Signal Processing 140 (2017) 45–52

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

A novel learning-based frame pooling method for event detection

Lan Wang[a], Chenqiang Gao[a,*], Jiang Liu[a], Deyu Meng[b]

[a] Chongqing Key Laboratory of Signal and Information Processing, Chongqing University of Posts and Telecommunications, Chongqing, China
[b] Institute for Information and System Sciences Faculty of Mathematics and Statistics, Xi'an Jiaotong University, Xi'an, China

ARTICLE INFO

Article history:
Received 18 July 2016
Revised 21 March 2017
Accepted 4 May 2017
Available online 5 May 2017

Keywords:
Optimal pooling
Event detection
Feature representation

ABSTRACT

Detecting complex events in a large video collection crawled from video websites is a challenging task. When applying directly good image-based feature representation, e.g., HOG, SIFT, to videos, we have to face the problem of how to pool multiple frame feature representations into one feature representation. In this paper, we propose a novel learning-based frame pooling method. We formulate the pooling weight learning as an optimization problem and thus our method can automatically learn the best pooling weight configuration for each specific event category. Extensive experimental results conducted on TRECVID MED 2011 reveal that our method outperforms the commonly used average pooling and max pooling strategies on both high-level and low-level features.

1. Introduction

Complex event detection aims to detect events, such as "marriage proposal", "renovating a home", in a large video collection crawled from video websites, like Youtube. This technique can be extensively applied to Internet video retrieval, content-based video analysis and machine intelligence fields and thus has recently attracted much research attention [1–5]. Nevertheless, the complex event detection encounters lots of challenges, mostly because events are usually more complicated and undefinable, possessing great intra-class variations and variable video durations, as compared with traditional concept analysis in constrained video clips, e.g., action recognition [6,7]. For example, identical events, as shown in Fig. 1, are entirely distinct in different videos, with various scenes, animals, illumination and views. Even in the same video, these factors are also changing. The above reasons make event detection far from being applicable to practical use with robust performance.

A large number of methods have been proposed to handle this challenging task [8–11]. Generally speaking, the video representation is one of the most important components. For many techniques to extract the video representation, namely feature descriptors, have to be carefully designed or selected for good detection performance. Different from images, video clips can be treated as spatial-temporal 3D cuboids. Lots of spatial-temporal oriented feature descriptors have been proposed and been proved effective, such as HOG3D [12], MoSIFT [13], 3DSIFT [14] and the state-of-the-art improved Dense Trajectory (IDT) [15]. Although these spatial-temporal descriptors can intrinsically describe videos, the 2D image descriptors are still very important for describing videos in the complex event detection community due to two aspects. On one hand, compared with 2D image descriptors, the spatial-temporal feature descriptors usually require larger data storage and higher computational complexity to be extracted and processed. This problem becomes more serious for large scale datasets. On the other hand, the TRECVID Multimedia Event Detection (MED) evaluation track [16] of each year, held by NIST, reveals that combining kinds of feature descriptors, including 2D and 3D features, usually outperforms those of using a single feature descriptor [17].

Profiting from the research development in image representations, a number of good features, including low-level ones of such HOG [18], SIFT [19], and high-level features of such Object-bank [20] along with the recently most successful Convolutional Neural Network (CNN) feature [21] can be directly applied to describe the video. The commonly used strategy is to extract the feature representation for each frame or selected key frames of the video (we will use *frame* hereinafter) and then pool all feature representations into one representation with *average pooling* or *max pooling* [22]. While the max pooling just uses the maximum response of all frames for each feature dimension, the average pooling uses their average value. It is hard to say which one of these two pooling strategies is better. Sometimes, average pooling is better than max pooling and vice versa. The performance heavily depends on the practical application or datasets. The actual strategy is manually choosing the better one through experiments conducted on a validation set. Therefore, intuitively, here comes two questions: 1) can we automatically choose the better one between the two

* Corresponding author.
E-mail address: gaocq@cqupt.edu.cn (C. Gao).

http://dx.doi.org/10.1016/j.sigpro.2017.05.005
0165-1684/

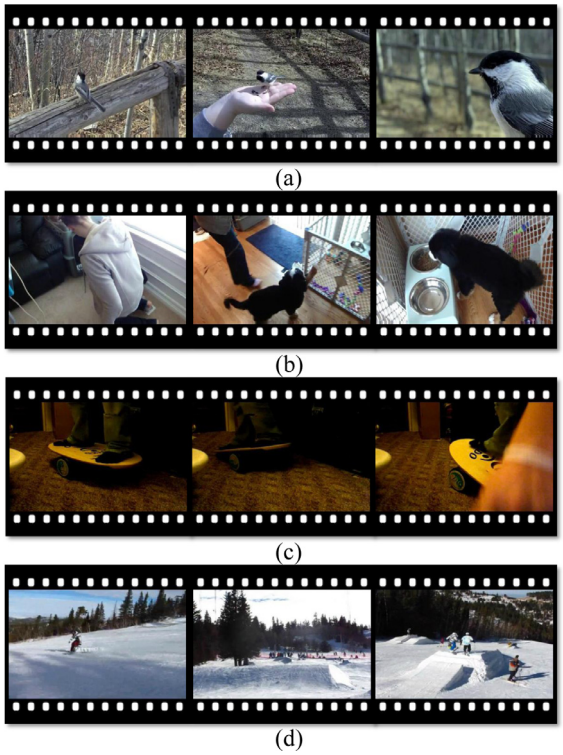

Fig. 1. Example frames of video clips in different events: **(a) (b)** Instances of the "Feeding an animal" event. **(c) (d)** Instances of the "Attempting a aboard trick" event.

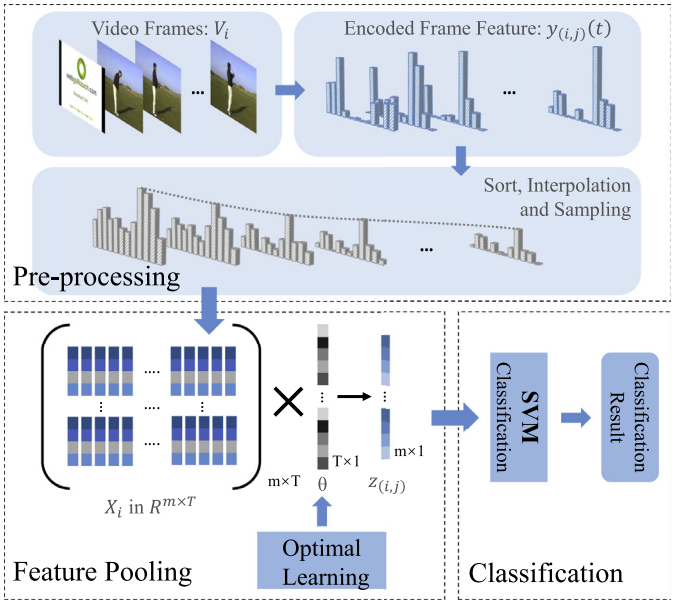

Fig. 2. The overall framework of our method. Given an input video V_i, the encoded frame level features $y_{(i,j)}(t)$ are firstly extracted. They are then sorted in descent order followed by Lagrange interpolation and are sampled for a feature matrix X_i including T frames. We employ the alternatively learned optimal pooling weight θ and the classifier parameters on X_i for the final classification over X_i.

previous pooling strategies? 2) is there any pooling method superior to these two strategies?

To answer these two questions mentioned above, we propose a novel learning-based frame pooling method. We notice that when human beings observe different events, they usually have different attention on various frames, i.e., the pooling weight for a particular event is inconsistent with the others. This phenomenon inspires us to adaptively learn the optimal pooling way from data. In other words, our approach can automatically derive the best pooling weight configuration for each specific event category. To this end, we design an alternative search strategy, which embeds the optimization process for frame pooling weight and classifier parameters into a unifying optimization problem. Experimental results conducted on TRECVID MED 2011 reveal that our learning-based frame pooling method outperforms the commonly used average pooling and max pooling strategies on both high-level and low-level image features.

The rest part of this paper is organized as following. In Section 2, we present our proposed methodology for video description task. Section 3 shows the experimental results with various low-level and high-level features. The conclusion is finally given in Section 4.

2. The proposed method

2.1. Overview of our framework

Our proposed algorithm consists of three main modules: pre-processing, feature pooling and classification, as shown in Fig. 2.

During the pre-processing stage, we extract the features of all frames and then independently sort feature values of all dimensions of the feature vectors in descent order. Then, the Lagrange interpolation and sampling operations are conducted on each video with different frames, to get fixed number features. In pooling stage, we pool the sampled features on interpolation functions with weights obtained by our learning method which will be described below. Finally, a classifier is employed for the event detection results.

2.2. Pre-processing

Our goal is to learn a uniform feature pooling weight setting for each specific event. However, the number of features extracted from videos are different due to different key frames or different video durations when we sample at fixed frame rates. To address this problem, the interpolation operation is adopted.

Given a video clip V_i with T_i frames, we can get T_i encoded feature vectors $y_{(i,j)}(t), t \in (1,2,3,\ldots,T_i), j \in (1,2,3,\ldots,m)$. Note that the feature vectors $y_{(i,j)}(t)$ could either be the raw frame level features like HOG and SIFT, spatial-temporal features like C3D [23], or their encoded counterparts including the Bag-of-Words representation of HOG, SIFT, etc. Hence, m is the dimension of the feature in each frame depending on the feature extractor or the size of codebook used for encoding. Ideally, we may directly construct a Lagrange interpolation function $\tilde{f}_{i,j}(u)$ for the jth feature dimension, i.e., the jth dimension of the input frame level features, as following:

$$\tilde{f}_{i,j}(u) = \sum_{t=1}^{T_i} \frac{\prod_{k=1}^{t-1}(u-k)\prod_{k=t+1}^{T_i}(u-k)}{\prod_{k=1}^{t-1}(t-k)\prod_{k=t+1}^{T_i}(t-k)} y_{i,j}(t), \quad (1)$$

where $\tilde{f}_{i,j}(u)$ can fit all the responses at each time (frame) u in the original video clip. With the interpolated functions for all feature dimensions, we may then re-sample a fixed number of the feature representations. Thus, videos with various durations are eventually able to re-normalized with the same number T of feature representations.

However, we would encounter the "over-fitting" problem if directly conducting interpolating operation on the original encoded features. This is due to the fact that the original feature values from one dimension may varies greatly even between consecutive frames and hence will cause the corresponding interpolation func-

tion to vary dramatically in the feature space. This would produce potential noise data.

For the sake of alleviating this problem, we sort independently all features for each dimension *before* constructing the Lagrange interpolation function. Specifically, for each dimension of m feature dimensions, T feature values are sorted in descent order. In this way, the interpolation function will tend to gradually decreasing in the feature space. Later, we sample along the temporal axis for the jth feature dimension with the interpolation function $\tilde{f}_{i,j}(u)$ denoted as $\bar{x}_{i,j}$:

$$\bar{x}_{i,j} = \left\{\tilde{f}_{i,j}\left(t_k^i\right)\right\}, k \in (1, 2, 3, \ldots, T), \tag{2}$$

where $t_k^i = 1+(k-1)\frac{T_i-1}{T-1}$, are the re-sampling points on the interpolated function. For a given video clip, we combine all sampled feature vectors together into a new feature matrix, denoted as $X_i = (\bar{x}_{i,1}, \bar{x}_{i,2}, \bar{x}_{i,3} \ldots \bar{x}_{i,m})^T \in \mathbb{R}^{m\times T}$.

2.3. Formulation

Given n training samples (X_i, y_i) $(i = 1, 2, \ldots, n)$, where the X_i is the feature matrix obtained by Section 2.2 and y_i is the sample label, our goal is to learn a weight parameter to pool the feature matrix X_i into a single feature vector. Actually, for both average and max pooling methods, the pooling operation is done independently for each feature dimension. Intuitively, we should learn an independent weight vector $\theta^j (j = 1, \ldots, m)$ for each dimension. However, this would make the model too complex to be learned effectively. Instead, we learn a single weight vector θ for all dimensions. Namely, we pool the features using the same weight vector for all feature dimensions as $X_i\theta$. Because our interpolation function $f_{i,j}$ will perform a decreasing property in feature space, we can easily know that the cases of $\theta = (1/T, \ldots, 1/T)$ and $\theta = (1, 0, 0, \ldots, 0)$ approximately correspond to average and max pooling strategies, respectively. Furthermore, the medium and minimum pooling strategies can also be approximately viewed as two specific cases, where $\theta = (0, \ldots, 1, \ldots, 0)$ (1 is located in the middle position of the vector) and $\theta = (0, 0, \ldots, 1)$, respectively. Since our goal is to learn an optimal pooling strategy for each event. To this end, the problem of pooling parameter θ learning is formulated as the following optimization problem:

$$\begin{cases} \min\limits_{w,b,\theta} \sum\limits_{i=1}^{n} \left(1 - y_i\left(w^T X_i \theta + b\right)\right)_+ + \dfrac{1}{2} w^T w, \\ \qquad s.t \quad \theta \geq 0, \sum\limits_{k=1}^{T} \theta_k = 1, \end{cases} \tag{3}$$

where $(\cdot)_+ = max(0, \cdot)$ means the hinge-loss in the loss function. Our model intends to minimize the objective function over w, b, which are the parameters of the hyperplane in the SVM classifier, along with our additional pooling parameter θ.

2.4. Solution

In order to solve the parameters of w, b, θ in Eq. (3) above, an alternative search strategy is employed. In general, our alternative search strategy can be viewed as an iteration approach with two steps in each round. The first step in each iteration is to update w, b with fixed θ by solving the following sub-optimization problem:

$$(w^*, b^*) = \arg\min_{w,b} \sum_{i=1}^{n} \left(1 - y_i\left(w^T X_i \theta + b\right)\right)_+ + \frac{1}{2} w^T w. \tag{4}$$

Here, we initialize θ using random values with constraint that $\theta \geq 0, \sum_{k=1}^{T} \theta_k = 1$. Eq. (4) is the standard formulation of a linear SVM problem and therefore can be solved via off-the-shelf tools like libsvm [24]. The second step in an iteration is to search θ by fixing the w, b obtained by the first step:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{n} \left(1 - y_i\left(w^T X_i \theta + b\right)\right)_+, \quad s.t \quad \theta \geq 0, \sum_{k=1}^{T} \theta_k = 1. \tag{5}$$

Directly solving this optimization problem would be very complex because the hinge loss and the constraints on θ make it a non-convex function. In this degree, a transformation of the above optimization problem needs to be conducted by relaxing the convex property. For a particular video sample V_i in the training set, we introduce a ε_i for it, measuring the corresponding upper bound of the classification error of V_i in the SVM classifier. According to the hinge loss property, the following two conditions are obtained:

$$\varepsilon_i \geq 1 - y_i\left(w^T X_i \theta + b\right), \tag{6}$$

$$\varepsilon_i \geq 0. \tag{7}$$

Eliminating the hinge loss using properties in Eqs. (6) and (7) gives the reformulation of the optimization problem:

$$\begin{aligned} \theta^* = \arg\min_{\theta,\varepsilon} \sum_{i=1}^{n} \varepsilon_i \quad s.t \quad \theta \geq 0, \sum_{k=1}^{T} \theta_k = 1, \\ y_i\left(w^T X_i \theta + b\right) \geq 1 - \varepsilon_i, \varepsilon_i \geq 0, \end{aligned} \tag{8}$$

We can further transform Eq. (8) into a constrained linear programming (LP) problem by defining $\alpha = [\theta, \varepsilon_1, \ldots, \varepsilon_d]$, $\rho = [0, \ldots, 0, 1, \ldots, 1]$, $\sigma = [1, \ldots, 1, 0, \ldots, 0]$, and $\eta_i = [y_i w^T X_i, e_i]$. Then, Eq. (8) can be rewritten as follows:

$$\alpha^* = \arg\min_{\alpha} \rho\alpha^T, \quad s.t \quad \alpha \geq 0, \sigma\alpha^T = 1, \eta_i\alpha^T \geq 1 - y_i b, \tag{9}$$

where d denotes the number of video clips. The number of zeros in ρ equals to the length of vector θ, and then follows by d ones. In other words, ρ plays a role as a selection variable which picks out the ε terms in α. On the opposite, σ selects out the θ from α as well. Therefore, in σ, the number of ones is $\|\theta\|$ and the number of zeros is d. e_i is a vector whose ith element is 1 and the others are zeros. This vector is used to select ε_i. Eq. (9) is a classical linear programming model, which can be optimized using existing tools.

In this way, the objective function in Eq. (3) can be minimized with expected convergence by iteratively searching for w, b and θ, respectively. The total times of iterations is N. The overall algorithm is illustrated in Algorithm 1.

Algorithm 1 Alternative search strategy to obtain optimum w, b, θ.

Input: X_i, y_i (the training set feature matrices and labels),
Output: learned parameter w, b, θ

1. Initialize $\theta^{(0)}$ with random values, $s.t.$ $\theta^{(0)} \geq 0, \sum_{j=1}^{T} \theta_j^{(0)} = 1$;
2. ***for k:=1 to N***
   (a) Fixing $\theta^{(k-1)}$ and updating $w^{(k)}, b^{(k)}$:
   $(w^{(k)}, b^{(k)}) = \arg\min_{w,b} \sum_{i=1}^{n} \left(1 - y_i\left(w^T X_i \theta^{(k-1)} + b\right)\right)_+ + \frac{1}{2} w^T w$;
   (b) Update $\alpha^{(k)}$:
   $\alpha^{(k)} = \arg\min_{\alpha} \rho\alpha^T \quad s.t \quad \alpha \geq 0, \sigma\alpha^T = 1, \eta_i\alpha^T \geq 1 - y_i b^{(k)}$;
   (c) Obtain $\theta^{(k)}$ according to $\alpha^{(k)}$;
   end for
3. Return $w^{(N)}$, $b^{(N)}$ and $\theta^{(N)}$.

Table 1
The definition of 18 events in TRECVID MED 2011.

	Event		Event
E001	Attempting a aboard trick	E010	Grooming an animal
E002	Feeding an animal	E011	Making a sandwich
E003	Landing a fish	E012	Parade
E004	Working on a woodworking project	E013	Parkour
E005	Wedding ceremony	E014	Repairing an appliance
E006	Birthday party	E015	Working on a sewing project
E007	Changing a vehicle tire	P001	Assembling shelter
E008	Flash mob gathering	P002	Batting a run
E009	Getting a vehicle unstuck	P003	Making a cake

3. Experiments

We evaluate our proposed model on the public large scale TRECVID MED 2011 dataset [16] with both low-level features: HOG [18], SIFT [19], high-level features: Object Bank-based feature [20], CNN-based feature [21] and a 3D feature C3D [23], and fused features. We adopt the most popular pooling methods of the *max* and *average* pooling as the baseline methods for comparisons. Meanwhile, we compare our method with several event detection methods evaluated by using the same metric. Finally, we visualize the learned pooling parameters to give more insights.

3.1. Dataset and evaluation metric

The TRECVID MED 2011 development set [16] is used to evaluate our method. It contains more than 13,000 video clips over 18 different kinds of events and background classes, which provides us with real life web video instances consisting of complex events under different scenes lasting from a few seconds to several minutes. The specific events ID and their explanations are listed in Table 1. We follow the original evaluation metric along with the pre-defined training/test splits of MED 2011 development set. In the pre-processing stage, we empirically sample $T = 20$ feature vectors for each video clips based on the interpolation functions. Besides, each learning-based frame pooling model for individual event class is trained with 100 times of iteration (N=100), which enables the objective function to be minimized to convergent. Finally, the average precision (AP) and the mean average precision (mAP) values are used as the evaluation metrics for different pooling approaches. Mean average precision is defined as the mean AP over all events.

3.2. Results on low-level features

We use the off-the-shelf toolkit *VLFeat* [25] to extract HOG and SIFT features with standard configurations for each frame. The HOG features are extracted on each frame with a fixed cell size of 8. The SIFT descriptors are densely sampled on each frame as well. Since the frame size from different V_i may be different, and thus the number of HOG and SIFT descriptors may vary. Therefore, instead of concatenating HOG and SIFT descriptors in each frame directly, the Bag-of-Words method is employed to encode them into a fixed length vector for each frame with 100 dimensions. The codebooks for HOG and SIFT descriptors are generated with the K-means method on the training set, respectively. Finally, the results are listed in Table 2.

From Table 2, it can be obviously observed that our method is effective on most events for both HOG and SIFT features encoded with Bag-of-Words method. For the HOG descriptor, our model leads to apparent AP improvements on 14 out of 18 events, and our learning-based method outperforms the max and average pooling strategies by 0.026 and 0.045 in mAP, respectively. As to the SIFT descriptor, the APs of overall 12 out of 18 events are improved by our method and our method outperforms the max and average pooling strategies by 0.008 and 0.018 in mAP, respectively. It is worth noting that it is very hard to improve mAP, even by 0.01 since the TRECVID MED 2011 is a very challenging dataset.

Table 2
The AP comparison among average pooling, max pooling and our optimal pooling method for low-level features on TRECVID MED 2011 dataset. The bold values indicate the average precision is the highest among different comparison methods.

Event ID	HOG			SIFT		
Method	Average	Max	Ours	Average	Max	Ours
E001	0.407	0.435	**0.457**	0.270	0.275	**0.298**
E002	0.302	0.320	**0.369**	0.207	0.217	**0.223**
E003	0.527	0.511	**0.586**	0.290	0.252	**0.294**
E004	0.279	**0.307**	0.285	0.140	**0.158**	0.130
E005	0.184	**0.217**	0.189	0.142	**0.185**	0.165
E006	0.179	0.175	**0.220**	0.098	**0.145**	0.138
E007	0.083	**0.112**	0.102	0.081	0.076	**0.082**
E008	0.162	0.269	**0.325**	0.197	0.181	**0.201**
E009	0.327	0.357	**0.362**	0.103	0.149	**0.180**
E010	0.151	0.136	**0.180**	0.113	**0.151**	0.125
E011	0.082	0.080	**0.096**	0.085	0.071	**0.112**
E012	0.107	0.144	**0.153**	0.141	0.206	**0.216**
E013	0.110	0.126	**0.130**	**0.107**	0.091	0.104
E014	0.192	0.177	**0.233**	0.150	**0.177**	0.154
E015	0.097	0.104	**0.157**	0.185	0.180	**0.195**
P001	0.123	**0.162**	0.147	0.105	0.129	**0.130**
P002	0.350	0.379	**0.424**	**0.362**	0.344	**0.362**
P003	0.057	0.066	**0.117**	0.058	0.044	**0.065**
mAP	0.207	0.226	**0.252**	0.158	0.168	**0.176**

3.3. Results on high-level features

We test two kinds of high-level features for video frame representation: the CNN-based feature and Object Bank-based feature. When it comes to the CNN-based feature, we directly employ the vgg-m-128 network [26], pre-trained on ILSVRC2012 dataset, to extract feature on each single frame. In detail, we use the 128 dimensional fully connected layer feature as the final feature descriptor, denoted as "CNN 128d". The Object Bank-based descriptor is a combination of several independent "object concept" filter responses, where we pre-train 1000 Object filters on the ImageNet dataset [27]. For each video frame, we employ the maximum response value for each filter as the image-level filter response. Thus, each frame is represented with a 1000 dimensional descriptor, denoted as "Max-OB". The experiment results are listed in Table 3.

Basically, consistent with the low-level feature descriptors, our learning-based pooling method is also effective for both two high-level features on most events. For some specific events, the improvements are large using our method. For example, in *E008*, the event of "Flash mod gathering" for object bank-based feature, our method improves the AP by more than 0.12 compared with average and max pooling methods. Averagely, our method has an improvement of around 0.02 in mAP compared to baseline methods for object bank-based feature, while around 0.002 in mAP for CNN-based feature.

From Tables 2 and 3, we can see that it is hard to determine which one of the baseline methods is better. Their performances rely heavily on the feature descriptors and event types. In contrast, our method performs the best in most cases (and in average).

In addition to using the CNN-based feature and Object Bank-based features as high-level video frame descriptor, we also evaluate the performance of our optimal pooling strategy using the recent proposed deep 3D feature: C3D, as a direct spatial-temporal level descriptor. The C3D is a simple yet effective spatial-temporal feature learned with 3-dimensional convolutional network (3D

Table 3
The AP comparison among different methods for high-level features on TRECVID MED 2011 dataset. The bold values indicate the average precision is the highest among different comparison methods.

Event ID	Max-OB			CNN 128d		
Method	Average	Max	Ours	Average	Max	Ours
E001	0.443	**0.445**	0.436	0.645	0.653	**0.654**
E002	0.321	0.338	**0.403**	**0.394**	0.388	**0.394**
E003	0.191	0.184	**0.216**	0.746	0.745	**0.747**
E004	0.128	0.129	**0.168**	**0.820**	0.818	0.813
E005	**0.153**	0.151	0.131	0.502	**0.590**	0.581
E006	0.370	0.368	**0.384**	0.387	**0.389**	**0.389**
E007	0.077	0.075	**0.132**	0.333	0.323	**0.337**
E008	0.120	0.121	**0.244**	0.423	0.446	**0.461**
E009	0.318	0.320	**0.362**	0.632	0.627	**0.636**
E010	0.124	**0.127**	0.119	0.214	0.269	**0.303**
E011	0.186	0.243	**0.268**	0.250	0.249	**0.252**
E012	0.178	**0.211**	0.183	0.371	**0.425**	**0.425**
E013	0.123	0.110	**0.125**	0.309	**0.327**	0.326
E014	0.175	**0.246**	0.169	**0.384**	0.381	**0.384**
E015	0.210	0.191	**0.219**	0.410	0.410	**0.422**
P001	0.201	0.172	**0.203**	0.426	**0.453**	0.447
P002	0.211	0.198	**0.224**	0.851	**0.956**	0.949
P003	0.118	0.133	**0.144**	0.224	0.219	**0.227**
mAP	0.203	0.209	**0.229**	0.484	0.481	**0.486**

Table 4
The AP comparison among average pooling, max pooling and our optimal pooling method for C3D features on TRECVID MED 2011 dataset. The bold values indicate the average precision is the highest among different comparison methods.

Event ID	C3D		
Method	Average	Max	Ours
E001	0.827	0.827	**0.828**
E002	0.345	0.345	**0.346**
E003	0.817	0.818	**0.819**
E004	0.805	0.805	**0.806**
E005	0.438	**0.439**	0.438
E006	**0.526**	**0.526**	**0.526**
E007	**0.342**	**0.342**	**0.342**
E008	0.728	0.727	**0.729**
E009	0.737	0.737	**0.738**
E010	**0.328**	**0.328**	**0.328**
E011	0.199	**0.200**	**0.200**
E012	0.591	0.591	**0.592**
E013	0.584	**0.585**	0.584
E014	0.432	0.433	**0.434**
E015	**0.394**	0.393	**0.394**
P001	0.464	0.464	**0.466**
P002	0.981	0.981	**0.982**
P003	0.144	**0.145**	**0.145**
mAP	0.538	0.538	**0.539**

ConvNets). It achieves the-state-of-the-art performance on various challenging event detection and recognition datasets, while to the best of our knowledge, has not been evaluated on the TRECVID MED task.

In our experiment, the C3D features are extracted with the released pre-trained model on the sport1m [28] dataset. The input to the network requires a video cuboid, in which the frame size is 128 × 171 pixels and the temporal length is 16 frames. Therefore, we cyclically sample to augment each video clip so that its length could be multiplied by 16. Then a sliding window with length of 16 in temporal domain is employed to scan over the augmented video clip. The fully connected layer features are extracted and then compressed to 128 dimension via Principal Component Analysis for each 3D video cuboid. We follow the strategy to sort, interpolate and sample these cuboid features into 20 feature matrix followed by applying Algorithm 1. The result is listed in Table 4. It is obvious that our optimal pooling approach still has obvious improvement than the traditional average and max pooling counterparts. In 13 out of 18 events, our method achieve better performance. However, we also notice that for event *E006, E007* and *E010*, the performances of three pooling strategy are equal. This implies that, with strong spatial-temporal features, recognizing these three events may not so rely on the absolute value of the feature responses. This is also supported by the mAP in Table 4: compared to "Max-OB" and "CNN 128d" in Table 3, the C3D feature has much higher mAP (around 0.54), which means more powerful representative ability. However, the absolute improvement of our optimal pooling method over the max and the average pooling is 0.001, which is not so significant. The reason may lie on that the C3D feature is already a 3D feature incorporating high-level temporal information. Therefore, the gain from further exploiting its potential in the pooling manner of video cuboids may not be as large as that of pure frame-level features.

3.4. Results on fused features

In addition to using individual low-level and high-level feature separately, it is intriguing to further investigate the ability of our optimal pooling method with their combinations. From Tables 2–4, we find out that the HOG feature obtains the best performance in low-level features, whereas the CNN and C3D features obtain outstanding performance among the high-level counterparts. Thus, we select "HOG+CNN" and "HOG+C3D" feature combinations using the late fusion strategy. The evaluation results are listed in Table 5.

Table 5
The AP comparison among average pooling, max pooling and our optimal pooling method for fused features on TRECVID MED 2011 dataset. The bold values indicate the average precision is the highest among different comparison methods.

Event ID	HOG+CNN			HOG+C3D		
Method	Average	Max	Ours	Average	Max	Ours
E001	0.493	0.498	**0.500**	0.835	0.835	**0.836**
E002	0.388	0.387	**0.395**	0.374	0.374	**0.375**
E003	0.747	0.750	**0.765**	0.825	0.831	**0.851**
E004	0.818	0.818	**0.819**	0.808	0.808	**0.813**
E005	**0.590**	**0.590**	**0.590**	0.491	0.491	**0.492**
E006	0.390	0.390	**0.392**	0.550	**0.551**	**0.551**
E007	0.323	0.323	**0.332**	**0.369**	**0.369**	0.368
E008	0.446	0.446	**0.468**	0.731	0.733	**0.736**
E009	0.627	0.626	**0.638**	0.741	0.740	**0.745**
E010	0.327	0.327	**0.329**	0.335	0.335	**0.336**
E011	0.249	0.249	**0.254**	0.246	0.246	**0.247**
E012	0.425	0.425	**0.429**	0.596	**0.597**	**0.597**
E013	0.269	0.270	**0.294**	0.590	0.590	**0.594**
E014	0.381	0.381	**0.393**	0.450	0.451	**0.456**
E015	0.410	0.410	**0.411**	0.403	0.402	**0.404**
P001	**0.468**	**0.468**	0.467	0.468	0.468	**0.469**
P002	0.950	**0.953**	0.952	0.985	0.985	**0.986**
P003	**0.219**	**0.219**	**0.219**	0.154	0.154	**0.155**
mAP	0.473	0.474	**0.480**	0.553	0.553	**0.556**

As illustrated in Table 5, in most events, the fused "HOG+CNN" feature combination outperforms using single HOG or CNN separately. The similar observation also applies to the "HOG+C3D" combination: the mAP of our optimal pooling strategy is 0.556, which is actually the best performance among all individual features and their combinations. Furthermore, we also observe that the number of events in which the APs are improved through our optimal pooling, increases after fusion. Specifically, in the "HOG+C3D" feature combination, the number of improved events is 17 out of 18 (except *E007*), in contrast to 14 out of 18 in merely using HOG, and 13 out of 18 in merely using C3D. This illustrates that the late fusion strategy can further promote our optimal pooling method positively.

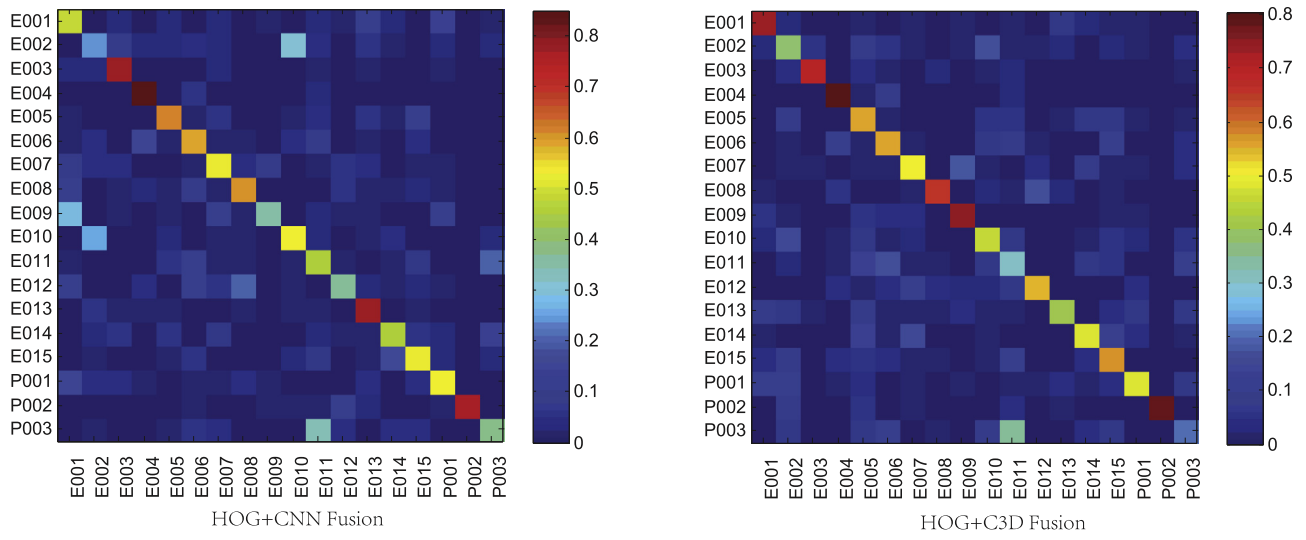

Fig. 3. The confusion matrices on TRECVID MED 2011 dataset with HOG+CNN feature combination **(left)** and HOG+C3D feature combination **(right)**, respectively.

Table 6
The AP comparison among different methods on TRECVID MED 2011 dataset. The bold values indicate the average precision is the highest among different comparison methods.

Event ID	HUT [29]	CRF [30]	EODL [10]	Ours (HOG)	Ours (HOG+C3D)
E001	0.158	0.229	0.188	0.457	**0.836**
E002	0.033	0.078	0.040	0.369	**0.375**
E003	0.189	0.303	0.194	0.586	**0.851**
E004	0.383	0.381	0.410	0.285	**0.813**
E005	0.112	0.215	0.163	0.189	**0.492**
E006	0.046	0.160	0.048	0.220	**0.551**
E007	0.023	0.319	0.077	0.102	**0.368**
E008	0.302	0.441	0.271	0.325	**0.736**
E009	0.035	0.163	0.188	0.362	**0.745**
E010	0.016	0.110	0.131	0.180	**0.336**
E011	0.028	0.143	0.029	0.096	**0.247**
E012	0.082	0.185	0.105	0.153	**0.597**
E013	0.133	0.261	0.220	0.130	**0.594**
E014	0.159	0.270	0.217	0.233	**0.456**
E015	0.014	0.125	0.105	0.157	**0.404**
mAP	0.114	0.226	0.160	0.256	**0.560**

In order to further evaluate the multiple event classification performance, the confusion matrices are drawn using the superior feature combinations certified by the previous experiments, i.e., the "HOG+CNN" and the "HOG+C3D" feature combinations. For a given video, we employ 18 individual event classifiers to vote for its scores and later, the event classifier which giving out maximum classification score assigns its corresponding label for the video. Fig. 3 shows their high precisions in classification: most event videos in the testing set are correctly classified as the groundtruth class, while the error ones tend to randomly distributed across other events with few apparent biases.

In addition, we compare our method to several state-of-the-art event detection methods, including HUT [29], CRF [30], and EODL [10]. All the methods are evaluated on 15 events following the official training and testing set split of TRECVID MED 2011. As shown in Table 6, it obviously that our optimal pooling strategy combined with low-level and spatial-temporal descriptors achieves excellent performance, showing an improvement by 0.334. For a more fair comparison, we list the results of our method using low-level features. As a result, even adopting the simple HOG descriptor, our method outperforms by 0.03 compared with the second best method. This illustrates the effectiveness of our learning-based frame pooling method.

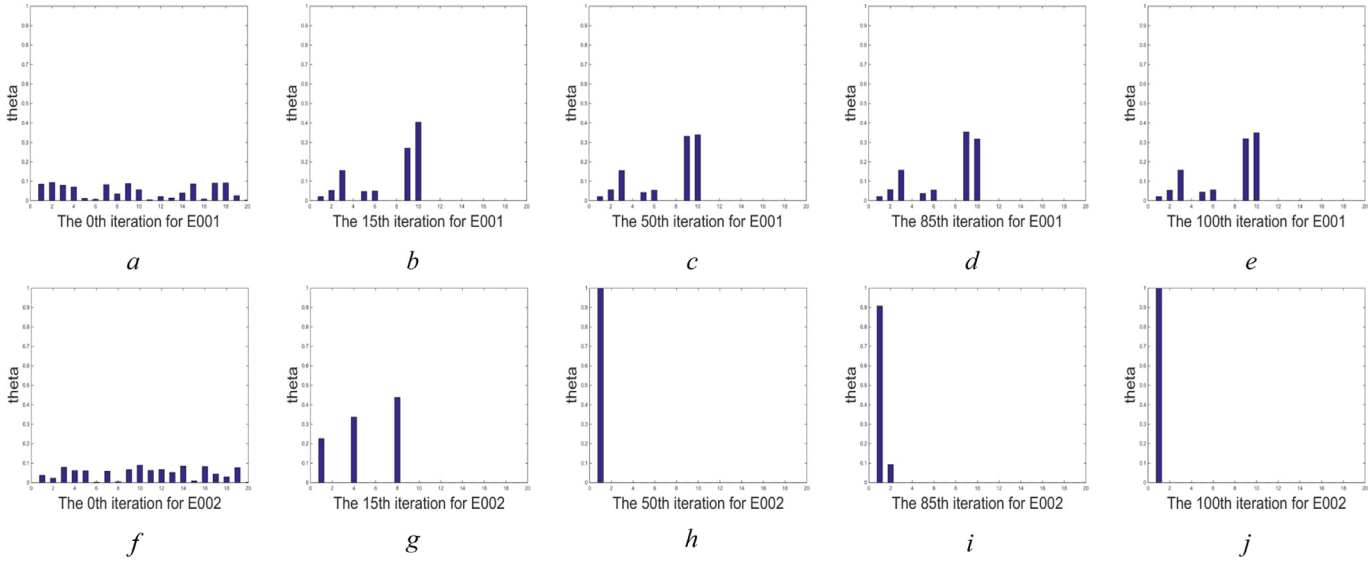

Fig. 4. The learned weights of *E001* and *E002* within 100 iterations. **(a)–(e)**: The learned weights in the 0th (the initial weights), 15th, 50th, 85th and 100th iteration for event E001. **(f)–(j)**: The learned weights in the 0th, 15th, 50th, 85th and 100th iteration for event E002.

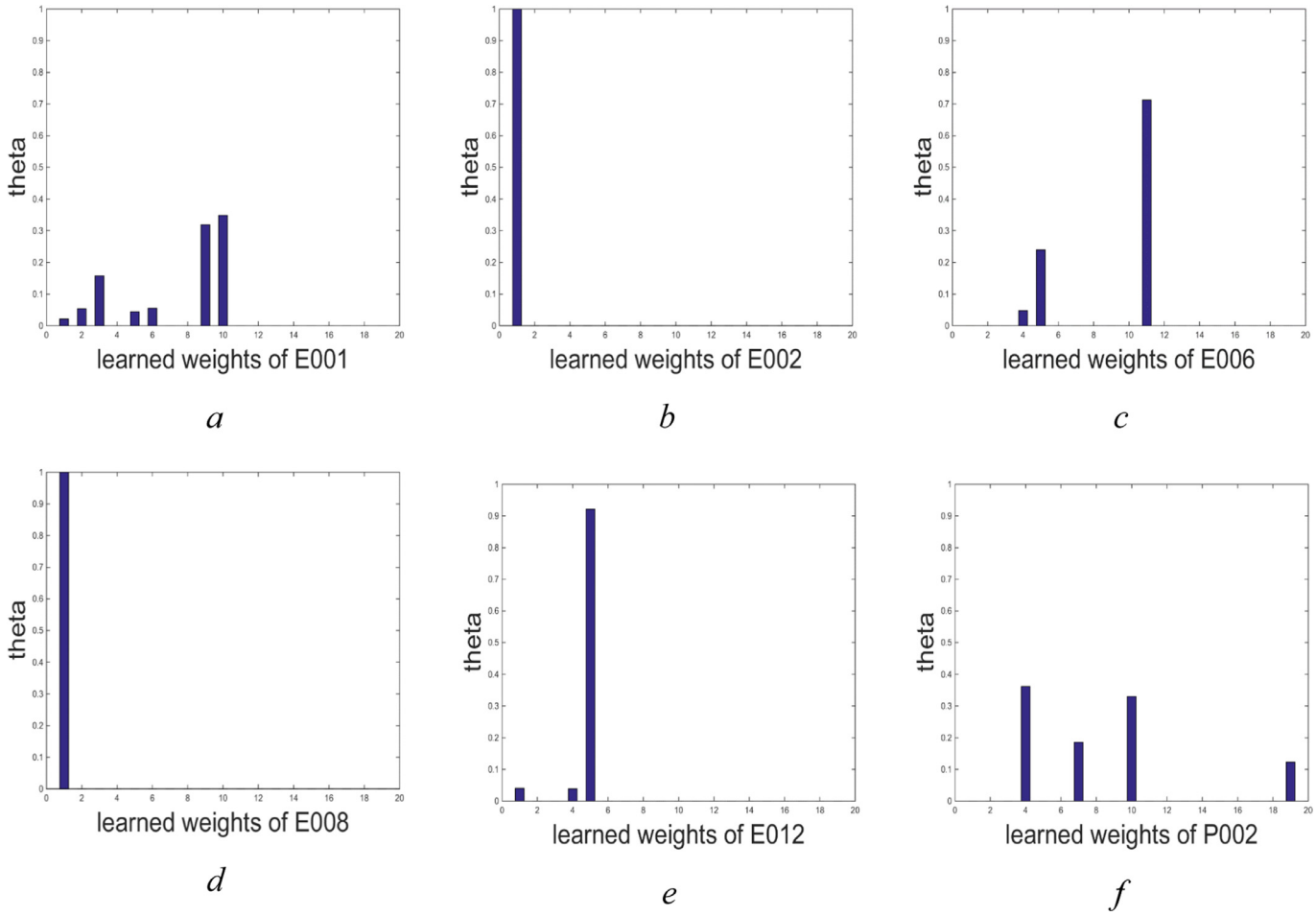

Fig. 5. The visualization results of learned weights from the last iteration in different events.

3.5. Visualization of the learned pooling parameters

In order to get an intuitive observation on the learned pooling parameters, we plot the parameter values of θ in several iterations of the training process. In detail, we visualize the pooling parameters in the models of *E001, E002* learned with the HOG descriptor. Fig. 4 shows the varying trend of these pooling parameters within the 100 iterations. The results illustrate that the weights in some feature dimensions increase quickly from the beginning, and then converge to stable values. In *E002*, the weights gradually concentrate in the first dimension, which reveals the max pooling strategy is an optional choice. As to the event *E001*, the medium feature responses, i.e., the 9th and 10th dimensions in θ, play a leading role. It implies that the most crucial weight for event detection is not always appearing in the order of the max pooling, the min pooling, or the average pooling.

We additionally compare and visualize the learned weights on different events in the last iteration. The visualization results are shown in Fig. 5. In event *E002*:"Feeding an animal" and event *E008*: "Flash mob gathering", features with maximum responses carry the greatest significance in θ. In other words, they obtain quite similar weights to the max pooling. However, the learned weights are not identical to that of max pooling, since other dimensions in θ also get values, whereas they are too small to be seen. This is also proved in Table 2: max pooling yields better mAP than average pooling for *E002* and *E008* with HOG features. In these cases, the optimal weights further outperform max pooling because of those tiny values in dimensions except the max one, i.e. the first dimension. Besides, the weight distributions of events *E001, E008, E012* and *P002* are scattered, reflecting that each dimension in θ are indispensable. Overall, the weight distributions of different events varies, which illustrates that the focus of video clip changes with event categories.

4. Conclusion

In this paper, we propose a learning-based frame pooling method to address the complex event detection task. Compared with commonly used average pooling and max pooling approaches, our method can automatically derive the pooling weight among frames for each event category. Through visualization, the learned weights reveal that weight distributions differ in all event categories. Even more, in each event, trivial weight components are also non-ignorable. Extensive experimental results demonstrate that our approach is more effective and robust for both low-level and high-level image descriptors compared with traditional pooling methods.

Acknowledgment.

This work is supported by the National Natural Science Foundation of China (No. 61571071), the Natural Science Foundation of Chongqing Science and Technology Commission (No. cstc2014jcyjA40048), Wenfeng innovation and start-up project of Chongqing University of Posts and Telecommunications (No. WF201404).

References

[1] K. Tang, L. Fei-Fei, D. Koller, Learning latent temporal structure for complex event detection, in: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 1250–1257.

[2] Z.-Z. Lan, L. Jiang, S.-I. Yu, S. Rawat, Y. Cai, C. Gao, S. Xu, H. Shen, X. Li, Y. Wang, et al., Cmu-informedia at trecvid 2013 multimedia event detection, in: TRECVID 2013 Workshop, vol. 1, 2013, p. 5.

[3] C. Gao, D. Meng, W. Tong, Y. Yang, Y. Cai, H. Shen, G. Liu, S. Xu, A.G. Hauptmann, Interactive surveillance event detection through mid-level discriminative representation, in: Proceedings of International Conference on Multimedia Retrieval, ACM, 2014, p. 305.

[4] Z. Xu, Y. Yang, A.G. Hauptmann, A discriminative cnn video representation for event detection, arXiv:1411.4006 (2014).

[5] L. Yang, C. Gao, D. Meng, L. Jiang, A novel group-sparsity-optimization-based feature selection model for complex interaction recognition, in: Computer Vision–ACCV 2014, Springer, 2015, pp. 508–521.

[6] Y. Yang, R. Liu, C. Deng, X. Gao, Multi-task human action recognition via exploring super-category, Signal Process. 124 (2016) 36–44.

[7] A.-A. Liu, W.-Z. Nie, Y.-T. Su, L. Ma, T. Hao, Z.-X. Yang, Coupled hidden conditional random fields for rgb-d human action recognition, Signal Process. 112 (2015) 74–82.

[8] X. Chang, Y. Yang, G. Long, C. Zhang, A.G. Hauptmann, Dynamic concept composition for zero-example event detection, arXiv:1601.03679 (2016).

[9] Y. Yan, H. Shen, G. Liu, Z. Ma, C. Gao, N. Sebe, Glocal tells you more: coupling glocal structural for feature selection with sparsity for image and video classification, Comput. Vision Image Understanding 124 (2014) 99–109.

[10] Y. Yan, Y. Yang, D. Meng, G. Liu, W. Tong, A.G. Hauptmann, N. Sebe, Event oriented dictionary learning for complex event detection, Image Process. IEEE Trans. 24 (6) (2015) 1867–1878.

[11] Z. Ma, Y. Yang, N. Sebe, A.G. Hauptmann, Knowledge adaptation with partiallyshared features for event detectionusing few exemplars, Pattern Anal. Mach. Intell. IEEE Trans. 36 (9) (2014) 1789–1802.

[12] A. Klaser, M. Marszałek, C. Schmid, A spatio-temporal descriptor based on 3d–gradients, in: BMVC 2008-19th British Machine Vision Conference, British Machine Vision Association, 2008. 275–1

[13] M.-y. Chen, A. Hauptmann, Mosift: recognizing human actions in surveillance videos (2009).

[14] P. Scovanner, S. Ali, M. Shah, A 3-dimensional sift descriptor and its application to action recognition, in: Proceedings of the 15th International Conference on Multimedia, ACM, 2007, pp. 357–360.

[15] H. Wang, A. Kläser, C. Schmid, C.-L. Liu, Dense trajectories and motion boundary descriptors for action recognition, Int. J. Comput. Vision 103 (1) (2013) 60–79.

[16] A.F. Smeaton, P. Over, W. Kraaij, Evaluation campaigns and trecvid, in: MIR '06: Proceedings of the 8th ACM International Workshop on Multimedia Information Retrieval, ACM Press, New York, NY, USA, 2006, pp. 321–330, doi:10.1145/1178677.1178722.

[17] Z.-Z. Lan, L. Jiang, S.-I. Yu, C. Gao, S. Rawat, Y. Cai, S. Xu, H. Shen, X. Li, Y. Wang, et al., Informedia e-lamp@ TRECVID 2013: multimedia event detection and recounting (MED and MER) (2013).

[18] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, vol. 1, IEEE, 2005, pp. 886–893.

[19] D.G. Lowe, Object recognition from local scale-invariant features, in: Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, vol. 2, IEEE, 1999, pp. 1150–1157.

[20] L.-J. Li, H. Su, L. Fei-Fei, E.P. Xing, Object bank: a high-level image representation for scene classification & semantic feature sparsification, in: Advances in Neural Information Processing Systems, 2010, pp. 1378–1386.

[21] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: a deep convolutional activation feature for generic visual recognition, arXiv:1310.1531 (2013).

[22] Y.-L. Boureau, J. Ponce, Y. LeCun, A theoretical analysis of feature pooling in visual recognition, in: Proceedings of the 27th International Conference on Machine Learning (ICML-10), 2010, pp. 111–118.

[23] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning spatiotemporal features with 3d convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4489–4497.

[24] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27.

[25] A. Vedaldi, B. Fulkerson, Vlfeat: an open and portable library of computer vision algorithms, in: Proceedings of the International Conference on Multimedia, ACM, 2010, pp. 1469–1472.

[26] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: delving deep into convolutional nets, in: British Machine Vision Conference, 2014.

[27] J. Deng, A.C. Berg, S. Satheesh, H. Su, A. Khosla, L. Fei-Fei, Imagenet large scale visual recognition challenge (ILSVRC) 2012, 2012.

[28] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, L. Fei-Fei, Large-scale video classification with convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1725–1732.

[29] A. Vahdat, G. Mori, Handling uncertain tags in visual recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 737–744.

[30] K. Tang, B. Yao, L. Fei-Fei, D. Koller, Combining the right features for complex event recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2696–2703.