Hybrid Super Vector with Improved Dense Trajectories for Action Recognition

Xiaojiang Peng[1,2], LiMin Wang[1,3], Zhuowei Cai[1], Yu Qiao[1,3], Qiang Peng[2]
[1]Shenzhen Key Lab of CVPR, Shenzhen Institutes of Advanced Technology, CAS, China
[2]Southwest Jiaotong University, Chengdu, P.R. China
[3]Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong
{xiaojiangp,07wanglimin,iamcaizhuowei}@gmail.com, yu.qiao@siat.ac.cn, qpeng@swjtu.edu.cn

Abstract

With recent improved dense trajectory features (HOG, warped HOF, and warped MBH), we employ two advanced super vector methods, namely Fisher Vector (FV) and soft Vector of Locally Aggregated Descriptors (VLAD-K) to encode them separately. The two individual super vectors are concatenated into a Hybrid Super Vector, and a linear SVM classifier is used to predict labels. We achieve 87.46%[1] in average accuracy of the three training/testing splits on the UCF101 dataset.

1. Introduction

Human action recognition in videos has been an active research area in recent years due to its wide range of potential applications, such as smart video surveillance, video indexing, and human-computer interface [1].

The key point in action recognition is how to represent an action video. Approaches mainly include dynamic model based methods which apply statistical sequential models such as HMM and Bayesian network to describe the temporal states of actions [12], human pose based approaches which utilize pose structure information [7], global action template based approaches which construct global templates to capture appearance and motion information of the whole motion body [10], and local feature based approaches which mainly extract spatial-temporal cuboids with appearance and motion descriptors [5, 8].

The local feature based approaches are robust to noise and illumination changes, and it can work without background subtraction or complex body-part modeling. Due to the above properties, in our implementation for the competition of *THUMOS challenge* [4, 11], we first extract the improved dense trajectory features as Wang *et al.* [14], and then leverage two advanced super vector methods, namely Fisher Vector (FV) [9, 15] and soft Vector of Locally Aggregated Descriptors (VLAD-K) to encode them separately. The two individual vectors are concatenated into a Hybrid Super Vector, and a linear SVM classifier is used to predict labels.

[1]It is different from the officially released result on the challenge website due to misorder of four class indexes compared with official one. For convenience, we just use the order of folder-list in Linux system as our class index, but the resulting index is different from the officially released one on the website. Totally, four action indexes are misordered: "HammerThrow"class index = 36) is placed before "Hammering"(class index = 35) in our case, while they are placed in the opposite order in the official released class index. The same problem is with "JumpRope"(class index = 48) and "JumpingJack"(class index = 47).

2. Pipeline

The pipeline of our method is shown in Figure 1. First, we densely extract improved trajectory features using the recent released code from Wang *et al.* [14], which contains warped trajectories, HOG, warped HOF, and warped MBH. Then, we employ two advanced super vector methods, namely the improved Fisher Vector (FV) and soft Vector of Locally Aggregated Descriptors (VLAD-K) to encode them separately. The codebooks are generated by GMM and K-means, respectively. The two individual super vectors are concatenated into a Hybrid Super Vector (HSV), and a linear SVM classifier is used to predict labels. We repeat all the steps three times since there exists three training/testing splits, and we report 87.46% in mean average accuracy.

In our experiment, we find that feature preprocessing and the normalization of coding vectors are very important for good result. We detail them in the following sections.

3. Feature preprocessing

Three preprocessing techniques are applied in our experiments. For all the histogram-based descriptors, we square root each dimension after L1 normalization as in [14]. Then, we reduce the dimensionality to 20, 48, 54, 48, and 48 for improved trajectory, HOG, HOF, MBHx, and MBHy, respectively. After that, all the features are whitened (i.e., divided by square root PCA eigenvalues).

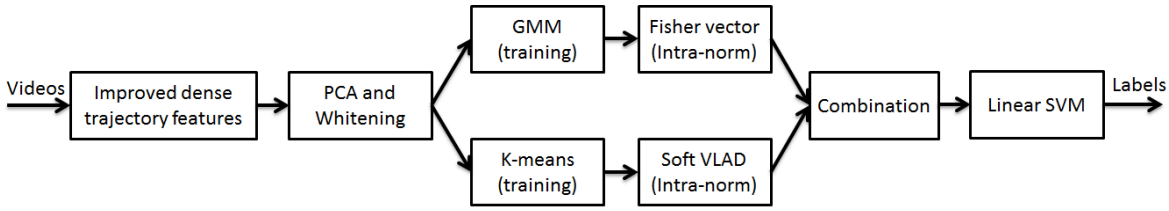

Figure 1. The pipeline of our method.

4. Encoding and fusion

Two different feature encoding methods are employed to all the features, namely FV [9] and VLAD-K. As for FV method, we train a GMM with 512 components within subsets of 500k descriptors for each type of features. Particularly, we use the `vl_gmm` command to learn GMM and leverage the `vl_fisher` command with the flag 'Improved' (i.e., root+L2 normalization) to encode all the features in the VLFeat toolbox [13], version 0.9.17. After encoding with `vl_fisher`, we apply intra-normalization [2] and L2 re-normalization which can efficiently suppress the "visual burstiness" problem, where each FV block (the first and second order blocks are separated) is L2 normalized separately.

As for the VLAD method [3], we first train a codebook with 512 words using K-means from the subsets of 500k descriptors for each type of features. We introduce a soft VLAD method: all the features are voted for k-nearest neighborhood words with weights like that in [6], where k is set to 5 empirically. Specially, we employ the `vl_vlad` command with intra-normalization and L2 re-normalization to implement, which supports soft voting scheme.

Finally, the two individual super vectors are concatenated into a Hybrid Super Vector, and one-vs-all linear SVM classifiers ($C = 100$) are used to predict labels.

5. Results

Table 1 shows our results of HSV and compares to that of VLAD-K and FV on UCF101 action datasets. The trajectory feature, HOG, HOF, MBHx, and MBHy are given the indexes from 1 to 5, respectively. The first column denotes the feature combination strategies. From the results, we observe that adding the trajectory feature would reduce the performance slightly, and there is a little complementary between the FV and VLAD-K representation. Our best result comes from the combination of the last four types of feature with our HSV representation. We report 87.46% in average accuracy.

Table 1. Performance of VLAD-K and FV with improved dense trajectory features on UCF101 action datasets.

Descs	Method	Split 1	Split 2	Split 3	Average
1-5	VLAD-K	85.04	85.70	86.81	85.85
	FV	85.46	87.14	87.26	86.62
	HSV	86.14	87.13	87.88	87.05
2-5	VLAD-K	85.41	86.20	86.87	86.16
	FV	85.85	87.80	87.93	87.19
	HSV	86.57	87.84	87.95	**87.46**

References

[1] J. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Computing Surveys (CSUR)*, 43(3):16, 2011.

[2] R. Arandjelovic and A. Zisserman. All about vlad. In *CVPR*, 2013.

[3] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, pages 3304–3311, 2010.

[4] Y.-G. Jiang, J. Liu, A. Roshan Zamir, I. Laptev, M. Piccardi, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/ICCV13-Action-Workshop/`, 2013.

[5] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, pages 1–8, 2008.

[6] L. Liu, L. Wang, and X. Liu. In defense of soft-assignment coding. In *ICCV*, pages 2486–2493, 2011.

[7] F. Lv and R. Nevatia. Single view human action recognition using key pose matching and viterbi path searching. In *CVPR*, pages 1–8, 2007.

[8] X. Peng, Y. Qiao, Q. Peng, and X. Qi. Exploring motion boundary based sampling and spatial-temporal context descriptors for action recognition. In *BMVC*, 2013.

[9] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*, pages 143–156, 2010.

[10] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, pages 1234–1241, 2012.

[11] 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.

[12] T. Starner and A. Pentland. Real-time american sign language recognition from video using hidden markov models. In *Motion-Based Recognition*, pages 227–243. 1997.

[13] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008.

[14] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[15] X. Wang, L. Wang, and Y. Qiao. A comparative study of encoding, pooling and normalization methods for action recognition. In *ACCV*, 2012.