IRDS-CASIA at TRECVid 2011: Surveillance Event Detection

Yanhu Shan, Zhang Zhang, Shiquan Wang, Kaiqi Huang, Tieniu Tan
National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, P.R.China
{yanhu.shan, zzhang, sqwang, kqhuang, tnt}@nlpr.ia.ac.cn

Abstract

This paper proposes the event detection system for TRECVid 2011 surveillance event detection. "CellToEar", "Embrace", "ObjectPut", "PeopleMeet", "PeopleSplit-Up", "PersonRuns" and "Pointing" are the 7 events we detect in our system. Firstly, interest points are detected in the Local spatial and temporal regions, and local feature is described with SFA (slow feature analysis) method. We apply lib-SVM to classify the 7 events and the 7 scores corresponding to foregoing events are the original result of the local region. Post-processing is used to generate the global result and reduce the false alarm.

1. Introduction

Activity recognition in real world datasets is always a challenge work in compute vision. TRECVid dataset is much more difficult because of its complexity of scenes. More occlusion and complex foreground are obvious

We detect all the seven events in the TRECVid dataset. The flowchart of our system is shown in figure 1. We ultimate all the videos of tv2008 except the test five ones for dry run and videos of scene 4 as training data. In the training step, we firstly labeled all the spatial locations of the ground truth in the training data which only have the temporal information (starting and ending frames). Then we get a local video volume of every event manually in the videos. Taking one local volume in one video for instance, we detect the STIP (spatial and temporal interest points) [2] in the local volume and get several cuboids around different interest points. Slow Feature Analysis is used in the system to extract slow feature functions of different actions. SFA [6] has been successfully used in the visual receptive fields of the cortical neurons successfully because of its ability of extracting slowly varying features from a quickly varying input signal. We apply Accumulated Squared Derivative (ASD) feature [6], which can encode the statistical distribution of slow features of an action sequence, to represent the action sequence. In the testing step, we get the local volume with multi-scale sliding window. The scale of windows are adapt to the complexity of the foreground. The method of feature extracting is the same with the foregoing. SVM [1] is used to classify the seven categories. We obtain the score vector of 7actions from the local volume. After recording all of the score vector corresponding to the local volumes in the global volume which is the original video clip from starting frame to ending frame. We define the maximum score of all the local volumes of one event to be the global score of the event. Non-maximum Suppression (NMS) and probability map, which is similar to the work of Yang et al. [5], are used as past-process after we get all the global scores in the whole video.

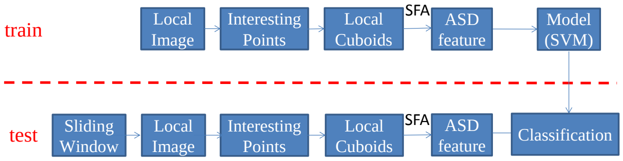


Figure 1. Flowchart of our system.

The rest of this paper is organized as follows. We introduce the detail of our method in the system in Section 2. Experimental results and discussion is presented in Section 3. We conclude our work and give our suggestion on TRECVid SED in Section 4.

2. Our Method

Out approach includes two parts, i.e., feature extraction and action classification. The core of our work is feature extraction, and it is presented in part one. The other part briefly shows action classification.

2.1. Label and Slice Window

The only information of TRECVid dataset, the starting and ending frames of different events, is not enough, because we should know the location of the events in the com-

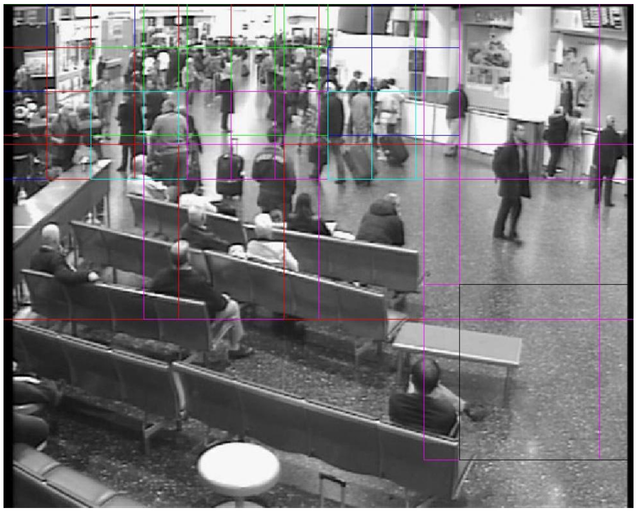


Figure 2. The results show of sliding windows.

plex foreground. So we label the spatial location of every event with a rectangular box, and a volume is obtained with the temporal information. This volume is the unit of training data applied in our training step.

To correspond to the local volume, we use sliding window method to get the local volume in test data. Both temporal slide and spatial slide are used in our system. The strategy of the slice window is that: 1) we detect the local event with spatial sliding windows in 50 frames. The spatial sliding window size is 100 by 100. The overlap of two adjacent windows is 20. We will skip the following steps when the number of the interest points is less than N^-. If the number is bigger than N^+, we will do the sub slide in the 100 by 100 rectangular box with a 50 by 50 window. The overlap is 10. So we can use different scale self-adaptively. Taking into account the computational complexity, we just use two scales. The sliding windows are shown in figure 2.

2.1.1 STIP

Dollar et all.'s work is used in our system because of its robustness to pose, image clutter, occlusion and complex foreground which characters TRECVid dataset has. The method is briefly reviewed here.The response function is

$$R = (I * g * h_{ev})^2 + (I * g * h_{od})^2 \tag{1}$$

Where I is the local volume and $g(x, y; \sigma$ is 2-D Gaussian smoothing kernel, applied only along the spatial dimensions, $h_{ev}(t; \tau, \omega) = -\cos(2\pi t\omega)e^{-t^2/\tau^2}$, $h_{od}(t; \tau, \omega) = -\sin(2\pi t\omega)e^{-t^2/\tau^2}$ and $omega = 4/tau$. Cuboids are generated around interesting points. A cuboid has the size of 131313 taking corresponding point as the center.

2.1.2 SFA and its discriminative derivation

As proposed in [6], SFA has been used for human action recognition. Mathematically, SFA is defined as follows [4]: Given an I-dimensional input signal $\mathbf{x}(t) = [x_1(t), ..., x_I(t)]^T$ with $t \in [t_0, t_1]$ indicating time, S-FA finds out a set of input-output functions $\mathbf{g}(x) = [g_1(x), ..., g_J(x)]^T$, so that the J-dimensional output signal $\mathbf{y}(t) = [y_1(t), ..., y_J(t)]^T$ with $y_j(t) = g_j(\mathbf{x}(\mathrm{t}))$ varies as slow as possible, i.e., for each $j \in [1, ..., J]$,

$$\Delta_j = \Delta(y_j) = \left\langle \dot{y}_j^2 \right\rangle_t isminimal, \tag{2}$$

subject to

$$\left\langle y_j \right\rangle_t = 0\ zeromean; \tag{3}$$

$$\left\langle y_j^2 \right\rangle_t = 1\ unitvariance; \tag{4}$$

$$and\ \forall j' < j : \left\langle y_{j'} y_j \right\rangle_t = 0\ decrrelation \tag{5}$$

where $\dot{y}$ denotes the operator of computing the first order derivative of y and $\left\langle y_j \right\rangle_t$ is the mean of signal y over time. Equation (2) is the primary objective of minimizing the temporal variation of the output signal, where the temporal variation is measured by the mean of the squared first order derivative. Constraint (3) presents for convenience only, so that Constraint (4) and (5) take a simple form. Constraint (4) means that the transformed signal should carry some information and avoid the trivial solution $y_j(t) = const.$ Constraint (5) ensures that different output components carry different types of information and it also induces an order, the first output signal being the slowest one, the second being the second slowest, etc.

If the transformation is linear, i.e., $g_j(\mathbf{x}) = w_j^T \mathbf{x}$, wherein x is input and w_j is weight, the solution of SFA is equivalent to the generalized eigenvalue problem:

$$AW = BWA \tag{6}$$

Where $A = \left\langle \dot{\mathbf{x}}\dot{\mathbf{x}}^T \right\rangle_t$ is the expectation of the covariance matrix of the temporal first order derivative of the input vector, $B = \left\langle \mathbf{x}\mathbf{x}^T \right\rangle_t$ is the expectation of the covariance matrix of the input vector,Λ is a diagonal matrix of the generalized eigenvalues and W is the corresponding generalized eigenvectors. Furthermore, the order of slow features is determined by eigenvalues and where the most slowly varying signal has the lowest index.

The nonlinear transformation can be deemed as the linear transformation in a nonlinear expansion space. The nonlinear expansion function **h(x)** is defined by

$$\mathbf{h(x)} := [h_1(\mathbf{x}), ..., h_M(\mathbf{x})] \tag{7}$$

Afterward, SFA can be performed in the expansion space to obtain nonlinear slow feature functions.

In summary, slow feature functions can be obtained by the following two steps:

- Nonlinear expansion Apply a nonlinear function **h(x)** to expand the original signal, and centralize **h(x)**.

$$\mathbf{z} := \mathbf{h}(\mathbf{x}) - \mathbf{h}_0 \tag{8}$$

where $\mathbf{h}_0 = \langle \mathbf{h}(\mathbf{x}) \rangle_t$. The centralization makes Constraint (3) valid. In this paper, we use the quadratic expansion, i.e., $\mathbf{h}(\mathbf{x}) = [\mathrm{x}_1, ..., \mathrm{x}_\mathrm{I}, \mathrm{x}_1\mathrm{x}_1, \mathrm{x}_1\mathrm{x}_2, ..., \mathrm{x}_\mathrm{I}\mathrm{x}_\mathrm{I}]$.

- Solve the generalized eigenvalue problem

$$AW = BWA \tag{9}$$

Where $A = \left\langle \dot{\mathbf{z}}\dot{\mathbf{z}}^T \right\rangle_t$ and $B = \langle \mathbf{z}\mathbf{z}^T \rangle_t$.

Assume the dimensionalities of matrix A and B are M, the first K eigenvectors $w_1, ..., w_K (K \ll M)$ associated with the smallest eigenvalues $\lambda_1 \leq \lambda_2 ... \leq \lambda_K$ are the nonlinear slow feature functions $g_1(\mathbf{x}), ..., g_K(\mathbf{x})$:

$$g_j(\mathbf{x}) = w_j^T(\mathbf{h}(\mathbf{x}) - \mathbf{h}_0), \tag{10}$$

which satisfies Constraints (3) - (5) and minimizes the objective function (2). Here, the input-output function computes the output signal instantaneously. Therefore, slow variation of the output signal cannot be achieved by using the temporal low-pass filter, but must be obtained by extracting aspects of the input signal that are inherently slow and useful for a higher level representation.

To properly introduce the supervised information to the SFA learning, we propose the discriminative SFA (D-SFA). D-SFA is inspired by discriminative sparse coding [3], where a number of sets of discriminative dictionaries are learnt, and each set of dictionaries are used to reconstruct a specific image class. Accordingly, D-SFA learns a number of sets of functions and each set of functions are used to slowdown a specific action class.

Given C classes of I-dims input signals $\{\mathbf{x}_\mathrm{c}(t) = [x_{c1}(t), ..., x_{cI}(t)] | c \in \{1, ..., C\}\}$, for the c-th class, D-SFA finds a set of J-dims functions $\mathbf{g}_c(\mathbf{x}) = [g_{c1}(\mathbf{x}), ..., g_{cJ}(\mathbf{x})]^T$ to minimize $\Delta(g_{cj}(\mathbf{x}_c)) - \gamma * \Delta(g_{cj}(\mathbf{x}_{c'}))$. Therefore each learnt function makes the intra-class signals $\mathbf{x}_c(t)$ (t) vary slowly, but makes the inter-class signals $\mathbf{x}_{c'}(t)$ that are different from class c vary quickly. Assume $\mathbf{g}_c(\mathbf{x}) = [g_{c1}(\mathbf{x}), ..., g_{cJ}(\mathbf{x})]^T$ are linear functions, for each $j \in \{1, ..., J\}$, D-SFA minimizes

$$\begin{aligned} &\Delta(g_{cj}(\mathbf{x}_c)) - \gamma * \Delta(g_{cj}(\mathbf{x}_{c'})) \\ &= \left\langle [g_{cj}\dot{(}\mathbf{x}_c)]^2 \right\rangle_t - \gamma * \left\langle [g_{cj}\dot{(}\mathbf{x}_{c'})]^2 \right\rangle_t \\ &= w_{cj}^T[\left\langle \dot{\mathbf{x}}_c\dot{\mathbf{x}}_c^T \right\rangle_t - \gamma * \left\langle \dot{\mathbf{x}}_{c'}\dot{\mathbf{x}}_{c'}^T \right\rangle_t]w_{cj} \end{aligned} \tag{11}$$

subject to:

$$\langle g_{cj}(\mathbf{x}_{c \cup c'}) \rangle = 0 \; zeromean; \tag{12}$$

$$\left\langle [g_{cj}(\mathbf{x}_{c \cup c'})]^2 \right\rangle = 1 \; unitvariance; \tag{13}$$

$$\forall j' < j : \langle g_{cj'}(\mathbf{x}_{c \cup c'}) g_{cj}(\mathbf{x}_{c \cup c'}) \rangle_t = 0 \; decorrelation, \tag{14}$$

where w_{cj} is the weight vector of the j-th slow feature function for the class c and γ is the tradeoff parameter. D-SFA can be written as a generalized eigenvalue problem

$$EW = BW\Lambda, \tag{15}$$

where $E = [\left\langle \dot{\mathbf{x}}_c\dot{\mathbf{x}}_c^T \right\rangle_t - \gamma * \left\langle \dot{\mathbf{x}}_{c'}\dot{\mathbf{x}}_{c'}^T \right\rangle_t]$, $B = \left\langle \mathbf{x}_{c \cup c'} {\mathbf{x}_{c \cup c'}}^T \right\rangle_t$ is a diagonal matrix of the generalized eigenvalues and W is the corresponding generalized eigenvectors. To obtain nonlinear slow feature functions, we can perform the nonlinear expansion before the D-SFA learning.

2.2. ASD Feature

In the SFA learning, cuboids are derived from d successive frames. Thus we compute a statistical feature from d frames to represent an action sequence. SFA minimizes the average squared derivative, so the fitting degree of a cuboid to a certain slow feature function can be measured by the squared derivative of the transformed cuboid. If the value is small, the cuboid fits the slow feature function. Otherwise, the cuboid does not fit the function. For cuboid C_i and slow function F_j, the squared derivative $v_{i,j}$ is

$$v_{i,j} = \frac{1}{d - \Delta t} \sum_{t=1}^{d - \Delta t} [C_i(t+1) \otimes F_j - C_i(t) \otimes F_j]^2, \tag{16}$$

where $\otimes$ is the transformation operation.

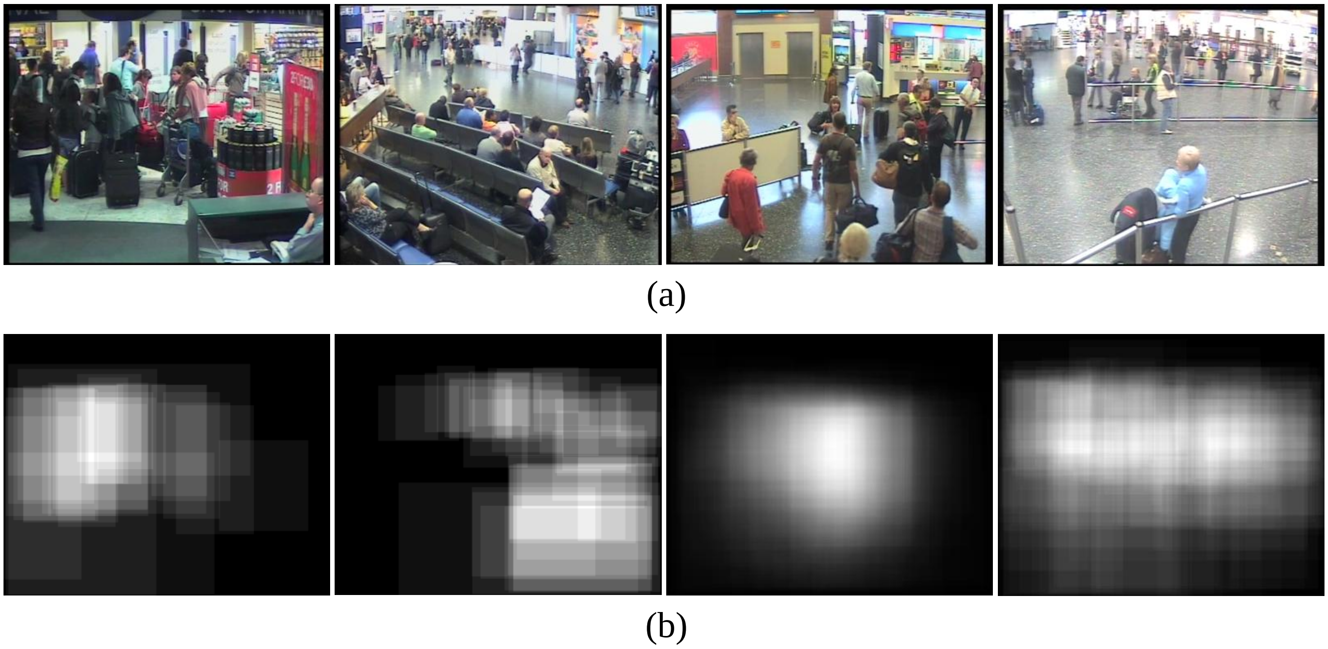
We then accumulated the squared derivatives over all cuboids to form the ASD feature.

$$f_{ASD} = \sum_i^N V_i, \tag{17}$$

where N is the total number of cuboids in current snippet, and $V_i = \langle v_{i,1}, v_{i,2}, ..., v_{i,K} \rangle^T$. Since the number of cuboids detected in a snippet may differ from that in another snippet, it is necessary to normalize the feature vector. Here, we perform the $L1$ normalization.

2.3. Classification

As the SFA function is learned separately for the 4 scenes, we learn 7*4 classifiers for every event in every scene. We obtaining the score vector of 7actions from the local volume by using the corresponding classifiers in the corresponding scene, and the location of windows are recorded for post-process.


(a)

(b)

Figure 3. (a) The frames of 4 different scenes; (b) corresponding probability maps of the 4 scenes of (a).

Event scene	CellToEar	Embrace	ObjectPut	PeopleMeet	PeopleSplitUp	PersonRuns	Pointing
1	0.5625	0.1055	0.9114	0.5125	0.6773	0.2925	0.9352
2	0.9418	0.6538	0.9767	0.4210	0.2274	0.3175	0.9601
3	0.9850	0.8749	0.9328	0.9417	0.2050	0.8172	0.8553
5	0.9273	0.9431	0.8625	0.5209	0.2980	0.3524	0.9880

Table 1. Min DCR we test for different scenes in the data which are used in dry run.

2.4. Post-process

As false alarms are bound to happen because one event can span across several clips, post-process is necessary. We use prior knowledge and NMS here to reduce false alarms.

2.4.1 Prior Knowledge

We build a probability map for every event in every scene. The maps are generated from the ground truth which we labeled manually and shown in figure 3. We process the score using

$$score_w = \alpha * w + (1 - \alpha) * score_{original} \qquad (18)$$

Where w is obtained by using the bounding box of the $score_{original}$. We set the average pixel value of the corresponding region in the map to be w. α is a parameter to balance w and $score_{original}$.

2.4.2 NMS

$score_w$ is the local score of one slice window. We define the maximum local score ($score_{original}$) as the global score of the video clip, then Non-maximum Suppression (NMS) is utilize to reduce the false alarms. Our strategy is:

- step 1: finding the maximum $score_w$ of one event in one video sequence and setting the scores, with the distance of less than R, to be zeros. R is decided by the average length of the time of the event in the ground truth of the training data.

- step 2: finding the next maximum score excluding the previous one;

- step 3: repeating step 2 if next maximum score exists and is greater than 0.

Event	CellToEar	Embrace	ObjectPut	PeopleMeet	PeopleSplitUp	PersonRuns	Pointing
Min DCR	1.0003	1.0003	0.9994	0.9997	0.9835	0.9979	1.0003
Best Min DCR	1.0003	0.8658	0.9983	0.9724	0.8809	0.8372	0.973

Table 2. Min DCR of the formal evaluation result of our baseline system in TRECVid .

Event scene	CellToEar	Embrace	ObjectPut	PeopleMeet	PeopleSplitUp	PersonRuns	Pointing
1	1.0016	1.0016	1.0016	1.0016	1.0016	1.0016	1.0829
2	1.0016	1.0082	1.0016	0.9959	0.9360	1.0016	1.0016
3	1.0033	1.0016	0.9940	1.0016	0.9752	1.0016	1.1017
5	1.0508	1.0016	1.0016	1.0016	1.0016	1.0016	1.0016

Table 3. Min DCR of the formal evaluation result for different scenes.

3. Experimental Analysis

The SFA functions and SVM models are trained in the training data excluding the 4 videos which are mentioned in the Experiment Control File: expt_2009_retroED_DEV09_ENG_s - camera_NIST_2.xml in dry run test, R (mentioned in 2.4.2) is decided by training data, and α (mentioned in 4.11)) of baseline system is set by test them in the 4 videos for dry run testing. We use Min DCR to reflect the performance of the system. The result we test in the foregoing data is showing in Table 1 when α is equal to 0, which means we do't use the prior knowledge. So the functions, models and R are all decided by training data excluding the 4 videos in dry run test. Appropriate α can contribute a small improvement for the system. The result looks perfect in individual scene. It seems that our system can detect the event well after learning the SFA function for every event in event scene, and prior knowledge and NMS really reduce the false alarms of the system.

Table 2 shows the formal evaluation result of our baseline system in TRECVid 2011 SED Evaluation data. The system seems not good enough from the result in Table 2, because every SFA function and every SVM model are trained separately in different scenes. They do not fit every scene, so the Min Dec. Threshes are not the same for one event in different scenes.

The results generated separately in different scenes are also provided in Table 3. In fact, they are too bad to our expectations. We got a perfect result without updating the parameters in the test data of dry run.

As plenty of time is needed to run the system in the formal evaluation data, we cannot adjust our system effectively.

4. Conclusions

We propose the event detection system for TRECVid 2011 surveillance event detection. This system utilizes sliding windows to detect the local regions and learn SFA function for ASD feature. NMS and prior knowledge are proposed as post-processing to reduce the false alarms. To be honest, our system is slow because of the dense sliding windows, so it need to be optimized.

5. Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No.61135002,61175007).

References

[1] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[2] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on*, 2005.

[3] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Discriminative learned dictionaries for local image analysis. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1 –8, june 2008.

[4] L. Wiskott and T. Sejnowski. Slow feature analysis: Unsupervised learning of invariances. *Neural Computation*, 14(4):1, 04 2002.

[5] W. Yang, T. Lan, and G. Mori. Sfu at trecvid 2009: Event detection.

[6] Z. Zhang and D. Tao. Slow feature analysis for human action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PP(99):1, 2011.