

Chained Multi-stream Networks Exploiting Pose, Motion, and Appearance for Action Classification and Detection (Supplementary Material)

Mohammadreza Zolfaghari , Gabriel L. Oliveira, Nima Sedaghat, and Thomas Brox
University of Freiburg
Freiburg im Breisgau, Germany

{zolfagha,oliveira,nima,brox}@cs.uni-freiburg.de

1. Network inputs

Figures 2 and 3 show sample frames of network inputs on different actions from J-HMDB [3], UCF101 [6], and HMDB51 [4] datasets.

2. Score aggregation

For action classification, there are many approaches to aggregate features or scores obtained during the evaluations. In [1], they reported the performance of different feature aggregation schemes. We analyzed two different augmentation and three fusion methods to yield the final label for each video at test time.

The two augmentation methods are (1) crop from the center of the frame, and horizontal flipping (**center-cropping**); (2) crop from the center and from 4 corners, and horizontal flipping (**fixed-cropping**). For fusion we used **mean**, **max**, and **majority voting** methods to calculate final scores.

In Table 1, we provide the result of action recognition on all datasets with different score fusion and augmentation schemes. Like [7], we observed that mean fusion in general provides better results than the other two fusion methods. In addition, fixed-cropping led to an increase of performance.

Since actions can span various time intervals, we analyzed videos at multiple temporal scales. In all networks, the length of the input clips was 16 frames, but for Net_{16} we used a sampling rate of 1, for Net_{32} the sampling rate is 2, and Net_W refers to selecting 16 frames randomly from the entire video. We use the combined score for our multi-grained action classification experiment in the main paper. A comparison of the single time scales is shown in Table 1.

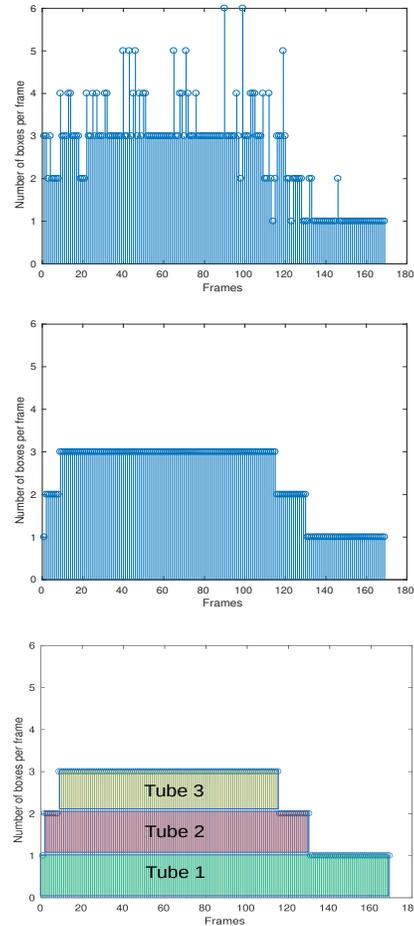


Figure 1: Preprocessing and extraction of the action tubes. The top figure shows the initial number of detected boxes per frame, N_t , the middle one shows the denoised version of it, \tilde{N}_t (section 3.2), and the bottom figure shows the corresponding tube proposals.

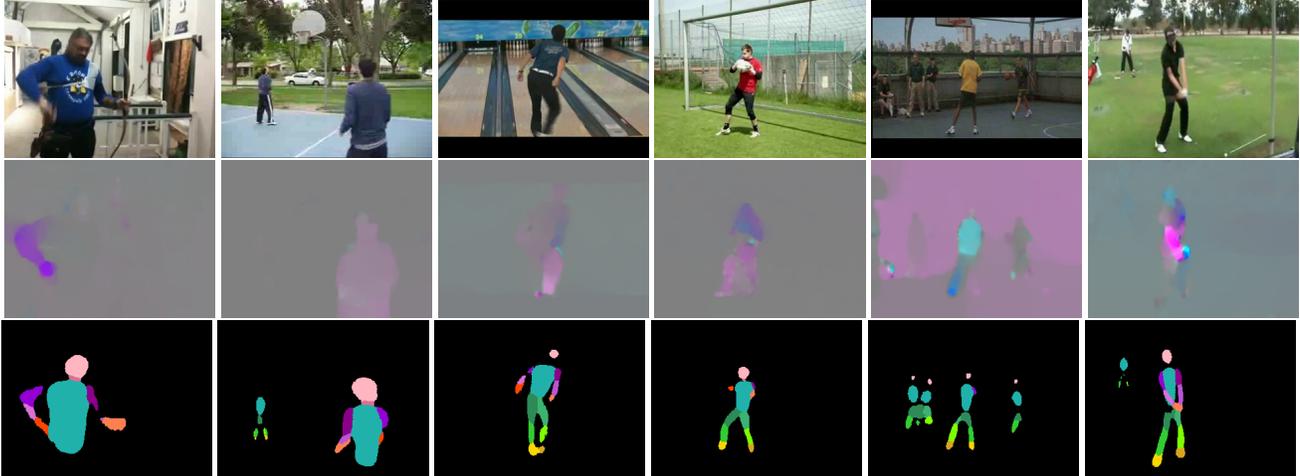


Figure 2: Network inputs. **First row:** Raw image. **Second row:** Optical flow. **Third row:** Pose.



Figure 3: Network inputs. **First row:** Raw image. **Second row:** Optical flow. **Third row:** Pose.

3. Action Detection

3.1. Spatial localization

We obtain human body part segmentation by our pose estimation network and simply generate the bounding box from the human body parts. We report quantitative results on J-HMDB dataset in the main paper. Here we additionally show some qualitative results on J-HMDB and UCF101 in Figure 4.

3.2. Spatio-temporal localization

Preprocessing To mitigate the effect of noise from our detections (false human detections, missed human instances), we define the *expected* number of boxes per each frame of the input video clip, based on the expected arrangement of the action-tubes. To this end, we define signal N_t , which represents the number of detected boxes versus frame

number – Figure 1. Then we use a median filter of size 80 to obtain a smoother version of it, \hat{N}_t , as in Figure 1. The final number of expected boxes in each frame is computed as follows:

$$N_t^* = \max(N_t, \hat{N}_t) \quad (1)$$

Whenever $N_t < N_t^*$ we simply create duplicate boxes from the box with the biggest size in that frame. Possible extra boxes are left intact and the box tracking algorithm that follows, picks the appropriate ones to use.

Iterative Extraction of Action Tubes We define action tube proposals in the input video, based on the continuous regions in \hat{N}_t – Figure 1. We start with the longest one, trim the video to include only its time-span, and feed the obtained video clip to the Box Tracking algorithm. The output is a single action tube in the specified time span, and the

Datasets	Networks	Center-Cropping			Fixed-Cropping		
		mean	max	MajorityVoting	mean	max	MajorityVoting
UCF101	Net ₁₆	87.9%	87.6%	85.4%	90.4%	89.8%	89.7%
	Net ₃₂	87.9%	87.3%	86.9%	90.3%	89.7%	89.5%
	Net _W	86.6%	86.8%	86.4%	89.6%	89.0%	88.8%
HMDB51	Net ₁₆	57.6%	56.7%	56.8%	62.1%	60.1%	60.3%
	Net ₃₂	58.7%	59.3%	58.2%	62.8%	61.1%	61.5%
	Net _W	62.6%	62.3%	62.0%	66.0%	65.9%	65.5%
J-HMDB	Net ₁₆	81.7%	76.1%	78.4%	79.1%	78.7%	74.3%
NTU RGB+D [5]	Net ₁₆	80.1%	77.2%	79.6%	80.8%	80.3%	76.6%

Table 1: Comparison of the performance of different score fusion and augmentation methods for action recognition on the UCF101, HMDB51, and J-HMDB datasets (split 1) and NTU RGB+D dataset (Cross subject).



Figure 4: Qualitative results. The first two rows correspond to detections on J-HMDB, the last ones on UCF101. In each row, the last two columns shows the failed detection examples. Ground truth bounding boxes are shown in green and detections in red.

detected boxes are removed from the initial set. Tube proposals of length less than 5 frames are ignored. We repeat these steps until no tube proposal is left.

Box Tracking For each action a in the video, we define the following optimization function:

$$\bar{B}_a = \arg \max_{\bar{B}} \frac{1}{T} \sum_{t=t_e}^{t_s} S_a(b_t, b_{t+1}) \quad (2)$$

where $\bar{B}_a = [b_s, b_{s+1}, \dots, b_e]$ is the sequence of linked detection boxes from frame s to frame e , T is the tube length and $S_a(b_t, b_{t+1})$ denotes the linking score between consecutive frames t and $t + 1$. In our framework, S_a is the IoU

overlap of the two detection boxes. We can find the path by solving the above optimization problem using Viterbi algorithm [2]. After finding the optimal path, we remove all boxes in \bar{B}_a and repeat this for finding the next optimal path until the set of boxes is empty.

Temporal Actionness Score As discussed in section 5.3 of the main paper, we calculate an actionness score for the frames of each video, based on the network outputs. More specifically, we leverage the scores obtained from the three streams as follows:

$$s_{total} = (s_{pose})^2 \cdot (s_{rgb})^{1/3} \cdot (s_{flow})^{1/3} \quad (3)$$

where s_{pose} , s_{rgb} and s_{flow} are the outputs of the softmax layers. We sum the per-frame scores over each single tube, to assign an overall actionness score to each tube.

References

- [1] G. Chéron, I. Laptev, and C. Schmid. P-CNN: Pose-based CNN Features for Action Recognition. In *ICCV*, 2015. 1
- [2] G. Gkioxari and J. Malik. Finding action tubes. 2015. 3
- [3] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black. Towards understanding action recognition. In *International Conf. on Computer Vision (ICCV)*, pages 3192–3199, 2013. 1
- [4] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2011. 1
- [5] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 3
- [6] k. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 1
- [7] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Val Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016. 1