Spatio-Temporal Action Detection Under Large Motion

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

Abstract

Current methods for spatio-temporal action tube detection often extend a bounding box proposal at a given keyframe into a 3D temporal cuboid and pool features from nearby frames. However, such pooling fails to accumulate meaningful spatio-temporal features if the position or shape of the actor shows large 2D motion and variability through the frames, due to large camera motion, large actor shape deformation, fast actor action and so on. In this work, we aim to study the performance of cuboid-aware feature aggregation in action detection under large action. Further, we propose to enhance actor feature representation under large motion by tracking actors and performing temporal feature aggregation along the respective tracks. We define the actor motion with intersection-over-union (IoU) between the boxes of action tubes/tracks at various fixed time scales. The action having a large motion would result in lower IoU over time, and slower actions would maintain higher IoU. We find that track-aware feature aggregation consistently achieves a large improvement in action detection performance, especially for actions under large motion compared to cuboid-aware baseline. As a result, we also report state-of-the-art on the large-scale MultiSports dataset.

1. Introduction

Spatio-temporal action detection, which classifies and localises actions in space and time, is gaining attention, thanks to the AVA [15] and UCF24 [40] datasets. However, most of the current state-of-the-art works [12, 21, 28, 37, 53] focus on pushing action detection performance usually by complex context modelling [28, 41, 53], larger backbone networks [11, 22, 25], or by incorporating an optical flow [37, 52] stream. The above methods use cuboid-aware temporal pooling for feature aggregation. In this work, we aim to study cuboid-aware action detection under varying degrees of action instance motion using the MultiSports [20] dataset which contains instances with large motions, unlike AVA [15] as shown in Fig. 1.

Large object motion can occur for various reasons, e.g.,

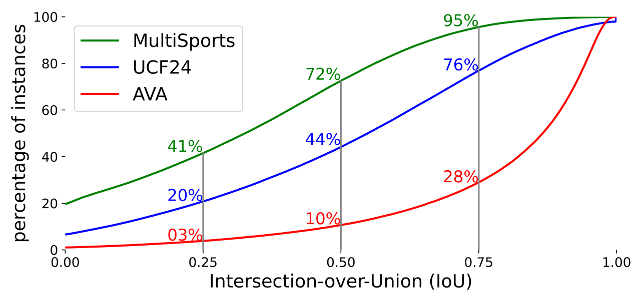


Figure 1: Cumulative density function of IoU measurements for ground-truth bounding box pairs taken one second apart in the training sets of AVA, UCF24, and MultiSports, plotted as percentage of instances falling in cumulative bins shown on the Y-axis. For example, 20% of MultiSports instances has an IoU less than or equal to 0.0 signifying that 20% of instances has very large motion present. In contrast, only 10% of AVA instances has an IoU less than 0.5, meaning that 90% of its instances have a large overlap after one second, *i.e.* large amount instance has small motion.

fast camera motion, fast action, body shape deformation due to pose change, or mixed camera and action motions. These reasons are depicted in Fig. 2. Furthermore, the speed of motions within an action class can vary because of a mixture of the above reasons and the nature of the action type, e.g., pose based or interaction based action. Either of these reasons can cause sub-optimal feature aggregation and lead to errors in action classification of a given reason.

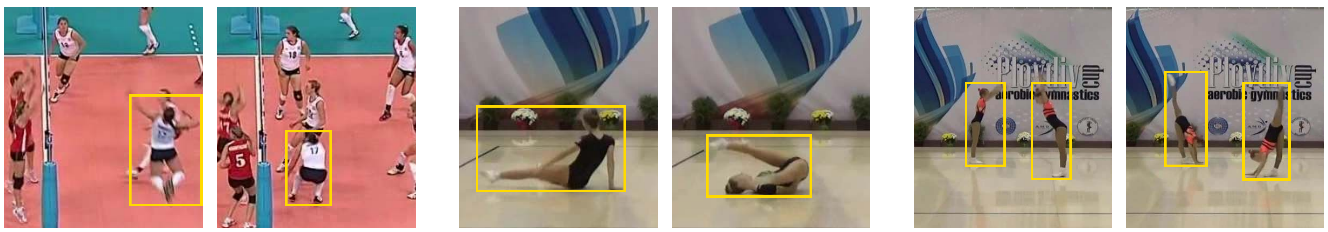
We propose to split actions into three categories: Large-motion, medium-motion, and small-motion, as shown in Figs. 1 and 3. The distinction is based on the IoU of boxes of the same actor over time, which we can compute using the ground truth tubes of the actors. We propose to study the performance on different motion categories of a baseline cuboid-aware method, without further bells and whistles like context features [27, 28, 41] or long-term features [41, 47], because large-motion happens quickly in a small time window, as seen in Fig. 1 and 2. In large-motion cases the IoU would be small (Fig. 3 (a)), and as a result a 3D

(a) Football block
(Large camera motion)

(b) Basketball drive
(Camera motion + fast action)

(c) Aerobic pike jump
(Fast action)

Figure 2: Reasons for large motions: (a) large camera motion (b) camera motion plus actor motion (c) static camera but super fast action. Note that, (b) shows camera zoom out and translation at the same time, and (c) shows Pike-jump action which involves jumping from standing position to air while bringing head and knee close to each other, then lending in horizontal shape on the ground, all this in close to one second. All these images contain pairs of boxes of the same actor, separated by one second in a tube with 0.0 IoU.



(a) Large movement
IoU: 0.00

(b) Medium movement
IoU: 0.44

(c) Small movement
IoU: 0.85

Figure 3: Varying degrees of motion observed for actors in bounding boxes with one second time window with mostly static camera: (a) large motion where actor performs spiking action, results in IoU of 0.0, meaning large-motion. (b) some pose change as resulting in 0.44 IoU, meaning medium-motion. (c) change in body pose at same location with IoU being close to 0.85, i.e. small-motion.

cuboid-aware feature extractor will not be able to capture features centred on the actor's location throughout the action. To handle the large-motion case, we propose to track the actor over time and extract features using Track-of-Interest Align (TOI-Align); resulting in Track Aware Action Detector (TAAD). Further, we study different types of feature aggregation modules on top of TOI-Aligned features for our proposed TAAD network, shown in Fig. 4.

To this end, we make the following contributions: (a) we are the first to study large-motion action detection systematically, using evaluation metrics for each type of motion, similar to object detection studies on MS COCO [23] based on object sizes. (b) we propose to use tube/track-aware feature aggregation modules to handle large motions, and we show that this type of module helps in achieving great improvements over the baseline, especially for instances with such large motion. (c) in the process, we set a new state-of-the-art for the MultiSports dataset by beating last year's challenge winner by a substantial margin.

2. Related Work

Action recognition [4, 11, 12, 22, 25, 34, 44, 46] models provide strong video representation models. However, action recognition as a problem is not as rich as action detection, where local motion in the video needs to be understood more precisely. Thus, action detection is the more relevant problem for understanding actions under large motion.

We are particularly interested in the spatio-temporal *action detection* problem [13, 14, 15, 47, 53], where an action instance is defined as a set of linked bounding boxes over time, called action tube. Recent advancements in online action detection [1, 18, 21, 37, 39, 49] lead to performance levels very competitive with (generally more accurate) offline action detection methods [15, 29, 31, 32, 35, 36, 43, 44, 52] on the UCF24 [40] dataset.

UCF24 has been a major benchmark for spatio-temporal action detection (i.e. action tube detection), rather than AVA [15]. The former is well suited for action tube detection research, as it provides dense action tube annotations, where every frame of the untrimmed videos is annotated (unlike AVA [15], in which videos are only annotated at one frame per second). More recently, Li *et al.* [20] proposed the MultiSports dataset, which resolves two main problems with the UCF24 dataset. Firstly, it has more fine-grained action classes. Secondly, it has multiple actors performing multiple types of action in the same video. As a result, the MultiSports dataset is comparable to AVA in terms of diversity and scale. Moreover, the MultiSports dataset is densely annotated, every frame at a rate of 25 frames per second, which makes it ideal to understand action under large motion, as shown in Fig. 1.

At the same time, there have been many interesting papers [6, 11, 12, 28, 41] that focus on keyframe based action detection on AVA [15]. AVA has been helpful in pushing action detection research on three fronts. Firstly, backbone model representations are much better now thanks to works like [6, 11, 12, 25, 44]. Secondly, long-term feature banks (LBF) [47] came to the fore [28, 41, 51], capturing some temporal context, but without temporal associations between actors. Thirdly, interactions between actors and object have been studied [27, 28, 41, 51]. Once again, the problem we want to study is action detection under large motion, which happens quickly at a small temporal scale. All the above methods use cuboid-aware pooling for local feature aggregation, which - as we will show - is not ideal when the motion is quick and large. As a result, we borrow the SlowFast [12] network as the baseline network for its simplicity and spatio-temporal representational power. Also, it has been used for MultiSports [20] as baseline and in many other works on UCF24 as a basic building block.

The work of Weinzaepfel *et al*. [45] is the first to use tracking for action detection. That said, their goal was different than ours. They used a tracker to solve the linking problem in the tube generation part, where action classification was done on a frame-by-frame basis given the bounding box proposals from tracks. We, on the other hand, propose action detection by pooling features from within entire tracks. Gabriellav2 [8] is another method that makes use of tracking to solve the problem of temporal detection of co-occurring activities, but it relies on background subtraction which would fail in challenging in-the wild videos. Singh *et al*. [36], Li *et al*. [21] and Zhao *et al*. [53] are the only works generating flexible micro-tube proposals without the help of tracking. However, these approaches are limited to a few frames (2-10). Without the possibility to scale to larger time windows of 1-2 seconds as required for multi-frame tube anchors/query to regress box coordinates on a large number of frames, performance drops after a few frames.

3. Methodology

In this section, we describe the proposed method to handle actions with large motions, which we call Track Aware Action Detector (TAAD). We start by tracking actors in the video, using a tracker described in Section 3.2. At the same time, we use a neural network designed for video recognition, SlowFast [12], to extract features from each clip. Using the track boxes and video features, we pool per-frame features with a RoI-Align operation [16]. Afterwards, a Temporal Feature Aggregation (TFA) module receives the per-track features and computes a single feature vector, from which a classifier predicts the final action label. Figure 4 illustrates each step of our proposed approach.

3.1. Baseline Action Detector

We select a SlowFast [12] network as our video backbone. The first reason for this choice is that its performance is still competitive to larger scale transformer models, such as VideoSwin [25] or MViT [10, 22], on the task of spatio-temporal action detection. Furthermore, SlowFast is computationally more efficient than the transformer alternatives, with a cost of 65.7 GFLOPS compared to 88, at least, and 170 for VideoSwin [25] and MViT [10] respectively, and offers features at two different temporal scales. Having different temporal scales is important, especially since we aim at handling fast and/or large motions, where a smaller scale is necessary. Finally, SlowFast is the default backbone network of choice for the MultiSports and UCF24 datasets, which are the main benchmarks in this work, facilitating comparisons with existing work.

We implement our baseline using pySlowFast [9] with a ResNet-50 [17] based SlowFast [12] architecture, building upon the works of Feichtenhofer *et al*. [12] and Li *et al*. [20]. First, we add background frames (+bg-frames), i.e. frames erroneously detected by our detector, YOLOv5, as extra negative samples for training the action detector. Next, we replace the multi-label with a multiclass classifier, switching from a binary cross entropy per class to a cross entropy loss (CE-loss). Finally, we also added a downward FPN block (see Sup.Mat. for details). Through these changes, we aimed to build the strongest possible baseline.

3.2. Tracker

We employ a class agnostic version of YOLOv5-DeepSort [2] as our tracker, which is based on YOLOv5 [30, 42] and TorchReID[54]. We fine-tune the medium size version of YOLOv5 as the detection model for 'person' classes. A pretrained OsNet-x0-25 [55] is used as re-identification (ReID) model. As we will show in the experiment section, a tracker with high recall, *i.e*. small number of missing associations, is key for improving performance action tube detection. We will also show that fine-tuning the detector is a necessary step, particularly for UCF24, where the quality and resolution of the videos is small.

The tracker can also be used as bounding box proposal filtering module. Sometimes, the detector produces multiple high scoring detections which are spurious and lead to false positives but these detections do not match to any of

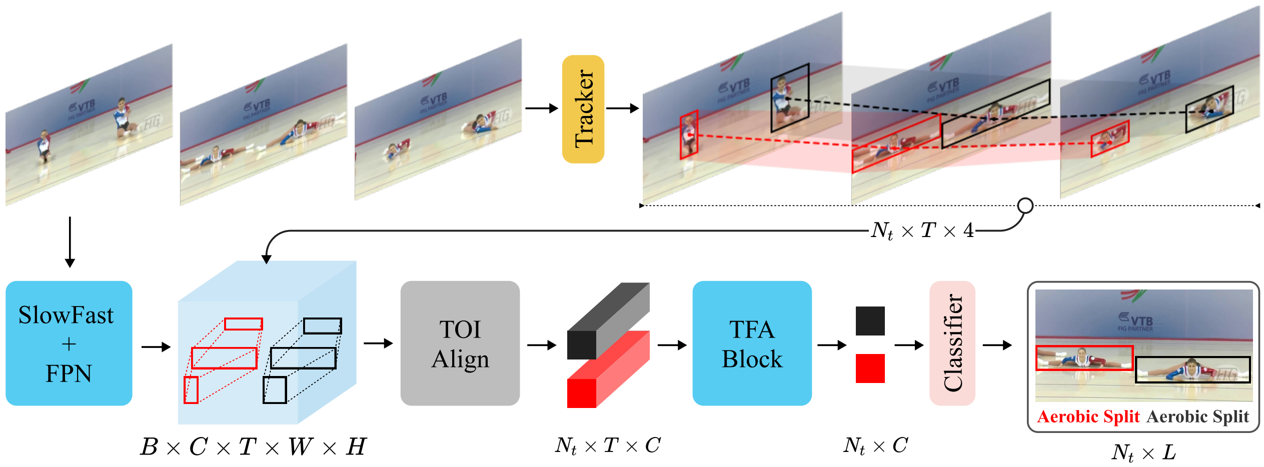


Figure 4: Proposed Track Aware Action Detector (TAAD): Given an input clip with T frames, we extract features using a video recognition network [12] and N_t per-actor tracks from a tracker. The TOI-Align operation extracts per-track features from the entire video sequence, using an RoI-Align operation and the track boxes, returning a $N_t \times T \times C$ feature array. Next, the Temporal Feature Aggregation (TFA) module aggregates the features along the temporal dimension and passes the resulting $N_t \times C$ array to the action classifier that predicts the action label.

the tracks being generated because they are not temporally consistent. The proposals generated by tracks can be used with the baseline methods at test-time. This helps improve the performance of the baseline method.

3.3. Temporal Feature Aggregation

Track-of-Interest Align (TOI-Align): The SlowFast video backbone processes the input clip and produces a $T \times H \times W$ feature tensor, while our tracker returns an array with size $N_t \times T \times 4$ that contains the boxes around the subjects. An RoI-Align [16] takes these two arrays as input and produces a feature array of size $N_t \times T \times H \times W$, *i.e.* one feature tube per track. In case the length of the track is smaller than the length of the input clip, we replicate the last available bounding box in the temporal direction, which occurs around 3% of input clips in MultiSports dataset.

Feature aggregation: In order to predict the label of a bounding box in a key-frame, we need to aggregate features across space and time. First we apply average pooling in spatial dimensions on features extracted by TOI-Align, then the Temporal Feature Aggregation role is performed by one of the following variants considered:

1. Max-pooling over the temporal axes (MaxPool).
2. A sequence of temporal convolutions (TCN).
3. A temporal variant of Atrous Spatial Pyramid Pooling (ASPP) [5]. We modify Detectron2's [48] ASPP implementation, replacing 2D with 1D convolutions.

We also tried a temporal version of ConvNeXt [24] and VideoSwin [25] blocks, however these resulted in unstable training, even with the tunning of learning rates and other hyperparameters. In our experiments, we only used one layer of temporal convolution for our TCN module, adding more layers did not help. See the **Sup. Mat.** for more details.

3.4. Tube Construction

Video-level tube detection requires the construction of action tubes from per-frame detections. This process is split into two steps [32]. The first links the proposals to form tube hypotheses (i.e. action tracks). The second trims these hypotheses to the part where there is an action. One can think of these two steps as a tracking step plus a temporal (start and end time) action detection step. The majority of the existing action tube detection methods [20, 21, 33, 36] use a greedy proposal linking algorithm first proposed by in [18, 37] for the first step. For the baseline approach, we use the same method for the tube linking process from [37]. Since for our method (TAAD) we already have tracks, the linking step is already complete. The temporal trimming of action tracks is performed using label smoothing optimisation [32], which is used by many previous works [18, 21]. In particular, we use class-wise the temporal trimming implementation provided by [37].

3.5. Datasets

We evaluate our idea on two densely annotated datasets (MultiSports [20] and UCF24 [40]) with frame and tube level evaluation metrics for actions detection, unlike AVA [15], which is sparsely annotated and mostly used for frame level action detection.

MultiSports [20] is built using 4 sports categories, collecting 3200 video clips annotated at 25 FPS, and annotating 37701 action tube instances with 902k bounding boxes. Although it contains 66 action classes, we follow the official evaluation protocol [1] that uses 60 classes. Due to the fine granularity of the action labels the length of each action segment in a clip is short, with an average tube length of 24 frames, equal to one second, while the average video length is 750 frames. Each video is annotated with multiple instances of multiple action classes, with well defined temporal boundaries. MultiSports contains action instances with large motions around actors, as shown in Fig. 1

UCF24 [40] consists of 3207 videos annotated at 25 FPS with 24 action classes from different sports, 4458 action tube instances with 560K bounding boxes. Videos are untrimmed, with an average video length of 170 frames and average action tube length of 120 frames. The disadvantages of UCF24 are (1) the presence of only one action class per video and (2) the low image quality, due to compression and the small resolution, namely 320×240 pixels. Even though UCF24 has less diversity, less motion, fewer classes and more labelling noise compared to MultiSports, it is still useful to evaluate action detection performance, thanks to its temporally dense annotations.

3.6. Implementation Details

We use 32 frames as input with sampling rate of 2, which means more 2 seconds of video clip. We use Slowfast-R50-8×8 [12], meaning speed ratio $\alpha = 8$ and channel ratio $\beta = 1/8$. We use stochastic gradient descent (SGD) to optimise the weights, with a learning rate of 0.05 and batch size of 32 on 4 GPUs. We use 1 epoch to warm up the learning rate linearly, followed by a cosine learning rate schedule [26], with a final learning rate of 0.0005, for a total of 5 epochs. Note that we only train for 3 epochs on UCF24. All our networks are trained with a batch size equal to 32 on 4 Titan X GPUs. We use the frame-level proposal released by [20] for MultiSports, for the fairness of comparison. More details can be found in **Sup. Mat.**.

4. Experiments

In this section, we evaluate our TAAD method along with TFA modules on the MultiSports and UCF24 datasets. We start by defining the metrics used in Sec. 4.1 and motion category classification in Sec. 4.2. Firstly, we study the impact of different TFA modules under different motion conditions in Sec. 4.3. Secondly, we compare our TAAD method with state-of-the-art methods in Sec. 4.4. Later, we discuss the baseline model and the impact the tracker has in Sec. 4.5. We finish with a discussion in section Sec. 4.6.

[1]https://github.com/MCG-NJU/MultiSports/

4.1. Metrics

We report metrics that measure our detector's performance both at frame- and video-level, computing frame and video mean Average Precision (mAP), denoted as f-mAP and v-mAP respectively. These metrics are common in action detection works [18, 21, 45]. A detection is correct if and only if its Intersection-over-Union(IoU) with a ground-truth box or tube, for frame and video metrics respectively, is larger than a given threshold (e.g. 0.5) and the predicted label matches the ground-truth one. From this, we compute the Average Precision (AP) for each class and the mean across classes, to get the desired mAP metric. Tube overlap is measured by spatio-temporal-IoU proposed by [45], similar to [20], we use the ACT[2] evaluation code.

4.2. Motion categories

We split actions into three motion categories: large, medium and small. Computing per-motion-category metrics requires labelling the ground-truth action tubes. We start this process by computing the IoU between a pair of boxes separated by offsets equal to $[4, 8, 16, 24, 36]$ in sliding window fashion. We average these 5 IoU values and get the final IoU value as a measure of speed. We then split the dataset into three bins of equal size. We can then assign a 'large, medium, or small' motion label to each instance:

$$\text{MultiSports} = \begin{cases} \text{Large}, & \text{IoU} \in [0.00, 0.21] \\ \text{Medium}, & \text{IoU} \in [0.21, 0.51] \\ \text{Small}, & \text{IoU} \in [0.51, 1.00] \end{cases} \tag{1}$$

$$\text{UCF24} = \begin{cases} \text{Large}, & \text{IoU} \in [0.00, 0.49] \\ \text{Medium}, & \text{IoU} \in [0.49, 0.66] \\ \text{Small}, & \text{IoU} \in [0.66, 1.00] \end{cases} \tag{2}$$

Given these labels, we can compute AP metrics per motion category. There are two options for these metrics. The first is to compute the AP for large, medium and small motions per action class and then average across actions. We call this metric *Motion-mAP*. The alternative is to ignore action classes and compute the AP for large, medium and small motions, irrespective of the action, which we call *MotionAP*. This essentially measures action detection accuracy w.r.t. to motion speed, irrespective of class. We compute the metrics both on a per-frame and on a per-video level, following the two methods just described. Video metrics are denoted with a *video* prefix. We will release the code for training and testing our TAAD network along with evaluation scripts for both MotionAP and Motion-mAP.

[2]https://github.com/vkalogeiton/caffe/tree/act-detector

Table 1: Motion-wise ablation of Temporal Feature Aggregation modules. We investigate the effect of different feature aggregation modules using frame- and video-mAP to measure model performance, both with the classic definition and with our proposed motion categories. Aggregating features across tracks, instead of cuboids, improves action detection performance across all categories, with a particularly noticeable improvement for large motions. For example, the TCN module improves large motion Motion-mAP by 8.4, with an improvement of only 4.5 points for small motions.

Method	f-mAP@0.5	Motion-mAP@0.5 Large	Medium	Small	v-mAP@0.5	Video Motion-mAP@0.5 Large	Medium	Small
MultiSports [20]								
Baseline (SlowFastR50 [12])	49.6	36.5	49.5	54.9	31.2	14.2	33.6	45.1
Baseline + track†	50.6	39.7	50.1	56.3	33.0	15.4	34.7	45.7
TAAD + MaxPool	53.9	43.8	52.7	57.7	34.8	16.7	35.5	47.4
TAAD + ASPP	54.4	44.2	52.9	58.4	36.0	**18.8**	37.5	46.0
TAAD + TCN	**55.3**	**44.9**	**53.4**	**60.4**	**37.0**	17.9	**38.1**	**47.3**
UCF24 [40]								
Baseline (SlowFastR50 [12])	75.9	67.0	77.3	70.6	45.4	33.3	47.0	46.0
Baseline + track†	78.3	68.6	79.0	72.1	47.4	34.8	47.9	50.7
TAAD + TCN	**81.5**	**74.9**	**83.7**	**75.1**	**52.0**	**38.3**	**51.2**	**50.2**

† tracks used a filtering module at frame-level and tube construction module at video-level.

4.3. Motion-wise (main) results

As the main objective for this work, we first study how the cuboid-aware baseline compares against our track-based TAAD under significant motion. We compare different choices for temporal feature aggregation. In Tab. 1, we measure the frame- and video-motion-mAP, for models trained with different TFAs, on MultiSports and UCF24. Pooling features across tracks, instead of neighbouring frames, even with a relatively simple pooling strategy, *i.e*. Max-Pool over the spatio-temporal dimensions, results in stronger action detectors, with a 5.7 % and 5.8 % frame and video mAP boost on MultiSports. More involved feature aggregation strategies, such as the temporal convolution blocks (TCN) or ASPP variant, lead to further gains. Note that the biggest improvements on MultiSports occur in the large motion category, + 8.4 % Motion-mAP, with smaller gains in medium (+3.9%) and small (+5.5 %) motions.

Tab. 2 contains MotionAP results on MultiSports for different TFA module choices. It is clear that TAAD combined with any of the TFA modules leads to large performance gains. Larger motions benefit the most, followed by medium and small motions. For example, the ASPP module helps more with large motions ($+7.9$) than with small motions ($+4.5$). We observe the same trend in Tab. 1, both for frame and video Motion-mAP.

These results signify that there is a large gap between the performance for large vs. small motion action instances for the baseline method. The combination of TAAD with any of the TFA modules helps to reduce this discrepancy and improves the overall performance for both datasets.

Table 2: Motion-wise ablation with MotionAP metric. We investigate the effect of different TFA modules using frame-level MotionAP to asses the quality of motion-wise action detection in comparison to baseline on MultiSports dataset.

Method	MotionAP @0.5 Large	Medium	Small
Baseline	63.2	77.7	82.4
Baseline + track†	64.6(+1.5)	78.7(+1.0)	84.4(+2.0)
TAAD +MaxPool	70.2(+7.0)	83.4(**+5.7**)	86.1(+3.9)
TAAD +ASPP	71.1(**+7.9**)	83.4(**+5.7**)	86.9(+4.5)
TAAD +TCN	70.4(+7.2)	83.3(+5.6)	87.3(**+4.9**)

† tracks used as filtering module.

4.4. Comparison to the State-of-the-art

We compare our proposed detector with the state-of-the-art for MultiSports and UCF24, for both frame and tube level action detection, unlike approaches [28, 41] which solely focus on frame level evaluation. It is important to note that, similar to the baseline, TAAD does not make use of any spatial context. Hence, gains are made using track aware feature aggregation rather than by using other spatio-temporal context modelling modules [53].

We report frame and video mAP for different methods in Tab. 3, namely SlowFast variants from the original MultiSports paper, Ning *et al*.'s [27] Person-Context Cross-Attention Modelling network and our improved baseline, and three versions of our model, the one with MaxPool along the temporal dimensions, the ASPP variant and the temporal convolutional network (TCN). Tab. 3 contains the

Table 3: Comparison of action detection performance of the proposed method to our baseline model and other state-of-the-art methods on MultiSports dataset. TAAD combined with TFA modules leads to state-of-the-art detection performance.

Method	f-mAP	v-mAP		
	0.5	0.2	0.5	.1:.9
YOWO [20, 21]	25.2	12.9	9.7	–
MOC [20, 21]	25.2	12.9	9.7	–
SlowFast-R50 [12, 20]	27.7	24.2	9.7	–
SlowFast-R101 [27]	29.5	28.1	8.4	12.3
SlowFast-R101+PCCA [27]	42.2	41.0	20.0	20.9
Baseline (ours)	49.6	54.1	31.3	28.9
Baseline + tracks (ours) †	50.6	56.3	33.0	30.9
TAAD + MaxPool (ours)	53.9	58.6	34.8	32.4
TAAD + ASPP (ours)	54.4	59.2	36.0	33.0
TAAD + TCN (ours)	**55.3**	**60.6**	**37.0**	**33.7**

* evaluated using tracks at test time.

Table 4: Comparison of action detection performance (f-mAP and v-mAP) of the proposed method along with our baseline model and other SOTA methods on UCF24 dataset. TAAD with TCN shows performance gain compared to baseline, with competitive performance to other methods that are specifically designed for UCF24, including spatial context [28, 53] module and sophisticated transformer head used by [53]. TAAD is even better than some of the approach that us optical flow ("F") stream as input along with visual stream ("V").

Methods	Input	f-mAP	v-mAP		
			0.2	0.5	0.5:0.9
ROAD [37]	V+F	–	76.4	45.2	20.1
AMTnet [31]	V+F	–	78.5	49.7	24.0
ACT [18]	V+F	67.9	76.5	49.2	23.4
TACNet [38]	V+F	72.1	77.5	52.9	24.1
FlowDance [52]	V+F	–	78.5	50.3	24.5
I3D [15]	V+F	76.3	–	59.9	–
MOC [21]	V+F	78.0	82.8	53.8	28.3
TubeR [53]	V+F	81.3	**85.3**	**60.2**	**29.7**
YOWO [19]	V	78.0	75.8	48.8	–
TubeR [53] *	V	80.1	**82.8**	**57.7**	**28.7**
Baseline	V	75.8	76.7	45.5	19.7
Baseline + tracks †	V	78.8	77.4	47.4	20.2
TAAD +TCN	V	**81.5**	79.6	52.0	23.0

† evaluated using tracks at test time.

* TubeR uses large transformer head plus complex context modelling.

results of these experiments, where we clearly see the benefit of using tracks for action detection. The addition of feature pooling along tracks, even with the simpler MaxPool version, outperforms our improved baseline by 4.3 % frame mAP. Better temporal fusion strategies, *i.e.* ASPP and TCN, lead to further benefits. As a result, we set a new state-of-the-art for the MultiSports dataset. Note that all our TFA modules add less than 1M FLOPS ($< 2\%$) to the computation time of the whole network.

Table 5: Baseline progression on MultiSports dataset with proposals released by [20]. Adding more negatives proposals from non-action frames, in the form of proposals erroneously detected by the per-frame detector, converting the problem from multi-label to multiclass classification and adding a FPN leads to a more effective action detector.

Method	SlowFast[20]	SlowFast	+bgFrames	+CE-loss	+FPN
#keyframes	unknown	288K	354K	354K	354K
f-mAP@0.5	27.7	34.5	39.7	49.0	49.6

Finally, we compare our proposed TAAD model on the older UCF24 dataset in Tab. 4. Our model outperforms most existing methods, with the exception of TubeR [53] and MOC [21]. We think the reason is that TubeR uses a set prediction framework [3] with a transformer head (plus 3 layers for each encoder- and decoder-transformer) on top a CNN backbone (CSN-152). Moreover, they use actor context modelling similar to [28]. It is also important that I3D based TubeR needs $132M$ FLOPS, which is much higher than the $97M$ needed by SlowFastR5-TCN based TAAD. MOC uses flow stream as additional input and uses DLA-34[50] as backbone network. Note that our goal is to analyse and improve action detection performance across different actor motion speeds. Hence, we do not use any spatial attention or context modelling [53] between actors. These are certainly very interesting topics, orthogonal to our proposed approach. This said, our network consistently shows improvements in all metrics for both datasets when compared to our baseline.

Additionally, the low quality on UCF24 given by YOLOv5 also hampers performance. We report the corresponding YOLOv5 +DeepSort metrics in Tab. 6. Finetuning the detector on each dataset is a necessary step, especially on UCF24 where the video quality is worse than MultiSports.

4.5. Building a Strong Baseline on MultiSports

Here, we investigate the effect of our proposed changes on the performance of the baseline action detector. Tab. 5 contains the f-mAP@0.5 values, computed on MultiSports, for our re-implementation of the ResNet-50 SlowFast network, the addition of the background negative frames, the conversion of the multi-label to a multiclass classification and finally the addition of the FPN. Each component improves the performance of the detector, leading to a much stronger baseline.

Tracker as filtering module: Using trackers as a post processing step for action detection has many advantages, which we demonstrate in all the above tables, including

Table 6: Recall of Class agnostic YOLOv5 based DeepSort tracker on MS and UCF24 dataset, with and without fine-tuning the detector on each dataset. Even though MultiSports is more complex, the tracker has better recall on it than UCF24, as the detector works better thanks to high resolution and quality of the MultiSports images.

		Recall			
	Fine-tune	MS		UCF24	
Tracker	detector	train	val	train	val
YOLOv5-DeepSort	✗	84.0	84.5	25.0	26.8
YOLOv5-DeepSort	✓	93.1	91.0	94.9	84.0

Tab. 4, where we get substantial improvement in f-mAP, labeled as "Baseline + tracks". Firstly, the tracker helps filter out false positive person detections with high scores that spuriously appear for a few frames. This reduces the load on person bounding box thresholding. Most of the current SOTA methods use a relatively high threshold to filter out unwanted false positive person detections, e.g. pySlowFast [9] uses 0.8 and mmaction2 [7] uses 0.9. Yet, such strict thresholds can eliminate some crucial true positives. In contrast to standard methods, we use a relatively liberal (0.05) threshold value for our track-based method. Secondly, using a good tracker greatly simplifies tube construction. Trackers are specifically designed to solve the linking problem, removing the need for greedy linking algorithms used in prior work [18, 21, 37]. The performance gains, both in v-mAP and f-mAP, obtained by "Baseline + tracks" rows of Tab. 3 and Tab. 4 over the "Baseline" row, clearly demonstrate this.

4.6. Discussion

In this work, our main objective is to study action detection under large motion. The experiments on MultiSports and UCF24, see Tabs. 1 and 2, demonstrate that TAAD, *i.e.* utilizing track information for feature aggregation, improves performance across the board. This does not mean that there is no room for further improvement. Our method is sensitive to the performance of the tracker, since this is the first step of our pipeline. Using a better state-of-the-art tracker and person detector, such as the ones employed by other contemporary methods [19, 27, 28, 41]), should boost performance further, especially on UCF24, where YOLOv5 struggles. Moreover, we can improve action detection performance by incorporating spatial/actor context modelling [28, 53], long-term temporal context [41], or a transformer head [53] or backbone [22, 25] into TAAD.

One could argue that our definition of motion categories is not precise. Unlike the object size categories in MS COCO [23], motion categories are not easy to define. Apart from the complex camera motion (incl. zoom, translation and rotation), which is pretty common, and quick actor motion, both of which we show in Fig. 2, special care has to be taken to avoid mislabelling cyclic motions.

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

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

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

Figure 5: Large-motion due to fast action and camera movement in Volley-spike instance (a) detected by all the methods including baseline, but in (b), Football-steal instance is only detected by ASPP and TCN. (c) Large-motion (0.07) due to camera, baseline fails to detect and ASPP module shows better overlap than TCN.

MultiSports for example contains multiple actions, *e.g.* in aerobics, where the actor starts and ends at the same position. This would result in a high IoU between the initial and last boxes and thus an erroneous small motion label. To solve this problem, we use an average of IoUs computed at different frame offsets. While our motion labelling scheme is not perfect, visual examples show that it correlates well with motion speed. Lastly, Fig. 5 shows examples where the baseline fails to detect action tubes but TAAD is detects them. We will provide more qualitative examples in the **Sup. Mat.** to illustrate this point.

5. Conclusion

In this work, we analyse and identify three coarse motion categories in action detection datasets. We observe that existing action detection methods struggle in the presence of large motions, *e.g.* motion due to fast actor movement or large camera motion, To remedy this, We introduce Track Aware Action Detector (TAAD), a method that utilizes actor tracks to solve this problem. TAAD aggregates information across actor tracks, rather than using a tube made from proposal boxes. We evaluate the proposed method on two datasets, MultiSports and UCF24. MultiSports is the ideal benchmark for this task, thanks to its large number of instances with fast-paced actions. TAAD not only bridges the performance gap between motion categories, but also sets a new state-of-the-art for MultiSports by beating last year's challenge winner by a large margin.

References

[1] Harkirat S. Behl, Michael Sapienza, Gurkirt Singh, Suman Saha, Fabio Cuzzolin, and Philip H. S. Torr. Incremental tube construction for human action detection. In *British Machine Vision Conference (BMVC)*, 2018.

[2] Mikel Broström. Real-time multi-object tracker using yolov5 and deep sort. `https://github.com/mikel-brostrom/Yolov5_DeepSort_Pytorch`, 2020.

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[4] Joao Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733, 2017.

[5] 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.

[6] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. Watch Only Once: An End-to-End Video Action Detection Framework. In *International Conference on Computer Vision (ICCV)*, pages 8178–8187, 2021.

[7] MMAction2 Contributors. Openmmlab's next generation video understanding toolbox and benchmark. `https://github.com/open-mmlab/mmaction2`, 2020.

[8] Ishan Dave, Zacchaeus Scheffer, Akash Kumar, Sarah Shiraz, Yogesh Singh Rawat, and Mubarak Shah. Gabriellav2: Towards better generalization in surveillance videos for action detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 122–132, 2022.

[9] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. Pyslowfast. `https://github.com/facebookresearch/slowfast`, 2020.

[10] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *International Conference on Computer Vision (ICCV)*, pages 6824–6835, 2021.

[11] Christoph Feichtenhofer. X3D: Expanding Architectures for Efficient Video Recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 203–213, 2020.

[12] 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.

[13] Rohit Girdhar, João Carreira, Carl Doersch, and Andrew Zisserman. A Better Baseline for AVA. *arXiv preprint arXiv:1807.10066*, 2018.

[14] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *Computer Vision and Pattern Recognition (CVPR)*, pages 759–768, 2015.

[15] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. AVA: A Video Dataset of Spatio-temporally Localized Atomic Visual Actions . In *Computer Vision and Pattern Recognition (CVPR)*, pages 6047–6056, 2018.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[18] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action Tubelet Detector for Spatio-Temporal Action Localization. In *International Conference on Computer Vision (ICCV)*, 2017.

[19] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll. You Only Watch Once: A Unified CNN Architecture for Real-Time Spatiotemporal Action Localization. *arXiv preprint arXiv:1911.06644*, 2019.

[20] 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.

[21] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *European Conference on Computer Vision (ECCV)*, 2020.

[22] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4804–4814, June 2022.

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common Objects in Context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[24] 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.

[25] 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.

[26] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[27] Zhiqing Ning, Qiaokang Xie, Wengang Zhou, Liangwei Wang, and Houqiang Li. Person-Context Cross Attention for Spatio-Temporal Action Detection. Technical report, Huawei Noah's Ark Lab, and University of Science and Technology of China, 2021.

[28] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-Context-Actor Relation Network for Spatio-Temporal Action Localization. In *Computer Vision and Pattern Recognition (CVPR)*, pages 464–474, 2021.

[29] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream R-CNN for action detection. In *European Conference on Computer Vision (ECCV)*, pages 744–759, 2016.

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection . In *Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[31] Suman Saha, Gurkirt Singh, and Fabio Cuzzolin. AMTnet: Action-Micro-Tube regression by end-to-end trainable deep architecture. In *International Conference on Computer Vision (ICCV)*, 2017.

[32] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. In *British Machine Vision Conference (BMVC)*, 2016.

[33] Gurkirt Singh, Stephen Akrigg, Manuele Di Maio, Valentina Fontana, Reza Javanmard Alitappeh, Suman Saha, Kossar Jeddisaravi, Farzad Yousefi, Jacob Culley, Tom Nicholson, et al. Road: The road event awareness dataset for autonomous driving. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 1(01):1–1, feb 5555.

[34] Gurkirt SingH and Fabio Cuzzolin. Recurrent convolutions for causal 3d cnns. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

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

[36] Gurkirt Singh, Suman Saha, and Fabio Cuzzolin. TraMNet-Transition Matrix Network for Efficient Action Tube Proposals. In *Asian Conference on Computer Vision (ACCV)*, pages 420–437. Springer, 2018.

[37] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online Real-time Multiple Spatiotemporal Action Localisation and Prediction. In *International Conference on Computer Vision (ICCV)*, pages 3637–3646, 2017.

[38] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. TAC-Net: Transition-aware context network for spatio-temporal action detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 11987–11995, 2019.

[39] Khurram Soomro, Haroon Idrees, and Mubarak Shah. Predicting the where and what of actors and actions through online action localization. In *Computer Vision and Pattern Recognition (CVPR)*, pages 2648–2657, 2016.

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

[41] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *European Conference on Computer Vision (ECCV)*, pages 71–87. Springer, 2020.

[42] ultralytics. Yolov5: Real-time object detector. `https://ultralytics.com/yolov5`, 2020.

[43] Jan C Van Gemert, Mihir Jain, Ella Gati, Cees GM Snoek, et al. APT: Action localization proposals from dense trajectories. In *British Machine Vision Conference (BMVC)*, volume 2, page 4, 2015.

[44] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local Neural Networks. In *Computer Vision and Pattern Recognition (CVPR)*, pages 7794–7803, 2018.

[45] 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.

[46] Philippe Weinzaepfel and Grégory Rogez. Mimetics: Towards understanding human actions out of context. *International Journal of Computer Vision*, 129(5):1675–1690, 2021.

[47] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-Term Feature Banks for Detailed Video Understanding . In *Computer Vision and Pattern Recognition (CVPR)*, pages 284–293, 2019.

[48] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[49] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. STEP: Spatio-Temporal Progressive Learning for Video Action Detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 264–272, 2019.

[50] Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018.

[51] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *Computer Vision and Pattern Recognition (CVPR)*, pages 9975–9984, 2019.

[52] Jiaojiao Zhao and Cees GM Snoek. Dance with Flow: Two-in-One Stream Action Detection . In *Computer Vision and Pattern Recognition (CVPR)*, pages 9935–9944, 2019.

[53] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Bing Shuai, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, Ivan Marsic, Cees G. M. Snoek, and Joseph Tighe. Tuber: Tubelet transformer for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13598–13607, June 2022.

[54] Kaiyang Zhou and Tao Xiang. Torchreid: A library for deep learning person re-identification in pytorch. *arXiv preprint arXiv:1910.10093*, 2019.

[55] Kaiyang Zhou, Yongxin Yang, Andrea Cavallaro, and Tao Xiang. Learning Generalisable Omni-Scale Representations for Person Re-Identification. *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.