Super Fast Event Recognition in Internet Videos

Yu-Gang Jiang, Qi Dai, Tao Mei, Yong Rui, and Shih-Fu Chang

***Abstract*—Techniques for recognizing high-level events in consumer videos on the Internet have many applications. Systems that produced state-of-the-art recognition performance usually contain modules requiring extensive computation, such as the extraction of the temporal motion trajectories, which cannot be deployed on large-scale datasets. In this paper, we provide a comprehensive study on efficient methods in this area and identify technical options for super fast event recognition in Internet videos. We start from analyzing a multimodal baseline that has produced good performance on popular benchmarks, by systematically evaluating each component in terms of both computational cost and contribution to recognition accuracy. After that, we identify alternative features, classifiers, and fusion strategies that can all be efficiently computed. In addition, we also provide a study on the following interesting question: for event recognition in Internet videos, what is the minimum number of visual and audio frames needed to obtain a comparable accuracy to that of using all the frames? Results on two rigorously designed datasets indicate that similar results can be maintained by using only a small portion of the visual frames. We also find that, different from the visual frames, the soundtracks contain little redundant information and thus sampling is always harmful. Integrating all the findings, our suggested recognition system is 2,350-fold faster than a baseline approach with even higher recognition accuracies. It recognizes 20 classes on a 120-second video sequence in just 1.78 seconds, using a regular desktop computer.**

***Index Terms*—Consumer videos, efficiency, event recognition, Internet videos, real time.**

I. Introduction

THE past decade has witnessed the explosion of user-generated videos on the Internet. As a result, there is a strong need of techniques for automatically recognizing high-level complex events in such videos, which are important in applications such as video search, personal video collection management, and smart advertising. See a few examples in Fig. 1. State-of-the-art event recognition systems often adopted a large set of features and classifiers in order to achieve a good accuracy, without seriously considering recognition efficiency. For instance, one of the popularly used features is based on the dense trajectories [1], which requires expensive analysis of every frame in a video sequence. While promising results have been achieved on several benchmark datasets [1], [2], these systems are computationally too slow to deal with large-scale data that is often seen in applications of the big data era.

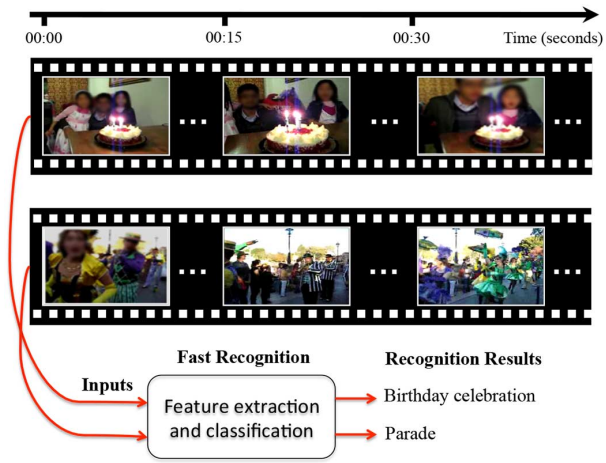


Fig. 1. This paper presents a very efficient system for recognizing events in Internet consumer videos, which only requires a few seconds to process a minutes-long video.

In this paper, we conduct a comprehensive study on many technical options with the goal of improving event recognition efficiency, while still obtaining a good accuracy. We start from evaluating the performance of a multimodal baseline, developed using several popular features that are the core components of many top-performing systems in the NIST TRECVID benchmark [3]. For each component of this baseline, we assess the computational cost and its contribution to the recognition accuracy. We then evaluate a large set of alternative methods or implementations to identify the best techniques for speed improvement. Valuable insights on the choices of features, feature quantization methods, classifiers and feature fusion schemes are attained from this comprehensive analysis. This finally leads to a super fast system for event recognition, which cleverly utilizes several features and classifiers that can be computed very efficiently.

In addition to evaluating popular features and classifiers, another question studied in this paper is: for event recognition in Internet videos, what is the minimum number of visual and audio frames needed to obtain a comparable accuracy to that of using all the frames? Popular systems normally extract features from the entire videos or on densely sampled frame sets without deeper investigation [1], [4]–[6]. However, humans can recognize many video semantics in a very short period of time, or

Manuscript received May 14, 2014; revised March 11, 2015; accepted May 10, 2015. Date of publication May 22, 2015; date of current version July 15, 2015. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Vasileios Mezaris.

Y.-G. Jiang and Q. Dai are with the School of Computer Science, Fudan University, Shanghai 201203, China (e-mail: ygj@fudan.edu.cn; daiqi@fudan.edu.cn).

T. Mei and Y. Rui are with Microsoft Research Asia, Beijing 100080, China (e-mail: tmei@microsoft.com; yongrui@microsoft.com).

S.-F. Chang is with the Department of Electrical Engineering and the Department of Computer Science, Columbia University, New York, NY 10027 USA (e-mail: sfchang@ee.columbia.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2015.2436813

sometimes even based on a single static visual frame. Our previous experience of event annotation using the Amazon MTurk indicates that, on average, human annotators only used 20 seconds to label an 80-second video (including operation time of marking all the found classes) [7]. These intuitions motivate us to evaluate the minimum number of frames required for event recognition by the automatic systems. In particular, we not only evaluate the number of visual frames needed for recognition, but also the number of audio frames (segments), as both modalities are important for video event recognition. This will provide unique insights to guide the design of an efficient system.

It is worth noting that this paper focuses on event recognition in user-generated consumer videos, a significant and probably the most dynamic portion of the Internet videos. Compared with other kinds of videos like movies or news, consumer videos have less or even no textual descriptions, and thus it is important to conduct content-based recognition to facilitate effective organization and retrieval. One property of the consumer videos is that the duration is normally very short and the content story is generally consistent. This implies that a very short segment may be only needed for event recognition. For other types of videos, the content may vary significantly over time and thus it is difficult to use just a subset of frames for recognition. In fact, for long videos like movies, it is not be suitable to assign video-level event labels, and segment/shot-level labels are more reasonable. Analyzing these professionally produced videos is beyond the focus of this work.

The rest of this paper is organized as follows. Section II discusses related works. Section III introduces various options for speeded up event recognition, including features, quantization methods, classifiers, fusion schemes, and frame sampling. We discuss a comprehensive set of experimental results on two datasets in Section IV. Finally, Section V summarizes the insights attained from this study and Section VI concludes this paper.

II. Related Work

The problem of recognizing complex events in unconstrained Internet videos is receiving significant research attentions. Existing methods on video content analysis such as human action recognition and video concept detection mostly only employed visual features [8], [4], [1], [9], while recent studies have proved that most video content recognition tasks can benefit from auditory clues and multimodal features should be jointly used [10], [5], [6], [2]. Typically an event recognition system first computes a large set of multimodal features, and then employs machine learning methods for classification, where SVM is the most popular option due to its robustness and efficiency [5], [6], [2], [11]. In the following we mainly discuss the works that have particularly considered the speed issue in the design of a recognition system.

Feature extraction is probably the slowest part of a recognition system. Popular features like the sparse keypoint-based SIFT [12] or the dense trajectories [1] are less suitable because they are too slow. In particular, the dense trajectory feature is extremely slow as it needs to process every video frame. Several studies have been devoted to speed up the feature extraction process. In [13], Bay *et al.* proposed a fast descriptor called speeded up robust feature (SURF), which is similar to the SIFT but is more efficient. In [14], Knopp *et al.* further extended SURF to the spatio-temporal space to locate descriptive local volumes for video analysis. It has also been frequently reported that dense sampling works similar to even better than the sparse local detectors [15], [1]. Different from the sparse detectors that find local maximums/minimums to locate invariant local patches, dense sampling computes descriptors densely on uniformly partitioned image/video patches. It is more efficient since the time-consuming detector phase is omitted, but it also requires to compute more descriptors as more patches are sampled. In [16], the authors showed that dense SIFT/SURF descriptors can be very efficiently computed using an engineering trick to reduce the computations on overlapped areas of nearby patches. Similar idea was also extended in [17] to implement fast versions of HOG and HOF features for human action recognition. In addition, recently the Convolutional Neural Networks (CNN) based features have been frequently used in video categorization [18], [19], [20], which can be quickly computed particularly on the GPU. In contrast to most visual features, audio features can be more efficiently extracted as the soundtrack is much more compact than the visual frames.

Many effective descriptors are extracted locally, and thus the number of the local descriptors varies across videos. A quantization step is needed to convert these sets of descriptors to fixed dimensional features, which are required by most learning algorithms. The most popular quantization method is called bag-of-words, which maps the descriptors to a set of pre-generated codewords, and the final representation is a frequency-based histogram. The mapping or quantization process is computationally slow if we compute the similarities between the local descriptors and the codewords in a straightforward way. In [21], Nister *et al.* used a tree-based structure to organize the codewords, so that the quantization can be done in a top-down mapping procedure to reduce the amount of similarity calculations. In [22], Moosmann *et al.* employed random forest, a special kind of trees, for fast computation of the bag-of-words features. This method has been used in [16] for fast image-based concept detection. In [23], the authors extended a method called semantic texton forests [24] from images to videos for fast human action recognition. More recently, the authors of [25] proposed a method for fast feature quantization, based on an assumption that local neighboring keypoints are visually and semantically similar.

Besides the features, classifier is another expensive component particularly when there are many classes to be recognized. Support Vector Machine (SVM) is the most popular classifier for video analysis, which has been widely used in many state-of-the-art systems. The nonlinear kernels such as the Histogram Intersection or the χ^2 are computationally expensive but are needed to achieve good recognition accuracy. In [26], Maji *et al.* used an approximation based method to reduce the number of support vectors to be compared by a test sample, which can largely reduce testing time and has been used in [16] for fast visual concept detection. In addition, the fast linear kernel has been observed to be suitable for high-dimensional features such as the Fisher vectors [27]. However, computing the Fisher vectors is slower than computing the bag-of-words representation

with efficient nearest neighbor search methods, as the former requires to compute the first and second order statistics instead of simple counting in the bag-of-words. Instead of the SVM, several recent works adopted neural networks for classification [18]–[20].

There are also a few works focusing on the selection or sampling of frames for recognition. In [28], Habibian *et al.* proposed to identify and remove stop frames in video event recognition, such as blank and blurry frames, and found that removing those frames is helpful. This work did not study the minimum number of required frames for recognition. In [29], Subhabrata *et al.* conducted an interesting human study to investigate the minimal needed evidence for event recognition. They found that a single microshot of just 1.5 seconds is sufficient for humans to predict the event class of many videos. Earlier, the authors of [30] evaluated the number of frames needed for action recognition in videos. The conclusion was that a single frame is adequate for many action classes. One reason of this observation is that the videos used in their experiments were captured in controlled environment and the samples are easy to be identified. Therefore we believe this may not generalize to the domain of complex consumer videos. Furthermore, in this work we not only study the number of needed visual frames, but also the audio counterpart.

This work is extended based on a conference paper [31]. The extensions include: 1) discussions and experiments on the accuracy and speed of the popular dense trajectory features and CNN features; 2) using random forest for feature quantization, which further improves the speed; 3) new experiments to evaluate a Kernel Regression classifier; and 4) additional experiments on a new dataset, which ensure that the observations and conclusions are more generalizable. In addition to our previous work [31], Lan *et al.* [32] recently studied the speed efficiency of several event recognition techniques, Ma *et al.* [33] discussed several options to improve speed, and Uijlings *et al.* [17] presented fast implementations of HOG/HOF features for human action recognition. However, Ma *et al.* [33] did not provide thorough experimental validations. Lan *et al.* [32] and Uijlings *et al.* [17] only focused on feature options without evaluating other factors such as classifiers and frame sampling strategies, which are also critical in the design of a fast recognition system.

III. Super Fast Event Recognition

In this section, we first introduce a multimodal baseline recognition system, which is used as a starting point for identifying alternative components to optimize the recognition speed.

A. A Multimodal Baseline System

For the baseline system, we consider critical components of several systems that have produced state-of-the-art results [5], [34], [2]. Features are the key factor in recognition performance and thus some of the strongest and the most popular features are adopted. For classification and fusion, we use the most popular option of SVM classifier and late fusion. Some recent approaches adopted sophisticated multimodal fusion strategies [35], [6], which may lead to slightly better performance but is computationally slow.

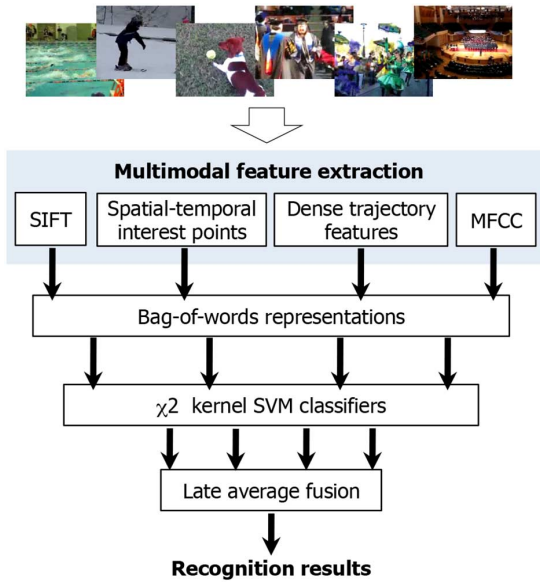


Fig. 2. Framework of the baseline event recognition system. We start from analyzing the components of this system and then identify alternative methods for speed improvements.

Fig. 2 shows the framework of the baseline approach. The extracted features cover all the major clues including static visual descriptor, motion-based visual descriptor, and audio descriptor. The descriptors are converted to the bag-of-words representation for classification with the SVM. Results of separate SVM classifiers trained with different features are combined by late fusion. More details are given in the following.

1) Static Visual Descriptor: We adopt SIFT as the static visual feature, which is the most popular feature in many image/video related applications. For local patch detection, we adopt the Difference of Gaussian (DoG) [12] and the Hessian Affine [36], which are complementary because they focus on different aspects of invariance. As computing features on every video frame is slow, we only extract static visual feature from a set of sampled frames (one frame every two seconds). This sampling rate has been frequently used in several systems. We will evaluate this factor in detail in the experiments.

2) Motion Descriptor: Different from the static SIFT features, motion features capture the moving characteristics in videos, which are important for events with strong motions. The first motion feature considered here is the spatial-temporal interest point (STIP) [37], which detects invariant spatial-temporal volumes, just like the Hessian Affine for detecting the static image patches. HOG and HOF descriptors are computed based on pixel values in each volume.

We also consider the dense trajectory descriptors [1], which have recently demonstrated top performance on various benchmark datasets. First, densely sampled local frame patches are tracked over time. Four kinds of descriptors are then computed for each trajectory, including a 30-d trajectory shape descriptor, a 96-d HOG descriptor, a 108-d HOF descriptor, and a 108-d Motion Boundary Histogram (MBH) descriptor. All these motion-based descriptors are expensive to compute as they need to process all the frames. Sampling the frames will always hurt the performance as the motion patterns will be destroyed.

3) Audio Descriptor: Neuroscientists have found that multiple senses can cooperate to enhance perception performance [38]. It has also been frequently reported that audio is a helpful clue for video event recognition. We therefore extract the well-known MFCC descriptors, which are computed over densely sampled audio segments/frames (32 ms of duration). Nearby segments have 50% (16 ms) overlap to minimize the information loss. As the soundtrack is much more compact than the visual frames, it is efficient to compute these descriptors although the audio frames are densely sampled.

4) Descriptor Quantization: All the aforementioned descriptors vary in set size across videos, requiring a quantization method to convert them into fixed-dimensional video representations. We adopt the well-known bag-of-words. Two codebooks, each with 500 codewords, are adopted for the SIFT descriptors. One for the DoG patches and the other for the Hessian Affine patches. To take the spatial locations of the SIFT descriptors into account, two spatial pyramid layers are adopted (1×1 and 2×2). This leads to a final representation of 5,000 dimensions ($2 \times 500 \times (1 + 2 \times 2)$) for each visual frame, and the representations of different frames of the same video are averaged to form the video-level representation.

For the STIP, the dense trajectory based descriptors, and the MFCC, codebooks of 5,000, 4,000, 4,000 codewords are adopted respectively. The spatial pyramids are not considered for STIP and dense trajectories, because the observed performance gain reported in prior works is not very significant. The four dense trajectory based descriptors are quantized separately and then combined by kernel-level fusion in the classification process, following the settings of [1].

For all the descriptors, we use a soft-weighting method to alleviate the quantization loss [39]. The similarities between the descriptors and the codewords are computed based on the inner product of L-2 normalized vectors, which is more efficient than brute-force computation of the Euclidean distances [16].

5) Classification and Fusion: As briefly mentioned earlier, we adopt the χ^2 kernel SVM for classification. Specifically, the one-vs-all strategy is employed to train a separate classifier for each event class using each feature. The prediction scores of the SVM classifiers using different features are combined using late fusion with average fusion weights. Adaptive weights may further improve the results, but the learned fusion weights (using methods like cross validation) were often observed to be less generalizable to new test data.

This baseline system contains critical components of many recent top-performing systems. The three features SIFT, STIP, and MFCC were jointly used in the top-performing system of TRECVID multimedia event detection task in 2010 [5], and the dense trajectory features were used as the central technique in the top-performing system of the same task in 2013 [2].

B. Alternative Techniques

This section discusses several alternative techniques that are potentially useful for improving the speed of the baseline system, covering all the components of a video event recognition system. The goal is to identify the best suitable techniques for each component of the system to obtain a high recognition accuracy while optimizing the speed. Notice that some of the adopted techniques were originally proposed to optimize the speed of various recognition problems, but they were mostly studied separately in the literature. This paper evaluates many technical options and integrates the findings to realize a fast video event recognition system. In the following we introduce the evaluated techniques.

1) Alternative Features: Because of slow computation, strong features like the dense trajectories cannot be deployed in a fast recognition system. We only evaluate alternative visual features in this work as the audio features like the MFCC are very efficient. The visual features listed below are selected because they are all relatively efficient. Notice that we do not consider all kinds of spatial-temporal features computed densely on all the frames like the fast HOG/HOF from [17], as they still require much more time than the static frame-based features, which violates our ultimate goal of developing a super fast recognition system.

- Convolutional Neural Network (CNN) based features: Recently, the off-the-shelf CNN features [40] have been popular in many visual recognition tasks. Several works have extracted frame-level features from a CNN model trained on the ImageNet data and reported promising video categorization performance [41], [42]. The appealing results clearly demonstrated that the CNN features are powerful and should be considered as an option. In this work, we adopt the AlexNet model [43] and use the outputs of the seventh fully-connected layer as features (4,096 dimensions).
- DIFT and DURF: The sparse SIFT has been observed to be very useful, but is computationally slow. We therefore adopt the fast versions of the dense SIFT (DIFT) and dense SURF (DURF) descriptors, developed by the authors of [16]. These descriptors have been shown to be effective for image-based concept detection. A visual codebook of 500 codewords is generated to quantize each of the two dense descriptors with the spatial pyramids.
- Self-Similarities (SSIM): Another static visual feature considered here is SSIM [44], which is also a local descriptor like the SIFT but is computed in a very different way. Instead of replying on the gradients within a local frame patch, SSIM quantizes a correlation map of the patch in a larger window. This descriptor is also computed on densely sampled patches, and the same form of quantization is performed using a codebook of 500 codewords.
- Color Moment (CM): Color is not considered in the baseline system, so we expect that this simple and efficient global descriptor may improve the results. *Lab* color space is adopted, and the first three moments in each of the three *Lab* channels are computed and concatenated. Each frame is divided into 25 grids, and the color moments are computed in each grid separately and then concatenated to form a final representation.
- GIST: This is another global descriptor, computed based on the outputs of Gabor-like filters over an image partition of 16 grids [45]. Eight orientations and four scales are used to generate the Gabor filters, leading to a representation of 512 dimensions.

- Local Binary Patterns (LBP) [46]: This feature compares each frame pixel to its neighboring pixels (8 neighbors in total). The pixel is then labeled using binary numbers based on the comparison results. The binary vectors ($2^8 = 256$ dimensions) of all the pixels are accumulated to form the LBP representation of a frame.
- Tiny Images (TINY) [47]: This feature is very simple, computed by concatenating pixel values of each frame resized to 32×32 pixels (3,072 dimensions). The resized tiny frames are used to control the dimension of the representation. In addition, when using very small frame sizes, it is also helpful for reducing the misalignments of similar semantics across frames.

2) Alternative Quantization Methods: The bag-of-words quantization is needed to consolidate the feature sets into fixed-dimensional video representations, including SIFT, STIP, dense trajectory descriptors and MFCC from the baseline, and DIFT, DURF and SSIM from the features listed in the previous subsection.

One option is to use the inner product of L-2 normalized vectors like the baseline system. We also adopt the random forest [22] to further reduce the quantization time. Following [22], we adopt a random forest of 4 trees, each with a depth of 7 levels. This generates a vocabulary of 512 words, similar to the adopted vocabulary sizes of most features in the inner product based quantization process. For the features that do not use the spatial pyramid in quantization (e.g., the MFCC), we use 4 trees of 10 levels, leading to vocabularies of 4,096 words.

3) Alternative Classification and Fusion Methods: Compared with feature extraction, classification and fusion are much more efficient. However, as our goal is trying to reach the speed limit of video event recognition, alternative methods saving even a half second of processing time are worth exploring. For classification, we consider SVM with the fast Histogram Intersection (HI) kernel from [26], where the authors proposed to use a small set of basis vectors to approximate the large number of support vectors in an SVM model. This greatly reduces the testing time as a test sample only needs to be compared with the basis vectors, and the distances are used to estimate the distances with the original support vectors. In addition, we also evaluate the kernel ridge regression (KRR) classifier [48], which is easy to implement and has been reported to be effective in several video categorization tasks [6].

Different from late fusion that combines the prediction outputs of separate classifiers trained with different features, early fusion concatenates multiple feature representations before classification with a single model. One drawback of early fusion is that representations of higher dimensions may overwhelm those in lower dimensional spaces, so some preprocessing techniques such as normalization should be used to avoid that. A slightly different version of the early fusion is called kernel fusion, which computes a kernel for each feature separately and then fuses the kernels together for classification. Early fusion and kernel fusion are the same for several simple forms of kernels, but are slightly different when using popular kernels like the χ^2 Gaussian. All the three fusion methods will be evaluated in this work.

Since weighted linear fusion is frequently used in all the aforementioned methods, the importance of each feature is measured by its weight used in the fusion process. Average weights have been the most popular option, which are also adopted throughout the experiments of this work. Advanced options for estimating adapted weights like multiple kernel learning [49] and robust late fusion [35] demand more computations but do not improve the results significantly. In addition, since using the adapted weights will have similar effects on all the three fusion strategies, we expect that uniform average fusion will be sufficient to judge or to compare the three methods, i.e., the conclusion on which strategy is better will be unlikely different when using the adapted weights.

The speed of the three fusion methods does not differ very greatly. Early and kernel fusion are slightly faster than the late fusion adopted in the baseline system as they only have one SVM model. Between early and kernel fusion, the speed is the same for kernels like HI. For the χ^2 Gaussian kernel ($\mathcal{K}(\mathbf{x}, \mathbf{y}) = e^{-\rho d_{\chi^2}(\mathbf{x},\mathbf{y})}$, where $d_{\chi^2}()$ is the χ^2 distance of $\mathbf{x}$ and $\mathbf{y}$), since the exponential function has to be computed multiple times for the kernel fusion, it is slightly slower than early fusion, for which only one time is needed.

4) Frame Sampling: Besides feature representation and classification, another factor that affects the speed significantly is frame sampling, which is often overlooked in the design of an efficient system. It is obvious that using too many frames will be computationally slow, and analyzing too few frames may not be sufficient because of information loss. In this work, we intend to study the relationships between the number of used frames and recognition performance, with the goal of identifying suitable frame numbers that represent a good tradeoff between speed and accuracy. This study is partly motivated by recent studies on human recognition of video events. In [7], the authors found that less than 20 seconds were needed for humans to accurately annotate an 80-second video. A recent work in [29] further shows that a very short segment with 1.5 seconds is already sufficient for many events.

There are mainly two ways for frame sampling. The first one is uniform sampling, which evenly selects frames over the entire videos. Another option is continuous sampling, i.e., using continuously selected frames at the beginning, in the middle, or at the end of a video. We will evaluate these options in the experiments. We noticed that there also exist more advanced methods for frame sampling, to maximize the information remained in the selected frames, which however require additional computation that is not desired in the design of a super fast recognition system.

IV. Experiments

A. Datasets and Performance Measures

We adopt two datasets to evaluate the aforementioned techniques. The first one is the Columbia Consumer Video (CCV) dataset [7], which has been widely adopted in several recent studies on Internet video analysis. There are 9,317 videos, divided evenly into a training set and a test set. The dataset contains 20 categories, 15 of which are events, annotated based

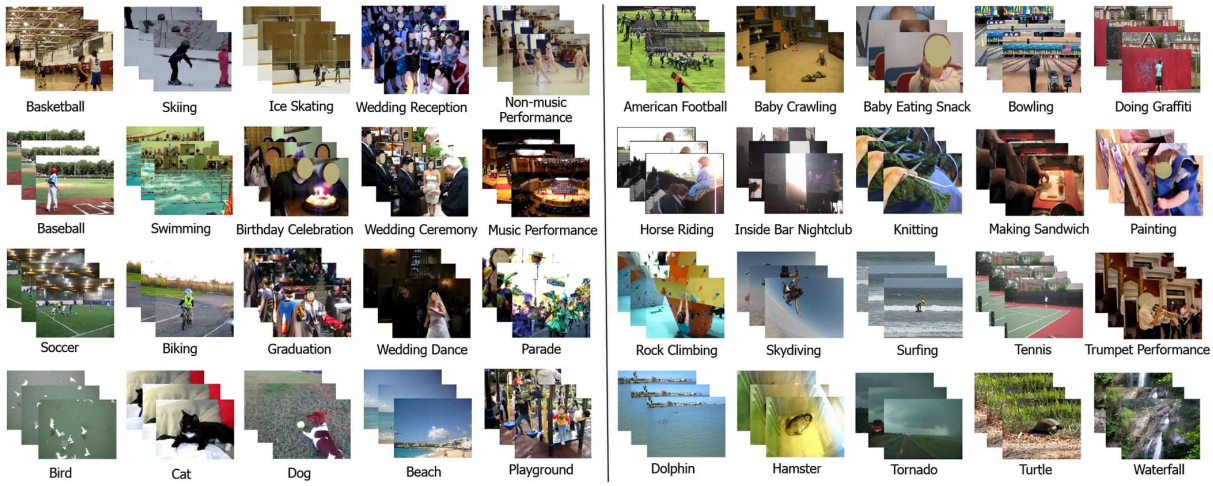


Fig. 3. Examples of the Columbia Consumer Video dataset (left) and the new CV20 dataset we collected (right). Among the 20 categories in each dataset, most are events, and a few (in the bottom row) are more related to objects and scenes. The task is very challenging due to the severe intra-class variations.

on Amazon's MTurk crowdsourcing platform. On average, the video duration is around 80 seconds. CCV is publicly available,[1] and interested readers are referred to [7] for more details.

To evaluate the generalizability of the findings, we collected another dataset of 20 categories from YouTube, named as CV20, containing 3,808 videos in total. Manual annotation was performed for all the categories. The average video duration of this dataset is 120 seconds. Similar to CCV, the dataset also has 15 events and 5 objects/scenes. Positive samples are evenly divided into training and test sets. Fig. 3 gives an example for each category in both the datasets.

The one-vs-all strategy is adopted to train separate classifiers for each category, and the test samples are ranked based on the predicted score values. The accuracy of recognition is measured by the widely adopted average precision (AP), which approximates the area under the precision-recall curve. For the overall accuracy of an entire dataset, we use mean AP (mAP) of all the categories.

The speed of recognition is evaluated as the time needed for feature extraction and classification. We evaluate the speed of feature extraction and classification (including fusion) methods separately in order to identify the most suitable techniques. For feature extraction, we report the average time of extracting features from CV20, i.e., the average time for processing a 120-second video. Notice that, for the visual features, one frame is sampled from every 2-second segment, meaning that the time is for the feature extraction of around 60 frames. For the classification speed, we report the average time needed for classifying a video using models of 20 categories. All the reported speed performances are evaluated on a regular PC with an Intel i7 4770 3.4 GHz CPU (using a single thread only) and 32 GB RAM. Note that this is different from the hardware used for the conference version [31].

For most of the evaluated techniques, we adopt publicly available codes from the original authors. For instance, the codes of sparse keypoint detectors and dense trajectories are from the LEAR group[2] at INRIA. The STIP codes are from Laptev [37], the DIFT/DURF codes are from Uijlings *et al.* [16], the CNN codes come from the Caffe framework of Jia *et al.* [50], and the fast SVM classifier codes are from Maji *et al.* [26]. These codes are adopted without major modifications, but may be further optimized for improved speed.

B. Evaluation Plan

Based on the discussions in Section III, we divide the evaluations into the following four parts.

Part 1: *Feature Representations.* We first evaluate the recognition accuracy and computational efficiency of the features. Feature extraction is the slowest step in a recognition system and it is very important to identify a set of efficient and reliable features. We examine both the baseline features and the alternative features in this part.

Part 2: *Quantization Methods.* In the second part, we compare the speed and accuracy of the two descriptor quantization methods. This part is much more efficient than computing the descriptors, but still occupies a considerable amount of computational time.

Part 3: *Classification and Fusion.* After selecting the most suitable features and quantization methods, we evaluate classifiers and fusion strategies, with the same goal of optimizing speed while obtaining a high accuracy.

Part 4: *Number of Audio/Visual Frames.* The last experiment is to evaluate the needed audio/visual frames for event recognition. This is not a part of the core techniques, but is very important for speed improvement.

C. Feature Representations

Results of the feature representations are summarized in Table I. By fusing the SIFT, STIP and MFCC features in the baseline system, we achieve a very good mAP of 0.595 on CCV and 0.868 on the new CV20 dataset. Fig. 4 shows the confusion matrices of both datasets, where we can clearly see

[1][Online]. Available: http://www.ee.columbia.edu/dvmm/CCV/

[2][Online]. Available: http://lear.inrialpes.fr/software

TABLE I
ACCURACY (MAP) AND SPEED OF THE BASELINE AND THE ALTERNATIVE FEATURES ON BOTH DATASETS. SPEED IS MEASURED IN SECONDS REQUIRED FOR PROCESSING A 120-SECOND VIDEO, INCLUDING QUANTIZATION TIME USING THE INNER PRODUCT-BASED METHOD. THE FUSION OF MULTIPLE FEATURES IS ACHIEVED BY THE LATE FUSION WITH EQUAL WEIGHTS. THE DENSE TRAJECTORY FEATURE IS ONLY EVALUATED ON CCV SINCE IT IS TOO SLOW AND CANNOT BE USED IN FAST RECOGNITION SYSTEMS

Feature Representation(s)	CCV	CV20	Time
SIFT	0.523	0.805	50
STIP	0.449	0.691	602
DenseTraj	0.642	—	3528
MFCC	0.331	0.514	3
SIFT+STIP	0.551	0.837	652
Base3 (SIFT+STIP+MFCC)	0.595	0.868	655
Base4 (SIFT+STIP+DenseTraj+MFCC)	0.669	—	4183
CNN	0.673	0.883	1.5
DIFT	0.493	0.727	6
DURF	0.513	0.731	5
SSIM	0.463	0.680	19
CM	0.324	0.475	4
GIST	0.325	0.566	4
LBP	0.285	0.486	1
TINY	0.229	0.340	0.4
CNN+DIFT	0.677	0.883	7.5
CNN+DURF	0.683	0.885	6.5
CNN+DURF+DIFT	0.681	0.884	12.5
CNN+DURF+SSIM	0.684	0.886	25.5
CNN+DURF+SSIM+CM	0.683	0.885	29.5
CNN+DURF+SSIM+GIST	0.682	0.885	29.5
CNN+DURF+SSIM+LBP	0.684	0.884	26.5
CNN+DURF+SSIM+TINY	0.682	0.884	25.9
Base3+CNN+DURF+SSIM	0.708	0.912	680.5
MFCC+CNN+DURF+SSIM	0.702	0.902	28.5
MFCC+CNN+DURF	0.701	0.900	9.5
MFCC+CNN	0.694	0.895	4.5

that CCV has more correlated categories like the last three related to wedding, which make it more difficult than CV20. As shown in the table, adding the dense trajectory based features can improve the results to 0.669 on CCV. We did not further evaluate this feature on CV20 as it is too slow to be used in our targeted fast recognition system. Even the SIFT (based on sparse local detectors) and STIP features are not acceptable as our goal is to spend just a few seconds to process a video.

Looking at the individual feature performance, the accuracy of STIP is not very good compared with the more efficient static SIFT. The main reason is that most events can be manifested based on static scenes. While motion is useful and can serve as a complementary clue to the static counterpart, purely computing motion (i.e., HOF) and appearance (i.e., HOG) on the dynamic STIP local volumes is not sufficient for recognizing these events. In contrast, the dense trajectory feature is the strongest as it is very comprehensive, covering not only visual appearance but also motion with special design to mitigate the effect of camera motion (by using the MBH descriptors). In addition, the dense trajectory features are computed densely, not only on the moving invariant volumes detected by the STIP, but also on other areas of the scene backgrounds. However, although these motion-based features produce good accuracies, their speed does not allow us to adopt them for fast recognition.

All the considered alternative features are efficient (see the middle group of results in the table). Compared with the sparse SIFT in the baseline, only 40% of the extraction time is needed by the slowest alternative feature SSIM, and the CNN feature only requires 3% of the time. Note that the speed of the CNN feature is evaluated on the same CPU with Intel's CBLAS library. If computed on GPU, the CNN feature can be even more efficiently extracted.

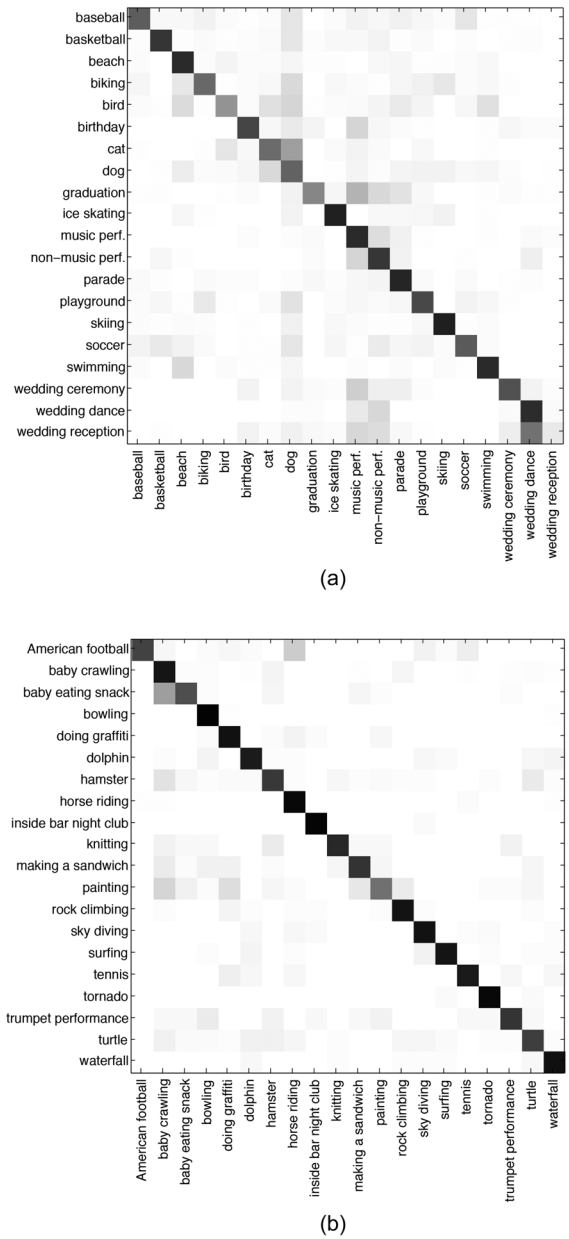


Fig. 4. Confusion matrices of CCV and CV20, using the baseline of 3 features ("Base3" in Table I). (a) CCV. (b) CV20.

Among the alternative features, CNN is very effective in terms of mAP on both datasets. Using itself is already better than the baseline that combines multiple traditional features. This is extremely appealing as the CNN feature is also efficient. For the others, DIFT and DURF are also good, while TINY is the worst but still demonstrates strong discriminative power (compared with a random prediction accuracy of just around 5% on both datasets).

We also evaluate the fusion of the alternative features, in order to study if they are complementary. For this, we start from the CNN and incrementally fuse it with the other features. A feature is dropped from fusion if it does not improve the result. As

shown in the table, fusing DIFT and DURF can both improve upon CNN, but combining three of them together is worse than the fusion of CNN and DURF. Therefore, DIFT is discarded. Similarly, most of the other features do not further improve the result, except SSIM that helps very slightly.

We further fuse the baseline features with the alternative features. We see that fusing the three more efficient baseline features (Base3) with the remaining three alternative features can achieve 0.708 on CCV and 0.912 on CV20. This is clearly better than both the baseline and the fusion of the three alternative features. As SIFT and STIP are fairly slow, we only use MFCC from the baseline and fuse it with the new features. This leads to the results shown in the bottom three rows of the table, which are suggested to be used for fast event recognition. Interestingly, all the three combinations are clearly better than the baseline because of the use of the CNN feature, which is not only more accurate but also very efficient. As for which of three combinations should be adopted, it would depend on the real application needs and hardware configurations. Notice that the computational time reported in the table includes feature quantization time. As the inner product based quantization is used in this experiment, we expect that using the tree-based quantization methods like the random forest will further improve speed, as will be discussed in the following subsection.

D. Quantization Methods

All the selected features, except the CNN, require a quantization process to produce a fixed dimensional representation. In this subsection, we compare the two descriptor quantization methods. Fig. 5 shows the results. Overall, the accuracies of the two methods are very similar on both datasets. For both the individual feature accuracy and the fusion accuracy, we do not observe significant differences. The speed of the random forest based quantization is much faster. As shown in Fig. 5(c), most of the time indicated by the light blue bars is for descriptor calculation, and quantization costs almost negligible amount of time. With the random forest based quantization, only around 2 seconds per video are needed for computing and quantizing both the MFCC and the DURF features (compared with 8 seconds needed by the inner product based method), which is a very significant improvement especially for large scale applications.

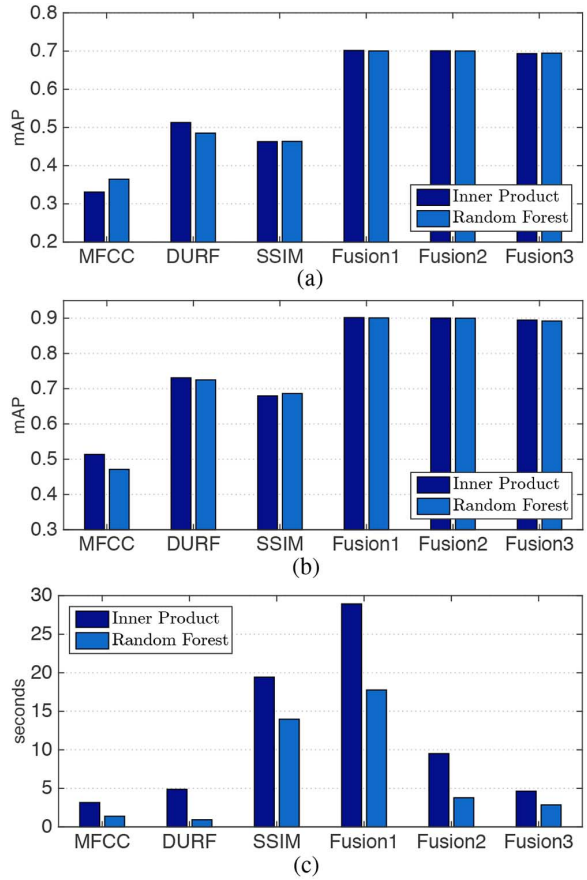


Fig. 5. Recognition accuracy and speed of various features using the inner product based quantization and the random forest based quantization. Accuracy is reported on both CCV and CV20, and speed is measured by the feature extraction time for processing a 120-second video. Fusion 1-3 represent the feature combinations listed in the bottom three rows of Table I, respectively. (a) Accuracy (mAP) on CCV. (b) Accuracy (mAP) on CV20. (c) Speed.

E. Classification and Fusion

Next, we discuss classification and fusion methods. We evaluate SVM with different kernel options, including the fastHI kernel, the traditional HI kernel and the χ^2 kernel from the baseline. By comparing all the three kernels, we can have a better understanding of the power of the fastHI. For KRR, we only report the performance of the RBF kernel as we observe that the speed of other kernels like χ^2 is slower, which are thus not desired in a fast recognition system. The three remaining alternative features and the MFCC are adopted in this evaluation.

Fig. 6 visualizes the results on both datasets. As shown in (a) and (b) of the figure, the χ^2 SVM is consistently better than the HI SVM and the fastHI SVM. This is not surprising as the χ^2 is a recognized kernel particularly suitable for the histogram-like bag-of-words representations. For all the features, the KRR is not as good as the SVM classifiers, but the gap is not significant. The speed comparisons are shown in (c) and (d), where we observe that the χ^2 SVM is the slowest and the fastHI SVM is extremely efficient. The speed-up from fastHI is around 50-100 times compared with the traditional HI, which is different from the observation of [16], where the authors only observed a 18-times speed-up. This is because the work of [16] used pre-computed SVM kernels, which could be reused for all the categories. In practice, the pre-computed kernels are not suitable as test data may arrive on the fly. Notice that the KRR may be used together with the kernel approximation methods for faster speed, but giving that it is similar or slightly worse than SVM, using the fastHI SVM is sufficient for our goal.

We now move on to compare the classifiers under different fusion settings. The fusion results are important as it is very unlikely that only one feature is used in a robust recognition system. As shown in Fig. 7, a very interesting observation is that, after feature fusion, the accuracy gap among different SVM kernels becomes smaller or even invisible in most cases. This is very appealing as we can adopt the fastHI kernel SVM without significant performance degradation. The gap between SVM and KRR remains similar before and after feature fusion.

Among the three fusion methods, early and kernel fusion tend to be slightly better than late fusion. This indicates that combining features before learning the classification models is de-

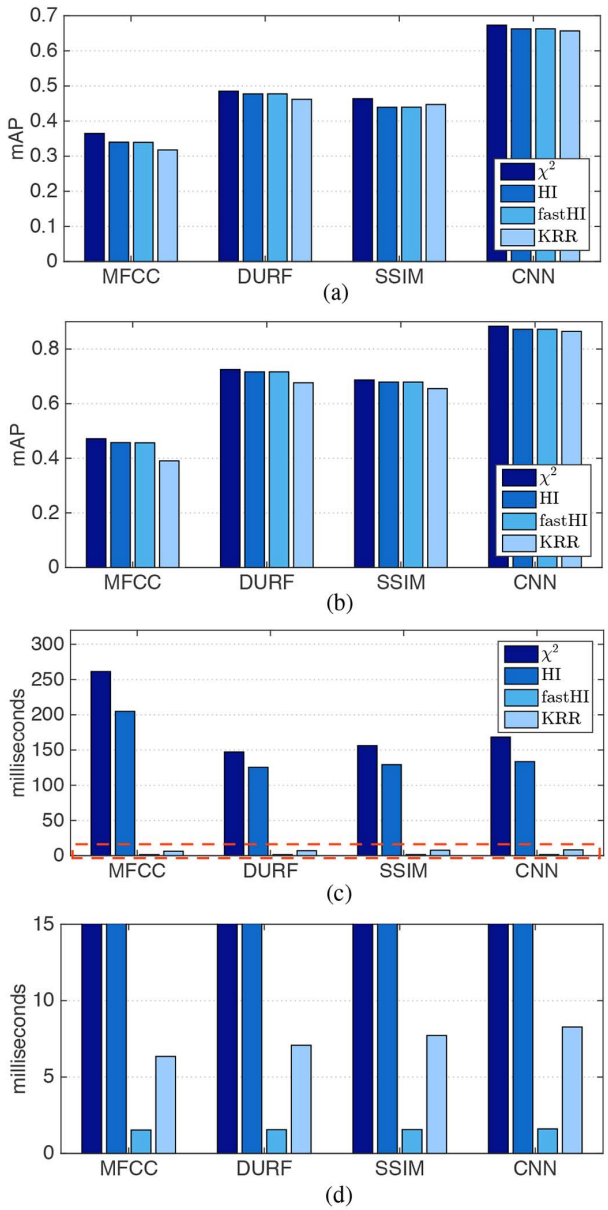


Fig. 6. Recognition accuracy (a), (b) and speed (c), (d) of the SVM classifier (three kernels) and the KRR classifier. Speed is measured by the average classification time to process a test video in CV20. (a) Accuracy (mAP) on CCV. (b) Accuracy (mAP) on CV20. (c) Speed. (d) Zoom-in view of the area in the dotted box in (c).

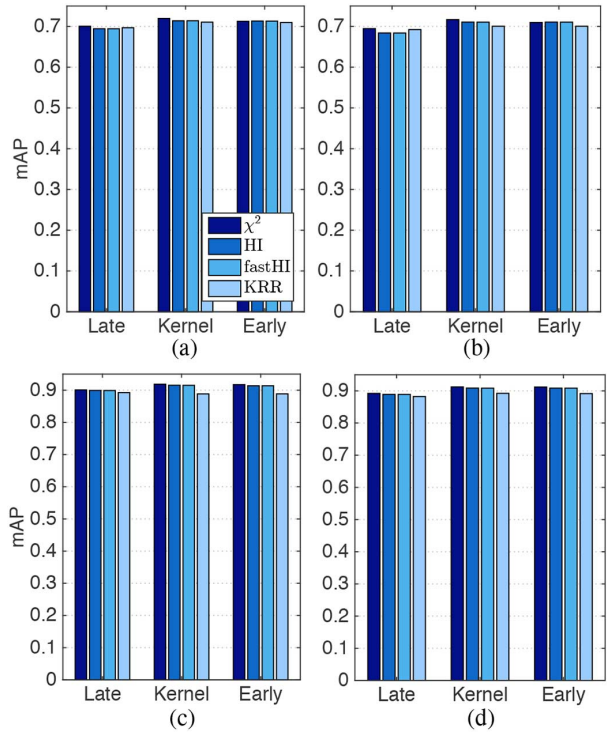


Fig. 7. Comparison of the three fusion methods on CCV and CV20, using different classification options. (a), (c) Fusion of MFCC, CNN, DURF and SSIM; (b), (d) fusion of MFCC and CNN. (a) CCV. (b) CCV. (c) CV20. (d) CV20.

sired, because early fusion of the features may form a new feature space where the positive and negative data samples can be better separated. Since early/kernel fusion is also slightly more efficient than late fusion because less models are needed to be trained, they are more suitable to be adopted by a fast recognition system.

F. Number of Audio/Visual Frames

Finally, we measure the number of required audio and visual frames in video event recognition. The MFCC and CNN features are used in this experiment. In addition to separately studying the number of needed audio and visual frames, we are also interested in comparing the two modalities.

We plot the results in Fig. 8, where the word "max" means that the entire sequence will be used if a video is shorter than a desired duration. We see that it is always harmful to sample the audio frames, which is not surprising as, even for humans, it is difficult to understand the content of a video by only listening a fraction of the audio soundtrack. In contrast, sampling the visual feature appears to be safe until reaching a certain number. On both datasets, we observe that 16 frames are suitable with almost invisible performance degradation. This verifies our expectation that the visual channel contains significant redundant information, while the audio channel does not.

Our observation on the number of needed visual frames is different from a few recent studies in [30], [29]. The authors of [30] found that only a single frame is needed for recognizing many human actions. However, the videos used in their study only contain a single subject with clean background, which are quite different from the complex videos on the Internet. In addition, compared with a recent human recognition study [29], where the authors found that a segment of 1.5 seconds is generally sufficient for humans to recognition many events in complex Internet videos, our observations indicate that longer segments are needed for the automatic machine algorithms. This is because the capability of current automatic techniques are still far below that of the humans.

Comparing the sampling strategies, uniform sampling is clearly better, implying that frames of the entire video are informative. Also, the continuous sampling method will involve redundant information as nearby frames tend to be similar. Among the continuous sampling methods, "Middle" is slightly better in most cases, indicating that the middle part of the videos may contain more important or representative information. In addition, on CV20 which has longer videos (average duration

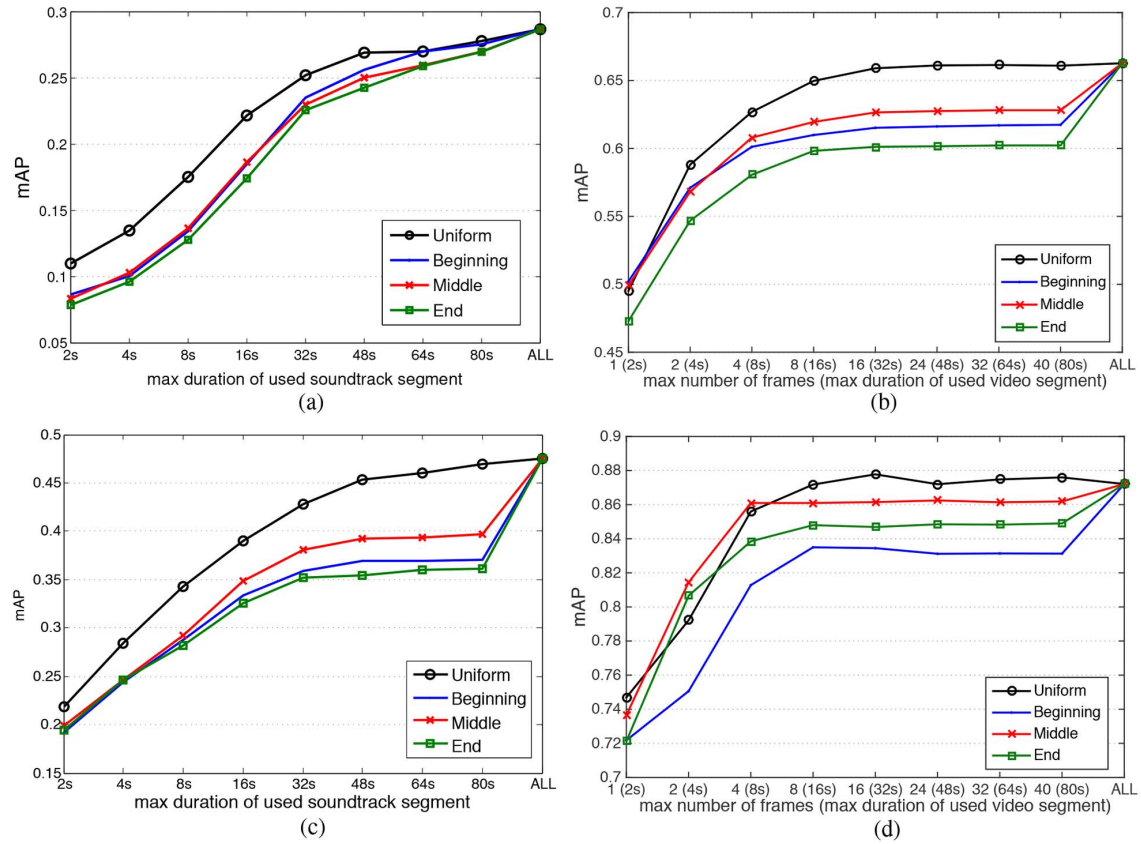


Fig. 8. Recognition accuracies with different numbers of sampled audio/visual frames. Sampling the audio frames is always harmful, while good recognition results can be maintained by sampling the visual frames. Observations on the two datasets are fairly consistent. (a) MFCC on CCV. (b) CNN on CCV. (c) MFCC on CV20. (d) CNN on CV20.

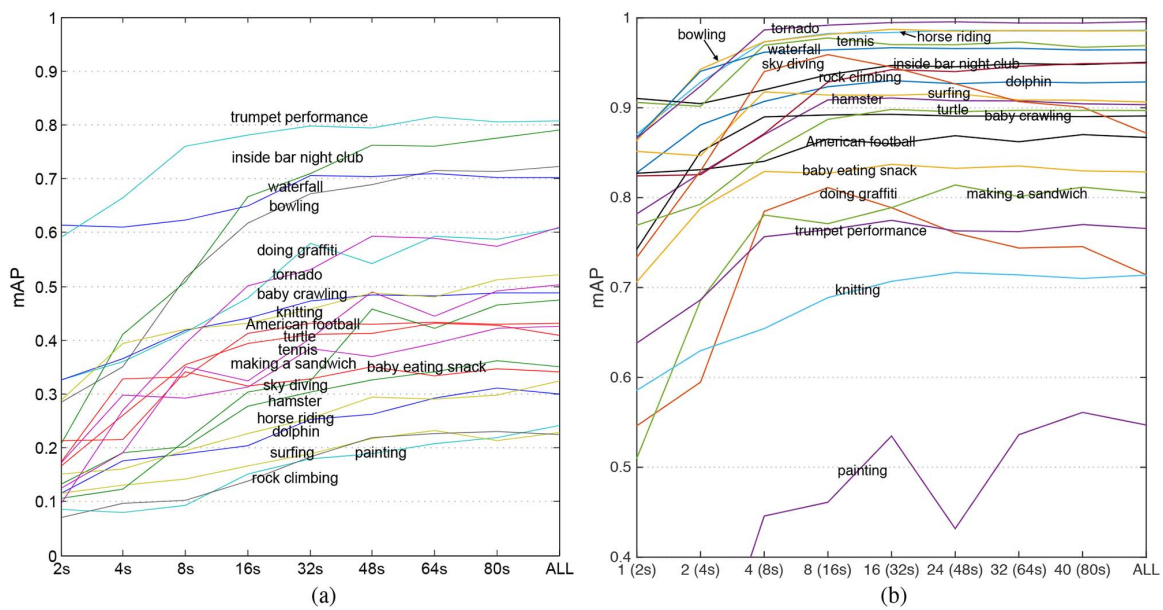


Fig. 9. Per-category accuracies of CV20, using different numbers of uniformly sampled frames. This figure is best viewed on screen with magnification. (a) MFCC. (b) CNN.

120 seconds; 50% longer than CCV), continuous sampling is significantly worse than uniform sampling, as more information loss will be incurred when continuously sampling the same number of frames from longer videos.

We further plot the per-category performances with different numbers of uniformly sampled frames in Fig. 9, using the CV20 dataset. We see that the performance trends across different categories are generally consistent. For MFCC, most categories

TABLE II
SUMMARY OF RECOMMENDED SYSTEM OPTIONS FOR FAST EVENT RECOGNITION, IN COMPARISON WITH THE 3-FEATURE BASELINE SYSTEM. THE TIME IS MEASURED IN SECONDS NEEDED FOR CLASSIFYING A 120-SECOND VIDEO, USING MODELS OF THE 20 CATEGORIES IN CV20

	Baseline		SUPER	
Visual frame sampling	–		Max 16	
Audio frame sampling	–		–	
Features	SIFT, STIP, MFCC		CNN, MFCC	
Quantization	Inner Product		Random Forest	
Classifier	χ^2 SVM		fastHI SVM	
Fusion	Late		Early	
	CCV	CV20	CCV	CV20
mAP	0.595	0.868	0.707	0.908
Processing time (120-s video)	655 seconds		1.78 seconds	

show performance degradation immediately when reducing the number of sampled frames. There are a few categories like "trumpet performance" and "waterfall" that do not suffer from audio sampling. After checking the videos in these categories, we find that the audio soundtracks are more consistent throughout these samples, i.e., containing similar sounds from the beginning to the end. For the visual features, the trends are also quite consistent, except two categories "sky diving" and "doing graffiti".

V. SUMMARY OF FINDINGS

We have evaluated many options to improve recognition speed. This section summarizes several important findings.

Selecting suitable features is the most critical part in designing a fast recognition system. We suggest using the audio feature MFCC, fused with a few fast static visual features like CNN and DURF. The motion-based features like STIP and dense trajectories are useful, but are not suggested because they need to process more frames. In terms of accuracy, the results are already good by using only the audio and the static visual features, which is consistent with human recognition as we can easily tell most of the events based on a set of static frames. In addition, the results also indicate that the random forest based quantization method should be adopted with significant speed improvement and no performance degradation.

For classification and fusion, we found that the fastHI SVM is very suitable, generating similar accuracies to the χ^2 SVM particularly under multi-feature fusion settings. The fastHI is much more efficient, with a speed-up of 50-100 times over the χ^2. KRR is also a good option in terms of speed but is slightly worse than SVM. We observed slightly better results from early and kernel fusion than the late fusion. Late fusion is more expensive as more classification models have to be trained.

For the suitable number of audio/visual frames, we found that audio frames should be sampled densely and down-sampling is always harmful. For the visual frames, we can uniformly select just 16 frames per video. This greatly reduces the feature extraction time and similar recognition accuracies can be obtained.

Based on these findings, we summarize the suggested system components for fast recognition in Table II. We name the final system as SUPER, standing for Speeded Up Event Recognition. Significant speed-up is achieved by using these alternative methods. Specifically, SUPER is 368 times faster than the 3-feature baseline as shown in Table II. The speed-up over the 4-feature baseline (with another dense trajectory feature) is as high as 2,350 times. The accuracy of SUPER is higher than the baseline on both datasets because of the strong CNN features. Compared with the extremely slow 4-feature baseline with the additional dense trajectories, the accuracy is also better ($0.669 \rightarrow 0.707$ on CCV). Furthermore, by including a few more efficient features like the DURF, the recognition accuracy may be further boosted with minor additional computation.

VI. CONCLUSION

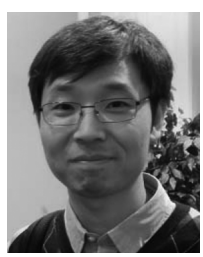
In this paper, we have conducted a comprehensive study on various technical options for event recognition in Internet videos. With the goal of optimizing speed while obtaining a high accuracy, our findings lead to a super efficient system that is 2,350 times faster than a strong baseline system using several widely adopted features. This is extremely important in the era of big video data, where the amount of data is increasing at a faster pace than the power of computational devices. Although event recognition can be processed off-line in most applications, it is difficult to deal with the Web-scale video data even with the most powerful clusters. In addition, this super fast recognition system can also be easily deployed on the less powerful mobile devices for applications like efficient personal video collection management.

The major message delivered in this work is that event recognition in Internet videos can be achieved efficiently, and research in this area should pay more attention to the computational efficiency rather than purely focusing on optimizing the recognition accuracy. While our observations are encouraging, there is still room for further improvements. One important direction is to exploit the CNN features as they are very powerful. Fine-tuning a neural network model using video annotations will probably produce better CNN features than directly using a network trained on images. As the numerous parameters in the neural networks require a huge amount of training samples to be well tuned, a large collection of videos with reliable annotations is needed to support future investigations on this learning paradigm. In addition, it is interesting to study the effectiveness of efficient features from automatic speech recognition (ASR), which contain useful clues different from the standard audio features used in this work.

REFERENCES

[1] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3551–3558.
[2] R. Aly *et al.*, "The AXES submissions at TrecVid 2013," in *Proc. NIST TRECVID Workshop*, 2013.
[3] "TRECVID multimedia event detection track," NIST. Gaithersburg, MD, USA, Dec. 2009 [Online]. Available: http://www.nist.gov/itl/iad/mig/med.cfm/, Accessed on: Jan. 2014
[4] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 1996–2003.
[5] Y.-G. Jiang *et al.*, "Columbia-UCF TRECVID2010 multimedia event detection: Combining multiple modalities, contextual concepts, and temporal matching," in *Proc. NIST TRECVID Workshop*, 2010.
[6] Z.-Z. Lan *et al.*, "CMU-Informedia@TRECVID 2013multimedia event detection," in *Proc. NIST TRECVID Workshop*, 2013.
[7] Y.-G. Jiang, G. Ye, S.-F. Chang, D. Ellis, and A. C. Loui, "Consumer video understanding: A benchmark database and an evaluation of human and machine performance," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2011.

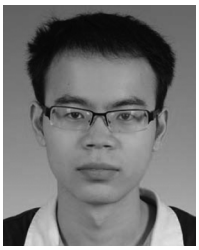
[8] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[9] Y.-G. Jiang, J. Yang, C.-W. Ngo, and A. G. Hauptmann, "Representations of keypoint-based semantic concept detection: A comprehensive study," *IEEE Trans. Multimedia*, vol. 12, no. 1, pp. 42–53, Jan. 2010.
[10] K. Lee and D. P. W. Ellis, "Audio-based semantic concept classification for consumer video," *IEEE Trans. Audio, Speech, Language Process.*, vol. 18, no. 6, pp. 1406–1416, Aug. 2010.
[11] A. Habibian, T. Mensink, and C. G. M. Snoek, "Videostory: A new multimedia embedding for few-example recognition and translation of events," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 17–26.
[12] D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, pp. 91–110, 2004.
[13] H. Bay, A. Ess, T. Tuytelaars, and L. van Gool, "SURF: Speeded up robust features," *Comput. Vis. Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.
[14] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. van Gool, "Hough transform and 3D SURF for robust three dimensional classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 589–602.
[15] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 490–503.
[16] J. R. R. Uijlings, A. W. M. Smeulders, and R. J. H. Scha, "Real-time visual concept classification," *IEEE Trans. Multimedia*, vol. 12, no. 7, pp. 665–680, Nov. 2010.
[17] J. R. R. Uijlings, I. C. Duta, N. Rostamzadeh, and N. Sebe, "Realtime video classification using dense HOF/HOG," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2014, pp. 145–152.
[18] A. Karpathy *et al.*, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 1725–1732.
[19] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," in *Proc. Int. Conf. Mach. Learning*, 2010, pp. 495–502.
[20] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576.
[21] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2006, vol. 2, pp. 2161–2168.
[22] F. Moosmann, E. Nowak, and F. Jurie, "Randomized clustering forests for image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 9, pp. 1632–1646, Sep. 2008.
[23] T.-H. Yu, T.-K. Kim, and R. Cipolla, "Real-time action recognition by spatiotemoral semantic and structural forests," in *Proc. Brit. Mach. Vis. Conf.*, 2010, pp. 52.1–52.12.
[24] J. Shotton, M. Johnson, and R. Cipolla, "Semantic texton forests for image categorization and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[25] N. Inoue and K. Shinoda, "Neighbor-to-neighbor search for fast coding of feature vectors," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1233–1240.
[26] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[27] F. Perronnin, J. Sanchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 143–156.
[28] A. Habibian and C. G. M. Snoek, "Stop-frame removal improves web video classification," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2014, pp. 499–502.
[29] S. Bhattacharya, F. X. Yu, and S.-F. Chang, "Minimally needed evidence for complex event recognition in unconstrained videos," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2014, pp. 105–112.
[30] K. Schindler and L. van Gool, "Action snippets: How many frames does human action recognition require?," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[31] Y.-G. Jiang, "Super: Towards real-time event recognition in internet videos," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2012, pp. 7–14.
[32] Z.-Z. Lan, Y. Yang, N. Ballas, and A. Hauptmann, "Resource constrained multimedia event detection," in *Proc. Int. Conf. Multimedia Modeling*, 2014, pp. 388–399.
[33] Z. Ma, S.-I. Yu, and A. G. Hauptmann, "How to efficiently handle large-scale multimedia event detection," *E-Letter IEEE Multimedia Commun. Tech. Committee*, vol. 9, no. 3, pp. 26–28, 2014.
[34] P. Natarajan *et al.*, "BBN VISER TRECVID 2011 multimedia event detection system," in *Proc. NIST TRECVID Workshop*, 2011.
[35] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang, "Robust late fusion with rank minimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 3021–3028.
[36] K. Mikoljczyk and C. Schmid, "Scale and affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, pp. 63–86, 2004.
[37] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, pp. 107–123, 2005.
[38] B. E. Stein and T. R. Stanford, "Multisensory integration: Current issues from the perspective of the single neuron," *Nature Rev. Neurosci.*, vol. 9, pp. 255–266, 2008.
[39] Y. G. Jiang, C. W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proc. ACM Int. Conf. Image Video Retrieval*, 2007, pp. 494–501.
[40] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," *CoRR*, vol. abs/1403.6382, 2014 [Online]. Available: http://arxiv.org/abs/1403.6382
[41] C. G. M. Snoek *et al.*, "MediaMill at TRECVID 2014: Searching concepts, objects, instances and events in video," in *Proc. NIST TRECVID Workshop*, Nov. 2014.
[42] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative CNN video representation for event detection," *CoRR*, vol. abs/1411.4006, 2014 [Online]. Available: http://arxiv.org/abs/1411.4006
[43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[44] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–8.
[45] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, pp. 145–175, 2001.
[46] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.
[47] A. Torralba, R. Fergus, and W. T. Freeman, "80 million tiny images: A large database for non-parametric object and scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.
[48] U. Brefeld, T. Gaertner, T. Scheffer, and S. Wrobel, "Efficient co-regularized least squares regression," in *Proc. Int. Conf. Mach. Learning*, 2006, pp. 137–144.
[49] F. R. Bach, G. R. Lanckriet, and M. I. Jordan, "Multiple kernel learning, conic duality, and the SMO algorithm," in *Proc. Int. Conf. Mach. Learning*, 2004.
[50] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," *CoRR*, vol. abs/1408.5093, 2014 [Online]. Available: http://arxiv.org/abs/1408.5093



Yu-Gang Jiang received the Ph.D. degree in computer science from the City University of Hong Kong, Kowloon, Hong Kong, in 2009.

From 2008 to 2011, he was with the Department of Electrical Engineering, Columbia University, New York, NY, USA. He is currently an Associate Professor of Computer Science with Fudan University, Shanghai, China. His current research interests include multimedia retrieval and computer vision.

Dr. Jiang is one of the organizers of the annual THUMOS Challenge on Large Scale Action Recognition, and is currently serving as a Program Chair of ACM ICMR 2015. He was the recipient of many awards, including the prestigious ACM China Rising Star Award in 2014.



Qi Dai received the B.Sc. degree in computer science from the East China University of Science and Technology, Shanghai, China, in 2011, and is currently working toward the Ph.D. degree at the School of Computer Science, Fudan University, Shanghai, China.

His current research interests include multimedia retrieval and computer vision.

Tao Mei received the B.E. degree in automation and Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2001 and 2006, respectively.

He is a Lead Researcher with Microsoft Research, Beijing, China. He has authored or coauthored over 100 papers in journals and conferences, and holds 13 U.S. granted patents. His current research interests include multimedia information retrieval and computer vision.

Dr. Mei is an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, *Multimedia Systems*, and *Neurocomputing*. He is the General Co-Chair of ACM ICIMCS 2013 and the Program Co-Chair of IEEE ICME 2015, IEEE MMSP 2015, and MMM 2013. He was the recipient of several awards from prestigious multimedia journals and conferences, including the IEEE Circuits and Systems Society Circuits and Systems for Video Technology Best Paper Award in 2014, the IEEE TRANSACTIONS ON MULTIMEDIA Prize Paper Award in 2013, and the Best Paper Awards at ACM Multimedia in 2009 and 2007.

Yong Rui received the B.S. degree from Southeast University, Dhaka, Bangladesh, the M.S. degree from Tsinghua University, Beijing, China, and the Ph.D. degree from the University of Illinois at Urbana-Champaign, Champaign, IL, USA. He also holds a Microsoft Leadership Training Certificate from the Wharton Business School, University of Pennsylvania, Philadelphia, PA, USA.

He is currently a Senior Director and Principal Researcher with Microsoft Research Asia, Beijing, China, leading research effort in the areas of multimedia search and mining, knowledge mining, and social computing. He has authored or coauthored 16 books and book chapters, and over 100 referred journal and conference papers. He holds 56 issued U.S. and international patents.

Dr. Rui is a Fellow of IAPR and SPIE, and a Distinguished Scientist of ACM. He is an Executive Member of ACM SIGMM, and the founding Chair of its China Chapter. He is the Editor-in-Chief of the *IEEE MultiMedia Magazine*, an Associate Editor of the *ACM Transactions on Multimedia Computing, Communication, and Applications*, a founding Editor of the *International Journal of Multimedia Information Retrieval*, and a founding Associate Editor of IEEE ACCESS. He was an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA (2004–2008), the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (2006–2010), the *ACM/Springer Multimedia Systems Journal* (2004–2006), and the *International Journal of Multimedia Tools and Applications* (2004–2006). He is on the Organizing Committee and Program Committee of numerous conferences. He was General Co-Chair of ACM Multimedia in 2009 and 2014, ACM ICMR in 2006 and 2012, and ICIMCS in 2010, and Program Co-Chair of ACM Multimedia in 2006, Pacific Rim Multimedia in 2006, and IEEE ICME in 2009. He was on the Steering Committees of ACM Multimedia, ACM ICMR, IEEE ICME and PCM.

Shih-Fu Chang is the Richard Dicker Professor and Senior Vice Dean with Columbia University Engineering School, New York, NY, USA. His research interests include multimedia information retrieval, computer vision, signal processing, machine learning, content-based image search, video recognition, image authentication, hashing for large image database, and novel application of visual search in brain machine interface and mobile communication. He has authored or coauthored over 300 peer-reviewed publications, and holds 25 issues patents and technologies licensed to companies.

Dr. Chang is a Fellow of the American Association for the Advancement of Science. He served as the Editor-in-Chief of the *IEEE Signal Processing Magazine* from 2006 to 2008. He was the recipient of the IEEE Signal Processing Society Technical Achievement Award, the ACM Multimedia SIG Technical Achievement Award, the IEEE Kiyo Tomiyasu Award, the IBM Faculty Award, and the Great Teacher Award from the Society of Columbia Graduates.