LCTS: Longest Continuous Temporal Sequences for Action Detection

Shaomeng Wang, Yan Song, Keke Chen, Zeyu Zhou, Rui Yan, Xiangbo Shu, Jinhui Tang

School of Computer Science and Engineering, Nanjing University of Science and Technology, China
Nanjing, China

wangshaomen@gmail.com

Abstract

This paper presents our solution to the Multi-Sports Track on Spatiotemporal Action Detection of the ICCV DeeperAction Challenge. Distinguishing the start and end time of multiple different actions and different participants in a video is a challenging task towards complex video understanding. Recent works have achieved good performance in simple video. In this paper, we not only model the single temporal segment on simple video, but also take into account multiple different temporal segments established upon complex video. We adopt SlowFast as the backbone to extract video features, and Actor Centric Relationship Network(ACRN) as the main module of the relational network to establish the model. In addition, aiming at detecting actions temporally and getting the time segments in different actions in a video, we propose the ***Longest Continuous Temporal Segment*** *module. The mAP of our method on the Multi-Sports dataset is 7.09 and ranks the third in the Multi-Sports Track on Spatiotemporal Action Detection of the ICCV DeeperAction Challenge.*

1. Introduction

Spatio-temporal action detection aims to recognize the actions of interest that are present in a video and localize them in both space and time and has been developing these years[1, 2, 3, 4, 5, 6]. Based on the technology of 2D image classification and localization, it has developed into the detection and recognition of actions which are represented by the 3D bounding box and start-to-end frame index(i.e., a sequence of bounding boxes of action). At present, for video understanding, previous works have designed many network models such as two-stream network[7], 3D convolution[8], TSN[9], BMN[10], SlowFast[11] and so on.

These models have achieved good performances on densely labeled highly abstract action datasets(i.e., J-HMDB and UCF101-24) or sparsely labeled datasets(i.e., AVA). However, these models do not get satisfactory results on new datasets which have a large number of simultaneous motion scenes and clear motion boundaries compared to the datasets mentioned above.

In this paper, we take the Actor Centric Relationship Network(ACRN)[12] as our baseline and propose a new model that handles the problem mentioned above.

We will introduce the framework of our model and demonstrate the experiment results on the Multi-Sports dataset.

2. Proposed Method

In this section, we present our approach for the relation modeling in spatio-temporal action detection. The main challenge is that the test samples are long videos without untrimming, and we need to submit not only the action detection results of each frame (spatial) but also the video segment corresponding to each action (temporal). To tackle the above challenges, We adopt SlowFast as the backbone to extract video features, and Actor Centric Relationship Network(ACRN)[12] as the main module of the relational network to establish the model. In addition, aiming at detecting actions temporally, we propose the longest continuous time sequence (LCTS) module and achieve good results in the competition.

2.1. Overall Framework

The action detection process can be simply expressed as Input → Backbone → Feature extraction and fusion module(FEFM)→Branch → Classification and Prediction. In the input part, we have done a lot of work in data adaptation and found that the organization's format of data is AVA[13]. See Data Preprocessing for details. The research of FAIR[6] shows that SlowFast network shows a very good effect on action recognition, so we use Resnet3d-SlowFast which depth in the slow path and fast path is 50 [14, 15] to extract video features. In the FEFM part, ACRN is added to the model, which computes and accumulates pairwise relation information from actor and global scene features, and generates relation features for action classification. After obtaining the features, we transfer them to the spatial branch,and we transfer bounding boxes to the temporal branch after the spatial branch obtains. The overall architecture of the proposed method is presented in Figure 1.

2.2. Backbone

We use ResNet-I3D-SlowFast[16] as our video feature extraction network. The network can be regarded as the superposition of two I3D models (called slow branch and fast branch respectively). For any branch, it is still an I3D model in essence. For both slow and fast branches, we use ResNet-I3D network with depth of 50. In addition, the network uses the later branch to fuse the output features of the slow branch and the fast branch. The basic method of the later branch is to perform 3D convolution conversion on the features of a certain position in the fast branch, and then perform full connection operation with the slow branch of the same layer. The result of full connection is the output of the slow branch. The premise of the later branch is that the structures of the slow branch and the fast branch should be the same except for the number of channels.

2.3. Feature Extraction and Fusion Module

In this module, we extract the region of interest of the input video features and add ACRN Head, which makes the features integrate context information and more global. The input of the model is a feature map of full size. After randomly screening the proposals, the bounding box region of interest extractor cuts the features in the corresponding feature map according to the size of the proposals. In addition, the extractor also performs Max pooling along the time dimension. After extracting features, we input them into ACRN head. At the same time, context features and the regions of interest are also input. The context feature is obtained by revolution of video feature. Firstly, ACRN operate maximum pooling on extracted features, and copies them with the size of context feature to obtain tile feature. Then the tile feature and context feature are spliced along one dimension, and finally the ROI features that have interacted with context feature is obtained after several times 1×1 and 3×3 convolution operation.

2.4. Spatial Branch

The spatial branch mainly solves a multitask learning problem that involves action classification and localization regression. The action detection model was proposed in [13], the input of which includes the features of the context information of the keyframe generating action prediction, and the output includes the 2D bounding boxes of the keyframe and the corresponding label. For actor localization, we use the region proposal network (RPN) from Faster R-CNN to generate 2D actor proposals. For action classification, we select the adjacent frames of keyframes. We expand the two-dimensional feature information of a single frame into three-dimensional feature information with time dimension by aligning the bounding boxes of adjacent frames. So the temporal information near keyframes can also be considered in action classification.

2.5. LCTS

The input bounding box denoted as $P^i \in R^{H \times W}$, where i is the person bounding box index. Since each person bounding box has its corresponding confidence, we can expand the box into a tuple $(P^i, score)$. After the bounding box gets the corresponding label through the spatial branch, it is input into the temporary branch. The input of the temporary branch is described as $X^i \in \{(P^i, score, label)|i \in I\}$, where I is the index set of person bounding boxes. After the temporary branch accepts the input, firstly, since there are multiple person bounding boxes in a frame, we select the key bounding box of each frame according to the Euclidean distance from the key object. Then temporal feature detection is performed on the box set F with the same label in a video. A continuous detection sequence with intervals is obtained, which can be intuitively described as $S = \{(start_1, end_1), (start_2, end_2), ..., (start_i, end_i)|start_i < end_i, start_j < end_k \ when \ j < k\}$. We provide two strategies to select the final temporal sequence in set S. Strategy 1: selecting the longest continuous temporal subsequence(LCTS),i.e. $max(length \ of \ x)$;Strategy 2: selecting the continuous subsequence with the maximum average confidence, i.e. $max(average \ confidence \ of \ x)$,where average confidence= $\frac{\sum score}{length \ of \ x}$. Eventually the temporal branch outputs the start frame and end frame of the same action. More details of the architecture are shown in Figure 2.

3. Experiments

3.1. Data Preprocessing

We use the data provided by DeeperAction Challenge for training, validation, and testing.

For training, we modify the original dataset format with reference to the AVA data format to adapt the ACRN. In addition, for the data loading, we specially designed a corresponding function which can adapt the variable length of dataset automatically to ensure that all the frames provided can be used for training and validation. Besides changing the format of data, we also create a new annotation file from

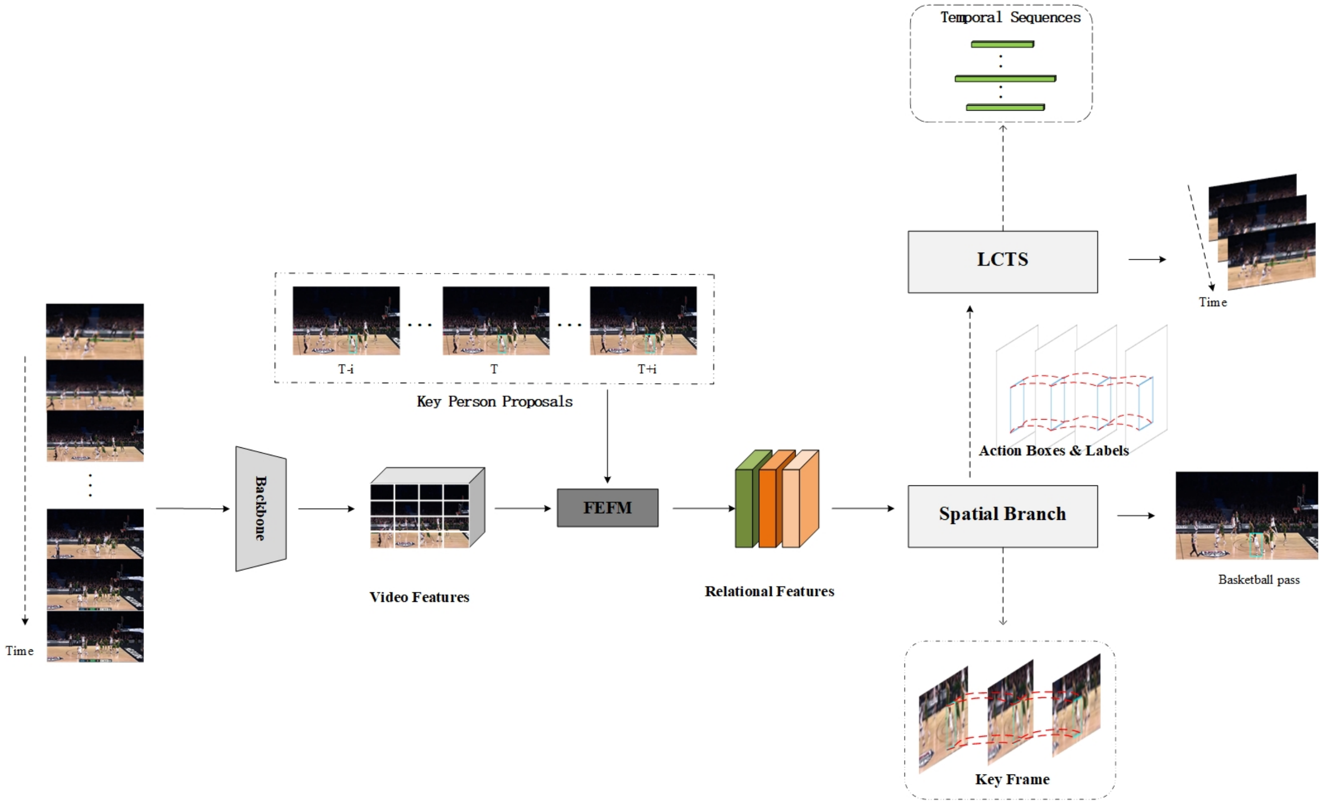

Figure 1. **Overview of our model**. Video clips are processed with a Backbone Network to produce video features. For key actor proposals, we extract actor features from the video features by RoIAlign. Given the actor and video features, the FEFM will generate relational features. These relational features will be sent to the Spatial branch to generate spatial-temporal boxes to classify action. These spatial-temporal boxes will be sent to LCTS to generate more accurate temporal sequences.

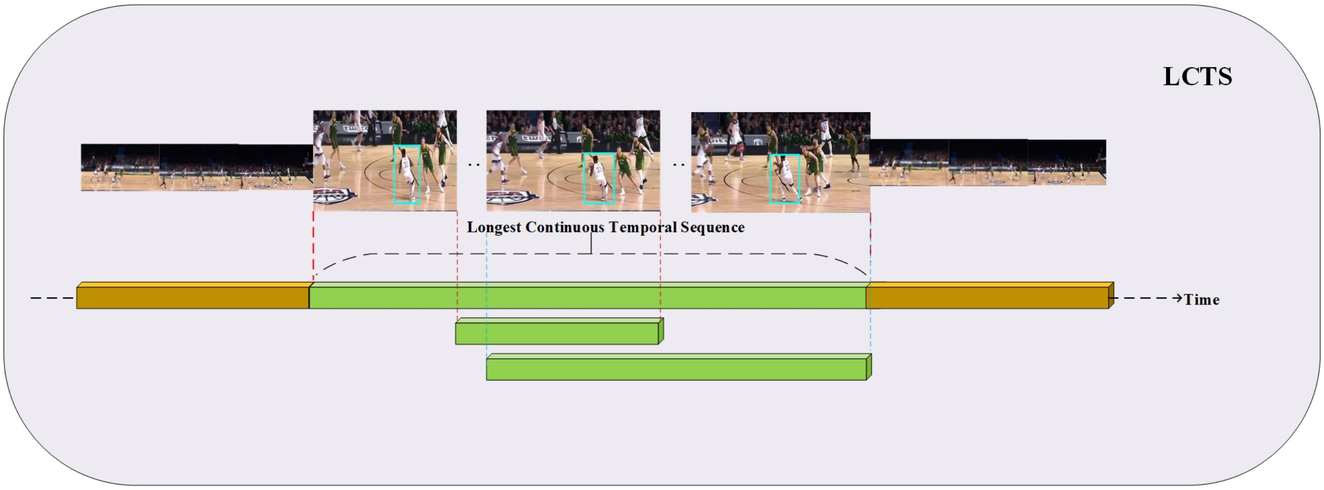

Figure 2. **Longest Continuous Temporal Sequence(LCTS) architectures**. Given spatial-temporal boxes with action labels, first, we generate all temporal sequences that contain actions, then for each sequence, we calculate its time interval, finally, we choose the max time interval as our temporal proposal.

the organizer's annotation file to make our model adopted to the Multi-Sports dataset suitably. Our new annotation file is a CSV file and contains many lines. Every line consists of video name, timestamp(901+frame index), normalized bounding box(*x1, y1, x2, y2*), label map, person ID, and the total frames of each video.

For testing, we keep the test set format consistent with the training dataset.

3.2. Experimental settings

Person Detector. We adopt the official prediction results for person bounding box. The results are obtained by using the model trained by Faster-RCNN[17] on COCO dataset to detect the characters of Multi-Sports. Before the official results are given, we also use the same method for character detection, and it is found that the accuracy of the two methods is almost the same.

Backbone. We set the resample rate = 4 (corresponding to τ in [18]), speed ratio = 4 (corresponding to α in [18]), and channel ratio = 8 (corresponding to $\frac{1}{\beta}$ in [18]), which means that the frame interval in the slow branch is $\tau = 4$ and the frame interval in the fast branch is $\frac{\tau}{\alpha} = 1$. In addition, in the slow branch, we use the lateral connection. The size of the first layer convolution kernel is set to $(1, 7, 7)$ and the step in the temporal dimension is 1. We set the step of the first pooling layer in the timing direction to 1, and use regularization during valuation.In the fast branch, we do not use the lateral connection. Although [18] shows that the two-way lateral connection can improve the accuracy of the model, but it also increase the time cost of the model. Under the balance, we choose to sacrifice accuracy for time to a small extent.At the same time, we set the size of the first layer convolution kernel to (5,7,7)and we do not use regularization, the other settings are consistent with the slow branch.

Heads. In the FEFM part, we combine ACRN-Head to capture the actor centered relationship network. This allows us to have more explicit ROI characteristics as input for task specific prediction.

Training and Inference. In order to achieve better training effect, we enhance the data from the following aspects. We first apply a random scaling with scales sampled from 256 to 320. Second, we randomly crop a region by a crop-size. In the last model training, we set crop-size to 256. Finally, we perform a flip with ratio = 0.5. We use SGD optimizer[19] with the momentum of 0.9 and the weight decay of 1e-5. The Cosine Annealing[20] algorithm is used as the learning rate attenuation strategy. According to the linear scaling rule, the average learning rate corresponding to each GPU should be set to 0.009375. Finally, we completed the training of the model on a single 3090 graphics card with batch-size (Single GPU)= 8,clip-len =32, frame-interval = 2, total epoch = 10. In the inference part, We use the model generated by 2, 3, 5, 7, 10 epoch respectively to test. Frame by frame detection is carried out on RTX 3090 with batch-size = 1, frame-interval = 1. Our model predicts each bounding box of each frame, and gives its corresponding action label and score. For temporary detection, we propose two schemes for the key bounding box of each frame: first, on the premise that the frame detection has the same label, we select the longest sequence in video as the final clip of the label in the video. Second, the premise remains unchanged, and the continuous frames with the highest average confidence in a video are selected as the result.

Strategy	**V@0.10:0.90**	**V @0.2**	**V@0.5**
LCTS	7.092	14.516	6.240
MCCTS	2.549	6.917	0.610
ACAR	1.288	3.009	0.754
Strategy	**V@0.05:0.45**	**V@0.50:0.95**	-
LCTS	13.055	1.810	-
MCCTS	5.644	0.123	-
ACAR	2.652	0.177	-

Table 1. **Comparison with results of video mAP.** The mAP calculation code is provided by the sponsor.

3.3. Results

To better highlight ACRN, we conduct a comparative experiment with ACAR and the results are listed in the table 1 and table 2.

Table 1 shows the results of video mAP the Multi-Sports. We list the prediction results of temporal branches under different strategies. The results show that the longest continuous temporal sequence(LCTS) strategy is better than the maximum confidence continuous temporal sequence(MCCTS) strategy in temporal detection. The mAP of LCTS strategy achieved 7.092 in the index test of V@0.10:0.90 and 14.516 in the index test of V@0.2. The result is finally evaluated by index V@0.10:0.90 in the competition. LCTS strategy is 4.543 higher than MC strategy.

Table 2 shows the frame test results of several epochs. It can be seen from the table that the performance of the mode is positively correlated with the number of training epoch l in the validation set. However, if the frame accuracy verification code given by the sponsor is used, the accuracy of the last generation model is not the highest for the test set. On the contrary, the test results of the 7th epoch are better than those of the 10th epoch.

3.4. Qualitative Results

In order to better demonstrate the effect of our method, we demonstrate the visualization work. As shown in Figure 3, we randomly select a video from 4 categories for visualization. The key person has been marked by ACRN with a blue rectangle and a video is cut into tubes as shown in the figure. The action labels are also predicted correctly.

Epoch	**Ours** Val mAP@0.5	**Ours** Test mAP@0.5
1	24.32	3.902
7	36.75	18.609
10	39.36	16.702

Epoch	**ACAR** Val mAP@0.5	**ACAR** Test mAP@0.5
1	10.01	0.94
7	33.48	2.87
10	37.59	3.12

Table 2. **The frame test results of several epochs compared with ACAR**. The val mAP@0.5 is the result obtained on the verification set using our own mAP calculation code. The test mAP@0.5 in the table is the result on the test set using the sponsor code.

References

[1] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross Girshick. Long-Term Feature Banks for Detailed Video Understanding. In *CVPR*, 2019.

[2] 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[3] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, and Wanli Ouyang. T-cnn: Tubelets with convolutional neural networks for object detection from videos. *arXiv preprint*, 2016.

[4] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object detection from video tubelets with convolutional neural networks. In *CVPR*, 2016.

[5] 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 (ICCV)*, Oct 2017.

[6] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[7] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[8] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks, 2015.

[9] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[10] Tianwei Lin, Xiao Liu, Xin Li, Errui Ding, and Shilei Wen. Bmn: Boundary-matching network for temporal action proposal generation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3889–3898, 2019.

[11] C. Feichtenhofer, H. Fan, J. Malik, and K He. Slowfast networks for video recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[12] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018.

[13] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.

[14] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Learning spatio-temporal features with 3d residual networks for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.

[15] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] Dahua Lin Yue Zhao, Yuanjun Xiong. Mmaction. `https://github.com/open-mmlab/mmaction`, 2019.

[17] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[18] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition, 2019.

[19] J. Michael Cherry, Caroline Adler, Catherine Ball, Stephen A. Chervitz, Selina S. Dwight, Erich T. Hester, Yankai Jia, Gail Juvik, TaiYun Roe, Mark Schroeder, Shuai Weng, and David Botstein. SGD: Saccharomyces Genome Database . *Nucleic Acids Research*, 26(1):73–79, 01 1998.

[20] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017.

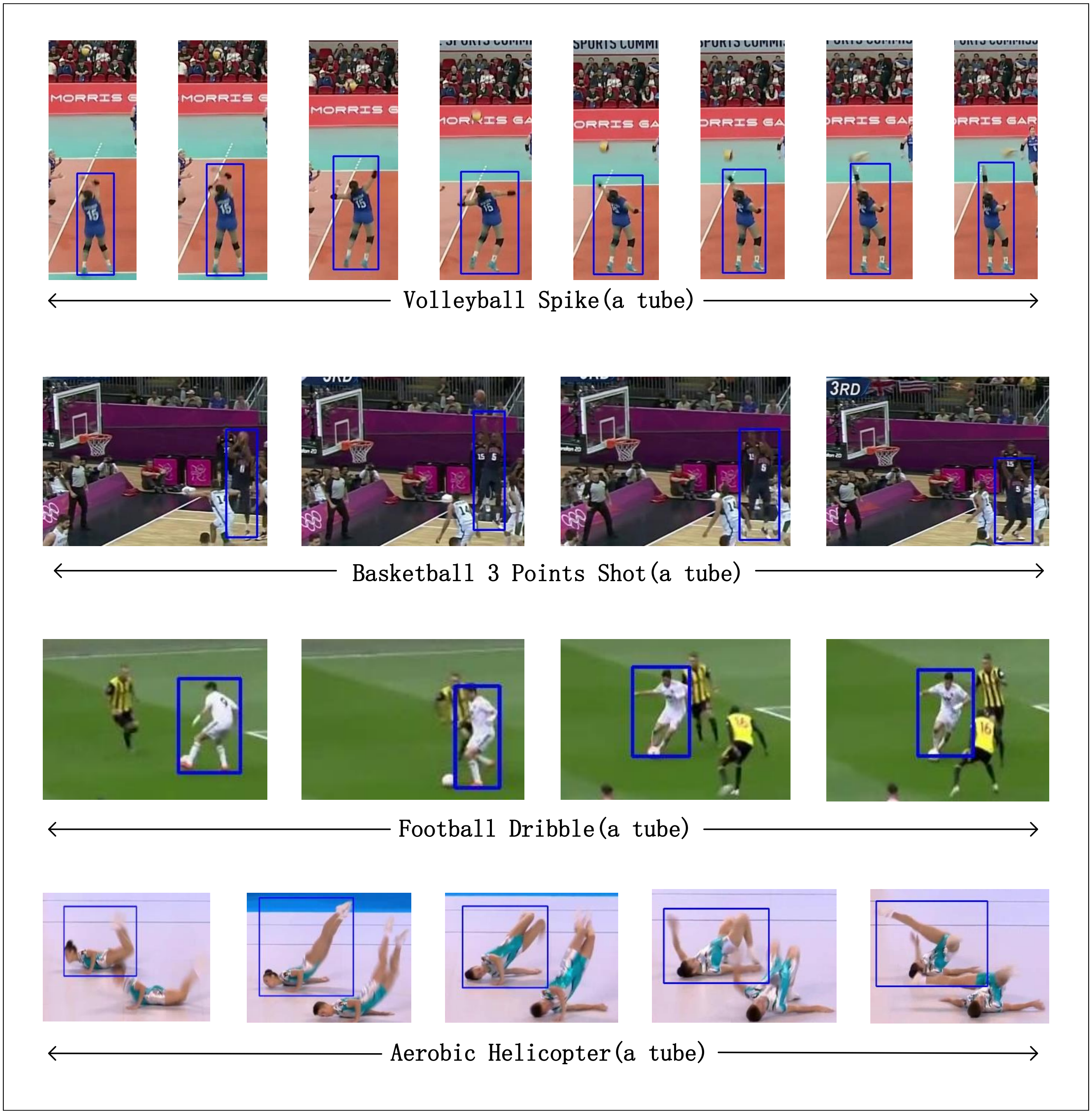
Volleyball Spike (a tube)
Basketball 3 Points Shot (a tube)
Football Dribble (a tube)
Aerobic Helicopter (a tube)

Figure 3. **Visualization of the results of 4 samples**.