LEAR-INRIA submission for the THUMOS workshop

Heng Wang and Cordelia Schmid
LEAR, INRIA, France
firstname.lastname@inria.fr

Abstract

This notebook paper describes the submission of the LEAR team from INRIA to the THUMOS workshop in conjunction with ICCV 2013. Our system is based on the recent improvement of dense trajectory feature [14]. After extracting the local features, we apply Fisher vector to integrate them into a compact representation for each video. We also use spatio-temporal pyramids to embed structure information. Finally a linear SVM with one-against-rest is employed for the multi-class action classification problem.

1. Introduction

Action recognition is an important area in computer vision and attracts lots of attention due to the large number of applications, such as, event analysis, video surveillance, human-computer interaction, *etc*. The THUMOS workshop [4] aims at action recognition in the wild with a large number of classes. This notebook paper describes the LEAR-INRIA submission. In section 2, we briefly describe the improved trajectory feature. The Fisher vector representation is explained in section 3 and experimental results in section 4.

2. Improved trajectories

Local space-time features are a successful representation for action recognition. Among the various local features, dense trajectories [13] perform best on a variety of datasets. The main idea is to densely sample feature points in each frame, and track them in the video based on optical flow. Multiple descriptors are computed along the trajectories of feature points to capture shape, appearance and motion information.

Recently, Wang and Schmid [14] further improved dense trajectories by estimating the camera motion explicitly. Here we follow exactly the same framework as in [14] and use the code from the website[1] to produce our results on UCF101 dataset [10]. We briefly review it in the following. For more details, please refer to [14].

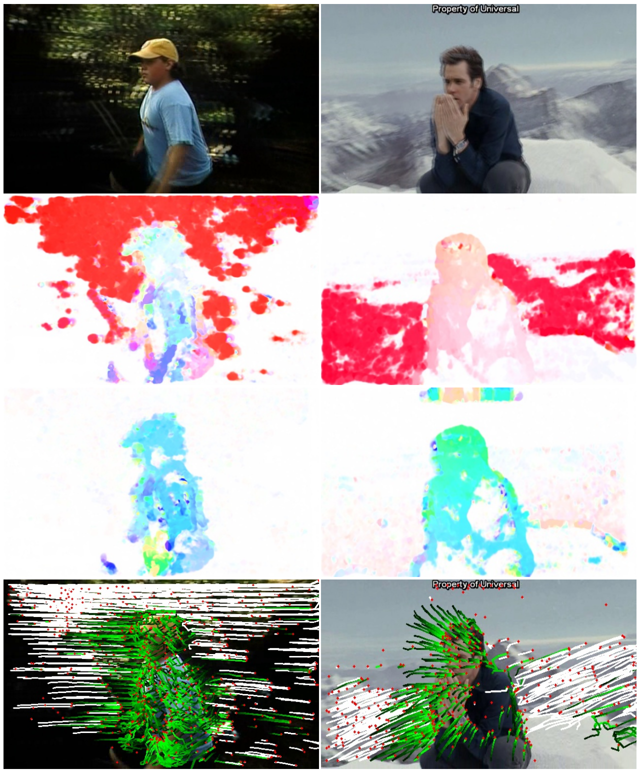

Figure 1. First row: images of two consecutive frames overlaid; second row: optical flow [3] between the two frames; third row: optical flow after removing camera motion; last row: trajectories removed due to camera motion in white.

To estimate the camera motion, we assume that two consecutive frames are related by a homography [11]. This assumption holds in most cases as the global motion between two frames is usually small. It excludes independently moving objects, such as humans and vehicles.

For homography estimation, the first step is to find the correspondences between two frames. We combine two approaches for feature extraction in order to generate sufficient and complementary matches. We extract SURF [1] features and match them based on the nearest neighbor rule.

[1] http://lear.inrialpes.fr/~wang/improved_trajectories

	HOG	HOF	MBH	HOG+HOF	HOG+MBH	HOF+MBH	HOG+HOF+MBH
—	72.4%	76.0%	80.8%	82.9%	83.3%	82.2%	84.8%
T2	72.8%	76.1%	81.1%	82.7%	83.3%	82.2%	84.8%
H3	73.2%	77.3%	80.5%	82.7%	83.4%	82.0%	84.6%
Combine	74.6%	78.3%	82.1%	83.9%	84.4%	83.3%	**85.9%**

Table 1. The average accuracy over three train-test splits on UCF101 dataset. "—" stands for no SPM; "T2" is two temporal blocks; and "H3" three horizontal strips.

We also sample motion vectors from the optical flow, which provides us with dense matches between frames. Here, we use an efficient optical flow algorithm based on polynomial expansion [3]. We select motion vectors for salient feature points using the good-features-to-track criterion [9]. Unlike [14], we do not use human detection to remove inconsistent matches from humans, as human detector is too computationally expensive to run on large datasets.

We estimate the homography using all the matches with RANSAC. This allows us to rectify the image to remove the camera motion. Figure 1 (two rows in the middle) demonstrates the difference of optical flow before and after rectification. Compared to the original flow (the second row of Figure 1), the rectified version (the third row) suppresses the background camera motion and enhances the foreground moving objects.

For dense trajectories, there are two major advantages of canceling out camera motion from optical flow. First, the motion descriptors can directly benefit from this. As shown in [13], the performance of the HOF descriptor degrades significantly in the presence of camera motion. The experimental results in [14] show that HOF can achieve similar performance as MBH when we have correct foreground optical flow.

Second, we can remove trajectories generated by camera motion. This can be achieved by thresholding the maximal magnitude of the displacement vectors of the trajectories in the warped flow field. If the displacement is small, the trajectory is considered to be similar to camera motion, and thus removed. Figure1 (last row) shows examples of removed background trajectories. This results in similar effects as sampling features based on visual saliency maps [6, 12].

3. Feature encoding

For each trajectory, we compute several descriptors (*i.e.*, HOG, HOF and MBH) with exactly the same parameters as [13]. The final dimensions of the descriptors are 96 for HOG, 108 for HOF and 192 for MBH. To encode features, we use Fisher vector. Unlike bag of features, Fisher vector [8] encodes first and second order statistics between the video descriptors and a Gaussian Mixture Model (GMM). In recent evaluations [2, 7], this shows an improved performance over bag of features for both image and action classification.

To compute Fisher vector, we first reduce the descriptor dimensionality by a factor of two using Principal Component Analysis (PCA), as in [8]. We set the number of Gaussians to $K = 256$ and randomly sample a subset of 256,000 features from the training set to estimate the GMM. Each video is, then, represented by a $2DK$ dimensional Fisher vector for each descriptor type, where D is the descriptor dimension after performing PCA. Finally, we apply power and L2 normalization to the Fisher vector, as in [8]. To combine different descriptor types, we concatenate their normalized Fisher vectors.

We also use spatio-temporal pyramids (SPM) [5] to embed structure information in the final representation. We split the video into two blocks in time (*i.e.*, T2) and into three horizontal strips (*i.e.*, H3) as in [7]. In all experiments, we use a linear SVM for classification and fix $C = 100$ for the SVM. In the case of multi-class classification, we use a one-against-rest approach and select the class with the highest score.

4. Experimental results

We present all the results in Table 1. As expected, the best single descriptor is MBH. HOF works better than HOG, as camera motion compensation significantly improves its performance. Unlike [14], we do not include the Trajectory descriptor in Table 1, as combining it does not result in a further improvement.

Combining different descriptors is a straightforward way to improve the results. For the case of only combining two descriptors, "HOG+MBH" works the best, as MBH is the best motion descriptor and HOG is complementary to it. Interestingly, combining HOF and MBH further improves the results as they are complementary to each other. HOF represents zero-order motion information, whereas MBH focuses on first-order derivatives. Combining all three descriptors gives the best performance, *i.e.*, 84.8%.

If we compare the different rows in Table 1, we find spatio-temporal pyramids always helps to improve the performance. The improvement is more significant for a single descriptor. We observe around 2% improvement for HOG, HOF and MBH. For their combined version, SPM results in a smaller improvement of around 1%. Finally combining everything gives the best performance 85.9%, which is the result we submitted to the workshop [4].

Acknowledgments. This work was supported by Quaero (funded by OSEO, French State agency for innovation), the European integrated project AXES, the MSR/INRIA joint project and the ERC advanced grant ALLEGRO.

References

[1] H. Bay, T. Tuytelaars, and L. V. Gool. SURF: Speeded up robust features. In *ECCV*, 2006.

[2] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *BMVC*, 2011.

[3] G. Farnebäck. Two-frame motion estimation based on polynomial expansion. In *SCIA*, 2003.

[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. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[6] S. Mathe and C. Sminchisescu. Dynamic eye movement datasets and learnt saliency models for visual action recognition. In *ECCV*, 2012.

[7] D. Oneata, J. Verbeek, and C. Schmid. Action and event recognition with Fisher vectors on a compact feature set. In *ICCV*, 2013.

[8] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, 2010.

[9] J. Shi and C. Tomasi. Good features to track. In *CVPR*, 1994.

[10] 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.

[11] R. Szeliski. Image alignment and stitching: a tutorial. *Foundations and Trends in Computer Graphics and Vision*, 2(1):1–104, 2006.

[12] E. Vig, M. Dorr, and D. Cox. Space-variant descriptor sampling for action recognition based on saliency and eye movements. In *ECCV*, 2012.

[13] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013.

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