The Many Shades of Negativity

Zhigang Ma, Xiaojun Chang, Yi Yang, Nicu Sebe, Senior Member, IEEE, and Alexander G. Hauptmann

Abstract—Complex event detection has been progressively researched in recent years for the broad interest of video indexing and retrieval. To fulfill the purpose of event detection, one needs to train a classifier using both positive and negative examples. Current classifier training treats the negative videos as equally negative. However, we notice that many negative videos resemble the positive videos in different degrees. Intuitively, we may capture more informative cues from the negative videos if we assign them fine-grained labels, thus benefiting the classifier learning. Aiming for this, we use a statistical method on both the positive and negative examples to get the decisive attributes of a specific event. Based on these decisive attributes, we assign the fine-grained labels to negative examples to treat them differently for more effective exploitation. The resulting fine-grained labels may be not optimal to capture the discriminative cues from the negative videos. Hence, we propose to jointly optimize the fine-grained labels with the classifier learning, which brings mutual reciprocity. Meanwhile, the labels of positive examples are supposed to remain unchanged. We thus additionally introduce a constraint for this purpose. On the other hand, the state-of-the-art deep convolutional neural network features are leveraged in our approach for event detection to further boost the performance. Extensive experiments on the challenging TRECVID MED 2014 dataset have validated the efficacy of our proposed approach.

Index Terms—Attribute representation, attribute selection, complex event detection, selective fine-grained labeling.

I. INTRODUCTION

VIDEO collections have been proliferating with the advance of social media. To effectively manage these videos, one needs video content analysis that is essentially a task of abstracting the semantics of multimedia data. This procedure is realized by bridging the semantic gap between the low-level features and the high-level descriptions [1], [2].

Earlier work of video analysis focused primarily on surveillance videos, sports videos and etc. [3]–[6]. To be more specific, video content detection is one of the tasks of video analysis by identifying the occurrence of a class of interest from a set of infinite classes. As it is impossible to know all the infinite classes during the training process, building up a discriminative classifying hyperplane is difficult, which in turn makes detection a challenging task. Detection is closely related to many real-world applications. For example, when a user submits a query of parkour to Youtube, the media portal actually detects this from its tremendous video archives and provides the user with the results. Video content detection has evolved from detecting simple objects (e.g. car), scenes (e.g. indoor and outdoor) or human actions (e.g. running) to more meaningful complex events such as parkour in recent years. Complex events are higher level descriptions of multimedia data which build upon necessary objects, scenes and human actions. For example, the event parkour includes several attributes such as people, outdoor, building, and running, jumping, etc. These attributes are the semantic concepts of objects, scenes and human actions.

Detection of these complex events is formidable as they contain many objects, various actions, multiple scenes and have significant intra-class variations. Additionally, they usually take place in much longer video clips where many frames can be uninformative so it is unlikely to infer an event with a single image or a few video frames. As it is practically appealing, researchers have been making great effort on complex event detection. Some work is focused on how to improve the low-level features whereas some work targets the detector learning algorithms. On the other hand, semantic based learning has proved to be promising among which Deep Convolutional Neural Network (DCNN) based methods have shown impressive performance for complex event detection [7]–[10].

As mentioned above, the research on complex event detection includes several directions. In this paper, we particularly focus on the learning process of event detector by exploring fine-grained labeling. Existing detector learning algorithms on complex event detection all treat negative examples equally, as far as we know. Given a closer look, however, we would find out that negative videos actually have different degrees of “negativity”. To better illustrate our argument, we give an example using the event Grooming an animal in Fig. 1. We can see that the first negative video, which is Harlem Shake, is quite clear to be negative. In contrast, it is very difficult to differentiate the second negative video representing Massaging an animal from the positive one. We find many common attributes between this video and the positive video such as animal and hand (object), touching (action). The last negative video is also ambiguous as it has animal and hand but does not look exactly like Grooming an animal. In fact, it is about A day in the life of a cute dog. As
propose to represent each video
ELATED
propose to model
W
propose to learn an embedding from videos and
: THE MANY SHADES OF NEGATIVITY
1559
have proposed a novel DCNN framework
main contributions of our work are as follows:
attained.
the two components are tightly coupled, mutual benefit is
at the same optimizes the fine-grained labels adaptively. As
labels are then integrated into the classifier learning, which
size the fine-grained labels accordingly. The initial fine-grained
label is an extension of our previous work [7] as positive examples are
suggested by the example in Fig. 1, negative videos for an event
are not equally negative. Some are definitely negative so using
0 as their labels during the classifier learning is precise. Some
have positive attributes so assigning them labels with different
values between 1 and 0, a.k.a. fine-grained labels in this paper,
may better characterize them. But manually assigning the fine-
gained labels is laborious, subjective and possibly not accurate
Hence, we study how to adaptively assign fine-grained
labels to negative videos, extracting more discriminating infor-
mation for event detector learning.
As mentioned before, the fine-grained labels of negative ex-
amples depend on the amount of positive attributes they have.
Hence, our approach first represents complex event videos with
a variety of attributes. Then we learn a decisive attribute set
to evaluate the “negativity” of the negative videos and initial-
ize the fine-grained labels accordingly. The initial fine-grained
labels are then integrated into the classifier learning, which
at the same optimizes the fine-grained labels adaptively. As
the two components are tightly coupled, mutual benefit is
attained.
This paper is the extension of our previous work [7]. The
main contributions of our work are as follows:
1) Based on the insights of [7] we leverage attributes to
evaluate the negativity of the negative examples. We show
that combining multiple sources of attributes is a more
efficient way for evaluation in comparison to the approach
in [7] that exploits each attribute source separately.
2) We boost the performance of complex event detection
by adaptively assigning fine-grained labels to negative
examples. The new formulation in this paper is much
faster than our previous formulation in [7], as will be
demonstrated in the experiments.
3) Our new formulation is enhanced by a constraint on the
labels of the positive training examples, which regulates
that the labels of positive examples are minimally changed
when optimizing the fine-grained labels for the negative
examples. This property makes our formulation match the
reality better than the one in [7] as positive examples are
usually carefully collected.
4) The state-of-the-art DCNN features are used for evalua-
tion in the experiments showing that the performance can
be further boosted, which is desirable in practice.

II. RELATED WORK
In this section, the state of the art on complex event detection
is reviewed and we also give a general introduction of visual
attributes for different applications.

A. Complex Event Detection
Complex event detection is an emerging research issue. Com-
pared to abnormality or sports event detection, complex events
usually involve multiple people, objects and scenes in much
longer videos with huge intra-class variations. Yet due to its
much interest to users in real life, researchers have been making
a lot of effort to improve the detection accuracy [11]. In general,
this problem has been tackled in different ways. From the ba-
sic level, a variety of features have been developed for complex
event detection. A new representation that captures the temporal
dynamics of windowed mid-level concept detectors is proposed in
[12]. Lai et al. propose to represent each video as multiple
instances defined as video segments of different temporal in-
tervals and learn an instance-level event detection model [13].
On the other hand, semantic based representation has also been
shown to be promising for complex event detection. Ye et al.
build a large-scale event-specific concept library that is used
as a mid-level representation of complex events in videos [14].
Habibian et al. propose to learn an embedding from videos and
their descriptions by combining correlated term labels to im-
prove complex event detection [15]. In line with the popularity
of DCNN features in computer vision community, several re-
searchers also apply this technique to complex event detection.
For instance, Xu et al. have proposed a novel DCNN framework
that shows the state-of-the-art performance for complex event
detection [16].
On top of various features, SVM is the main learning tool
for event detection [16]–[18]. On the other hand, some other
learning algorithms have been proposed to tackle complex event
detection from different angles. Vahdat et al. propose to model a
video using both global and segment-level features and define
a multiple kernel learning latent SVM to integrate different
feature types [8]. Lai et al. propose to represent each video as
multiple instances, defined as video segments of different
temporal intervals, and learn an instance-level event detection
model [13]. Transfer learning has been exploited by Ma et al.
for complex event detection when there are only few positive
examples [10].

Though various learning algorithms have contributed to com-
plex event detection in different ways, they overlook the possi-
bility of further boosting the performance by utilizing negative
examples in a different manner, i.e., treating them differently. In
this paper, we dwell in the usage of negative examples, which
is an extension of our previous work [7]. Complex events usu-
ally happen in long video clips. The negative examples may

Fig. 1. Example showing that negative videos for an event are not equally
negative.
reassemble the positive examples in many frames, which justifies fine-grained labels that reflect the different degrees of “negativity” of negative examples. We therefore propose to uncover extra informative cues from the fine-grained labels to boost the complex event detection performance. It is worth mentioning that the proposed approach in this paper has two major differences from our former approach. On one hand, the newly proposed approach is much more efficient than our previous method. On the other hand, we re-designed our algorithm by introducing a constraint on the labels of the positive training examples so that the labels of positive examples were not optimized, which is more reasonable. This is a more advanced feature of our algorithm compared to the previous version.

B. Semantic Representation

Complex event detection, as a higher-level of video analysis, could benefit from semantic concepts, which has been proved by recent work. Essentially, using semantic concepts for complex event detection is to learn semantic representation of event videos via a large corpus of concepts covering object, scene and action. A multitude of approaches have been proposed for such purpose and they all seem promising for better performance gain. Merler et al. have introduced semantic model vectors that are learned with many semantic classifiers, each being an ensemble of SVM models trained from thousands of labeled web images, as an intermediate level semantic representation for complex event detection [19]. Habibian et al. study how to create an effective vocabulary for complex event detection by considering the number, type, specificity and quality of the concept detectors [20]. Ye et al. build a large-scale event-specific concept library that is used as a mid-level representation of complex events in videos [14].

Inspired by the promising results on complex event detection with proper exploitation of semantic concepts and the high availability from public resources, e.g., UCF101 [21], we propose to treat the semantic concepts of these data as attributes and learn the attribute representations of complex events. The attribute representations can be exploited to evaluate the “negativity” of negative examples, resulting in the fine-grained labels.

III. FRAMEWORK OVERVIEW

Fig. 2 demonstrates our framework for complex event detection using Grooming an animal as an example. We briefly introduce each step of our approach as follows and leave the details in the next section.

Step 1: We leverage other multimedia sources related to objects, scenes (e.g., some concepts defined in TRECVID Semantic Indexing Task) and actions to train the attribute representations of the event videos. We propose to use multi-source attributes, in other words, multiple datasets as our attribute sources. Technically, binary linear SVM is used for training concept detectors based on these external sources. Then we apply all the concept detectors on the event videos to obtain the confidence scores as the attribute representation.

Step 2: For a specific event, some attributes are particularly indicative while others may be not. Therefore, we select and leverage these indicative attributes to evaluate the “negativity” of negative examples.

Step 3: We represent both the positive and negative videos with the indicative attributes. Among these indicative attributes, we learn a decisive attribute set which is decisive enough to evaluate the “negativity” of negative examples. Specifically, we calculate the distances of the negative examples to the positive ones based on the decisive attribute set. The initial fine-grained labels are then allocated from 0 to 1 w.r.t. the distances.

Step 4: Multiple visual features can be extracted from the videos and be combined with the indicative attributes based representation. Nonlinear mapping such as full rank principal component analysis [22] can be further adopted for its efficacy for video analysis. In this work, we extract the deep convolutional neural network (DCNN) feature and use vlfeat to generate the VLAD representation as the nonlinear mapping.

Fig. 2. Our framework for complex event detection.
Step 5: We jointly optimize the classifier and the initial fine-grained labels. As the labels for positive examples should be consistent with the ground truth, we apply a constraint on the fine-grained labels. The optimized fine-grained labels help us attain extra useful information for learning a more robust classifier.

IV. PROBLEM FORMULATION

In this section, we detail how to initialize fine-grained labels, how to adaptively optimize them, and how to obtain the classifier.

The first step is to evaluate the “negativity” of the negative examples. We realize it by utilizing visual attributes, i.e., representing event videos by visual attributes that are semantic concepts of objects, scenes and human actions. A specific event is usually characterized by several indicative attributes which are capable of differentiating this event from others. Hence, if a negative example has low “negativity” the negative example should also have many of these indicative attributes. Inspired by this, we seek to discover the indicative attributes of an event followed by assessing the “negativity” of the negative examples.

What attributes are indicative for a specific event? Suppose we have $n_p$ positive examples $A_p = \{a_p^1, a_p^2, \ldots, a_p^{n_p}\} \in \mathbb{R}^{d_{attr} \times n_p}$ represented by attributes. The indicative attributes should be those who have consistently higher values over all the positive examples. Let us see an exemplar event Grooming an animal in Fig. 3. The indicative attributes for this event could be animal, fur, hand, etc. which are very informative. Consequently, all the positive examples would have consistently higher values on these attributes. There are also some attributes that are not relevant to the event, e.g., photo and floor in the example and they would show large variance among positive examples. The observation from Fig. 3 indicates that if we calculate $\sum_{(a_p^j, a_p^k) \in A_p} (a_p^j - \frac{1}{n_p} \sum_{j=1}^{n_p} a_p^j)^T (a_p^k - \frac{1}{n_p} \sum_{j=1}^{n_p} a_p^j)$, the dimensions corresponding to the indicative attributes should have lower values (ideally close to zero). Meanwhile, we define an attribute selection matrix $S \in \mathbb{R}^{d_{attr} \times n_p}$ that is supposed to be sparse for selecting the indicative attributes by finding the homogenous pattern across all data. Thus, we propose the following objective function for attribute selection:

$$\min_S \quad Tr(S^T A_p H A_p^T S) + \lambda \|S\|_{2,1}$$

s.t. $S^T S = I$ (1)

where $A_p H A_p^T = \sum_{(a_p^j, a_p^k) \in A_p} (a_p^j - \frac{1}{n_p} \sum_{i=1}^{n_p} a_p^i)^T (a_p^k - \frac{1}{n_p} \sum_{j=1}^{n_p} a_p^j)$, $\|\cdot\|_{2,1}$ is the $\ell_2,1$-norm, $H$ is a centering matrix and $\lambda$ is a parameter. The solution of the above function is similar to the optimization approach proposed in [23]. The attained $S$ is event specific but the same for all videos of one same event.

The following procedure is performed to select the indicative attributes using $S$. We calculate the $\ell_2$ norm of each row of $S$ to get a vector of size $d_{attr}$ corresponding to all attributes. A threshold of $10^{-3}$ is set and we get the indices of those elements in the vector that are larger than the threshold. The indices point to the attributes to be selected. If more than two thirds of indices are obtained we double the threshold and regenerate the indices. The above procedure is iterated until no more than two thirds of indices are obtained. The attributes corresponding to these indices are the indicative attributes and we use them to represent the videos. Then we evaluate the “negativity” of the negative examples. We again take the event Grooming an animal as an example to demonstrate our idea. In Fig. 4, the indicative attributes for Grooming an animal are animal, fur, water, hand, scrub, indoor and etc. The negative video Massaging a horse as shown on the right is similar to the positive video. Only a few attributes such as animal, fur, hand and scrub existing in the negative video make it look like Grooming an animal whereas other attributes, e.g., water and indoor are not important.

In light of this, we learn a decisive attribute set from the indicative attributes. The decisive attribute set is used for evaluating the “negativity” of the negative examples. Suppose $k$ attributes are considered to be decisive. We sort the indicative attributes of the negative videos and select the top $k$ attributes. Next we need to initialize the fine-grained labels for the negative examples. We formulate our strategy based on the fact that negative examples are not equally negative. For instance, suppose animal and hand are the decisive attributes for Grooming an animal. The two negative examples in Fig. 5 both have these two attributes. However, the first video looks very much like Grooming an animal but we can judge more easily that the second one is not Grooming an animal. Hence, to reflect such differences, we calculate the distances between negative examples and positive examples which are represented by the $k$-decisive attributes.
is the Hadamard operator. For two matrices $U$ and $V$ with the same dimensions, the Hadamard product of them is given by $(U \odot V)_{i,j} = (U)_{i,j}(V)_{i,j}$.

In the above function, the first term makes the optimization of $w$, $p$ and $f$ mutually influential and the second term guarantees that $f$ does not deviate drastically from $g$ initialized by our labeling strategy. The joint optimization enables us to add extra useful information from fine-grained labels to the classifier $w$.

Our regression model adaptively optimizes the labels for negative examples but considers that those for positive examples should be consistent with the ground truth. Thus, we name it Selective Fine-grained Labeling Regression (SFLR).

V. OPTIMIZATION

We propose an alternating approach to solve (2).

1) Fixing $f$ to optimize $w$ and $p$:

By defining a diagonal matrix $D$ whose diagonal elements $D_{ii} = f$, the problem in (2) converts to

$$
\min_{w,p} \left\| D(\hat{X}^T w - p) \right\|_2^2 + \alpha Tr \left( (p - y)^T U (p - y) \right)
+ \beta \|w\|_2^2 + \gamma \|f - g\|_2^2.
$$

(3)

By setting the derivative of (4) w.r.t. $w$ to 0, we have

$$
w = (\hat{X} D^2 \hat{X}^T + \beta I)^{-1} \hat{X} D^2 p = K \hat{X} D^2 p
$$

where $I_d \in \mathbb{R}^{d \times d}$ is an identity matrix and $K = (\hat{X} D^2 \hat{X}^T + \beta I_d)^{-1}$. Substituting (5) into (4) and setting its derivative w.r.t. $p$ to 0, we have

$$
p = \alpha J U y
$$

(5)

where $J = (D^2 \hat{X}^T K \hat{X} - I_n) D^2 (\hat{X}^T K \hat{X} D^2 - I_n) + \alpha U)^{-1}$ and $I_n \in \mathbb{R}^{n \times n}$ is an identity matrix.

2) Fixing $w$ and $p$ to optimize $f$:

The problem equals to

$$
\min_f \left\| (\hat{X}^T w - p) \odot f \right\|_2^2 + \alpha Tr \left( (p - y)^T U (p - y) \right)
+ \beta \|w\|_2^2 + \gamma \|f - g\|_2^2.
$$

(6)

Denoting $X^Tw - p = [z^1, ..., z^n]^T$, we define a diagonal matrix $Q$ with its diagonal elements $Q_{ii} = z^i$. In this way, we can rewrite (7) as

$$
\min_f \|Qf\|_2^2 + \gamma \|f - g\|_2^2.
$$

(7)

By setting the derivative of (8) w.r.t. $f$ to 0, we obtain

$$
f = \gamma (Q^2 + \gamma I_n)^{-1} g
$$

(8)

Next, we propose Algorithm 1 to solve the objective function in (2). It can be proved by that the objective function value monotonically decreases in each iteration until convergence using Algorithm 1. See appendix for details. The computational complexity is discussed as follows. For training, the most complex part of Algorithm 1 is optimizing $w$, $p$ and $f$, i.e., solving (5), (6) and (9). We can do it by solving linear equations, which results in $O(d^2)$ for updating $w$; $O(n^2)$ for updating $p$; and $O(n^2)$ for updating $f$. Thus, the complexity is $O(\max(d, n)^3)$. During testing, the prediction is a linear process. Suppose there

Specifically, the Euclidean distance between a negative example and all positive examples is computed and averaged. Once we get the disparate distance of all negative examples to positive examples, we can assign fine-grained labels to negative examples, e.g., $0$, $0.1$,...,0.9. Technically, we sort the negative examples based on the distance in a descending order. The sorted negative examples are partitioned into 10 groups with the first group labeled as 0, the second group as 0.1 until the last group as 0.9. Higher value indicates that they have lower “negativity,” in other words, they resemble positive examples more. Remind that we use attributes from multiple datasets to represent the event videos, for which we need to integrate the fine-grained labels calculated from each dataset. Considering ease of implementation, we average the fine-grained labels from different dataset. Combining the resulted labels and the labels of positive examples (all ones), we get the initial fine-grained labels and denote them as $g$.

However, the initial fine-grained labels may be not accurate enough to reflect the “negativity” of negative examples. As a robust classifier depends on the label information, we consider adaptively optimizing the fine-grained labels together with the classifier learning. Consequently, they are mutually beneficial for achieving a better classifier as well as more accurate fine-grained labels. We denote the adaptively assigned fine-grained labels as $f$ and propose to jointly optimize $f$ and the classifier $w$ based on the feature representation denoted by $\hat{X}$. $\hat{X}$ is obtained as follows. We first extract DCNN features of the training data and then apply the nonlinear mapping (VLAD) to generate the representation which is subsequently fused with the semantic attribute representation.

On the other hand, there is no point to adaptively change the labels of the positive examples. Hence, we should keep the labels of the positive examples as close to their ground truth as possible when optimizing the fine-grained labels for the negative examples. We introduce a selection matrix $U$ whose diagonal element $U_{ii} = \infty$ if $\hat{X}_i$ indicates a positive example and $U_{ii} = 1$ otherwise. By integrating the aforementioned constraint with the learning of $f$ and classifier $w$, we arrive at the following objective function:

$$
\min_{w,p,f} \left\| (\hat{X}^T w - p) \odot f \right\|_2^2 + \alpha Tr \left( (p - y)^T U (p - y) \right)
+ \beta \|w\|_2^2 + \gamma \|f - g\|_2^2.
$$

(2)

In (2), $\alpha$, $\beta$ and $\gamma$ are regularization parameters and $\beta \|w\|_2^2$ is added to avoid over-fitting; $p$ denote the predicted labels; $\odot$
are \( n_{te} \) testing videos, we need \( d n_{te} \) multiplications with the complexity of \( O(d n_{te}) \).

## VI. Experiments

In this section, we test the performance of our method on complex event detection by comparing it to several state-of-the-art classification approaches.

### A. Datasets

The TRECVID MED’14 (MED14) [24] dataset is used in our experiments. MED14 has 20 complex events which are: Bike trick, Cleaning an appliance, Dog show, Giving directions, Marriage proposal, Renovating a home, Rock climbing, Town hall meeting, Winning race without a vehicle, Working on a metal crafts project, Beekeeping, Wedding shower, Non-motorized vehicle repair, Fixing a musical instrument, Horse riding competition, Felling a tree, Parking a vehicle, Playing fetch, Tailgating, Tuning a musical instrument (E021-E040). Standard partitions for training and testing are provided. The training set comprises 100 positive examples and 5,002 negative examples per event. The testing set consists of 23,953 videos.

To measure the “negativity” of the negative examples, we need to represent the videos with semantic concepts, for which we use the following datasets: TRECVID 2014 semantic indexing task (SIN14) dataset (346 concepts) [25], the Google sports dataset (Sports) (478 concepts) [26], the UCF101 dataset (101 concepts) [21], the Do It Yourself (DIY) dataset (1601 concepts) [27] and the Yahoo Flickr Creative Commons (YFCC) dataset (609 concepts) [28]. Technically, for each concept from these sources, we represent the data by improved dense trajectories to get the prediction scores as the semantic representation.

When learning the event detector using our algorithm SFLR, we use two types of feature: the semantic features mentioned above and the state-of-the-art DCNN features introduced in [16]. The following is the detailed procedure of how we extract the DCNN features. Following [16], we extract the DCNN features based on the network architecture released by [30], i.e., the configuration with 19 weight layers. In our work, we extracted the features from the activation of the last pooling layer, the first and second fully-connected layers, which are denoted as pool5, fc6 and fc7 respectively. We sample every five frames in the videos and follow the pre-processing of [31], [32] on DCNN descriptor extraction. We extract the features from the center crop only. DCNN descriptors are extracted using Caffe with the publicly available model [30], and we utilize vflfeat to generate the VLAD representation. We first apply PCA with whitening on the \( \ell \)-2 normalized DCNN descriptors. After that, we utilize 256 components for VLAD as common choices, PCA projections and centers in K-means for VLAD are learned from approximately 256,000 sampled frames in the training set.

### B. Settings

To show the advantage of the proposed method, we primarily compare it to the state-of-the-art results in [16]. The comparison is based on the VLAD representation of the three DCNN features combined with semantic attribute representation. The three DCNN features are pooling layer, fc6 layer and fc7 layer. Linear SVM is used as the baseline. On the other hand, we apply SVM on the semantic attribute representation only to get additional detection results. Furthermore, a state-of-the-art fusion algorithm, i.e., Robust Late Fusion (RLF) [33], is used to attain detection scores with both the DCNN and semantic features; our previous method (GMMC) as presented in [7] leverages the semantic and DCNN features simultaneously. Correspondingly, the newly proposed method in this paper, namely SFLR, is applied using both DCNN and semantic features to get the detection results.

There are two types of parameters. The first one is the size of the decisive attribute set, i.e., \( k \). We observe that the performance is usually good when it is 8 to 15, though there is variation, so we empirically set it as 10 in the experiment. The second type includes the regularization parameters. We tune them from \{0.001, 0.1, 10, 1000\} for all the algorithms and report the best result for each algorithm.

We perform complex event detection on two scenarios in TRECVID MED challenge. One is EK100, for which all the 100 positive examples and 5002 negative examples of each event are used for training. The other one is EK10, for which only 10 positive examples as specified and 5002 negative examples of each event are used for training.

### C. Complex Event Detection Results

All the detection results using different features are listed in Table I. It can be seen that our proposed method SFLR is the most robust algorithm. The other three algorithms are similarly competitive while GMMC has slight advantage over all. We also notice that using semantic attribute representation alone is limited for performance, which indicates that the DCNN features are indeed more discriminative for complex event detection.
<table>
<thead>
<tr>
<th>Feature</th>
<th>SVM</th>
<th>RLJ</th>
<th>GMCC</th>
<th>SPLR</th>
<th>Event ID</th>
<th>100EX</th>
<th>100EX</th>
</tr>
</thead>
<tbody>
<tr>
<td>fpc+semantic</td>
<td>0.207</td>
<td>0.214</td>
<td>0.201</td>
<td>0.218</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>fcp+semantic</td>
<td>0.249</td>
<td>0.233</td>
<td>0.249</td>
<td>0.249</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ftc+semantic</td>
<td>0.188</td>
<td>0.192</td>
<td>0.188</td>
<td>0.216</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>semantic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.302</td>
<td>0.317</td>
<td>0.288</td>
<td>0.311</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.175</td>
<td>0.165</td>
<td>0.175</td>
<td>0.191</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.279</td>
<td>0.285</td>
<td>0.264</td>
<td>0.305</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.165</td>
<td>0.15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.939</td>
<td>0.949</td>
<td>0.939</td>
<td>0.942</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.785</td>
<td>0.739</td>
<td>0.708</td>
<td>0.769</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.417</td>
<td>0.911</td>
<td>0.907</td>
<td>0.924</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.091</td>
<td>0.109</td>
<td>0.087</td>
<td>0.099</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.051</td>
<td>0.036</td>
<td>0.037</td>
<td>0.073</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.028</td>
<td>0.059</td>
<td>0.053</td>
<td>0.071</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.063</td>
<td>0.075</td>
<td>0.061</td>
<td>0.084</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.051</td>
<td>0.036</td>
<td>0.037</td>
<td>0.042</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.028</td>
<td>0.059</td>
<td>0.053</td>
<td>0.071</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.216</td>
<td>0.229</td>
<td>0.195</td>
<td>0.258</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.164</td>
<td>0.162</td>
<td>0.157</td>
<td>0.213</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.189</td>
<td>0.187</td>
<td>0.179</td>
<td>0.264</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.046</td>
<td>0.046</td>
<td>0.050</td>
<td>0.053</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.251</td>
<td>0.214</td>
<td>0.227</td>
<td>0.263</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.231</td>
<td>0.247</td>
<td>0.204</td>
<td>0.276</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.208</td>
<td>0.222</td>
<td>0.237</td>
<td>0.289</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.238</td>
<td>0.238</td>
<td>0.238</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.444</td>
<td>0.462</td>
<td>0.441</td>
<td>0.463</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.505</td>
<td>0.435</td>
<td>0.493</td>
<td>0.535</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.427</td>
<td>0.428</td>
<td>0.425</td>
<td>0.449</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.382</td>
<td>0.382</td>
<td>0.382</td>
<td>0.449</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.360</td>
<td>0.375</td>
<td>0.343</td>
<td>0.377</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.296</td>
<td>0.284</td>
<td>0.269</td>
<td>0.313</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.343</td>
<td>0.343</td>
<td>0.343</td>
<td>0.355</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.247</td>
<td>0.247</td>
<td>0.247</td>
<td>0.264</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.146</td>
<td>0.168</td>
<td>0.175</td>
<td>0.215</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.177</td>
<td>0.166</td>
<td>0.174</td>
<td>0.228</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.148</td>
<td>0.144</td>
<td>0.164</td>
<td>0.218</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0.138</td>
<td>0.138</td>
<td>0.238</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.873</td>
<td>0.889</td>
<td>0.842</td>
<td>0.905</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.795</td>
<td>0.741</td>
<td>0.748</td>
<td>0.811</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.849</td>
<td>0.847</td>
<td>0.852</td>
<td>0.869</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.584</td>
<td>0.584</td>
<td>0.584</td>
<td>0.584</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.316</td>
<td>0.327</td>
<td>0.297</td>
<td>0.351</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.355</td>
<td>0.349</td>
<td>0.323</td>
<td>0.383</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.285</td>
<td>0.292</td>
<td>0.308</td>
<td>0.327</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.584</td>
<td>0.575</td>
<td>0.583</td>
<td>0.625</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.628</td>
<td>0.597</td>
<td>0.599</td>
<td>0.668</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.548</td>
<td>0.548</td>
<td>0.548</td>
<td>0.548</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.447</td>
<td>0.447</td>
<td>0.447</td>
<td>0.447</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.503</td>
<td>0.516</td>
<td>0.499</td>
<td>0.518</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.497</td>
<td>0.497</td>
<td>0.497</td>
<td>0.518</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.483</td>
<td>0.487</td>
<td>0.479</td>
<td>0.474</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.149</td>
<td>0.149</td>
<td>0.149</td>
<td>0.149</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.492</td>
<td>0.503</td>
<td>0.488</td>
<td>0.504</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.453</td>
<td>0.458</td>
<td>0.459</td>
<td>0.503</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.466</td>
<td>0.478</td>
<td>0.479</td>
<td>0.474</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.251</td>
<td>0.251</td>
<td>0.251</td>
<td>0.251</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.247</td>
<td>0.238</td>
<td>0.227</td>
<td>0.233</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.262</td>
<td>0.253</td>
<td>0.248</td>
<td>0.242</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.208</td>
<td>0.202</td>
<td>0.204</td>
<td>0.216</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.251</td>
<td>0.251</td>
<td>0.251</td>
<td>0.251</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.374</td>
<td>0.382</td>
<td>0.371</td>
<td>0.391</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.446</td>
<td>0.458</td>
<td>0.459</td>
<td>0.495</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.352</td>
<td>0.352</td>
<td>0.352</td>
<td>0.352</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.167</td>
<td>0.167</td>
<td>0.167</td>
<td>0.167</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.104</td>
<td>0.106</td>
<td>0.102</td>
<td>0.116</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.072</td>
<td>0.072</td>
<td>0.072</td>
<td>0.087</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.084</td>
<td>0.085</td>
<td>0.085</td>
<td>0.088</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.046</td>
<td>0.046</td>
<td>0.046</td>
<td>0.046</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.487</td>
<td>0.504</td>
<td>0.479</td>
<td>0.498</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.419</td>
<td>0.412</td>
<td>0.412</td>
<td>0.419</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.463</td>
<td>0.474</td>
<td>0.479</td>
<td>0.478</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.403</td>
<td>0.403</td>
<td>0.403</td>
<td>0.403</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.242</td>
<td>0.251</td>
<td>0.262</td>
<td>0.265</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.203</td>
<td>0.204</td>
<td>0.226</td>
<td>0.264</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.158</td>
<td>0.158</td>
<td>0.221</td>
<td>0.249</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.148</td>
<td>0.148</td>
<td>0.148</td>
<td>0.148</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.349</td>
<td>0.370</td>
<td>0.355</td>
<td>0.382</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.316</td>
<td>0.324</td>
<td>0.333</td>
<td>0.363</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.316</td>
<td>0.343</td>
<td>0.343</td>
<td>0.367</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0.241</td>
<td>0.241</td>
<td>0.241</td>
<td>0.241</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Table I**: Comparison of KE100 and KE10 Detection Results. The best results are highlighted in bold.
We summarize our observations as follows: 1) Using pool5 combined with semantic features, SFLR gets the best detection performance on 16 events for EK100 and on 17 events for EK10; SVM, RFL and GMMC obtain similar competitive performance; SFLR attains the highest MAP over all the 20 events for both EK100 and EK10; the advantage of SFLR over other algorithms is more visible for EK10. 2) Using fc6 combined with semantic features, SFLR gets the best detection performance on 16 events for EK100 and on 17 events for EK10; SVM, RLF and GMMC are comparable; SFLR attains the highest MAP over all the 20 events for both EK100 and EK10 with visible improvement over the comparison algorithms. 3) Using fc7 combined with semantic features, SFLR gets the best detection performance on 18 events for EK100 and on 17 events for EK10; GMMC is generally the second best algorithm; RLF also achieves good performance; SFLR attains the highest MAP over all the 20 events for both EK100 and EK10 and the advantage over other algorithms is promising. 4) Using semantic attribute representation alone is generally disadvantageous. 5) Considering the overall MAP, the best result is attained by our method using pool5 combined with semantic features for EK100 whereas our method using fc6 combined with semantic features gets the best results for EK10. For other algorithms, pool5 shows better performance than fc6 and fc7 when combined with semantic features in general. 6) SFLR could further boost the performance over GMMC and RLF since it sophisticatedly learns from negative examples by extracting fine-grained information from them. In this way, we could attain more discriminating informative cues from the training samples, thus resulting in a more robust event detector. The complex event detection results have validated the effectiveness of our approach of utilizing negative examples in a delicate way.

We additionally test the computation efficiency of SFLR and GMMC to see how the new version improves over the old one. It turns out that SFLR is 4.5 times faster than GMMC to train the models. To better visualize the advantage of the new method, we show the running time comparison using the first three events as examples in Fig. 6. The result verifies that the newly proposed approach is also more preferable than the old one for efficiency.

### D. Parameter Sensitivity

Our algorithm has three regularization parameters denoted as α, β and γ in (2). The performance variance w.r.t. the parameters is shown in Fig. 7 using Bike trick as an example. The results are obtained by fixing one parameter while varying the other two. From Fig. 7 we notice that the performance changes corresponding to different combinations of the regularization parameters, which is presumably related to the trait of the data.

On our experimental event, better results are generally obtained when α, β and γ vary from 0.1 to 10.

### E. Using Near-Miss Videos

In TRECVID MED, NIST also provides a set of training samples called near-miss videos in the training set, which is similar to the problem focused in this paper. The difference is that these videos are pre-defined whereas we aim to uncover such characteristic of negative videos in a statistical way. It is interesting to see the performance when near-miss videos are included. We use the near-miss videos in the same way as in [34] for all the comparison algorithms and show the results in Table II. It can be seen that all the algorithms benefit from the exploitation of near-miss videos. However, if we compare Tables I and II, we can see that our method SFLR without near-miss videos still outperforms SVM using near-miss videos. This indicates that our method is effective in mining the negative examples in a statistical way, compared to which SVM using near-miss videos is less robust and needs more manual effort (i.e., pre-define the near-miss videos).

### F. Without Fine-Grained Labels?

In this subsection, we investigate the performance change when the fine-grained labels are not utilized. That being said, all the negative examples are labeled as 0, which is the traditional way of handling negative examples. Our algorithm is equivalent to the ridge regression model in this scenario. We perform this experiments based on all the three types of DCNN features combined with semantic features. The experimental results on EK100 under the same parameter setting are displayed in Fig. 8 in comparison with the results obtained by exploiting fine-grained labels. It can be seen that using fine-grained labels clearly contributes to the performance gain with different features. We have similar observation on EK10. This experiment validates that the proposed fine-grained labeling is helpful for complex event detection.

### G. Examples of Decisive Attributes

In Fig. 9, we show some of the most decisive attributes selected by our method on a few events. We discuss our observations based on each exemplar event. Intuitively, the actions for Bike trick is fundamental for effective detection. Some attributes such as “dirt jumping”, “bike”, “biking acrobatics” selected by our method are all highly related to this event. They are mostly trained from the sports datasets. For Rock climbing, we see that “sport climbing”, “rock climbing”, “person climbing vertically” are top attributes which are quite reasonable. Similarly, these attributes are also trained from sports datasets. In contrast, Town hall meeting is an event with much fewer visible actions. Our method selected “cheering”, “talking”, “speech”, “conference
Fig. 7. Performance variation w.r.t. \( \alpha \), \( \beta \), and \( \gamma \) on Bike trick. (a) \( \alpha \) \& \( \beta \), (b) \( \alpha \) \& \( \gamma \), (c) \( \beta \) \& \( \gamma \).

TABLE II
MAP COMPARISON OF EK100 AND EK10 DETECTION WITH NEAR-MISS VIDEOS

<table>
<thead>
<tr>
<th>Feature</th>
<th>SVM</th>
<th>RLF</th>
<th>GMMC</th>
<th>SFLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>pool5+semantic</td>
<td>0.251</td>
<td>0.259</td>
<td>0.268</td>
<td>0.284</td>
</tr>
<tr>
<td>fc6+semantic</td>
<td>0.238</td>
<td>0.249</td>
<td>0.258</td>
<td>0.291</td>
</tr>
<tr>
<td>fc7+semantic</td>
<td>0.153</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

Fig. 8. Performance comparison of EK100 between leveraging fine-grained labels and not leveraging them. Judged by the t-test, using fine-grained labels is significantly better than not using them.

Fig. 9. Examples of some of the most decisive attributes selected by our method.

room” and “group of people” as the decisive attributes. These attributes well characterize this event and the action attributes are trained from YFCC whereas the object attributes are trained from SIN which is a dataset of many static concepts. These examples show that our method is fairly reasonable.

VII. CONCLUSION
In this paper, we have introduced the exploration on adaptively assigning fine-grained labels to negative videos to better characterize their properties. As a result, more useful informative cues can be obtained, which are incorporated into the
classifier learning for complex event detection. Our approach is summarized by the following steps: attribute representation using multiple video sources; indicative attributes selection; evaluation of “negativity”, and joint optimization of fine-grained labels and event detector learning. In this way, we have added useful knowledge from the fine-grained labels to the classification. A constraint has also been introduced to keep the labels of positive examples unchanged during the optimization. We have conducted experiments on the challenging TRECVID MED 2014 dataset using the state-of-the-art DCNN features. The comparison with several approaches has shown the effectiveness of our method. We should point out that our method depends on attribute representation but current attribute detectors are not yet reliable, which could restrain the performance of our method. Should we develop more accurate attribute detectors in the future, better complex event detection performance can be expected.

ACKNOWLEDGMENT

The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

REFERENCES


Zhigang Ma received the Ph.D. degree in computer science from the University of Trento, Trento, Italy, in 2013. He is currently a Postdoctoral Research Associate with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include machine learning and its applications to multimedia analysis and computer vision.
Xiaojun Chang received the Ph.D. degree in computer science from the Centre for Quantum Computation and Intelligent Systems, University of Technology Sydney, Ultimo NSW, Australia, in 2016. He has been working as a Research Associate with the Language Technology Institute, Carnegie Mellon University, Pittsburgh, PA, USA. His main research interests include machine learning, data mining, and computer vision.

Yi Yang received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2010. From 2011 to 2013, he was a Postdoctoral Research Fellow with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently an Associate Professor with the Center for Quantum Computation and Intelligent Systems, University of Technology at Sydney, Sydney, NSW, Australia. His research interests include multimedia and computer vision.

Nicu Sebe (M’01–SM’11) received the Ph.D. degree in computer science from Leiden University, Leiden, The Netherlands, in 2001. He is currently with the Department of Information Engineering and Computer Science, University of Trento, Trento, Italy, where he is leading the research in the areas of multimedia information retrieval and human–computer interaction in computer vision applications. Dr. Sebe is a Senior Member of ACM and a Fellow of IAPR. He was a General Co-Chair of the IEEE Automatic Face and Gesture Recognition Conference 2008 and ACM Multimedia 2013, and a Program Chair of the ACM International Conference on Image and Video Retrieval 2007 and 2010, and ACM Multimedia 2007 and 2011. He is a Program Chair of ECCV 2016 and ICCV 2017.

Alexander G. Hauptmann received the B.A. and M.A. degrees in psychology from Johns Hopkins University, Baltimore, MD, USA, in 1982 and 1984, respectively, and the M.S. degree 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 currently with the faculty of the Department of Computer Science and the Language Technologies Institute, CMU. From 1984 to 1994, he worked on speech and machine translation, when he joined the Informedia project for digital video analysis and retrieval, and led the development and evaluation of news-on-demand applications. His research interests include several different areas: man–machine communication, natural language processing, speech understanding and synthesis, video analysis, and machine learning.