Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors

Limin Wang[1,2], Yu Qiao[2], Xiaoou Tang[1,2]
[1]Department of Information Engineering, The Chinese University of Hong Kong. [2]Shenzhen Institutes of Advanced Technology, CAS, China.

Visual features are of vital importance for human action understanding in videos. Currently, there are mainly two types of video features available for action recognition. The first type of representations are the *hand-crafted* local features, and typical local features include STIPs, Cuboids, Dense Trajectories, and Improved Trajectories. Among these local features, improved trajectories [5] with rich descriptors of HOG, HOF, MBH have shown to be successful on a number of challenging datasets and contests. Improved trajectories includes several important ingredients in their extraction process. First, these extracted trajectories are mainly located at regions with high motion salience. Second, these local descriptors of the corresponding regions in several successive frames, are aligned and pooled along the trajectories. However, these hand-crafted descriptors may lack discriminative capacity for action recognition. The second type of representations are the *deep-learned* features and Two-Stream ConvNets [4] are probably the most successful architecture at present. They are composed of two neural networks, namely spatial nets and temporal nets. Spatial nets mainly capture the discriminative appearance features for action understanding, while temporal nets aim to learn the effective motion features. However, most of current deep learning based action recognition methods largely ignore the intrinsic difference between temporal domain and spatial domain, and just treat temporal dimension as feature channels when adapting the architectures of ConvNets to model videos.

Motivated by the above analysis, this paper proposes a new kind of video feature, called *trajectory-pooled deep-convolutional descriptor* (TDD). The design of TDD aims to combine the benefits of both hand-crafted and deep-learned features. To achieve this goal, our approach integrates the key factors from two successful video representations, namely improved trajectories [5] and two-stream ConvNets [4]. We utilize deep architecture to learn multi-scale convolutional feature maps, and introduce the strategies of trajectory-constrained sampling and pooling to encode deep features into effective representations. As shown in Figure 1, we first train two-stream ConvNets on a relatively large dataset, while more labeled action videos will make ConvNet training more stable and robust. Then, we treat the resulted two-stream ConvNets as generic feature extractors, and use them to obtain multi-scale convolutional feature maps for each video. Meanwhile, we detect a set of point trajectories with the method of improved trajectories. Based on convolutional feature maps and improved trajectories, we pool the local ConvNet responses over the spatiotemporal tubes centered at the trajectories, where the resulting descriptor is called TDD.

To enhance the robustness of TDDs, we apply the normalization strategy to the convolutional feature maps of two-stream ConvNets to suppress the activation burstiness of some neurons. We design two kinds of normalization methods:

- *Spatiotemporal Normalization.* For spatiotemporal normalization, we normalize the feature map for each channel independently across the video spatiotemporal extent. The spatiotemporal normalization method ensures that each convolutional feature channel ranges in the same interval, and thus contributes equally to final TDD recognition performance.
- *Channel Normalization.* For channel normalization, we normalize the feature map for each pixel independently across the feature channels. This channel normalization is able to make sure that the feature value of each pixel range in the same interval, and let each pixel make the equal contribution in the final representation.

In order to verify the effectiveness of TDDs, we conduct experiments on two public large datasets, namely HMDB51 and UCF101. Table 1 shows the performance of TDDs of different layers and different nets and Table 2 compares our recognition results with several recently published methods. On the HMDB51 dataset, our best result outperforms other methods by 4.8%, and on the UCF101 dataset, our best result outperforms by 3.5%. This superior performance of TDDs indicates the effectiveness of introducing trajectory-constrained sampling and pooling into deep-learned features.

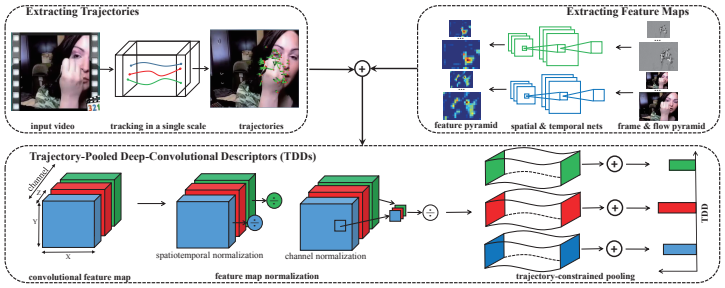

Figure 1: **Pipeline of TDD.** The whole process of extracting TDD is composed of three steps: (i) extracting trajectories, (ii) extracting multi-scale convolutional feature maps, (iii) calculating TDD.

Algorithm	HMDB51	UCF101
HOG	40.2%	72.4%
HOF	48.9%	76.0%
MBH	52.1%	80.8%
HOF+MBH	54.7%	82.2%
iDT	**57.2%**	**84.7%**
Spatial net	40.5%	73.0%
Temporal net	54.6%	83.7%
Two-stream ConvNets	**59.4%**	**88.0%**
Spatial conv4	48.5%	81.9%
Spatial conv5	47.2%	80.9%
Spatial conv4 and conv5	**50.0%**	**82.8%**
Temporal conv3	54.5%	81.7%
Temporal conv4	51.2%	80.1%
Temporal conv3 and conv4	**54.9%**	**82.2%**
TDD	63.2%	90.3%
TDD and iDT	**65.9%**	**91.5%**

Table 1: Performance of TDD on the HMDB51 and UCF101 dataset. We compare our proposed TDD with iDT features and two-stream ConvNets.

[1] Zhuowei Cai, Limin Wang, Xiaojiang Peng, and Yu Qiao. Multi-view super vector for action recognition. In *CVPR*, 2014.

[2] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[3] Xiaojiang Peng, Limin Wang, Xingxing Wang, and Yu Qiao. Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice. *CoRR*, abs/1405.4506, 2014.

[4] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *NIPS*, 2014.

[5] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[6] Limin Wang, Yu Qiao, and Xiaoou Tang. Motionlets: Mid-level 3D parts for human motion recognition. In *CVPR*, 2013.

The TDD code and learned two-stream ConvNet models are available at https://wanglimin.github.io/tdd/index.html

HMDB51		UCF101	
Motionlets [6]	42.1%	Deep Net [2]	63.3%
DT+MVSV [1]	55.9%	DT+MVSV [1]	83.5%
iDT+FV [5]	57.2%	iDT+FV [5]	85.9%
iDT+HSV [3]	61.1%	iDT+HSV [3]	87.9%
Two Stream [4]	59.4%	Two Stream [4]	88.0%
TDD+FV	63.2%	TDD+FV	90.3%
Our best result	**65.9%**	Our best result	**91.5%**

Table 2: Comparison of TDD to the state of the art.

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.