# ObjectGraphs: Using Objects and a Graph Convolutional Network for the Bottom-up Recognition and Explanation of Events in Video

Nikolaos Gkalelis, Andreas Goulas, Damianos Galanopoulos, Vasileios Mezaris
CERTH-ITI
6th Km Charilaou-Thermi Road, P.O. BOX 60361 Thessaloniki, Greece
{gkalelis,dgalanop,bmezaris}@iti.gr, goulasand@gmail.com

## Abstract

*In this paper a novel bottom-up video event recognition approach is proposed, ObjectGraphs, which utilizes a rich frame representation and the relations between objects within each frame. Following the application of an object detector (OD) on the frames, graphs are used to model the object relations and a graph convolutional network (GCN) is utilized to perform reasoning on the graphs. The resulting object-based frame-level features are then forwarded to a long short-term memory (LSTM) network for video event recognition. Moreover, the weighted in-degrees (WiDs) derived from the graph's adjacency matrix at frame level are used for identifying the objects that were considered most (or least) salient for event recognition and contributed the most (or least) to the final event recognition decision, thus providing an explanation for the latter. The experimental results show that the proposed method achieves state-of-the-art performance on the publicly available FCVID and YLI-MED datasets*[1].

## 1. Introduction

The recognition of high-level events in unconstrained videos is one of the major research topics in multimedia understanding. Adopting popular definitions in the literature [3], a high-level event is a long-term spatially and temporally dynamic activity, e.g. a "birthday party", encompassing multiple objects or actions [23, 13], e.g., visitors, birthday cake and dancing stage, that are loosely organized spatially and temporally. This definition clearly differentiates the above research domain from the human action recognition one, which deals with the recognition of fine-grained elementary actions of a human being by capturing the subtle differences between similar actions, e.g., "running" and "walking" or "drinking beer" and "drinking wine".

[1]Source code is made publicly available at: https://github.com/bmezaris/ObjectGraphs

A key element of event recognition approaches is the method used to extract the features for representing the video. According to this, the various approaches can be categorized as follows. i) Handcrafted: Mostly older methods using low-level features, e.g. improved dense trajectories [25]. ii) C2D: Techniques that utilize deep convolutional neural networks (DCNNs) with 2D convolutional kernels to extract the static event-related information at frame-level, and subsequently utilize an appropriate technique to capture the temporal dynamics of the event [26, 22, 15, 32, 30, 31, 17, 19]. iii) C3D: DCNNs that use 3D convolutional kernels to encode simultaneously the spatiotemporal event information in videos [24, 28, 8].

The two latter categories described above have shown superior event recognition performance due to the ability of DCNNs to extract the appropriate features that separate well the different event classes. The majority of them operate directly into the overall video frame (C2D) or the entire video (C3D) in a top-down manner, i.e., utilize a single event label for each video through a cross-entropy loss function to learn to focus implicitly into the video regions that are mostly related with the specified event. However, in this way they fail to fully exploit the discriminant information carried by the multiple semantic entities related with the underlying event, as well as to provide a human understandable explanation of their event classification decisions.

The above limitations can be alleviated by either utilizing a suitable top-down approach and an appropriate video dataset for holistic representation learning [5], or by employing a bottom-up mechanism to attain a rich representation of the video content at each frame. To this end, inspired from recent advances in other video understanding domains [1, 28, 13], we follow the latter direction. Firstly, an object detector (OD) [20, 1] and a graph convolutional network (GCN) [16] are utilized to derive a feature vector representation for the objects most likely depicted in the frame as well as the relationships among them, thus, obtaining an object-level representation of the video content at each frame. Subsequently, a long short-term memory (LSTM)

[11] is used to encode the temporal dynamics of the frame representations and recognize the underlying event. Furthermore, during the testing phase, the weighted in-degrees (WiDs) of the graph vertices are used to identify the objects and their regions at frame- and video-level that mostly contributed to recognizing the event. In this way, our approach effectively provides a recounting of the recognized event: an object-grounded explanation of the model's outcome [10, 7]. The proposed approach is evaluated in two publicly available datasets, namely FCVID [14] and YLI-MED [3], producing state-of-the-art results. In summary, the main contributions of the paper are the following:

- We present a bottom-up video event recognition approach that, combining effectively relevant deep learning technologies (OD, GCN and LSTM), identifies and exploits the objects appearing in the video and their semantic relations.
- We utilize the WiDs of the derived graph's adjacency matrix in order to provide a recounting of the recognized event consisting of its key semantic entities (objects) at frame- and video-level.

The paper is structured as: Related work is discussed in Section 2. The proposed method is presented and evaluated in Sections 3 and 4. Conclusions are drawn in Section 5.

## 2. Related work

Early event recognition methods used hand-crafted features with quite good results [25]. However, over the last years, DCNN-based approaches have dominated this domain due to their groundbreaking performance in a variety of tasks. The C2D approaches extract 2D spatial convolutional features and model the temporal dimension independently. For instance, in [26] short snippets are extracted, modeling the long-range temporal structure of the video more effectively. Spatiotemporal VLAD (ST-VLAD) is presented in [22], encoding convolutional features across different segments to represent the video. In [15], PivotCorrNN is proposed, exploiting correlations among different video modalities. S2L is introduced in [32], utilizing a pretrained ResNet and an LSTM to model separately the spatial and temporal video information. LiteEval in [30] uses a coarse and a fine LSTM operating cooperatively through a conditional gating module. In [31], AdaFrame exploits a policy gradient method to select future frames for faster and more accurate video predictions. In [17], SCSampler uses a lightweight saliency model to select the most salient temporal clips within a long video. In [19], the adaptive resolution network (AR-Net) selects on-the-fly the optimal frame resolution for classifying the video, outperforming the other methods in the FCVID dataset. In contrast to C2D approaches, C3D ones learn the space and time information jointly by exploiting 3D convolutions. For instance, in [24], convolutional 3D features are exploited by a linear support vector machine (C3D+LSVM) for video classification. In [8], the large-scale Kinetics dataset is used to derive 3D-CNNs of high depth for transfer learning applications.

The above methods learn to recognize a specified event in a top-down manner, i.e. a single event label is used for implicitly teaching the deep neural network to focus on the most salient features for the specified event in the video. A major drawback of this approach is that discriminant information contained in the multitude of semantic entities appearing in a video may not be fully exploited for recognizing the underlying event. Recently, the utilization of a bottom-up mechanism to provide a richer representation of the video content has been explored in the domains of visual question answering [1] and action recognition [28, 13]. More specifically, in [1], a bottom-up mechanism is implemented using a Faster R-CNN [20], resulting in improved image captioning. In [13], Faster R-CNN and RelDN [34] are used to extract objects and visual relationships, and construct spatiotemporal scene graphs [29] for action recognition. In [28], a 3D-ResNet backbone is combined with Faster R-CNN and GCN to represent videos as space-time region graphs for the classification of elementary actions. Inspired from the above works, a bottom-up event recognition approach is proposed here, which in contrary to [28], utilizes a Faster R-CNN with ResNet-101 backbone, a pre-trained ResNet-152 as feature extractor and a GCN to derive an object graph for each frame. That is, we represent each video as a sequence of graphs instead of a space-time region graph because, despite the fact that C3D approaches have shown promising performance in the recognition of elementary human actions, recent studies suggest that C2D methods can better encode the long-term dependencies and compositional nature of complex high-level events [12]. This is due to the very different nature of the two problems, as for instance, only subtle differences may be observed in subsequent video frames depicting a "short-term" human action (e.g. "lifting the telephone"), while, in event videos such differences may be dramatic, and thus, difficult to capture using a C3D in the entire video. Furthermore, during the testing phase, the use of a C2D backbone network (instead of C3D) allows the association of each frame with a graph and subsequently the utilization of a mechanism (i.e. the computation of the WiDs of the derived graph's adjacency matrix) to derive the most salient objects in the frame related with the recognized event.

## 3. Proposed method

### 3.1. Problem formulation

Suppose an annotated training dataset of $N$ videos and $C$ event classes. Keyframe sampling is performed to obtain

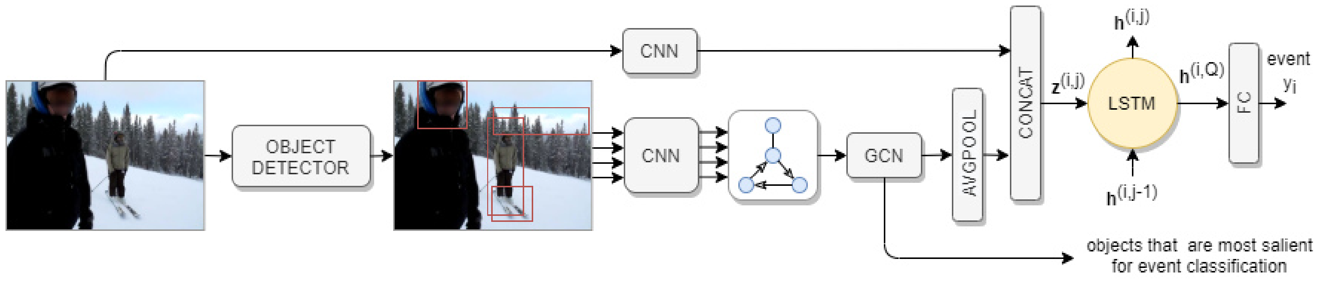

Figure 1. Block diagram of the proposed *ObjectGraphs*. The object relations within each frame are encoded in a graph. A GCN is used to perform reasoning on the graph. The sequence of frame features is forwarded to an LSTM layer.

a sequence of $Q$ frames for each video, and an OD is used to detect $K$ objects at each frame, representing each object with its label, a bounding box (BB), a feature vector and a degree of confidence (DoC) value. Based on this formulation, the overall dataset can be described as

$$(\mathbf{X}^{(i,j)}, y_i), i = 1, \ldots, N; j = 1, \ldots, Q, \tag{1}$$

where, $y_i \in [1, \ldots, C]$ is the event class label, $\mathbf{X}^{(i,j)}$ represents the $j$th frame of the $i$th video,

$$\mathbf{X}^{(i,j)} = [\mathbf{x}_1^{(i,j)}, \ldots, \mathbf{x}_K^{(i,j)}]^T, \tag{2}$$

where $\mathbf{x}_k^{(i,j)} \in \mathbb{R}^F$ is the feature vector representation of the $k$th object detected by the OD at frame $(i, j)$, the feature vectors in the rows of $\mathbf{X}^{(i,j)}$ are sorted in descending order based on their DoC value, $F$ is the dimensionality of the feature space $\mathbb{R}^F$, $u_k^{(i,j)} \in [1, \ldots, P]$ is the object class label of $\mathbf{x}_k^{(i,j)}$, and $P$ is the number of object classes. Given the above formulation, a network architecture combining a GCN and an LSTM structure is used to learn the spatiotemporal dynamics of high-level events. The overall architecture is shown in Fig. 1 and explained in detail in the next subsections.

## 3.2. Object detector

In order to obtain a precise representation of the underlying event at each frame, we need to focus on the frame regions conveying the high-level semantic information that can help us recognize the event and discard the noisy or irrelevant frame parts. To this end, a bottom-up procedure is adopted in order to obtain $K$ objects at each frame [1]. Specifically, a ResNet-101 [9] pretrained and fine-tuned in ImageNet1K [21] and Visual genome [18] datasets, respectively, is used as a backbone network of a Faster R-CNN architecture [20]. Applying the latter to an input frame $(i, j)$ means that a convolutional feature map output is derived, region proposals are produced using the region proposal network (RPN) module sliding over the obtained feature map, and several BBs with corresponding DoC values at multiple scales and aspect ratios are generated using the box-regression and -classification networks. All the BBs are sorted according to their DoC value, and a non-maximum suppression procedure with an intersection-over-union (IoU) threshold is applied to retrieve the top-$K$ proposed regions along with the corresponding object class labels $u_k^{(i,j)}, k = 1, \ldots, K$. Subsequently, the region of interest (RoI) pooling layer is used to extract an $H \times H$ feature map and obtain the respective coordinates of the $K$ regions in the input frame.

Each of the $K$ regions derived above is fed to a feature extractor retrieving a feature vector $\mathbf{x}_k^{(i,j)}$ capturing the appearance of the $k$th object in the frame. As feature extractor we utilize the pool5 layer of a ResNet-152 trained on the ImageNet11K dataset [21].

## 3.3. Graph construction

The appearances of the objects in frame $(i, j)$ and the interrelations among them are encoded by constructing a directed graph $G^{(i,j)}(\mathcal{V}^{(i,j)}, \mathcal{E}^{(i,j)})$ where, $\mathcal{V}^{(i,j)}$ is the set of vertices and $\mathcal{E}^{(i,j)}$ of the edges. In order to avoid a notation clutter, the superscript $(i, j)$ is dropped in the rest of this and next subsection. In our setting, the vertices in $\mathcal{V}$ are represented by the $K$ vectors associated with the objects in the frame, sorted in descending order according to their DoC values, $\mathbf{x}_1, \ldots, \mathbf{x}_K$, as explained in Eq. (2). A matrix $\mathbf{S} \in \mathbf{R}^{K \times K}$ is then constructed using the following pairwise similarity measure [27, 28]

$$[\mathbf{S}]_{l,k} = \tilde{\mathbf{v}}_l^T \check{\mathbf{v}}_k, \tag{3}$$

where $[\mathbf{S}]_{l,k}$ is the element of $\mathbf{S}$ in the $l$th row and $k$th column. In the expression above, similarly to [27, 28], $\tilde{\mathbf{v}}_l$ and $\check{\mathbf{v}}_k$ are derived using two different affine transformations on the object feature vectors,

$$\check{\mathbf{v}}_l = \check{\mathbf{W}}\mathbf{x}_l + \check{\mathbf{b}}, \quad \tilde{\mathbf{v}}_k = \tilde{\mathbf{W}}\mathbf{x}_k + \tilde{\mathbf{b}}, \tag{4}$$

where, $\tilde{\mathbf{W}}, \check{\mathbf{W}} \in \mathbb{R}^{F \times F}$ and $\tilde{\mathbf{b}}, \check{\mathbf{b}} \in \mathbb{R}^F$ are optimized during the network training procedure. The adjacency matrix $\mathbf{A} \in \mathbb{R}^{K \times K}$ of the graph is then computed as [33]

$$[\mathbf{A}]_{l,k} = \frac{[\mathbf{S}]_{l,k}^2}{\sum_{k=1}^K [\mathbf{S}]_{l,k}^2}, \tag{5}$$

i.e. the weighted out-degree of each vertex is normalized to one.

## 3.4. Graph convolutional network

An $M$-layer GCN is used to exploit objects' information encoded in the frame-level graphs in order to learn discriminant graph embeddings for event recognition. Given the adjacency matrix $\mathbf{A}$ (Eq. (5)), the $m$th graph convolutional layer is implemented as [28, 16]

$$\mathbf{X}^{[m]} = \mathsf{ReLU}(\mathsf{LN}(\mathbf{A}\mathbf{X}^{[m-1]}\mathbf{W}^{[m]})), \tag{6}$$

where, $\mathsf{LN}()$, $\mathsf{ReLU}()$ are the layer normalization [2] and rectified linear unit operators, $\mathbf{W}^{[m]} \in \mathbb{R}^{F^{[m-1]} \times F^{[m]}}$ is the weight matrix at layer $m$, the rows of $\mathbf{X}^{[m]} \in \mathbb{R}^{K \times F^{[m]}}$ are the hidden feature vectors corresponding to the $K$ objects in the frame, $\mathbf{X}^{[0]}$ equals $\mathbf{X}$ defined in Eq. (2), i.e. consists of the object feature vectors extracted using the OD, and $F^{[m]}$ is the dimensionality of the feature vectors at the $m$th layer.

## 3.5. Event recognition

A single feature vector for the $(i, j)$ frame is obtained as explained in the following. The output of the GCN is passed through an average pooling layer, yielding a local feature vector $\bar{\mathbf{z}}^{(i,j)} \in \mathbb{R}^{F^{[M]}}$. A global feature vector $\hat{\mathbf{z}}^{(i,j)} \in \mathbb{R}^{F}$ is also obtained by applying the feature extractor of the OD to the entire frame $(i, j)$ (i.e. the frame is fed to the ResNet-152 pretrained on the ImageNet11K dataset and the output of the pool5 layer is used to represent $\hat{\mathbf{z}}^{(i,j)}$, similarly to what is described for specific regions in Section 3.2). The two feature vectors are then concatenated to form $\mathbf{z}^{(i,j)} \in \mathbb{R}^{F+F^{[M]}}$, encoding both the local and global frame information. Next, a standard LSTM layer [11] is utilized to capture the temporal dynamics of the event along the different frames

$$\mathbf{h}^{(i,j)} = \mathsf{LSTM}(\mathbf{z}^{(i,j)}, \mathbf{h}^{(i,j-1)}), \tag{7}$$

where $\mathbf{h}^{(i,j)}$ is the hidden state vector[2]. The hidden state vector $\mathbf{h}^{(i,Q)}$ at the last time step of the video sequence is forwarded to a fully connected (FC) classification head (in our experiments we use two FC layers with an appropriate nonlinearity, i.e. softmax or sigmoid), providing a score value for each event in the dataset.

## 3.6. Explanation of event recognition results

During the forward signal propagation in the proposed network architecture the adjacency matrix amplifies the contribution of the objects relevant to the event depicted in the scene, and in contrary attenuates the contribution of the irrelevant ones. To this end, during the testing phase, the adjacency matrix $\mathbf{A}^{(i,j)}$ (Eq. (5)) associated with the frame $(i, j)$ is employed to derive the set of objects whose contribution to the feature vector $\bar{\mathbf{z}}^{(i,j)}$ was amplified and thus mostly contributed to the recognition of the specified event. Firstly, the WiD $\gamma_k^{(i,j)}$ of the $k$th graph vertex is computed as follows

$$\gamma_k^{(i,j)} = \sum_{l=1}^{K} [\mathbf{A}^{(i,j)}]_{l,k}, \quad k = 1, \ldots, K. \tag{8}$$

The computed $\gamma_k^{(i,j)}$ corresponds to the $k$th detected object and thus can be associated with its object class label $u_k^{(i,j)}$ (see Eq. (2)) and the respective BB. We treat $\gamma_k^{(i,j)}$ as an indicator for the contribution of the $k$th object in associating the frame $(i, j)$ with the recognized event. Therefore, these quantities can be used (e.g. by means of mean- or max-pooling) to provide some form of explanation for the event recognition result. Here, for each object class $p$ we compute the "average" WiDs at frame- and video-level, $\zeta_p^{(i,j)}$, $\delta_p^{(i)}$, respectively, as shown below

$$\zeta_p^{(i,j)} = \frac{1}{N_p^{(i,j)}} \sum_{\substack{k \\ u_k^{(i,j)}==p}} \gamma_k^{(i,j)}, \tag{9}$$

$$\delta_p^{(i)} = \sum_{j} \frac{N_p^{(i,j)}}{N_p^{(i)}} \zeta_p^{(i,j)}, \tag{10}$$

where $N_p^{(i,j)}$, $N_p^{(i)}$ denote the number of objects belonging to class $p$ detected in frame $(i, j)$ and entire video $i$, respectively. Then, a set of indices $\mathcal{Z}^{(i,j)}$ corresponding to the $\vartheta$ most salient objects at frame $(i, j)$ can be derived using

$$\mathcal{Z}^{(i,j)} \equiv \mathsf{sort}_{\vartheta}(\zeta_1^{(i,j)}, \ldots, \zeta_P^{(i,j)}), \tag{11}$$

where the $\mathsf{sort}_{\vartheta}$ operator returns the indices of the $\vartheta$ largest values in its input. Following a similar procedure, the $\vartheta$ largest $\delta_p^{(i)}$ (Eq. (10)) can be derived and used to obtain the most salient objects in video $i$.

## 4. Experimental evaluation

### 4.1. Datasets

We run experiments on two publicly available video datasets: i) FCVID [14] is a multilabel video dataset consisting of 91223 YouTube videos annotated according to 239 categories. It covers a wide range of topics, with the majority of them being real-world events such as "group dance", "horse riding", "birthday", "making cake" and other. The dataset is evenly split into training and testing partitions with 45611 and 45612 videos, respectively. Among them, 436 videos in the training partition and 424

[2]Note that the expressions associated with the computation of the rest of the LSTM vectors (i.e. input gate, forget gate, etc.) are not shown here for notational convenience.

videos in the testing partition were corrupted and thus could not be used. ii) YLI-MED [3] is a TRECVID-style video corpus based on YFCC100M, containing 1823 videos and 10 event categories. The dataset is divided into standard training and testing partitions of 1000 and 823 videos, respectively.

## 4.2. Setup

Uniform sampling is first applied to represent each video with a sequence of $Q = 9$ frames. Noting that in both datasets, videos' duration ranges from few seconds to several minutes, this yields sparsely sampled video sequences. In the FCVID dataset, our model is trained as follows: The OD described in Section 3.2 is used to derive $K = 50$ objects for each video frame, where each object is associated with a BB, an object class label and a feature vector of dimensionality $F = 2048$. The size of the feature maps extracted from the RoI pooling layer of the Faster R-CNN is set to $14 \times 14$ (i.e. $H = 14$). Moreover, the feature extractor described in Section 3.2 (i.e. the pool5 layer of a pretrained ResNet-152 on ImageNet11K) is applied on the entire frame to derive a 2048-dimensional feature vector, encoding the global appearance information. The extracted feature vectors are then utilized for learning the GCN, LSTM and FC layer parameters of our model, as explained in Section 3. We use a two-layer GCN with 2048 hidden size for each layer (i.e. $M = 2, F^{[m]} = 2048, m = 1, 2$), an LSTM layer of hidden size 4096, two FC layers with 2048 and 239 units, respectively, and a sigmoid nonlinearity is utilized on the last FC layer to facilitate multilabel learning.

A two-stage procedure is applied for training our model. Initially, the overall network is trained for 60 epochs using Adam optimizer, batch size 64, cross-entropy (CE) loss, learning rate $10^{-4}$ reduced by a factor of 10 at epoch 50, and a dropout rate of 0.5 is applied between the two FC layers. In the second stage, the GCN is frozen and utilized as a feature extractor for further optimizing the parameters of the LSTM and FC layers, using a learning rate of $10^{-5}$ and 10 epochs in total.

For the experiments on YLI-MED we use the same network architecture as above, except that the last FC layer consists of 10 units with a softmax activation function, which is preferred in multiclass classification problems. The GCN pretrained on FCVID is exploited as a feature extractor for training the LSTM and FC layers (generally, different methods pretrain on different datasets, e.g. C3D+LSVM [24] from Table 2 is pretrained on Sports-1M and 3D-CNN [8] on Kinetics). The training is performed using Adam optimizer, batch size 16, exponential schedule with initial learning rate $10^{-4}$, decay factor 0.9 at every epoch, and 30 epochs in total.

Table 1. Performance comparison on FCVID.

| | mAP(%) |
|---|---|
| *ST-VLAD [22]* | 77.5 |
| *PivotCorrNN [15]* | 77.6 |
| *LiteEval [30]* | 80.0 |
| *AdaFrame [31]* | 80.2 |
| *SCSampler [17]* | 81.0 |
| *AR-Net (ResNet backbone) [19]* | 81.3 |
| *AR-Net (EfficientNet backbone) [19]* | 84.4 |
| *ObjectGraphs (proposed; ResNet backbone)* | **84.6** |

Table 2. Performance comparison on YLI-MED.

| | Top-1 accuracy(%) |
|---|---|
| *C3D+LSVM [24]* | 65.61 |
| *3D-CNN [8]* | 72.66 |
| *TSN [26]* | 74.12 |
| *ActionVLAD [6]* | 76.67 |
| *S2L [32]* | 79.46 |
| *ObjectGraphs (proposed)* | **83.60** |

## 4.3. Results

The proposed approach is evaluated on FCVID using the mean average precision (mAP) and compared against the top-scoring approaches of the literature, i.e. PivotCorrNN [15], LiteEval [30], AdaFrame [31], SCSampler [17], ST-VLAD [22] and AR-Net [19]. On YLI-MED, the top-1 accuracy is utilized, and the comparison is performed against the top-scoring literature approaches for this dataset, i.e. C3D+LSVM [24], 3D-CNN [8], TSN [26], ActionVLAD [22] and S2L [32]. The results on FCVID and YLI-MED are shown in Table 1 and 2, respectively. From the obtained results we observe the following: i) The proposed approach achieves the best performance in both datasets. On YLI-MED, we significantly improve the-state-of-the art by a large margin. On the much larger FCVID dataset, a small but significant performance gain of 0.2% is obtained over the previous best method in this dataset, despite the latter using a very strong backbone network (EfficientNet); its variant that uses a ResNet backbone has a mAP that is lower by 3 percentage points. Comparing our method, which has only been tested with a ResNet backbone, with the equivalent AR-Net variant, a significant performance gain of 3.3% is observed. ii) From the results on YLI-MED (Table 2) we observe that the C3D approaches underperform in this task. This may be due to overfitting as this dataset is relatively small [8], or because the C2D approaches operating at the first level on individual frames and combined with the LSTM can capture more effectively the loose spatiotemporal structure and dynamics of the high-level events [12].

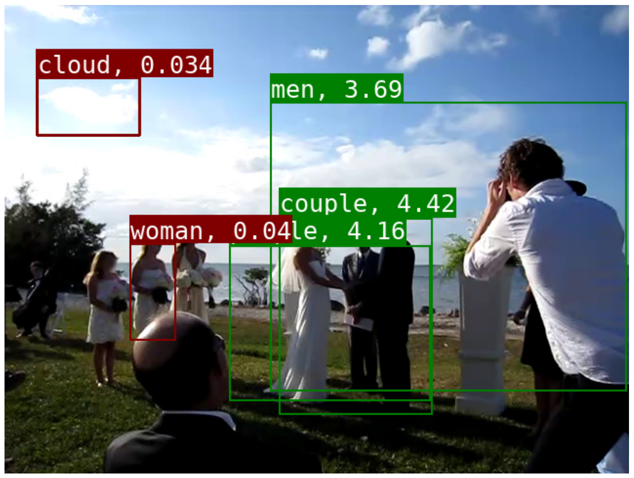

Figure 2. Frame sampled from a video labeled "Wedding ceremony". The three most salient objects in terms of WiD are depicted in green BBs. The name of the object and the computed WiD value are shown at the left top corner of each BB. These objects are strongly related with each other and with the recognized event. The three objects that are the least correlated with the event are shown in a red BBs. Our model tends to ignore such objects.

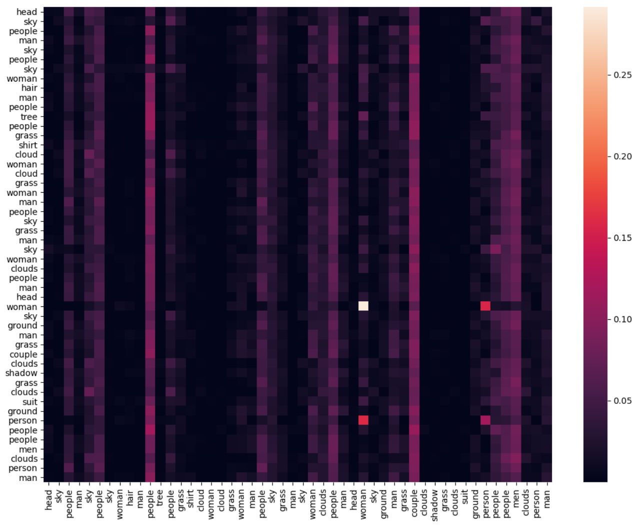

Figure 3. Visualization of the adjacency matrix (Eq. (5)) for the frame of Fig. 2. We observe that certain object classes tend to produce highly influential graph nodes, indicating that they are very strong predictors of the recognized event.

## 4.4. Explanations of the event recognition results

In addition to the event label, our model can provide visual explanations concerning the event recognition outcome. This is performed by exploiting the WiDs of the graphs' adjacency matrix, as described in Section 3.6.

To illustrate how our model can be used to provide visual explanations at frame-level, Fig. 2 presents one frame of a video labeled "Wedding ceremony", while the graphs' adjacency matrix (Eq. (5)) corresponding to this frame is depicted in Fig. 3. Moreover, in Fig. 4 the top bar plot presents the average DoC values derived using the OD, and similarly, the bottom bar plot shows the "average" WiDs (Eq. (9)) of the detected objects.

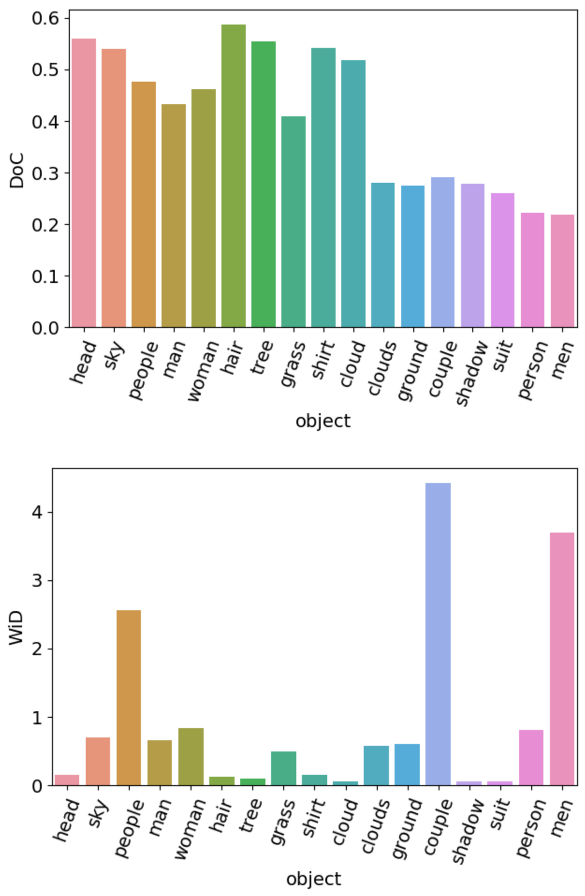

Figure 4. Bar plots of the average DoC values and the "average" WiDs (Eq. (9)), corresponding to the objects detected in the frame of Fig. 2. We observe that objects detected with a high DoC value are mostly unrelated with the recognized event. On the other hand, the objects associated with a high WiD (couple, men, people, woman, etc.) strongly correlate with the event ("Wedding ceremony").

Using the WiDs, the detected objects can be ranked and used to produce visual explanations of the model's result. For instance, the three most and three less salient objects with relation to the recognized event along with their "average" WiD values are shown in Fig. 2 with green and red BBs, respectively.

From the example above, we see that our model tends to focus on the objects that are the most visually relevant to the recognized event and to ignore the irrelevant ones. Additionally, in contrary to the DoC values, which can provide a general overview of the scene, we observe that the WiDs contain valuable information about the event and can

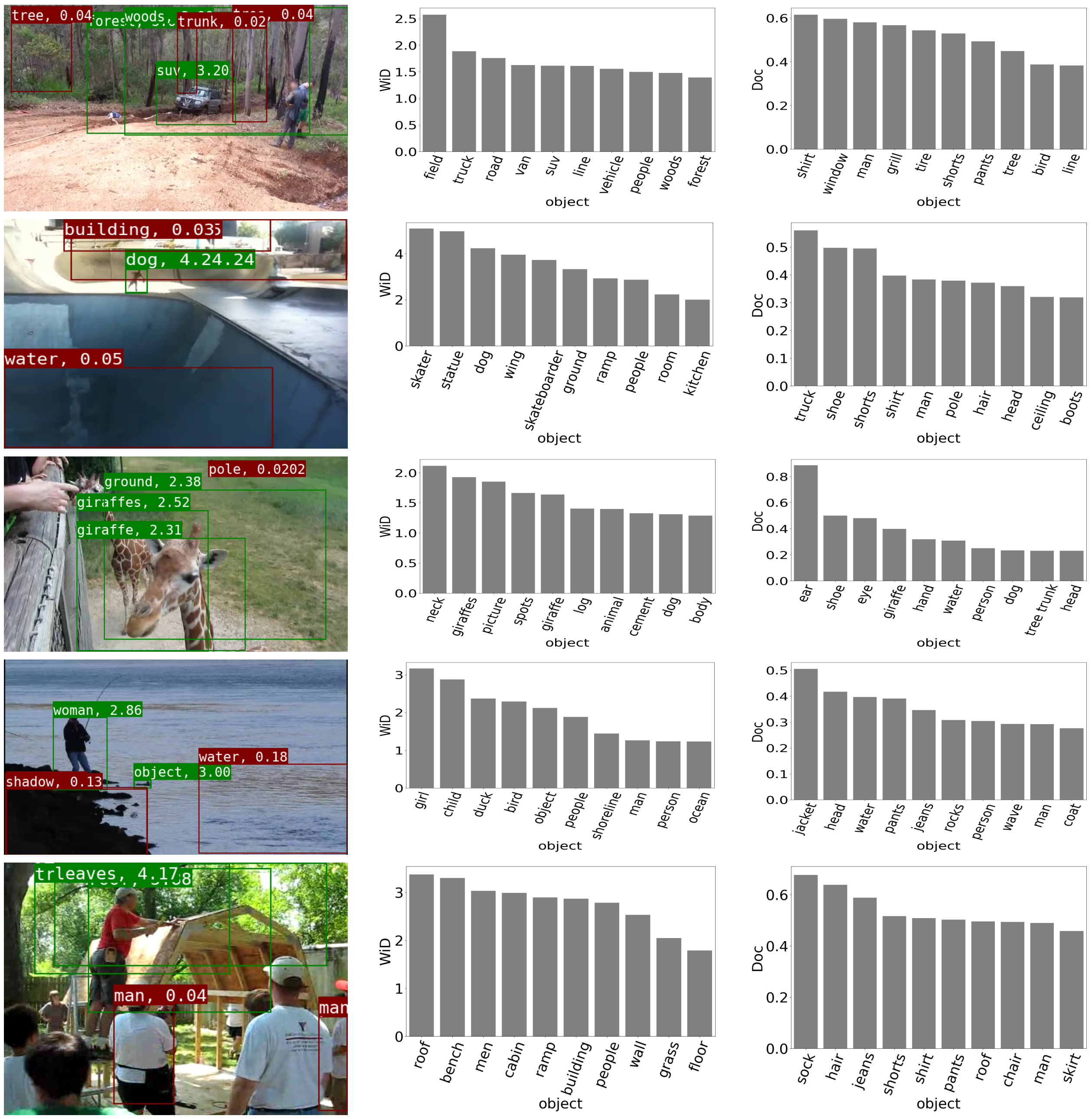

Figure 5. One visual explanation example for five event categories (from top to bottom): a) "Getting a vehicle unstuck", b) "Person attempting a board trick", c) "Person hand-feeding an animal", d) "Person landing a fish", e) "Working on a woodworking project".

be used to identify the visually salient objects accurately.

In Fig. 5 we demonstrate the use of our model to provide explanations at video-level. Each row of this figure corresponds to a video from a different event category, consisting of a video frame, one bar plot depicting the $\vartheta = 10$ objects with the highest "average" WiDs at video level (Eqs. (10), (11)) and a similar bar plot depicting the objects with the highest average DoC values along the video. We again observe that in all examples the proposed method focuses on a usually small part of the frame where the recognized event is occurring. For instance, the objects depicting a skater (although mislabeled as "dog" by the OD) and a woman fishing (labeled as "woman" by the OD) are identified as the most salient in the frame sampled from the video of the event "Person attempting a board trick" (second row of Fig. 5) and "Person landing a fish" (fourth row of Fig. 5), respectively. We also see that the top ten object class labels derived with our method in all cases can provide a sensible

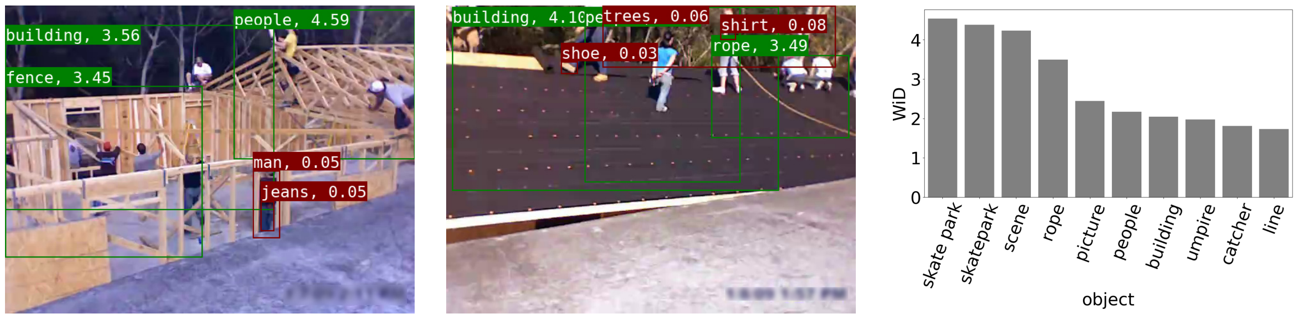

Figure 6. Visual explanation example for a video depicting "Working on a woodworking project" but mis-recognized as "Person attempting a board trick". From the bar plot we see that the most salient objects based on the "average" WiDs at video level (Eq. (10)) are "skate park" and "skatepark"'. These objects refer to the roof of the wood construction, which, as shown in the second frame, highly resemble a skate park, explaining why our model mislabeled this video.

recounting of the recognized event, while this is not true for the DoC-based recountings.

Finally, in Fig. 6 we provide an example where our model provided a wrong event recognition decision. From the bar plot in this figure we see that the top most salient objects are "skate park" and "skatepark", both associated with very high "average" WiDs. We also see that the roof of the wood construction depicted in the second video frame of Fig. 6 is very similar to a skate park, which explains why our model mislabeled this video.

## 4.5. Ablation study

We perform two ablation studies in order to gain further insight into the proposed approach. Firstly, we examine the influence of the main components of the proposed network architecture. More specifically, we evaluate the following three architectures: i) Global: the bottom-up mechanism, graph construction mechanism and GCN are removed from the network, i.e. only "global" features $\hat{\mathbf{z}}^{(i,j)}$ (Eq. (7)) are utilized for learning the event. ii) Global + local + FC: the graph construction mechanism and GCN are replaced by an FC layer, i.e the adjacency matrix $\mathbf{A}$ (Eq. (5)) is removed from Eq. (6). iii) Global + local + GCN: the entire network architecture is utilized. For simplicity, all the above networks are evaluated on FCVID using just the first-stage training procedure described in Section 4.2 (without freezing the GCN – thus, there is a small performance loss compared to the results reported in Table 1), i.e., end-to-end training for 60 epochs with Adam optimizer and CE loss, batch size 64, initial learning rate $10^{-4}$ reduced to $10^{-5}$ at epoch 50, and FC dropout rate of 0.5. The results are provided in Table 3. We see that the information provided by both the bottom-up mechanism and the graph related parts of the network (the graph construction mechanism and the GCN) have a strong impact on the performance of the network, providing an absolute gain of 2.3% and 1.3%, respectively. We also see that the graph construction mechanism and GCN cannot be sufficiently replaced by an FC layer, as it is shown that the FC layer cannot model with the same effectiveness the relations among objects.

Table 3. Influence of different components.

| | mAP(%) |
|---|---|
| Global | 80.6 |
| Global + local + FC | 82.9 |
| Global + local + GCN | **84.2** |

Table 4. Influence of different GCN depths.

| # Layers | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| mAP(%) | 84.0 | **84.2** | 84.0 | 83.9 |

In a second study, we examine the impact of the number of GCN layers in the performance of the model. The different architectures are evaluated on FCVID using the same training procedure as above. The results are shown in Table 4. We observe that the performance of the network slightly reduces when more than two layers are used, due to the known problem of over-smoothing [4].

## 5. Conclusions

We presented a new approach for video event recognition which exploits the relations among objects within each frame. More specifically, a graph, constructed using the appearance features of the objects, is exploited by our model to recognize the video event. Moreover, using the weighted in-degrees of the graph's adjacency matrix, our model is able to provide insightful explanations for its decisions. It is experimentally verified that this approach achieves state-of-the-art results on two popular video datasets.

**Acknowledgment.** This work was supported by the EU Horizon 2020 programme under grant agreements 832921 (MIRROR) and 951911 (AI4Media).

## References

[1] P. Anderson, X. He, et al. Bottom-up and top-down attention for image captioning and visual question answering. In *Proc. ICVGIP*, pages 6077–6086, Hyderabad, India, Dec. 2018.

[2] J. L. Ba, J. R. Kiros, and G. E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016.

[3] J. Bernd, D. Borth, et al. The YLI-MED corpus: Characteristics, procedures, and plans. *CoRR*, abs/1503.04250, 2015.

[4] M. Chen, Z. Wei, et al. Simple and deep graph convolutional networks. In *Proc. MLR*, volume 119, pages 1725–1735, July 2020.

[5] A. Diba, M. Fayyaz, et al. Large scale holistic video understanding. In *Proc. ECCV*, pages 593–610, Glasgow, UK, Aug. 2020.

[6] R. Girdhar, D. Ramanan, et al. ActionVLAD: Learning spatio-temporal aggregation for action classification. In *Proc. IEEE CVPR*, pages 3165–3174, Honolulu, HI, USA, July 2017.

[7] N. Gkalelis, V. Mezaris, I. Kompatsiaris, and T. Stathaki. Video event recounting using mixture subclass discriminant analysis. In *Proc. IEEE ICIP*, pages 4372–4376, Melbourne, Australia, Sept. 2013.

[8] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In *Proc. IEEE CVPR*, pages 6546–6555, Salt Lake City, Utah, USA, June 2018.

[9] K. He, X. Zhang, et al. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, pages 770–778, Las Vegas, NV, USA, June 2016.

[10] L. A. Hendricks, Z. Akata, et al. Generating visual explanations. In *Proc. ECCV*, pages 3–19, Amsterdam, The Netherlands, Oct. 2016.

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8), 1997.

[12] M. Hutchinson, S. Samsi, et al. Accuracy and performance comparison of video action recognition approaches. *CoRR*, abs/2008.09037, 2020.

[13] J. Ji, R. Krishna, L. Fei-Fei, and J. C. Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proc. IEEE CVPR*, pages 10236–10247, June 2020.

[14] Y.-G. Jiang, Z. Wu, et al. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(2):352–364, 2018.

[15] S. Kang, J. Kim, et al. Pivot correlational neural network for multimodal video categorization. In *Proc. ECCV*, pages 413–431, Munich, Germany, 2018.

[16] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. In *Proc. ICLR*, Toulon, France, Apr. 2017.

[17] B. Korbar, D. Tran, and L. Torresani. SCSampler: Sampling salient clips from video for efficient action recognition. In *Proc. IEEE ICCV*, pages 6231–6241, Seoul, Korea (South), Oct./Nov. 2019.

[18] R. Krishna, Y. Zhu, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73, May 2017.

[19] Y. Meng, C.-C. Lin, et al. AR-Net: Adaptive frame resolution for efficient action recognition. In *Proc. ECCV*, pages 86–104, Glasgow, UK, Aug. 2020.

[20] S. Ren, K. He, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Proc. NIPS*, volume 28, 2015.

[21] O. Russakovsky, J. Deng, et al. ImageNet large scale visual recognition challenge. *Int. J. Comput. Vision*, 115(3):211–252, 2015.

[22] M. Soltanian, S. Amini, and S. Ghaemmaghami. Spatio-temporal VLAD encoding of visual events using temporal ordering of the mid-level deep semantics. *IEEE Trans. Multimedia*, 22:1769–1784, 2020.

[23] E. Spyrou and Y. Avrithis. Detection of high-level concepts in multimedia. In Borko Furht, editor, *Encyclopedia of Multimedia*, pages 151–158, Boston, MA, 2008. Springer US.

[24] D. Tran, L. Bourdev, et al. Learning spatiotemporal features with 3D convolutional networks. In *Proc. IEEE ICCV*, pages 4489–4497, Santiago, Chile, 2015.

[25] H. Wang and C. Schmid. Action recognition with improved trajectories. In *Proc. IEEE ICCV*, pages 3551–3558, Sydney, NSW, Australia, Dec. 2013.

[26] L. Wang, Y. Xiong, et al. Temporal segment networks: Towards good practices for deep action recognition. In *Proc. ECCV*, pages 20–36, Amsterdam, The Netherlands, Oct. 2016.

[27] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proc. IEEE CVPR*, pages 7794–7803, Salt Lake City, Utah, USA, June 2018.

[28] X. Wang and A. Gupta. Videos as space-time region graphs. In *Proc. ECCV*, volume 11209, pages 413–431, Munich, Germany, Sept. 2018.

[29] C.-Y. Wu, C. Feichtenhofer, et al. Long-term feature banks for detailed video understanding. In *Proc. IEEE CVPR*, pages 284–293, Long Beach, CA, USA, June 2019.

[30] Z. Wu, C. Xiong, et al. LiteEval: A coarse-to-fine framework for resource efficient video recognition. In *Proc. NIPS*, pages 7778–7787, Vancouver, Canada, 2019.

[31] Z. Wu, C. Xiong, et al. AdaFrame: Adaptive frame selection for fast video recognition. In *Proc. IEEE CVPR*, pages 1278–1287, Long Beach, CA, USA, June 2020.

[32] Z. Xu, L. Su, et al. S2L: Single-streamline for complex video event detection. In *Proc. IEEE ICMEW*, pages 1–6, San Diego, CA, USA, 2018.

[33] J. Yang, W. S. Zheng, et al. Spatial-temporal graph convolutional network for video-based person re-identification. In *Proc. IEEE CVPR*, pages 3286–3296, Seattle, WA, USA, June 2020.

[34] J. Zhang, K. J. Shih, et al. Graphical contrastive losses for scene graph parsing. In *Proc. IEEE CVPR*, pages 11535–11543, Long Beach, CA, USA, June 2019.