Predicting Action Tubes

Gurkirt Singh, Suman Saha, and Fabio Cuzzolin

Oxford Brookes University, UK
gurkirt.singh-2015@brookes.ac.uk

Abstract. In this work, we present a method to predict an entire 'action tube' (a set of temporally linked bounding boxes) in a trimmed video just by observing a smaller subset of it. Predicting where an action is going to take place in the near future is essential to many computer vision based applications such as autonomous driving or surgical robotics. Importantly, it has to be done in real-time and in an online fashion. We propose a **T**ube **P**rediction network (TPnet) which jointly predicts the past, present and future bounding boxes along with their action classification scores. At test time TPnet is used in a (temporal) sliding window setting, and its predictions are put into a tube estimation framework to construct/predict the video long action tubes not only for the observed part of the video but also for the unobserved part. Additionally, the proposed action tube predictor helps in completing action tubes for unobserved segments of the video. We quantitatively demonstrate the latter ability, and the fact that TPnet improves state-of-the-art detection performance, on one of the standard action detection benchmarks - J-HMDB-21 dataset.

1 Introduction

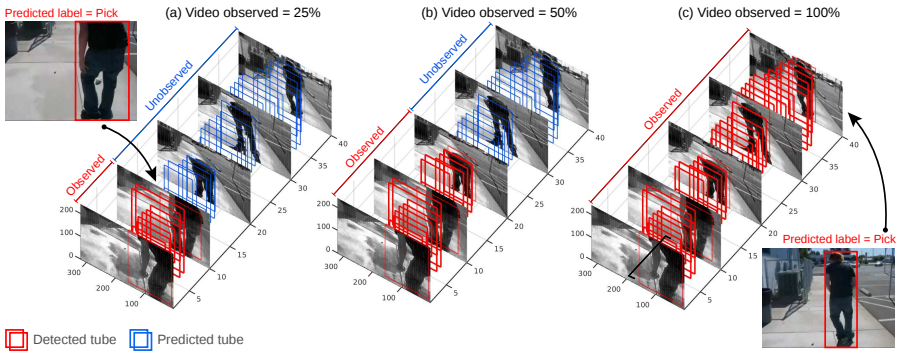

Fig. 1. *An Illustration of the action tube prediction problem using an example in which a "pickup" action is being performed on a sidewalk. As an ideal case, we want the system to predict an action tube as shown in (c) (i.e. when 100% of the video has been processed) just by observing 25% of the entire clip (a). We want the tube predictor to predict the action class label (shown in red) alongside predicting the spatial location of the tube. The red shaded bounding boxes denote the detected tube in the observed portion of the input video, whereas, the blue coloured bounding boxes represent the future predicted action tube for the unobserved part of the clip.*

Imagine a pedestrian on the sidewalk, and an autonomous car cruising on the nearby road. If the pedestrian stays on the sidewalk and continues walking, they are of no concern for the self-driving car. What, instead, if they start approaching the road, in a possible attempt to cross it? Any future prediction about the pedestrian's action and their possible position on/off the road would crucially help the autonomous car avoid any potential incident. It would suffice to foresee the pedestrian's action label and position half a second early to avoid a major accident. As a result, awareness about surrounding human actions is essential for the robot-car.

We can formalise the problem as follows. We seek to predict both the class label and the future spatial location(s) of an action instance as early as possible, as shown in Figure 1. Basically, it translates into early spatiotemporal action detection [31], achieved by completing action instance(s) for the unobserved part of the video. As commonly accepted, action instances are here described by 'tubes' formed by linking bounding box detections in time.

In an existing relevant work by Singh *et al.* [31], early label prediction and online action detection are performed jointly. The action class label for an input video is predicted early on just by observing a smaller portion (a few frames) of it, whilst the system incrementally builds action tubes in an online fashion. In contrast, the proposed approach can predict both the class label of an action and its future location(s) (i.e., the future shape of an action tube). In this work, by *'prediction'* we refer to the estimation of both an action's label and location in *future*, unobserved video segments. We term *'detection'* the estimation of action labels/locations in the observed segment of video up to any given time, i.e., for *present* and *past* video frames.

The computer vision community is witnessing a rising interest in problems such as early action label prediction [31, 24, 9, 2, 32, 39, 40, 16, 42, 28, 20], online temporal action detection [23, 39, 6, 33], online spatio-temporal action detection [31, 32, 38], future representation prediction [34, 16] or trajectory prediction [1, 15, 21]. Although, all these problems are interesting, and definitely encompass a broad scope of applications, they do not entirely capture the complexity involved by many critical scenarios including, e.g., surgical robotics or autonomous driving. In opposition to [31, 32], which can only perform early label prediction and online action detection, in this work we propose to predict both future action location and action label. A number of challenges make this problem particularly hard, e.g., the temporal structure of an action is obviously not completely observed; locating human actions is itself a difficult task; the observed part can only provide clues about the future locations. In addition, camera movement can make it even harder to extrapolate an entire tube. We propose to solve these problems by regressing the future locations from the present tube.

The ability to predict *action micro-tubes* (sets of temporally connected bounding boxes spanning k video frames) from pairs of frames [29] or sets of k frames [13, 10] provides a powerful tool to extend the single frame-based online approach by Singh *et al.* [31] in order to cope with action location prediction, while retaining its incremental nature. Combining the basic philosophies of [31] and [29] has thus the potential to provide an interesting and scalable approach to action prediction.

Briefly, the action micro-tubes network (AMTnet, [29]), divides the action tube detection problem into a set of smaller sub-problems. Action 'micro-tubes' are produced

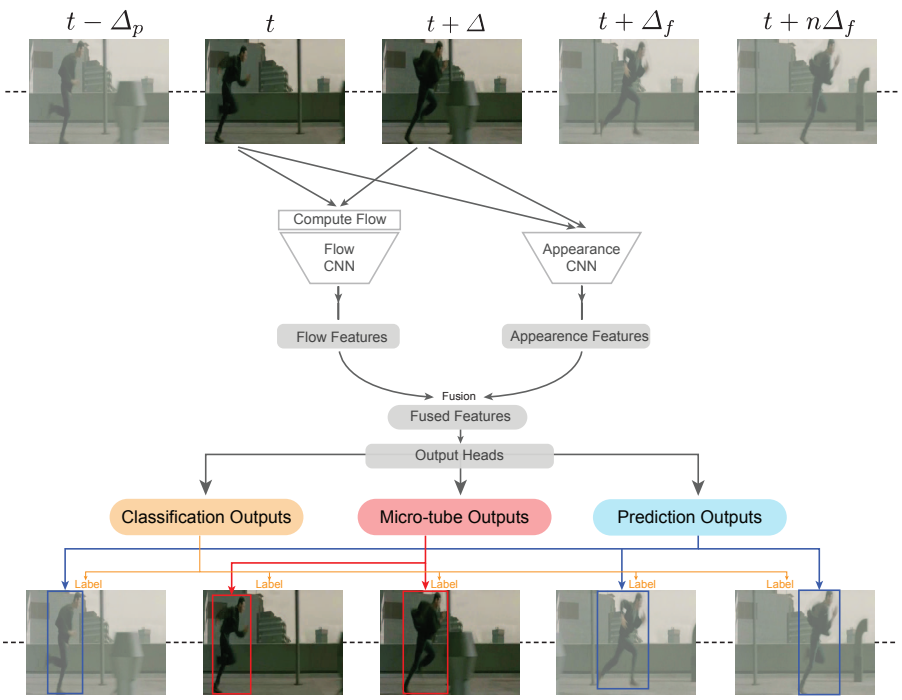

Fig. 2. *Workflow illustrating the application of TPnet to a test video at a time instant t. The network takes frames f_t and $f_{t+\Delta}$ as input and generates classification scores, the micro-tube (in red) for frames f_t and $f_{t+\Delta}$, and prediction bounding boxes (in blue) for frames $f_{t-\Delta_p}$, $f_{t+\Delta_f}$ up to $f_{t+n\Delta_f}$. All bounding boxes are considered to be linked to the micro-tube. Note that predictions also span the past: a setting called* smoothing *in the estimation literature. Δ_p, Δ_f and n are network parameters that we cross-validate during training.*

by a convolutional neural network (a 3D region proposal network) processing two input frames that are Δ apart. Each micro-tube consists of two bounding boxes belonging to the two frames. When the network is applied to consecutive pairs of frames, it produces a set of consecutive micro-tubes which can be finally linked [31] to form complete action tubes. The detections forming a micro-tube can be considered as implicitly linked, hence reducing the number of linking subproblems. Whereas AMTnet was originally designed to generate micro-tubes using only appearance (RGB) inputs, here we augment it by introducing the feature-level fusion of flow and appearance cues, drastically improving its performance and, as a result, that of TPnet.

Concept: We propose to extend the action micro-tube detection architecture by Saha *et al.* [29] to produce, at any time t, past ($\tau < t$), present, and future ($\tau > t$) detection bounding boxes, so that each (extended) micro-tube contains bounding boxes for both observed and not yet observed frames. All bounding boxes, spanning presently observed frames as well as past and future ones (in which case we call them predicted bounding boxes), are considered to be linked, as shown in blue in Figure 2.

We call this new deep network 'TPnet'.

Once bounding boxes are regressed, the online tube construction method of Singh *et*

al. [31] can be incrementally applied to the observed part of the video to generate one or more 'detected' action tubes at any time instant t.
Further, in virtue of TPnet and online tube construction, the temporally linked micro-tubes forming each currently detected action tube (spanning the observed segment of the video) also contain past and future estimated bounding boxes. As these predicted boxes are implicitly linked to the micro-tubes which compose the presently detected tube, the problem of linking to the latter the future bounding boxes, leading to a whole action tube, is automatically addressed.

The proposed approach provides two main benefits: i) future bounding box predictions are implicitly linked to the present action tubes; ii) as the method relies only on two consecutive frames separated by a constant distance Δ, it is efficient enough to be applicable to real-time scenarios.

Contributions: In Summary we present a Tube Predictor network (TPnet) which:

- given a partially observed video, can (early) predict video long action tubes in terms of both their classes and the constituting bounding boxes;
- demonstrates that training a network to make predictions also helps in improving action detection performance;
- demonstrates that feature-based fusion works better than late fusion in the context of spatiotemporal action detection.

2 Related work

Early label prediction. Early, online action label prediction has been studied using dynamic bag of words [28], structured SVMs [9], hierarchical representations [20], LSTMs[39] and Fisher vectors [6]. Recently, Yeung *et al.* [39, 40] have proposed a variant of long short-term memory (LSTM) deep networks for modelling these temporal relations via multiple input and output connections. Kong *et al.* [16], instead, make use of variational auto-encoders to predict a representation for the whole video and use it to determine the action category for the whole video as early as possible. Probabilistic approaches based on Bayesian networks [24], Conditional Random Fields [17] or Gaussian processes [12] may help in activity anticipation. However, inference in such generative approaches is often expensive. None of these methods address the full online label and spatiotemporal location prediction setting considered here.

Online Action Detection. Soomro *et al.* [32] have recently proposed an online method which can predict an action's label and detect its location by observing a relatively smaller portion of the entire video sequence. They use segmentation to perform online detection via SVM models trained on fixed length segments of the training videos. Similarly, Singh *et al.* [31] have extended online action detection to untrimmed videos with help of an online tube construction algorithm built on the top of frame-level action bounding box detections. Similiarly, Behl *et al.* [3] solve online detection with help of tracking formulation. However, these approaches [32, 31, 3] only perform action localisation for the observed part of the video and adopt the label predicted for the currently detected tube as the label for the whole video.

To the best of our knowledge, no existing method generates predictions concerning both labels and action tube geometry. Interestingly, Yang *et al.*[38] use features from

current, frame t proposals to 'anticipate' region proposal locations in $t + \Delta$ and to generate detections at time $t + \Delta$, thus failing to take full advantage of the anticipation trick to predict the future spatiotemporal extent of the action tubes.

Advances in action recognition are always going to be helpful in action prediction from a general representation learning point of view. For instance, Gu *et al.*[8] have recently improved on [25, 13] by plugging in the inflated 3D network proposed by [5] as a base network on multiple frames. Although they use a very strong base network pre-trained on the large "Kinetics" [14] dataset, they do not handle the linking process within the network as the AVA [8] dataset's annotations are not temporally linked. Analogously, learning to predict future representation [34] can be useful in general action prediction (cfr. e.g. [16]).

Recently, inspired by the record-breaking performance of CNN-based object detectors [26, 27, 22], a number of scholars [31, 30, 7, 25, 35, 37, 41, 3] have tried to extend frame-level object detectors to videos for spatio-temporal action localisation. These approaches, however, fail to tackle spatial and temporal reasoning jointly at the network level, as spatial detection and temporal association are treated as two disjoint problems. More recent works have attempted to address this problem by reducing the amount of linking required with the help of 'micro-tubes' [29] or 'tubelets' [13, 10] for small sets of frames taken together, where micro-tube boxes from different frames are considered to be linked together. AMTnet [29] by Saha *et al.* is particularly interesting, because of its compact (GPU memory-wise) and flexible nature, as it can exploit pairs of successive frames Δ sampling intervals apart, that it can also leverage sparse annotations [36] as well. For these reasons in this work we build on AMTnet as base network, improving its feature representation by feature-level fusion of motion and appearance cues.

3 Methodology

In this section, we describe our tube prediction framework for the problem formulation described in § 3.1. Our approach has four main components. Firstly, we tie the future action tube prediction problem (§ 3.1) with action micro-tube [29] detection. Secondly, we devise our tube prediction network (TPnet) to predict future bounding boxes along with current micro-tubes, and describe its training procedure in § 3.3. Thirdly, we use TPnet in a sliding window fashion (§ 3.4) in the temporal direction while generating micro-tubes and corresponding future predictions. These, eventually, are fed to a tube prediction framework (§ 3.4) to generate the future of any current action tube being built using micro-tubes.

3.1 Problem Statement

We define an *action tube* as a connected sequence of detection boxes in time without interruptions and associated with a same action class c, starting at first frame f_1 and ending last frame f_T, in trimmed video: $\mathcal{T}_c = \{b_1, ...b_t, ...b_T\}$. Tubes are constrained to span the entire video duration, like in [7]. At any time point t, a tube is divided into two parts, one needs to be detected $\mathcal{T}_c^d = \{b_1, ...b_t\}$ up to f_t and another part needs to

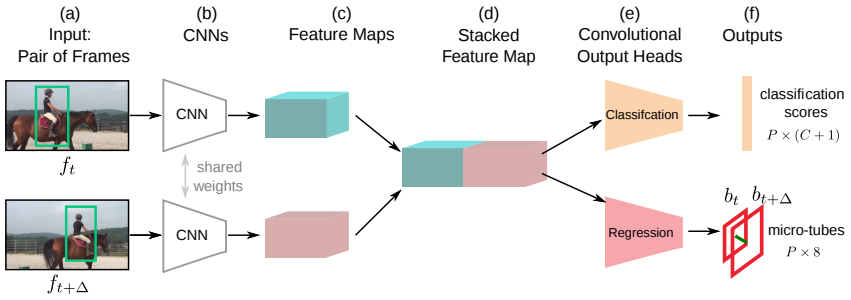

Fig. 3. *Overview of the action micro-tube detection network (AMTnet). As it only predicts micro-tubes and their scores, here we modify it to predict the future locations associated with the given micro-tubes, as shown in Figure 2.*

be predicted/estimated $\mathcal{T}_c^p = \{b_{t+1}, ...b_T\}$ from frame f_{t+1} to f_T along with its class c. The observed part of the video is responsible for generating $\mathcal{T}_c^d$ (red in Fig 1), while we need to estimate the future section of the tube $\mathcal{T}_c^p$ (blue in Fig 1) for the unobserved segment of the video. The first sub-problem, the online detection of $\mathcal{T}_c^d$, is explained in § 3.2. The second sub-problem (the estimation of the future tube segment $\mathcal{T}_c^p$) is tackled by a tube prediction network (TPnet, § 3.3) in a novel tube prediction framework (§ 3.4).

3.2 From micro-tubes to full action tubes

Saha *et al.* [29] introduced *micro-tubes* in their action micro-tube network (AMTnet) proposal, shown in Figure 3. AMTnet decomposes the problem of detecting $\mathcal{T}_c$ into a set of smaller problems, detecting micro-tubes $m_t = \{b_t, b_{t+\Delta}\}$ at time t along with their classification scores for $C+1$ classes, using two successive frames f_t and $f_{t+\Delta}$ as an input (Fig. 3(a)). Subsequently, the detection micro-tubes $\{m_1...m_{t-\Delta}\}$ are linked up in time to form action tube $\mathcal{T}_c^d$. Similar to [22], one background class is added to the class list which takes the number classes to $C+1$.

AMTnet employs two parallel CNN streams (Fig. 3(b)), one for each frame, which produce two feature maps (Fig. 3(c)). These feature maps are stacked together into one (Fig 3(d)). Finally, convolutional heads are applied in a sliding window (spatial) fashion over predefined 3×3 anchor regions [22], which correspond to P prior [22] or anchor [27] boxes. Convolutional heads produce a $P \times 8$ output per micro-tube (Fig. 3(f)) and $P \times (C+1)$ corresponding classification scores (Fig 3(g)). Each micro tube has 8 coordinate, 4 for the bounding box b_t in frame f_t and 4 for bounding box $b_{t+\Delta}$ in frame $f_{t+\Delta}$. As shown in Figure 3(f), the pair of boxes can be considered as implicitly linked together, hence the name micro-tube.

Originally, Saha *et al.* [29] employed FasterRCNN [27] as base detection architecture. Here, however, we switch to Single Shot Detector (SSD) [22] as a base detector for efficiency reasons. Singh *et al.* [31] used SSD to propose an online and real-time action tube generation algorithm, while Kalogeiton *et al.* [13] adapted SSD to detect micro-tubes (or, in their terminology, 'tubelets') k frames long.

More importantly, we make two essential changes to AMTnet. Firstly, we enhance its

feature representation power by fusing appearance features (based on RGB frames) and flow features (based on optical flow) at the feature level (see the fusion step shown in Fig 4), unlike the late fusion approach of [13] and [31]. Note that the original AMTnet framework does not make use of optical flow at all. We will show that feature-level fusion dramatically improves its performance. Secondly, the AMTnet-based tube detection framework proposed in [29] is offline, as micro-tube linking is done recursively in an offline fashion [7]. Similar to Kalogeiton *et al.* [13], we adapt the online linking method of [31] to link micro-tubes in to a tube $\mathcal{T}_c^d$.

Micro-tube linking details: Let B_t be the set of detection bounding boxes from frame f_t, and B_{t+1} the corresponding set from f_{t+1}, generated by a frame-level detector. Singh *et al.* [31] associate boxes in B_t to boxes in B_{t+1}, whereas, in our case, we need to link micro-tubes $m_t \in M_t \doteq B_t^1 \times B_{t+\Delta}^2$ from a pair of frames $\{f_t, f_{t+\Delta}\}$ to micro-tubes $m_{t+\Delta} \in M_{t+\Delta} \doteq B_{t+\Delta}^1 \times B_{t+2\Delta}^2$ from the next set of frames $\{f_{t+\Delta}, f_{t+2\Delta}\}$. This happens by associating elements of $B_{t+\Delta}^2$, coming from M_t, with elements of $B_{t+\Delta}^1$, coming from $M_{t+\Delta}$. Interestingly, the latter is a relatively easier sub-problem, as all such detections are generated based on the same frame, unlike the across frame association problem considered in [31]. The association is achieved based on Intersection over Union (IoU) and class score, as the tubes are built separately for each class in a multi-label scenario. For more details, we refer the reader to [31].

Since we adopt the online linking framework of Singh *et al.* [31], we follow most of the linking setting used by them, e.g.: linking is done for every class separately; the non-maximal threshold is set to 0.45. As shown in Figure 5(a) to 5(b), the last box of the first micro-tube (red) is linked to the first box of next micro-tube (red). So, the first set of micro-tubes is produced at f_1, the following one at f_Δ the one after that at $f_{2\Delta}$, and so on. As a result, the last micro-tube is generated at $f_{t-\Delta}$ to cover the observable video duration up to time t. Finally, we solve for the association problem as described above.

3.3 Training the tube prediction network (TPnet)

AMTnet allow us to detect current tubes $\mathcal{T}_c^d$ by generating a set of successive micro-tubes $\{m_1 ... m_{t-\Delta}\}$, where $m_{t-\Delta} = \{b_{t-\Delta}, b_t\}$. However, our aim is to predict the future section $\mathcal{T}_c^p$ of the tube using the latter linked micro-tubes, up to time t.

To address this problem, we propose a tube prediction framework aimed at simultaneously estimating a micro-tube m_t, a set $z_t = \{b_{t-\Delta_p}, b_{t+\Delta_f}, ... b_{t+n\Delta_f}\}$ of *past and future detections*, and the classification scores for the $C+1$ classes. Δ_p measures how far in the past we are looking into, whereas Δ_f is a future step size, and n is the number of future steps. This is performed by a new Tube Prediction network (TPnet).

The underlying architecture of TPnet is shown in Figure 4. TPnet takes two successive frames from time t and $t+\Delta$ as input. The two input frames are fed to two parallel CNN streams, one for appearance and one for optical flow. The resulting feature maps are fused together, either by concatenating or by element-wise summing the given feature maps. Finally, three types of convolutional output heads are used for P prior boxes as shown in Figure 4. The first one produces the $P \times (C+1)$ classification outputs; the second one regresses the $P \times 8$ coordinates of the micro-tubes, as in AMTnet; the last one regresses $P \times (4(1+n))$ coordinates, where 4 coordinates correspond to the frame

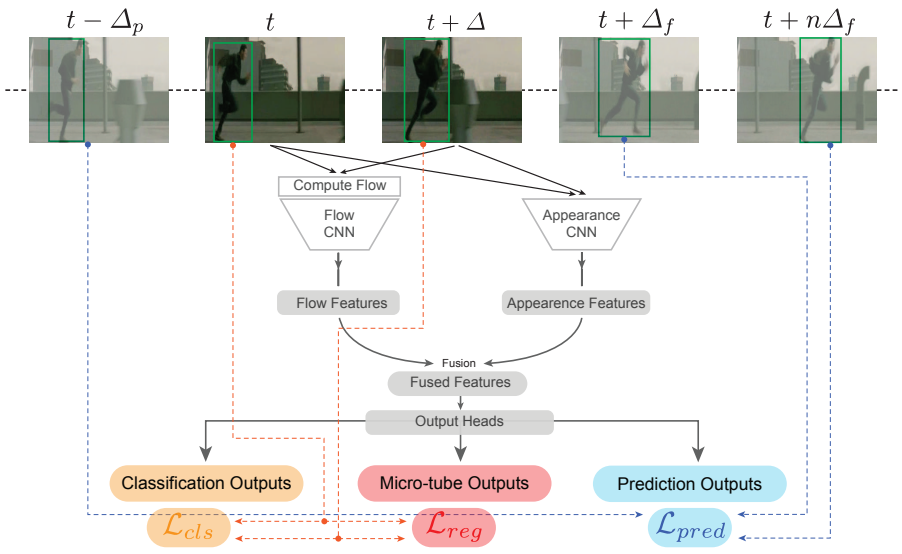

Fig. 4. *Overview of the tube prediction network (TPnet) architecture at training time.*

at $t-\Delta_p$, and the remaining $4n$ are associated with the n future steps. The training procedure of the new architecture is illustrated below.

Multi-task learning TPnet is designed to strive for three objectives, for each prior box p. The first task (i) is to classify the P prior boxes; the second task (ii) is to regress the coordinates of the micro-tubes; the last (iii) is to regress the coordinates of the past and future detections associated with each micro-tube.

Given a set of P anchor boxes and the respective outputs we compute a loss following the training objective of SSD [22]. Let $x^c_{i,j} = \{0,1\}$ be the indicator for matching the i-th prior box to the j-th ground truth box of category c. We use the bipartite matching procedure described in [22] for matching the ground truth micro-tubes $G = \{g_t, g_{t+\Delta}\}$ to the prior boxes, where g_t is a ground truth box at time t. The overlap is computed between a prior box p and micro-tube G as the mean IoU between p and the ground truth boxes in G. A match is defined as positive ($x^c_{i,j} = 1$) if the overlap is more than or equal to 0.5.

The overall loss function $\mathcal{L}$ is the following weighted sum of classification loss ($\mathcal{L}_{cls}$), micro-tube regression loss ($\mathcal{L}_{reg}$) and prediction loss ($\mathcal{L}_{pred}$):

$$\mathcal{L}(x,c,m,G,z,Y) = \frac{1}{N}\big(\mathcal{L}_{cls}(x,c) + \alpha\mathcal{L}_{reg}(x,m,G) + \beta\mathcal{L}_{pred}(x,z,Y)\big), \quad (1)$$

where N is the number of matches, c is the ground truth class, m is the predicted micro-tube, G is the ground truth micro-tube, z assembles the predictions for the future and the past, and Y is the ground truth of future and past bounding boxes associated with the ground truth micro-tube G. The values of α and β are both set to 1 in all of our experiments: different values might result in better performance.

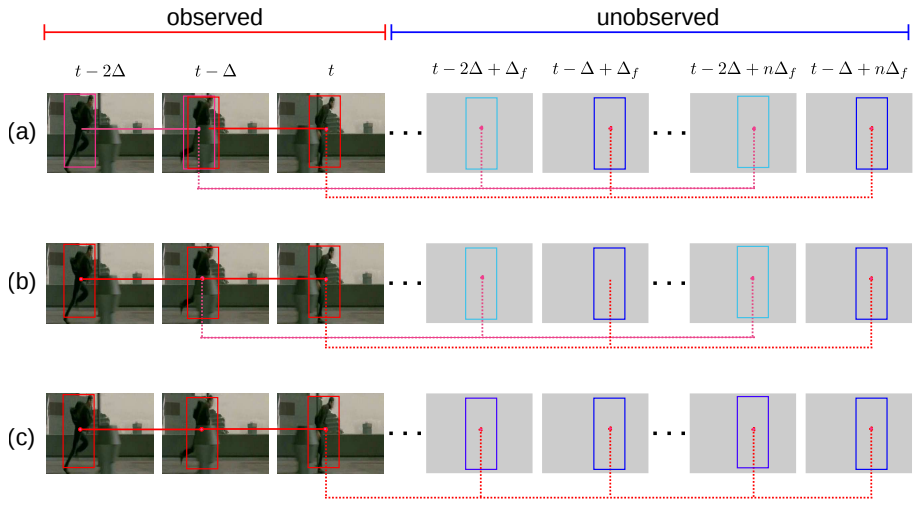

Fig. 5. *Overview of future tube ($\mathcal{T}_c^p$) prediction using the predictions that are linked to micro-tubes. The first row (a) shows two output micro-tubes in light red and red and their corresponding predictions in future in light blue and blue. In row (b) two micro-tubes are linked together, after which they are shown in the same colour (red). By induction on the previous step, in row (c) we show that the predictions associated with two micro-tubes are linked together as well, hence forming one single tube. The observed segment is shown in red, while the predicted segment for the part of the video yet to observe is shown in blue.*

The classification loss $\mathcal{L}_{cls}$ is a softmax cross-entropy loss; a hard negative mining strategy is also employed, as proposed in [22]. The micro-tube loss $\mathcal{L}_{reg}$ is a Smooth L1 loss [27] between the predicted (m) and the ground truth (G) micro-tube. Similarly, the prediction loss $\mathcal{L}_{pred}$ is also a Smooth L1 loss between the predicted boxes (z) and the ground truth boxes (Y). As in [22, 27], we regress the offsets with respect to the coordinates of matched prior box p matched to G for both m and z. We use the same offset encoding scheme as used in [22].

3.4 Tube prediction framework

TPnet is shown in Figure 2 at test time. As in the training setting, it observes only two frames that are Δ apart at any time point t. The outputs of TPnet at any time t are linked to a micro-tube, each micro-tube containing a set of bounding boxes $\{m_t = \{b_t, b_{t+\Delta}\}; z_t = \{b_{t-\Delta_p}, b_{t+\Delta_f}, ..., b_{t+\Delta_f}\}\}$, which are considered as linked together.

As explained in § 3.2, given a set of micro-tubes $\{m_1 ... m_{t-\Delta}\}$ we can construct $\mathcal{T}_c^d$ by online linking [31] of the micro-tubes. As a result, we can use predictions for $t+\Delta_f$ up to $t+n\Delta_f$ to generate the future of $\mathcal{T}_c^d$, thus extending it further into the future as shown in Figure 5. More specifically, as it is indicated in Figure 5(a), a micro tube at $t-2\Delta$ is composed by $n+2$ bounding boxes ($\{b_{t-2\Delta}, b_{t-\Delta}, b_{t-2\Delta+\Delta_f}, ...b_{t-\Delta+n\Delta_f}\}$) linked together. The last micro-tube is generated from $t-\Delta$. In the same fashion, putting together the predictions associated with all the past micro-tubes ($\{m_1 ... m_{t-\Delta}\}$)

yields a set of linked future bounding boxes ($\{b_{t+1}, ..., b_{t+\Delta+\Delta_f}, ..., b_{t-\Delta+n\Delta_f}\}$) for the current action tube $\mathcal{T}_c^d$, thus outputting a part of the desired future $\mathcal{T}_c^p$.

Now, we can generate future tube $\mathcal{T}_c^p$ from the set of linked future bounding boxes ($\{b_{t+1}, ...b_{t-\Delta+\Delta_f}, ...b_{t-\Delta+n\Delta_f}\}$) from $t+1$ to $t-\Delta+n\Delta_f$ and simple linear extrapolation of bounding boxes from $t-\Delta+n\Delta_f$ to T. Linear extrapolation is performed based on the average velocity of the each coordinates from last 5 frames, predictions outside the image coordinate are trimmed to the image edges.

4 Experiments

We test our action tube prediction framework (§ 3) on four challenging problems: i) action localisation (§ 4.1), ii) early action prediction (§ 4.2), iii) online action localisation (§ 4.2), iv) future action tube prediction (§ 4.3) Finally, evidence of real time capability is quantitatively demonstrated in (§ 4.4).

J-HMDB-21. We evaluate our model on the J-HMDB-21 [11] benchmark. J-HMDB-21 [11] is a subset of the HMDB-51 dataset [19] with 21 action categories and 928 videos, each containing a single action instance and trimmed to the action's duration. It contains atomic action which are 20-40 frames long. Although, videos are of short duration (max 40 frames), we consider this dataset because tubes belong to the same class and we think it is a good dataset to start with for action prediction task.

Evaluation metrics. Now, we define the evaluation metrics used in this paper. i) We use a standard mean-average precision metric to evaluate the detection performance when the whole video is observed. ii) Early label prediction task is evaluated by video classification accuracy [32, 31] as early as when only 10% of the video frames are observed.
iii) Online action localisation (§ 4.2) is set up based on the experimental setup of [31], and use mAP (mean average precision) as metric for online action detection i.e. it evaluates present tube ($\mathcal{T}_c^d$) built in online fashion.
iv) The future tube prediction is a new task; we propose to evaluate its performance in two ways. Firstly, we evaluate the quality of the whole tube prediction from the start of the videos to the end as early as when only 10% of the video is observed. The entire tube predicted (by observing only a small portion (%) of the video) is compared against the ground truth tube for the whole video. Based on the detection threshold we can compute mean-average-precision for the complete tubes, we call this metric *completion-mAP* (c-mAP). Secondly, we measure how well the future predicted part of the tube localises. In this measure, we compare the predicted ($\mathcal{T}_c^p$) tube with the corresponding ground truth future tube segment. Given the ground truth and the predicted future tubes, we can compute the mean-average precision for the predicted tubes, we call this metric *prediction-mAP* (p-mAP).

We report the performance of previous three tasks (i.e. task ii to iv) as a function of *Video Observation Percentage*, i.e., the portion (%) of the entire video observed.
Baseline. We modified AMTnet to fuse flow and appearance features 3.2. We treat it as a baseline for all of our tasks. Firstly, we show how feature fusion helps AMTnet in Section 4.1, and compare it with other action detection methods along with our TP-net. Secondly in Section 4.3, we linearly extrapolate the detection from AMTnet to

Table 1. Action localisation results on JHMDB dataset. The table is divided into four parts. The first part lists approaches which takes a single frame as input; the second part presents approaches which takes multiple frames as input; the third part contemplates different fusion strategies of our feature-level fusion (based on AMTnet); lastly, we report the detection performance of our TPnet by ignoring the future and past predictions and only use the detected micro-tubes to produce the final action tubes.

Methods	δ = 0.2	δ = 0.5	δ = 0.75	δ = .5:.95	Acc %
MR-TS Peng *et al.* [25]	74.1	73.1	–	–	–
FasterRCNN Saha *et al.* [30]	72.2	71.5	43.5	40.0	–
OJLA Behl *et al.* [3]*	–	67.3	–	36.1	–
SSD Singh *et al.* [31]*	73.8	72.0	44.5	41.6	–
AMTnet Saha *et al.* [29] rgb-only	57.7	55.3	–	–	–
ACT kalogeiton *et al.* [13]*	74.2	73.7	52.1	44.8	61.7
T-CNN (offline) Hou *et al.* [10]	**78.4**	76.9	–	–	67.2
MR-TS [25] + I3D [5] Gu *et al.* [8]	–	**78.6**	–	–	–
AMTnet-LateFusion*	71.7	71.2	49.7	42.5	65.8
AMTnet-FeatFusion-Concat*	73.1	72.6	59.8	48.3	68.4
AMTnet-FeatFusion-Sum*	73.5	72.8	59.7	48.1	69.6
Ours TPnet$_{053}$*	72.6	72.1	58.0	46.7	67.5
Ours TPnet$_{453}$*	73.8	73.0	59.1	47.3	68.2
Ours TPnet$_{051}$*	74.6	73.1	60.5	49.0	**69.8**
Ours TPnet$_{451}$*	74.8	74.1	**61.3**	**49.1**	68.9

TPnet$_{abc}$ represents our TPnet where $a = \Delta_p$, $b = \Delta_f$ and $c = n$.; * means online methods

construct the future tubes, and use it as a baseline for tube prediction task. **Implementation details.** We train all of our networks with the same set of hyper-parameters to ensure the fair comparison and consistency, including TPnet and AMTnet. We use an initial learning rate of 0.0005, and the learning rate drops by a factor of 10 after $5K$ and $7K$ iterations. All the networks are trained up to $10K$ iterations. We implemented AMTnet using pytorch (https://pytorch.org/). We initialise AMTnet and TPnet models using the pretrained SSD network on J-HMDB-21 dataset on its respective train splits. The SSD network training is initialised using image-net trained VGG network. For, optical flow images, we used optical flow algorithm of Brox *et al.* [4]. Optical flow output is put into a three channel image, two channels are made of flow vector and the third channel is the magnitude of the flow vector.

TPnet$_{abc}$. The training parameters of our TPnet are used to define the name of the setting in which we use our tube prediction network. The network name TPnet$_{abc}$ represents our TPnet where $a = \Delta_p$, $b = \Delta_f$ and $c = n$, if Δ_p is set to 0 it means network doesn't learn to predict the past bounding boxes. In all of our settings, we use $\Delta = 1$.

4.1 Action localisation performance

Table 1 shows the traditional action localisation results for the whole action tube detection in the videos of J-HMBD-21 dataset.

Feature fusion compared to the late fusion scheme in AMTnet shows (Table 1) remarkable improvement, at detection threshold $\delta = 0.75$ the gain with feature level fusion is

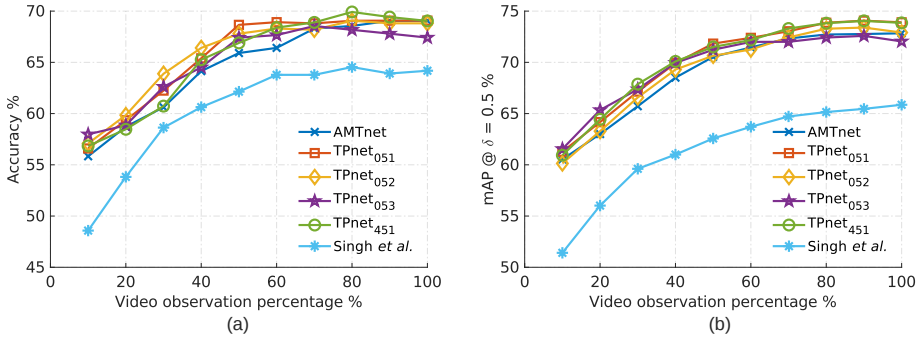

Fig. 6. *Early label prediction results (video-level label prediction accuracy) on J-HMDB-21 dataset in sub-figure (a). Online action detection results (mAP with detection threshold $\delta = 0.5$) on J-HMDB-21 dataset are shown in sub-figure (b). TPnet$_{abc}$ represents our TPnet where $a = \Delta_p$, $b = \Delta_f$ and $c = n$.*

10%, as a result, it is able to surpass the performance of ACT [13], which relies on set of 6 frames as compared to AMTnet which uses only 2 successive frames as input. Looking at the average-mAP ($\delta = 0.5 : 95$), we can see that the fused model improves by almost 8% as compared to single frame SSD model of Singh *et al.* [31]. We can see that concatenation and sum fusion perform almost similar for AMTnet. Sum fusion is little less memory intensive on the GPUs as compared to the concatenation fusion; as a result, we use sum fusion in our TPnet.

TPnet for detection is shown in the last part of the Table 1, where we only use the detected micro-tubes by TPnet to construct the action tubes(§ 3.2). We train TPnet to predict future and past (i.e. when $\Delta_p > 0$) as well as present micro-tubes. We think that predicting bounding boxes for both the past and future video segments acts as a regulariser and helps improving the representation of the whole network. Thus, improving the detection performance (Table 1 TPNet$_{051}$ and TPNet$_{451}$). However, that does not mean adding extra prediction task always help when a network is asked to learn prediction in far future, as is the case in TPNet$_{053}$ and TPNet$_{453}$, we have a drop in the detection performance. We think there might be two possible reasons for this, i) network might starts to focus more on prediction task, and ii) videos in J-HMDB-21 are short and number of training samples decreases drastically ($19K$ for TPNet$_{051}$ and $10K$ for TPNet$_{453}$), because we can not use edge frames of the video in training samples as we need a ground truth bounding box which is 15 frames in the future, as $\Delta_f = 5$ and $n = 3$ for TPNet$_{053}$. However, in Section 4.3, we show that the TPNet$_{053}$ model is the best to predict the future very early.

4.2 Early label prediction and online localisation

Figure 6 (a) & (b) show the early prediction and online detection capabilities of Singh *et al.* [31], AMTnet-Feature Fusion-sum and our TPnet.

Soomro *et al.* [32]'s method also perform early label prediction on J-HMDB-21; however, their performance is deficient, as a result the plot would become skewed (Figure 6(a)), so we omit theirs from the figure. For instance, by observing only the initial

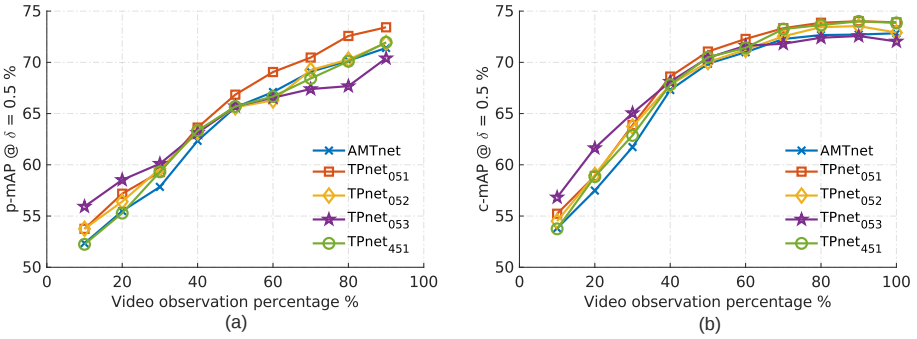

Fig. 7. *Future action tube prediction results (a) (prediction-mAP (p-mAP)) for predicting the tube in unobserved part of the video. Action tube prediction results (b) (completion-mAP (c-mAP)) for predicting video long tubes as early as possible on J-HMDB-21 dataset in sub-figure (b). We use p-mAP (a) and c-mAP (b) with detection threshold $\delta = 0.5$ as evaluation metrics on J-HMDB-21 dataset. TPnet$_{abc}$ represents our TPnet where $a = \Delta_p$, $b = \Delta_f$ and $c = n$.*

10% of the videos in J-HMDB-21, TPnet$_{453}$ able to achieve a prediction accuracy of 58% as compared to 48% by Singh *et al.* [31] and 5% by Soomro *et al.* [32], which is in fact higher than the 43% accuracy achieved by [32] after observing the *entire* video. As more and more video observed, all the methods improve, but TPnet$_{451}$ show the most gain, however, TPnet$_{053}$ observed the least gain from all the TPnet settings shown. Which is in-line with action localisation performance discussed in the previous section 4.1. We can observe the similar trends in online action localisation performance shown in Figure 6(b). To reiterate, TPnet$_{053}$ doesn't get to see the training samples from the end portion of the videos, as it needs a ground truth bounding box from 15 frames ahead. So, the last frame it sees of any training video is $T - 15$, which is almost half the length of the most extended video(40 frames) in J-HMDB-21. This effect magnifies when online localisation performance measured at $\delta = 0.75$, we provide the evidence of it in the supplementary material.

4.3 Future action tube prediction

Our main task of the paper is to predict the future of action tubes. We evaluate it using two newly proposed metrics (*p-mAP* and *c-mAP*) as explained earlier at the start of the experiment section 4. Result are shown in Figure 7 for future tube prediction (Figure 7 (a)) with p-mAP metric and tube completion with c-mAP as metric.

Although, the TPnet$_{053}$ is the worst setting of TPnet model for early label prediction (Fig. 6(a)), online detection(Fig. 6(b)) and action tube detection (Table 1), but as it predicts furthest in the future (i.e. 15 frame away from the present time), it is the best model for early future tube prediction (Fig. 7(a)). However, it does not observe as much appreciation in performance as other settings as more and more frames are seen, owing to the reduction in the number of training samples. On the other hand, TPnet$_{451}$

observed large improvement as compared to TPnet$_{051}$ as more and more portion of the video is observed for tube completion task (Fig.7(b)), which strengthen our arguement that predicting not only the future but also the past is useful to achieve more regularised predictions.

Comparision with the baseline. As explained above, we use AMTnet as a baseline, and its results can be seen in all the plots and the Table. We can observe that our TPnet performs better than AMTnet in almost all the cases, especially in our desired task of early future prediction (Fig 7(a)) TPnet$_{043}$ shows almost 4% improvement in p-mAP (at 10% video observation) over AMTnet.

Discussion. Predicting further into the future is essential to produce any meaningful predictions (seen in TPnet$_{053}$), but at the same time, predicting past is helpful to improve overall tube completion performance. One of the reasons for such behaviour could be that J-HMDB-21 tubes are short (max 40 frames long). We think training samples for a combination of TPnet$_{053}$ and TPnet$_{451}$, i.e. TPnet$_{453}$ are chosen uniformly over the whole video while taking care of absence of ground truth in the loss function could give us better of both settings. We show the result of TPnet$_{453}$ in current training setting in supplementary material. The idea of regularising based on past prediction is similar to the one used by Ma *et al.* [23].

4.4 Test Time Detection Speed

Singh *et al.* [31] showcase their method's online and real-time capabilities. Here we use their online tube generation method for our tube prediction framework to inherit those properties. The only question mark is TPnet's forward pass speed. We thus measured the average time taken for a forward pass for a batch size of 1 as compared to 8 by [31]. A single forward pass takes 46.8 milliseconds to process one text example, showing that it can be run in almost real-time at 21fps with two streams on a single 1080Ti GPU. One can improve speed even further by testing TPnet with Δ equal to 2 or 4 and obtain a speed improvement of $2\times$ or $2\times$. However, use of dense optical flow [4], which is slow, but as in [31], we can always switch to real-time optical [18] with small drop in performance.

5 Conclusions

We presented TPnet, a deep learning framework for future action tube prediction in videos which, unlike previous online tube detection methods [31, 32], generates future of action tubes as early as when 10% of the video is observed. It can cope with the future uncertainty better than the baseline methods while remaining state-of-the-art in action detection task. Hence, we provide a scalable platform to push the boundaries of action tube prediction research; it is implicitly scalable to multiple action tube instances in the video as future prediction is made for each action tube separately. We plan to scale TPnet for action prediction in temporally untrimmed videos in the future.

References

1. Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S.: Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 961–971 (2016)
2. Aliakbarian, M.S., Saleh, F.S., Salzmann, M., Fernando, B., Petersson, L., Andersson, L.: Encouraging lstms to anticipate actions very early. In: IEEE International Conference on Computer Vision (ICCV). vol. 1 (2017)
3. Behl, H.S., Sapienza, M., Singh, G., Saha, S., Cuzzolin, F., Torr, P.H.: Incremental tube construction for human action detection. arXiv preprint arXiv:1704.01358 (2017)
4. Brox, T., Bruhn, A., Papenberg, N., Weickert, J.: High accuracy optical flow estimation based on a theory for warping (2004)
5. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4724–4733. IEEE (2017)
6. De Geest, R., Gavves, E., Ghodrati, A., Li, Z., Snoek, C., Tuytelaars, T.: Online action detection. arXiv preprint arXiv:1604.06506 (2016)
7. Gkioxari, G., Malik, J.: Finding action tubes. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition (2015)
8. Gu, C., Sun, C., Vijayanarasimhan, S., Pantofaru, C., Ross, D.A., Toderici, G., Li, Y., Ricco, S., Sukthankar, R., Schmid, C., et al.: Ava: A video dataset of spatio-temporally localized atomic visual actions. arXiv preprint arXiv:1705.08421 (2017)
9. Hoai, M., De la Torre, F.: Max-margin early event detectors. International Journal of Computer Vision **107**(2), 191–202 (2014)
10. Hou, R., Chen, C., Shah, M.: Tube convolutional neural network (t-cnn) for action detection in videos. In: IEEE Int. Conf. on Computer Vision (2017)
11. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M.: Towards understanding action recognition (2013)
12. Jiang, Y., Saxena, A.: Modeling high-dimensional humans for activity anticipation using gaussian process latent crfs. In: Robotics: Science and Systems, RSS (2014)
13. Kalogeiton, V., Weinzaepfel, P., Ferrari, V., Schmid, C.: Action tubelet detector for spatio-temporal action localization. In: IEEE Int. Conf. on Computer Vision (2017)
14. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., et al.: The kinetics human action video dataset. arXiv preprint arXiv:1705.06950 (2017)
15. Kitani, K.M., Ziebart, B.D., Bagnell, J.A., Hebert, M.: Activity forecasting. In: European Conference on Computer Vision. pp. 201–214. Springer (2012)
16. Kong, Y., Tao, Z., Fu, Y.: Deep sequential context networks for action prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1473–1481 (2017)
17. Koppula, H.S., Gupta, R., Saxena, A.: Learning human activities and object affordances from rgb-d videos. The International Journal of Robotics Research **32**(8), 951–970 (2013)
18. Kroeger, T., Timofte, R., Dai, D., Van Gool, L.: Fast optical flow using dense inverse search. arXiv preprint arXiv:1603.03590 (2016)
19. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: Hmdb: a large video database for human motion recognition. In: Computer Vision (ICCV), 2011 IEEE International Conference on. pp. 2556–2563. IEEE (2011)
20. Lan, T., Chen, T.C., Savarese, S.: A hierarchical representation for future action prediction. In: Computer Vision–ECCV 2014. pp. 689–704. Springer (2014)

21. Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M.: Desire: Distant future prediction in dynamic scenes with interacting agents. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 336–345 (2017)
22. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. arXiv preprint arXiv:1512.02325 (2015)
23. Ma, S., Sigal, L., Sclaroff, S.: Learning activity progression in lstms for activity detection and early detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1942–1950 (2016)
24. Nazerfard, E., Cook, D.J.: Using bayesian networks for daily activity prediction. In: AAAI Workshop: Plan, Activity, and Intent Recognition (2013)
25. Peng, X., Schmid, C.: Multi-region two-stream R-CNN for action detection. In: ECCV 2016 - European Conference on Computer Vision. Amsterdam, Netherlands (Oct 2016), https://hal.inria.fr/hal-01349107
26. Redmon, J., Farhadi, A.: Yolo9000: Better, faster, stronger. arXiv preprint arXiv:1612.08242 (2016)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. pp. 91–99 (2015)
28. Ryoo, M.S.: Human activity prediction: Early recognition of ongoing activities from streaming videos. In: IEEE Int. Conf. on Computer Vision. pp. 1036–1043. IEEE (2011)
29. Saha, S., Singh, G., Cuzzolin, F.: Amtnet: Action-micro-tube regression by end-to-end trainable deep architecture. In: IEEE Int. Conf. on Computer Vision (2017)
30. Saha, S., Singh, G., Sapienza, M., Torr, P.H.S., Cuzzolin, F.: Deep learning for detecting multiple space-time action tubes in videos. In: British Machine Vision Conference (2016)
31. Singh, G., Saha, S., Sapienza, M., Torr, P., Cuzzolin, F.: Online real-time multiple spatiotemporal action localisation and prediction. In: IEEE Int. Conf. on Computer Vision (2017)
32. Soomro, K., Idrees, H., Shah, M.: Predicting the where and what of actors and actions through online action localization (2016)
33. Tahmida Mahmud, M.H., Roy-Chowdhury, A.K.: Joint prediction of activity labels and starting times in untrimmed videos. In: IEEE Int. Conf. on Computer Vision. vol. 1 (2017)
34. Vondrick, C., Pirsiavash, H., Torralba, A.: Anticipating the future by watching unlabeled video. arXiv preprint arXiv:1504.08023 (2015)
35. Weinzaepfel, P., Harchaoui, Z., Schmid, C.: Learning to track for spatio-temporal action localization. In: IEEE Int. Conf. on Computer Vision and Pattern Recognition (June 2015)
36. Weinzaepfel, P., Martin, X., Schmid, C.: Human action localization with sparse spatial supervision. arXiv preprint arXiv:1605.05197 (2016)
37. Weinzaepfel, P., Martin, X., Schmid, C.: Towards weakly-supervised action localization. arXiv preprint arXiv:1605.05197 (2016)
38. Yang, Z., Gao, J., Nevatia, R.: Spatio-temporal action detection with cascade proposal and location anticipation. In: BMVC (2017)
39. Yeung, S., Russakovsky, O., Jin, N., Andriluka, M., Mori, G., Fei-Fei, L.: Every moment counts: Dense detailed labeling of actions in complex videos. arXiv preprint arXiv:1507.05738 (2015)
40. Yeung, S., Russakovsky, O., Mori, G., Fei-Fei, L.: End-to-end learning of action detection from frame glimpses in videos. CVPR (2016)
41. Zolfaghari, M., Oliveira, G.L., Sedaghat, N., Brox, T.: Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In: IEEE Int. Conf. on Computer Vision. pp. 2923–2932. IEEE (2017)
42. Zunino, A., Cavazza, J., Koul, A., Cavallo, A., Becchio, C., Murino, V.: Predicting human intentions from motion cues only: A 2d+ 3d fusion approach. In: Proceedings of the 2017 ACM on Multimedia Conference. pp. 591–599. ACM (2017)