WARD-CMU@TRECVID 2015 Surveillance Event Detection

Xingzhong Du[1], Xuanchong Li[2], Xiaofang Zhou[1], Alexander Hauptmann[2]
[1]The University of Queensland, [2]Carnegie Mellon University

I. Abstract

We present a retrospective system for event detection in surveillance videos automatically, which is built on the Gatwick development data. It is an enhanced version of the retrospective system in [1]. The changes come from four aspects. First, dense trajectory [2] and improved dense trajectory [3] are used together in the proposed system. Second, the PCA features used in Gaussian Mixture Model are changed to whiten-PCA features. Third, we implement a learning-based probability function for LIBLINEAR [4]. Forth, instead of averaging all the detection scores we have, we select two kinds of them to get better results. We think all the changes are beneficial to the final submission 'WARD-CMU p-fusion_1' which wins 4 events in all 7 events. Specifically, it is worth noting that the PersonRuns sets a new record in recent years' SED competitions. Through the results in our internal evaluation, we think dense trajectory and improved dense trajectory are complementary for event detection in a complex surveillance environment. We also notice that current system is bad at detecting the short events like CellToEar, ObjectPut and Pointing. We are considering to introduce other methods such as pose estimation and pedestrian detection to enhance current system in the future.

II. Retrospective System

The retrospective system consists of five components : video preprocessing, feature extraction, feature encoding, model training and score fusion. In this section, we will briefly introduce their functionalities and our changes in them.

A. Video preprocessing

The video preprocessing prepares the input for the feature extraction. In the first step, all the videos are resized to 320×240. This accelerates the feature extraction. After that, the resized videos are slid into video clips. Specially, each clip's length is 60 frames and 30-frame overlapped with the adjacent clips. It is worth noting that the dense trajectory and improved dense trajectory track the feature points for 15 frames by default. It means that, if we want to extract all the feature points in a clip, we need the clip's length to be 75 frames. Therefore, for each video clip, we append 15 subsequent frames behind the original 60 frames. After the sliding, roughly 350k clips are generated. The clips whose size are 0 are removed at the end of video preprocessing.

B. Feature Extraction

The feature extraction processes video clips and generates the raw features. Last year, we used improved dense trajectory only. This feature can capture the wrapped object motion from the video, which is achieved by removing camera motion through homography plus RANSAC [6]. We check the released code[1], and find that improved dense trajectory removes the dominant motion between two frames actually [7]. In SED, there is no camera motion in the videos. The dominant motion comes from the stream of people. Applying improved dense trajectory in this situation can remove the interference from the irrelevant persons. That's why we used improved dense trajectory in last year's SED.

However, after last year's SED, we find that some positive clips only contain a few persons. In this situation, the dense trajectory is more suitable because removing the dominant motion is unnecessary. Therefore, the features in use for this year's SED are dense trajectory and improved dense trajectory.

C. Feature Encoding

The feature encoding encodes the raw feature array into a vector for each clip. In event detection, this step is necessary because it makes the feature representation more robust. The state-of-the-art encoding method is the spatial-temporal fisher [8]. This method consists of four steps. The first step learns a projection matrix based the raw features by Principle Components Analysis (PCA). After dimension reduction, the dimension of the raw features are reduced by a factor of two. The second step learns a Gaussian Mixture Model (GMM) based on the reduced features. The components in GMM act as the visual words for inferring the soft assignment information. The third step transforms the reduced features of a clip into fisher vectors by the soft assignment information, and averages them into one fisher vector for this clip. The last but not the least step is the normalization. It enhances the class-specific information by power and l_2 normalization [10].

In recent literature [9], the whiten PCA has been proved to be superior to PCA for action recognition. The whiten PCA rescales the reduced features to make them have similar variances. It could make the distribution of GMM components more uniform, so that the fisher vector can discriminate more patterns. Therefore, in this year's system, we use whiten PCA instead of PCA in feature encoding.

D. Model Training

The model training creates detectors for each event under Camera 1, 2, 3, 5. It consists of three steps. In the first step, we treat the clips which have 50% overlap with the ground truth as the positive, then use LIBLINEAR [4] to train detectors and two-fold cross-validation to choose parameters. Using

[1] https://lear.inrialpes.fr/people/wang/improved_trajectories

LIBLINEAR has two reasons. First, the features for learning are all fisher vectors which are suitable for linear kernels naturally. Second, the dimension of the fisher vector in use is 116736. It requires 0.5MB space of each vector on disk. linear SVM avoids storing the support vectors, which saves a lot of storage space for a detector. After the detectors are trained, the decision values from the detectors need to be transformed into probabilities, so that the decision values from different models are comparable. In LIBSVM [5] python version, we can use its packaged probability function. But in LIBLINEAR, we need to implement the probability function by ourselves. In last year's submission, the probability function is simply implemented by curve fitting. In this year's submission, the probability function is implemented as [11], which is more robust than the curve fitting. The python code can be downloaded from *https://github.com/domainxz/pytools.git*. We verify this code by reproducing the action recognition experiment in [3]. The results show this code can work properly. The third step of model training learns a threshold for each detector, then applies Non-Maximum Suppression (NMS) to merge the adjacent positive clips. When the model training is finished, We will have 7×4 detectors per feature. Each of them only focuses on one event under one camera.

E. Score Fusion

TABLE I. EVALUATIONS FOR FUSION STRATEGY

event	idtwfv	dtfv+idtwfv	idtfv+idtwfv	dtwfv+idtwfv
CellToEar	1.0058	**1.0013**	1.0036	1.0040
Embrace	1.0068	0.9253	0.9197	**0.9105**
ObjectPut	1.0042	1.0023	1.0026	**1.0020**
PeopleMeet	0.9520	**0.9238**	0.9369	0.9297
PeopleSplitUp	0.9613	0.8931	0.9036	**0.8861**
PersonRuns	0.6440	0.6478	0.6549	**0.6299**
Pointing	1.0140	0.9920	0.9891	**0.9858**

The score fusion averagely fuses the probabilities from different selected features to form the final submission. Therefore, the key problem is which features are fused together. After evaluating the detectors on the development set, we get four group of detection scores at hand. They are predicted by the fisher vectors in terms of dense trajectory with normal PCA (dtfv), dense trajectory with whiten PCA (dfwfv), improved dense trajectory with normal PCA (idtfv) and improved dense trajectory with whiten PCA (idtwfv). We try all the possible combinations to select the best, and find the combination of dtwfv and idtwfv is the best.

III. Result Analysis

TABLE II. COMPARISON BETWEEN OUR RESULTS AND OTHERS' BEST RESULTS

Event	Our retro results		Best retro results		Best inter results	
	aDCR	mDCR	aDCR	mDCR	aDCR	mDCR
CellToEar	**1.0046**	**1.0006**	1.3071	1.0006	2.1010	1.0006
Embrace	0.8680	0.8453	0.7909	0.7909	0.8540	0.8540
ObjectPut	1.0160	0.9884	1.0120	0.9965	0.9930	0.9867
PeopleMeet	**0.8939**	**0.8848**	1.0426	0.9981	0.9978	0.9919
PeopleSplitUp	**0.8934**	**0.8785**	0.9387	0.9253	0.9164	0.9164
PersonRuns	**0.5768**	**0.5466**	0.9700	0.9545	0.9411	0.9411
Pointing	1.0140	0.9940	1.0040	0.9989	0.9939	0.9939

In this section, we firstly compare our retrospective submission to the others' best results in terms of actual DCR (aDCR) and min DCR (mDCR) by Table II. We make the event names bold which we get the first positions in SED 2015 [12]. In total, we win 4 events in no matter the retrospective or the interactive competitions. But we notice that our method is only good at handling the events driven by long duration actions.

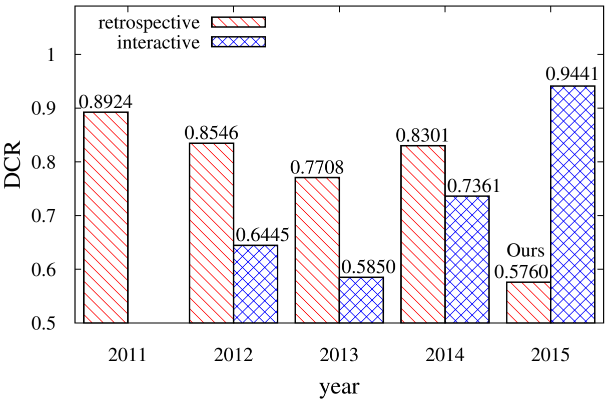

Fig. 1. The accuracy evaluated in terms of actual DCR for PersonRuns event in recent five years' SED retrospective and interactive competitions.

In this year, we achieve the lowest DCR in PersonRuns event. We compare this result to recent five years' best results in Fig. 1. We find this score achieves the new record in recent years' SED competitions, even though the test data have been changed since 2014.

References

[1] S.-I. Yu, L. Jiang, Z. Mao, X. Chang, X. Du, C. Gan, Z. Lan, Z. Xu, X. Li, Y. Cai *et al.*, "Informedia @ TRECVID 2014," in *NIST TRECVID Video Retrieval Evaluation Workshop*, 2014.

[2] H. Wang, A. Kläser, C. Schmid, and C. Liu, "Action recognition by dense trajectories," in *CVPR*, 2011, pp. 3169–3176.

[3] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *ICCV*, 2013, pp. 3551–3558.

[4] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin, "LIBLINEAR: A library for large linear classification," *JMLR*, vol. 9, pp. 1871–1874, 2008.

[5] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," *TIST*, vol. 2, no. 3, p. 27, 2011.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *CACM*, vol. 24, no. 6, pp. 381–395, 1981.

[7] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *SCIA*, 2003, pp. 363–370.

[8] J. Krapac, J. J. Verbeek, and F. Jurie, "Modeling spatial layout with fisher vectors for image categorization," in *ICCV*, 2011, pp. 1487–1494.

[9] X. Peng, L. Wang, X. Wang, and Y. Qiao, "Bag of visual words and fusion methods for action recognition: Comprehensive study and good practice," *CoRR*, vol. abs/1405.4506, 2014.

[10] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," in *ECCV*, 2010, pp. 143–156.

[11] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *ADVANCES IN LARGE MARGIN CLASSIFIERS*. MIT Press, 1999, pp. 61–74.

[12] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, G. Quenot, and R. Ordelman, "Trecvid 2015 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2015*. NIST, USA, 2015.