DeeperAction workshop at ICCV 2021: MultiSports Challenge on Spatio-Temporal Action Detection Track Technical Report: A Solution to Detect Key Actions in Complicated Multi-person Scene

Yanbin Chen, Jiangyuan Mei, Zhicai Ou, Feifei Feng and Jian Tang
AIIC vision group, Midea Group
2388 Houhai avenue, Shenzhen, Guangdong, China
{chenyb60, meijy3, zhicai.ou, feifei.feng and tangjian22}@midea.com

Abstract

This article introduces the solution of the Midea_AIIC team for the multisports challenge on spatio- temporal action detection track of the 2021 deeperaction competition. In this work, we tried to handle three main challenges in MultiSports dataset: the complicated multi-person scene, inaccurate boundary segmentation of tubes and some action is related to environmental information. Thus, two main innovations are presented in our work. Fristly, we improve MOC-Detector by adding a new background branch which provide a further information to distinguish background and actions. Secondly, we propose an adjust tube postprocess method, which improves tubelet linking phase, including dealing with blurred time boundaries and using environment information like ball object to reclass similar actions. We achieved 19.13 video-mAP@$0.10:0.90$ *on test dataset and rank the second place in 2021 ICCV DeeperAction track 2.*

1. Introduction

As a challenging task in video understanding, spatio-temporal action detection not only needs to extract the temporal information of the action in video, but also needs to return the specific target location with a box where the action occurs. This enables it to deal with more complex visual tasks such as video surveillance [11] [5], video information search, sports event analysis and other scenarios where specific action needs to be analyzed, but at the same time, simultaneously extracting time and space information greatly increases the difficulty task processing.

On the DeeperAction track 2, MultiSports [8] is a new dataset for spatio-temporal action detection. It is a densely annotated high-level actions dataset like J-HMDB [6] and UCF101-24 [18], which provides frame-wise action labels of the video.

The comparison between JHMDN, UCF101_24 and MultiSports is shown in Fig. 1. In comparison, J-HMDB and UCF101-24 are the classical benchmarks in spatio-temporal action detection, and have made a huge contribution to the development of the field, but there are some problems on them. One is that the video has a low resolution limited by early technology, and by clipping some videos are not very clear. What's more, each video has only an action category, and doesn't contain background frame while the action usually occupies the main part of the video, which sometimes is different from the actual, reducing their practical application value. Compared with them, MultiSports is a larger densely annotated high-level actions dataset with 720p high resolution. It focuses on the detection of sports actions, giving a clear action boundary both in time and space. Each video has multiple action categories and many people behave at the same time, but only the specified actions should be detected. The smaller target, and the plenty of similar multiple concurrent action instances bring more challenges in spatio-temporal action detection.

On task processing, in the past, spatio-temporal action detection mainly take a frame of videos as the input, and output the detection result of these static images and then merge results to action tubes [3][12][15][20][21][16]. This has achieved some effects for actions that have great correlation with certain human posture and scenes, but for other complicated actions like similar sport actions, it is limited due to the lack of timing dynamic information. In recent years, the number of excellent models proposed for spatio-temporal detection has gradually increased. They use a sequence of frames as input to capture dynamic information, thereby greatly improving the effect of action detection [14][7][4][22][24][17].

Among them, MOC-Detector [9] is the state of art on JH-

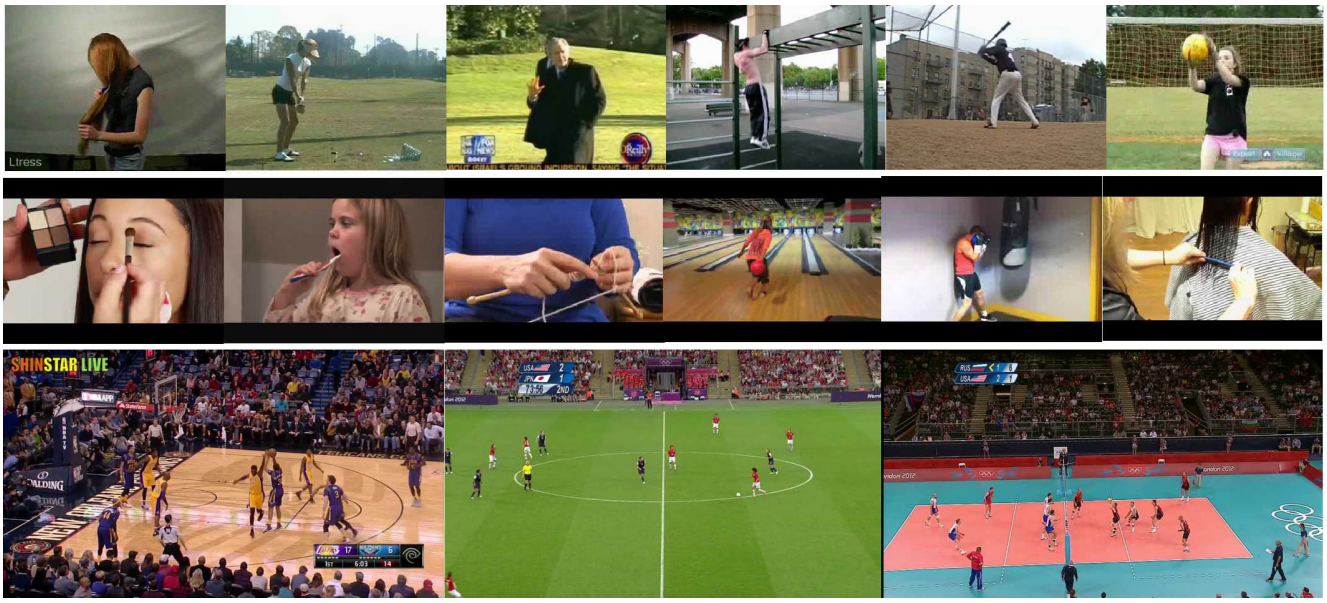

Figure 1. The comparison between JHMDN (row 1), UCF101_24 (row 2) and MultiSports (row 3) data types

MDN and UCF101-24. Most of the tubelet detection methods like ACT-Detecor [7] and SlowFast [2] is mainly based on the classic target detection network such as SSD [10] and Faster RCNN [13]. However, this kind of method relies on manual anchors, which makes network inconvenient to design, and the calculation is very large. When further to process videos, the computation increase more. In comparison, MOC-Detector inherits the concept of CenterNet [1], elegantly returns to the target box of the object in a free-anchor method, and has a great reduction in computational cost. The most important is that, MOC-Detector designs a Movement Branch to predict the trajectory of the human movement. This method clearly return the center position of the human short-term movement, which is equivalent to link k frames predicted target boxes. It replaces the complex task in anchor-based detection method that extends the predicted target box of key frames to other frames to get a 3D ROI features and then use the ROI features to predict the action category, proposing an excellent idea for simplifying the task of spatio-temporal action detection. While simplifying the task processing, MOC-Detector still maintains an excellent accuracy. This is the main reason why we choose MOC-Detector as the baseline.

By looking into the evaluation results of MOC-Detector, we found that: 1) accurate target detection from the background is crucial to final results; 2) inaccurate boundry segmentation of tubes affects the TIOU betweenthe predicted tubes and the ground truth severely. Thus, two main innovations are proposed in our work.

Firstly, based on MOC-Detector, a new branch called background branch is added to predict the confidence of actions. This modification (MOC-B-Detector) help to improve the overall performance, including accuracy and converge speed. Besides, it can help to screen extra detection tubes.

Secondly, a new post process called Adjust Post Process (ATP) is proposed to handle the blurred time boundary of tubes. This post process improves the video mAP greatly.

In addition, we use environment information like ball, racket to build the relation between action and sport. By this way, the false classifcation rate is reduced.

2. Method

Our framework is mainly developed from MOC-Detector. MOC-Detector is an excellent detector for processing spatio-temporal action detection.The input of MOC-Detector is consecutive k frames. Then, the model uses a 2D shared weights backbone to extract features of k-frame. Next, the k-frame feature maps are sent to three branches called center branch, movemet branch and box branch. The center branch outputs center points of instances and categories of actions of key frames. The movement brach aims at analyzing the movement between the key frame and other frames, and estimate the trajectories of moving points.The box branch is designed to regress the size of bounding box in each frame. After that, the outputs from these three braches generate predicted tubelet results. With a matching strategy, these tubelets are further linked to yield video-level tubes.

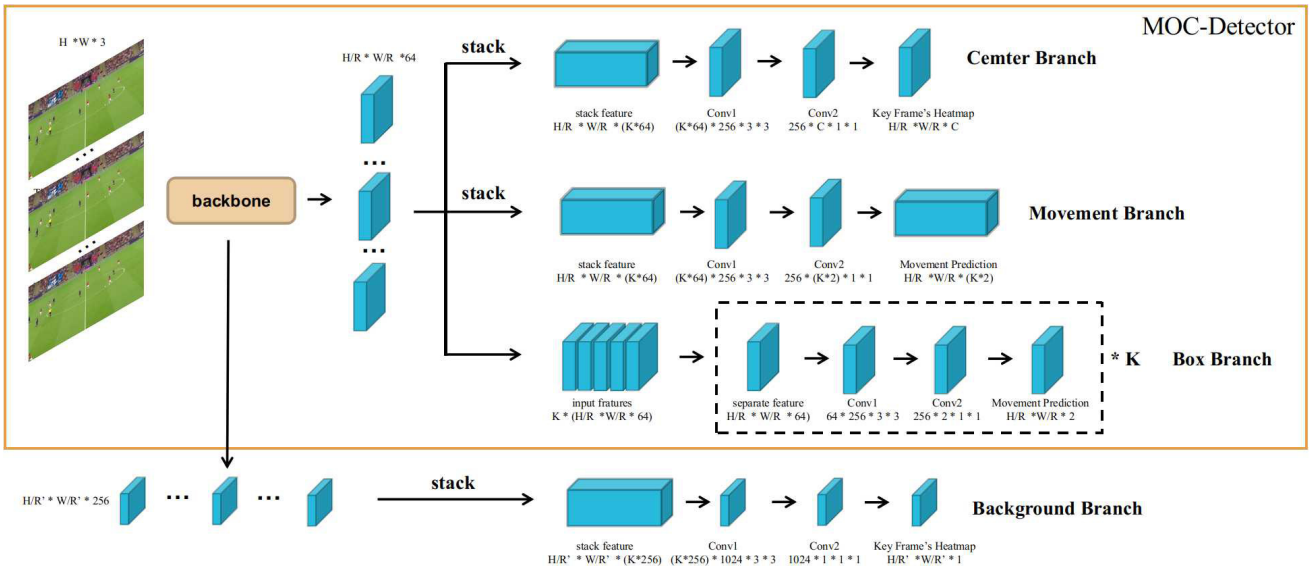

Figure 2. MOC-B framework.

2.1. Framework

Although MOC-Detector achieves the state-of-the-art results on J-HMDB and UCF101-24 datasets. In MultiSports challenge, the situation is different. MultiSports dataset has a complicated multi-person scene. In the video, behaviors from many people are often similar, especially for football players. However, only specific actions need to be detected and the others belong to background. In addition, top-N results from center branch are obatained to process tubelet linking, but the value N usually larger than the number of action instances. This will cause a large amount of redundant results. Thus, in our framework, we try to add an another branch to distinguish the background actions and ground truth actions.

Inspired by auxiliary classifiers from GoogLeNet [19] and [23], MOC-B adds another scale feature map as the input of new branch on the basis of MOC-Detector, and we call it background branch. This branch is only used to predict the confidence of actions. The framework is shown in Fig. 2.

We concatenate $16\times$ down-sampling k frames feature maps of the backbone network to predict action confidence via two convolution layers. Compared with using final feature maps, this makes a negligible calculation on the original model. Secondly, the branch establishes a shortcut between the relatively middle layer and loss calculation. The loss can directly affect the middle layer, which is more conducive to the gradient backpropagation of the shallow network layers, so as to accelerate the loss reduction speed. And $16\times$ down-sampling feature maps own smaller size. In order to predict correctly, the branch leads less information loss and more information is tend to be saved to improve the overall performance.

In addtition, compared to detect multiple action categories at the same time, it is easier to just detect the possibility of actions and has a higher recall rate and accuracy to distinguish background and actions. In inference, the confidence from background branch can work together with the heatmap from center branch to improve detection quality in tubelet linking phase.

The training objective of the MOC-B framework is expressed as

$$l = l^c + al^m + bl^b + cl^{bg} \tag{1}$$

where l^c stands the center branch loss, it is a variant of focal loss. l^m is the movement branch loss, and it is a l_1 loss function. l^b represents the box branch loss, and it is also a l_1 loss function. The details of these three loss functions can be found in the work [9]. We add a brackground branch loss function l^{bg} here.

The l^{bg} function is similar to center branch loss l^c. The diffenece is that the center branch loss has a category parameter where brackground branch only considers background or actions. So, our the l^{bg} function is expressed as a focal loss as following,

$$l^{bg} = -\frac{1}{n}\sum_{x,y} l^{bg}_{xy}, \tag{2}$$

where

$$l^{bg}_{xy} = \begin{cases} \left(1-\widehat{H}_{xy}\right)^{\alpha} log\left(\widehat{H}_{xy}\right), & H_{xy}=1 \\ \left(\widehat{H}_{xy}\right)^{\alpha} log\left(1-\widehat{H}_{xy}\right), & H_{xy}=0 \end{cases} \tag{3}$$

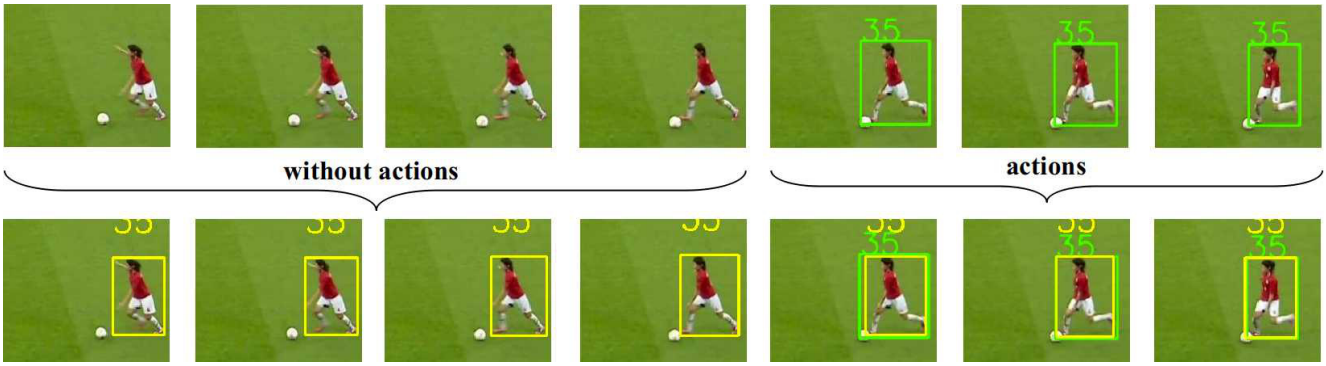

Figure 3. Example of that partly actions in consecutive 7 frames contribute their score to frames without action. The green box is ground truth and the yellow is predicted.

The ground truth heatmap H_{xy} follows a binary distribution. For the ith action instance, consider a point (x_i, y_i), if point (x_i, y_i) is in the bound box of ground truth label, $H_{xy} = 1$ no matter what kind of action categories. If (x_i, y_i) is not in the bound box of ground truth, H_{xy} is set as 0, which means it is a background. The $\widehat{H}_{xy}$ is predicted result at point (x, y), n is the number of ground truth instances and α is hyperparameter of the focal loss.

2.2. An Adjust Tube Post Process Method

Tubelet linking algorithm from ACT-Detector is used to merge clip-level tubes into video-level tubes. In multisports dataset, the situation is more complicated. Thus, we design a series of post process methods to improve the tubelet linking algorithm. These post process methods are called adjust tube post process (ATP) algorithm. And we will introduce some details of the ATP algorithm.

Firstly, we'd like to deal with the blurred time boundary of tubelets. We find that the predicted tubes are always longer than the ground truth. There are three reasons for this phenomena. The first reason is that the length of tubes is larger than the parameter of k. A large k always means more available temporal information. However, some ground truth action tubes are shorted than k. The second reason is that the clip-level detector with k frames as input is usually used to generated the heatmap of key frame, but the result works on all k frames. On validation, each frame is needed to generate a tubelet, but not every frame has actions. If consecutive k frames own partly actions, it will contribute action score to other frames without actions, extending the time on both ends, as showed in Fig. 3.

Another situation is that when some actions occur, it is difficult to have a perfect boundary between the start time and the end time. Action is a process and inevitably difficult to distinguish accurately. The ambiguity will also make action prediction start faster or end slower than the actual.

In response to this phenomenon, we designed an efficient method called time boundary cut to deal with this problem. For predicted action, each frame has a corresponding score. The score is the average score of the category heatmap of the tubelet overlap frames when tubelets are linked. It can be used as an important information to delete frames without action. The action information from the frames without action must be insufficient, and will have a lower score. Therefore, a threshold can be selected to eliminate them.

For difference in score between easily detected actions and hard detected actions, in order to improve robustness, for one tube we calculate the median value of all frames scores as a dynamic threshold δ and set a weight w to make the score compare with the median score $median\,(s_i)$ multiply by the weight . For all frame score less than the δ, we will remove the frames at both ends.

$$\delta = median\,(s_i)\,w, i = 1, 2, \cdots, n, \tag{4}$$

$$\overline{T} = Clip\,(T\,[s_i > \delta])\,, i = 1, 2, \cdots, n, \tag{5}$$

where T and $\overline{T}$ are tubes before and after time boundary cut. s_i is the ith frame action score. $Clip\{\}$ is the operation to delete frame on both ends whose score lower than the threshold δ. Time boundary cut strategy brings a huge improvement for video-mAP.

More action information from trainval can be used to improve tublete linking. Based on statistical methods, we counted the durations from all action category on trainval dataset to improve predicted result further. Action's duration should be reasonable. For example, volleyball spike cannot last long and lasts up to about 20 frames. Basketball dribble may be long or short. We can delete actions that are too long or too short. On the other hand, the predicted action duration is greater than or equal to the value of k, but there are many actions whose duration is less than the value of k, so we need to separately tailor the time boundaries for different action categories.

2.3. Environment information

Environment information like ball object can be used to screen extra actions. MultiSports mainly focuses on the detection of sports events, in which volleyball, basketball and football are all strongly related to ball object. We used public yolov5 trained on coco dataset to detect balls and by introducing ball position, it can be used to filter some false negative samples.

Meanwhile, Some actions are indistinguishable with background. For example, in multi-person sports scene, like football, whether or not action is occurred, football player are walking and running almost all the time, so their postures are very similar. The similar human postures lead a large amount of redundant detection actions , and the ball is an important information to distinguish them. So we crop a larger area centered on predicted action boxes to contain ball object and trained a small classification network to reclassify the tubes.

2.4. Finetune

In the training stage, we firstly trained the model 20 epochs and then analyzed the results. Some actions such as basketball dribble steal, have a very high miss rate. The main reasons for this phenomenon can be explained from two aspects. On the one hand, the amount of these actions type samples is not enough, which leads a poor predictive ability on val or test dataset. On the other hand, some action types are too complicated to detect. They have very different tubelets in different situations. The model can't correctly predict them or can't distinguish them from the other actions.

In the final analysis, the main reason is that the distribution of actions number of MultiSports dataset is unbalanced. So we need to increase the numbers of training times for some data. For actions with higher miss rate, we train it more times each epoch if one frame has actions with high miss rate. The higher miss rate, the more repetitions.

$$t_m = floor(max(m_i)//10) \tag{6}$$

where t_m is the repeating times of training the mth frame. $miss_i$ is the miss rate of the ith category. $//$ represents divisible. $floor\,()$ represents the downward integration.

3. Experiment

3.1. Dataset

We perform experiments on MultiSports dataset [15]. MultiSports is a high quality dataset for spatio-temporal action detection. It mainly focuses on the detection of sport actions from four sports event, including aerobic gymnastics, volleyball, football and basketball, which totally contains 66 sport action categories. For each action, the time boundary and target box are labeled clearly. The train dataset has 1574 videos and 18422 instances, and the val dataset has 555 videos and 6577 instances.

3.2. Implementation Details

In the training stage, we set $a = 1$ and $b = 0.1$ which follows the work [18] in Equ. 1. For the parameter c, we set it as 1 after some experments. In the Equ. 3, we set $\alpha = 2$ which is same to the work [9].

The original video resolution is 1280×720. we crop it to each frame and normalize each frame to 320×320 for training and validation. We input 11 consecutive frames each time. Then, we trained 20 epochs first and each epoch selects one scale from five scales $[384 \times 384, 352 \times 352, 320 \times 320, 288 \times 288, 256 \times 256]$ for training. The initial learning rate is $5e^{-4}$, and it drops 10 times after the 6th and 8th epoch. We perform two more epoch of fine-tuning training after 20 epochs with learning rate $5e^{-7}$. We trained on 8 Tesla P100 and consumed about 4 days.

We use 320×320 as the input image size with horizontal flip test. And the parameter N is set as 5. All the length of tube were saved. Then we follow MOC-Detector to link tubelets and use our proposed ATP method to improve linking results.

	AP	Extra	Cls	Miss
MOC-B*	34.69	18.36	15.46	23.93
MOC-B	33.12	25.32	18.58	13.58

Table 1. Error analysis. MOC-B* represents the postprocess select tubes over 15 frames length. MOC-B represnets saves all tubes.

	Video-mAP(%)				
	@0.10:0.90	@0.2	@0.5	@0.05:0.45	@0.50:0.95
MOC-B*	13.91	30.63	10.81	26.74	2.79
MOC-B	13.22	29.33	10.11	25.49	2.61
MOC-B*+ATP	20.12	37.24	22.04	34.31	7.23
MOC-B+ATP	20.22	37.51	22.05	34.52	7.24

Table 2. The comparison of video-mAP result on MOC-B* and MOC-B.

3.3. Ablation Studies

In the first experiment, we'd like to introduce the difference about miss rate in tubelet linking phase. In the original tubelet linking algorithm, it only selects the tublets whose length over 15 frames. Although it can improve the AP socre by reducing redundant results. It also brings too high miss rate, as shown in table 1.

In our method, we choose to save all tubes, and use the ATP method to post process these redundant results and reduce the miss rate. Although video-mAP from all tubes is

Model	Frame-mAP @0.5(%)	Video-mAP(%)				
		@0.10:0.90	@0.2	@0.5	@0.05:0.45	@0.50:0.95
BS	25.74	11.82	26.08	8.79	22.87	2.28
+MS	26.45	12.37	27.42	9.34	23.82	2.50
+BB	27.06	13.16	28.87	10.11	25.31	2.61
+K11	27.14	13.22	29.33	10.36	25.49	2.63
+FT	27.56	13.30	29.02	10.41	25.57	2.72
+ATP	27.56	20.18	36.92	22.32	34.13	7.39
+EN	27.56	20.32	37.23	22.37	34.23	7.41

Table 3. Ablation experimental results on val dataset. BS is the baseline MOC-detector with parameters $k = 7$, $N = 5$ and save all tubes. MS represents use multi-scale images as the training input. BB represents MOC-B framework which adds the background branch. K11 means $k = 11$. FT means fine-tune two epochs in the last. ATP use the proposed adjust post process method. EN is using environment information in post process.

lower than the tubes with a length greater than or equal to 15 frames for more redundant results, it has lower miss rate. It means a higher potential in improving video-mAP after time boundary cut and statistics-based post-process method, and the comparison of results before and after ATP method is showed in table 2.

In the second experiment, we mainly evluate peformance of each module, as shown in table 3. In this table, BS represents baseline MOC-detector. And the parameters are set as $k = 7$, $N = 5$ and save all tubes. The frame-mAP@0.5 is 25.74%, and the video-mAP@$0.10 : 0.90$ is 11.82%.

MS represents using multi-scale images as the training input. The original input resolution is 320×320. The MS method select one scale from five scales $[384 \times 384, 352 \times 352, 320 \times 320, 288 \times 288, 256 \times 256]$ for training in epoch . The frame-mAP@0.5 increases 0.71%, and the video-mAP@$0.10 : 0.90$ increases 0.55%.

BB represents MOC-B framework which adds the proposed background branch. It improves the overall perfomance from two aspects. On one hand, it converges faster than original MOC detector. On the other hand, the final converged loss value is lower than that of the original MOC detector. With background branch, the frame-mAP@0.5 improves 0.61%, and the video-mAP@$0.10 : 0.90$ improves 0.79%.

We select a large parameter k in the training stage. Larger k means that richer temporal information are used. The good thing is that it can increase recall rate, and the bad point is it also bring redundant results and long tublets. So we can see, when we increase k from 7 to 11, the frame-mAP@0.5 and video-mAP@$0.10 : 0.90$ increase a little. However, it help reduce miss rate. Using the following ATP module, they two together can improve the video-mAP@$0.10 : 0.90$ index a lot.

The FT module is to deal with some action categories which has a low accuracy because of small number of training samples. These actions are very hard to detect. We perform two more epoches of fine-tuning training after 20 epochs. In these two epochs, these frames with actions without enough number of samples would be trained more times. The fine-tuned two epoch with learning rate $5e^{-7}$. And FT module earns 0.42% of frame-mAP@0.5 and 0.08% of video-mAP@$0.10 : 0.90$.

The ATP is our proposed adjust post process method to improve action time boundary, and it affect the video-mAP@$0.10 : 0.90$ index a lot. For the reason that predicted action duration is usually longer than the actual. The TIOU value between predicted tubes and ground truth tubes is relatively low, leading a bad overall performance, so we optimized the accuracy of the boundary and cut the unreasonable actions with ATP method. From the table 3, we can see, the proposed ATP method make a significant improvement. The video-mAP@$0.10 : 0.90$ raised from 13.30% to 20.18%.

At last, environmental information is also important in distinguishing different actions and background. In our method, we only use ball object information to help reclassify tubes with a classification network. And the video-mAP@$0.10 : 0.90$ raised from 20.18% to 20.38%. In fact, there is various environmental information which be used to improve the performance. For example, "football trap" and "football steal" are almost the same actions. The way to distinguish them is to compare cloth colours between the two people who pass and trap the ball.

Frame-mAP @0.5(%)	Video-mAP(%)				
	@0.10:0.90	@0.2	@0.5	@0.05:0.45	@0.50:0.95
12.9872	19.13	35.05	20.83	32.48	7.11

Table 4. The final result on testing dataset.

The result on testing dataset is shown in table 4. We achieve 19.13 video-mAP@$0.10 : 0.90$ on test dataset and rank the second place in 2021 ICCV DeeperAction track 2.

4. Conclusion

In this work, we provide a solution for Multisports datasets. Firslty, we added a new backgroud branch for MOC-Detector and use the multi-scale images as the inputs, which is appropriate to deal with the complicated multi-person scene of Multisports datasets. Secondly, we proposed an ATP method to deal with the blurred time boundary of tubelets in the tubelet linking phase, and make use of environmental information to improve the classification results. In the future, we'll spend more effort to merge the post-process into the model and make it a conceptually simple framework.

References

[1] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

[2] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019.

[3] Georgia Gkioxari and Jitendra Malik. Finding action tubes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 759–768, 2015.

[4] Rui Hou, Chen Chen, and Mubarak Shah. Tube convolutional neural network (t-cnn) for action detection in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 5822–5831, 2017.

[5] Weiming Hu, Tieniu Tan, Liang Wang, and Steve Maybank. A survey on visual surveillance of object motion and behaviors. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 34(3):334–352, 2004.

[6] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 3192–3199, 2013.

[7] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4405–4413, 2017.

[8] 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. *arXiv preprint arXiv:2105.07404*, 2021.

[9] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *European Conference on Computer Vision*, pages 68–84. Springer, 2020.

[10] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[11] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3153–3160. IEEE, 2011.

[12] Xiaojiang Peng and Cordelia Schmid. Multi-region two-stream r-cnn for action detection. In *European conference on computer vision*, pages 744–759. Springer, 2016.

[13] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[14] Suman Saha, Gurkirt Singh, and Fabio Cuzzolin. Amtnet: Action-micro-tube regression by end-to-end trainable deep architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4414–4423, 2017.

[15] Suman Saha, Gurkirt Singh, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Deep learning for detecting multiple space-time action tubes in videos. *arXiv preprint arXiv:1608.01529*, 2016.

[16] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3637–3646, 2017.

[17] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. Tacnet: Transition-aware context network for spatio-temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11987–11995, 2019.

[18] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. A dataset of 101 human action classes from videos in the wild. *Center for Research in Computer Vision*, 2(11), 2012.

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[20] Limin Wang, Yu Qiao, Xiaoou Tang, and Luc Van Gool. Actionness estimation using hybrid fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2016.

[21] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE international conference on computer vision*, pages 3164–3172, 2015.

[22] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2019.

[23] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017.

[24] Jiaojiao Zhao and Cees GM Snoek. Dance with flow: Two-in-one stream action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9935–9944, 2019.