Spatiotemporal Deformable Scene Graphs for Complex Activity Detection

BMVC 2021 Submission # 706

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

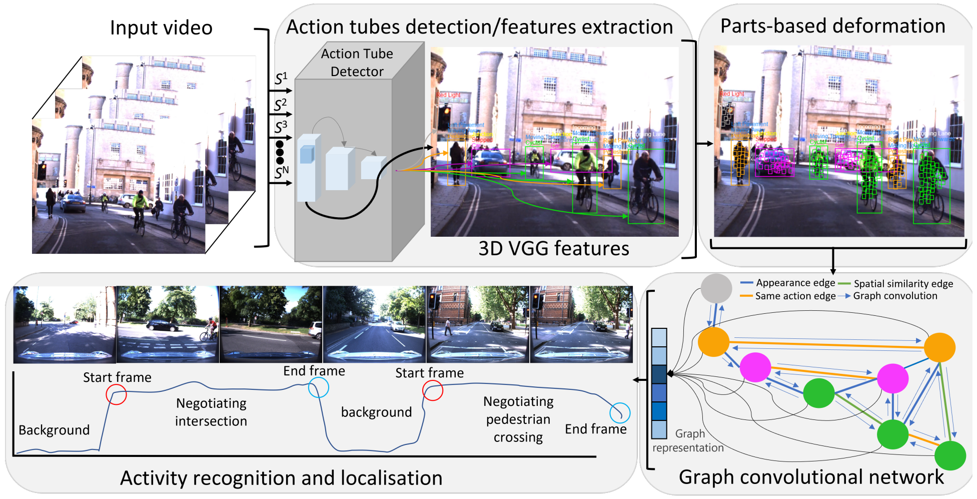

Figure 1: Overall pipeline of our long-term complex activity detection framework. (i) The input video is first divided into s nippets. (ii) Snippets are passed to an action tube detector module one by one. (iii) The parts-based deformation module receives 3D VGG features and action tube locations and 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.

v1.3 [5]. 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 autonomous driving and surgical robotics but of general applicability and extendable to other domains. For instance, one of the complex activities in autonomous driving detect by our proposed method is *Negotiating intersection* which is made up of the following 'atomic' actions: *AV-move* + *Vehicle traffic light / Green* + *AV-stop* + *Vehicle(s) / Stopped / At junction*+ *AV-move*. 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 prepossessing step, aimed at spatially and temporally locating the atomic actions present [3, 12, 24, 33, 34, 40, 41]. The tube detector needs to ensure a fixed-size representation for each activity part (atomic action). Here in particular we adopt AMTNet [35], 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, we propose a Graph Convolutional Network (GCN) module in which a graph is constructed by considering individual tubes as nodes and connecting them with 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 datasets related to completely different domains: ROAD [42] for situation awareness in autonomous driving and SARAS-ESAD [2] for surgical action detection, which both provide 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 activities (road activities in ROAD vs surgical phases in SARAS-ESAD). The main contributions of this paper are therefore:

- A novel framework for long-term complex activity recognition and localisation.
- An original deformable 3D RoI pooling approach for flexibly pooling features from the components of 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 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 deformable pooling of features and a spatiotemporal graph representation which employs multiple semantics.

3.1 Action Tube Detection

To provide a fixed-size representation for the instances of atomic actions composing a complex activity, 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, and add four extra convolutional layers at the end. AMTNet takes sequence of RGB frames as an input with a fixed temporal interval $\triangle$ between consecutive frames, i.e., $\{f_t, f_{t+\triangle}\}$. 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 (300×300 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 12 frames, aligned

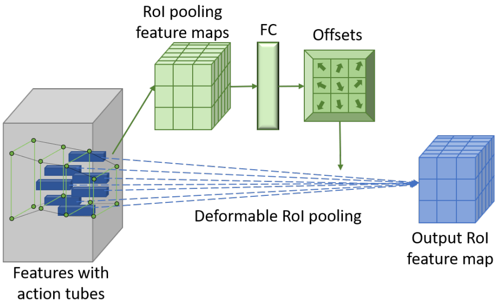

Figure 2: Our 3D deformable RoI pooling layer takes feature maps and action tube locations as input and arranges them into a fixed-size grid of components (here illustrated for size 3 × 3). For each grid component an offset is generated and multiplied by the original tube feature to produce the final component features.

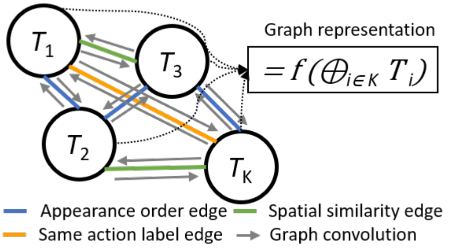

Figure 3: Our graph module takes as input the features generated for each tube by the ROI pooling layer $T_{1,2,3...K}$ and builds edges between them according to different semantics (appearance order, spatial similarity, and same action label). The overall graph is processed by a GCN to deliver a fixed-size graph representation.

with our snippet length. Complete action tubes are incrementally generated by AMTNet by temporally linking the micro-tubes predicted by the network.

3.2 3D Part-Based Deformable Layer

The feature extractor in our framework is a novel 3D deformable RoI pooling layer which encodes 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] that has the ability to extract and learn features from an action tube rather than a 2D bounding box. The main rational behind using 3D deformable RoI pooling layer is to learn the geometric transformation of each atomic action during the training time rather than using the feature of pretrained model. 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.

Firstly, standard RoI pooling is applied to the provided feature map X and bounding boxes locations forming an action tube ($M \times [x,y,w,h]$), by subdividing the tube into a pooled feature map grid of fixed-size in both the spatial and the temporal dimension $M \times k \times k$. M is the snippet size fixed during action tube detection while k is a free parameter for selecting the bin size in RoI pooling. 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 the element-wise product with original RoI's width and height. Offsets are also multipled by a scalar value to modulate the magnitude of offsets (empircally set to 0.1) which makes the it invariant to different sizes of 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 shape 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) 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. For each snippet, a separate graph is built with a variable number of nodes corresponding to the number of detected activity parts (action tubes). The initial representation of the nodes is provided by their ROI features. We consider three different types of connections: (i) the *order of appearance* (in a specified direction) 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 the same action label, as this provides very relevant information for the determination of the activity class. As a result three heterogeneous spatiotemporal scene graphs are constructed having the same nodes but with different edges. These graphs are then combined by taking a union of all edges to create a single homogeneous graph representing the overall scene.

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 representation, resulting in a fixed-sized feature representation which is 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 $S_{1,2,3,...,N}$, 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 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., 7×7) grid of components whose dimensionality is equal to the number (64) of convolutional layers: ${X_{DRoI}}^{K \times 64 \times M \times 7 \times 7}$. These features are then fed to our 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 with a dual verification mechanism. Namely, since we target long-term activities, if there is a random false positive or a random false negative between two same class snippets we simply ignore it and consider it as a same activity.

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 and extract tube parts of a tube rather than of a 2D object. In our experiments we also convert a more recent version of deformable RoI pooling called 'modulated' deformable RoI pooling to the 3D case. In the GCN module we construct graphs online for each snippet during training using a PyTorch data loader [32]. For the design of the GCN architecture we use 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, architecture is implemented using the PyTorch [32] deep learning library 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, in which 12 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’ 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 the **supplementary material**.

SARAS-ESAD [2]: ESAD (Endoscopic Surgeon Action Detection Dataset) is a benchmark devised for surgeon action detection in real-world endoscopic surgery videos. In ESAD, surgeon actions are divided 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 (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 reported in the **supplementary material**. For more details please see [2].

Evaluation Metrics: For the evaluation of action tube detection performance we use the standard frame/video mean Average Precision (*mAP*) with different IoU thresholds δ, namely 0.2, 0.3, 0.5, 0.75, on both datasets. Complex activity recognition is evaluated using classification accuracy, precision, recall and F-score. For complex activity localisation we use 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

	ROAD				SARAS-ESAD			
Methods / IoU threshold δ	0.2	0.3	0.5	0.75	0.2	0.3	0.5	0.75
Singh et al. [42] (frame-*mAP*)	-	-	25.9	-	-	-	-	-
Singh et al. [42] (video-*mAP*)	17.5	-	4.6	-	-	-	-	-
Bawa et al. [2] (frame-*mAP*)	-	-	-	-	-	24.3	12.2	-
AMTNet (frame-*mAP*)	22.3	18.1	15.4	11.0	30.4	24.6	18.7	7.9
AMTNet (video-*mAP*)	11.6	7.9	3.8	-	13.7	10.1	8.8	5.4

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.

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.

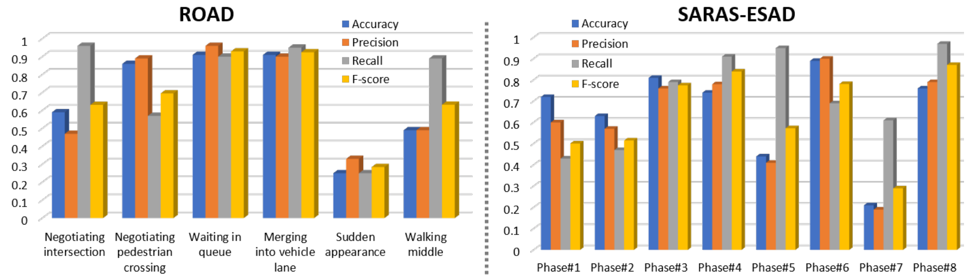

Figure 4: Complex activity classification performance on both ROAD and SARAS-ESAD.

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 both datasets are exposed for the first time to complex activity detection. The major changes we made during reimplementation are: (i) data loading (as both methods are designed to train and test on already extracted features), and (ii) replacing the regression part with a sliding window approach for the SARAS-ESAD dataset, as the latter lacks a background class.

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. In the first

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

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

Method / Activities	Negotiating intersection	Negotiating pedestrian crossing	Waiting in queue	Merging into vehicle lane	Sudden appearance	Walking middle of road
P-GCN [55]	44.3	53.8	74.4	50.1	21.7	34.1
G-TAD [54]	47.8	57.3	70.6	55.2	**24.3**	37.1
Ours	**51.2**	**72.3**	**89.8**	**84.1**	17.8	**41.3**

Table 3: ROAD activity localisation performance (*mAP*, %) for each complex road activity, at a standard IoU threshold of 0.5.

case, when no action tube is detected, we used the entire frame as RoI for our parts deformation module to understand the overall scene. The next scenario is considered as same as other activities to learn the model for actions without contribution to any activity.

Temporal activity detection on ROAD, measured via *mAP* at five different IoU thresholds, is reported in Table 2 for 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. For localisation, however, it is important to detect start and end time of each phase. 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 can be clearly observed that 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 and, from our results, the existing tube detectors are not reliable enough to perform well over challenging real-world datasets such as those we chose as benchmarks. 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.

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

Method / Activities	Phase#1	Phase#2	Phase#3	Phase#4	Phase#5	Phase#6	Phase#7	Phase#8
P-GCN [55]	56.7	43.2	52.3	59.1	**33.8**	59.4	14.8	41.2
G-TAD [54]	51.1	46.6	57.2	63.8	29.4	62.2	**19.3**	45.7
Ours	**57.5**	**54.1**	**69.3**	**60.2**	31.1	**71.3**	16.5	**52.4**

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

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 element of a dynamic scene, such as object, agent, action, location, 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 that make them suitable candidates to be used as 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.

References

[1] Mohammad M Arzani, Mahmood Fathy, Ahmad A Azirani, and Ehsan Adeli. Switching structured prediction for simple and complex human activity recognition. *IEEE transactions on cybernetics*, 2020.

[2] Vivek Singh Bawa, Gurkirt Singh, Francis KapingA, Alice Leporini, Carmela Landolfo, Armando Stabile, Francesco Setti, Riccardo Muradore, Elettra Oleari, Fabio Cuzzolin, et al. Esad: Endoscopic surgeon action detection dataset. *arXiv preprint arXiv:2006.07164*, 2020.

[3] Harkirat Singh Behl, Michael Sapienza, Gurkirt Singh, Suman Saha, Fabio Cuzzolin, and Philip HS Torr. Incremental tube construction for human action detection. *arXiv preprint arXiv:1704.01358*, 2017.

[4] Pratool Bharti, Debraj De, Sriram Chellappan, and Sajal K Das. Human: Complex activity recognition with multi-modal multi-positional body sensing. *IEEE Transactions on Mobile Computing*, 18(4):857–870, 2018.

[5] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 961–970, 2015.

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[7] Fanta Camara, Nicola Bellotto, Serhan Cosar, Florian Weber, Dimitris Nathanael, Matthias Althoff, Jingyuan Wu, Johannes Ruenz, André Dietrich, Gustav Markkula,

et al. Pedestrian models for autonomous driving part ii: high-level models of human behavior. *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[8] Lei Chen, Jiwen Lu, Zhanjie Song, and Jie Zhou. Part-activated deep reinforcement learning for action prediction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 421–436, 2018.

[9] Yunpeng Chen, Marcus Rohrbach, Zhicheng Yan, Yan Shuicheng, Jiashi Feng, and Yannis Kalantidis. Graph-based global reasoning networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 433–442, 2019.

[10] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[11] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.

[12] Rocco De Rosa, Nicolò Cesa-Bianchi, Ilaria Gori, and Fabio Cuzzolin. Online action recognition via nonparametric incremental learning. In *BMVC*, 2014.

[13] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. doi: 10.1109/CVPR.2008.4587597.

[14] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[15] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. Cascade object detection with deformable part models. In *2010 IEEE Computer society conference on computer vision and pattern recognition*, pages 2241–2248. IEEE, 2010.

[16] Jiyang Gao, Zhenheng Yang, Kan Chen, Chen Sun, and Ram Nevatia. Turn tap: Temporal unit regression network for temporal action proposals. In *Proceedings of the IEEE international conference on computer vision*, pages 3628–3636, 2017.

[17] Ross Girshick, Forrest Iandola, Trevor Darrell, and Jitendra Malik. Deformable part models are convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 437–446, 2015.

[18] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9785–9795, 2019.

[19] Guoqiang Gong, Xinghan Wang, Yadong Mu, and Qi Tian. Learning temporal co-attention models for unsupervised video action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9819–9828, 2020.

[20] Jun-Wei Hsieh, Chi-Hung Chuang, Sin-Yu Chen, Chih-Chiang Chen, and Kuo-Chin Fan. Segmentation of human body parts using deformable triangulation. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 40(3):596–610, 2010.

[21] Guyue Hu, Bo Cui, Yuan He, and Shan Yu. Progressive relation learning for group activity recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 980–989, 2020.

[22] Yifei Huang, Yusuke Sugano, and Yoichi Sato. Improving action segmentation via graph-based temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14024–14034, 2020.

[23] Haroon Idrees, Amir R Zamir, Yu-Gang Jiang, Alex Gorban, Ivan Laptev, Rahul Sukthankar, and Mubarak Shah. The thumos challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23, 2017.

[24] Saumya Jetley and Fabio Cuzzolin. 3d activity recognition using motion history and binary shape templates. In *Asian Conference on Computer Vision*, pages 129–144. Springer, 2014.

[25] Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10247, 2020.

[26] Hyeokhyen Kwon, Catherine Tong, Harish Haresamudram, Yan Gao, Gregory D Abowd, Nicholas D Lane, and Thomas Ploetz. Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–29, 2020.

[27] Jin Li, Xianglong Liu, Zhuofan Zong, Wanru Zhao, Mingyuan Zhang, and Jingkuan Song. Graph attention based proposal 3d convnets for action detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 4626–4633, 2020.

[28] Junwei Liang, Lu Jiang, Juan Carlos Niebles, Alexander G Hauptmann, and Li Fei-Fei. Peeking into the future: Predicting future person activities and locations in videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5725–5734, 2019.

[29] Xingyu Liu, Joon-Young Lee, and Hailin Jin. Learning video representations from correspondence proposals. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4273–4281, 2019.

[30] Fuchen Long, Ting Yao, Zhaofan Qiu, Xinmei Tian, Jiebo Luo, and Tao Mei. Learning to localize actions from moments. *arXiv preprint arXiv:2008.13705*, 2020.

[31] Chenxu Luo and Alan L Yuille. Grouped spatial-temporal aggregation for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5512–5521, 2019.

[32] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learn
pdf.

[33] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. *arXiv preprint arXiv:1608.01529*, 2016.

[34] Suman Saha, Gurkirt Singh, and Fabio Cuzzolin. Amtnet: Action-micro-tube regression by end-to-end trainable deep architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4414–4423, 2017.

[35] Suman Saha, Gurkirt Singh, and Fabio Cuzzolin. Two-stream amtnet for action detection. *arXiv preprint arXiv:2004.01494*, 2020.

[36] Baifeng Shi, Qi Dai, Yadong Mu, and Jingdong Wang. Weakly-supervised action localization by generative attention modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1019, 2020.

[37] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1049–1058, 2016.

[38] Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision*, pages 510–526. Springer, 2016.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[40] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3637–3646, 2017.

[41] Gurkirt Singh, Suman Saha, and Fabio Cuzzolin. Predicting action tubes. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.

[42] Gurkirt Singh, Stephen Akrigg, Manuele Di Maio, Valentina Fontana, Reza Javanmard Alitappeh, Suman Saha, Kossar Jeddisaravi, Farzad Yousefi, Jacob Culley, Tom Nicholson, et al. Road: The road event awareness dataset for autonomous driving. *arXiv preprint arXiv:2102.11585*, 2021.

[43] Vivek Singh Bawa, Gurkirt Singh, Francis KapingA, Inna Skarga-Bandurova, Elettra Oleari, Alice Leporini, Carmela Landolfo, Pengfei Zhao, Xi Xiang, Gongning Luo, et al. The saras endoscopic surgeon action detection (esad) dataset: Challenges and methods. *arXiv e-prints*, pages arXiv–2104, 2021.

[44] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[45] Swathikiran Sudhakaran, Sergio Escalera, and Oswald Lanz. Lsta: Long short-term attention for egocentric action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9954–9963, 2019.

[46] Nirmalya Thakur and Chia Y Han. An improved approach for complex activity recognition in smart homes. In *International Conference on Software and Systems Reuse*, pages 220–231. Springer, 2019.

[47] Limin Wang, Yu Qiao, and Xiaoou Tang. Action recognition and detection by combining motion and appearance features. *THUMOS14 Action Recognition Challenge*, 1(2): 2, 2014.

[48] Minjie Wang, Lingfan Yu, Da Zheng, Quan Gan, Yu Gai, Zihao Ye, Mufei Li, Jinjing Zhou, Qi Huang, Chao Ma, Ziyue Huang, Qipeng Guo, Hao Zhang, Haibin Lin, Junbo Zhao, Jinyang Li, Alexander J Smola, and Zheng Zhang. Deep graph library: Towards efficient and scalable deep learning on graphs. *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019. URL https://arxiv.org/abs/1909.01315.

[49] Xiaolong Wang and Abhinav Gupta. Videos as space-time region graphs. In *Proceedings of the European conference on computer vision (ECCV)*, pages 399–417, 2018.

[50] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.

[51] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.

[52] Zhuyang Xie, Junzhou Chen, and Bo Peng. Point clouds learning with attention-based graph convolution networks. *Neurocomputing*, 402:245–255, 2020.

[53] Huijuan Xu, Abir Das, and Kate Saenko. R-c3d: Region convolutional 3d network for temporal activity detection. In *Proceedings of the IEEE international conference on computer vision*, pages 5783–5792, 2017.

[54] Mengmeng Xu, Chen Zhao, David S Rojas, Ali Thabet, and Bernard Ghanem. G-tad: Sub-graph localization for temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10156–10165, 2020.

[55] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019.

[56] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 793–803, 2019.

[57] Hang Zhao, Antonio Torralba, Lorenzo Torresani, and Zhicheng Yan. Hacs: Human action clips and segments dataset for recognition and temporal localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8668–8678, 2019.

[58] Xiaokang Zhou, Wei Liang, I Kevin, Kai Wang, Hao Wang, Laurence T Yang, and Qun Jin. Deep-learning-enhanced human activity recognition for internet of healthcare things. *IEEE Internet of Things Journal*, 7(7):6429–6438, 2020.

[59] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019.

[60] Aneeq Zia, Andrew Hung, Irfan Essa, and Anthony Jarc. Surgical activity recognition in robot-assisted radical prostatectomy using deep learning. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 273–280. Springer, 2018.