Spatio-Temporal Action Detection Under Large Motion
Supplemental Material

Gurkirt Singh Vasileios Choutas Suman Saha Fisher Yu
Luc Van Gool
Computer Vision Lab, ETH Zürich

1. Overview

Here, we aim to provide additional details about the certain parts of main paper. First, we show architecture of backbone network, where feature pyramid network structure (FPN) incorporated into SlowFast [2] network in figure Fig. 1. Second, we show the structure of TFA modules used in our TAAD model in Sec. 2. Then, we present frame-level Motion-mAP and MotionAP o n individual time scales in Sec. 3 Finally, we show visual results of detected action-tube instances under different motion types in Sec. 4.

2. Structure of Temporal Feature Aggregation Modules

Listing 1, 2, and 3 contain the PyTorch implementation of our MaxPool, TCN and ASPP TFA modules, used with our Track Aware Action Detector (TAAD) method. Similar to 1D-ASPP module (see Listing-Fig. 3), In addition to these blocks, we tried 1D-ConvNeXt [4] and 1D-Swin [5] blocks as well, but observed that training was unstable and that the final performance was worse. For example, 1D-ConvNext could only reach up to 44% f-mAP compared to 53.3% using MaxPool module (Listing-Fig. 1). In all the listings that follow, C is the number of channels and T is the number of input frames.

3. Individual time scales results

Frame-level Motion-mAP and MotionAP on individual time scales are shown in following tables:

(1) MotionAP on MultiSports in Tab. 1

(2) MotionAP on UCF24 in Tab. 2

(3) Motion-mAP on MultiSports in Tab. 4

(4) Motion-mAP on UCF24 in Tab. 3

The main take-away from these tables is that results are consistent across different time scales compared to the average over all time scales. Through the MotionAP metric, we show that that large-motion action instances are harder to detect compared to medium motions, which in turn are even harder to detect compared to small-motion action instances, or in other words performance of large-motion $<$ medium-motion $<$ small-motion. This result is consistent across both our benchmarks, *i.e*. MultiSports and UCF24. Such pattern is desirable and intuitive to understand, it is missing in Motion-mAP because some class might not have any (or very few) ground-truth instances in one or two motion type categories resulting very small value of mAP

Table 1: MotionAP ablation on MultiSports [3]. We investigate the effect of different feature aggregation modules using frame MotionAP to asses the quality of motion-wise action detection. Aggregating features across tracks, instead of cuboids, improves action detection performance across all categories, with a particularly noticeable improvement for large motions.

Method	f-mAP @0.5	MotionAP Large	Medium	Small
Speed-IoU measured as mean over time scales [4,8,16,24,26]				
Baseline	49.6	63.2	77.7	82.4
Baseline + track*	50.6	64.6	78.7	84.4
TAAD +MaxPool	53.9	70.2	**83.4**	86.1
TAAD +ASPP	54.4	**71.1**	**83.4**	86.9
TAAD +TCN	**55.3**	70.4	83.3	**87.3**
Speed-IoU measured at time scales of 16 frames				
Baseline	49.6	62.7	78.8	81.8
Baseline + track*	50.6	64.2	79.8	83.7
TAAD +MaxPool	53.9	69.9	83.9	85.9
TAAD +ASPP	54.4	**70.8**	84.3	86.2
TAAD +TCN	**55.3**	70.0	**84.5**	**86.4**
Speed-IoU measured at time scales of 24 frame				
Baseline	49.6	61.9	78.7	82.9
Baseline + track*	50.6	63.3	79.7	85.0
TAAD +MaxPool	53.9	68.6	**83.9**	87.3
TAAD +ASPP	54.4	**69.6**	83.5	**88.3**
TAAD +TCN	**55.3**	69.0	83.7	**88.3**

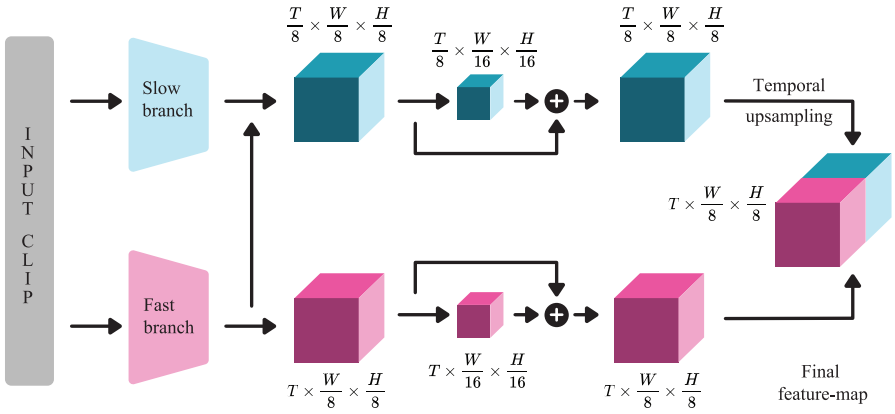


Figure 1: Single backbone with a single spatial upsample/downward step from res_5 to res_4. We add a single feature pyramid network (FPN) block to increase the spatial resolution, because the average size (26×54) of bounding boxes is very small, compared to the size (256×256) of the input image fed to network, *e.g.* when using MultiSports data.

```
{
(tube_temporal_pool): AdaptiveMaxPool1d(output_size=1)
}
```

Listing 1: MaxPool module with input feature of size $T \times C$ and output of $1 \times C$.

Table 2: MotionAP ablation on UCF24 [6]. We investigate the effect of different feature aggregation modules using frame MotionAP to asses the quality of motion-wise action detection. Aggregating features across tracks, instead of cuboids, improves action detection performance across all categories, with a particularly noticeable improvement for large motions.

Method	f-mAP @0.5	MotionAP Large	MotionAP Medium	MotionAP Small
Speed-IoU measured as mean over time scales [4,8,16,24,26]]				
Baseline	75.9	78.8	86.5	88.5
Baseline + track*	78.3	81.4	87.8	88.4
TAAD +TCN	**81.5**	**82.5**	**88.7**	**90.0**
Speed-IoU measured at time scales of 16 frames				
Baseline	75.9	79.0	86.7	88.2
Baseline + track*	78.3	81.4	88.3	87.8
TAAD +TCN	**81.5**	**82.7**	**89.0**	**89.5**
Speed-IoU measured at time scales of 24 frame				
Baseline	75.9	79.0	86.3	88.7
Baseline + track*	78.3	80.9	87.9	88.6
TAAD +TCN	**81.5**	**82.1**	**88.8**	**90.2**

Table 3: Motion-mAP ablation on UCF24 [6]. We investigate the effect of different feature aggregation modules using frame Motion-mAP to asses the quality of motion-wise action detection. Aggregating features across tracks, instead of cuboids, improves action detection performance across all categories, with a particularly noticeable improvement for large motions.

Method	f-mAP @0.5	Motion-mAP Large	Motion-mAP Medium	Motion-mAP Small
Speed-IoU measured as mean over time scales [4,8,16,24,26]				
Baseline	75.9	67.0	77.3	70.6
Baseline + track*	78.3	68.6	79.0	72.1
TAAD +TCN	**81.5**	**74.9**	**83.7**	**75.1**
Speed-IoU measured at time scales of 16 frames				
Baseline	75.9	68.9	78.6	72.1
Baseline + track*	78.3	70.6	80.0	73.5
TAAD +TCN	**81.5**	**76.1**	**84.2**	**78.1**
Speed-IoU measured at time scales of 24 frame				
Baseline	75.9	67.4	78.3	72.1
Baseline + track*	78.3	68.4	79.7	74.7
TAAD +TCN	**81.5**	**73.6**	**83.9**	**79.1**

or zero mAP, since medium is middle motion category it has more classes with some instances with medium motion class, hence fewer classes with zero MotionAP resulting in higher mean-AP i.e. Motion-mAP for medium motion type.

```
{
(TCN): Conv1d(576, 576, kernel_size=3,
        stride=1, padding=2, dilation=2)
(tube_temporal_pool): AdaptiveMaxPool1d(output_size=1)
}
```

Listing 2: TCN module with input feature of size $T \times C$ with output of $1 \times C$.

Table 4: Motion-mAP ablation on MultiSports [3]. We investigate the effect of different feature aggregation modules using frame Motion-mAP to asses the quality of motion-wise action detection. Aggregating features across tracks, instead of cuboids, improves action detection performance across all categories, with a particularly noticeable improvement for large motions. In "Baseline+track*", the track boxes are scored using baseline, with tracks acting as a false-positive filtering mechanism.

Method	f-mAP @0.5	Motion-mAP Large	Medium	Small
Speed-IoU measured as mean over time scales [4,8,16,24,26]				
Baseline	49.6	36.5	49.5	54.9
Baseline + track*	50.6	39.7	50.1	56.3
TAAD + MaxPool	53.9	43.8	52.7	57.7
TAAD + ASPP	54.4	44.2	52.9	58.4
TAAD + TCN	**55.3**	**44.9**	**53.4**	**60.4**
Speed-IoU measured at time scales of 16 frames				
Baseline	49.6	36.4	51.7	52.8
Baseline + track*	50.6	39.5	52.5	55.3
TAAD +MaxPool	53.9	42.4	54.4	56.4
TAAD +ASPP	54.4	**43.7**	54.2	56.3
TAAD +TCN	**55.3**	43.2	**55.6**	**58.9**
Speed-IoU measured at time scales of 24 frame				
Baseline	49.6	32.4	51.4	55.8
Baseline + track*	50.6	35.7	51.6	57.5
TAAD +MaxPool	53.9	38.4	54.2	58.7
TAAD +ASPP	54.4	39.0	53.9	59.2
TAAD +TCN	**55.3**	**39.2**	**54.7**	**61.0**

4. Motion-wise visual results

In this section we show visual results obtained using our baseline and TAAD model. We discuss some interesting observation in the caption of figures. The figures are best viewed in colour. The qualitative results contain the following scenarios:

(1) Large-motion due to fast execution of actions in Fig. 2.

(2) Large-motion due to fast camera motion in Fig. 3.

(3) Medium-motion action instances in Fig. 4.

(4) Small-motion action instance in Fig. 5.

(5) An action instance where TAAD fails due to tracking error shown in Fig. 6.

In all the captions, "Overlap" denotes the spatiotemporal overlap of the detected tube with the ground-truth tube, as defined by Weinzaepfel *et al*. [7]. Ground-truth boxes and frames (dot at the bottom of the frame) are shown in green colour, while the detected track is shown in red colour. We use "baseline+tracks" in these figure as "baseline" method. Since all the methods use same set of tracks, red box is used to annotate track boxes. Each method's score has a separate colour, described in the sub-caption.

References

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, 2018.

[2] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 6202–6211, 2019.

[3] Yixuan Li, Lei Chen, Runyu He, Zhenzhi Wang, Gangshan Wu, and Limin Wang. MultiSports: A Multi-Person Video Dataset of Spatio-Temporally Localized Sports Actions. In *International Conference on Computer Vision (ICCV)*, pages 13536–13545, 2021.

[4] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. *Computer Vision and Pattern Recognition (CVPR)*, 2022.

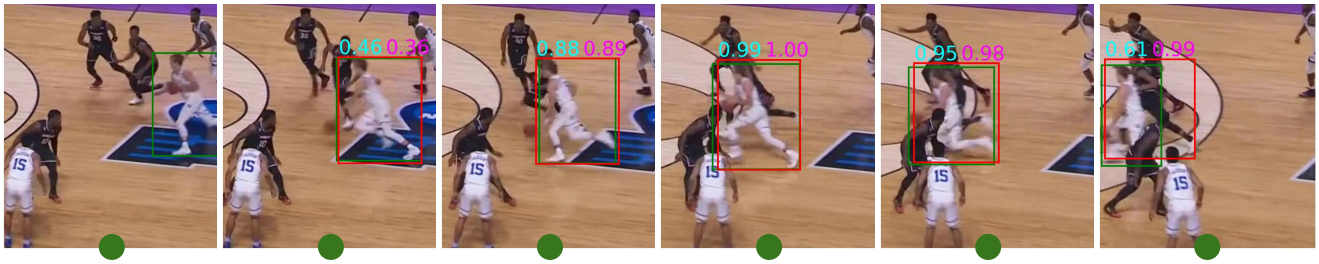
[5] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022.

[6] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild, 2012.

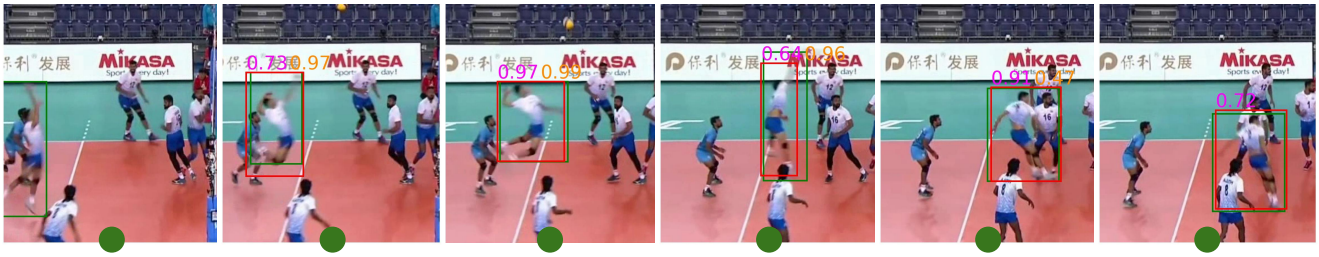
[7] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *International Conference on Computer Vision (ICCV)*, pages 3164–3172, 2015.

```
{
(ASPP): ASPP1D(
    (convs): ModuleList(
      (0): Conv1d(576, 256, kernel_size=1,
                  stride=1)
      (1): Sequential(
        (0): Conv1d(256, 576, kernel_size=1,
                    stride=1)
        (1): ReLU()
      )
      2: ASPPConv1D(
        (0): Conv1d(256, 576, kernel_size=3,
            stride=1, padding=1)
        (1): ReLU()
      )
      (3): ASPPConv1D(
        (0): Conv1d(256, 576, kernel_size=3,
            stride=1, padding=3, dilation=3)
        (1): ReLU()
      )
      (4): ASPPConv1D(
        (0): Conv1d(256, 576, kernel_size=3,
            stride=1, padding=(5), dilation=5)
        (1): ReLU()
      )
      (5): ASPPPooling1D(
        (0): AdaptiveAvgPool1d(output_size=1)
        (1): Conv1d(256, 576, kernel_size=1,
            stride=1)
        2: ReLU()
      )
    )
    (project): Sequential(
      (0): Conv1d(2880, 576, kernel_size=1,
                  stride=1, bias=False)
      (1): ReLU()
      )
)

(tube_temporal_pool): AdaptiveMaxPool1d(output_size=1)

}
```

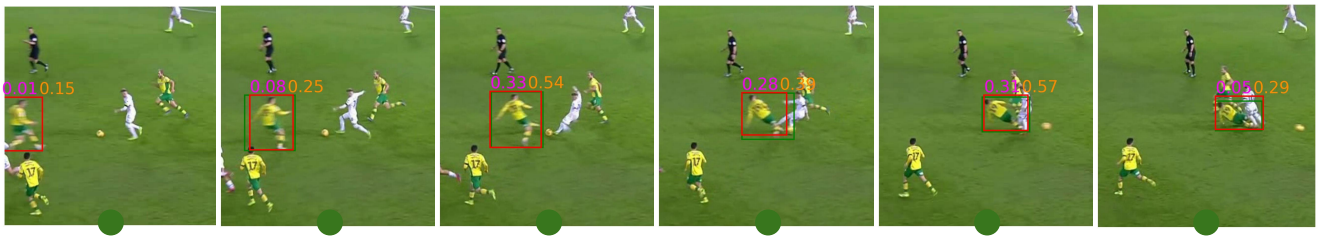
Listing 3: 1D-ASPP [1] module with input feature of size $T \times C$ with output of $1 \times C$.



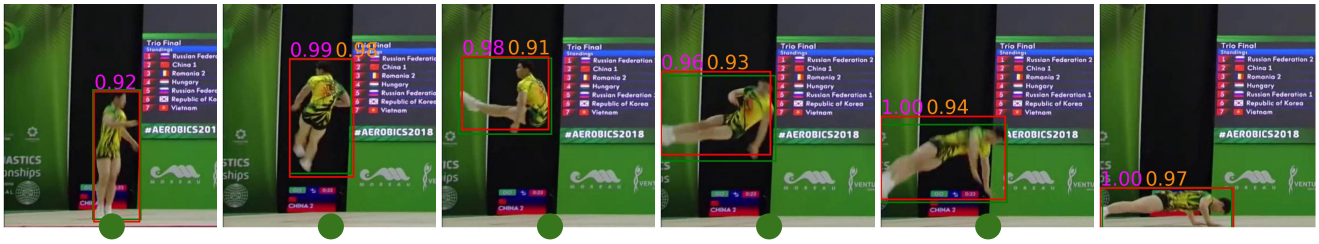
(a) Basketball-drive: Large-motion: Speed 0.05 IoU; Overlap: Baseline 70%, ASPP 67%



(b) Volleyball-spike: Large-motion: Speed 0.05 IoU; Overlap: ASPP 73%, TCN 72%



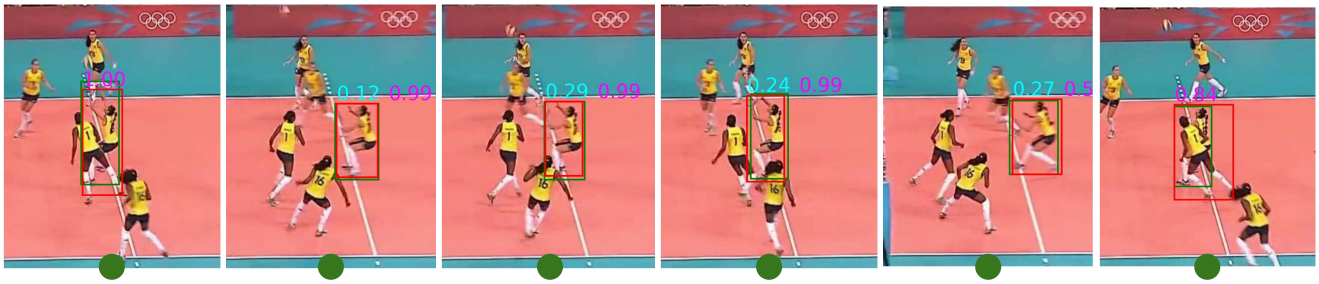
(c) Football-steal: Large-motion: Speed 0.03 IoU; Overlap: ASPP 77%, TCN 77%



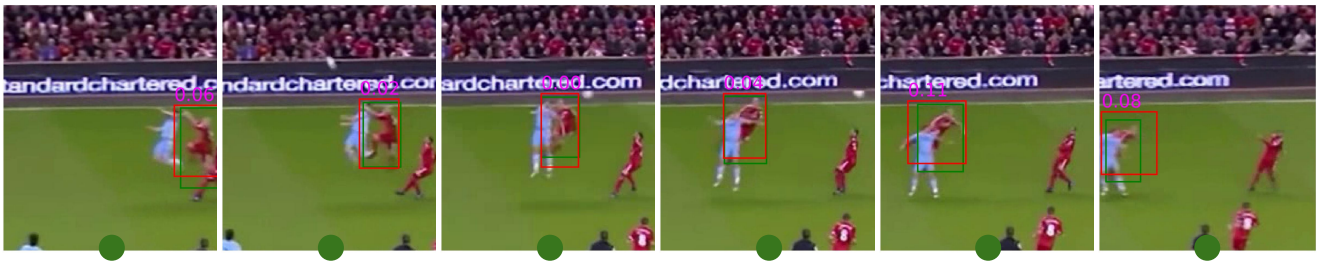
(d) Aerobic-pike-jump: Large-motion: Speed 0.17 IoU; Overlap: ASPP 71%, TCN 67%

Figure 2: Large-motion due to fast action; (a) TCN fail to detect it and others fail to detect initial few frames. (b) Volley-spike instance detected with high overlap both by ASPP and TCN. Similarly, baseline fails to detect fast action instances in (c) and (d). (d) connect back to instances shown in Fig 2 (c) in the introduction Section of main paper.

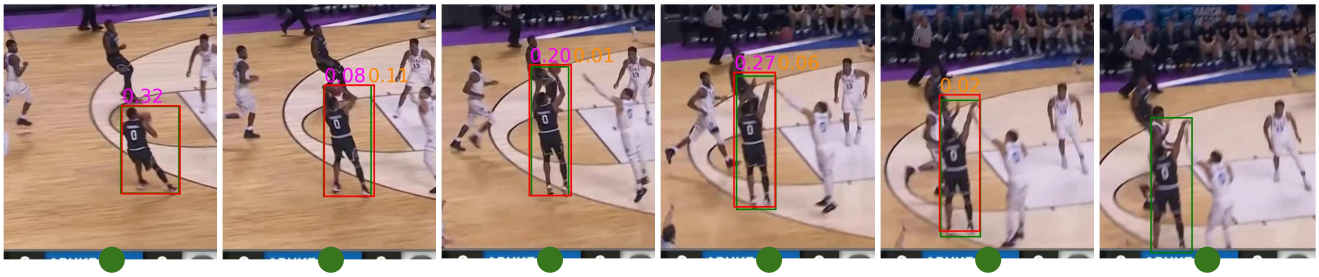
(a) Volleyball-serve: Large-motion: Speed 0.17 IoU; Overlap: Baseline 79%, ASPP 79%, TCN 79 %



(b) Volleyball-defend: Large-motion: Speed 0.10 IoU; Overlap: Baseline 54%, ASPP 65%

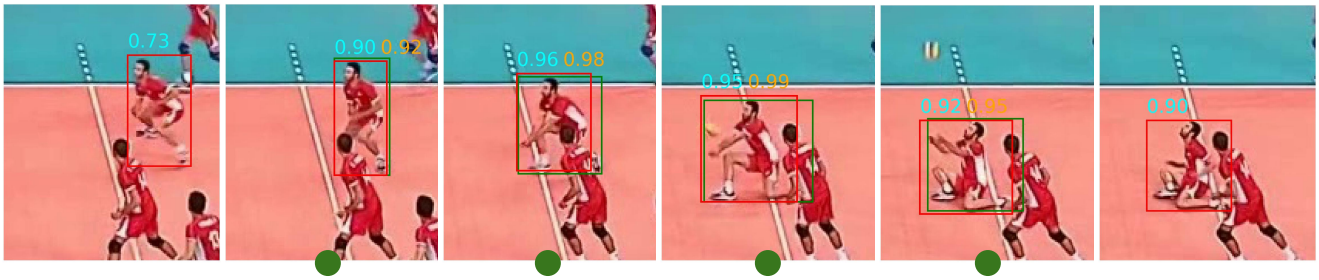


(c) Football-aerial-duels: Large-motion: Speed 0.00 IoU; Overlap: ASPP 67%

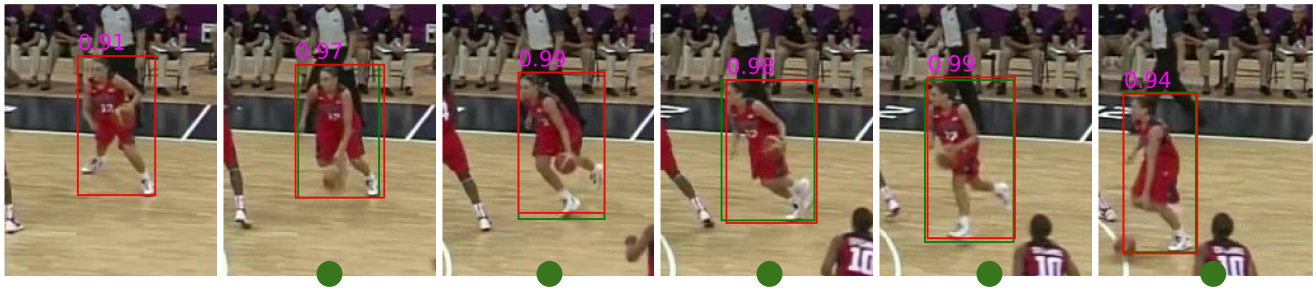


(d) Basketball-3-point-shot: Large-motion: Speed 0.07 IoU; Overlap: ASPP 68%, TCN 57 %

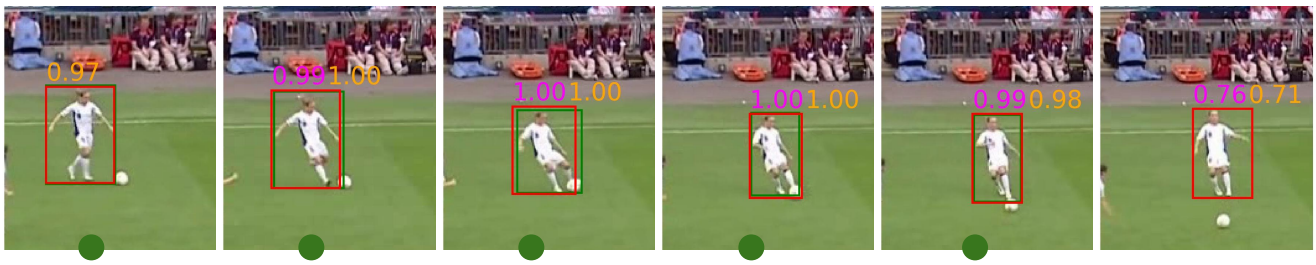
Figure 3: Large-motion due to camera motion; (a) shows an instance of "Volleyball-serve" which is correctly detected by all three methods, it happens quite often. In contrast, ASPP is better at detection large motion instances as shown in (b), (c) and (d).



(a) Volleyball-defend: Medium-motion: Speed 0.26 IoU; Overlap: Baseline 53%, TCN 63%



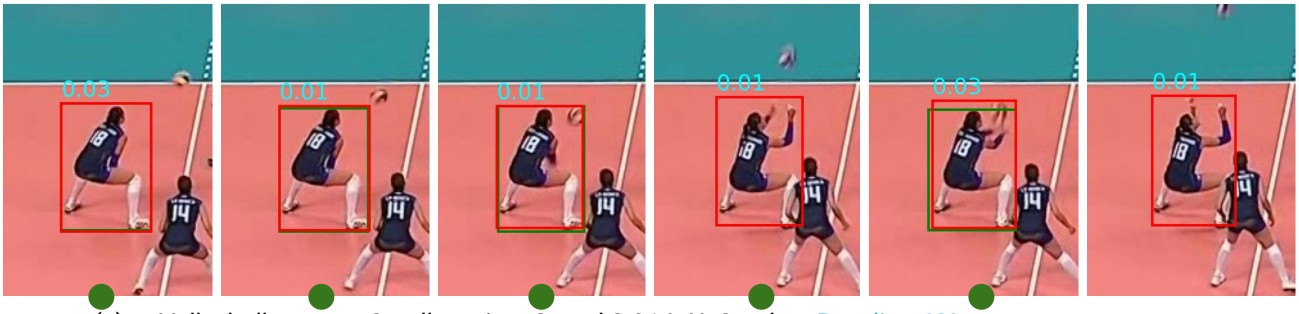
(b) Basketball-dribble: Medium-motion: Speed 0.29 IoU; Overlap: ASPP 90%



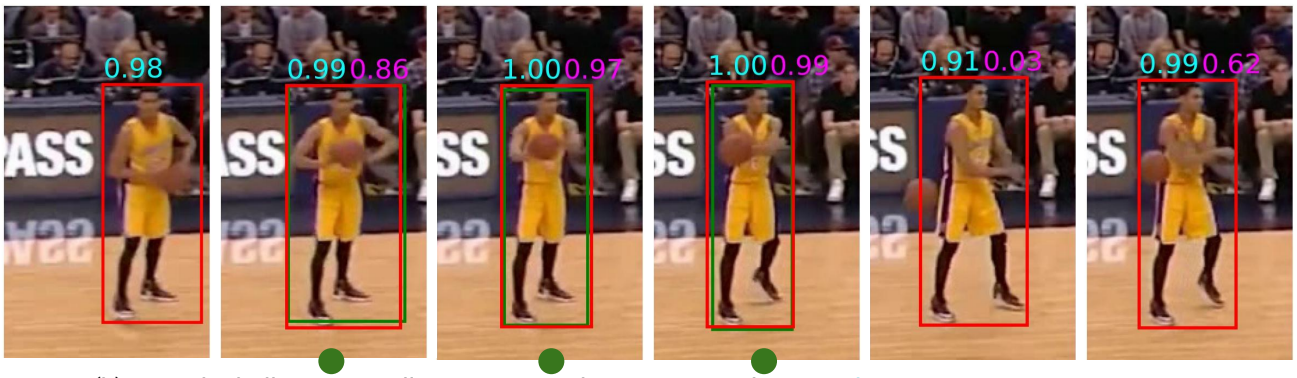
(c) Football-short-pass: Medium-motion: Speed 0.23 IoU; Overlap: ASPP 53%, TCN 58%

(d) Aerobic-straddle: Medium-motion: Speed 0.43 IoU; Overlap: ASPP 55%, TCN 63%

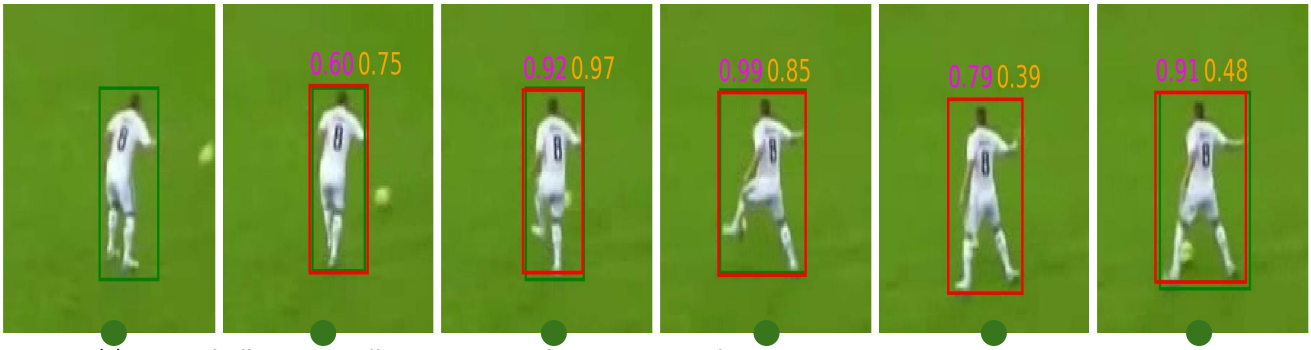
Figure 4: Medium-motion; all these instances show where temporal detection bounding is longer than ground truth tube, which happen often for medium-motion instances. Accuracy of temporal boundary detection is directly proportional to stability in continuous scores of consecutive boxes in a track, where ASPP seems to be better in these examples as a result having higher overlap in (b) and (c).



(a) Volleyball-protect: Small-motion: Speed 0.91 IoU; Overlap: Baseline 62%

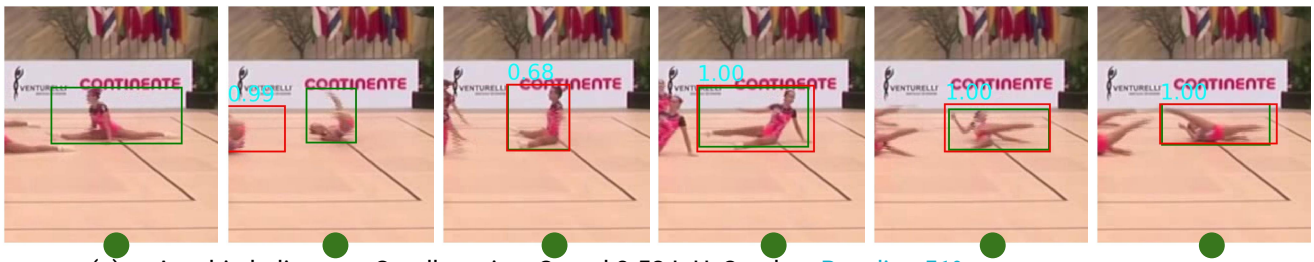


(b) Basketball-pass: Small-motion: Speed 0.51 IoU; Overlap: Baseline 51%, ASPP: 61%



(c) Football-trap: Small-motion: Speed 0.59 IoU; Overlap: ASPP: 74%, TCN: 74%

Figure 5: Small-motion; SubFig. (a) show instances where baseline is able to detect an instance of "Volleyball-protect", where others not, even though the confidence of the detection is very low for the baseline. (b) shows instances where TAAD +TCN fails to detect it. (a) shows an action instance where baseline fails, although other methods are able to detect it with high overlap but not in initial few frames.



(a) Aerobic-helicopter: Small-motion: Speed 0.52 IoU; Overlap: Baseline 51%

Figure 6: Tracking-error; This figures show that if tracking fails, *e.g.* in the first two frames of this action instance, TAAD fails to detect such instances.