Supplementary Material: 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

1. Implementation details

Training data preparation. We used all the training video frames from J-HMDB-21 dataset to train both appearance- and flow-based SSD networks. As UCF101-24 has larger training set, we used every 4-th frame from each training video along with their associated ground-truth labels and bounding boxes. Frames that do not contain any human action annotation (background frames) were not used in training. Using those video frames (background frames) and modifying the cost function of SSD could further lead to improvement in temporal localisation of actions.

Optical flow based video frame generation. We computed dense optical flow between each pair of successive video frames using the algorithms of [1] (for accurate non real-time flow) and [3] (for real-time flow). A 3-channel optical flow image is constructed from flow components (i.e., flow-x, flow-y and the flow magnitude) by taking a similar approach as in [2].

CNN weight initialisation. SSD [4] utilises ParseNet [5], a fully convolutional network, based on VGG net [7] as a base model. In addition, it adds few convolution layers on the top. VGG network weights were initialised with weights from a pre-trained ImageNet model [5][1] for UCF101-24 appearance- and accurate flow-based SSD networks. Following the transfer learning approach [8], UCF101-24 real-time flow based SSD network weights were initialised with accurate flow based SSD network weights upto fc7 layer. We transfer VGG network weights learned on UCF101-24 to train appearance- and flow-based SSD networks on J-HMDB-21 dataset.

CNN solver configuration setting. We trained network with slightly modified training parameters. For UCF-101, we use a base learning rate of 0.0001 instead 0.0004. We observed better convergence with relatively lower learning rate 0.0001. For J-HMDB-21 dataset, we dropped the learning rate further to 0.00004. The step size was set to 40000 and 15000 iterations for UCF101-24 and J-HMDB-21 respectively. After predefined number of training iterations (i.e. the step size), the learning rate was dropped by a factor of 10. We used a single Titan-X GPU to train networks with batch size of 32 for both the datasets. It took 2 days to train SSD network to train on UCF101-24 and 1 day to train on JHMDB-21.

2. Ablation study

We report an ablation study of the online spatio-temporal action localisation performance on UCF-101 dataset. Table 1 and Table 2 show the class-specific video AP (average precision in $\%$) for each action category of UCF-101 generated by the appearance (A) model, appearance plus real-time flow (A & RTF) fusion model and appearance plus accurate flow (A & AF) fusion model. Results are generated at a spatio-temporal overlap threshold of $\delta = 0.2$ in Table 1 and $\delta = 0.5$ in Table 2.

Difference between two fusion can be observed (Table 1) in classes with multiple actors "IceDancing", "SalsaSpin" and "Biking", union-set fusion shows significant improvement when compared with boost-fusion strategy.

Difficult classes. "Basketball", "CricketBowling", "VolleyballSpiking" and "TennisSwing" are the most difficult classes. Most of the "Basketball" training videos have at least one actor (basketball player) present in the video, however, the "Basketball" action is performed within a small temporal extent. As temporal localisation is difficult, action categories with relatively large number of temporally untrimmed test videos such as "CricketBowling" , "VolleyballSpiking" and "TennisSwing" show lower APs. Similarly in "CricketBowling" class, an actor is present in most part of the video, but the action is annotated within a smaller temporal extent. Further, running (during the "CricketBowling" action) is not considered as a part of the action which makes it even more difficult to detect. "VolleyballSpiking" videos contain many potential actors (Volleyball players) which are difficult to distinguish. It is clear from evidences that, it is necessary to retrain network using background frame, which is not done. To achieve that we will need to modify the cost function of SSD to accept an image

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

[1] https://gist.github.com/weiliu89/2ed6e13bfd5b57cf81d6

Table 1. Spatio-temporal detection results (video APs in %) on UCF101-24 at $\delta = 0.2$ along with class-wise statistics about UCF101-24 dataset in first two rows (number of action instance per video and action instance duration compared to video duration).

Actions	Basketball	BasketballDunk	Biking	CliffDiving	CricketBowling	Diving	Fencing	FloorGymnastics
Number of Actions/Video	1.0	1.0	1.8	1.0	1.1	1.0	2.4	1.0
Action/Video duration (ratio)	0.34	0.59	0.70	0.64	0.36	0.65	0.89	1.00
Saha *et al*. [6]	39.6	49.7	66.9	73.2	14.1	93.6	85.9	99.8
Ours-Appearance (A)	43.0	67.4	75.8	67.2	50.5	100.0	88.5	97.9
Ours-A + RTF (boost-fusion)	42.2	69.0	71.7	73.1	41.3	100.0	87.6	99.1
Ours-A + RTF (union-set)	42.0	64.6	73.7	75.2	41.5	100.0	86.5	97.9
Ours-A + AF (boost-fusion)	45.0	86.4	67.6	78.2	44.2	100.0	89.8	99.9
Ours-A + AF (union-set)	43.9	81.6	73.6	73.7	49.0	100.0	90.2	97.9
SSD+[6] A + AF (union-set)	43.2	78.5	65.8	72.0	43.6	100.0	86.1	98.1

Actions	GolfSwing	HorseRiding	IceDancing	LongJump	PoleVault	RopeClimbing	SalsaSpin	SkateBoarding
Number of Actions/Video	1.0	1.0	2.3	1.0	1.1	1.0	4.9	1.0
Action/Video duration (ratio)	0.67	0.95	0.85	0.98	0.81	1.00	0.34	1.00
Saha *et al*. [6]	68.3	94.1	63.1	57.2	75.1	89.6	31.1	85.1
Ours-Appearance (A)	59.9	95.9	56.5	59.7	80.8	93.4	36.9	86.1
Ours-A + RTF (boost-fusion)	64.3	96.0	72.8	68.8	72.7	94.5	19.7	85.6
Ours-A + RTF (union-set)	62.1	96.0	77.6	69.7	76.1	96.1	22.2	87.4
Ours-A + AF (boost-fusion)	65.8	96.0	74.0	81.4	80.3	95.8	23.1	88.3
Ours-A + AF (union-set)	62.0	96.0	76.3	82.9	82.7	98.1	25.7	87.8
SSD+[6] A + AF (union-set)	61.3	96.0	60.6	83.4	84.6	98.6	20.1	88.4

Actions	Skiing	Skijet	SoccerJuggling	Surfing	TennisSwing	TrampolineJumping	VolleyballSpiking	WalkingWithDog
Number of Actions/Video	1.0	1.0	1.1	1.6	1.3	2.5	1.1	1.1
Action/Video duration (ratio)	1.00	0.95	0.86	0.53	0.30	0.65	0.31	0.84
Saha *et al*. [6]	79.6	96.1	89.1	63.2	33.6	52.7	20.9	75.6
Ours-Appearance (A)	78.6	94.6	71.0	65.4	37.6	59.4	24.8	83.7
Ours-A + RTF (boost-fusion)	78.8	89.9	82.1	62.1	31.7	57.7	27.0	83.2
Ours-A + RTF (union-set)	81.0	87.1	82.4	62.1	37.4	59.4	21.7	85.1
Ours-A + AF (boost-fusion)	80.0	91.1	93.1	65.4	38.7	57.4	28.3	84.3
Ours-A + AF (union-set)	81.8	93.2	93.0	65.8	38.2	58.5	26.1	86.9
SSD+[6] A + AF (union-set)	82.0	93.5	92.7	61.1	38.8	56.8	30.7	86.6

without any positive instance. At the moment it requires at least one positive instance present in any video frame.

Easy classes. "FloorGymnastics", "HorseRiding" and "SoccerJuggling" are the most easy classes to detect. Possibly, because these classes contain mostly one actor at a time and have salient appearance features. For instance, presence of horse in the "HorseRiding" class.

Dataset statistics of UCF101-24 dataset. The first two rows of Table 1 shows class-wise statistics of UCF101-24 dataset. Number of action instances per video are shown in first row, averaged over test-list of split 1. Duration of action instance compared to duration of video is shown in second row, averaged over test-list of split 1. It is clear that the most difficult classes are those which have lower temporal duration of action instances than the entire video duration. For instance, on an average a basketball action is only performed in 34% of the entire video sequence. We can also see is huge performance difference when detection threshold is increased from $\delta = 0.2$ to 0.5 in Table 2.

3. Algorithm pseudo-code

In Algorithm 1, we provide a pseudocode of the proposed online incremental tube generation algorithm. Note that, tubes are sorted at every time step in decreasing order as per their confidence scores, so that, the best tube is assigned the highest scoring box first. In all our experiments, we set $\lambda = 0.1$, $n = 10$, $k = 5$ and $\alpha_c = 3$. Code is made available online at https://github.com/gurkirt/realtime-action-detection.

Table 2. Spatio-temporal detection results (video APs in %) on UCF101-24 at $\delta = 0.2$ along with class-wise statistics about UCF101-24 dataset in first two rows (number of action instance per video and action instance duration compared to video duration).

Actions	Basketball	BasketballDunk	Biking	CliffDiving	CricketBowling	Diving	Fencing	FloorGymnastics
Number of Actions/Video	1.0	1.0	1.8	1.0	1.1	1.0	2.4	1.0
Action/Video duration (ratio)	0.34	0.59	0.70	0.64	0.36	0.65	0.89	1.00
Saha *et al*. [6]	0.1	0.0	29.6	13.4	0.7	39.7	46.7	82.7
Ours-Appearance (A)	0.1	5.3	56.9	16.6	1.8	20.2	74.8	95.2
Ours-A & RTF (boost-fusion)	0.1	5.3	41.2	17.6	1.0	16.8	75.7	99.1
Ours-A & RTF (union-set)	0.0	5.3	57.6	24.0	1.3	21.3	72.5	95.3
Ours-A & AF (boost-fusion)	0.0	5.3	45.2	27.3	2.0	17.8	69.2	97.2
Ours-A & AF (union-set)	0.0	5.3	55.1	31.4	1.5	19.6	71.9	95.2
SSD+[6] A + AF (union-set)	0.0	5.3	55.6	20.0	1.6	14.6	55.6	87.5

Actions	GolfSwing	HorseRiding	IceDancing	LongJump	PoleVault	RopeClimbing	SalsaSpin	SkateBoarding
Number of Actions/Video	1.0	1.0	2.3	1.0	1.1	1.0	4.9	1.0
Action/Video duration (ratio)	1.00	0.95	0.86	0.53	0.30	0.65	0.31	0.84
Saha *et al*. [6]	34.3	90.0	44.2	32.7	05.1	65.4	2.9	82.2
Ours-Appearance (A)	33.7	88.8	42.7	32.3	13.2	84.5	3.0	82.8
Ours-A + RTF (boost-fusion)	36.0	91.5	45.6	48.9	13.2	89.5	1.1	79.1
Ours-A + RTF (union-set)	34.5	90.9	47.1	45.3	15.9	90.0	1.8	79.9
Ours-A + AF (boost-fusion)	38.9	91.4	43.0	62.4	21.5	90.3	1.4	82.3
Ours-A + AF (union-set)	37.0	90.9	48.4	60.5	32.8	92.5	1.6	81.6
SSD+[6] A + AF (union-set)	36.3	91.0	43.9	53.5	24.5	92.8	2.2	81.9

Actions	Skiing	Skijet	SoccerJuggling	Surfing	TennisSwing	TrampolineJumping	VolleyballSpiking	WalkingWithDog
Number of Actions/Video	1.0	1.0	1.1	1.6	1.3	2.5	1.1	1.1
Action/Video duration (ratio)	0.67	0.95	0.85	0.98	0.81	1.00	0.34	1.00
Saha *et al*. [6]	68.0	78.1	74.9	25.5	0.8	11.8	0.0	44.2
Ours-Appearance (A)	73.7	75.5	55.5	50.0	0.8	18.2	0.0	55.8
Ours-A + RTF (boost-fusion)	75.8	71.6	73.1	41.4	0.8	28.4	0.0	53.7
Ours-A + RTF (union-set)	77.0	73.8	76.3	40.0	0.8	23.4	0.0	56.9
Ours-A + AF (boost-fusion)	71.3	76.8	85.6	43.9	0.6	27.2	0.0	55.9
Ours-A + AF (union-set)	77.0	85.8	87.4	47.3	0.8	23.6	0.0	63.0
SSD+[6] A + AF (union-set)	74.7	89.0	86.8	45.9	0.8	19.8	0.0	55.8

%% Imgae at time t is represented by img_t;
for *t = 1 to* $end - 1$ **do**
  compute flow image $flowimg_t$ using img_t and img_{t+1};
  get appearance detection dt_a using appearance network;
  get flow detection dt_f using flow network;
  for *each class c* **do**
    get top n dt_a and dt_f detection boxes;
    get fused detections dt after fusion of dt_a and dt_f;
    apply non-maximal suppression on dt and keep top n detections dt ;
    %% detections dt is an array of structure with $dt[1].box$ and $dt[1].scores$ as fields for first box;
    if *t == 1* **then**
      initialise $n_c = n$ $tubes$ with top n boxes from dt;
      %% $tubes$ is a 2D array of structure with $tubes[c][1].boxes[t_1]$ and $tubes[c][1].socres[t_1]$
      %% $tube[c][1].cost[t_1, :]$ as fields for first tube of class c at time t_1 or $t = 1$
    else
      sort n_c $tubes$ in decreasing order;
      for $id = 1$ *to* n_c **do**
        $scores = zeros(n)$;
        for $b = 1$ *to* n **do**
          if *IoU(*$tube[c][id].boxes[t\text{-}1]$, $dt[b].box$*)*$> \lambda$ **then**
            $scores[b] = dt[b].scores[c]$;
        if *any(*$scores > 0$*)* **then**
          $maxindex = argmax(scores)$;
          $tube[c][id].boxes[t] = dt[maxindex].box$;
          $tube[c][id].scores[t] = scores[maxindex]$;
          $dt[maxindex].scores[c] = 0$;
          %% appended the box to tube and removed the box by setting score to zero
        if *detection not found for k frames* **then**
          terminate($tubes[c][id]$)
        %% updating temporal labelling costs for $tube[c][id]$
        $score_c$ = $tube[c][id].scores[t]$; $score_0$ = $1 - tube[c][id].scores[t]$
        for l_t *in c and* 0 **do**
          $V(l_t, l_{t-1}) = 0\ if\ \ l_t == l_{t-1}\ alpha_c otherwise$;
          $tube[c][id].cost[t, l_t] = score_{l_t} + \max_{l_{t-1}}(tube[c][id].cost[t-1, l_{t-1}] - V(l_t, l_{t-1}))$;
        if *get tube labelling OR tube terminated* **then**
          %% get tube label recursively $l_t^* = argmax_{l_i}(tube[c][id].cost[t, l_t] - V(l_t, l_{t-1}^*))$
      if *any detection dt left unassigned* **then**
        initialise a new $tube$ with $dt.box$ and $dt.scores[c]$;

Algorithm 1: Incremental online tube generation.

4. Qualitative results

Fig. 1 and 2 show qualitative results of spatiotemporal action localisation on temporally untrimmed "Fencing" and "Surfing" action sequences taken from UCF-101 test-set. Fig. 3 shows sample early action label prediction and online action localisation qualitative results on J-HMDB-21 dataset. Fig. 4 provides additional evidence on the action localisation performance of our method on UCF101 dataset.

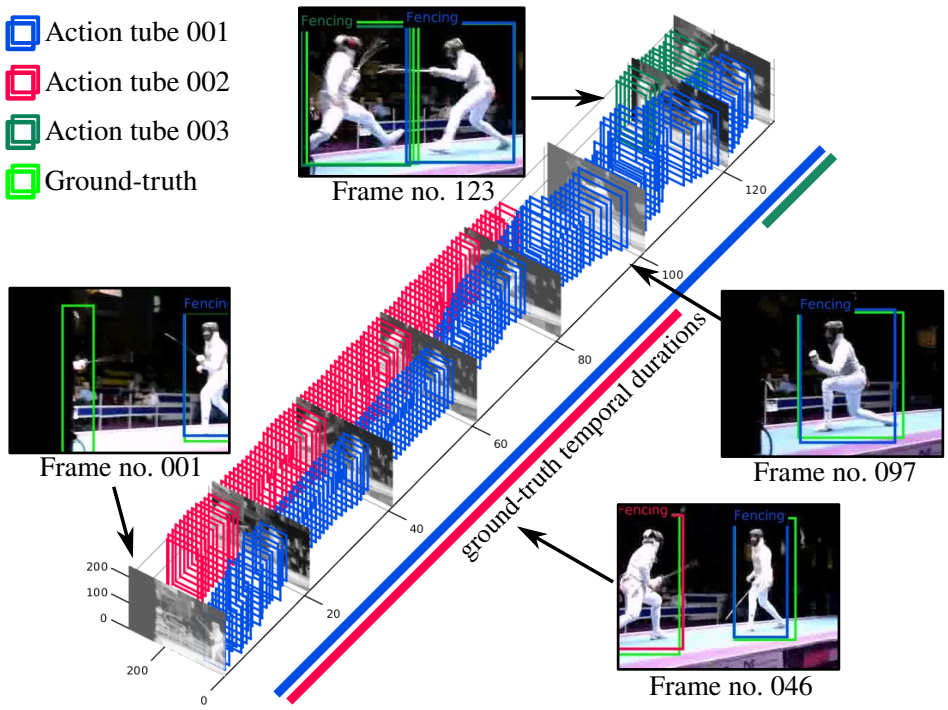

Figure 1. *Sample spatiotemporal action localisation results on a "Fencing" action sequence taken from UCF101 testset. The detected action tubes are plotted in 3D and drawn in different colour indicating 3 different action instances. The ground-truth temporal duration of each action instance is shown by the coloured bars. Note that the temporal durations of the detected and the ground-truth tubes are closely matched (with good temporal overlaps).*

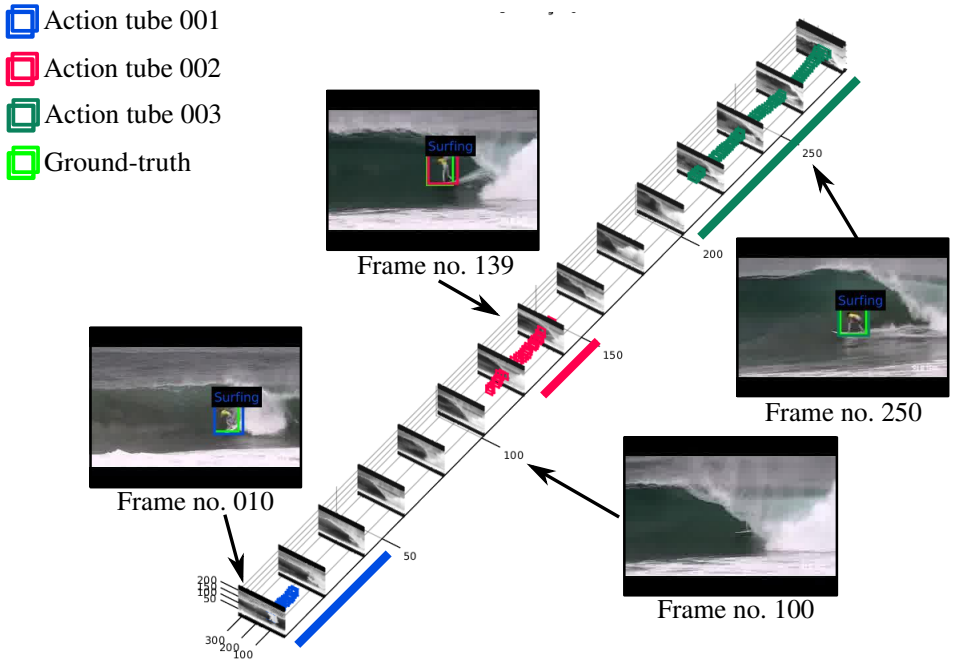

Figure 2. *Sample spatiotemporal action localisation results on a "Surfing" action sequence taken from UCF101 testset. The detected action tubes are plotted in 3D and drawn in different colour indicating 3 different action instances. The ground-truth temporal duration of each action instance is shown by the coloured bars. Note that the temporal durations of the detected and the ground-truth tubes are closely matched.*

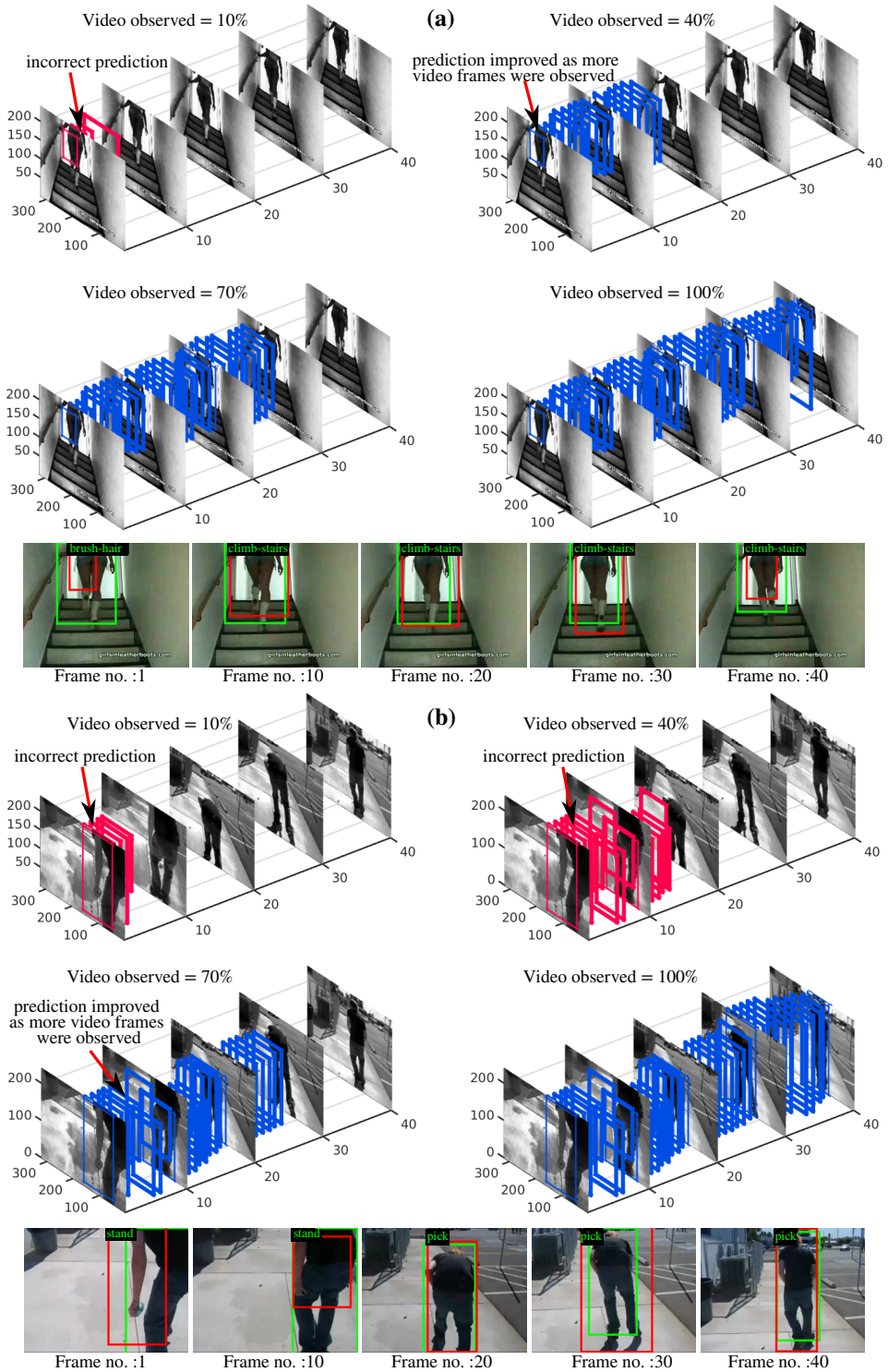

Figure 3. *Sample early action label prediction and online action localisation results on J-HMDB-21 dataset.* ***(a)*** *and* ***(b)*** *show prediction results of 2 different test videos with ground-truth action labels 'climb-stairs' and 'pick' respectively. Each video and its corresponding space-time detection tube were plotted in 3D at different time points (i.e., % of video observed). Detection tubes are drawn in two different colours to indicate the wrong early label prediction and the improved prediction as more video frames were observed in time. Just below the 3D plot, the predicted action labels for the same video at different time points are overlaid on the corresponding video frames in which the green box depicts the ground-truth and red depicts the predicted bounding box.*

Figure 4. Sample action localisation results on UCF-101. Each row represents a UCF-101 test video clip. Ground-truth bounding boxes are drawn in green and detection boxes are in red.

References

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

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

[3] T. Kroeger, R. Timofte, D. Dai, and L. Van Gool. Fast optical flow using dense inverse search. *arXiv preprint arXiv:1603.03590*, 2016. 1

[4] 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. 1

[5] W. Liu, A. Rabinovich, and A. C. Berg. Parsenet: Looking wider to see better. *arXiv preprint arXiv:1506.04579*, 2015. 1

[6] 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. 2, 3

[7] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1

[8] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector cnns. *CVPR*, 2016. 1