

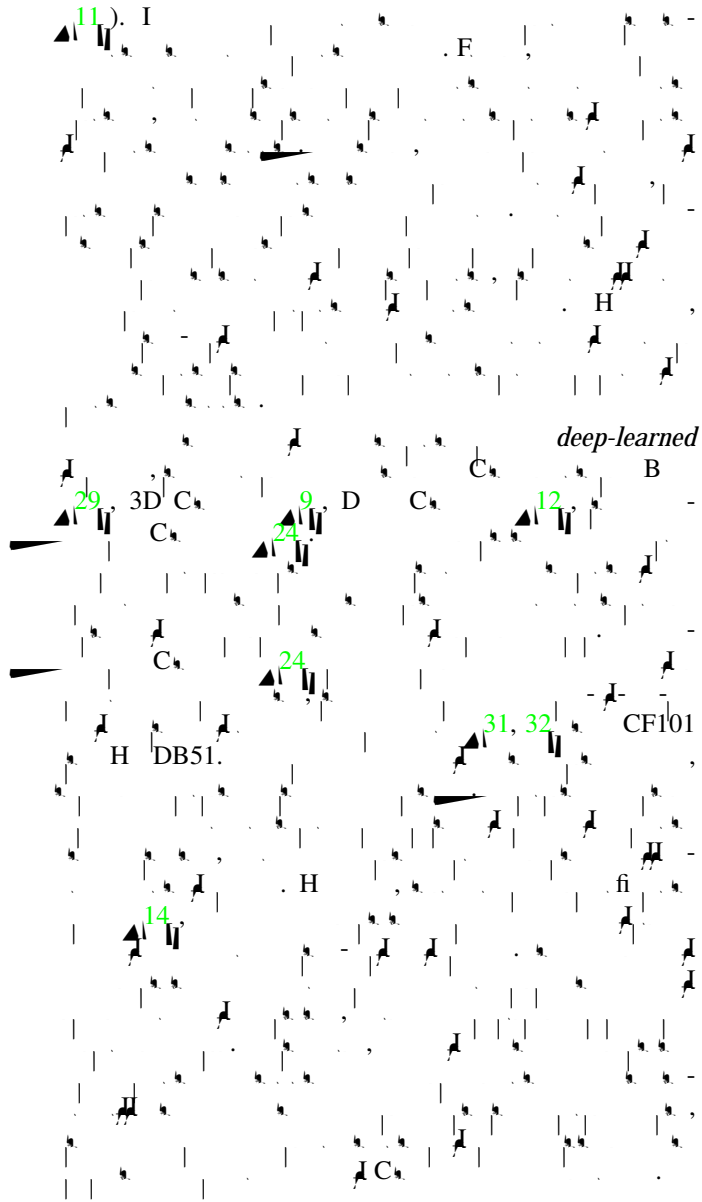
# Action Recognition with Trajectory-Pooled Deep-Convolutional Descriptors

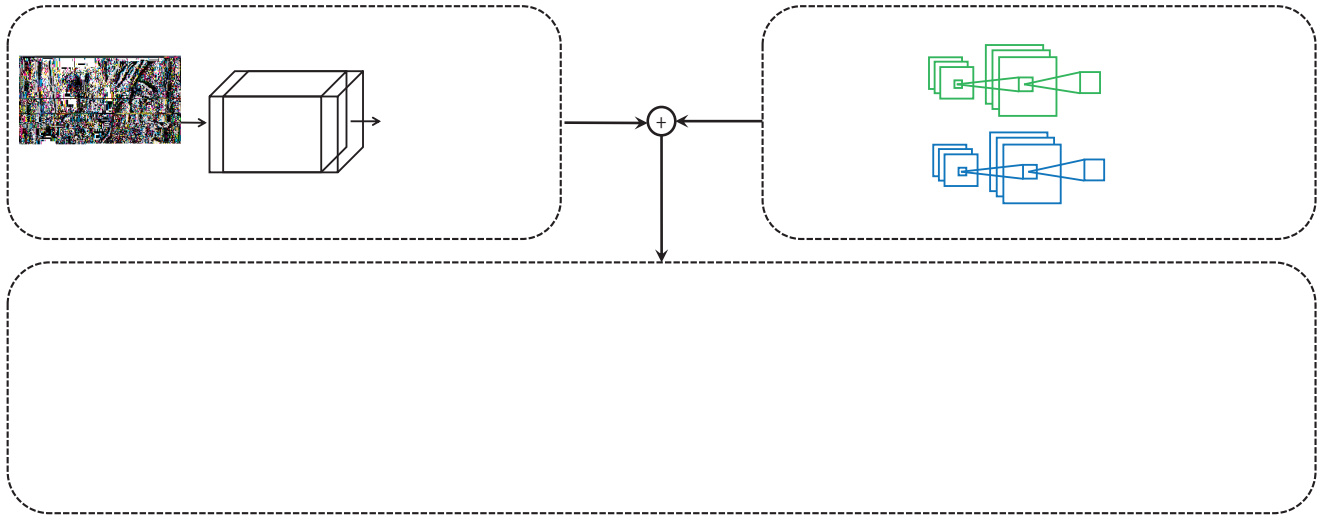
Wanglimin<sup>1</sup>, Qiao<sup>1,2</sup>, Tang<sup>1,2</sup>  
07wanglimin@gmail.com, yu.qiao@siat.ac.cn, xtang@ie.cuhk.edu.hk

## Abstract

Visual features are of vital importance for human action understanding in videos. This paper presents a new video representation, called

(TDD), which shares the merits of both hand-crafted features [31] and deep-learned features [24]. Specifically, we utilize deep architectures to learn discriminative convolutional feature maps, and conduct trajectory-constrained pooling to aggregate these convolutional features into effective descriptors. To enhance the robustness of TDDs, we design two normalization methods to transform convolutional feature maps, namely spatiotemporal normalization and channel normalization. The advantages of our features come from (i) TDDs are automatically





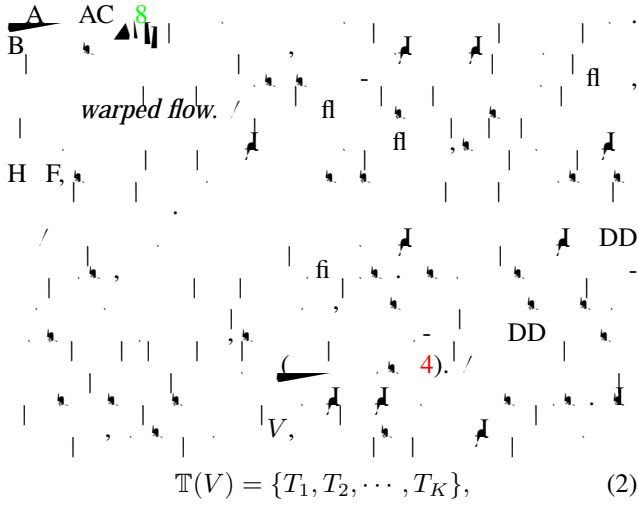
**2. Pipeline of TDD.**

DD. / G. (.) (.)

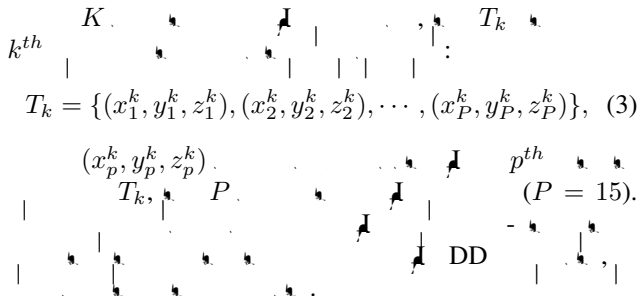
14, 25, 28, 41

9, 12, 24,

29. et al. 29



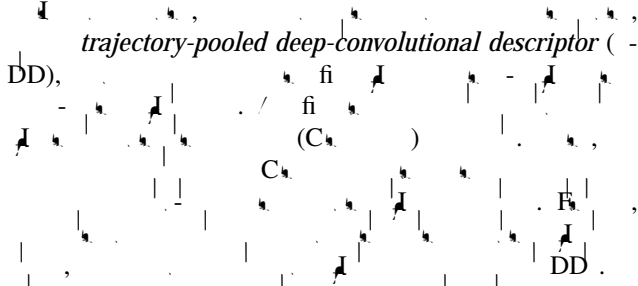
$$\mathbb{T}(V) = \{T_1, T_2, \dots, T_K\}, \quad (2)$$



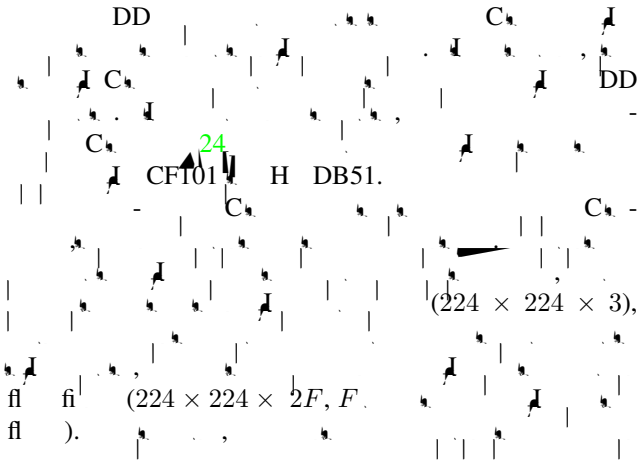
$$T_k = \{(x_1^k, y_1^k, z_1^k), (x_2^k, y_2^k, z_2^k), \dots, (x_P^k, y_P^k, z_P^k)\}, \quad (3)$$

$$(x_p^k, y_p^k, z_p^k) \quad T_k \quad P \quad p^{th} \quad (P = 15).$$

#### 4. Deep Convolutional Descriptors



##### 4.1. Convolutional networks



	1	1	2	2	3	4	5	5	6	7	8
	7 × 7	3 × 3	5 × 5	3 × 3	3 × 3	3 × 3	3 × 3	3 × 3	-	-	-
	2	2	2	2	1	1	1	2	-	-	-

$(x_p^k, y_p^k, z_p^k)$ ,  $p^{th}$   
 $T_k, r_m$ ,  $m^{th}$   
 $1, (\cdot)$   
 $D(T_k, \tilde{C}_m^a)$   
*trajectory-pooled*  
*deep convolutional descriptor,*

**Multi-scale TDD extension.**

DD DD, F  
 fi  $32 \times 32$  H G, H F, BH  
 fi  
 fi  
 C, F, B  
 DD  
 $r_m$  E (7)  $r_m \times s$ , s  
 $s = 1/2, 1/\sqrt{2}, 1, \sqrt{2}, 2$

**5. Experiments**

**5.1. Datasets**

DD, H D-  
 B51 15 CF101 26 H DB51  
 6, 766 51 100  
 70 30  
 CF101 100 101

13, 320  
 25  
 CF101 H DB51, CF101  
 H DB51  
 C, DD H DB51

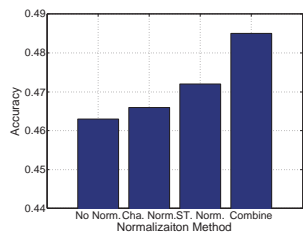
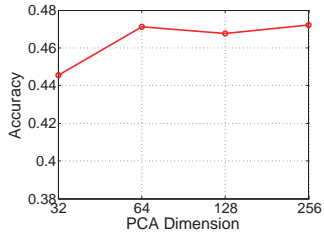
**5.2. Implementation details**

**Two-stream ConvNets training**

C, CF101 1  
 C, CF101 10  
 (256) (0.9), F 256, I  
 $224 \times 224$  fi  
 $10^{-2}$ , CF101  $10^{-3}$   
 14K, 20K, 3D, 1, fi  
 F, fi, fi  
 40  
 C, fi, F, fi  
 0-255, 10, CF101  
 A, 0.9, 6, 0.8, 7  
 $224 \times 224 \times 20$   
 $10^{-2}$ ,  $10^{-3}$ , 50K, I  
 $10^{-4}$ , 70K, 90K

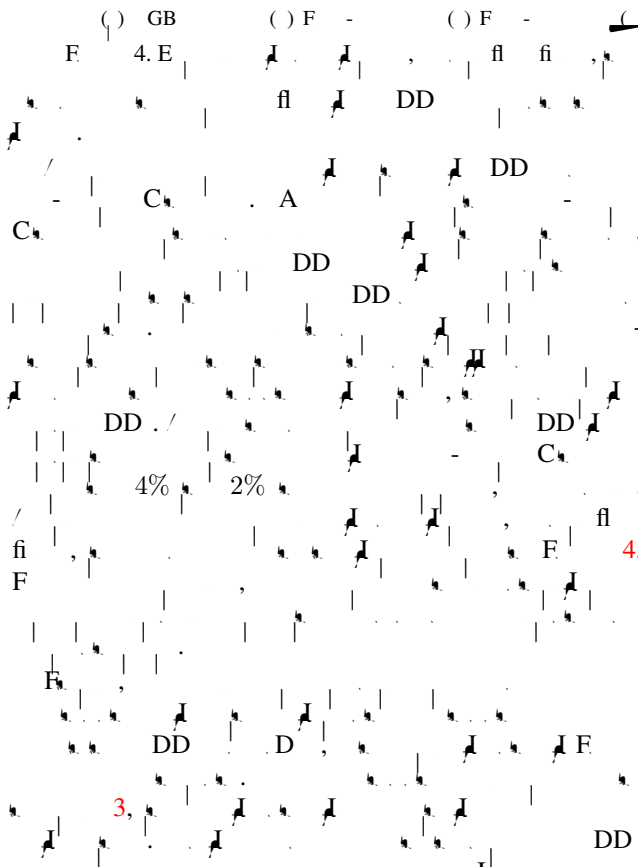
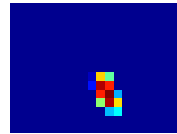
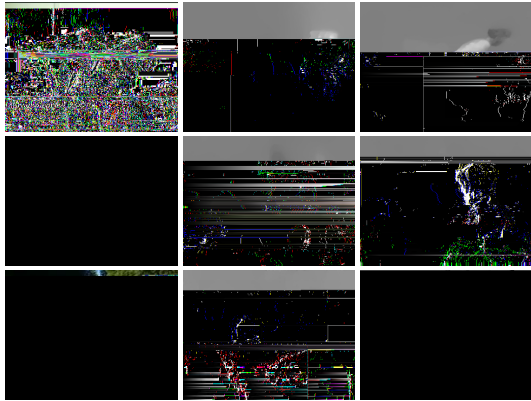
**Results of two-stream ConvNets.**

24, 25  
 10  
 80.1% 71.2%  
 C, C, 84.7%,  
 24 (85.6%)



F 3.

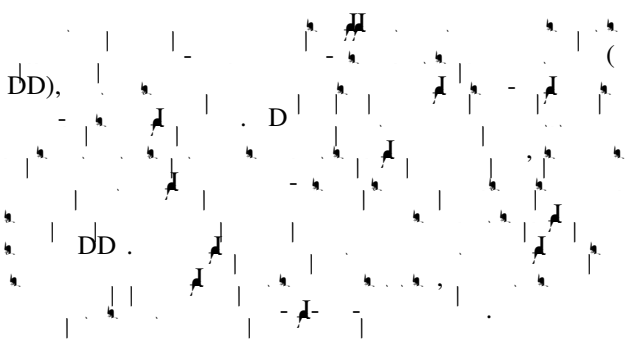
C <sub>1</sub>	C <sub>2</sub>					C <sub>3</sub>				
	1	2	3	4	5	1	2	3	4	5
	24.1%	33.9%	41.9%	<b>48.5%</b>	<b>47.2%</b>	39.2%	50.7%	<b>54.5%</b>	<b>51.2%</b>	46.1%



H DB51		CF101	
I + B	23.0%	I + B	43.9%
D + B	42.1%	D	63.3%
D +	46.6%	D + AD	79.9%
D + F	55.9%	D +	83.5%
D + H	57.2%	D + F	85.9%
DD + F	61.1%	D + H	87.9%
	59.4%	DD + F	88.0%
	63.2%		90.3%
	<b>65.9%</b>		<b>91.5%</b>

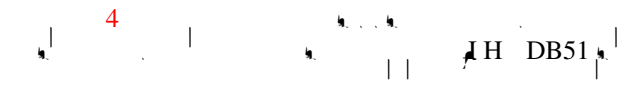


## 6. Conclusions



## Computational costs.

## 5.5. Comparison to the state of the art



## Acknowledgement

G. J. IDIA C. K40  
H. K. D F  
F. C. (91320101, 61472410)  
B. (JC J20120903092050890, J-  
C J20120617114614438, JC J20130402113127496), 100  
I. CA, G.  
(.201001D0104648280).

## References

1. J. K. A. H. : A  
*ACM Comput. Surv.*, 43(3):16, 2011. 1
2. H. B. J. G. F:  
*ECCV*, 2006. 3
3. C. :  
*CVPR*, 2014. 8
4. K. C. fi. K. , A. A. :  
*BMVC*, 2014. 6
5. D. B. H.

- ▲ 37. *ECCV*, 2014. 2
- ▲ 38. ACCV, 2012. 7
- ▲ 39. G. J. G. A. *ECCV*, 2008. 2
- ▲ 40. C. H. B. A. *29th DAGM Symposium on Pattern Recognition*, 2007. 6
- ▲ 41. D. F. *ECCV*, 2014. 3, 4, 5
- ▲ 42. J. B. *ICCV*, 2013. 2