Supplementary Material of Point3D: tracking actions as moving points with 3D CNNs

Shentong Mo*[1]
shentonm@andrew.cmu.edu
Jingfei Xia*[1]
jingfeix@andrew.cmu.edu
Xiaoqing Tan[1,2]
xit31@pitt.edu
Bhiksha Raj†[1]
bhiksha@cs.cmu.edu

[1] Carnegie Mellon University
Pittsburgh, PA 15213, United States

[2] University of Pittsburgh
Pittsburgh, PA 15260, United States

In this supplementary material, we provide more experimental results, ablation studies, and visualizations to evaluate the performance of our Point3D in a comprehensive manner.

1 Experimental Results

In this section, we closely follow [2, 4] to conduct an error analysis on the frame mAP in order to better explore our proposed Point3D. Specifically, we investigate five kinds of action detection errors as described in [2], which are localization errors (EL), classification errors (EC), time errors (ET), other errors (EO), and missed detection errors (EM). Among these action detection errors, EL, EC, ET, and EO identify false positive detection and we follow the calculation of frame mAP and measure the area under the curve when plotting the percentage of each category at all recall values. On the other hand, EM refers to the actions that we fail to detect at all. It is computed by measuring the percentage of ground truth boxes for which there are no correct detections. We report the results for $T = 1, 7, 16$ on the JHMDB and UCF101-24 dataset in Figure 1.

From Figure 1, we make the following observations: *First*, when experimenting with $T = 7$ on the JHMDB dataset, we achieve lower EL, EC, EO, and EM than the MOC [4] by 0.09%, 5.78%, 0.07%, and 0.20%, respectively. Our Point3D outperforms the MOC by 6.14% in terms of frame-mAP. Similar improvements can be seen on the UCF101-24 dataset. This shows the advantage of our Point3D in spatio-temporal action recognition. *Second*, with the increase of T, i.e., the length of the input clip, the frame-mAP increases and all the errors except ET decreases. This agrees with the common sense that the length of the input clip is crucial for action recognition. But there is a trade-off between the precision and the speed using different Ts. *Third*, the increasing gap from $T = 1$ to $T = 7$ is insignificant compared to the increasing magnitude from $T = 1$ and $T = 7$. Thus, we set $T = 7$ in our case for a better trade-off between the precision and the speed. From the error analysis, we can observe that our classification error EC (see the blue bar in Figure 1 is still high. Our Point3D sometimes

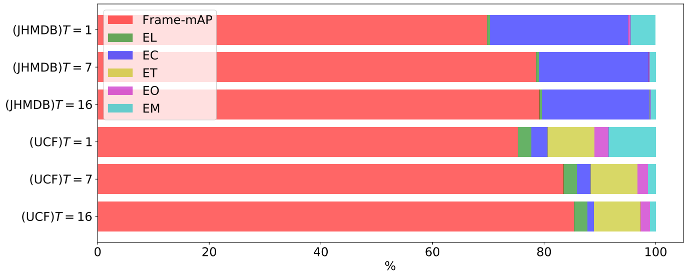

Figure 1: Error analysis of our Point3D for $T = 1, 7, 16$ on the JHMDB and UCF101-24 dataset. We show frame-mAP and different sources of error.

makes some false positive classifications at initial frames since it is hard to recognize the action before it happens.

Following previous work [4] closely, we also evaluate the two-stream offline speed of our Point3D's on a single Tesla V100 GPU. Point3D reaches a competitive speed of 20 fps compared with existing 2D detectors. In Figure 2, we compare our Point3D with some current state-of-the-art methods which have reported their speed in the original paper [2, 3, 4, 6, 8, 10, 12]. Point3D achieves the best performance against the existing methods in terms of video mAP within reasonable runtime. This further confirms the advantages of our Point3D in tracking action as moving points with 3D-CNNs.

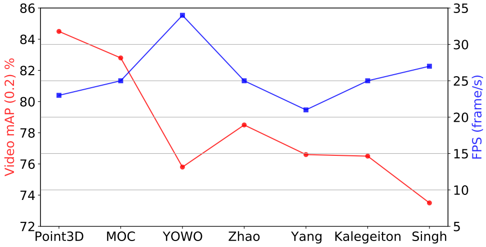

Figure 2: Runtime comparison with the state-of-the-art methods using video mAP with a threshold of 0.2 and FPS of frame per second. A high video mAP (in red) and a small FPS (in blue) indicates a better performance. Results of [6] are omitted as their video mAP is much lower compared to other methods.

2 Ablation Study

Input of 3D Head. In this ablation study, we explore three types of 3D Head input, including the raw clip, frame heatmaps and the output from Point Head, as shown in Table 1. Output from the Point Head achieves better performance against the raw clip and the heatmap alone, which shows the advantage of the Point Head. Combining two of three input types can improve the performance further. Especially, using the raw clip and the output from Point Head achieves the best performance among all input combinations in terms of four video-mAP metrics. Feeding all three types of input into the 3D Head achieves the best frame-mAP, which further demonstrate the flexibility of the proposed Point3D architecture.

[*] These authors contributed equally to this work. [†] Corresponding author.

Table 1: Exploration study on the input of 3D Head.

Raw clip	Heatmap	Point Head output	Frame-mAP(%)	Video-mAP(%)			
			0.5	0.2	0.5	0.75	0.5:0.95
✓			76.1	87.4	84.5	69.6	58.7
	✓		77.9	87.9	84.8	69.9	59.1
		✓	79.2	89.1	86.1	71.5	60.9
✓	✓		78.1	89.0	85.2	70.3	59.4
	✓	✓	79.4	89.3	86.2	71.7	61.1
✓		✓	79.5	**89.4**	**86.5**	**71.8**	**61.3**
✓	✓	✓	**79.6**	89.2	86.1	71.5	61.0

3D Head backbone. Furthermore, we explore the design of 3D Head backbone, as shown in Table 2. Specifically, we employ the 3D backbone with 3D-ResNet with different depths [9] and with other popular 3D-CNN architectures such as MobileNets [1, 7] and ShuffleNet [5, 11]. We can observe that the stronger 3D Head backbone we use, better the achieved results. This further demonstrates the importance of the proposed 3D Head in our Point3D. Moreover, applying the light-weight backbone in our 3D Head achieves the competitive performance compared to the MOC [4].

Table 2: Exploration study on the design of 3D Head backbone.

3D backbone	FPS	Frame-mAP(%)	Video-mAP(%)			
		0.5	0.2	0.5	0.75	0.5:0.95
MobileNetV1 2.0x	**31**	67.5	72.6	71.8	64.9	53.8
MobileNetV2 2.0x	**31**	69.1	75.5	73.6	66.8	55.6
ShuffleNetV1 2.0x	30	70.2	77.6	75.8	67.5	56.4
ShuffleNetV2 2.0x	30	70.8	78.9	77.5	68.1	57.6
ResNet-18	27	72.6	80.4	78.5	68.4	58.3
ResNet-50	25	74.5	82.5	80.3	68.7	58.8
ResNet-101	20	77.4	86.5	84.2	69.4	59.8
ResNeXt-101	23	**79.2**	**89.1**	**86.1**	**71.5**	**60.9**

Weight of localization and classification loss. In Table 3, we ablate λ_{loc} and λ_{cls}, two important parameters in our Point3D to balance the weight of Point Head and 3D Head in the overall loss $L_{overall}$. We see that the performance gap among different settings are small, which further shows the robustness of our Point3D to the weight hyper-parameters. In our case, we set $\lambda_1 = 10$ and $\lambda_2 = 1$ to achieve the best performance.

Table 3: Exploration study on λ_{loc} and λ_{cls}.

λ_{loc}	λ_{cls}	Frame-mAP(%)	Video-mAP(%)			
		0.5	0.2	0.5	0.75	0.5:0.95
1	1	79.2	89.1	86.1	71.5	60.9
1	5	79.3	89.2	86.2	71.6	61.1
1	10	79.5	89.4	86.4	71.8	61.4
1	15	79.1	88.9	85.9	71.3	60.8
5	1	79.7	89.5	86.4	71.7	61.3
10	1	**79.9**	**89.8**	**86.8**	**71.9**	**61.5**
15	1	79.6	89.4	86.3	71.6	61.2

3 Visualizations

We provide more qualitative examples of action recognition on the JHMDB and UCF101-24 datasets to demenstrate the effectiveness of our proposed Point3D. In general, our Point3D architecture exhibits satisfactory performance at localizing and classifying actions in videos. As can be seen in the first row of Figure 3, the heatmaps generated from our Knot-Point (KP) detector track spatial and temporal changes of the action from frame to frame, without needing any anchor boxes. For true positive examples, the actions are correctly classified and the bounding boxes predicted from Point3D are nearly identical to the ground truth bounding boxes. For false positive examples, our results show even though the action is misclassified, the localization result is still robust. We provide more visualization examples using videos in JHMDB and UCF101-24 in Figure 4 and Figure 5, respectively.

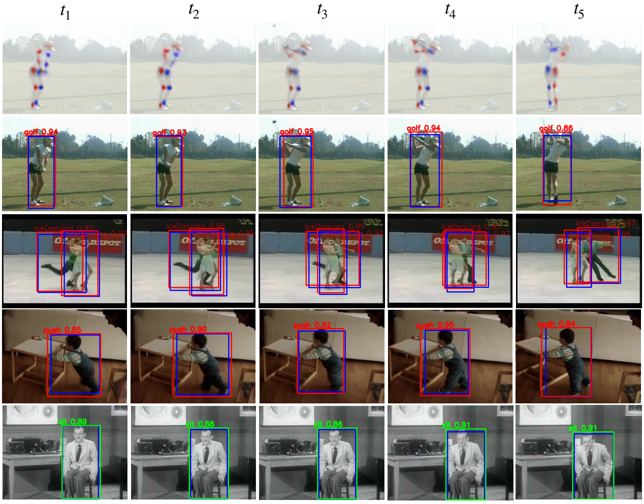

Figure 3: **Visualization results of how our Point3D conducts action recognition and some examples from JHMDB and UCF101-24.** The first row denotes the heatmaps generated from our KP detector, tracking spatial and temporal changes of the action from frame to frame. The second, third, and fourth rows show examples of true positive detections. The last row shows an false positive detection example that the action “stand up" is misclassified into “sit" but the localization of this action is still robust. The **blue** bounding boxes are ground truths while the **red** and **green** boxes are true and false positive detections, respectively. Zoom in for a better view.

4 Codes to replicate the experiments

We attach the codes to replicate our experiments and the usage of Point3D as part of supplementary material.

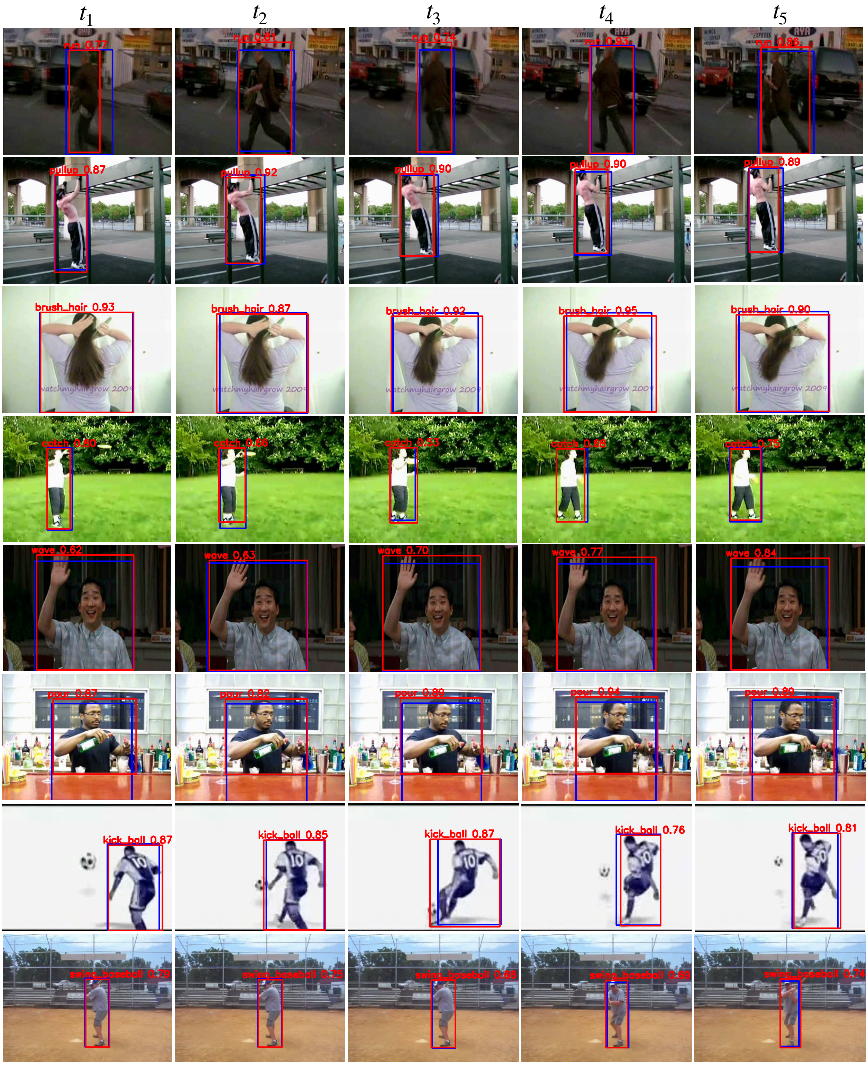

Figure 4: **Visualization results of examples from JHMDB**. The **blue** bounding boxes are ground truths while the **red** are true positive detections, respectively. The actions from top to bottom are "run", "pullup", "brush hair", "catch", "wave", "pour", "kick ball", and "swing baseball". Zoom in for a better view.

References

[1] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[2] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Ac-

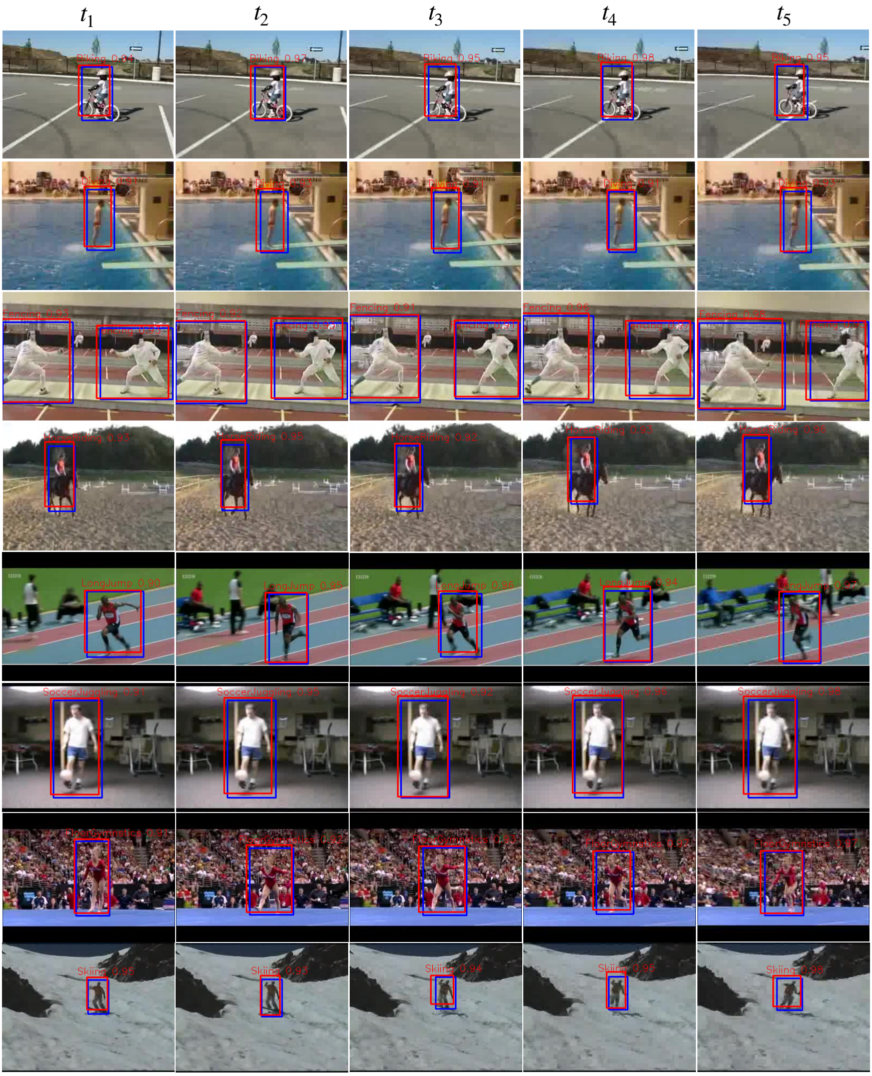

Figure 5: **Visualization results of examples from UCF101-24**. The **blue** bounding boxes are ground truths while the **red** are true positive detections, respectively. The actions from top to bottom are "Biking", "Diving", "Fencing", "Horse Riding", "Long Jump", "Soccer Juggling", "Floor Gymnastics", and "Skiing". Zoom in for a better view.

tion tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4405–4413, 2017.

[3] 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.

[4] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In

Proceedings of the European Conference on Computer Vision, pages 68–84, 2020.

[5] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European Conference on Computer Vision*, pages 116–131, 2018.

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

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetv2: inverted residuals and linear bottlenecks. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.

[8] 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/CVF International Conference on Computer Vision*, pages 3637–3646, 2017.

[9] Sasha Targ, Diogo Almeida, and Kevin Lyman. Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*, 2016.

[10] 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.

[11] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6848–6856, 2018.

[12] 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.