Complex event recognition using constrained low-rank representation

Afshin Dehghan a,⁎, Omar Oreifej b, Mubarak Shah a

a University of Central Florida, Orlando, FL, United States
b University of California, Berkeley, CA, United States

Abstract

Complex event recognition is the problem of recognizing events in long and unconstrained videos. In this extremely challenging task, concepts have recently shown a promising direction where core low-level events (referred to as concepts) are annotated and modeled using a portion of the training data, then each complex event is described using concept scores, which are features representing the occurrence confidence for the concepts in the event. However, because of the complex nature of the videos, both the concept models and the corresponding concept scores are significantly noisy. In order to address this problem, we propose a novel low-rank formulation, which combines the precisely annotated videos used to train the concepts, with the rich concept scores. Our approach finds a new representation for each event, which is not only low-rank, but also constrained to adhere to the concept annotation, thus suppressing the noise, and maintaining a consistent occurrence of the concepts in each event. Extensive experiments on large scale real world dataset TRECVID Multimedia Event Detection 2011 and 2012 demonstrate that our approach consistently improves the discriminativity of the concept scores by a significant margin.

© 2015 Published by Elsevier B.V.

1. Introduction

The increasing popularity of digital cameras has been creating a tremendous growth in social media websites like YouTube. Along with the increased number of user-uploaded videos, the need to automatically detect and recognize the type of activities occurring in these videos has become crucial. However, in such unconstrained videos, automatic content understanding is a very challenging task due to the large intra-class variation, dynamic and heterogeneous background, and different capturing conditions. Therefore, this problem has recently gained significant attention.

Most activity recognition methods are developed for constrained and short videos (5–10 s) as in [3–6]. These videos contain simple and well-defined human actions such as waving, running, and jumping. In contrast, in this paper, we consider more practical videos with realistic events, complicated contents, and significantly variable lengths. Refer to Fig. 2. The standard activity recognition methods do not incorporate evidences for detecting a particular action/event when deal with such unconstrained videos. To this end, the most recent approaches resorted to using low-level events called “concepts” as an intermediate representation [2,1]. In that, a complex event is described using concept scores, which is the occurrence confidence for the concepts in the video. For example, the event Birthday Party can be described as the occurrence of singing, laughing, blowing candles, jumping … etc.

In the context of concept-based event representation, substantial consequences arise as a result of the complex nature of these unconstrained videos. First: The examples used to train each concept have significant variations, and thus the resulting concept models are noisy. Second: The concept-content of each event may still vary among the samples of each event, mainly because of the variable temporal lengths and capturing conditions. Therefore, the obtained concept scores used to describe each event are also significantly noisy. Third: The automatically obtained concept representation strictly relies on local visual features, and lacks context and semantic cues, which humans naturally infer (refer to Fig. 1). In this paper, we address these consequences using a novel low-rank formulation, which combines the precisely annotated videos used to train the concepts, with the rich concept scores. Our approach is based on two principles: First, the videos of the same event should share similar concepts, and thus should have consistent responses to the concept detectors. Therefore, the matrix containing the concept scores of each event must be of low-rank. Second, since the videos used to train the concept models were manually and carefully annotated, the resulting low-rank matrix should also follow the annotation. For example, concepts like person falling or person flipping may falsely fire in the Birthday Party event, and concepts like person opening present or person dancing may not fire in some videos of events like Birthday Party where these concepts actually occurred. Therefore, by enforcing our constraints, such hurdles can be avoided.

Fig. 3 summarizes the steps involved in our method. We split the training data into two sets; (1) the event-level annotated data, which
has only event labels and (2) the concept-level annotated data, which has both event-level and concept-level labels. We use the concept-level annotated data to train concept detectors, which we run on the event-level annotated data and obtain their concept scores. Consequently, we stack the concept scores for each event in a matrix and find their low-rank representation such that it also follows the basis of the concept annotation. The resulting training data combines the two sets in one rich and consistent training set.

The low-rank constraint has been vigorously employed in different computer vision problems such as tracking [7], feature fusion [8], face recognition [9], and saliency detection [10]. However, to the best of our knowledge, low-rank estimation of concept scores has never been used before. More importantly, our formulation is more general than the standard RPCA [11] in that we allow the estimated low-rank matrix to follow a prior pattern (the annotation in our scenario). On the other hand, since we exploit the low-rank constraint, our method is more robust against noisy concepts and cluttered background than [2,1,12], and significantly outperforms the state-of-the-art, as we demonstrate in the experiments.

The main contribution of this paper is a novel low-rank formulation, through which we find a new representation for each event, which is not only low-rank, but also constrained by the concept annotation, thus suppressing the noise, and maintaining a consistent occurrence of the concepts in each event. Our constrained low-rank representation is not restricted to a certain type of features; which allows us to employ a combination of state-of-the-art features, including STIP [6], DTF-MBH [3], and DTF-HOG [3].

The rest of the paper is organized as follows: Section 2 reviews the related work. Section 3 describes the process of computing the constrained low-rank event representation. In Section 4, we describe how to find the optimal solution for the low-rank optimization. The experimental results are presented in Section 5. Finally, Section 6 concludes the paper.

2. Related work

Compared to the traditional action recognition, complex event recognition is more challenging, mainly because the complex events have significantly longer lengths and diverse contents. Early methods for complex event recognition used low-level features such as SIFT [13], MBH [3], and MFCC [6], and showed promising results as in [14–16]. Additionally, pooling of these low-level features was proposed in [17], where features such as SIFT and color were fused in order to improve the complex event recognition. Moreover, [12] used seven different low-level features in a Bag-of-Words (BoW) framework. However, these approaches reveal limitation in representing semantic concepts because they seek high-level class labels using only low-level features.

On the other hand, concept-based complex event recognition [4,18,1,2,19,20,37,38] has recently flourished and shown promising results in high-level semantic representation of videos with complicated contents. This approach is particularly appealing for the purposes of retrieval and filtering of consumer media. The semantic concepts inherently represent the building blocks of the complex events; therefore, they naturally fit the complex event recognition task. For instance, in [18]
and [4], a large dataset is collected to train concept detectors. However, their concepts are not suitable for complex videos as they have been recorded in well constrained conditions. Additionally, Loui et al. in [18] collected a benchmark dataset containing 25 concepts; however, the concepts are based on static images, not videos. On the other hand, concepts have been also employed in other computer vision problems such as image ranking and retrieval [21], and object classification [22]. In that, the concepts were used in the form of attributes [22], which can be considered as concepts with small granularity [1].

The most recent works on complex event recognition are [2] and [1]. The former utilized 62 action concepts as well as low-level features in a latent SVM framework, while the latter used an unsupervised approach (deep learning [23]) to find the data driven concepts in order to describe the high level semantics of the complex videos. Data driven concepts showed promising performance; however, they do not provide any conceptual description for the video.

On the other hand, low-rank matrix decomposition and completion are recently becoming very popular since Candès et al. in [11] proved that a robust estimation of an underlying subspace can be obtained by decomposing the observations into a low rank matrix and a sparse error matrix. The low-rank decomposition found applications in numerous computer vision problems such as video denoising [24–26], tracking [7], feature fusion [8], face recognition [9], rank aggregation [27], and saliency detection [10]. Evidently, one of the closest approaches to ours is [8], where the low-rank constraint was enforced on the detection scores for the purpose of late fusion of different training models. In our method, we estimate the low-rank subspace of concept scores, which, to the best of our knowledge, has never been used before. More importantly, we not only obtain a low-rank concept representation, but also show how to incorporate the user annotation in order to encourage the low-rank estimation to follow a semantic pattern.

3. Low-rank complex events

Given training event samples \( X = \{x_i\} \) with event labels \( Y = \{y_i\} \), we manually annotate a portion of the training data with 93 predefined low-level events, which occur frequently. These low-level events are called concepts, and they are similar to the concepts used in [1]. Compared to [1] which used 62 action concepts, we selected more concepts in order to cover the events in MED 12 as well. This generates two training subsets: \( M \), which has only event-level annotation (the labels), and \( Z \), which has both event-level and concept-level annotations. A video is annotated by tagging the beginning and ending frames for every concept which occurs along the duration of the video. Therefore, we end up with training clips for each concept, from which we extract low-level features (DTF-MBH, DTF-HOG and STIP) and train a SVM model for each concept. Consequently, each video in the subset \( M \) is tested against each concept model. In that, the video is uniformly divided into clips, and the features are extracted from each clip, then tested on each concept model. We then use max-pooling such that for each concept, the maximum confidence among the clips of the video is stored. This generates a confidence vector of length 93 for each video in subset \( M \), which represents our concept score feature vector.

Similarly, we construct confidence feature vectors for the samples in subset \( Z \) using the concepts’ annotation. For each video in \( Z \), we set the confidence for each concept which was tagged in the video to the maximum possible confidence (we obtain that from subset \( M \)), and we set the confidence for the non-existing concepts to zero. Note that, typically, the manually annotated set \( Z \) is significantly smaller than the automatically obtained set \( M \), because annotating the concepts for a large dataset is often not feasible. We discuss this further in the Experiments section.

We stack the features of the videos from subsets \( M \) and \( Z \) in the rows of matrices \( M \) and \( Z \), respectively. Note the difference between the notation used to refer to the subset of videos, and the notation used to refer to the matrix containing the corresponding features. The training samples in \( M \) and \( Z \) exhibit inherently different cues. The features in \( Z \) are exact and strict because they are collected manually. In contrast, \( M \) has automatically extracted features; therefore, it is noisy, however, contains a rich description of all the details in the videos.

Our goal is to combine the benefits of each subset in one rich and less noisy training set. To the best of our knowledge, this problem has never been explored before. One trivial solution is to simply combine the samples from the two sets and train a SVM on that. In this case, it is likely that the trained model will select support vectors from each of the subsets, and therefore the final decision will be a weighted sum of the dot product between the support vectors and the testing sample (in a linear SVM). However, it is clear that the weighted sum cannot denoise the noisy matrix \( M \), or enrich the highly strict matrix \( Z \). In fact, as we demonstrate later in the experiments, combining the subsets naively can decrease the classifier’s discriminativity.

Since we train a model for each event separately, it is expected that the matrix containing the concept scores of all the samples of an event stacked in the rows will be of low-rank (similarly if stacked in the columns). We exploit this observation in our method. However, in our case, the matrix needs to be not only of low-rank, but also close to the annotation. We formulate this problem for an event \( s \) as

\[
\min_{A_i, \alpha, \beta} \text{Rank}(A_i) \quad s.t. \quad M_i = A_i + \beta, \quad \|E_i\|_0 \leq s, \quad f(A_i, Z_i) \leq \alpha, \tag{1}
\]

where matrices \( M_i \in \mathbb{R}^{m_i \times d} \) and \( Z_i \in \mathbb{R}^{z_i \times d} \) contain the features of event \( i \) extracted from the videos in subsets \( M \) and \( Z \), respectively, and stacked in the rows. \( A_i \in \mathbb{R}^{m_i \times d} \) is the low-rank estimate of \( M_i \), and \( E_i \in \mathbb{R}^{m_i \times d} \) is the error matrix. \( d \) is the dimensionality of the feature vector (93 in our case), and \( m_i \) and \( z_i \) are the number of samples of event \( i \) in subsets \( M \) and \( Z \), respectively. \( s \) and \( \alpha \) are constants which control the relative weights in the minimization. The function \( f : \mathbb{R}^{m_i \times d} \times \mathbb{R}^{z_i \times d} \rightarrow \mathbb{R}^1 \) measures the similarity between the estimated low-rank matrix and the annotation matrix. Since there is no correspondence between the samples in \( M \) and \( Z \), the function cannot be defined as \( \|A_i - Z_i\|_F \), where \( \| \cdot \|_F \) is the Frobenius norm. Note that \( A_i \) and \( Z_i \) can also have different sizes. To this end, we consider the following distance measure

\[
f(A_i, Z_i) = \|A_i - M_i U_i U_i^T\|_F^2 \tag{2}
\]

where matrices \( U_i \in \mathbb{R}^{d \times q} \) is the matrix with the most significant \( q \) principal components of \( Z_i \) obtained by SVD(\( Z_i \)), and stacked in the columns. In other words, Eq. (2) represents the difference between the low-rank estimation \( A_i \) and the components of \( M_i \) along the major directions of the annotation matrix \( Z_i \). Therefore, minimizing Eq. (1) under the constraint in Eq. (2) encourages the low-rank estimation to additionally follow the concept-level annotation. Note that in order to constrain the low-rank solution using the annotation, it is possible to derive other forms of \( f(A_i, Z_i) \). However, as we discuss further in the following section, having the similarity measure in the form \( \|A_i - X\|_F \), where \( X \) is a constant matrix, allows us to obtain a closed form solution at each iteration of the optimization.

Fig. 3 shows an example for the event “Birthday Party” before and after the annotation. It is clear that the final training matrix \( A_i \) is of low-rank and also adopts the patterns in the annotation. In the coming subsection, we discuss our approach to solve the optimization in Eq. (1) using the method of Augmented Lagrange Multiplier (ALM) [28].

4. Optimizing the constrained low-rank problem

Our method decomposes the matrix containing the examples of an event by extracting the noise such that the resulting matrix is both low-rank and follows the concept-annotation. This is achieved
Table 1

Average precision for TRECVID MED 2011 event collection using only concept representation.

<table>
<thead>
<tr>
<th>Event</th>
<th>Base</th>
<th>Naive</th>
<th>SSVM</th>
<th>LR</th>
<th>ACCLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boarding trick</td>
<td>83.7</td>
<td>78</td>
<td>63.7</td>
<td>83.5</td>
<td>83.5</td>
</tr>
<tr>
<td>Feeding animal</td>
<td>69.9</td>
<td>67.8</td>
<td>60.9</td>
<td>71.4</td>
<td>69.4</td>
</tr>
<tr>
<td>Landing fish</td>
<td>56.4</td>
<td>56.8</td>
<td>66.4</td>
<td>58.5</td>
<td>74.3</td>
</tr>
<tr>
<td>Wedding</td>
<td>78.8</td>
<td>76</td>
<td>66.0</td>
<td>85.7</td>
<td>85.5</td>
</tr>
<tr>
<td>Wood project</td>
<td>67.8</td>
<td>66</td>
<td>63.2</td>
<td>71.1</td>
<td>69.6</td>
</tr>
<tr>
<td>Birthday party</td>
<td>73.1</td>
<td>76</td>
<td>59.2</td>
<td>75.5</td>
<td>77.7</td>
</tr>
<tr>
<td>Changing tire</td>
<td>55.2</td>
<td>56</td>
<td>63.2</td>
<td>59.1</td>
<td>74.7</td>
</tr>
<tr>
<td>Flash mob</td>
<td>82.0</td>
<td>84</td>
<td>53.2</td>
<td>85.8</td>
<td>91.1</td>
</tr>
<tr>
<td>Vehicle unstuck</td>
<td>69.8</td>
<td>66</td>
<td>57.6</td>
<td>75.0</td>
<td>57.9</td>
</tr>
<tr>
<td>Grooming animal</td>
<td>78.1</td>
<td>52</td>
<td>59.2</td>
<td>61.4</td>
<td>85.3</td>
</tr>
<tr>
<td>Making sandwich</td>
<td>72.8</td>
<td>52</td>
<td>57.6</td>
<td>69.7</td>
<td>70.3</td>
</tr>
<tr>
<td>Parade</td>
<td>76.0</td>
<td>74</td>
<td>56.7</td>
<td>81.7</td>
<td>82.7</td>
</tr>
<tr>
<td>Parkour</td>
<td>56.4</td>
<td>82</td>
<td>64.0</td>
<td>62.5</td>
<td>69.1</td>
</tr>
<tr>
<td>Repairing appliance</td>
<td>77.6</td>
<td>74</td>
<td>53.0</td>
<td>79.9</td>
<td>80.5</td>
</tr>
<tr>
<td>Sewing machine</td>
<td>71.0</td>
<td>64</td>
<td>57.4</td>
<td>72.9</td>
<td>75.9</td>
</tr>
<tr>
<td>Mean AP</td>
<td>71.1</td>
<td>68.2</td>
<td>60.0</td>
<td>74.4</td>
<td>77.8</td>
</tr>
</tbody>
</table>

using Eqs. (1) and (2) as discussed in the previous section. When solving Eq. (1), it is convenient to consider the Lagrange form of the problem:

$$\min_{A_i} \text{Rank}(A_i) + \lambda \sum_{i} \|E_i\|_1 + \frac{\gamma}{2} \|A_i - M_i U_i U_i^T\|_F^2$$ s.t. \(M_i = A_i + E_i\),

where \(\lambda\) and \(\gamma\) are the weighting parameters. The optimization of Eq. (3) is not directly tractable since the matrix rank and the \(\ell_0\)-norm are nonconvex and extremely difficult to optimize. However, it was recently shown in [11] that when recovering low-rank matrices from sparse errors, if the rank of the matrix to be recovered is not too high and the number of non-zero entries in \(E_i\) is not too large, then minimizing the nuclear norm of \(A_i\) (sum of singular values) and the \(\ell_1\)-norm of \(E_i\) can recover the exact matrices. Therefore, the nuclear norm and the \(\ell_1\)-norm are the natural convex surrogates for the rank function and the \(\ell_0\)-norm, respectively. Applying this relaxation, our new optimization becomes:

$$\min_{A_i} \|A_i\|_* + \lambda \sum_{i} \|E_i\|_1 + \frac{\gamma}{2} \|A_i - M_i U_i U_i^T\|_F^2$$ s.t. \(M_i = A_i + E_i\),

where \(\|A_i\|_*\) denotes the nuclear norm of matrix \(A_i\). We adopt the Augmented Lagrange Multiplier method (ALM) [28] to solve the optimization problem (Eq. (4)), which is currently the method of choice among the computational methods for low-rank optimization [29,30]. Define the augmented Lagrange function for the problem as:

$$L(A_i, E_i, Y_i) = \|A_i\|_* + \lambda \sum_{i} \|E_i\|_1 + \frac{\gamma}{2} \|A_i - M_i U_i U_i^T\|_F^2 + \beta \|Y_i - (M_i - A_i - E_i)\|_F^2,$$

where \(Y_i \in \mathbb{R}^{m \times d}\) is a Lagrange multiplier matrix for event \(i\), \(\beta\) is a positive scalar, and \((\cdot)\) denotes the matrix inner product (trace(\(A^T B\))). Minimizing the function in Eq. (5) can be used to solve the constrained optimization problem in Eq. (4). We use the ALM algorithm to estimate both the Augmented Lagrange multiplier and the optimal solution by iteratively minimizing the augmented Lagrangian function:

$$\begin{align*}
\begin{aligned}
A_i^{k+1} & = \arg \min_{A_i} L(A_i, E_i, Y_i^k) \\
E_i^{k+1} & = Y_i^k - \beta^{-1} \left( M_i^{k+1} - A_i^{k+1} - E_i^{k+1} \right)
\end{aligned}
\end{align*}$$

where \(k\) denotes the iteration number. When \(\beta_k\) is a monotonically increasing positive sequence, the iterations converge to the optimal solution of problem (Eq. (4)) [31]. However, solving Eq. (6) directly is difficult; therefore, the solution is approximated using an alternating strategy minimizing the augmented Lagrange function with respect to each component separately:

$$A_i^{k+1} = \arg \min L(A, E_k, Y_k),$$

$$E_i^{k+1} = \arg \min E_i A_i^{k+1}, E, Y_k.$$
Table 3
Average precision for TRECVID MED 2012 event collection using only concept representation.

<table>
<thead>
<tr>
<th>Event</th>
<th>Base</th>
<th>Naive</th>
<th>SSVM</th>
<th>LR</th>
<th>ACLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bike trick</td>
<td>65.35</td>
<td>63.67</td>
<td>70.15</td>
<td>65.69</td>
<td>69.82</td>
</tr>
<tr>
<td>Cleaning appliance</td>
<td>58.49</td>
<td>71.52</td>
<td>53.81</td>
<td>60.25</td>
<td>66.11</td>
</tr>
<tr>
<td>Dog show</td>
<td>64.33</td>
<td>66.24</td>
<td>56.69</td>
<td>62.89</td>
<td>69.45</td>
</tr>
<tr>
<td>Giving direction</td>
<td>56.57</td>
<td>63.03</td>
<td>47.76</td>
<td>66.60</td>
<td>78.40</td>
</tr>
<tr>
<td>Marriage proposal</td>
<td>58.95</td>
<td>62.46</td>
<td>62.34</td>
<td>70.93</td>
<td>67.73</td>
</tr>
<tr>
<td>Renovating home</td>
<td>64.94</td>
<td>68.39</td>
<td>59.44</td>
<td>65.86</td>
<td>71.59</td>
</tr>
<tr>
<td>Rock climbing</td>
<td>63.05</td>
<td>80.49</td>
<td>64.65</td>
<td>66.55</td>
<td>69.26</td>
</tr>
<tr>
<td>Town hall meeting</td>
<td>67.84</td>
<td>75.30</td>
<td>63.45</td>
<td>60.05</td>
<td>72.56</td>
</tr>
<tr>
<td>Winning a race</td>
<td>67.97</td>
<td>71.66</td>
<td>59.91</td>
<td>68.93</td>
<td>76.15</td>
</tr>
<tr>
<td>Metal project</td>
<td>73.73</td>
<td>66.57</td>
<td>65.09</td>
<td>72.76</td>
<td>69.60</td>
</tr>
<tr>
<td>Mean AP</td>
<td>64.08</td>
<td>68.93</td>
<td>60.33</td>
<td>66.05</td>
<td>71.03</td>
</tr>
</tbody>
</table>

Algorithm 1
Complex event recognition

Input:
- Training samples for event $i$: $M_i \in \mathbb{R}^{n_i \times d}$
- Annotated training samples for event $i$: $Z_i \in \mathbb{R}^{n_i \times d}$

Output:
- Annotation-constrained low-rank estimation $A_i \in \mathbb{R}^{n_i \times d}$

\[ W \Sigma V^T = \text{svd}(Z_i) \]
\[ \{\text{Select the most significant } q \text{ vectors} \} \]
\[ U_i = W(1 \ldots q) \]

while not converged do

\[ W \Sigma V^T = \text{svd}((\beta^k M_i - \beta^k Z_i + Y_i + \tau M_i U_i^T)/\alpha^k) \]
\[ A_i^{k+1} = U_i Z_i V_i \]
\[ Y_i^{k+1} = Y_i^k + \beta^k (M_i A_i^{k+1} - \beta_i^{k+1}) \]
\[ \beta_{i+1} = \rho \beta_i \]
end

5. Experiments

We extensively experimented on our method using a subset of the most challenging multimedia event datasets, TRECVID MED 2011 and 2012, which exhibit a wide range of challenges including camera motion, cluttered background and illumination changes. Inarguably, one of the biggest challenges in these datasets is the significantly varying video length, which ranges from 30 s to over 30 min. The frame rate also ranges from 12 to 30 fps, and the resolution ranges from 320 × 480 to 1280 × 2000. We report our performance using average precision (AP), similar to [12] and [1].

The closest work to our method which directly uses concept scores to tackle the recognition problem is [1]. Therefore, in order to have a fair comparison with [1] and also [2], we selected the same subset of TRECVID data which contains 4062 videos from event collection (2062 for MED 11 and 2000 for MED 12). Similar to [1], we split the data into 70% for training and 30% for testing. The concept models are learned using 700 videos randomly selected from the training set. We found this number of videos sufficient to obtain enough exemplars (at least 40) for training the concept classifiers.

In order to compute the concept scores for a video, we divide it uniformly into overlapping clips, where the clip length is 180 frames, and the step size is 90 frames. Consequently, the confidence scores of each detector are computed for every clip, followed by max-pooling. Thus, our concept score for a video is a 93 dimensional vector, which contains the scores of the concept detectors. Consequently, we process the feature vectors for each event using our constrained low-rank optimization. The inputs to our method are two matrices. The first matrix contains the automatically obtained concept scores of the event. The second matrix contains the manually annotated concept scores of the same event, such that for every annotated video we construct a 93 dimensional vector which is set to zero for all the concepts except the annotated ones. We set the score for the annotated concepts to 0.2, which is the maximum score that a concept can have in our original training matrix.

All the parameters of the low-rank model including $\lambda$, $\tau$, and $q$ are selected using cross validation on the training data. On the other hand, the portion of training videos which are manually annotated with concepts (subset $\mathcal{Z}$) can also be selected by cross validation over the optimal training split. However, this requires manually annotating the whole training set, which is impractical. In our experiments, we use the 700 videos previously selected for training the concept models in order to build the annotation matrix. Note that the constrained low-rank representation is not very sensitive to the number of manually annotated videos. This is because videos of the same event typically share similar concept content such that even a small subset of videos is sufficient to discover the dominant basis of the matrix $Z$. Fig. 4 better illustrates the scalability of the proposed method with respect to the number of annotated videos.

We compare our approach (ACLR) to seven methods:

- Our baseline (Base): Here we use a SVM directly on the obtained concept scores, and skip the constrained low-rank optimization. We refer to this setup in the tables as Base.

Table 4
The average precision for TRECVID MED 2012 event collection using both concept scores and low-level features. Columns 1-3 show the average precision for the low-level features. Columns 4 and 5 show the average precision obtained using our baseline concept models and the proposed one. Column 6 shows the average precision for the combination of low-level features. Column 7 shows the combination of low-level features and baseline concept models. Finally, the last column shows the final result obtained by combining the features from columns 1, 2, 3, and 5.

<table>
<thead>
<tr>
<th>Event</th>
<th>ISA (1)</th>
<th>HOG (2)</th>
<th>STIP (3)</th>
<th>Base (4)</th>
<th>ACLR (5)</th>
<th>(1 + 2 + 3)</th>
<th>(1 + 2 + 3 + 4)</th>
<th>(1 + 2 + 3 + 5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIT</td>
<td>65.52</td>
<td>63.67</td>
<td>67.30</td>
<td>65.35</td>
<td>69.82</td>
<td>74.55</td>
<td>75.6</td>
<td>75.77</td>
</tr>
<tr>
<td>CA</td>
<td>59.80</td>
<td>71.52</td>
<td>71.82</td>
<td>58.49</td>
<td>66.11</td>
<td>71.31</td>
<td>72.6</td>
<td>72.40</td>
</tr>
<tr>
<td>DS</td>
<td>52.48</td>
<td>66.24</td>
<td>58.52</td>
<td>64.33</td>
<td>69.45</td>
<td>64.89</td>
<td>64.5</td>
<td>68.32</td>
</tr>
<tr>
<td>GD</td>
<td>66.39</td>
<td>62.46</td>
<td>62.34</td>
<td>70.93</td>
<td>78.40</td>
<td>68.66</td>
<td>68.9</td>
<td>67.49</td>
</tr>
<tr>
<td>MP</td>
<td>59.53</td>
<td>62.46</td>
<td>77.33</td>
<td>58.95</td>
<td>67.34</td>
<td>75.84</td>
<td>74</td>
<td>75.31</td>
</tr>
<tr>
<td>RH</td>
<td>70.19</td>
<td>68.39</td>
<td>68.86</td>
<td>64.94</td>
<td>71.59</td>
<td>73.56</td>
<td>71.8</td>
<td>77.06</td>
</tr>
<tr>
<td>RC</td>
<td>70.02</td>
<td>80.49</td>
<td>73.38</td>
<td>63.05</td>
<td>69.26</td>
<td>81.21</td>
<td>81.3</td>
<td>85.79</td>
</tr>
<tr>
<td>THM</td>
<td>65.84</td>
<td>75.30</td>
<td>60.83</td>
<td>67.84</td>
<td>72.56</td>
<td>75.40</td>
<td>76.4</td>
<td>70.48</td>
</tr>
<tr>
<td>WR</td>
<td>62.25</td>
<td>71.66</td>
<td>69.62</td>
<td>67.97</td>
<td>76.15</td>
<td>68.04</td>
<td>71.8</td>
<td>71.17</td>
</tr>
<tr>
<td>MP</td>
<td>71.45</td>
<td>66.57</td>
<td>75.79</td>
<td>73.73</td>
<td>69.60</td>
<td>74.70</td>
<td>73.7</td>
<td>78.89</td>
</tr>
<tr>
<td>Mean AP</td>
<td>64.35</td>
<td>68.93</td>
<td>68.68</td>
<td>64.08</td>
<td>71.03</td>
<td>72.82</td>
<td>73.1</td>
<td>74.26</td>
</tr>
</tbody>
</table>
• Naive Augmentation (Naive): We attempt to enforce the annotation in the event model by directly training a SVM using both the automatically obtained concept representation M and the manually annotated concepts Z. (We augment the two sets and form one combined training set.) We refer to this setup in the tables as Naive.

• Structural SVM (SSVM) [33]: A structured SVM learns a discriminative classifier for a general structured output space instead of binary labels. In that, the model is trained using the joint input–output space, and thus can predict the structured output by evaluating the compatibility score of any input–output pair. In the context of our problem, where we aim to refine the concept scores such that they follow the annotation, we can consider the annotation as a structured output. In particular, we use the automatically annotated features M as an input for training, and associate each sample in M with a corresponding structured output from the manually annotated features in Z. Consequently, we use SSVM to learn a classifier which predicts the annotation for the concept scores. Since the relation between the annotation in Z and the event label is predefined and one-to-one, we can predict the event label using the predicted annotation. We use the SSVM implementation provided from [33], with margin rescaling and 1-slab algorithm as described in [33]. We refer to this approach in the tables as SSVM.

• Low-Rank optimization (LR): Here we obtain a low-rank model for the events without using the annotation constraint (corresponds to setting $\tau$ to zero in Eq. (3)). We refer to this setup in the tables as LR.

• Low-level features and SVM (Bag of Words): In this, we cluster the low-level features (STIP, ISA, DTF-HOG) to obtain a dictionary. Consequently, we compute a histogram of word frequency for each feature. We use 10,000 codebook size for all the features, and min–max normalization for the histograms. Note that the low-level features perform well, however, they lack the semantic cues contained in the concept-representation. We refer to this approach in the tables by the name of the features used in the BOW framework (i.e. STIP, ISA, or DTF-HOG).

• Izadinia and Shah [1]: Here the presence/absence of the low-level events in the complex video is considered as a latent variable. Consequently, a Latent-SVM is employed to learn a discriminative event model. This is similar to the Structural SVM formulation which we described above, except that here the annotation matrix $Z$ is considered missing/unobservable instead of being considered as a structured output.

• Yang and Shah [2]: In this, deep learning is used to compute data-driven concepts. In particular, low-level features are first learned using Topographic Independent Component Analysis (TICA) [23]. Consequently, a Deep Belief Network (DBN) is employed for dimensionality reduction. Finally, the data-driven concepts are learned by clustering the training data in a low-dimensional space using vector quantization (VQ) [34].

5.1. TRECVID MED 2011 event collection

TRECVID MED 2011 contains 15 complex events including, Boarding trick (BT), Feeding animal (FA), Landing fish (LF), Wedding (WE), Woodworking project (WP), Birthday Party (BP), Changing tire (CT), Flash mob (FM), Vehicle unstuck (VU), Grooming animal (GA), Making sandwich (MS), Parade (PA), Parkour (PR), Repairing appliance (RA) and Sewing project (SP). Each event has an event-kit, which includes a verbal description. We selected 110 mostly human action-centric concepts based on the description in the kits.

In the event kits, for each event there are 4 to 8 predefined activities, which we used as a resource to select our action concepts. For example, for event Feeding animal, activities such as holding food in front of animal, filling food container, holding bottle and eating are provided in the kit. Some of these concepts like Person Dancing and Person Jumping are common among multiple events and some belong to one specific event, e.g. Person Blowing Candle occurs only in the Birthday Party event. The concepts which have over 40 examples were retained, and the rest were discarded. Therefore, we finally ended up with 93 action concepts. We show in the experiments that the selected 93 concepts are representative of the 25 TRECVID events on which we evaluate our method. Additionally, considering the fact that [1] and [2] employed 62 concepts to recognize 15 events, we employ a considerably larger concepts’ set in order to accommodate
the additional events. We extracted various low-level features to represent the concepts, including: Dense trajectory-based MBH and HOG [3], as well as STIP [6].

We present the performance results for TRECVID MED 11 in two tables: First, Table 1, which shows the average precision for the methods which are based on concept representation. These include: the baseline (Base), the naive augmentation (Naive), Structured SVM (SSVM), Low-Rank optimization (LR), and our annotation constrained Low-Rank optimization (ACLR). As can be clearly observed, our method outperforms all other approaches by a significant margin, and improves the performance of the baseline concept representation by 7.7%. Second: Table 2 shows the average precision for both the methods which are based on concept representation and low-level features (BOW of STIP/DTF-HOG/ISA features). In the table, we also demonstrate the performance of different feature combinations including: Combining all the low-level features (STIP + HOG + ISA), and combining all the low-level features with the concept scores obtained after applying our ACLR approach.

It can be clearly observed in Table 2 that the performance of the combination of the low-level features and our concept representation attains the highest average precision on TRECVID MED 11. Note that the performance results of the STIP and HOG features are slightly higher than our concept representation, which is expected since low-level features generally outperform concept-representation as also noted in [35] and [36]. Note that our method focuses on improving the performance of concept scores. However it is important to show that combining low-level features with our concept representation can achieve state of the art on TRECVID MED datasets. Moreover concept representation is important and preferable because it carries semantically meaningful description of the video, which can be employed in other relevant tasks such as TRECVID Multimedia Event Recounting (MER), which recounts the important concepts that led to the conclusion that a particular multimedia clip contains an instance of particular event.

In Table 2, we also compare with the results from [1] and [2], which we significantly outperform. It is important to note that [1] and [2] use different features, which are not publicly available, and some of them are manually annotated and impossible to regenerate. However, it is necessary to compare with them since they constitute the state-of-the-art on TRECVID 11.

5.2. TRECVID MED 2012

TRECVID MED 2012 is similar to TRECVID MED 2011, except that it contains 10 more events including: Bike trick (BiT), Cleaning an appliance (CA), Dog show (DS), Giving direction (GD), Marriage proposal (MP), Renovating a home (RH), Rock climbing (RC), Town hall meeting (THM), Race winning (TRW) and Metal craft project (MP). The concept detectors are trained similar to TRECVID MED 2011; therefore, we also ended up with 93 action concepts, and a corresponding 93-dimensional feature vector for each video.

Similar to TRECVID MED 2011, we present the performance results in two tables: First, Table 3, which shows the average precision for the methods which are based on concept representation. Our method outperforms all other approaches by a significant margin, and improves the performance of the baseline concept representation by 6.95%. Second: Table 4 shows the average precision for the methods which are based on concept representation, low-level features, and different feature combinations. It can be clearly observed in Table 4 that the performance of the combination of the low-level features and our concept representation attains the highest average precision on TRECVID MED 12. Additionally, unlike in TRECVID MED 11, here the low-level features are inferior to the concept representation, unless all the low-level features are combined.

5.3. Refining the concept representation

Using our constrained low-rank optimization framework, we obtain more meaningful concept representations, in which the noisy concepts
are suppressed, and replaced with new concepts which are more semantically meaningful. This is illustrated in Fig. 5, which shows the average concept scores of all the training samples before and after applying our method for the events Attempting a Board Trick from TRECVID MED 11 and Attempting a Bike Trick from TRECVID MED 12. In the event Attempting a Board Trick, several essential concepts are missing from the original features (have insignificant scores) such as falling, flipping the board, and spinning the board. Additionally, several irrelevant concepts are falsely firing, such as riding bike on one wheel, running next to dog, and scaling walls trees. Similarly, in the event Attempting a Bike Trick, several essential concepts are missing from the original features such as spinning the bike, spinning the bike handle, and standing on top of bike. Moreover, several irrelevant concepts are falsely firing, such as animal grabs food, running next to dog, and standing on the board. After applying our method (bottom row in each example), the missing concepts are induced, while the irrelevant concepts are suppressed, thus generating a new concept representation which is less noisy and more semantically meaningful.

### 6. Conclusion

We presented a novel, simple, and easily implementable method for complex event recognition. We first divide the training data into two sets, one where we annotate the concepts manually, and another where we detect the concepts automatically with models trained using the first set. Consequently, we exploit the inherent low-rank structure in the examples of an event, and combine the two training sets in one set which is not only low-rank but also encouraged to follow the annotation. Thus, combining the rich descriptors in the automatically annotated set, with the accurate manual annotations. This suppresses the miss-detected concepts and generates robust and discriminative event models, which, as we verified in the experiments, outperform all previous approaches on the relevant benchmark datasets.

### Acknowledgment

This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via the Department of Interior National Business Center contract number D11PC20066. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

### References