Invited Paper

# Strategies for Searching Video Content with Text Queries or Video Examples

Features, Semantic Detectors, Fusion, Efficient Search and Reranking

Shoou-I Yu* (student member)[†1], Yi Yang (member)[†2], Zhongwen Xu (student member)[†2], Shicheng Xu (student member)[†1], Deyu Meng (member)[†3], Zexi Mao (member)[†1], Zhigang Ma (member)[†1], Ming Lin (member)[†1], Xuanchong Li (student member)[†1], Huan Li (member)[†1], Zhenzhong Lan (student member)[†1], Lu Jiang (student member)[†1], Alexander G. Hauptmann (member)[†1], Chuang Gan (student member)[†4], Xingzhong Du (student member)[†5], Xiaojun Chang (student member)[†2]

**Abstract** The large number of user-generated videos uploaded on to the Internet everyday has led to many commercial video search engines, which mainly rely on text metadata for search. However, metadata is often lacking for user-generated videos, thus these videos are unsearchable by current search engines. Therefore, content-based video retrieval (CBVR) tackles this metadata-scarcity problem by directly analyzing the visual and audio streams of each video. CBVR encompasses multiple research topics, including low-level feature design, feature fusion, semantic detector training and video search/reranking. We present novel strategies in these topics to enhance CBVR in both accuracy and speed under different query inputs, including pure textual queries and query by video examples. Our proposed strategies have been incorporated into our submission for the TRECVID 2014 Multimedia Event Detection evaluation, where our system outperformed other submissions in both text queries and video example queries, thus demonstrating the effectiveness of our proposed approaches.

**Key words**: Content-based Video Retrieval, Motion & Image Features, Multimedia Event Detection, Multimodal Fusion, Semantic Concept Detectors, Reranking

## 1. Introduction

As we see an unprecedented growth of user-generated videos on the Internet, it is crucial to have an effective indexing and searching mechanism for these videos. To perform search, current existing video search engines mainly rely on user-generated text metadata. However, text metadata is often not a comprehensive representation of the video as: 1) users often do not provide metadata, and 2) even if users do provide metadata, a user cannot possibly annotate all facets of the video. Therefore, content-based video retrieval (CBVR), which directly analyzes the visual and audio channels of a video to perform search, has attracted the attention of many researchers and the annual TRECVID Multimedia Event Detection (MED) evaluation[1] was created. In this independent evaluation, participants design systems which utilize the wealth of information in the visual and audio channels to perform effective and efficient content-based video search for different query types, including 1) text queries and 2) query by video example.

Compared with the already mature text-based search, CBVR is significantly more challenging. One big challenge is the *low-level feature extraction* challenge. A big problem with raw visual and audio channels is that videos which depict similar semantics will still look very different if one directly compared the raw values of the two channels. To make matters worse, user-generated videos are usually very unstructured, have low resolution, severe camera motion, and very large variability. Therefore, representing videos with features which have certain invariance and generalization capabilities is a crucial part of CBVR. Another challenge is the *text/video semantic gap* challenge[2]. The main problem is that the aforementioned feature representations for video often do not contain semantic information, but to query video data with textual queries, it is crucial to bridge the semantic

Received November 30, 2015; Accepted January 6, 2016

†1 Language Technologies Institute, Carnegie Mellon University. (Pittsburgh, PA, USA.)
†2 Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney. (Sydney, NSW, Australia.)
†3 School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xian Jiaotong University. (Xi'an, China.)
†4 Institute for Interdisciplinary Information Sciences, Tsinghua University (Beijing, China.)
†5 School of Information Technology and Electrical Engineering, The University of Queensland. (Brisbane, QLD, Australia.)
* Authors are sorted in reverse alphabetical order.

gap between a pure text-based query and the non-semantic representation of a video. The final challenge is the *indexing/search* challenge. As new features are used to represent the visual and audio channels, traditional text-search techniques are not directly applicable, and new search techniques need to be developed. To enable search over large video collections, these new search techniques have to be both effective and efficient.

In light of the aforementioned challenges, we propose multiple strategies to tackle these problems. For the *low-level feature extraction* challenge, we propose two different features to significantly enhance CBVR performance. The first feature is a variant of the Improved Dense Trajectory feature[3)4)] (Section 4. 1), and the second feature is a deep learning feature (Section 4. 2) trained on ImageNet[5)6)] data. For the *text/video semantic gap* challenge, we propose a method which utilizes large amounts of weakly-labeled videos to learn semantic concept detectors encompassing a large vocabulary (Section 5). This enlarged vocabulary is crucial in bridging the semantic gap between a text-query and non-semantic video representations. For the *indexing/search* challenge, we first propose to utilize Explicit Feature Maps[7)] and Product Quantization[8)] to perform efficient yet effective video search (Section 6. 1). We then propose a novel fusion method called Multistage Hybrid Late Fusion (MHLF) to effectively fuse search results from multiple feature modalities (Section 6. 2). Finally, we propose a self-paced reranking method[9)] to automatically enhance search results through pseudo-relevance feedback (Section 6. 3). The aforementioned methods were all integrated into our TRECVID MED 2014 system, which was the leading system in all eight MED subtasks, thus demonstrating the effectiveness of our proposed strategies.

In the following sections, we first give an overview of a general CBVR system and related work in Section 2. Then we summarize our results in the TRECVID MED 2014 task in Section 3. Details of each proposed strategy are given in Sections 4, 5 and 6. Finally, Section 7 concludes the paper.

## 2. Content-based Video Retrieval Preliminaries

A general pipeline of a CBVR system is shown in Figure 1. There are mainly two phases: the offline phase and the online phase. In the offline phase, low-level and semantic features are extracted for a large video repository and indexed so that the online phase is sufficiently efficient. The semantic features are predictions of semantic concept detectors, which takes low-level features as input and predicts whether a given concept such as dog, cat, or car exists in a video. In the online phase, users will provide different types of queries to search for relevant videos. There are mainly two types:

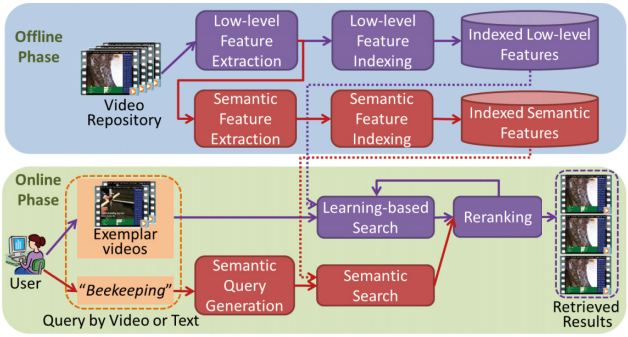



**Fig. 1** Pipeline of a general CBVR system. The purple boxes and arrows correspond to components designed for querying by video examples. The red boxes and arrows corresponds to components designed for text queries.

(1) Query by video example: The user provides one or multiple example videos to search for related videos.
(2) Text queries: The user types in a pure text query to search for videos of interest.

As the input from different query types are of different modalities (i.e. videos or text), different kinds of features and search techniques are designed for each case. For the query by video examples case, the learning-based search component retrieves related videos by first training a model which distinguishes the exemplar videos from the non-related videos. The model is trained based on the features extracted in the offline phase. Then, the model is applied to the video repository to search for other related videos. Lastly, the search results goes through an iterative reranking process, which performs pseudo-relevance feedback to automatically improve the search results. For searching by text queries, the first step is semantic query generation, where the text query is mapped to the system vocabulary. The system vocabulary constitutes of all concepts that could be detected by the available semantic concept detectors. Then the generated semantic query is utilized to perform semantic search. The initial ranked list also goes through the reranking process to acquire a more accurate ranked list. In the following sections, we will explain the details and also briefly review the related work for each component.

### 2. 1 Searching by Video Examples

For the query by video examples scenario, low-level features combined with discriminatively learned search models play the key role in achieving good performance[10)11)]. In the following sections, we will review the related work on these two topics.

(1) Low-level Features

Many low-level audio and visual features have been utilized to enhance CBVR performance. The most popular low-level audio feature used is the Mel-Frequency Cepstral

Coefficients (MFCC)[11], which have been shown to be the most cost-effective feature[12]. Other audio features including Acoustic Unit Descriptors[13] and Large-scale Pooling Features[14] have also been utilized.

Low-level visual features can be split into two categories: *static image features* and *motion features*. Static image features are essentially image-based features extracted from all or selected frames of a video. The temporal relation between the frames are not taken into account. Before the introduction and success of deep features, mainstream static image features were frequently hand-crafted SIFT-based features[15][16]. Currently, deep convolutional neural network (DCNN) features[11][17] are significantly outperforming hand-crafted features and represent the current mainstream. In this paper, we present architectural improvements for static-image deep-networks to enhance CBVR performance.

Motion features utilize the temporal relations between frames to capture motion characteristics of a video. Optical flow is typically used to compute motion features. Currently, one of the most effective features is Improved Dense Trajectories (IDT)[18], which significantly outperforms the previously proposed popular motion features such as Space Time Interest Points (STIP)[19] and Motion SIFT (MoSIFT)[20]. In this paper, we present two enhancements which further improves the performance of IDT.

One problem with the previously mentioned low-level features is that they will generate a different number of feature vectors depending on the length, resolution and contents of the video, thus leading to varying length vector representations for each video. It is very difficult to compare two videos with different length representations. Therefore, the varying length representations of each video need to be converted to a fixed-length vector representation, thus many different encoding/pooling techniques have been proposed, including Bag-of-Words (BoW)[21] and Spatial Pyramid BoW (SpBoW)[22], Fisher Vectors (FV)[23][24], and Vector of Locally Aggregated Descriptors (VLAD)[25]. FV and VLAD are the current mainstream encoding methods[17].

Overall, a "complete" feature is a combination of a low-level feature and an encoding method. For example, SIFT can be encoded with SIFT-SpBoW or SIFT-FV, and MFCC can also be encoded with BoW (MFCC-BoW) or FV (MFCC-FV) respectively. Once these encodings have been computed, they are indexed for the subsequent learning-based search.

(2) Learning-based Search

There are two key components to learning-based search: the learning component, and the fusion component.

Utilizing machine learning models such as Support Vector Machines (SVM)[10][26] and Kernel Ridge Regression (KRR)[11] has shown to be very effective for querying with video examples. The main idea is to treat the example videos as positive training data, and when combined with a large pool of negative videos, a classifier can be trained to determine whether an input testing video is relevant or not.

Fusion enables the incorporation of search results from different features which capture the multiple aspects of a video. The main challenge of fusion is to effectively estimate the reliability of each feature source so that the fusion algorithm knows which features to rely more on when dealing with different videos. Many fusion method such as early fusion, late fusion, double fusion[27] and other more complex methods[28] have been proposed. In this paper, we propose a Multistage Hybrid Late Fusion method, which shows superior performance and robustness over other fusion methods.

(3) Reranking

Reranking utilizes pseudo-relevance feedback (PRF) to automatically enhance an initial rank list. The intuition of PRF is that the top-ranked results in an initial rank list are highly likely to be correct, and adding these instances back into the training set may improve performance. This simple method has shown to be effective in many different scenarios. However, previous PRF methods usually operate on a single ranked list[29], but the CBVR task inherently outputs multiple ranked lists from different features, and effectively fusing these ranked lists becomes a challenging task[30].

In this paper, we introduce self-paced reranking, which further improves the performance of existing PRF approaches. Our system incorporates MMPRF[30] and SPaR[9] to conduct 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 by Jiang et al.[9], in that the model is trained iteratively as opposed to simultaneously. Our methods are able to leverage high-level and low-level features which generally leads to increased performance[31]. The high-level features used are ASR, OCR, and semantic visual concepts. The low-level features include DCNN, IDT and MFCC features.

### 2.2 Searching with Text Queries

This scenario takes a pure-text query as input, and outputs a ranked list of relevant videos. It is an interesting task because it resembles a real-world video search scenario, where users typically search videos by using query words instead of providing example videos.

The main challenge of the text-to-video search scenario is to bridge the semantic gap between text and video. In current state-of-the-art systems, this gap is usually bridged

with automatic speech recognition (ASR), optical character recognition (OCR), and semantic concept detectors. Semantic concept detectors are trained to detect whether a certain object, scene, or action exists in a video or not. Given a pool of concept detectors, these detectors can be applied on an input video to acquire a semantic feature representation of the video, which corresponds to the confidence score of detecting a concept in the video. This feature representation is very different from the low-level feature representations, where each dimension in the vector does not have a clear semantic meaning. Popular datasets to train concept detectors include the ImageNET[5] dataset, the SUN397[32] scene dataset, and the TRECVID Semantic Indexing (SIN)[1] dataset. To train effective static-image-based detectors, the current mainstream is deep convolutional neural network models[6][33]. To train video-based detectors, the current mainstream is combining deep static-image detectors with motion features such as Improved Dense Trajectories[18].

According to Jiang et al.[2][30], a text-to-video search system consists of three major components, namely Semantic Query Generation (SQG), Semantic Search and Reranking/PRF as shown in Figure 1. The Semantic Query Generation component translates the description of the user's information need into a set of multimodal system queries that can be processed by the system. There are two challenges in this step. Since the semantic vocabulary of the system is usually limited, the first challenge is to map the user's query words into the system vocabulary. The second challenge is assigning a given query word its modality as well as its weight associated with that modality. A preliminary study of these challenges is detailed in Jiang et al.[2].

The semantic search component retrieves multiple ranked lists for a given text query. Our system incorporates various retrieval methods such as the Vector Space Model, tf-idf, BM25, language model[34], etc. Surprisingly, a better retrieval model on worse features actually outperforms a worse retrieval model on better features. This observation suggests that the role of retrieval models in our semantic search system may be underestimated in much current research. 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.

Reranking is also performed for text query search. One key advantage of reranking is that it "bridges" the semantic search and the learning-based search[30]. Once the text query search component generates an initial ranked list, the positives in this ranked list can be used to perform learning-based search, which can often further improve search performance.

## 3. TRECVID Multimedia Event Detection 2014

The TREC Video Multimedia Event Detection (MED)[1] task is a standardized task held every year since 2010 to evaluate the performance of different CBVR systems on the MED task. Different CBVR systems are presented with multiple queries, and the CBVR system needs to retrieve relevant videos from the evaluation set. Example queries are shown in Figure 2. The query consists of two parts, the textual query and the video examples. Different parts of the query will be utilized for different query settings as described below. For the TRECVID MED14 task, the organizers split the data set into four standard sets: the positive examples, the background set, the validation set and the testing set. The positive examples are videos relevant to a queried event. The number of positive examples will vary according to the four different query settings defined by the organizers:

( 1 ) Semantic Query (SQ): Only the textual query was given.
( 2 ) 0 Exemplar (000Ex): The textual query and background videos were given. No positive videos were given.
( 3 ) 10 Exemplar (010Ex): In addition to what was given for 000Ex, 10 positive videos were given.
( 4 ) 100 Exemplar (100Ex): In addition to what was given for 000Ex, 100 positive videos were given.

The background set, which can be viewed as the negative videos, contained 4992 videos. The validation set, which is also know as MEDTEST14, contained around 24,000 videos. The testing set contained around 198,000 videos (8000 hours of video) which does not contain any text metadata. In the competition, competitors trained their system on the positive and background sets, and then tuned the system on the validation set. Finally, the resulting system performed search over the testing set and the results were submitted to the organizers. Label information was only available for the positive, background and validation sets. The labels for the testing set were never released to prevent overfitting on the testing set. In the competition, there were two types of queries, pre-specified and ad-hoc. For pre-specified queries, the names of events were given a few months beforehand, so participants could design specialized detectors for these events. On the other hand, ad-hoc queries were given a few days before the deadline, leaving no time to design specialized detectors. There were a total of 20 events/queries for the MED14 pre-specified run and 10 events for the MED14 ad-hoc run. The evaluation metric used was Mean Average Precision (MAP)[1].

**CMU MED14 Submission Overview**

For MED14, we had a system[11] for text queries (SQ, 000Ex) and another system for query by video examples (010Ex, 100Ex). For text queries, our system utilized ASR,

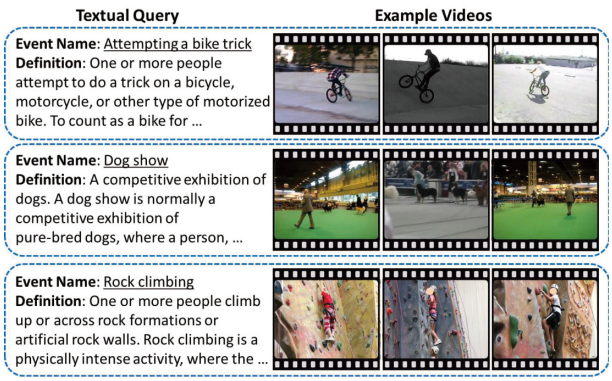



**Fig. 2** Examples queries from the MED14 task. Each query consists of a textual description and video examples.

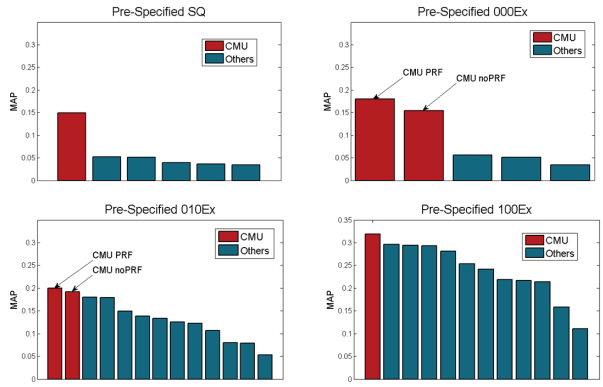



**Fig. 3** Official[1] MAP performance on the MED14 testing set in different settings for pre-specified events.

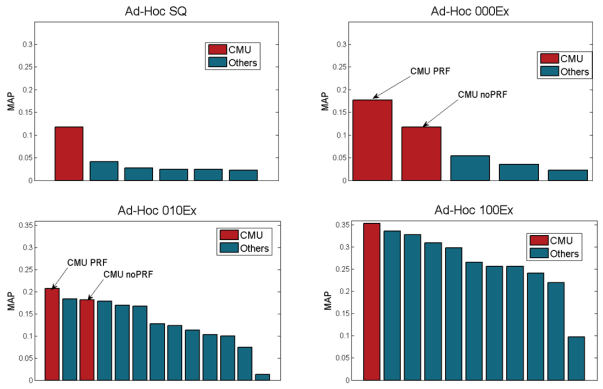



**Fig. 4** Official[1] MAP performance on the MED14 testing set in different settings for ad-hoc events.

| Metric: MAP | 100Ex | 010Ex |
|---|---|---|
| IDT | 0.274 | 0.133 |
| IDT + SPM | 0.286 | 0.136 |
| MIFS (L=0,2,5) | 0.297 | 0.153 |
| MIFS (L=0,2,5) + STED | **0.298** | **0.162** |

**Table 1** Performance comparison of MIFS and STED over baseline methods.

OCR, and more than 3000 concept detectors (Section 5). The semantic query generation utilized WordNet similarity[35], Point-wise Mutual Information on Wikipedia, and word2vec[35)36)] to generate a mapping that maps the textual event description to the concepts in our vocabulary. For semantic event search[2], our system incorporated various retrieval methods such as Vector Space Model, tf-idf, BM25, language model[34], etc. For query by video examples, our system extracted 47 low-level and semantic features, which were all provided to the learning-based search component. The search components utilized two classifiers: SVM and KRR. For 100Ex, both SVM and KRR were used. However, for 010Ex, experiments had shown that only using KRR achieves better performance, so the 010Ex runs only utilized prediction results from KRR. The output of the 47 features from the classifiers were given to Multistage Hybrid Late Fusion to acquire the final fusion results. More details of the 47 features are in Yu et al.[11].

Figures 3 and 4 present the results of our system and other competing systems on the MED14 task. We can see that our system is significantly better than other competing systems, thus demonstrating the effectiveness of our strategies. In the following sections, we will detail each of our strategies. However, as the labels for the testing set were never released, we can only present experimental results on the 20 pre-specified events on the validation set MEDTEST14.

## 4. Improvements in Low-Level Features

### 4.1 Enhancements for Improved Dense Trajectories

We improve the original Improved Dense Trajectory[18] in two ways. First, temporal scale-invariance is achieved by extracting features under different video playback speeds, which are generated by skipping frames at certain intervals. We denote this new way of feature extraction as Multi-skIp Feature Stacking (MIFS)[4]. Different from what has been described in Lan et al.[4], we use the combination of level 0, 2 and 5 to balance speed and performance.

Second, we propose a new space-time encoding method, dubbed Space-Time Extended Descriptors (STED), that attaches spatial $(x, y)$ and temporal $(t)$ location information to the raw features after PCA-projection[4].

As illustrated in Table 1, by using MIFS, we improve MAP of both 100Ex and 010Ex on MEDTEST14 by about 2%, absolute. We further add STED to the results of MIFS and compared it with Spatial Pyramid Matching (SPM)[22], a classical space-time encoding method. As can be seen, STED can get similar or better results compared to the results of only using MIFS. SPM can also improve the baseline results, but due to its high dimensionality, it needs large space for storing

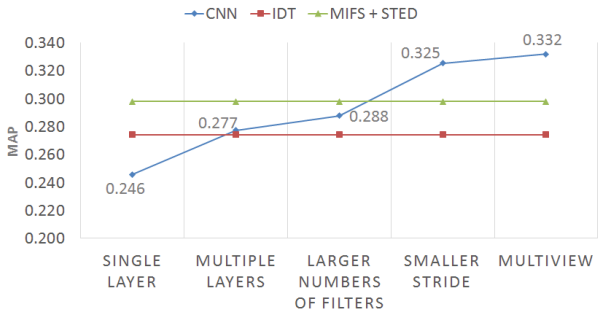



**Fig. 5** Performance gains on MEDTEST14 100Ex after CNN structure modifications.

the resulting feature vectors and is computationally expensive to run the classifiers, thus STED is a more space efficient alternative to incorporate spatial and temporal information into a feature. For details, please see Lan et al.[4].

### 4.2 Features from ImageNet DCNN Models

In order to leverage the powerful deep learning models in MED, we improved existing DCNN models in two directions: 1) by utilizing more data and 2) by modifying the network structure. In total, we have extracted a total of 15 different Deep Convolutional Neural Network (DCNN) features in our MED14 system. All models were trained on different subsets of ImageNet.

We utilized more data by training 3 models[12] on the whole ImageNet dataset consisting of around 14 million labeled images and 28,000 classes. We took the networks at epoch 5, 6 and 7 and generated features for MED keyframes using the first fully connected layer and probability layer. To generate video features from keyframe-level features, we used both maximum pooling and average pooling for the probability layer and only average pooling for the fully connected layer. This procedure results in 9 DCNN-ImageNet representations for each video.

To explore the performance of deep models under varying network structures, another 5 models were trained on the standard ILSVRC 2012 dataset[5] which had around 1.28 million images belonging to 1,000 classes. Two models were trained with six convolutional layers, two models were trained with smaller filters, and one was trained with a larger number of filters. A multi-view representation was used for one of the models. The network structure is as described in Zeiler et al.[37]. Except for different structures among models, the models with the same structures differ in initialization. The training process was tuned on the ImageNet ILSVRC 2012 validation set with 50 thousand images. These models result in another 6 different feature representations. More details and also some further improvements after the MED14 evaluation are described in[17].

Figure 5 illustrates the improvements on different network structures we have explored within the ILSVRC 2012 training setting. We started with the standard AlexNet but with 6 convolutional layers and the features were computed from the 1,000 dimensional probability layer. Intuitively, the probability output provides a semantic feature representation for each video, where each dimension corresponds to a specific object. We can regard this feature as Bag-of-Words with a vocabulary of 1,000 visual objects. The features were then fed into a $\chi^2$-exponential SVM for classification. We only achieved 0.246 MAP on MEDTEST14 100Ex, which is far below IDT. We then explored features from other layers, e.g., $\text{pool}_5$, $\text{fc}_6$ and $\text{fc}_7$. Adding multiple layers into the video representation increases the MAP to 0.277.

We further explored a wider network by doubling the number of filters in each convolutional layer. For example, the standard AlexNet had 256 filters in the 5-th convolutional layer, while we explored the 5-th convolutional layer with 512 filters. This way, the network learns more complex patterns in the images and improved the MAP to 0.288. Following Zeiler et al.[37], we made the filter size of the first convolutional layer smaller, i.e., reducing it from 11 to 4, and decreased the stride of this layer from 4 to 2. Though this dramatically increased training time for the network due to much more time-consuming convolutional operations on the first convolutional layer, the smaller filter size and stride helped the network capture finer patterns and boosted performance to 0.325, which outperforms the previous versions significantly. In the stages described above, we only utilized a single crop from the central 224-by-224 pixels of the video frames, which may lose some helpful visual information. Therefore, we generated a multi-view DCNN feature by producing 10 crops per input frame, which included the top-left, top-right, bottom-left, bottom-right and center crops along with their corresponding mirrored crops. The features obtained from the 10 views are subsequently averaged together to acquire a single vector representation. This further improved performance to 0.332. The whole exploration of utilizing features extracted from ImageNet pretrained models with different structures raised performance from 0.246 to 0.332, which is a big improvement over state-of-the-art hand-crafted features.

## 5. Bridging the Text/Video Semantic Gap

Our shot-based semantic concepts were directly trained on video shots and not still images for the following two reasons: 1) shot-based concepts have minimal domain difference; 2) this allows for action detection. We have found that detectors trained on still images usually do not work well on video, which may suggest that the domain difference between static

images and video data such as MED data is significant.

The shot-based semantic concept detectors were trained with our pipeline based on our previous study on CascadeSVM and a new study on self-paced learning[38)39)]. Our system included more than 3,000 shot-based concept detectors which were trained on around 2.7 million shots using the standard improved dense trajectory features[18)]. The detectors are generic and include people, scenes, activities, sports, and fine-grained actions described in[40)]. The detectors were trained on several datasets including Semantic Indexing[1)], YFCC100M[41)] and Google Sports[42)]. YFCC100M and Google Sports are weakly-labeled datasets, i.e. the labels for each video were inferred from the metadata of the videos and not annotated by a human. The notable increase in quantity and quality of our detectors significantly contributed to the improvement in the text-to-video system performance.

Training large-scale concept detectors on big data is very challenging, thus requiring research on both theoretical and practical perspectives. Regarding theoretical progress, we adapted self-paced learning theory, which provided theoretical justification for concept training. Self-paced learning is inspired by the learning process of humans and animals[39)43)], where samples were not learned randomly but organized in a meaningful order: from easier samples to gradually more complex ones. We advanced the theory in two directions: augmenting the learning schemes[9)] and learning from easy and diverse samples[38)]. The two studies offer a theoretical foundation for our detector training system.

As for practical progress, we optimized our pipeline for high-dimensional features (around 100K 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[38)]. We use around 768 cores in Pittsburgh Supercomputing Center for about 5 weeks, which could be roughly broken down into two parts: low-level feature extraction for 3 weeks and concept training for 2 weeks. For testing, we converted our models to linear models to achieve around 1,000 times speedup in prediction.

In summary, our theoretical and practical progress provided the foundation for developing critical tools for large-scale concepts training on big data. For instance, if we had 500 concepts over 0.5 million shots, then, optimistically speaking, we can finish 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.

## 6. Improvements in Indexing/Retrieval

### 6.1 Efficient Learning-based Search

The most natural way for a human to utilize a system is through an interactive process. Therefore, to strive for interactive MED, we targeted completing learning-based search over 200,000 videos in 15 minutes on a single machine. This is a big challenge for the query by video example pipeline, as we utilized 47 features and around 100 classifiers (SVM & KRR) to create the final ranked list. The text search pipeline is a lot simpler thus timing is not a big issue. Therefore, we will focus on the query by video example system in the remaining section. To speed up, we performed optimizations in three different directions: 1) decreasing computational requirements, 2) decreasing I/O requirements and 3) utilizing GPUs. Computational requirements were decreased by replacing kernel classifiers with linear classifiers. I/O requirements were decreased by compressing features vectors with Product Quantization[8)] (PQ). GPUs were utilized for fast linear regression and prediction.

(1) Replacing Kernel Classifiers by Linear Classifiers

Kernel classifiers are slow during prediction time because to perform prediction on a testing video vector, it is often required to compute the dot-product between the testing video feature and each vector in the training set. For MED14, we had around 5000 training videos, so 5000 dot products were required to predict one video. This is too slow, as preliminary experiments showed that prediction of improved trajectory fisher vectors (IDT-FV, 109056 dimensions) on 200,000 videos required 50 minutes on a NVIDIA K-20 GPU. Therefore, to accelerate this process, we switched to linear classifiers, which requires only one dot product per testing vector, thus in theory we have sped up by 5000x. However, bag-of-word features do not perform well with linear kernels. Therefore, we used the Explicit Feature Map (EFM)[7)] to map all bag-of-word features 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. Figures 6 and 7 show the performance difference before and after EFM approximations. For most features, we suffer a slight drop in performance, which is still cost-effective given that prediction speed was sped up by 5000x.

(2) Feature Compression with Quantization

In order to improve I/O performance, we compressed our features using Product Quantization[8)] (PQ). Compression is crucial because reading uncompressed features can take a lot of time. PQ compresses feature vectors by first splitting each feature vector into multiple chunks, and then quantizing each chunk with a 256 word codebook. A 256 word codebook

is ideal because cluster assignments can be stored with 1 byte. Therefore, the chunk, which we set to 8 floating point numbers (32 bytes) in our system, is simply represented by 1 byte, thus achieving 32X compression. Also, faster classifier prediction can be done based on the PQ codebooks. More details are in Jegou et al.[8] and Yu et al.[44]. However, as PQ performs lossy compression, the quality of the final ranked list may degrade. Figures 6 and 7 shows the performance drop before and after PQ approximation. We can see that there is nearly no performance drop before and after PQ. Figure 8 further shows MEDTEST14 010Ex performance when performing quantization under different compression ratios. We show performance of two quantization methods, PQ and Uniform Quantization (UQ). The basic idea of UQ is to quantize each dimension of all feature vectors into $k$ bins, and each dimension can be represented with $\log_2(k)$ bits. As we can see, PQ and UQ have similar performance. The problem with UQ is that one can at most achieve 32X compression when $k = 2$, but PQ can achieve higher compression ratios by adjusting the size of each chunk.

(3) Utilizing GPUs for Fast Linear Regression* and Linear Classifier Prediction

Following the TRECVID MED 2014 guidelines, we were limited to a single workstation for learning-based search. Therefore, we utilized all available computing resources on our 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. Our workstation had 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.

(4) Overall Speed Improvements

As both EFM and PQ are approximations, we quantified the drop in performance when both methods were used. The results are shown in Table 2. 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. We have sped up our system by 16 times for learning-based search with a cost of 3% relative drop in performance, which is negligible given the large efficiency gain.

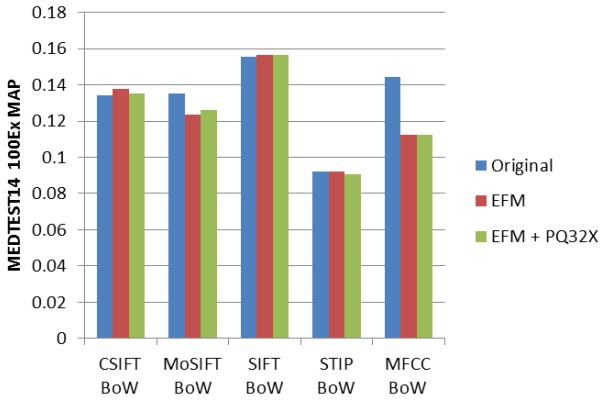



**Fig. 6** Performance difference before and after EFM and PQ approximations for MEDTEST14 100Ex.

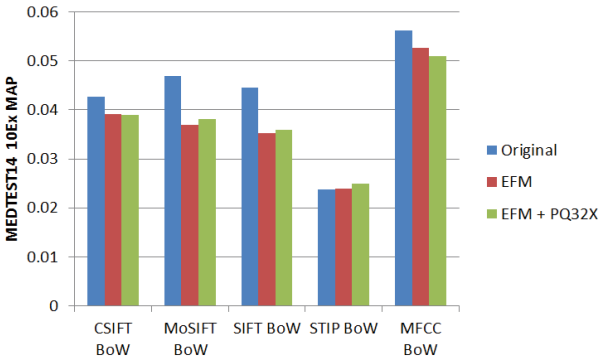



**Fig. 7** Performance difference before and after EFM and PQ approximations for MEDTEST14 010Ex.

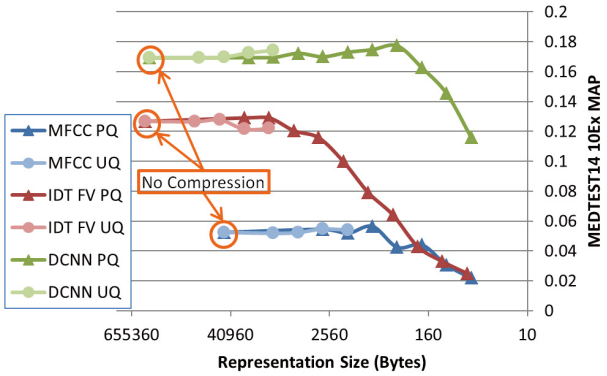



**Fig. 8** Performance on MEDTEST14 010Ex under different quantization methods and compression ratios.

| | MEDTEST14 MAP | | 100Ex Learning-based Search Timing (s) |
|---|---|---|---|
| | 100Ex | 010Ex | |
| No EFM, No PQ, with GMM features$^{\alpha}$ | 0.405 | 0.266 | 17580$^{\beta}$ |
| EFM, PQ, no GMM features | 0.394$^{\gamma}$ | 0.270 | 1068 |
| Relative Improvement | -2.7% | 1.5% | 1646% |

**Table 2** Performance of different features and fusion methods.

*For linear features, the KRR model effectively becomes linear regression.

$\alpha$ : Used in our MED13 system[27].

$\beta$ : Extrapolated timing for our MED13 system[27].

$\gamma$ : A modified MHLF was used so that it is compatible with features of the MED13 system, thus leading to slightly different numbers than Table 3.

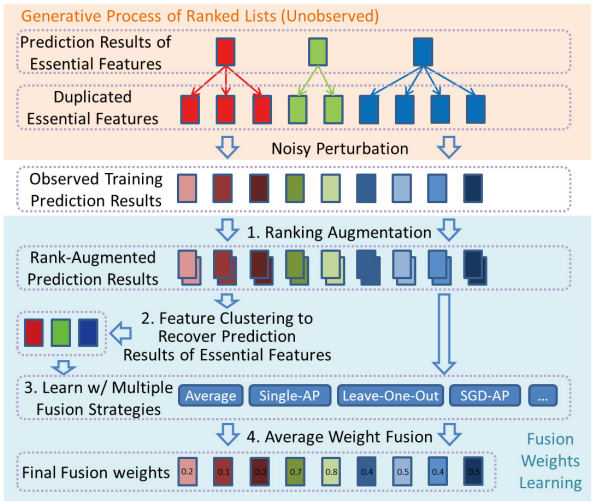



**Fig. 9** Intuition and pipeline of Multistage Hybrid Late Fusion.

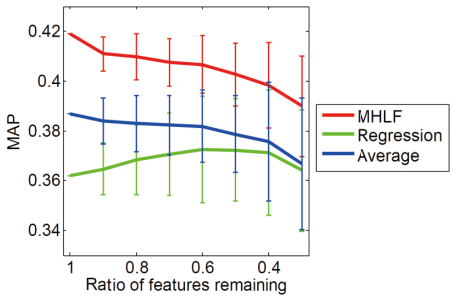



**Fig. 10** Results of fusion based on randomly sampled subsets of features, where the number of randomly sampled features varies from all features to 30% of the features. The sampling was repeated 60 times, and the 95% confidence intervals are shown.

### 6.2 Multistage Hybrid Late Fusion Method

For a given query, the goal of fusion is to learn the weights of different modalities according to the effectiveness of each feature. A simple way to learn modality specific weights can be by training a linear regression model on held-out data from the training set. However, this approach is usually not as stable as if the held-out set is small, the learned weights tend to overfit. To this end, we propose a new learning based late fusion algorithm, named the "Multistage Hybrid Late Fusion" (MHLF) as shown in Figure 9. The MHLF is designed based on the following three key observations:

**1. Ranking information is not explicitly modeled in the prediction scores.** Therefore, step 1 in Figure 9 augments the original prediction scores with ranking information.

**2. Prediction scores from different features contain duplicate information and should not be naïvely averaged.** Duplicate information comes from different features using the same basic feature. For example, SIFT-BoW and CSIFT-FV are all SIFT-based and their ranked lists are usually highly correlated. We propose to model such highly correlated ranked-lists as a generative process. The assumption is that there are many "essential features", whose classifiers generate noise free ranked lists. However, these essential features goes through a duplication and noisy perturbation process, thus what we observe are noisy prediction results. Therefore, to recover the essential features, we perform PCA-Tree clustering as shown in step 2 of Figure 9. The cluster centers corresponds to a "cleaner" version of the prediction results and can be viewed as an estimate of an essential feature. These recovered essential features, and also the original prediction scores are all provided to the next hybrid fusion step.

**3. Prediction scores contain random noise and directly learning fusion weights on top may lead to overfitting.** To deal with this issue, MHLF utilizes hybrid strategies to acquire a more robust fusion weight estimate. The intuition is that each fusion strategy can be viewed as a random observation of a "ground-truth fusion strategy". Since there is no single fusion strategy that performs better than others on all queries, sampling multiple strategies and averaging them is a simple and effective method to acquire a more stable estimate of fusion weights. The key fusion strategies include:

(1) Average fusion: each feature gets equal weight.
(2) Single-AP: the weights of each feature is its average precision (AP) on the held-out set.
(3) Leave-One-Out: the weights of a feature is the AP performance drop when removing this feature from an average fusion run.
(4) SGD-AP: performs stochastic gradient descent which maximizes average precision as the loss function.

Results on the key features of MEDTEST14 and final fusion results are shown in Table 3. All these results were based on prediction scores from 32X PQ. As we can see, MHLF is superior than average fusion and linear regression fusion. We also performed robustness tests on our fusion algorithm as shown in Figure 10. In this experiment, we randomly removed a subset of features from the original set of 47 features and ran the different fusion methods. As we can see, as we gradually remove features, MHLF is still consistently better than the other two baseline fusion algorithms, thus demonstrating the robustness of MHLF.

From Table 3, we can also compare the relative performance of single features and encoding methods. For static image features, it is clear that multi-view DCNN outperforms handcrafted features such as SIFT and CSIFT. For motion features, MIFS + STED is significantly better than MoSIFT and STIP. Encoding wise, we can see that Fisher vector (FV) encodings are in general better than BoW encodings. Also, in general KRR performs better than SVM in the 010Ex scenario, so for the 010Ex fusion runs we only used pre-

| Condition | 10Ex | | 100Ex | |
|---|---|---|---|---|
| Classifier | KRR | SVM | KRR | SVM |
| STIP SpBoW EFM | 0.026 | 0.025 | 0.087 | 0.091 |
| SIFT SpBoW EFM | 0.042 | 0.036 | 0.145 | 0.157 |
| MoSIFT SpBoW EFM | 0.045 | 0.038 | 0.110 | 0.126 |
| CSIFT SpBoW EFM | 0.046 | 0.039 | 0.143 | 0.135 |
| MFCC BoW EFM | 0.057 | 0.051 | 0.101 | 0.112 |
| CSIFT FV | 0.065 | 0.051 | 0.157 | 0.140 |
| SIFT FV | 0.066 | 0.060 | 0.162 | 0.157 |
| STIP FV | 0.073 | 0.074 | 0.140 | 0.140 |
| MoSIFT FV | 0.081 | 0.083 | 0.179 | 0.184 |
| IDT FV | 0.135 | 0.128 | 0.270 | 0.268 |
| MIFS + STED | 0.161 | 0.142 | 0.292 | 0.277 |
| Multi-view DCNN | 0.187 | 0.167 | 0.319 | 0.299 |
| MHLF, MIFS + STED & multi-view DCNN | 0.215 | | 0.353 | |
| MHLF, MIFS + STED & multi-view DCNN & MFCC BoW | 0.237 | | 0.389 | |
| Linear Regression Fusion, 47 Features | 0.250 | | 0.362 | |
| Average Fusion, 47 Features | 0.252 | | 0.387 | |
| MHLF, 47 Features | **0.285** | | **0.419** | |

**Table 3** MAP performance of different features and fusion methods. The testing features have all gone through 32X PQ compression, so the results are slightly lower than the non-approximated results reported in Table 1 and Figure 5.

diction results from KRR. Finally, if there were resource constraints in feature extraction, combining the 3 core features: MFCC BoW, MIFS + STED, and multi-view DCNN can achieve around 90% of the full system's performance. Among these 3 core features, MFCC excels on events such as "Tuning musical instrument" and "Town hall meeting", where audio such as instrument sounds or speech is an important cue. MIFS + STED performs well on events such as "Rock climbing" and "Winning a race without a vehicle", where the action of people is crucial in determining if a video is relevant. Finally, multi-view DCNN achieves high performance on events which have discriminative objects, such as honeycomb for the "Beekeeping" event, and cars in the "Parking a vehicle" event.

### 6.3 Self-Paced Reranking

Our PRF system was implemented according to Self-Paced Reranking (SPaR) detailed in Jiang et al.[9]. 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 was used, since we empirically found it outperforms the binary self-paced function on the validation set[2]. Since the starting values can significantly affect final performance, we used the reasonable starting values generated by MMPRF[30]. The high-level features used were ASR, OCR, and semantic visual concepts. The low-level features were DCNN, IDT and MFCC features. We did not run PRF for SQ since our 000Ex and SQ runs are very similar. The final results were computed by averaging the initial ranked list with the reranked list. This is beneficial because for the 000Ex case, the initial ranked list is from semantic search (high-level features), whereas the reranked list is from learning-based search (low-level features), and leveraging high-level and low-level features usually yields better performance[31]. To be prudent, the number of iterations is no more than 2 in our final submissions.

The contribution of our reranking methods is evident because the reranking method is the only difference between our noPRF runs and PRF runs as shown in Figure 3 and 4. According to the MAP on the testing set of MED14, our reranking method boosted the MAP of the 000Ex system by a relative 16.8% for pre-specified events and a relative 51.2% for ad-hoc events. Besides, it also boosted 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 previous work[9][30]. Note that the ad-hoc queries are very challenging because the query is unknown to the system beforehand. As we can see, our reranking methods still managed to yield significant improvement on ad-hoc events. More reranking results on MEDTEST14 data can be found in Jiang et al.[2].

It is interesting that our 000Ex system for ad-hoc events outperforms 010Ex systems from many other teams. In MED14, the difference between the best 000Ex with PRF (17.7%) and the best 010Ex noPRF (18.2%) is marginal. In MED13, however, this difference was very large where the best 000Ex and 010Ex system was 10.1% and 21.2%* respectively. This observation suggests that the gap of real-world 000Ex event search system is shrinking rapidly. We attribute the improvement of the 000Ex system to the following key reasons: 1) improved semantic concept detectors (Section 5), 2) improvement achieved by the reranking algorithm SPaR, and 3) reasonable queries formulated by human experts.

## 7. Conclusion and Future Work

We have described multiple strategies to enhance both the accuracy and speed of content-based video retrieval systems. Overall, the main conclusions are: 1) IDT-based and CNN-

*The runs in different years are not comparable since different queries were used.

based features are the current best motion and static image feature, 2) semantic concept detectors trained from big data are effective, 3) EFM and PQ compression can significantly speed up the system with only a negligible drop in accuracy, 4) MHLF fusion, which fuses multiple fusion strategies, is robust, and 5) reranking is an effective way to enhance accuracy. Looking into the future, we believe that current systems can already achieve reasonable accuracy, but speed is still a big issue. Efficiently extracting features, indexing and searching the billions of videos online will be the next big challenge.

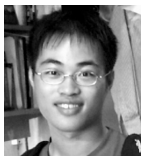
## Acknowledgments

This work was partially supported by the US Department of Defense the U. S. Army Research Office (W911NF-13-1-0277), National Science Foundation under Grant Number IIS-12511827 and 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).

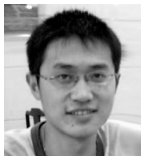
## References

1) P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Smeaton, and G. Quénot, "TRECVID 2014 – An Overview of the Goals, Tasks, Data, Evaluation Mechanisms and Metrics," in *TRECVID 2014* (2014).
2) L. Jiang, S.-I. Yu, D. Meng, T. Mitamura, and A. G. Hauptmann, "Bridging the Ultimate Semantic Gap: A Semantic Search Engine for Internet Videos," in *International Conference on Multimedia Retrieval (ICMR)* (2015).
3) H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," in *International Conference on Computer Vision* (2013).
4) Z. Lan, M. Lin, X. Li, A. G. Hauptmann, and B. Raj, "Beyond Gaussian Pyramid: Multi-skip Feature Stacking for Action Recognition," in *Computer Vision and Pattern Recognition (CVPR)* (2015).
5) J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *Computer Vision and Pattern Recognition (CVPR)* (2009).
6) A. Krizhevsky, I. Sutskever, and G. E. Hinton., "Imagenet Classification with Deep Convolutional Neural Networks," in *Neural Information Processing Systems (NIPS)* (2012).
7) A. Vedaldi and A. Zisserman, "Efficient Additive Kernels via Explicit Feature Maps," in *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2012).
8) H. Jégou, M. Douze, and C. Schmid, "Product Quantization for Nearest Neighbor Search," in *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2011).
9) L. Jiang, D. Meng, T. Mitamura, and A. Hauptmann, "Easy Samples First: Self-paced Reranking for Zero-Example Multimedia Search," in *ACM Multimedia* (2014).
10) P. Natarajan, S. Wu, S. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, and R. Prasad, "Multimodal Feature Fusion for Robust Event Detection in Web Videos," in *Computer Vision and Pattern Recognition (CVPR)* (2012).
11) S.-I. Yu, L. Jiang, Z. Xu, Z. Lan, S. Xu, X. Chang, X. Li, Z. Mao, C. Gan, Y. Miao, X. Du, Y. Cai, L. Martin, N. Wolfe, A. Kumar, H. Li, M. Lin, Z. Ma, Y. Yang, D. Meng, S. Shan, P. D. Sahin, S. Burger, F. Metze, R. Singh, B. Raj, T. Mitamura, R. Stern, and A. Hauptmann, "CMU-Informedia @ TRECVID 2014," in *TRECVID Video Retrieval Evaluation Workshop* (2014).
12) Z.-Z. Lan, Y. Yang, N. Ballas, S.-I. Yu, and A. Haputmann, "Resource Constrained Multimedia Event Detection," in *Multimedia Modeling (MMM)* (2014).
13) S. Chaudhuri, M. Harvilla, and B. Raj, "Unsupervised Learning of Acoustic Unit Descriptors for Audio Content Representation and Classification," in *Interspeech* (2011).
14) F. Metze, S. Rawat, and Y. Wang, "Improved Audio Features for Large-Scale Multimedia Event Detection," in *International Conference on Multimedia and Expo (ICME)* (2014).
15) D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," in *International Journal of Computer Vision (IJCV)* (2004).
16) K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating Color Descriptors for Object and Scene Recognition," (2010).
17) Z. Xu, Y. Yang, and A. G. Hauptmann, "A Discriminative CNN Video Representation for Event Detection," in *Computer Vision and Pattern Recognition (CVPR)* (2015).
18) H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," in *International Conference on Computer Vision (ICCV)* (2013).
19) I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning Realistic Human Actions from Movies," in *Computer Vision and Pattern Recognition (CVPR)* (2008).
20) M. Chen and A. Hauptmann, "MoSIFT: Reocgnizing Human Actions in Surveillance Videos," tech. rep., (2009).
21) J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," in *Computer Vision and Pattern Recognition (CVPR)* (2003).
22) S. Lazebnik, C. Schmid, and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," in *Computer Vision and Pattern Recognition (CVPR)* (2006).
23) K. Chatfield, A. V. V. Lempitsky, and A. Zisserman, "The Devil is in the Details: an Evaluation of Recent Feature Encoding Methods," in *British Machine Vision Conference (BMVC)* (2011).
24) F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher Kernel for Large-scale Image Classification," in *European Conference on Computer Vision (ECCV)* (2010).
25) H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating Local Descriptors into a Compact Image Representation," in *Computer Vision and Pattern Recognition (CVPR)* (2010).
26) X. Chang, Y. Yang, E. P. Xing, and Y.-L. Yu, "Complex Event Detection using Semantic Saliency and Nearly-Isotonic SVM," in *International Conference on Machine Learning (ICML)* (2015).
27) 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 Video Retrieval Evaluation Workshop* (2013).
28) G. Ye, D. Liu, I.-H. Jhuo, S.-F. Chang, *et al.*, "Robust Late Fusion with Rank Minimization," in *Computer Vision and Pattern Recognition (CVPR)* (2012).
29) T. Joachims, "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization.," tech. rep., (1996).
30) L. Jiang, T. Mitamura, S.-I. Yu, and A. Hauptmann, "Zero-Example Event Search using MultiModal Pseudo Relevance Feedback," in *International Conference on Multimedia Retrieval (ICMR)* (2014).
31) L. Jiang, A. Hauptmann, and G. Xiang, "Leveraging High-level and Low-level Features for Multimedia Event Detection," in *ACM Multimedia* (2012).
32) J. Xiao, K. A. Ehinger, J. Hays, A. Torralba, and A. Oliva, "Sun Database: Exploring a Large Collection of Scene Categories," in *International Journal of Computer Vision (IJCV)* (2014).
33) S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-time Object Detection with Region Proposal Networks," in *Neural Information Processing Systems (NIPS)* (2015).
34) C. Zhai and J. Lafferty, "A Study of Smoothing Methods for Language Models Applied to Ad-hoc Information Retrieval," in *SIGIR* (2001).
35) "WordNet Similarity for Java, https://code.google.com/p/ws4j/,").
36) T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," in *Neural Information Processing Systems (NIPS)* (2013).
37) M. D. Zeiler and R. Fergus, "Visualizing and Understanding Convolutional Networks," in *European Conference on Computer Vision (ECCV)* (2014).
38) L. Jiang, D. Meng, *et al.*, "Self-paced Learning with Diversity," in *Neural Information Processing Systems (NIPS)* (2014).
39) M. P. Kumar, B. Packer, and D. Koller., "Self-paced Learning for Latent Variable Models," in *Neural Information Processing Systems (NIPS)* (2010).
40) S.-I. Yu, L. Jiang, and A. Hauptmann, "Instructional Videos for Unsupervised Harvesting and Learning of Action Examples," in *ACM Multimedia* (2014).

41) B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "The New Data and New Challenges in Multimedia Research," in *arXiv preprint arXiv:1503.01817* (2015).
42) A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale Video Classification with Convolutional Neural Networks," in *Computer Vision and Pattern Recognition (CVPR)* (2014).
43) Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum Learning," in *International Conference on Machine Learning (ICML)* (2009).
44) S.-I. Yu, L. Jiang, Z. Xu, Y. Yang, and A. G. Hauptmann, "Content-Based Video Search over 1 Million Videos with 1 Core in 1 Second," in *International Conference on Multimedia Retrieval (ICMR)* (2015).



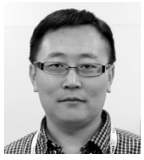
**Shoou-I Yu** received the B.S. in Computer Science and Information Engineering from National Taiwan University, Taiwan in 2009. He is now a Ph.D. student in Language Technologies Institute, Carnegie Mellon University. His research interests include multi-object tracking and multimedia retrieval.



**Yi Yang** Yi Yang received the PhD degree from Zhejiang University in 2010. He was a postdoc research fellow with the School of Computer Science at Carnegie Mellon University. He is now an Associate Professor with University of Technology Sydney. His research interest include multimedia, computer vision and machine learning.

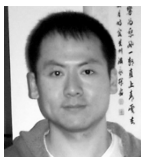
**Zhongwen Xu** received the B.E. in Computer Science and Technology from Zhejiang University, China in 2013. He is now a Ph.D. student at Centre for Quantum Computation & Intelligent Systems, University of Technology Sydney. His research interests are on computer vision and deep learning, especially for video analysis.

**Shicheng Xu** received the B.S. in Computer Science and Engineering from Zhejiang University, Hangzhou, Zhejiang, China, in 2014. He is now a Visiting Researcher in Language Technology Institute, Carnegie Mellon University. His research interests include multimedia analysis and multimedia retrieval.



**Deyu Meng** received the B.Sc., M.Sc., and Ph.D degrees in 2001, 2004, and 2008, respectively, from Xian Jiaotong University, Xian, China. He is currently an associate professor with the Institute for Information and System Sciences, School of Mathematics and Statistics, Xian Jiaotong University. From August 2012 to July 2014, he was a visiting scholar in Carnegie Mellon University. His current research interests include machine learning, computer vision, multimedia analysis and other related topics.

**Zexi Mao** Zexi Mao received the B.S. in Computer Science and Technology from Zhejiang University, China in 2013, and the M.S. in Language Technologies from Carnegie Mellon University, USA in 2015. He is currently a Data Engineer at Jetlore, Inc.



**Zhigang Ma** is now a Postdoctoral Research Fellow with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA. His research interest is mainly on machine learning and its applications to multimedia analysis and computer vision. He has authored or co-authored more than 20 scientific articles at top venues, including the IEEE T-PAMI, T-MM, IJCV, ACM MM, CVPR, AAAI and IJCAI. He was a PC member for ACM MM 2014; a TPC member for ICME 2014 and 2015; a TPC member for ICMR 2015 and a PC member for IJCAI 2015. He is also an invited reviewer for IEEE Transactions on Multimedia, IEEE Transactions on Cybernetics, Multimedia Tools and Applications, Neurocomputing, Computer Vision and Image Understanding. Dr. Ma received the Outstanding PhD thesis award from SIGMM and the best PhD thesis award from Gruppo Italiano Ricercatori in Pattern Recognition, Italy.

**Ming Lin** Ming Lin received his Bachelor and Doctor degree in the Department of Automation from Tsinghua University, Beijing, China, in 2008 and 2014. He is a Postdoctoral Research Fellow in the School of Computer Science at Carnegie Mellon University. His research interest is mainly on machine learning theory and its applications in computer vision.

**Xuanchong Li** Xuanchong Li received B.E. in computer science and technology from Zhejiang University, China in 2012. He is now a master student in Carnegie Mellon University. His research interest includes computer vision, machine learning.

**Huan Li** Huan received the PhD degree of computer science from Beihang University, China in 2012. She is currently a software engineer at Microsoft. Her research interest is mainly on machine learning and its applications to multimedia analysis and computer vision.

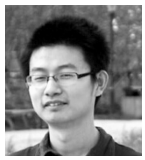
**Zhenzhong Lan** received the B.S. in software engineering and statistics from Sun Yat-sen University, China in 2010. He is now a Ph.D. student at Language Technologies Institute, Carnegie Mellon University. His research interests include computer vision and multimedia retrieval.

**Lu Jiang** Lu Jiang received his M.Sc. degree in Computer Science in 2011 and B.Sc. degree in Software Engineering in 2008, both from Xian Jiaotong University. Currently, he is a Ph.D candidate at school of computer science, Carnegie Mellon University. His research is focused on multimedia, machine learning, and big data.

**Alexander G. Hauptmann** received the B.A. and M.A. degrees in psychology from The Johns Hopkins University, Baltimore, MD, USA, in 1982, the "Diplom" in computer science from the Technische Universität Berlin,Berlin, Germany, in 1984, and the Ph.D. degree in computer science from Carnegie Mellon University (CMU), Pittsburgh, PA, USA in 1991. He is a Principal Systems" Scientist in the CMU Computer Science Department and also a faculty member with CMU's Language Technologies Institute. His research combines the areas of multimedia analysis and retrieval, man-machine interfaces, language processing, and machine learning. He is currently leading the Informedia project which engages in understanding of video data ranging from news to surveillance, Internet video for applications in general retrieval as well as healthcare.

**Chuang Gan** Chuang Gan received the B.E. in Electronic Engineering from Beihang University, China in 2013. He is now a Ph.D. student at Institute for Interdisciplinary Information Sciences, Tsinghua University. His research interests are on computer vision and machine learning, especially for large-scale video analysis.

**Xingzhong Du** received the B.S. in Software Engineering and M.S. in Computer Science from Nanjing University, China in 2010 and 2013 respectively. He is now a Ph.D. student at the University of Queensland. His research interests are on content-based video recommender system, video database and surveillance event detection.



**Xiaojun Chang** is a Ph.D. student at University of Technology Sydney. His research interests include machine learning, data mining and computer vision. His publications appear in proceedings of prestigious international conference like ICML, ACM MM, AAAI, IJCAI and etc.