View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Oxford Brookes University: RADAR

Except for this watermark, it is identical to the version available on IEEE Xplore.

# Online Real-time Multiple Spatiotemporal Action Localisation and Prediction

Gurkirt Singh[1] Suman Saha[1] Michael Sapienza[2*] Philip Torr[2] Fabio Cuzzolin[1]
[1]Oxford Brookes University [2]University of Oxford
{gurkirt.singh-2015,suman.saha-2014,fabio.cuzzolin}@brookes.ac.uk
m.sapienza@samsung.com, philip.torr@eng.ox.ac.uk

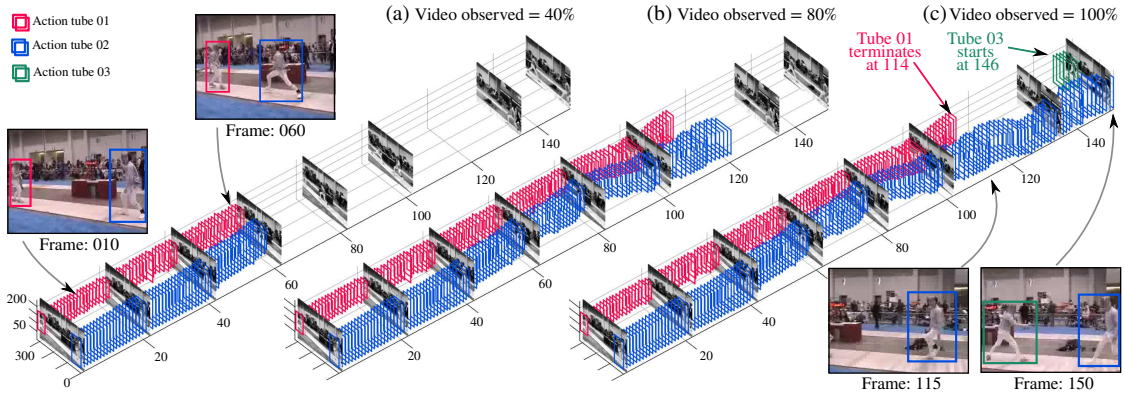


Figure 1: Online spatio-temporal action localisation in a test 'fencing' video from UCF-101-24 [43]. **(a)** to **(c)**: A 3D volumetric view of the video showing detection boxes and selected frames. At any given time, a certain portion (%) of the entire video is observed by the system, and the detection boxes are linked up to incrementally build space-time action tubes. Note that the proposed method is able to detect multiple co-occurring action instances (3 tubes shown here).

## Abstract

*We present a deep-learning framework for real-time multiple spatio-temporal (S/T) action localisation and classification. Current state-of-the-art approaches work offline, and are too slow to be useful in real-world settings. To overcome their limitations we introduce two major developments. Firstly, we adopt real-time SSD (Single Shot Multi-Box Detector) CNNs to regress and classify detection boxes in each video frame potentially containing an action of interest. Secondly, we design an original and efficient online algorithm to incrementally construct and label 'action tubes' from the SSD frame level detections. As a result, our system is not only capable of performing S/T detection in real time, but can also perform early action prediction in an online fashion. We achieve new state-of-the-art results in both S/T action localisation and early action prediction on the challenging UCF101-24 and J-HMDB-21 benchmarks, even when compared to the top offline competitors. To the best of our knowledge, ours is the first real-time (up to 40fps) system able to perform online S/T action localisation on the untrimmed videos of UCF101-24.*

## 1. Introduction

Spatio-temporal human action localisation [53, 33, 28] in videos is a challenging problem that is made even harder if detection is to be performed in an online setting and at real-time speed. Despite the performance of state-of-the-art S/T action detection systems [33, 28] being far from real time, current systems also assume that the entire video (taken as a 3D block of pixels) is available ahead of time in order to detect action instances. Here, an action instance is made up of a sequence of detection boxes linked in time to form an 'action tube' [7, 53]. For such a detector to be applicable to real-world scenarios such as video surveillance and human-robot interaction, video frames need to be processed in real time. Moreover, the action detection system needs to construct action tubes in an incremental and online fashion, as each new frame is captured.

With the rise of Convolutional Neural Networks (CNNs), impressive progress has been made in image classification [15] and object detection [6], motivating researchers to apply CNNs to action classification and localisation. Although the resulting CNN-based state-of-the-art S/T action detectors [33, 7, 53, 28] have achieved remarkable results, these methods are computationally expensive and their detection accuracy is still below what is needed for real-world deployment. Most of these approaches [7, 53] are based

*M. Sapienza performed this research at the University of Oxford, and is currently with the Think Tank Team, Samsung Research America, CA.

on unsupervised region proposal algorithms [48, 61] and on an expensive multi-stage training strategy mutuated from object detection [6]. For example, Gkioxari *et al*. [7] and Weinzaepfel *et al*. [53] both separately train a pair of (motion and appearance) CNNs and a battery of one-vs-rest Support Vector Machines (SVMs). This limits detection accuracy as each module is trained independently, leading to sub-optimal solutions.

The most recent efforts by Saha *et al*. [33] and Peng *et al*. [28] use a supervised region proposal generation approach [30], and eliminate the need for multi-stage training [6] by using a single end-to-end trainable CNN for action classification and bounding box regression. Although [33, 28] exhibit the best spatio-temporal action localisation accuracies to date, test time detection involves the use of computationally expensive optical flow [1], and remains a two-step region proposal network (RPN) [30] and RCNN [30] process, limiting real-time deployment. Also, [33, 28] both employ offline tube generation methods which process the entire video in two passes: one to link detection boxes into tubes which stretch from start to end of the video, and one to temporally trim and label the video-long constructed tubes.

In this work, we propose an online framework, outlined in Figure 2, which overcomes all the above limitations. The pipeline takes advantage of the more recent SSD (Single Shot MultiBox Detector) object detector [22] to address issues with accuracy and speed at frame level. This is possible as SSD eliminates the region proposal generation step and is single-stage, end-to-end trainable.

To leverage the performance of SSD, we design a novel single pass online tube building method which leads to both superior accuracy (compared to [53, 33, 28]), especially at realistic detection precision, and real-time detection speed. Unlike previous tube-generation approaches [7, 33, 28, 53], our algorithm works in an online fashion as tubes are updated frame by frame, together with their overall action-specific scores and labels. As soon as non-real-time optical flow [1] is replaced by the less accurate (but real-time) optical flow algorithm [16], the resulting system performs in real time (28fps), with just a little performance degradation, an essential feature for real-world applications.

The incremental nature of our system makes it possible to accurately foresee the class label of an entire test video and localise action instances within it by just observing a small fraction of frames (*early action prediction and localisation*). Such a system has been recently proposed by Soomro *et al*. [42], who showed that both action prediction and online localisation performance improve over time as more and more video frames become available. Using [42] as a baseline, we report here new state-of-the-art results on the temporally trimmed J-HMDB-21 videos. Furthermore, compared to [42], we are able to demonstrate action prediction and localisation capabilities from partially observed *untrimmed* streaming videos on the challenging UCF101-24 dataset, while retaining real-time detection speeds.

**Contributions.** In summary, we present a holistic framework for the real-time, online spatial and temporal localisation of multiple action instances in videos which:

1. incorporates the newest SSD [22] neural architecture to predict frame-level detection boxes and the associated action class-specific confidence scores, in a single-stage regression and classification approach (§ 3.2);
2. devises an original, greedy algorithm capable of generating multiple action tubes incrementally (§ 3.4);
3. provides early action class label predictions and online spatio-temporal localisation results (Fig. 1) from partially observed action instances in untrimmed videos;
4. functions in real-time, while outperforming the previous (offline) state of the art on the untrimmed videos of UCF101-24 dataset.

To the best of our knowledge, our framework is the first with a demonstrated ability to perform online spatial and temporal action localisation. An extensive empirical evaluation demonstrates that our approach:

- significantly outperforms current offline methods, especially on realistic detection thresholds of $0.5$ or greater;
- is capable of superior early action prediction performance compared to the state of the art [42];
- achieves a real-time detection speed (upto 40fps), that is 5 to 6 times faster than previous works (§ 4.4).

Our code is available online at `https://github.com/gurkirt/realtime-action-detection`.

## 2. Related work

Deep learning architectures have been increasingly applied of late to action classification [13, 14, 37, 47], spatial [7], temporal [36] and spatio-temporal [53] action localisation, and event detection [55].

*Spatial action localisation* is typically addressed using segmentation [23, 41, 11] or region proposal and actionness [7, 52] -based approaches. Gkioxari and Malik [7], in particular, have built on [6] and [37] to tackle spatial action localisation in temporally trimmed videos only, using Selective-Search region proposals, fine-tuned CNN features and a set of one-vs-rest SVMs. These approaches are restricted to trimmed videos.

*Temporal action detection* is mostly tackled using expensive sliding window [20, 5, 46, 27, 51] approaches. Recently, deep learning-based methods have led to significant advances. For instance, Shou *et al*. [36] have employed 3D CNNs [13, 47] to address temporal action detection in long videos. LSTMs are also increasingly being used [56, 3, 38, 57] to address the problem. Dynamic programming has been employed to solve the problem efficiently [18, 4, 40]. Some of the above works [56, 3, 4, 56]

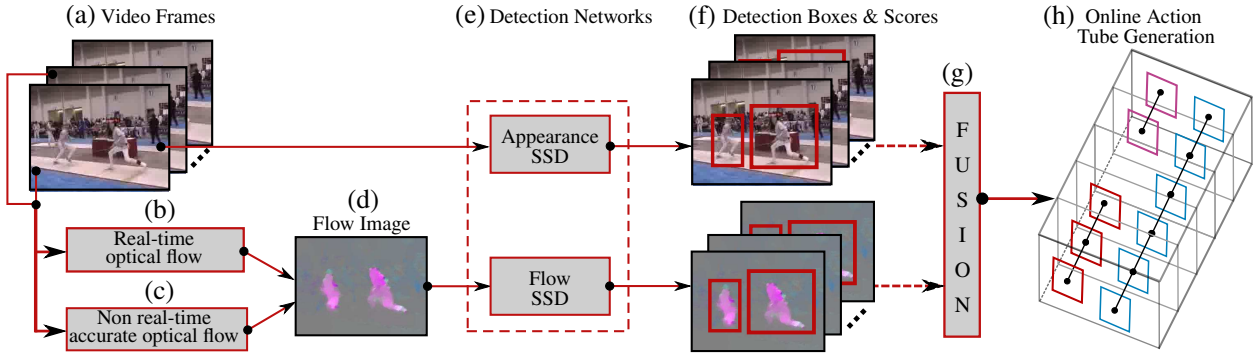


Figure 2. *At test time, the input to the framework is a sequence of RGB video frames* ***(a)****. A real-time optical flow (OF) algorithm* ***(b)*** *[16] takes the consecutive RGB frames as input to produce flow images* ***(d)****. As an option,* ***(c)*** *a more accurate optical flow algorithm [1] can be used (although not in real time).* ***(e)*** *RGB and OF images are fed to two separate SSD detection [22] networks (§ 3.2).* ***(f)*** *Each network outputs a set of detection boxes along with their class-specific confidence scores (§ 3.2).* ***(g)*** *Appearance and flow detections are fused (§ 3.3). Finally* ***(h)****, multiple action tubes are built up in an online fashion by associating current detections with partial tubes (§ 3.4).*

can perform action detection in an online fashion. However, unlike our framework, all these methods only address temporal, as opposed to spatial and temporal, action detection.

*Spatio-temporal action localisation* can be approached in a supervised [28, 33], semi-supervised [49, 53], or weakly supervised [34, 45] manner. Inspired by Oneata *et al*. [27] and Jain *et al*. [10], Gemert *et al*. [49] use unsupervised clustering to generate a small set of bounding box-like spatio-temporal action proposals. As their method is based on dense-trajectory features [50], it fails to detect actions characterised by small motions [49]. Weinzaepfel *et al*.'s work [53] performs both temporal and spatial detections by coupling frame-level EdgeBoxes [61] region proposals with a tracking-by-detection framework. However, temporal trimming is still achieved via a multi-scale sliding window over each track, making the approach inefficient for longer video sequences. More recently, Saha *et al*. [33] and Peng *et al*. [28] have made use of supervised region proposal networks (RPNs) [30] to generate region proposal for actions on frame level, and solved the S/T association problem via 2 recursive passes over frame level detections for the entire video by dynamic programming. Using a non real-time and 2-pass tube generation approach, however, makes their methods offline and inefficient. In opposition, our framework employs a real-time OF algorithm [16] and a single shot SSD detector [22] to build multiple action tubes in a fully incremental way, and in real time.

*Real-time methods*. Relatively few efforts have been directed at simultaneous real time action detection and classification. Zhang *et al*. [60], for example, accelerate the two-stream CNN architecture of [37], performing action classification at 400 frames per second. Unlike our method, however, theirs cannot perform spatial localisation. Yu *et al*. [59] evaluate their real-time continuous action classification approach on the relatively simpler KTH [35] and UT-interaction [32] datasets. To the best of our knowledge, this is the first work to address real-time action localisation.

*Online action prediction*. Early, online action prediction has been studied using dynamic bag of words [31], structured SVMs [9], hierarchical representations [19], LSTMs and Fisher vectors [3]. Once again, unlike our framework, these approaches [31, 9, 19] do not perform online action localisation. Soomro *et al*. [42] recently proposed an online method which can predict an action's label and location by observing a relatively smaller portion of the entire video sequence. However, [42] only works on temporally trimmed videos and not in real-time, due to expensive segmentation.

## 3. Methodology

As outlined in Fig. 2, our approach exploits an integrated detection network [22] (§ 3.2-Fig. 2e) to predict detection boxes and class-specific confidence scores for appearance and flow (§ 3.1) video frames independently. One of two alternative fusion strategies (§ 3.3-Fig. 2g) is then applied. Finally, action tubes are built incrementally in an online fashion and in real time, using a new efficient action tube generation algorithm (§ 3.4-Fig. 2h), which can be applied to early action prediction (§ 3.5).

### 3.1. Optical flow computation

The input to our framework is a sequence of RGB images. As in prior work in action localisation [33, 7, 53], we use a two-stream CNN approach [37] in which optical flow and appearance are processed in two parallel, distinct streams. As our aim is to perform action localisation in real-time, we employ real-time optical flow (Fig. 2b) [16] to generate the flow images (Fig. 2d). As an option, one can compute optical flow more accurately (Fig. 2c), using Brox *et al*.'s [1] method. We thus train two different networks for the two OF algorithms, while at test time only one network is used depending on whether the focus is on speed rather than accuracy. Following the transfer learning approach on

motion vectors of [60], we first train the SSD network on accurate flow results, to later transfer the learned weights to initialise those of the real time OF network. Performance would degrade whenever transfer learning was not used.

### 3.2. Integrated detection network

We use a single-stage convolutional neural network (Fig. 2e) for bounding box prediction and classification, which follows an end-to-end trainable architecture proposed in [22]. The architecture unifies a number of functionalities in single CNN which are, in other action and object detectors, performed by separate components [7, 53, 30, 33], namely: (i) region proposal generation, (ii) bounding box prediction and (iii) estimation of class-specific confidence scores for the predicted boxes. This allows for relatively faster training and higher test time detection speeds.

**Detection network design and training.** For our integrated detection network we adopt the network design and architecture of the SSD [22] object detector, with an input image size of $300 \times 300$. We do not use the $512 \times 512$ SSD architecture [22], as detection speed is much slower [22]. As in [22], we also use an ImageNet pretrained VGG16 network provided by [22] (https://gist.github.com/weiliu89/2ed6e13bfd5b57cf81d6). We adopt the training procedure described by [22] along with their publicly available code for network training (https://github.com/weiliu89/caffe/tree/ssd). We use a learning rate of $0.0005$ for the appearance stream and of $0.0001$ for the flow stream on UCF101-24, whereas that for JHMDB is set to $0.0001$ for both appearance and flow. All implementation details are in the supplementary material.

### 3.3. Fusion of appearance and flow cues

The detection boxes generated by the appearance and flow detection networks (Fig. 2f) need to be merged to improve robustness and accuracy (Fig. 2g). We conducted experiments using two distinct fusion strategies.

**Boost-fusion**. Here we follow the approach in [33], with a minor modification. Firstly, we perform L-1 normalisation on the detection boxes' scores after fusion. Secondly, we retain any flow detection boxes for which an associated appearance based box was not found, as we found that discarding the boxes lowers the overall recall.

**Fusion by taking the union-set.** A different, effective fusion strategy consists in retaining the union $\{b_i^a\} \cup \{b_j^f\}$ of the two sets of appearance $\{b_i^a\}$ and flow $\{b_j^f\}$ detection boxes, respectively. The rationale is that in UCF-101, for instance, several action classes (such as 'Biking', 'IceDancing', or 'SalsaSpin') have concurrent action instances in the majority of video clips: an increased number of detection boxes may so help to localise concurrent action instances.

### 3.4. Online action tube generation

Given a set of detections at time $t = 1..T$, for each given action class $c$, we seek the sets of consecutive detections (or *action tubes*) $\mathcal{T}_c = \{b_{t_s}, , b_{t_e}\}$ which, among all possible such collections, are more likely to constitute an action instance. This is done separately for each class, so that results for class $c$ do not influence those for other classes. We allow the number of tubes $n_c(t)$ to vary in time, within the constraint given by the number of available input detections. We allow action tubes to start or end at any given time. Finally, we require: (i) consecutive detections part of an action tube to have spatial overlap above a threshold $\lambda$; (ii) each class-specific detection to belong to a single action tube; (iii) the online update of the tubes' temporal labels. Previous approaches to the problem [7, 33] constrain tubes to span the entire video duration. In both [33] and [28], in addition, action paths are temporally trimmed to proper action tubes using a second pass of dynamic programming.

In opposition, we propose a simple but efficient online action tube generation algorithm which incrementally (frame by frame) builds multiple action tubes for each action class in parallel. Action tubes are treated as 'tracklets', as in multi-target tracking approaches [26]. We propose a greedy algorithm (3.4.1) similar to [25, 39] for associating detection boxes in the upcoming frame with the current set of (partial) action tubes. Concurrently, each tube is temporally trimmed in an online temporal labelling (3.4.2) setting.

#### 3.4.1 A novel greedy algorithm

The input to the algorithm is the fused frame-level detection boxes with their class specific scores (Sec. 3.3). At each time step $t$, the top $n$ class-specific detection boxes $\{b_c\}$ are selected by applying non-maximum suppression on a per-class basis. At the first frame of the video, $n_c(1) = n$ action tubes per class $c$ are initialised using the $n$ detection boxes at $t = 1$. The algorithm incrementally grows the tubes over time by adding one box at a time. The number of tubes $n_c(t)$ varies with time, as new tubes are added and/or old tubes are terminated.

At each time step, we sort the existing partial tubes so that the best tube can potentially match the best box from the set of detection boxes in the next frame $t$. Also, for each partial tube $\mathcal{T}_c^i$ at time $t-1$, we restrict the potential matches to detection boxes at time $t$ whose IoU (Intersection over Union) with the last box of $\mathcal{T}_c^i$ is above a threshold $\lambda$. In this way tubes cannot simply drift off, and they can be terminated if no matches are found for $k$ consecutive frames. Finally, each newly updated tube is temporally trimmed by performing a binary labelling using an online Viterbi algorithm. This is described in detail in Sec. 3.4.2.

Summarising, action tubes are constructed by applying the following 7 steps to every new frame at time $t$:

1. Execute steps 2 to 7 for each class $c$.
2. Sort the action tubes generated up to time $t-1$ in decreasing order, based on the mean of the class scores of the tube's member detection boxes.
3. **LOOP START:** $i = 1$ to $n_c(t-1)$ - traverse the sorted tube list.
4. Pick tube $\mathcal{T}_c^i$ from the list and find a matching box for it among the $n$ class-specific detection boxes $\{b_c^j, j = 1, ..., n\}$ at frame $t$ based on the following conditions:
   (a) for all $j = 1, ..., n$, if the IoU between the last box of tube $\mathcal{T}_c^i$ and the detection box $b_c^j$ is greater than $\lambda$, then add it to a potential match list $\mathcal{B}^i$;
   (b) if the list of potential matches is not empty, $\mathcal{B}^i \neq \emptyset$, select the box $b_c^{max}$ from $\mathcal{B}^i$ with the highest score for class $c$ as the match, and remove it from the set of available detection boxes at time $t$;
   (c) if $\mathcal{B}^i = \emptyset$, retain the tube anyway, without adding any new detection box, unless more than $k$ frames have passed with no match found for it.
5. Update the temporal labelling for tube $\mathcal{T}_c^i$ using the score $s(b_c^{max})$ of the selected box $b_c^{max}$ (see § 3.4.2).
6. **LOOP END**
7. If any detection box is left unassigned, start a new tube at time $t$ using this box.

In all our experiments, we set $\lambda = 0.1$, $n = 10$, and $k = 5$.

#### 3.4.2 Temporal labelling

Although $n$ action tubes per class are initialised at frame $t = 1$, we want all action specific tubes to be allowed to start and end at any arbitrary time points $t_s$ and $t_e$. The online temporal relabelling step 5. in the above algorithm is designed to take care of this.

Similar to [33, 4], each detection box $b_r$, $r = 1, ..., T$ in a tube $\mathcal{T}_c$, where $T$ is the current duration of the tube and $r$ is its temporal position within it, is assigned a binary label $l_r \in \{c, 0\}$, where $c$ is the tube's class label and 0 denotes the background class. The temporal trimming of an action tube thus reduces to finding an optimal binary labelling $\mathbf{l} = \{l_1, ..., l_T\}$ for all the constituting bounding boxes. This can be achieved by maximising for each tube $\mathcal{T}_c$ the energy:

$$E(\mathbf{l}) = \sum_{r=1}^{T} s_{l_r}(b_r) - \alpha_l \sum_{r=2}^{T} \psi_l(l_r, l_{r-1}), \qquad (1)$$

where $s_{l_r}(b_r) = s_c(b_r)$ if $l_r = c$, $1 - s_c(b_r)$ if $l_r = 0$, $\alpha_l$ is a scalar parameter, and the pairwise potential $\psi_l$ is defined as: $\psi_l(l_r, l_{r-1}) = 0$ if $l_r = l_{r-1}$, $\psi_l(l_r, l_{r-1}) = \alpha_c$ otherwise.

**Online Viterbi.** The maximisation problem (1) can be solved by Viterbi dynamic programming [33]. An optimal labelling $\hat{\mathbf{l}}$ for a tube $\mathcal{T}_c$ can be generated by a Viterbi backward pass at any arbitrary time instant $t$ in linear time. We keep track of past box-to-tube associations from the start of the tube up to $t-1$, which eliminates the requirement of an entire backward pass at each time step. This makes temporal labelling very efficient, and suitable to be used in an online fashion. This can be further optimised for much longer videos by finding the coalescence point [44]. As stated in step 5. above, the temporal labelling of each tube is updated at each time step whenever a new box is added. In the supplementary material, we present a pseudocode of our online action tube generation algorithm.

### 3.5. Early action prediction

As for each test video multiple tubes are built incrementally at each time step $t$ (§3.4), we can predict at any time instant the label of the whole video as the label of the current highest-scoring tube, where the score of a tube is defined as the mean of the tube boxes' individual detection scores: $\hat{c}(t) = \arg\max_c \left( \max_{\mathcal{T}_c} \frac{1}{T} \sum_{r=1}^{T} s(b_r) \right)$.

## 4. Experiments

We test our online framework (§ 3) on two separate challenging problems: i) early action prediction (§ 4.1), ii) online spatio-temporal action localisation (§ 4.2), including a comparison to offline action detection methods. Evidence of real time capability is provided in (§ 4.4).

In all settings we generate results by running our framework in five different 'modes': 1) *Appearance (A)* – only RGB video frames are processed by a single SSD; 2) *Real-time flow (RTF)* – optical flow images are computed in real-time [16] and fed to a single SSD; 3) *A+RTF*: both RGB and real-time optical flow images are processed by a SSD in two separate streams; 4) *Accurate flow (AF)* optical flow images are computed as in [1], and 5) *A+AF*: both RGB and non real-time optical flow frames [1] are used.
Modes 1), 2) and 3) run in real-time whereas modes 4) and 5)'s performances are non real-time (while still working incrementally), due to the relatively higher computation time needed to generate accurate optical flow.

**Datasets.** We evaluate our model on the UCF-101-24 [43] and J-HMDB-21 [12] benchmarks. **UCF101-24** is a subset of UCF101 [43], one of the largest and most diversified and challenging action datasets. Although each video only contains a single action category, it may contain multiple action instances (upto 12 in a video) of the same action class, with different spatial and temporal boundaries. A subset of 24 classes out of 101 comes with spatio-temporal localisation annotation, released as bounding box annotations of humans with THUMOS-2013 challenge[1]. On average there are 1.5 action instances per video, each action instance covering 70% of the duration of the video. For some classes, instances avergae duration can be as low as 30%. As in previous spatio-temporal action detection works

[1] http://crcv.ucf.edu/ICCV13-Action-Workshop/download.html

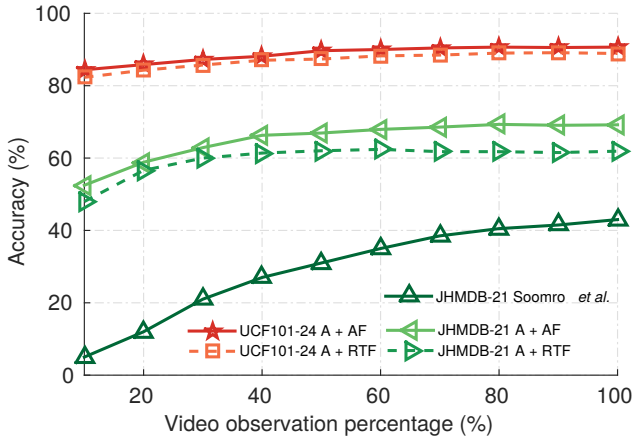


Figure 3. *Early action label prediction results (accuracy %) on the UCF101-24 and J-HMDB-21 datasets.*

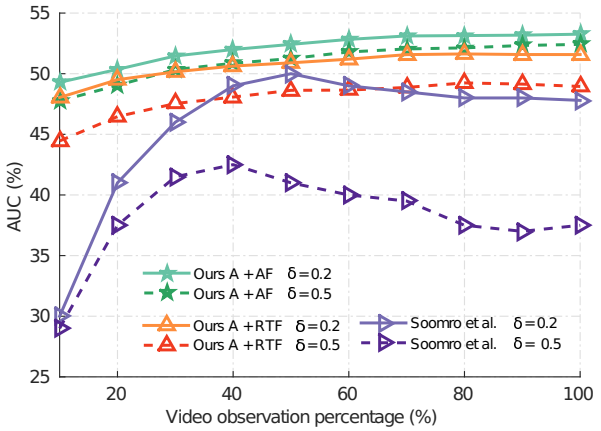


Figure 4. *Online action localisation results using the AUC (%) metric on J-HMDB-21, at IoU thresholds of* $\delta = 0.2, 0.5$.

[33, 58, 28, 53], we test our method on split 1. **J-HMDB-21** [12] is a subset of the HMDB-51 dataset [17] with 21 action categories and 928 videos, each containing a single action instance and trimmed to the action's duration.

Note that the THUMOS [8] and ActivityNet [2] datasets are not suitable for spatiotemporal localisation, as they lack bounding box annotation.

**Evaluation metrics.** For the early action label prediction (§ 4.1) and the online action localisation (§ 4.2) tasks we follow the experimental setup of [42], and use the traditional localisation metrics AUC (area under the curve) and mAP (mean average precision). We report performance as a function of *Video Observation Percentage*, i.e., with respect to the portion (%) of the entire video observed before predicting action label and location. We also report a performance comparison to offline methods [33, 58, 28, 53] using the protocol by Weinzaepfel *et al*. [53].

## 4.1. Early action label prediction

Although action tubes are computed by our framework frame by frame, we sample them at 10 different time 'check-points' along each video, starting at 10% of the total number of video frames and with a step size of 10%. We use the union-set and boost fusion strategies (§ 3.3) for UCF101-24 and J-HMDB-21, respectively. Fig. 3 compares the early action prediction accuracy of our approach with that of [42], as a function of the portion (%) of video observed. Our method clearly demonstrates superior performance, as it is able to predict the actual video label by observing a very small portion of the entire video at a very initial stage. For instance, by observing only the initial 10% of the videos in J-HMDB-21, we are able to achieve a prediction accuracy of 48% as compared to 5% by Soomro *et al*. [42], which is in fact higher than the 43% accuracy achieved by [42] after observing the *entire* video. We do not run comparisons with the early action prediction work by Ma *et al*. [24] for they only show results on ActivityNet [2], as dataset which has only temporal annotations. The early prediction capability of our approach is a subproduct of its being online, as in [42]: thus, we only compare ourselves with Soomro *et al*. [42] re early action prediction results.

Compared to [42] we take one step further, and perform early label prediction on the untrimmed videos of UCF101-24 as well (see Fig. 3). It can be noted that our method performs much better on UCF101-24 than on J-HMBD-21 at the prediction task. This relatively higher performance may be attributed to the larger number of training examples, subject to more modes of variations, present in UCF101-24, which improves the generalisation ability of the model and prevents it from overfitting. Interestingly, we can observe that the performances of the real-time (A + RTF) and non real-time (A + AF) modalities are quite similar, which suggests that accurate optical flow might be not so crucial for action classification on UCF101-24 dataset.

## 4.2. Online spatio-temporal action localisation

### 4.2.1 Performance over time

Our action tubes are built incrementally and carry associated labels and scores at each time step. At any arbitrary time $t$, we can thus compute the spatio-temporal IoU between the tubes generated by our online algorithm and the ground truth tubes, up to time $t$.

Fig. 4 plots the AUC curves against the observed portion of the video at different IoU thresholds ($\delta = 0.2$ and 0.5) for the proposed approach versus our competitor [42]. Our method outperforms [42] on online action localisation by a large margin at all the IoU thresholds and video observation percentage. Notice that our online localisation performance (Fig. 4) is a stable function of the video observation percentage, whereas, Soomro *et al*. [42]'s method needs some 'warm-up' time to reach stability, and its accuracy slightly decreases at the end. In addition, [42] only reports online spatial localisation results on the temporally trimmed J-HMDB-21 test videos, and their approach lacks

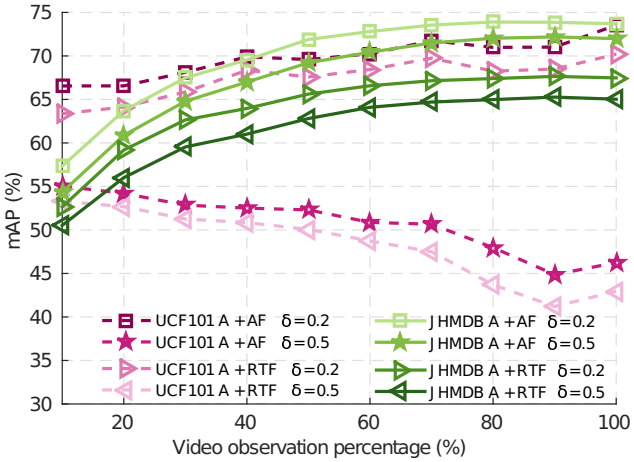


Figure 5. *Action localisation results using the mAP (%) metric on UCF101-24 and JHMDB-21, at IoU thresholds of* $\delta = 0.2, 0.5$.

temporal detection capabilities.

Our framework, instead, can perform online spatio-temporal localisation: to demonstrate this, we present results on the temporally untrimmed UCF101-24 test videos as well. In Fig. 5 we report online spatial-temporal localisation results on UCF101-24 and JHMBD-21 using the standard mAP metric (not reported in [42]). Interestingly, for UCF101-24, at a relatively smaller IoU threshold ($\delta = 0.2$) the performance gradually increases over time as more video frames are observed, whereas at a higher IoU threshold ($\delta = 0.5$) it slightly degrades over time. A reason for this could be that UCF101-24 videos are temporally untrimmed and contain multiple action instances, so that accurate detection may be challenging at higher detection thresholds (e.g. $\delta = 0.5$). If temporal labelling is not very accurate, as required at high thresholds ($\delta = 0.5$), this might result in more false positives as the video progress, hence the observed drop in performance over time.

#### 4.2.2 Global performance

To demonstrate the strength of our online framework, we compare as well its absolute detection performances to those of the top offline competitors [33, 53, 28, 58]. To ensure a fair comparison with [33], we evaluate their offline tube generation method using the detection bounding boxes produced by the SSD net. As in [21], we report the mAP averaged over thresholds from 0.5 to 0.95 in steps of 0.05.

**Improvement over the top performers**. Results on UCF101-24 are reported in Table 1. In an online real-time setting we achieve an mAP of 70.2% compared to 66.6% reported by [33] at the standard IoU threshold of $\delta = 0.2$. In non-real time mode, we observe a further performance improvement of around 3.3%, leading to a 73.5% mAP, comparable to the 73.5 reported by the current top performer [28]. The similar performance of our method (A+AF) to [28] at $\delta = 0.2$ suggests that SSD and the multi-region adaptation of Faster-RCNN by [28] produce similar quality frame level detection boxes.

Table 1. S/T action localisation results (mAP) on untrimmed videos of UCF101-24 dataset in split1.

| IoU threshold $\delta$ | 0.2 | 0.5 | 0.75 | 0.5:0.95 |
|---|---|---|---|---|
| Yu *et al.* [58]‡ | 26.5 | – | – | – |
| Weinzaepfel *et al.* [53]‡ | 46.8 | – | – | – |
| Peng and Schmid [28]† | **73.5** | 32.1 | 02.7 | 07.3 |
| Saha *et al.* [33]† | 66.6 | 36.4 | 07.9 | 14.4 |
| Ours-Appearance (A)* | 69.8 | 40.9 | **15.5** | 18.7 |
| Ours-Real-time-flow (RTF)* | 42.5 | 13.9 | 00.5 | 03.3 |
| Ours-A + RTF (boost-fusion)* | 69.7 | 41.9 | 14.1 | 18.4 |
| Ours-A + RTF (union-set)* | 70.2 | 43.0 | 14.5 | 19.2 |
| Ours-Accurate - flow (AF)** | 63.7 | 30.8 | 02.8 | 11.0 |
| Ours-A + AF (boost-fusion)** | 73.0 | 44.0 | 14.1 | 19.2 |
| Ours-A + AF (union-set)** | **73.5** | **46.3** | 15.0 | **20.4** |
| SSD+ [33] A + AF (union-set)† | 71.7 | 43.3 | 13.2 | 18.6 |

‡ These methods were using different annotations to [28, 33] and ours.
* Incremental & real-time ** Incremental, non real-time † Offline

**Performance under more realistic requirements**. Our method significantly outperforms [33, 28] at more meaningful higher detection thresholds $\delta = 0.5$ or higher. For instance, we achieve a 46.2% mAP at $\delta = 0.5$ as opposed to the 32.1% by [28] and the 36.4% by [33], an improvement of **14%** and **9.8%**, respectively. This attests the superiority of our tube building algorithm when compared to those of [28, 33]. In fact, even in real time mode our pipeline (A + RTF) still performs better than both [33, 28] at $\delta = 0.5$ or higher.

It is important to note that, our proposed fusion method (*union-set-fusion*) significantly outperforms *boost-fusion* proposed by [33] on UCF101-24 dataset (see Table 1). UCF-101 includes many co-occurring action instances, we can infer that the union-set fusion strategy improves the performance by providing a larger number of high confidence boxes from either the appearance or the flow network. When a single action is present in each video, as in JHMDB, *boost-fusion* perform better (Table 2). In the supplementary material we present a complete class-wise performance comparison of the two fusion strategies on both datasets.

**Evaluation on J-HMDB-21**. Table 2 reports action detection results averaged over the three splits of *J-HMDB-21*, and compares them with those to our closest (offline) competitors. Our framework outperforms the multi-stage approaches of [7, 52, 53] in non real-time mode at the standard IoU threshold of 0.5, while it attains figures very close to those of [33, 28] (73.8 versus 74.1 and 72.6, respectively) approaches, which make use of a two-stage Faster-RCNN.

Once again it is very important to point out that [28] employs a battery of frame-level detectors, among which one based on strong priors on human body parts. Our approach does not make any prior assumption on the object(s)/actors(s) performing the action of interest, and is thus arguably more general-purpose.

Table 2. S/T Action localisation results (mAP) on J-HMDB-21.

| IoU threshold $\delta$ | 0.2 | 0.5 | 0.75 | 0.5:0.95 |
|---|---|---|---|---|
| Gkioxari and Malik [7]† | – | 53.3 | – | – |
| Wang *et al*. [52]† | – | 56.4 | – | – |
| Weinzaepfel *et al*. [53]† | 63.1 | 60.7 | – | – |
| Saha *et al*. [33]† | 72.6 | 71.5 | 43.3 | 40.0 |
| Peng and Schmid [28]† | **74.1** | **73.1** | – | – |
| Ours-Appearance (A)* | 60.8 | 59.7 | 37.5 | 33.9 |
| Ours-Real-time-flow (RTF)* | 56.9 | 47.4 | 20.2 | 19.3 |
| Ours-A + RTF (union-set)* | 66.0 | 63.9 | 35.1 | 34.4 |
| Ours-A + RTF (boost-fusion)* | 67.5 | 65.0 | 36.7 | 38.8 |
| Ours-Accurate - flow (AF)** | 68.5 | 67.0 | 38.7 | 36.1 |
| Ours-A + AF (union-set)** | 70.8 | 70.1 | 43.7 | 39.7 |
| Ours-A + AF (boost-fusion)** | 73.8 | 72.0 | **44.5** | **41.6** |
| SSD+ [33] A + AF (boost-fusion)† | 73.2 | 71.1 | 40.5 | 38.0 |

* Incremental & real-time ** Incremental, non real-time † Offline

### 4.3. Discussion

**Contribution of the flow stream.** The optical flow stream is an essential part of the framework. Fusing the real-time flow stream with the appearance stream (A+RTF mode) on UCF101-24 leads to a 2.1% improvement at $\delta = 0.5$. Accurate flow adds a further 3.3%. A similar trend can be observed on JHMDB-21, where A+RTF gives a 5% boost at $\delta = 0.5$, and the A+RTF mode takes it further to 72%. It is clear from Table 1 and Table 2 that optical flow plays a much bigger role on the JHMDB dataset as compared to UCF101-24. Real-time OF does not provide as big a boost as accurate flow, but still pushes the overall performance towards that of the top competitors, with the invaluable addition of real-time speed.

**Relative contribution of tube generation and SSD.** As anticipated we evaluated the offline tube generation method of [33] using the detection bounding boxes produced by the SSD network, to both provide a fair comparison and to understand each component's influence on performance. The related results appear in the last row of Table 1 and Table 2. From comparing the figures in the last two rows of both tables it is apparent that our online tube generation performs better than the offline tube generation of [33], especially providing significant improvements at higher detection thresholds for both datasets. We can infer that the increase in performance comes from both the higher-quality detections generated by SSD, as well as our new online tube generation method. The fact that our tube genration is online, gready and outperforms offline methods, so it suggests that offline approaches has big room for improvements.

The reason for not observing a big boost due to the use of SSD on JHMDB may be its relatively smaller size, which does not allow us to leverage on the expressive power of SSD models. Nevertheless, cross validating the CNNs' hyper-parameters (e.g. learning rate), might lead to further improvements there as well.

### 4.4. Test time detection speed

To support our claim to real time capability, we report the test time detection speed of our pipeline under all three types of input A (RGB), A+RTF (real-time flow), A + AF (accurate flow) in Table 3. These figures were generated using a desktop computer with an Intel Xeon CPU@2.80GHz (8 cores) and two NVIDIA Titan X GPUs. Real-time capabilities can be achieved by either not using optical flow (using only appearance (A) stream on one GPU) or by computing real-time optical flow [16] on a CPU in parallel with two CNN forward passes on two GPUs. For action tube generation (§ 3.4) we ran 8 CPU threads in parallel for each class. We used the real-time optical flow algorithm [16] in a customised setting, with minimum number of pyramid levels set to 2 instead of 3, and patch overlap 0.6 rather than 0.4. OF computation averages $\sim$ 7 ms per image.

Table 3 also compares our detection speed to that reported by Saha *et al*. [33]. With an overall detection speed of 40 fps (when using RGB only) and 28 fps (when using also real time OF), our framework is able to detect multiple co-occurring action instances in real-time, while retaining very competitive performance.

Table 3. Test time detection speed.

| Framework modules | A | A+RTF | A+AF | [33] |
|---|---|---|---|---|
| Flow computation (ms*) | – | 7.0 | 110 | 110 |
| Detection network time (ms*) | 21.8 | 21.8 | 21.8 | 145 |
| Tube generation time (ms*) | 2.5 | 3.0 | 3.0 | 10.0 |
| Overall speed (fps** ) | 40 | 28 | 7 | 4 |

* ms - milliseconds ** fps - frame per second.

## 5. Conclusions and future plans

We presented a novel online framework for action localisation and prediction able to address the challenges involved in concurrent multiple human action recognition, spatial localisation and temporal detection, in real time. Thanks to an efficient deep learning strategy for the simultaneous detection and classification of region proposals and a new incremental action tube generation approach, our method achieves superior performances compared to the previous state-of-the-art on early action prediction and online localisation, while outperforming the top offline competitors, in particular at high detection overlap. Its combination of high accuracy and fast detection speed at test time paves the way for its application to real-time applications such as autonomous driving, human robot interaction and surgical robotics, among others.

A number of future extensions can be envisaged. Motion vectors [60], for instance, could be used in place of optical flow to achieve faster detection speeds. An even faster frame level detector, such as YOLO [29], could be easily incorporated. More sophisticated online tracking algorithms [54] for tube generation could be explored.

## References


[1] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. 2004. 2, 3, 5

[2] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. 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

[3] R. De Geest, E. Gavves, A. Ghodrati, Z. Li, C. Snoek, and T. Tuytelaars. Online action detection. *arXiv preprint arXiv:1604.06506*, 2016. 2, 3

[4] G. Evangelidis, G. Singh, and R. Horaud. Continuous gesture recognition from articulated poses. In *ECCV Workshops*, 2014. 2, 5

[5] A. Gaidon, Z. Harchaoui, and C. Schmid. Temporal localization of actions with actoms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2782–2795, 2013. 2

[6] R. Girshick, J. Donahue, T. Darrel, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014. 1, 2

[7] G. Gkioxari and J. Malik. Finding action tubes. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2015. 1, 2, 3, 4, 7, 8

[8] A. Gorban, H. Idrees, Y. Jiang, A. R. Zamir, I. Laptev, M. Shah, and R. Sukthankar. Thumos challenge: Action recognition with a large number of classes, 2015. 6

[9] M. Hoai and F. De la Torre. Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202, 2014. 3

[10] M. Jain, J. Van Gemert, H. Jégou, P. Bouthemy, and C. G. Snoek. Action localization with tubelets from motion. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 740–747. IEEE, 2014. 3

[11] S. D. Jain and K. Grauman. Supervoxel-consistent foreground propagation in video. In *Computer Vision–ECCV 2014*, pages 656–671. Springer, 2014. 2

[12] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. Black. Towards understanding action recognition. 2013. 5, 6

[13] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):221–231, Jan 2013. 2

[14] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2014. 2

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012. 1

[16] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool. Fast optical flow using dense inverse search. *arXiv preprint arXiv:1603.03590*, 2016. 2, 3, 5, 8

[17] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011. 6

[18] K. Kulkarni, G. Evangelidis, J. Cech, and R. Horaud. Continuous action recognition based on sequence alignment. *International Journal of Computer Vision*, 112(1):90–114, 2015. 2

[19] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *Computer Vision–ECCV 2014*, pages 689–704. Springer, 2014. 3

[20] I. Laptev and P. Pérez. Retrieving actions in movies. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007. 2

[21] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 7

[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. *arXiv preprint arXiv:1512.02325*, 2015. 2, 3, 4

[23] J. Lu, r. Xu, and J. J. Corso. Human action segmentation with hierarchical supervoxel consistency. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, June 2015. 2

[24] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1942–1950, 2016. 6

[25] B. Majecka. Statistical models of pedestrian behaviour in the forum. *Master's thesis, School of Informatics, University of Edinburgh*, 2009. 4

[26] A. Milan, K. Schindler, and S. Roth. Multi-target tracking by discrete-continuous energy minimization. *IEEE TPAMI*, 2016. 4

[27] D. Oneata, J. Verbeek, and C. Schmid. Efficient action localization with approximately normalized fisher vectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2545–2552, 2014. 2, 3

[28] X. Peng and C. Schmid. Multi-region two-stream R-CNN for action detection. In *ECCV 2016 - European Conference on Computer Vision*, Amsterdam, Netherlands, Oct. 2016. 1, 2, 3, 4, 6, 7, 8

[29] J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016. 8

[30] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 2, 3, 4

[31] M. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1036–1043. IEEE, 2011. 3

[32] M. S. Ryoo and J. Aggarwal. Ut-interaction dataset, icpr contest on semantic description of human activities (sdha). In *IEEE International Conference on Pattern Recognition Workshops*, volume 2, page 4, 2010. 3

[33] S. Saha, G. Singh, M. Sapienza, P. H. S. Torr, and F. Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *British Machine Vision Conference*, 2016. 1, 2, 3, 4, 5, 6, 7, 8

[34] M. Sapienza, F. Cuzzolin, and P. H. Torr. Learning discriminative space-time action parts from weakly labelled videos. *Int. Journal of Computer Vision*, 2014. 3

[35] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: a local svm approach. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 3, pages 32–36. IEEE, 2004. 3

[36] Z. Shou, D. Wang, and S. Chang. Action temporal localization in untrimmed videos via multi-stage cnns. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, June 2016. 2

[37] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems 27*, pages 568–576. Curran Associates, Inc., 2014. 2, 3

[38] B. Singh and M. Shao. A multi-stream bi-directional recurrent neural network for fine-grained action detection. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2016. 2

[39] G. Singh. Categorising the abnormal behaviour from an indoor overhead camera. *Bachelor's thesis, VIT University*, 2010. 4

[40] G. Singh and F. Cuzzolin. Untrimmed video classification for activity detection: submission to activitynet challenge. *arXiv preprint arXiv:1607.01979*, 2016. 2

[41] K. Soomro, H. Idrees, and M. Shah. Action localization in videos through context walk. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. 2

[42] K. Soomro, H. Idrees, and M. Shah. Predicting the where and what of actors and actions through online action localization. 2016. 2, 3, 6, 7

[43] K. Soomro, A. R. Zamir, and M. Shah. UCF101: A dataset of 101 human action classes from videos in the wild. Technical report, CRCV-TR-12-01, 2012. 1, 5

[44] R. Šrámek, B. Brejová, and T. Vinař. On-line viterbi algorithm and its relationship to random walks. *arXiv preprint arXiv:0704.0062*, 2007. 5

[45] W. Sultani and M. Shah. What if we do not have multiple videos of the same action? - video action localization using web images. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2016. 3

[46] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2642–2649. IEEE, 2013. 2

[47] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *Proc. Int. Conf. Computer Vision*, 2015. 2

[48] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *Int. Journal of Computer Vision*, 2013. 2

[49] J. C. van Gemert, M. Jain, E. Gati, and C. G. Snoek. APT: Action localization proposals from dense trajectories. In *BMVC*, volume 2, page 4, 2015. 3

[50] H. Wang, A. Kläser, C. Schmid, and C. Liu. Action Recognition by Dense Trajectories. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2011. 3

[51] H. Wang, D. Oneata, J. Verbeek, and C. Schmid. A robust and efficient video representation for action recognition. *International Journal of Computer Vision*, pages 1–20, 2015. 2

[52] L. Wang, Y. Qiao, X. Tang, and L. Van Gool. Actionness estimation using hybrid fully convolutional networks. *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, 2016. 2, 7, 8

[53] P. Weinzaepfel, Z. Harchaoui, and C. Schmid. Learning to track for spatio-temporal action localization. In *IEEE Int. Conf. on Computer Vision and Pattern Recognition*, June 2015. 1, 2, 3, 4, 6, 7, 8

[54] Y. Wu, J. Lim, and M.-H. Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013. 8

[55] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. *arXiv preprint arXiv:1411.4006*, 2014. 2

[56] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *arXiv preprint arXiv:1507.05738*, 2015. 2

[57] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. *CVPR*, 2016. 2

[58] G. Yu and J. Yuan. Fast action proposals for human action detection and search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1302–1311, 2015. 6, 7

[59] T.-H. Yu, T.-K. Kim, and R. Cipolla. Real-time action recognition by spatiotemporal semantic and structural forests. In *BMVC*, volume 2, page 6, 2010. 3

[60] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector cnns. *CVPR*, 2016. 3, 4, 8

[61] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014. 2, 3