Abstract

Long-term complex activity recognition and localisation can be crucial for decision making in autonomous systems such as smart cars and surgical robots. Here we address the problem via a novel deformable, spatiotemporal scene graph approach, consisting of three main building blocks: (i) action tube detection, (ii) the modelling of the deformable geometry of parts, and (iii) a graph convolutional network. Firstly, action tubes are detected in a series of snippets. Next, a new 3D deformable RoI pooling layer is designed for learning the flexible, deformable geometry of the constituent action tubes. Finally, a scene graph is constructed by considering all parts as nodes and connecting them based on different semantics such as order of appearance, sharing the same action label and feature similarity. We also contribute fresh temporal complex activity annotation for the recently released ROAD autonomous driving and SARAS-ESAD surgical action datasets and show the adaptability of our framework to different domains. Our method is shown to significantly outperform graph-based competitors on both augmented datasets.

1 Introduction

Complex activity recognition is attracting much attention in the computer vision research community due to its significance for a variety of real-world applications, such as autonomous driving [6, 7], surveillance [28], medical robotics [60] or team sports analysis [21]. In autonomous driving, for instance, it is extremely important that the vehicle understands dynamic road scenes, in order, e.g., to accurately predict the intention of pedestrians and forecast their trajectories to inform appropriate decisions. In surveillance, group activities rather than actions performed by individuals need to be monitored. Robotic assistant surgeons need to understand what the main surgeon is doing throughout a complex surgical procedure composed by many short-term actions and events [43], in order to suitably assist them.

Recent methods for action or activity recognition and localisation can be broadly divided into two categories; single atomic action [19, 30, 36, 54] and multiple atomic action recognition/localisation [22, 25, 31, 45, 51, 57]. The former methods only focus on identifying the start and end of an action performed in a short video portraying a single instance, leveraging datasets such as UCF-101 [44] or Charades [38]. The latter set of approaches consider videos which contain a number of atomic actions or multiple repetitions of the same action. Methods in this category do address complex activity recognition, as their aim is to understand an overall, dynamic scene by detecting and identifying its constituent components. Datasets used for complex activity detection are Epic-kitchens [11], THUMOS14 [23] or ActivityNet
Figure 1: Overall pipeline of our long-term complex activity detection framework. (i) The input video is first divided into snippets. (ii) Snippets are passed to an action tube detector module one by one. (iii) A part-based deformation module receives 3D VGG features and action tube locations and returns features for the salient (non-background) parts of each tube instance. (iv) A GCN module represents activity parts (action tubes) as nodes with the features generated by ROI pooling and builds edges with different semantics to construct a spatiotemporal graph representation. Finally (v), the graph representation features produced by GCN inference on the consolidated graph are used for temporal activity detection.

Both classes of methods are geared towards merely recognising and localising short term action or activities that lasts for only a few frames or seconds.

Unlike all existing methods, in this work we present a framework capable of recognising complex, long-term activities, validated in the fields of autonomous driving and surgical robotics but of general applicability and extendable to other domains. More precisely, by ‘complex activity’ we mean an ensemble of ‘atomic’ actions, each performed by an individual agent present in the environment, which extends over a period of time and which collectively has a meaning. For instance, in the autonomous driving context, one can define the complex activity Negotiating intersection as composed of the following atomic actions: an autonomous vehicle (AV) moves along its lane; the traffic light regulating the vehicle’s lane turns red, while the light turns green for a traversing road; a number of other vehicles pass through the intersection from both sides; the AV’s light turns green again; the AV resumes moving and crosses the intersection.

The proposed pipeline (Fig. 1) is divided into three parts: (i) action tube detection, (ii) part-based feature extraction and learning via 3D deformable RoI pooling, and (iii) a graph generation strategy to process a variable number of parts and their connections, aimed at learning the overall semantics of a dynamic scene representing a complex activity. Action tube detection is a necessary pre-processing step, aimed at spatially and temporally locating the atomic actions present [1, 2, 3, 4, 5, 6]. Note that the tube detector needs to ensure a fixed-size representation for each activity part (atomic action). Here, in particular, we adopt AMTNet [5], as the latter describes action tubes of any duration using a fixed number of bounding box detections.
Our contribution is twofold. Firstly, our novel 3D deformable RoI pooling layer, inspired by standard deformable and modulated RoI pooling [10, 59], is not only designed to work with 3D data but is also capable of learning feature representations for tubes of variable spatiotemporal shape. Secondly, an original Graph Convolutional Network (GCN) module constructs a graph by considering individual tubes as nodes and connecting them via edges encoding diverse semantics, namely: appearance order, sharing of the same action label, and spatial feature similarity. The spatiotemporal scene graph so constructed is then processed by a stack of graph convolutional layers resulting in graph representation features, which are used to train a classifier for recognising complex activities, followed by a localisation stage which uses a sliding window approach.

The framework is evaluated using two real-world datasets springing from completely different domains: ROAD [42] for situation awareness in autonomous driving and SARAS-ESAD [2] for surgical action detection, both providing video-level annotation in the form of (atomic) action tubes. In this work we augment these datasets with suitable annotation on the start and end time of each instance of complex activity (road activities in ROAD vs surgical phases in SARAS-ESAD). The main contributions of this paper are therefore:

- An original deformable 3D RoI pooling approach for flexibly pooling features from the various components of the detected tubes, to create an overall representation for activity parts.
- A spatiotemporal scene graph generation and processing mechanism able to cope with a variable number of parts while learning the overall semantics of an activity class.
- Augmented annotation for two newly-released datasets aimed at making them suitable benchmarks for future work on complex activity detection.

Our results clearly indicate that the detection task (both at atomic and complex activity level) is extremely challenging on the real-world data which forms these newly annotated benchmarks, when compared to existing academic datasets. We hope this will stimulate further original thinking to address these challenges. Our method is shown to clearly outperform two recent state of the art graph-based competitors [54, 55] on both augmented datasets.

2 Related Work

Complex activity recognition. Most recent work on complex activity recognition concerns scalar sensors [4, 46, 58] or combination of both scalar and vision sensors [1, 26]. Recently, though, several vision-based complex activity recognition methods have been proposed [22, 25, 31, 45, 51, 57] with the goal of understanding an overall scene by recognising and segmenting atomic actions. These methods can be further divided into (i) sliding windows approaches [37, 47], in which an activity classifier is applied to each snippet, and (ii) boundaries analyses [16, 53], in which a model is trained to identify the start and end time of each action. Overall, current activity recognition methods are geared to recognise short-term activities via a combination of small atomic actions.

Unlike existing approaches, our objective is to understand long-term activities in dynamic scenes, such as the phases a surgical procedure is broken into, whose detection is crucial to inform the decision making of automated robotic assistants.

Deformable parts-based models. Deformable part-based models have been used by the research community for more than a decade [13, 14, 15, 20] for object detection and segmentation. Following the rapid development of Convolution Neural Networks (CNNs), Girshick
et al. [17] first recognised that deformable part-based models can be implemented for object detection in a CNN formulation, in which each convolution pyramid is fed to a distance transform pooling and a geometric filter layer. The main limitation of this method is that it is not end-to-end trainable and requires a heuristic selection of part sizes and components. A subsequent end-to-end deformable CNN formulation was proposed in [10], which uses two new CNN layers (deformable convolution and deformable RoI pooling) that reproduce the functionalities of traditional part deformation. The latest version of deformable CNN is Deformable ConvNets v2 [59], which introduces a modulation mechanism in both deformable convolution and RoI pooling.

To the best of our knowledge, all deformable models proposed to date focus on either object or short-term action detection, whereas here, for the first time, we design a novel 3D deformable RoI pooling layer for learning long-term complex activities.

**Graph convolutional network.** Recently, GCNs have been widely used for action and activity detection and recognition, building on their success in different areas of computer vision such as point cloud segmentation [50, 52] and 3D object detection [18]. Relevant GCN approaches have been broadly focussing on either action recognition [9, 29, 49] or temporal action localisation [54, 55]. In the former, videos are represented in different spatiotemporal formats such as 3D point clouds and time-space region graphs, and methods focus on recognising atomic actions only. In contrast, Zeng et al. [55] use GCN for temporal activity localisation by considering action proposals as nodes and a relation between two proposals as an edge. In opposition, in our model nodes are action tubes and their connections are based on an array of semantics. In another recent study, Xu et al. [54] generate graphs by considering temporal snippets as nodes and drawing connections between them based on temporal appearance and semantic similarity.

Most graph-based activity detection methods [27, 54, 55] construct a graph for a whole video by taking snippets as nodes and their temporal linkage as edges, not paying much attention to the constituent atomic actions within each snippet, and are typically limited to shorter videos and memory dependent. In contrast, our proposed framework is designed to construct a graph for each snippet which reflects the structure of a dynamic scene in terms of atomic action tubes (nodes) and the different types of relationships between them.

### 3 Proposed Method

Crucial to the identification of complex video activities is the modelling of the relations among the constituent actions. In this paper we propose to achieve this via a combination of the deformable pooling of features and a spatiotemporal graph representation employing multiple semantics.

#### 3.1 Action Tube Detection

To provide a fixed-size representation for the instances of atomic actions composing a complex activity, here we adopt AMTnet [35]. AMTnet is a two-stream online action tube detector that uses both RGB and optical flow information (although here we only use the RGB stream). The main rationale for using AMTnet is that it generates tubes in an incremental manner while preserving a fixed-size representation.

**Architectural Details.** AMTNet uses VGG-16 [39] as baseline CNN feature extractor. The last two fully-connected layers of VGG-16 are replaced by two convolutional layers. Four extra convolutional layers are added at the end. AMTNet takes as input a sequence of RGB frames with a fixed temporal interval $\Delta$ between consecutive frames, i.e., $\{f_t, f_{t+\Delta}\}$.
The input to AMTNet is in the format $[BS \times Sq \times D \times H \times W]$, where $BS$ is the training batch size, $Sq$ is the sequence length (in this case a pair), $D$ is the dimensionality (equal to 3 as we are dealing with RGB frames), while $H$ and $W$ are the height and width of each frame ($300 \times 300$ in our case). As typical in action detection, AMTNet uses both a classification and a regression layer for recognition and detection, respectively, with the goal of predicting action ‘micro-tubes’ defined by pairs of consecutive detections. The method predicts bounding boxes for a pair of frames separated by fixed gap $\triangle$, while the bounding boxes for intermediate frames are generated by interpolation. In this work, atomic action instances are represented as 3 micro-tubes with $\triangle = 3$ for an overall tube length of $L = 12$ frames, aligned with our snippet length. Complete action tubes are incrementally generated by AMTNet by temporally linking the micro-tubes predicted by the network [35].

### 3.2 3D Part-Based Deformable Layer

The feature extractor in our framework is a novel 3D deformable RoI pooling layer encoding the spatiotemporal geometry of the action tubes which correspond to the activity parts. This is an extension of the existing standard deformable RoI pooling layer [10], and has the ability to extract and learn features from an action tube rather than a 2D bounding box. The rationale behind using the 3D deformable RoI pooling layer is that it allows us to learn at training time how the geometric shape of atomic action tubes (as regions of the video considered as a spatiotemporal volume) varies across instances of the same class. Intuitively, the shape of the bounding box detections forming a tube (and therefore the shape of the tube itself) will vary with, e.g., the viewpoint, as well as the particular style with which the action is performed by a certain agent (for instance, a cyclist can turn right making a narrower as opposed to a wider turn).

The principle of our 3D deformable RoI pooling operation is shown in Figure 2. Like the classical deformable RoI pooling layer, our module also includes standard RoI pooling (used in all region proposal-based object detection methods), a fully connected layer, and offsets which encode the amount of geometric deformation. Firstly, standard RoI pooling is applied to the provided feature map $X$ and bounding box locations forming an action tube $(L \times [x, y, w, h])$, by subdividing the tube into a pooled feature map grid of fixed-size in both the spatial and the temporal dimensions: $L \times k \times k$. Here $L = 12$ is the fixed action tube length, while $k$ is a free parameter which determines the ‘bin size’, i.e., the number of grid locations detections are divided into in each spatial dimension (see Figure 2 again). Next, for each bin in the grid, normalised offsets (representing the degree of deformation of the grid components of each action tube) are generated for these feature maps using a fully-connected layer, which are then transformed using an element-wise product with the original RoI’s width and height. Offsets are also multiplied by a scalar value to modulate their magnitude (empirically set to 0.1), making them invariant to the different possible sizes of the RoI. In our framework, this layer takes the VGG features extracted by AMTNet and each detected action tube separately as an input, and returns an overall feature map which encodes both the appearance and the shape (through the above offsets) of each atomic action.

### 3.3 Graph Convolutional Network

As our purpose is to achieve a comprehensive understanding of the dynamic scene which comprises a complex activity, we propose to use a graph convolutional network to model and exploit the relations between the constituent action tubes. Unlike the tree structure of classical part-based models (which requires to fix the number of parts [8]), (spatiotemporal)
Figure 2: Our 3D deformable RoI pooling layer takes feature maps and action tube locations as input and arranges them into a fixed-size grid of components (here illustrated for size $3 \times 3$). For each grid component an offset is generated and multiplied by the original tube feature to produce the final component features.

graphs allow us to flexibly describe a complex activity composed by a variable number of actions (nodes) of different type, and to encode the different semantic relationships between them. The functioning of our GCN module is illustrated in Figure 3.

**Graph Construction.** When constructing the activity graphs, the input video is subdivided into consecutive, non-overlapping, fixed-length snippets. For each snippet, a separate graph is built with a variable number of nodes corresponding to the number of detected activity parts (action tubes) within the snippet. The initial representation of the nodes is provided by their RoI features. We consider three different types of connections: (i) the order of appearance (from left to right) of each action tube (bearing in mind that in autonomous driving, for instance, road activities tend to follow a specific order, e.g., pedestrian crossing the road followed by vehicles engaging an intersection); (ii) the spatial similarity of node features, measured using the distance proposed in [50]; (iii) node type, meant as the sharing of the same action label, as this provides relevant information for the determination of the activity class. As a result, three spatiotemporal scene graphs are constructed, having the same nodes but with different edges. While the second and third graph are undirected, the appearance order graph is a directed one. However, when merging the three graphs an undirected version of the order graph is used. These graphs are then combined by taking a union of all edges to create a single homogeneous graph representing the overall scene. Namely, two nodes are connected in the merged graph iff they are connected in at least one of the three graphs.

**Graph Convolution and Representation.** Given the final graph, global graph embedding is applied to extract the context of each snippet portraying a complex activity. In our GCN approach we apply a stack of three graph convolutional layers followed by a graph readout layer. The latter encapsulates the final graph representation by taking the mean of the hidden convolutional representations, resulting in fixed-sized feature vectors which are invariant to the number of nodes and edges.

### 3.4 Complete Framework

The complete framework is the concatenation of the aforementioned three modules. Firstly, we divide the video $V$ into $N$ snippets $S_1, S_2, ..., S_N$, with each snippet $S_i$ consisting of a fixed ($M$) number of frames: $S_i = F^{1, 2, 3, ..., M}$. Each snippet is passed to the action tube detection
module \( AT \) which returns \( K \) action tubes each composed by \( L = 12 \) bounding boxes with labels \( B_A \) and intermediate VGG features \( X_A \), represented as \( B_A^{M \times K \times 5}, \ X_A^{M \times 64 \times 300 \times 300} \in AT \). Action tube locations and features are then passed to our 3D deformable RoI pooling layer \( DRoI \) which returns a fixed-sized (i.e., \( 7 \times 7 \)) grid of components whose dimensionality is equal to the number (64) of convolutional layers: \( X_{DRoI}^{K \times 64 \times 7 \times 7} \). These features are then fed to the GCN module \( G \), where a graph with \( K \) nodes and \( E \) edges is processed to yield a fixed-sized feature representation \( X_G^{2048} \in G \). Finally, the latter features are fed to a softmax classifier to classify the snippet into their respective activity category.

For localisation, we use a sliding window approach in which snippets are sequentially classified, using a dual verification mechanism. Namely, as our purpose is to recognise and detect long-term activities, if there appears to be a random false positive or false negative between two snippets belonging to the same class we simply ignore it and consider it as having the same activity label. The detection algorithm is described in detail in the Supplementary material, Algorithm 1.

Implementation. Before training our overall architecture, we separately train AMTNet for action tube detection over both datasets. Note that we had to design from scratch suitable data loaders for the two datasets, as the format of the annotation there is completely different from that of the original datasets AMTNet was validated upon. As mentioned, our 3D RoI pooling layer includes a temporal dimension to learn the deformation of the components of a tube, rather than of a 2D object. In our experiments we also adapt a more recent version of deformable RoI pooling, termed modulated deformable RoI pooling, to the 3D case. In the GCN module, we construct a graph for each snippet in an online fashion at training time using a PyTorch data loader [32]. For the design of the GCN architecture we used the Deep Graph Library (DGL) [48] with a PyTorch back-end, which supports the processing of graphs of various length in a single mini-batch. Overall, our architecture is implemented using the PyTorch deep learning library [32] with OpenCV and Scikit-learn. For training we used a machine equipped with 4 Nvidia GTX 1080 GPUs with 12GB VRAM each.

4 Experimental Results

4.1 Datasets and Evaluation Metrics

In this paper we used two datasets for evaluating our approach, both already annotated at video level for action tubes detection.

ROAD [42]: ROAD (the ROad event Awareness Dataset for autonomous driving) is annotated for road action and event detection. Each event is described in terms of three different labels: (road) agent (e.g., cyclist, bus), action performed by the agent (e.g., turning left, right), and event location (w.r.t. the autonomous vehicle). The ROAD dataset consists of total 22 videos carefully selected from the Oxford RobotCar Dataset because of their diverse weather and lighting conditions. ROAD comprises 560K bounding boxes in 122K annotated frames with 560K agent labels, 640K action labels and 499K location labels.

For this work we augmented the annotation of the ROAD dataset for complex road activity detection. We used a total of 19 videos with an average duration of 8 minutes each, 12 of which were selected for training and the remaining 7 for testing. We temporally annotated the ROAD videos by specifying the start and end frame for six different classes of complex road activities we inferred from video inspection. For example, a ‘Negotiating intersection’ activity class can be defined which is made up of the following ‘atomic’ events: Autonomous Vehicle (AV)-move + Vehicle traffic light / Green + AV-stop + Vehicle(s) / Stopped / At junction+ AV-move. Activity class statistics are listed and described in more
SARAS-ESAD [2]: ESAD (the Endoscopic Surgeon Action Detection Dataset) is a benchmark devised for surgeon action detection in real-world endoscopic surgery videos. In ESAD, surgeon actions are classed into 21 different categories and annotated with the help of professional surgeons. Here we took a step forward and annotated ESAD in terms of complex activities corresponding to the different phases of the surgical procedure portrayed by the videos (namely, radical prostatectomy). For example, Phase # 3 corresponds to ‘Bladder neck transection’, in which a scissor cuts the neck of the bladder until it is transected. Phases and their statistics are again reported in the Supplementary material. The complex activities (surgical phases) in SARAS were defined by professional surgeons experts in radical prostatectomy. As standard in the surgical context, such phases are consecutive without the need for any background activity class. For more details please see [2].

Evaluation Metrics: For the evaluation of action tube detection performance we used the standard frame/video mean Average Precision (mAP) at different IoU thresholds $\delta$ (namely, 0.2, 0.3, 0.5, 0.75) on both datasets. Complex activity recognition was evaluated using classification accuracy, precision, recall and F-score. For complex activity localisation we used the standard protocol mAP over the temporal dimension used by all relevant methods.

4.2 Action Tube Detection

A detailed comparative analysis of AMTNet over different action detection datasets can be found in the original paper [35]. Here we briefly report the performance of AMTNet on our two datasets of choice, as AMTNet was never tested there. Table 1 reports both frame-mAP and video-mAP results, and compares AMTNet with the proposed baselines for the two datasets: the ROAD baseline (termed 3D RetinaNet [42]), and the ESAD baseline [2], a vanilla implementation of RetinaNet (only providing frame-level results). AMTNet performed better than [2] on SARAS-ESAD, while being inferior to [42] on ROAD. Remember that the main rationale for using AMTNet is that it can provide a fixed-size representation for the tubes (as required by our framework), motivating us to compromise on accuracy.

4.3 Complex Activity Recognition

Next, we provide a detailed analysis of complex activity classification using our approach on both the ROAD and SARAS-ESAD datasets. The performance for each class in both datasets is illustrated in Fig. 4 using all metrics. It is apparent that the ROAD dataset is characterised by significant fluctuations in class-wise performance, with higher recognition accuracy for activities that appear more often, e.g. ‘waiting in a queue’, as opposed to infrequent ones (e.g. ‘sudden appearance’). In SARAS-ESAD each activity class does contain enough samples for good training, while the diversity in phases still poses a challenge. In fact, performance on this dataset is a function of the complexity of the phases and the visual similarity of the constituent atomic actions, not just the amount training data. For example, Phase 5 and 6 both include ‘fat removal’ actions, complicating their differentiation.

4.4 Temporal Activity Detection - Comparison with State-of-the-Art

To evaluate the performance of our complex activity detection approach we reimplemented two state-of-the-art activity localisation methods – P-GCN [55] and G-TAD [54], as neither ROAD nor ESAD were ever used for complex activity detection. The major changes we made during re-implementation are: (i) data loading (as both P-GCN and G-TAD were designed to be trained and tested on pre-extracted features), and (ii) replacing the regression
S. KHAN AND F. CUZZOLIN: COMPLEX ACTIVITY DETECTION

Table 1: Action tube detection performance on both the ROAD and SARAS-ESAD datasets. Both Frame-mAP and Video-mAP at different IoU thresholds are reported for evaluation.

<table>
<thead>
<tr>
<th>Methods / IoU threshold δ</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Singh et al. [42] (frame-mAP)</td>
<td>-</td>
<td>-</td>
<td>25.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Singh et al. [42] (video-mAP)</td>
<td>17.5</td>
<td>4.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Bawa et al. [2] (frame-mAP)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>24.3</td>
<td>12.2</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>AMTNet (frame-mAP)</td>
<td>22.3</td>
<td>18.1</td>
<td>15.4</td>
<td>11.0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>AMTNet (video-mAP)</td>
<td>11.6</td>
<td>7.9</td>
<td>3.8</td>
<td>-</td>
<td>-</td>
<td>13.7</td>
<td>10.1</td>
<td>8.8</td>
<td>5.4</td>
<td></td>
</tr>
</tbody>
</table>

ROAD

SARAS-ESAD

Table 2: Comparative analysis of temporal activity localisation performance on ROAD and SARAS-ESAD, reporting mAP (%) at five different IoU thresholds.

<table>
<thead>
<tr>
<th>Methods / IoU threshold δ</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
<th>0.1</th>
<th>0.2</th>
<th>0.3</th>
<th>0.4</th>
<th>0.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-GCN [55]</td>
<td>60.0</td>
<td>56.7</td>
<td>53.9</td>
<td>50.5</td>
<td>46.4</td>
<td>57.9</td>
<td>55.6</td>
<td>53.4</td>
<td>49.0</td>
<td>45.1</td>
</tr>
<tr>
<td>G-TAD [54]</td>
<td>62.1</td>
<td>59.8</td>
<td>55.6</td>
<td>52.2</td>
<td>48.7</td>
<td>59.1</td>
<td>56.7</td>
<td>54.5</td>
<td>49.8</td>
<td>46.9</td>
</tr>
<tr>
<td>Ours</td>
<td>77.3</td>
<td>74.6</td>
<td>71.2</td>
<td>66.7</td>
<td>59.4</td>
<td>62.9</td>
<td>59.6</td>
<td>58.2</td>
<td>55.3</td>
<td>51.5</td>
</tr>
</tbody>
</table>

ROAD. For activity detection in ROAD we used an additional ‘background’ class, which indicates either no action or presence of action(s) without any solid indication. Whenever no action tube was detected we would use the entire frame as RoI for our parts deformation module to understand the overall scene. Temporal activity detection performance on ROAD, measured via mAP at five different IoU thresholds, is reported in Table 2 for both our approach and the two competitors. Class-wise results for each complex activity at a standard IoU threshold of 0.5 are reported in Table 3.

SARAS-ESAD. Temporal activity detection on this dataset much relates to activity recognition, as surgical phases are contiguous. Both the average mAP of the methods at five IoU thresholds and the class-wise performance for each activity (phase) at a standard IoU threshold of 0.5 are reported in Table 2 and 4, respectively. From the results it is clear how our method outperforms the chosen state-of-the-art methods by a reasonable margin.

4.5 Limitations and Future Work

The main limitation of this work is that it relies on action tube detection. From our results, the existing tube detectors are not reliable enough to perform well over challenging

Figure 4: Complex activity classification performance on both ROAD and SARAS-ESAD.
Table 3: ROAD activity localisation performance (mAP, %) for each complex road activity, at a standard IoU threshold of 0.5.

<table>
<thead>
<tr>
<th>Method / Activities</th>
<th>Negotiating intersection</th>
<th>Negotiating pedestrian crossing</th>
<th>Waiting in queue</th>
<th>Merging into vehicle lane</th>
<th>Sudden appearance</th>
<th>Walking middle of road</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-GCN</td>
<td>44.3</td>
<td>53.8</td>
<td>74.4</td>
<td>50.1</td>
<td>21.7</td>
<td>34.1</td>
</tr>
<tr>
<td>G-TAD</td>
<td>47.8</td>
<td>57.3</td>
<td>70.6</td>
<td>55.2</td>
<td>24.3</td>
<td>37.1</td>
</tr>
<tr>
<td>Ours</td>
<td>51.2</td>
<td>72.3</td>
<td>89.8</td>
<td>84.1</td>
<td>17.8</td>
<td>41.3</td>
</tr>
</tbody>
</table>

Table 4: SARAS-ESAD activity localisation performance (mAP, %) for each activity at a fixed IoU threshold of 0.5.

<table>
<thead>
<tr>
<th>Method / Activities</th>
<th>Phase#1</th>
<th>Phase#2</th>
<th>Phase#3</th>
<th>Phase#4</th>
<th>Phase#5</th>
<th>Phase#6</th>
<th>Phase#7</th>
<th>Phase#8</th>
</tr>
</thead>
<tbody>
<tr>
<td>P-GCN</td>
<td>56.7</td>
<td>43.2</td>
<td>52.3</td>
<td>59.1</td>
<td>33.8</td>
<td>59.4</td>
<td>14.8</td>
<td>41.2</td>
</tr>
<tr>
<td>G-TAD</td>
<td>51.1</td>
<td>46.6</td>
<td>57.2</td>
<td>63.8</td>
<td>29.4</td>
<td>62.2</td>
<td>19.3</td>
<td>45.7</td>
</tr>
<tr>
<td>Ours</td>
<td>57.5</td>
<td>54.1</td>
<td>69.3</td>
<td>60.2</td>
<td>31.1</td>
<td>71.3</td>
<td>16.5</td>
<td>52.4</td>
</tr>
</tbody>
</table>

real-world datasets such as those we adopted here. Clearly, if the tube detector misses an important atomic action this will affect the overall activity detection performance. Nevertheless our results show that, even when using a suboptimal detector, our approach is capable of significantly outperforming state of the art methods on our new benchmarks.

Detection is challenging on SARAS and ROAD because of their real-world nature: surgical images are indistinct, road scenes come with incredible variations. These benchmarks show how even the best detectors suffer a huge drop in performance when moving from ‘academic’ benchmarks to real ones. We hope the realism of these two extremely challenging datasets will stimulate real progress and new original thinking in the field.

In the future our primary target will be the design of a more accurate action tube detector with the ability to perform better in challenging scenarios such as those portrayed in ROAD or SARAS-ESAD. We will also explore the end-to-end training of the entire model in all its three components. Further down the line, we will update our S/T scene graph approach to properly model the heterogenous nature of the graph [56], and extend it to a more complete representation of complex dynamic events in which nodes (rather than correspond all to action tubes) may be associated with any relevant elements of a dynamic scene, such as objects, agents, actions, locations and their attributes (e.g. red, fast, drivable, etc).

5 Conclusions

In this paper we presented a spatiotemporal complex activity detection framework which leverages both part deformation and a heterogenous graph representation. Our approach is based on three building blocks; action tube detection, part-based deformable 3D RoI pooling for feature extraction and a GCN module which processes the variable number of detected action tubes to model the overall semantics of a complex activity. In an additional contribution, we temporally annotated two recently released benchmark datasets (ROAD and ESAD) in terms of long-term complex activities. Both datasets come with video-level action tube annotation, making them suitable benchmarks for future work in this area. We thoroughly evaluated our method, showing the effectiveness of our 3D part-based deformable model approach for the detection of complex activities.
Acknowledgements

The work reported in this paper was supported by Huawei Technologies Co., Ltd. and the European Union’s Horizon 2020 research and innovation programme, under Grant Agreement no. 779813 (SARAS).

References


