10.1% whereas the best 010Ex system is 21.2% (The runs in different years are not comparable since they are on different datasets). This observation suggesting that the gap of real-world 000Ex event search system is shrinking rapidly.

We observed two scenarios where the proposed reranking methods could fail. First, when the initial top-ranked videos retrieved by queries are completely off-topic. This may be due to irrelevant queries or poor quality of the high-level features, e.g. ASR and semantic concepts. In this case, SPaR may not recover from the inferior original ranked list, e.g. the query brought by ``E022 Cleaning an appliance" are off-topic (on cooking in kitchen). Second, SPaR may not help when the features used in reranking are not discriminative to the queries, e.g. for ``E025 Marriage Proposal", our system lacks of meaningful features/detectors such as ``stand on knees". Therefore even if 10 true positives are used (010Ex), the AP is still bad (0.3%) on the MED14test dataset.

1.5 Efficient EQG and ES

To strive for the ultimate goal of interactive MED, we targeted completing Semantic/Event Query Generation (EQG) in 30 minutes (1800 seconds) and Event Search (ES) in 5 minutes (300 seconds). This is a big challenge for the 010Ex and 100Ex pipeline, as we utilized 47 features and 100 classifiers to create the final ranked list. The semantic query and 000Ex pipelines are a lot simpler thus timing is not a big issue. Therefore, we will focus on 010Ex and 100Ex timing in the next few paragraphs. To speed up EQG and ES for the 010Ex and 100Ex system, we performed optimizations in three different directions: 1) decreasing computation requirements, 2) decreasing I/O requirements and 3) utilizing GPUs. Computational requirements for EQG and ES are decreased by replacing kernel classifiers with linear classifiers. I/O requirements for ES are decreased by compressing features vectors with Product Quantization (PQ). GPUs are utilized to compute fast matrix inverse for linear regression and for fast prediction of videos.

1.5.1 Replacing Kernel Classifiers by Linear Classifiers

Kernel classifiers are slow during prediction time because to perform prediction on an evaluation video vector, it is often required to compute the dot-product between the evaluation video feature and each vector in the training set. For MED14, we have around 5000 training videos, so 5000 dot products are required to predict one video. This is a very slow process, and preliminary experiments show that prediction of improved trajectory fisher vectors (109056 dimensions) on 200,000 videos requires 50 minutes on a NVIDIA K-20 GPU. Therefore, in order to perform ES in 5 minutes, we switched to linear classifiers, which require only one dot product per evaluated vector, so in theory we sped up the prediction process by 5000x for MED14. However, bag-of-word features do not perform well with linear kernels. Therefore, we used the Explicit Feature Map (EFM) [36] to map all bag-of-words to a linearly separable space before applying the linear classifier. As the EFM is an approximation, we run the risk of a slight drop in performance. Figure 4 shows the performance difference of before (“Original”, blue bar) and after (“Mapped”, red bar) EFM. For most features, we suffer a slight drop in performance, which is still cost-effective given that we sped up our prediction (ES) speed by 5000x. EQG speed is also improved because we need to search over less parameters during cross-validation when using linear classifiers. We see a 15x speed up for SVM training and a 5x speed up for Linear Regression training. On the other hand, we no longer use GMM supervector-based features [37], because they perform best with the RBF-kernel which is not supported by EFM.

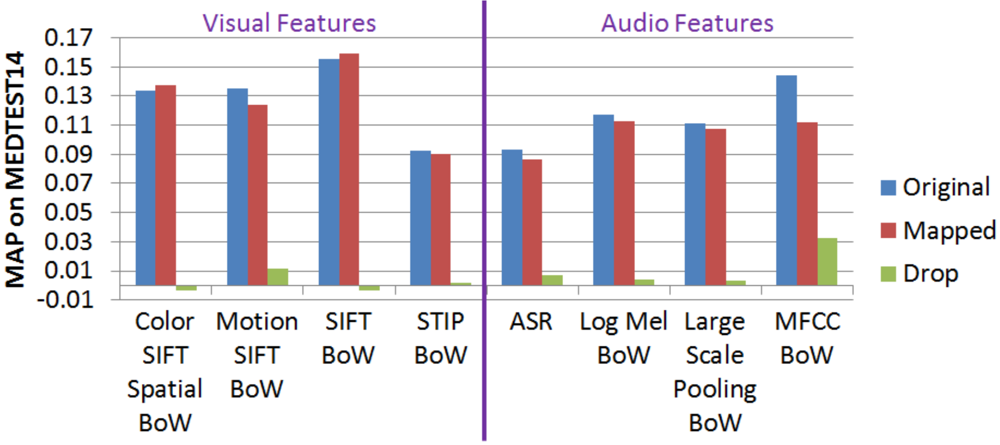


Figure 4: Performance before and after EFM for selected features

1.5.2 Feature Compression with Product Quantization

In order to improve I/O performance, we compress our features using Product Quantization (PQ). Compression is crucial because reading uncompressed features can take a lot of time. However, as PQ performs lossy compression, the quality of the final ranked list may degrade. To quantify the degradation, we performed experiments on MEDTEST14 for 23 features which is a subset of the features we used this year. Table 2 shows the relative drop in performance when using different quantization parameters. On average, we see a relative 2% drop in performance after performing 32X PQ compression, which is a worthwhile tradeoff given that we have decreased the I/O requirements by a factor of 32. In our final submission, we use a compression factor of 32X.

Configuration (Average over 23 features)	PQ 16X Compression		PQ 32X Compression	
	Average Drop	Max Drop	Average Drop	Max Drop
EK100 Linear SVM	0.50%	6.80%	0.93%	6.72%
EK100 Linear Regression	1.42%	11.81%	2.01%	12.42%
EK10 Linear SVM	1.05%	19.60%	1.30%	19.39%
EK10 Linear Regression	0.04%	8.64%	0.60%	12.03%

Table 2: Performance drop under different PQ compression factors

1.5.3 Utilizing GPUs for Fast Linear Regression and Linear Classifier Prediction

As we are limited to a single workstation for EQG and ES, we utilized all available computing resources on the workstation, which includes CPUs and GPUs. Exploiting the fact that matrix inversion on GPUs are faster than CPUs, we trained our linear regression models on GPUs, which is 4 times faster than running on a 12 core CPU. We also ported the linear classifier prediction step to the GPU, which runs as fast as a 12 core CPU. All EQG and ES are performed on a single workstation which has 2 Intel(R) Xeon(R) CPU E5-2640 6 core processors, 4 NVIDIA TESLA K20's, 128GB RAM, and 10 1T SSDs setup in RAID 10 to increase I/O bandwidth.

1.5.4 Overall Speed Improvements

As both EFM and PQ are approximations, we quantified the drop in performance when both methods are used. The results are shown in Table 3 below. We see a 3% relative drop in performance for 100Ex and a slight gain in performance for 010Ex. Despite slight drop in performance, speed has been substantially decreased as shown in Table 3. We have sped up our system by 19 times for EQG and 38 times for ES with a cost of 3% relative drop in performance, which is negligible given the large efficiency gain.

Runs (MEDTEST14)	MAP Performance		Timing (s) for 100Ex	
	100Ex	010Ex	EQG	ES
Original (no EFM, no PQ, with GMM features)	0.405	0.266	12150[1]	5430[1]
With EFM, PQ 32X, no GMM features	0.394	0.270	926	142
Improvement	-2.7%	1.5%	1940%	3823%

Table 3: Performance difference after utilizing EFM and PQ

We further break down the pipeline and report timing information for each step. In the EQG phase, the first step is the *classifier training* phase, where we train 47 SVM classifiers, 47 linear regression models and 6 early fusion SVM classifiers. SVMs are trained using CPUs [38], while linear regression models are trained using GPUs. The second step is the *fusion weight learning* phase, where we run our Multistage Hybrid Late Fusion method to learn weights for the 100 classifiers learned. The average timing information and standard deviation for the 10 events in the adhoc submission (E041-E050) are shown in Table 4. The 010Ex scenario is faster than the 100Ex during classifier training because 010Ex does not perform cross-validation to tune parameters, which is the same as last year's system [1]. In sum, it took on average 6 minutes 52 seconds for 010Ex EQG and 15 minutes 26 seconds for 100Ex EQG.

[1] Extrapolated timing for MED13 pipeline

Setting	Classifier Training (s)	Fusion Weights Learning (s)	Total (s)
010Ex	385.3 ± 6.4	26.2 ± 0.63	411.5 ± 6.38
100Ex	864 ± 42.7	62 ± 0.47	926 ± 42.54

Table 4: EQG timing for 010Ex/100Ex for adhoc events

In the ES phase, both the 010Ex and 100Ex pipelines perform *classifier prediction* followed by *Fusion of Predictions & Threshold Learning*. The 010Ex pipeline further goes through MER generation, reranking and MER generation for reranked results. The average timing information and standard deviation for the 10 events in the adhoc submission (E041-E050) are shown in Table 5. On average, the 010Ex pipeline with reranking requires 5 minutes 15 seconds. However, the 010Ex pipeline without reranking only requires 3 minutes 31 seconds. The 100Ex pipeline requires 2 minutes 22 seconds on average.

Setting	Classifier Prediction (s)	Fusion of Predictions & Threshold Learning (s)	MER (s)	Reranking (s)	MER on Reranked Results (s)	Total (s)
010Ex	133.6 ± 7.41	13.3 ± 0.67	64.2 ± 21.49	56.9 ± 2.28	46.6 ± 1.26	314.6 ± 20.31
100Ex	128.7 ± 3.56	13.2 ± 0.79				141.9 ± 3.78

Table 5: ES timing for 010Ex/100Ex for adhoc events

2. MER System

Our MER system takes event query xml from the I/O server, threshold and detection results from MED system, and use features and models from metadata store to compute recounting evidences for all videos above the R0 threshold. Around 2000 high quality concepts have been renamed and are available for recounting.

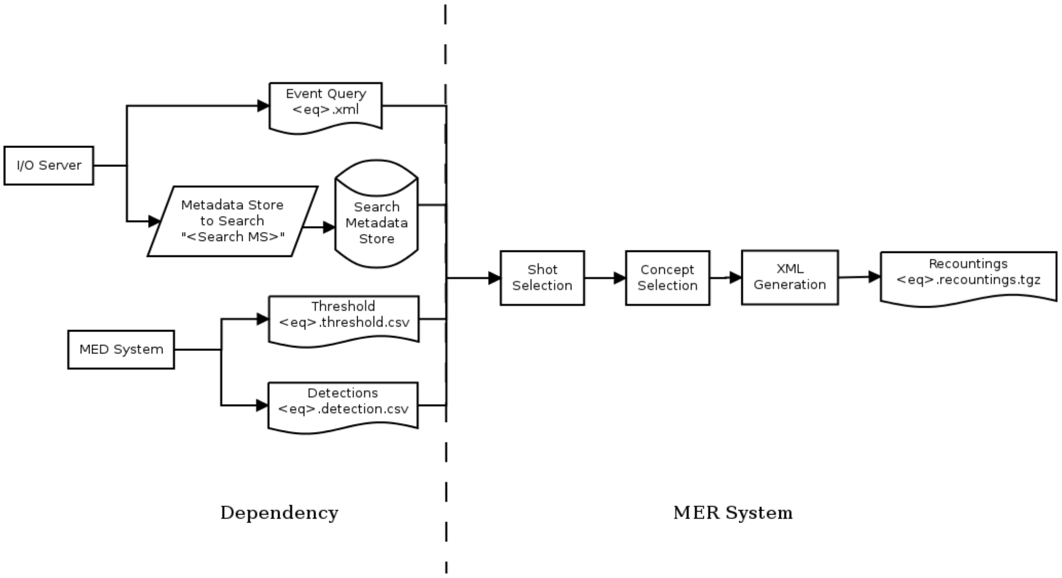

I/O Server
Event Query <eq>.xml
Metadata Store to Search "<Search MS>"
Search Metadata Store
MED System
Threshold <eq>.threshold.csv
Detections <eq>.detection.csv
Shot Selection
Concept Selection
XML Generation
Recountings <eq>.recountings.tgz
Dependency
MER System

Figure 5: MER system dependency and workflow

For each video, the evidences are computed in three steps. First, we select top five confident shots by applying video model on shot features. Second, one concept with the highest detection score is selected for each shot as a visual-audio evidence. The time period of the shot is used for evidence timing localization. The evidences from top three shots are marked as key evidence, the other two are marked as non-key evidence. Finally, the recounting xml is generated by filling evidence information into the event query xml. Figure 5 shows the dependency and work flow of our MER system.

We have submitted our recounting results for both 010Ex noPRF and 010Ex PRF run. Our system uses 8.2% of original video duration to localize key evidence snippets, which is the shortest among all teams. But we achieve relatively good results on evidence quality. Table 6 shows our judge results on *query conciseness* and *key evidence convincing*.

	Query Conciseness	Key Evidence Convincing
Strongly Disagree	7%	11%
Disagree	15%	15%
Neutral	18%	17%
Agree	48%	34%
Strongly Agree	12%	23%

Table 6: MER results on *Query Conciseness* and *Key Evidence Convincing*

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.

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. Specifically, it used the Blacklight system at the Pittsburgh Supercomputing Center (PSC).

Reference

[1] Z.-Z. Lan, L. Jiang, S.-I. Yu, S. Rawat, Y. Cai, C. Gao, S. X. al. and et, "CMU-Informedia at TRECVID 2013 multimedia event detection," in *TRECVID Workshop*, 2013.

[2] M. P. Kumar, B. Packer and D. Koller., "Self-paced learning for latent variable models," in *NIPS*, 2010.

[3] Lu Jiang, Deyu Meng, Shoou-I Yu, Zhen-Zhong Lan, Shiguang Shan, Alexander Hauptmann, "Self-paced Learning with Diversity," in *NIPS*, 2014.

[4] L. Jiang, D. Meng, T. Mitamura and A. Hauptmann, "Easy Samples First: Self-paced Reranking for Zero-Example Multimedia Search," in *ACM MM*, 2014.

[5] Z. Lan, M. Lin, X. Li, A. G. Hauptmann and B. Raj, "Beyond Gaussian Pyramid: Multi-skip Feature Stacking for Action Recognition," in *arXiv preprint arXiv:1411.6660*, 2014.

[6] H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," in *ICCV*, 2013.

[7] A. Krizhevsky, I. Sutskever and G. E. Hinton., "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012.

[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li and L. Fei-Fei, "ImageNet: A Large-Scale

Hierarchical Image Database," in *CVPR*, 2009.

[9] H. Jegou, M. Douze and C. Schmid., " Product quantization for nearest neighbor search," in *PAMI*, 2011.

[10] L. Jiang, T. Mitamura, S.-I. Yu and A. Hauptmann, "Zero-Example Event Search using MultiModal Pseudo Relevance Feedback," in *ICMR*, 2014.

[11] "WordNet Similarity for Java, https://code.google.com/p/ws4j/".

[12] T. Mikolov, I. Sutskever, K. Chen, G. Corrado and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *NIPS*, 2013.

[13] C. Zhai and J. Lafferty, "A study of smoothing methods forlanguage models applied to ad hoc information retrieval," in *SIGIR*, 2001.

[14] L. Jiang, A. Hauptmann and G. Xiang, "Leveraging High-level and Low-level Features for Multimedia Event Detection," in *ACM MM*, 2012.

[15] K. v. d. Sande, T. Gevers and C. Snoek, "Evaluating color descriptors for object and scene recognition," *TPAMI*, 2010.

[16] M. Chen and A. Hauptmann, "MoSIFT: Reocgnizing Human Actions in Surveillance Videos," Carnegie Mellon University, 2009.

[17] H. Wang, M. M. Ullah, A. Klaser, I. Laptev and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.

[18] S. Chaudhuri, M. Harvilla and B. Raj, "Unsupervised Learning of Acoustic Unit Descriptors for Audio Content Representation and Classification," in *Interspeech*, 2011.

[19] 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," in *TRECVID*, 2014.

[20] K. Soomro, A. R. Zamir and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," in *arXiv preprint arXiv:1212.0402*, 2012.

[21] "Yahoo Flickr Creative Commons, http://webscope.sandbox.yahoo.com/catalog.php?datatype=i&did=67".

[22] C. Busso and a. et, "IEMOCAP: Interactive emotional dyadic motion capture database," in *Language resources and evaluation*, 2008.

[23] S. Lazebnik, C. Schmid and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *CVPR*, 2006.

[24] K. Chatfield, A. V. V. Lempitsky and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," in *BMVC*, 2011.

[25] Z.-z. Lan, L. Bao, S.-I. Yu, W. Liu and A. G. Hauptmann, "Multimedia classification and event detection using double fusion," in *Multimedia Tools and Applications*, 2013.

[26] S.-I. Yu, L. Jiang and A. Hauptmann, "Instructional Videos for Unsupervised Harvesting and Learning of Action Examples," in *ACM MM*, 2014.

[27] "Google Sport Concept Detectors, http://gr.xjtu.edu.cn/web/dymeng/4".

[28] Y. Bengio, J. Louradour, R. Collobert and J. Weston, "Curriculum learning," in *ICML*, 2009.

[29] "Cascade SVM, https://code.google.com/p/cascadesvm/".

[30] Z.-Z. Lan, Y. Yang, N. Ballas, S.-I. Yu and A. Haputmann, "Resource Constrained Multimedia Event Detection," in *Multimedia Modeling*, 2014.

[31] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *ECCV*, 2014.

[32] Z. Xu, Y. Yang and A. G. Hauptmann, "A Discriminative CNN Video Representation for Event Detection," in *arXiv preprint arXiv:1411.4006*, 2014.

[33] D. Povey and e. al, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[34] H. Soltau, F. Metze, C. Fügen and A. Waibel, "A One-pass Decoder based on Polymorphic Linguistic Context Assignment," in *ASRU*, 2001.

[35] F. Eyben, F. Weninger, F. Gross and B. Schuller, "Recent Developments in openSMILE, the Munich Open-Source Multimedia Feature Extractor," in *ACM MM*, 2013.

[36] A. Vedaldi and A. Zisserman, "Efficient additive kernels via explicit feature maps," in *PAMI*, 2012.

[37] W. Campbell and D. Sturim, "Support vector machines using GMM supervectors for speaker verification," *IEEE Signal Processing Letters*, 2006.

[38] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," in *ACM Transactions on Intelligent Systems and Technology*, 2011.

CMU Informedia @TRECVID 2014: Semantic Indexing

Lu Jiang, Xiaojun Chang, Zexi Mao, Anil Armagan, Zhengzhong Lan, Xuanchong Li, Shoou-I Yu, Yi Yang, Deyu Meng, Pinar Duygulu-Sahin, Alexander Hauptmann
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We report on our system used in the TRECVID 2014 Semantic Indexing (SIN) task. We highlight the following new components: 1) self-paced learning pipeline for concept training, 2) dense trajectory with fisher vector encoding, 3) multi-modal pseudo relevance feedback for final results reranking and 4) deep convolutional neural networks directly trained on SIN keyframes. With the help of above components, we were ranked top 3 among all type A runs (using only TRECVID IACC training data).

1. System Description

The training set used is identical to the set used last year, which includes around 370 thousand shots from IACC.1.tv10.training and IACC.1.A-C collections [11]. Our system includes the implementations of two pipelines: SVM-based self-paced learning pipeline and Deep Convolutional Neural Networks (DCNN)-based pipeline.

For the self-paced learning pipeline, we used the features listed in Table 1 for this year's submission.

Table 1 Summary of features used in our SVM-based self-paced learning pipeline.

Raw feature	Representation
SIFT harrislaplace [1]	Spatial Bag-of-words [5]
SIFT densesampling [1]	Spatial Bag-of-words [5]
Color SIFT harrislaplace [1]	Spatial Bag-of-words [5]
Color SIFT densesampling [1]	Spatial Bag-of-words [5]
Improved Dense Trajectory [2]	Fisher Vector (non-spatial) [3]
Metadata [4]	Bag-of-words

The spatial bag-of-word feature is determined with the help of JS-Tiling described in [5]. The codebook size of the BoW features is 4096. Among all features, improved dense trajectory [2] is the only shot-based motion feature, which is encoded by the fisher vector [6]. The dimension of the final fisher vector is 109,056. We used non-spatial fisher vector as we observed adding spatial information only leads to marginal improvements. For the bag-of-words features, the intersection kernel is used, whereas for fisher vector, linear kernel is used. No audio features are used in our system. The metadata of a video includes its title, uploader [4] and description information extracted from XML file.

The concept models are trained based on self-paced learning, which provides theoretically justification for the large-scale concept training [7][8]. The learning paradigm is inspired by the learning principle underlying the cognitive process of humans and animals [9][10], which generally starts with learning easier aspects of an aimed task, and then gradually takes more complex examples into consideration. Since the complexity of the training samples usually varies in large-scale real-world dataset, the samples should not be learned randomly but organized in a meaningful order which illustrates from easy to gradually more complex ones. Figure 1 illustrates representative positive samples in TRECVID SIN 2014 dataset for the concept "bus", where a palpable difference between easy and complex examples can be observed.

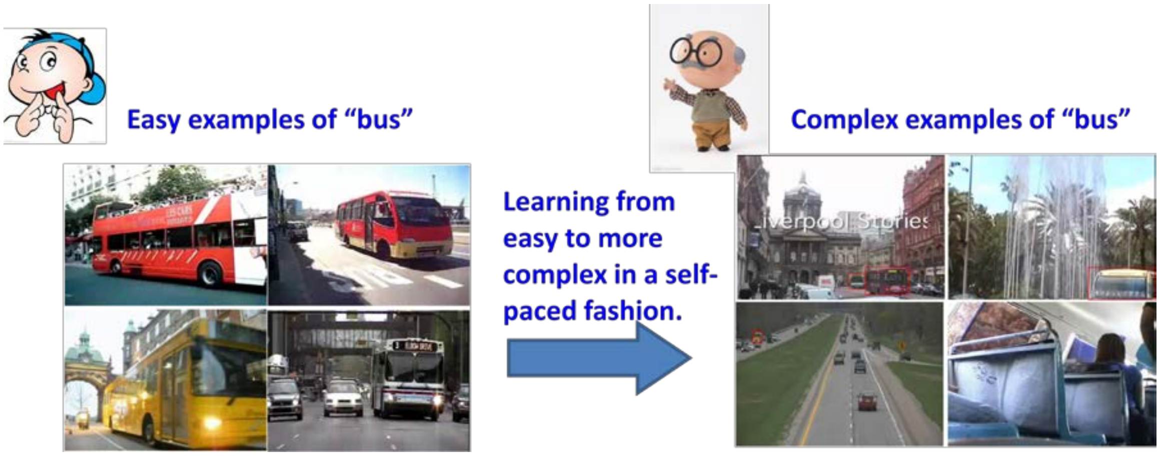

Easy examples of "bus"
Complex examples of "bus"
Learning from easy to more complex in a self-paced fashion.
Liverpool Stories

Figure 1: the positive examples for the concept "bus" in the TRECVID SIN dataset

Self-paced concept training is interesting for the following reasons: first it represents a novel framework that has never been studied by any of the TRECVID team; second, it offers a theoretically sound way to approach large-scale concept training, as opposed to heuristic methods in most of the existing work such as cascadeSVM. We advance the theory in two directions: augmenting the learning schemes [8] and learning from easy and diverse samples [7]. Above two studies offer a theoretical foundation for our detector training system. We recommend reading the papers for the details of the approach. This pipeline is also very efficient, and we are able to finish training the full SIN dataset (346 concepts from 0.6 million shots) with no more than 48 hours on 512 CPU cores.

For the DCNN-based pipeline: in this year's submission, rather than using DCNN as concept detectors, we train DCNN models directly on the provided keyframes [17]. The DCNN models are pre-trained on the ImageNet ILSVRC2012 [13] dataset. Every layer except the last in the ImageNet model is used to initialize the SIN models. The structure of the last layer is changed in order to produce 347 output probabilities (346 concept + null). Two models are trained on the SIN training data based after the initialization using different strategies: 1) duplicate the positive training examples; 2) do not duplicate positive training examples. The final result of DCNN for SIN is given by the average fusion of the two models.

For the no-annotation task, we design a different approach. Instead of learning the concepts with complex methods, in this method we prefer to use web images to learn simple SVM models for indexing. It would be easier to index if we manage to learn discriminative models with the data in hand, instead of using the complicated learning methods. For this purpose, we collected a set from the Bing Image Search Engine and we use it for learning. Since the web data is noisy, we only need to use the relevant images. Therefore, we use a subset of the collected set based on the ranking of the search engine, since the less relevant images are ranked low on the search engine. We used the concept names as it is since it is not allowed to extend or change the concept names. We tried to collect 1000 images for each concept, but the number images provided by the search engine differs for each concept. Therefore, if the number of images provided is less than 1000, we were collected the maximum number of images that is provided by the engine.

The collected images are described with SIFT and Opponent SIFT descriptors. Before finding the interest points and describing them, all images are down sampled to have 15000 pixels and the height to width ratio is kept the same. Then, BoW model is applied to SIFT and Opponent SIFT descriptors. A codebook with 1000 words is generated for BoW model by using 4000 frames from IACC.2.A set and applied to the frames using spatial five tiling. The resulted dimension of a feature vector for an image is 5000 for both descriptors since we apply five tiling with 1000 words on SIFT and Opponent SIFT descriptors.

We experimented on multi-class and binary class SVM classifiers with RBF kernel on IAAC.2.A set and got better results by learning with binary class SVM models. It has been observed that some images that do not show enough characteristics in terms of intensity are ranked higher. To prevent this situation we applied a simple intensity selection procedure by forcing to rank images lower that have the average intensity value lower than 20 or higher than 230.

2. Submitted Runs

We submitted 4 runs for the main task, all of which are under type A, i.e. using only TRECVID IACC training data:

- CMU_Run1: Our Safe run trains all features except the metadata by our self-paced learning pipeline. The weights in fusion are determined based on heuristic rules. For examples, for action related concepts, dense trajectory and SIFT features are average fused. This run also includes the related concept propagation [16], which proves to be beneficial in our last year's submissions.
- CMU_Run2: This run average fuses CMU_Run1 and the run generated by the DCNN pipeline. Here only 15 out of 60 concepts in DCNN run that shows improvements on the validation set are fused.
- CMU_Run3: After removing junk shots (by the junk/black frame detectors), MultiModal Pseudo Relevance Feedback (MMPRF) [12] is conducted on top of the CMU_Run2. Two modalities including the metadata and visual fusions are used in the reranking.

- CMU_Run4: This run is based on CMU_Run2. Instead of determining the fusion weights heuristically, the optimal weight of each feature for each concept is learned by grid search based on the validation dataset. Then the confidence scores of all the features are fused with the learned weights for SIN14.

Besides, we also submitted two runs for the no-annotation condition. In CMU_Run5 we used the models that are learned from the highest ranked 200 images of the search engine. To see the difference between learning from more trusty images and learning from all images in our collection we made a second submission (CMU_Run6) by using all images in our set. We were expecting to see much better results by the method that is using first 200 images comparing with the method that uses all images. However, the results of both methods are pretty similar.

3. Results

In this section, we summarize our final results returned by NIST in Table 2. Comparing CMU_Run1 with CMU_Run2, it suggests that fusing DCNN pipeline may not yield significant improvements. Comparing CMU_Run2 and CMU_Run3, we can see that MMPRF offers a relative 8.0% (1.8% absolute) infMAP improvement over the CMU_Run2. Comparing CMU_Run1 and CMU_Run4, we see that tuning parameters such as fusion weights on the validation set can also significantly improves the final results (relative 4.6% and absolute 1.1%). Our submission is ranked the top 3 teams among all submission using only IACC training data (type A). Figure 2 illustrates the comparison with other teams.

Table 2. CMU's final results on IACC.2.B for the main task.

Run ID	infMAP	infNDCG	P@10	P@100
CMU_Run1	0.2265	0.4660	0.6700	0.5583
CMU_Run2	0.2297	0.4710	0.6900	0.5683
CMU_Run3	**0.2480**	**0.4975**	**0.7000**	**0.5900**
CMU_Run4	0.2403	0.4844	0.6900	0.5730

Using the ground-truth data on IACC.2.B provided by NIST after the submission, we are able to diagnose the performance for individual features in our system. Table 3 lists the comparison results. It seems the dense trajectory feature is the best low-level feature which significantly outperforms others. However, when combined with others, it can be further greatly improved (see Table 2). For comparison, we also include ImageNet1000 features, in which the outputs of the 1000 concepts detectors trained by DCNN on ImageNet are used as the mid-level features in the SIN training [17]. Note we did not use ImageNet1000 concepts in our final submissions.

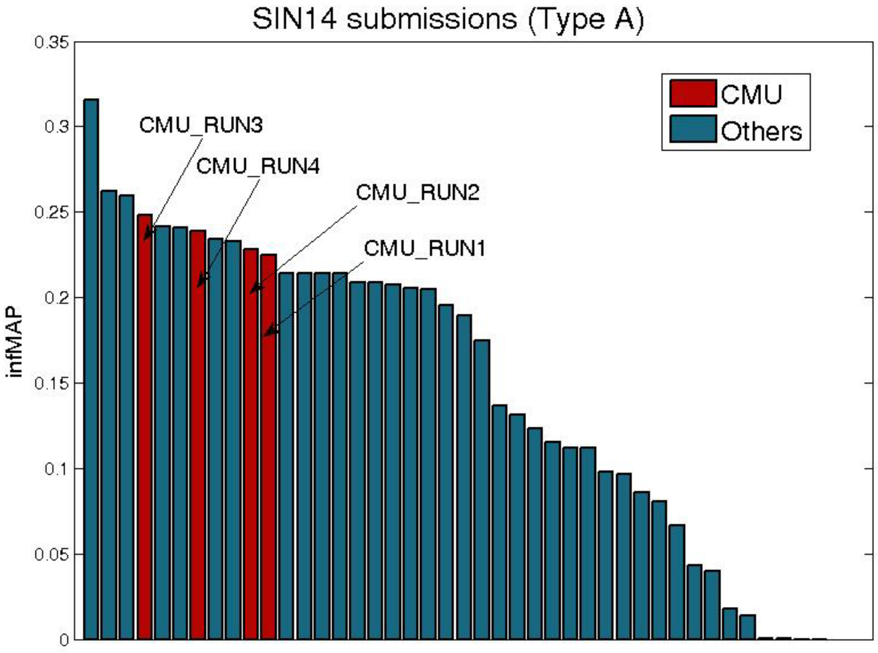


Figure 2 Comparison of CMU runs with the runs (type A) of other teams.

Using the ground-truth data on IACC.2.B provided by NIST after the submission, we are able to diagnose the performance for individual features in our system. Table 3 lists the comparison results. It seems the dense trajectory feature is the best low-level feature which significantly outperforms others. However, when combined with others, it can be further greatly improved (see Table 2). For comparison, we also include ImageNet1000 features, in which the outputs of the 1000 concepts detectors trained by DCNN on ImageNet are used as the mid-level features in the SIN training [17]. Note we did not use ImageNet1000 concepts in our final submissions.

Table 3. The performance for individual features on IACC.2.B

Run ID	Pipeline	infMAP	infNDCG	P@10	P@100
sift_harrislaplace	SVM-based	0.0866	0.2816	0.4822	0.3482
csift_harrislaplace	SVM-based	0.0842	0.2669	0.4967	0.3294
sift_multiple_keyframes_shots	SVM-based	0.0903	0.2896	0.4278	0.3326
csift_multiple_keyframes_shots	SVM-based	0.0909	0.2857	0.4422	0.3112
sift_densesampling	SVM-based	0.1096	0.3175	0.5367	0.3683
csift_densesampling	SVM-based	0.0988	0.291	0.4911	0.3686
dense_trajectory	SVM-based	0.1844	0.4001	0.6778	0.5083
DCNN pipeline	DCNN-based	0.134	0.3834	0.5111	0.4243
ImageNet1000 concepts*	SVM-based	0.0368	0.1871	0.2611	0.1904

* ImageNet1000 concepts were not used in our final submissions.

For the static image features such as SIFT/CSIFT, following [15], we compare the prediction using a single and multiple keyframes within a shot. Due to our computational constraints, 3.25 key frames are sampled for 10,7806 shot in IACC.2.B. The following table lists the comparison results. As we see, for both SIFT/CSIFT features using multiple keyframes seems to be better than using a single keyframe, though not significant. However, the precision of multiple keyframes decreases suggesting it may lose the focus for key frame of interest. This strategy also leads to 3.25 times of feature extraction and prediction time.

Table 4. Comparison of SIFT/CSIFT on single and multiple keyframes of a shot.

Run ID	infMAP	infNDCG	P@10	P@100
sift_harrislaplace_single_keyframe	0.0866	0.2816	0.4822	0.3482
sift_harrislaplace_multiple_keyframe	0.0903	0.2896	0.4278	0.3326
csift_harrislaplace_single_keyframe	0.0842	0.2669	0.4967	0.3294
csift_harrislaplace_multiple_keyframe	0.0909	0.2857	0.4422	0.3112

Table 5. CMU’s final results on IACC.2.B for the no-annotation task.

Run ID	Pipeline	infMAP	infNDCG	P@10	P@100
CMU_Run5	no-annotation	0.0118	0.1099	0.1100	0.0757
CMU_Run6	no-annotation	0.0085	0.0956	0.0967	0.0680

4. Conclusions

Based on the final results, we reached the following observations: 1) MultiModal Pseudo Relevance Feedback (MMPRF) yields a decent improvement in our final runs. 2) Self-paced concept training offers an effective and efficient pipeline for semantic concept training. 3) Dense trajectory features with fisher vector are the best low-level features, which by itself can obtain 18% infMAP. 4) Bag-of-words features on static images are weak but offer complementary information when combined with the dense trajectory; 5) SIFT/CSIFT dense-sampling seems to be better than SIFT/CSIFT harrislaplace; 6) Tuning the fusion weights on the validation set seems to beneficial.

Acknowledgments

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number OCI-1053575. Specifically, it used the Blacklight system at the Pittsburgh Supercomputing Center (PSC).

References

[1] K. Sande, T. Gevers and C. Snoek, "Evaluating color descriptors for object and scene recognition," TPAMI, 2010.

[2] H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," *in ICCV*, 2013.

[3] K. Chatfield, A. Lempitsky and A. Zisserman, "The devil is in the details: an evaluation of recent feature encoding methods," *in BMVC*, 2011.

[4] U. Niaz, and B. Merialdo. "Improving video concept detection using uploader model," in *ICME*, 2013.

[5] L. Jiang, W. Tong, D. Meng, A. Hauptmann, "Towards Efficient Learning of Optimal Spatial Bag-of-Words Representations," in *ICMR*. 2014.

[6] 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," in *TRECVID*, 2014.

[7] L. Jiang, D. Meng, S. Yu, Z. Lan, S. Shan, A. Hauptmann, "Self-paced Learning with Diversity," in *NIPS*, 2014.

[8] L. Jiang, D. Meng, T. Mitamura, A. Hauptmann, "Easy Samples First: Self-paced Reranking for Zero-Example Multimedia Search," in *MM*, 2014.

[9] M. P. Kumar, B. Packer and D. Koller, "Self-paced learning for latent variable models," in *NIPS*, 2010.

[10] Y. Bengio, J. Louradour, R. Collobert and J. Weston, "Curriculum learning," in *ICML*, 2009.

[11] S. Ayache and G. Quénot, "Video Corpus Annotation using Active Learning", in *ECIR*, 2008.

[12] L. Jiang, T. Mitamura, S. Yu, A. Hauptmann, "Zero-Example Event Search using MultiModal Pseudo Relevance Feedback, " in *ICMR*, 2014.

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z.Huang, A.Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge," arXiv:1409.0575, 2014.

[14] J. Deng, et al. "Imagenet: A large-scale hierarchical image database."Computer Vision and Pattern Recognition," in *CVPR* 2009.

[15] C. G. M. Snoek, et al. "MediaMill at TRECVID 2013: Searching concepts, objects, instances and events in video." in *TRECVID*, 2013.

[16] L. Jiang, A. Hauptmann, G. Xiang. "Leveraging High-level and Low-level Features for Multimedia Event Detection," in *MM*, 2012.

[17] Z. Z. Lan, Y. Yi, N. Ballas, S. Yu, A. Hauptmann , "Resource Constrained Multimedia Event Detection, " in *MMM*, 2014.

Informedia@TRECVID 2014 Surveillance Event Detection

Xingzhong Du[3], Yang Cai[1], Yicheng Zhao[2], Huan Li[1], Yi Yang[3] and Alexander Hauptmann[1]

Carnegie Mellon University[1] Beijing Institute of Technology[2] The University of Queensland[3]

1 Introduction

We present a generic event detection system for the SED task of TRECVID 2014. It consists of two parts: the retrospective system and the interactive system. The retrospective system uses STIP [1], MoSIFT [2] and Improved Dense Trajectory [3] as the low level features, and uses Fisher Vector encoding [4] to represent shots generated by sliding window approach. The linear SVM is in use to perform event detection. To improve the performance further, we perform several spatial schemas to generate the fisher vector in our experiments. For interactive system, we apply a general visualization schemes for all the events and a temporal locality based search method for user feedback utilization. Among the primary runs of all teams, our retrospective system ranked 1st for 3 / 7 events, in terms of actual DCR.

2 Retrospective System

2.1 Data Preprocessing

In our generic event detection system, each video is resized into 320 * 240 to accelerate the feature extraction. The resized videos are then split into shots by setting the window as 60 frames and step as 30 frames. In the experiments, we regard the shots which have 50% overlap among the annotations as the positive.

2.2 Feature Extraction and Encoding

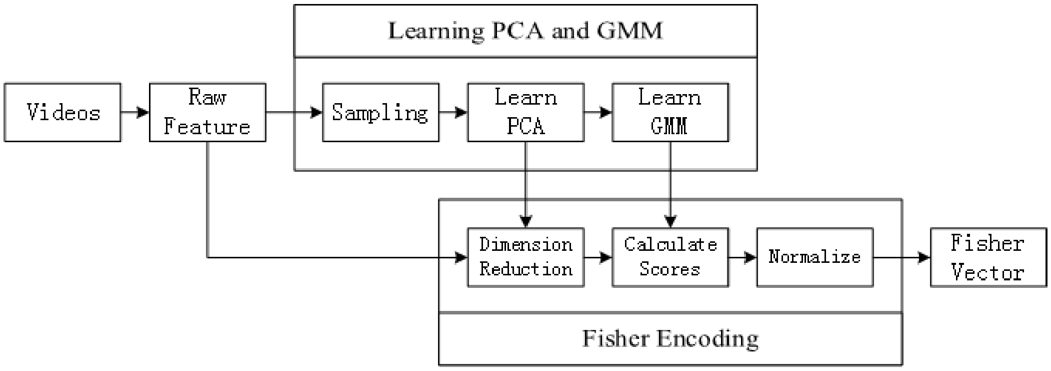


Figure 1: The pipeline of extracting fisher vector

Based on the last year's system [5], we add Improved Dense Trajectory to improve this year's performance. The feature of Improved Dense Trajectory has five parts, namely trajectory, HOG, HOF, MBHx and MBHy. Since these parts are extracted along the trajectory, they could better capture the motion information from the videos. We use PCA to reduce the dimension of each part to half. This step is important for fisher vector encoding, due to the covariance matrix becoming diagonal after PCA. Based on the transformed features, we learn GMM models with 256 Gaussians for five parts respectively. Then, each part is encoding separately by fisher vector and concatenated in the end. We also append spatial information into the fisher vector [9]. Finally, we normalize the concatenation fisher vector by power and l_2 normalization like [4]. We use 256 threads to extract the features and transform them into fisher vectors. The time cost is almost two days.

2.2 Detection

The fisher vector has very high dimension when we use Improved Dense Trajectory, thus the Linear SVM is in use to accelerate the model training and detecting. We use Liblinear [6] to perform Linear SVM. The outputs of Liblinear are distances, which are not the probabilities. Therefore, we use curve fitting method in [7] to transform the distances into probabilities. These probabilities will be used to do the non-maximum compression.

2.3 Non-maximum Suppression

The duration of events are various in different cameras. Some of the events like embrace and people meet can last for very long time. Others like cell to ear and pointing just happen in a short moment. Therefore, we do not perform the exhaustive search. Instead, we filter the shots by the thresholds we get in the cross validation and attribute the adjacent shots' labels to the shot whose confidence is the local maximum.

3 Interactive System

In this year's interactive task, we still utilize the interactive system in [8]. The time schedule on different tasks is different from the previous reports. In previous task, the time schedule is made based on the event occurrence histogram on different cameras. However, the statistical method causes one problem. That is, only several events have patterns under specific cameras, such as embrace in camera 3 and pointing in camera 1, while the others like cell to ear and person run do not have the consistent high occurrences under the specific cameras. Therefore, we make a camera-wise time schedule in the submission

4 Experiments

4.1 Model Training with Bounding Box

We think accurate temporal or spatial could improve the performances. With the annotation files, we can make the temporal information more accurate rather than sliding videos into same length. Besides that, we try to draw bounding boxes for the positive shots. So we design two experiments to verity these assumptions. The first one is use temporal information to fetch exact shots. Since Improve Dense Trajectory has the best performance when it tracks features for 15 frames, we append 15 frames after each shots to ensure that the interest features are captured. The results are shown in Table 1, in which we use IDT_FV to represent the feature extracted by Improved Dense Trajectory and encoded by Fisher Vector. The IDT_FV1 is the model trained on the shots of same length 60 frames. The IDT_FV2 is the model trained on the shots of same length as the positive annotations. The IDT_FV3 is the model trained on the IDT_FV2's features within the bound boxes. As a baseline, we also attach the results of MoSIFT Fisher Vector.

Table 1: The actual DCR and min DCR of different spatial temporal models

	MoSIFT_FV		IDT_FV1		IDT_FV2		IDT_FV3	
	aDCR	mDCR	aDCR	mDCR	aDCR	mDCR	aDCR	mDCR
PersonRuns	0.8676	0.8065	0.7835	0.7497	0.8466	0.7843	0.8655	0.8337
CellToEar	1.0090	0.9993	0.9905	0.9891	1.0075	0.9865	1.0540	0.9928
ObjectPut	1.0072	1.0001	1.0127	0.9994	1.0104	1.0005	1.0801	1.0006
PeopleMeet	0.9927	0.9652	0.9581	0.9501	0.9810	0.9710	0.9759	0.9627
PeopleSplitUp	0.9665	0.9456	0.9555	0.9324	0.9786	0.9514	1.0029	0.9779
Embrace	0.9671	0.9305	1.0218	0.9520	1.0408	0.9871	1.0321	0.9999
Pointing	1.0000	0.9955	0.9965	0.9875	1.0101	0.9972	1.0655	0.9972

The results show that only train models on the fine-tuned features cannot improve the performances. We need a similar process on the test data.

4.2 Template Bounding Box

Based on above results, we propose a template bounding box method to improve the results. The idea is that we learn the template bounding box on the training data and apply them on the test data. We apply k-means on the positions of bounding boxes and get the centroids. These centroids are used as the template positions. With several combinations of width and height, we collect the PMISS results in the cross validation. It seems that we can have a significant performance gain when the number of template bounding boxes is 5.

With the template bounding boxes, we test PersonRuns and ObjectPut under Camera 1, the results are shown in the Figure 2.

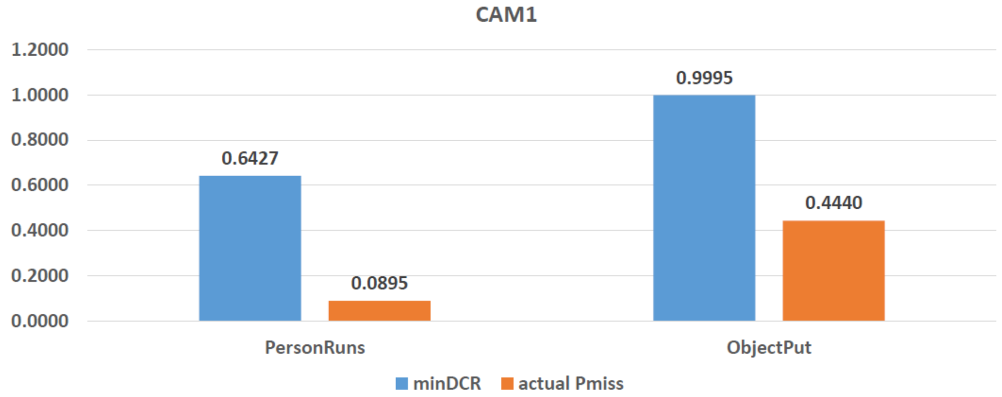


Figure 2: The preliminary results from template bounding boxes

4.3 Evaluation of the submission

In the final submission, we fuse the detection results from STIP, MoSIFT and Improved Dense Trajectory by average fusion. Due to the time cost, we do not apply template bounding box in this year’s submission. The results of retrospective event detection are shown in Table 2.

Table 2: The results in the task of retrospective event detection

	CMU14		Others Best	
	aDCR	mDCR	aDCR	mDCR
PersonRuns	0.8551	0.8500	0.8301	0.8301
CellToEar	1.0032	1.0005	0.9921	0.9911
ObjectPut	1.0023	1.0005	0.9713	0.9761
PeopleMeet	0.9008	0.8975	0.8587	0.8583
PeopleSplitUp	0.8353	0.8330	0.8698	0.8594
Embrace	0.8503	0.8462	0.8113	0.8113
Pointing	1.0035	0.9959	0.9998	0.9953

We win one event in the retrospective event detection. This is because the Improve Dense Trajectory generates a lot of positives and brings more false alarms into the detection results. We can correct these false alarms in the interactive system, and we get the following results:

Table 3: The results in the task of interactive event detection

	CMU14		Others Best	
	aDCR	mDCR	aDCR	mDCR
PersonRuns	0.7361	0.7356	0.7895	0.7895
CellToEar	1.0041	1.0009	0.9555	0.9555
ObjectPut	0.9280	0.9276	0.9641	0.9641

PeopleMeet	0.8872	0.8849	0.7960	0.7960
PeopleSplitUp	0.8115	0.8097	0.8390	0.8390
Embrace	0.8417	0.8357	0.6978	0.6978
Pointing	0.9746	0.9745	0.9744	0.9744

The results in Table 3 reveal that we can get a significant improvement after the interactive task. This is because we can effectively eliminate the performance loss from false alarm by human effort. Such improvement cannot achieve in the last year’s general interactive task, because the STIP and MoSIFT cannot detect so many positives, which lose the opportunity to decrease the PMISS.

Reference

[1] I. Laptev and T. Lindeberg, " Space-time Interest Points", in ICCV, 2013.

[2] M. Chen and A. Hauptmann, "MoSIFT: Reocgnizing Human Actions in Surveillance Videos," Carnegie Mellon University, 2009.

[3] H. Wang and C. Schmid, "Action Recognition with Improved Trajectories", in ICCV, 2013

[4] J. Perronnin and T. Mensink, "Improving the Fisher Kernel for Large-Scale Image Classification", in ECCV, 2010

[5] C. Gao, Y. Cai, H. Shen, W. Tong, Y. Yang, N. Ballas, D. Meng, Y. Yan and A. Hauptmann, "Informedia@TRECVID 2013 : Surveillance Event Detection".

[6] R. Fan, K. Chang, C. Hsieh, X. Wang and C. Lin, "LIBLINEAR : A Library for Large Linear Classification".

[7] J. Platt, "Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods", in Advances in Large Margin Classifiers, 1999

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

[9] J. Krapac, J. Verbeek and F. Jurie, "Modeling Spatial Layout with Fisher Vectors for Image Categorization", in ICCV, 2011