Cross-Dataset Action Detection

Liangliang Cao*
Beckman Institute
Department of ECE, UIUC
cao4@ifp.illinois.edu

Zicheng Liu
Microsoft Research
Redmond, WA, USA
zliu@microsoft.com

Thomas S. Huang
Beckman Institute
Department of ECE, UIUC
huang@ifp.illinois.edu

Abstract

In recent years, many research works have been carried out to recognize human actions from video clips. To learn an effective action classifier, most of the previous approaches rely on enough training labels. When being required to recognize the action in a different dataset, these approaches have to re-train the model using new labels. However, labeling video sequences is a very tedious and time-consuming task, especially when detailed spatial locations and time durations are required. In this paper, we propose an adaptive action detection approach which reduces the requirement of training labels and is able to handle the task of cross-dataset action detection with few or no extra training labels. Our approach combines model adaptation and action detection into a Maximum a Posterior (MAP) estimation framework, which explores the spatial-temporal coherence of actions and makes good use of the prior information which can be obtained without supervision. Our approach obtains state-of-the-art results on KTH action dataset using only 50% of the training labels in tradition approaches. Furthermore, we show that our approach is effective for the cross-dataset detection which adapts the model trained on KTH to two other challenging datasets[1].

1. Introduction

This paper considers the problem of cross-dataset action detection, which aims to generalize action detection models built from a *source* dataset to a *target* dataset. The two datasets are likely to be collected in very different environments. For example, the human actions in the source dataset may be recorded with clean backgrounds, and each video clip may involve only one type of action and only a single person, who keeps performing the same action for the entire video clip. In contrast, in the target dataset, the background may be cluttered, and there may be multiple people moving around with occasional occlusions.

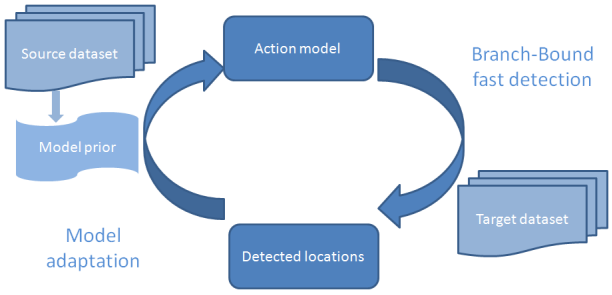

Figure 1. The framework of our cross dataset action detection method.

The problem of cross dataset detection is important especially for real surveillance applications. Conventional detector usually builds a classifier from labeled examples and assumes the testing samples are generated from the same distribution. In the case where a new dataset has a different distribution from the training dataset, a new classifier needs to be trained, which requires large amount of training labels from the new dataset. Such a labeling process is time-consuming and labor intensive. Especially, when the background is cluttered and there are multiple people appearing in the same frame, the labelers need to provide a bounding box for every subject together with the starting/ending frames of an action instance. For a video as long as several hours, the labeling process might take several weeks or even longer. The cross-dataset learning aims to alleviate such a problem. We aim to adapt the existing classifier from a source dataset to a new target dataset, while requiring only a small amount of labeling samples or even no labels at all.

Cross-dataset action detection is challenging because the video are usually taken in different occasions. As Figure 2 shows, the videos are of different backgrounds, and the actions may appear differently with the change of performers, lighting conditions, scales and performing speeds. However, we believe such a task is possible due to two factors. First, the actions in different datasets still share some similarities to some degree. Second, the classifier adaptation

*The majority of this work is carried out when Liangliang Cao worked as a summer intern at Microsoft Research.

[1]Please check http://www.ifp.illinois.edu/~cao4/crossdataset_action for more results.

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

Figure 2. Comparing differences between source and target datasets. The first row shows four actions from KTH dataset: boxing, hand waving, hand clapping, and running. The second row shows the corresponding actions from MSR dataset (boxing, hand waving and clapping) and TRECVID dataset (running).

can leverage the spatial and temporal coherence of the individual actions in a new dataset. This paper combines action detection and classifier adaptation into a single framework, which we believe is beneficial to the cross-dataset detection task.

Cross-dataset learning and semi-supervised learning share the similarity that they both aim to learn a model with limited amount of labeled data. However, semi-supervised learning assumes the labeled data and unlabeled data are generated from the same distribution, which is not the case in cross-dataset learning. In cross-dataset learning, we assume the actions of interests share some similarities but are not exactly the same. The background models are usually quite different from dataset to dataset. Therefore, semi-supervised learning technique is not suitable to our problem. We instead treat the action model in the source dataset as prior, and employ maximum a posterior estimation to adapt the model to the new dataset. Moreover, the model developed in this paper aims at both classification and detection (spatial and temporal localization), while the conventional semi-supervised learning algorithm considers only the first task.

Following the recent progress in the field of action recognition, we employ Spatial-Temporal Interest Points (STIPs) [17] [2] [10] as the features fed into our system. More specifically, we extract both histogram of oriented gradients and histogram of optical flow features at salient locations as in [10], and the feature dimension is 162. STIPs are motivated by the recent success of spatial salient patches in image domain, especially the well known SIFT and HOG features. The STIPs in videos provide rich representation of local gradients of edges and motions. However, such features are often affected by lighting conditions, viewpoints, and partial occlusions. For this reason, the STIPs from different datasets may not have the same distribution.

A majority of recent work on STIPs takes quantized STIPs as input and builds histograms based on quantization indices. The quantized STIPs are also called video codewords. The histograms can be fed into discriminative SVM classifiers [17] or generative topic models [15]. The collection of quantized codewords is also named as a codebook. The use of the codebook and histogram is preferred because it can condense different number of STIPs into a fixed length feature vector. However, quantized codeword representation is not fit for cross-dataset scenarios due to the variety of STIPs in different scenarios. Given two videos captured with different viewing points and light conditions, the corresponding distributions of STIPs are likely quite different. If we build a new codebook on the new dataset, the word histogram representation will be totally different, so the old model cannot be applied. In summary, quantized codeword representation overlooks the differences of STIPs distribution in different scenarios and may fail to correctly transfer the knowledge from source dataset to target dataset.

In this work, we employ a probabilistic representation of the original STIPs instead of quantized one. In our approach, we first apply PCA to STIPs, and then model them with Gaussian Mixture Models (GMMs). GMM with large number of components is known to have the ability to model any given probability distribution function. Although more components might work better, we employ 512 components GMM which works well for all the dataset. To model the correlation between a source dataset and a target dataset, we introduce a prior distribution of the GMM parameters and propose an adaptation approach to incorporate the prior information for cross-dataset analysis.

Our work also explores the spatial-temporal coherence nature of the video actions. We use a 3D subvolume to represent a region in the 3D video space that contains an action instance. A 3D subvoume is parameterized as a 3D cube with six degrees of freedom in (x, y, t) space. Spatial and temporal localization of an action in a video sequence is rendered as searching for the optimal subvolume. Our approach locates the action and updates the GMM parameters simultaneously. Figure 1 illustrates the framework of our method. The benefits of such a formulation are twofold. First, in contrast to classification, action detection provides much more useful information to the user. Second, by locating the spatial-temporal subvolumes, it allows us to iteratively filter out the STIPs in the background thus refining the model estimation.

2. Related Works

In the past several years, much research work is based on modeling spatial-temporal interest points (STIPs) [17], [2], [6], [20], [11], [14]. Models with STIPs avoid the alignment problem in temporal domain [5], and can effectively distinguish periodic or short-interval actions such as running and jogging with clean background.

When the background is cluttered and when there are multiple actors in the scene, it becomes crucial to locate where the action happens in addition to simply estimat-

ing the action category for video clips. Haritaoglu *et al*. built a tracking based system to monitor the region containing persons [3]. Ke *et al*. constructed action models using hand-labeled template and matched the template against over-segmented spatio-temporal volumes [7]. Hu *et al*. employed a multiple-instance learning framework to allow rough annotations instead of accurate ones [4]. In this paper, we follow the work of Yuan *et al*. [22] which employs a branch-and-bound approach [8] to locate the action instances. However, our work is different from [22] since [22] did not address the data mismatch problem. In contrast, we integrate model adaptation and action detection into a single framework, thus providing an effective solution for handling data mismatches in action detection. Moreover, [22] models STIPs with the mutual information scores, while this paper employs GMM to represent STIP whose posterior can be effectively estimated from the source dataset.

GMM-based adaptation was first developed for audio-based speaker identification system [16]. These work used GMM to train a universal background model and then model the speaker by adapting the background model. Following this approach, a lot of recent work employs GMM to model the temporal structure of audio-visual events [23] or the visual similarities between two images or videos [24]. However, the fundamental difference between those work and this paper is that those systems used GMM-based adaptation for classification task, instead of detecting the action of interests. As in classical audio processing, those work does not consider the cross dataset scenarios or the spatial configuration, which are essential constraints for our problem. Our work also shares some similarity with Stauffer and Grimson's GMM-based tracking [18], which used a mixture of Gaussians to model the background pixels and adapted the model for each new frame. This paper also employs GMM to model the distribution of STIPs. However, the differences are: (1) Their approach aims to keep up with the slow but continuous change of background over time, while this paper aims to adapt the model to a very different dataset. (2) Our approach combines action detection and model adaptation into a single framework, while [18] only considers the adaptation issue.

To the best of our knowledge, there has been no previous work considering cross-dataset action detection. However, some researchers recognize the importance of cross-dataset learning and address this problems in different scenarios. Yang *et al*. [21] developed adaptive SVMs for classifying video concepts such as studio, outdoor, and sports. Their approache requires new labels in the target dataset and thus cannot be applied to our problem. On the contrary, this paper explores the the spatial-temporal configuration of the action instances, and updates the model without new labels. Lampert et al considered transfer learning of object recognition [9], which is different from ours in several aspects: (1) Lampert et al. considered the knowledge transfer from one class to another, while we consider the model adaptation from one dataset to another dataset. (2) Lampert et al. assumed the attributes are shared across different categories, while we don't use attributes. (3) [9] does not address the localization problem, which plays a critical role in our scheme.

3. Cross-Dataset Action Detection

We represent a video sequence as a collection of spatial-temporal interest points (STIPs), where each STIP is represented by a feature vector q. To model the probability of each STIP , we employ a Gaussian Mixture Model (GMM) to represent universal background distribution. GMMs is known to have the ability to model any given probability distribution function. Suppose a GMM contains K components, the probability can be written as $Pr(q|\theta) = \sum_{k=1}^{K} w^k \mathcal{N}(q; \mu^k, \Sigma^k)$, where $\mathcal{N}(\cdot)$ denotes the normal distribution, and μ^k and Σ^k denote the mean and variance of kth normal component. Each component is associated a weight w^k which satisfies $\sum_{k=1}^{K} w^k = 1$. The parameter of GMM is denoted by $\theta = \{\mu^k, \Sigma^k, w^k\}$, of which the prior takes the form of normal-Wishart density distribution. In cross-data set detection, $Pr(\theta)$ represents the prior information obtained from source dataset. We believe that in different datasets, the action information might be correlated, and thus the prior $Pr(\theta)$ from source dataset will be beneficial for the action detection in the target dataset.

For the task of action detection, we need to distinguish the action of interest from the background. Thus we employ two GMM models: the background model $\theta_b = \{\mu_b^k, \Sigma_b^k, w_b^k\}$ and the model for action of interest $\theta_c = \{\mu_c^k, \Sigma_c^k, w_c^k\}$. The corresponding prior distributions are denoted as $Pr(\theta_b)$ and $Pr(\theta_c)$, respectively.

This paper models the task of action detection as finding 3D subvolumes in spatial and temporal domains that contain the actions of interests. Let $\mathbf{Q} = \{Q_1, Q_2, \ldots\}$ denote the set of subvolumes each of which contains an instance of action. We take the union of $\mathbf{Q}$ as $U_{\mathbf{Q}} = \bigcup_{Q \in \mathbf{Q}} Q$, and let $\overline{U}_{\mathbf{Q}}$ denotes the complement of $U_{\mathbf{Q}}$. By assuming that each STIP is independent to each other, we can write the log likelihood of STIPs as

$$\begin{aligned} \mathcal{L} &= \log Pr(U_{\mathbf{Q}}; \theta_c) + \log Pr(\overline{U}_{\mathbf{Q}}; \theta_b) \qquad (1) \\ &= \sum_{q \in U_{\mathbf{Q}}} \log Pr(q|\theta_c) Pr(\theta_c) + \sum_{q \in \overline{U}_{\mathbf{Q}}} \log Pr(q|\theta_b) Pr(\theta_b) \end{aligned}$$

To detect action in the the dataset, we need to find the optimal action model parameters θ_c together with the action subvolume set $\mathbf{Q}$ in the new dataset.

$$(\theta_c^*, \mathbf{Q}^*) = \arg\max_{\theta_c, \mathbf{Q}} \mathcal{L} \qquad (2)$$

However, directly optimizing (2) is intractable. An effective approach in practice is to find the solution in an iterative way:

$$\text{given } \mathbf{Q}, \theta_b, \qquad \theta_c^* = \arg\max_{\theta_c} \mathcal{L}(\mathbf{Q}, \theta_c), \quad (3)$$

$$\text{given } \theta_c, \theta_b, \qquad \mathbf{Q}^* = \arg\max_{\mathbf{Q}} \mathcal{L}(\mathbf{Q}, \theta_c), \quad (4)$$

where θ_c^* and $\mathbf{Q}^*$ are updated action model and subvolumes. Note that the background model θ_b is fixed in our model. Eqn (3) and (4) optimize the same objective cost in (1), where the objective value is guaranteed to decrease in each iteration. Thus iteratively minimizing (3) and (4) will converge to a Kuhn-Tucker point.

It is easy to see that (3) gives us a powerful tool to incorporate the cross-dataset information. Suppose we have a labeled source dataset S and an unlabeled target dataset T. We can estimate θ_b by fitting the GMM with all the STIPs in T. However, it is difficult to estimate θ_c since there is no label information of $\mathbf{Q}$ in T. In contrast, we can apply (3) to the source dataset S and obtain θ_c. By using the obtained θ_c as initialization, we leverage the label information in S so that we can iteratively update the estimation of θ_c and $\mathbf{Q}$ in T. The model can be updated efficiently and good results can be obtained after 2 or 3 iterations. Algorithm 1 describes the procedure of our adaptive action detection approach. Note that our algorithm takes determined initialization $Pr(\theta_b)$, and updates $Pr(\theta_c)$ and $\mathbf{Q}$ using Eqn (3) and (4). Unlike the classical EM, Algorithm 1 does not rely on extra initialization algorithm such as K-means or random initialization.

Algorithm 1 Cross-dataset Action Detection

1: **Input**: labeled source dataset S and target dataset T.
2: Train background model $Pr(\theta_b)$ based on all the STIPs in the T.
3: In **Source** dataset S:
4: apply (3) to S and obtain θ_c.
5: In **Target** dataset T:
6: update $\mathbf{Q}$ using (4).
7: update θ_c using (3).
8: repeat the last two steps for several rounds.
9: **Output** the action model and the detected regions in T.

Next we will discuss in detail how to compute θ_c^* and $\mathbf{Q}^*$, respectively.

3.1. Model adaptation

The optimal parameter θ^c should maximize (1):

$$\theta_c^* = \arg\max_{\theta_c} \mathcal{L}(\mathbf{Q}, \theta_c)$$

When $\mathbf{Q}$ is given and θ_b is fixed, the problem is simplified as

$$\theta_c^* = \arg\max_{\theta^c} \sum_{q \in U_{\mathbf{Q}}} \log Pr(q|\theta_c) Pr(\theta_c) \quad (5)$$

We take the model of μ_k^c in source dataset as prior. Since Gaussian distribution is the conjugate prior for Gaussian, and we can obtain the MAP estimation for (5) in a simple form

$$\mu_c^k = \alpha^k \, E_c^k(x) + (1 - \alpha^k)\mu_c^k \quad (6)$$
$$\Sigma_c^k = \beta^k \, E_c^k(x^2) + (1 - \beta^k)(\Sigma_c^k + \mu_c^{k^T}\mu_c^k) - \mu_c^{k^T}\mu_c^k$$

where α^k is the weights which adjust the contribution of model prior to the updated model. E_c^k is the weighted summation of samples in the new dataset. In this paper, we update only μ^k but not Σ^k for the sake of faster speed[2]. The variable E_c^k can be estimated as follows:

$$E_c^k = \frac{1}{\sum_j p_{kj}} \sum_{q_j \in U_{\mathbf{Q}}} p_{kj} q_j \quad (7)$$
$$p_{kj} = \frac{w_k \mathcal{N}(q_j|\mu_k, \Sigma_k)}{\sum_k w_k \mathcal{N}(q_j|\mu_k, \Sigma_k)}$$

Note that the weighting parameter α^k can also be simplified as

$$\alpha_k = \frac{\sum_j p_{kj}}{\sum_j p_{kj} + r}$$

where r is the controlling variable for adaptation.

The adaptation approach in (6) effectively makes use of the prior information from source dataset, and requires only a small amount of training data to obtain the adaptation model.

3.2. Subvolume Detection

Given the model θ_c, we can find the best subvolumes containing the action of interests.

$$\mathbf{Q}^* = \arg\max_{\mathbf{Q}} \mathcal{L}(\mathbf{Q}, \theta_c) \quad (8)$$

To maximize (8), we can write the objective function as

$$\begin{aligned}\mathcal{L} &= \sum_{q \in U_{\mathbf{Q}}} \log Pr(q|\theta_c) Pr(\theta_c) + \sum_{q \in \overline{U}_Q} \log Pr(q|\theta_b) \\ &= \sum_{q \in U_{\mathbf{Q}}} [\log Pr(q|\theta_c) Pr(\theta_c) - \log Pr(q|\theta_b) Pr(\theta_b)] \\ &\quad + \sum_{q \in U_{\mathbf{Q}} \cup \overline{U}_Q} \log Pr(q|\theta_b) Pr(\theta_b) \quad (9)\end{aligned}$$

[2]GMM with a updated Σ^k might fit the data better, but often results in computation instability especially when the number of training samples is small.

The second term is constant given the universal background model. So we can use a simplified form of subvolume detection.

$$\mathbf{Q}^* = \arg\max_{\mathbf{Q}} \sum_{q \in U_{\mathbf{Q}}} \log \frac{Pr(q|\theta_c)Pr(\theta_c)}{Pr(q|\theta_b)Pr(\theta_b)} \quad (10)$$

By assigning each STIP a score

$$f(q) = \log \frac{Pr(q|\theta_c)}{Pr(q|\theta_b)} - T = \log \frac{Pr(q|\theta_c)Pr(\theta_c)}{Pr(q|\theta_b)Pr(\theta_b)},$$

Equation (10) can be rewritten as

$$\mathbf{Q}^* = \arg\max_{\mathbf{Q}} \sum_{q \in U_{\mathbf{Q}}} f(q). \quad (11)$$

This is a multi-instance subvolume search problem. To ensure that the returned subvolumes in $\mathbf{Q}$ are not fragmented, we require that each $Q \in \mathbf{Q}$ is *maximal*, which means that there does not exist any cube Q' such that $f(Q') > f(Q)$ and $Q' \cap Q \neq \emptyset$. As in [22], such a multi-instance search problem can be converted to a series of single max-subvolume search problem. That is, we first search for a 3D cube Q_1^* so that

$$Q_1^* = \arg\max_{Q} \sum_{q \in Q} f(q). \quad (12)$$

and then we set the scores of all the pixels in Q_1^* to be zeros, and then find the second optimal 3D cube Q_2^*, *etc*. The iteration stops when the score of the returned max subvolume is below zero or a small threshold value.

The single subvolume search problem (12) can be solved by Branch-and-bound algorithm [8] [22]. Branch-and-bound approach was first developed for integer programming problems. In recently years, it has been shown be to an efficient technique for object detection in images and action detection in videos [8], [1], [22]. In this paper, we perform max-subvolume search using the 3D branch-and-bound algorithm in [22], which is an extension of the 2D branch-and-bound technique [8]. The detailed technical description of the 3D branch-and-bound algorithm is omitted due to limited space.

4. Experimental Results

Three datasets are employed to test our algorithm: KTH dataset [17], Microsoft Research Action Dataset II [3] (we call it MSR dataset in this paper), and a dataset of the running action in TRECVID surveillance data [19]. The KTH data contains six types of human actions: walking, jogging, running, boxing, hand waving and hand clapping. The MSR dataset contains three actions: boxing, hand waving and hand clapping. The TRECVID dataset contains running actions only. KTH is a popular public dataset, however, it is limited to clean background and a single actor. Since each video sequence exhibits one individual action from beginning to end, locating the actions of interest is trivial. The other two datasets are taken in realistic scenarios, with cluttered background and multiple people in each frame. The actions in the other two datasets are the same type as KTH, but of a more challenging scenario. In this paper, we first compare the performance of our algorithm on KTH dataset with those of previous work. Then we test our algorithm in the cross-dataset setting, where classifiers take only KTH labels for training but MSR or TRECVID for testing.

Table 1. Comparing the accuracy on KTH

Work	Accuracy	Num of training
Schuldt *et al*. [17]	71.71%	16 persons
Dollar *et al*. [2]	80.66%	16 persons
Niebles and Fei-Fei [15]	83.92%	16 persons
Huang *et al*. [6]	91.6%	16 persons
Laptev *et al*. [10]	91.8%	16 persons
Yuan *et al*. [22]	93.3%	16 persons
Liu and Shah [12]	94.16%	16 persons
Our work	**95.02%**	16 persons
Our work	**94.01%**	8 persons
Our work	**90.63%**	4 persons

In KTH dataset, each human action is performed several times by 25 actors. We follow the standard experimental setting of KTH dataset as in [17]. Among the 25 persons, 16 of them are used for training and the rest 9 are used for testing. The training dataset contains 2391 sequences, each of which is associated with one of the six actions. We build background model θ_b using all the STIPs in the dataset, and take the training videos as source dataset. In each video of the KTH dataset, we need not estimate Q since there is only one actor repeating the same action without background motions involved, and all the STIPs in the video are associated with the action. According to (3), we update θ_c for $c = 1, 2, \ldots, 6$, and estimate the action category of testing videos by

$$c_{esti} = \arg\max_{c} \mathcal{L}(\theta_c).$$

Figure 3 shows the confusion matrix of our approach. Table. 4 compares the accuracy of our method with the previous works on KTH dataset using the same experimental setting. Our accuracy 95.02% outperforms the state-of-the-art on KTH dataset. We also test the situation using fewer training labels, and find that with half of the training labels (8 persons), our accuracy 94.01% is comparable with the state of the art 93.3% in [22] and 94.16% in [12].

Next we apply our approach to our MSR dataset. MSR dataset is a new dataset collected by us for detecting ac-

[3] http://research.microsoft.com/∼zliu/ActionRecoRsrc

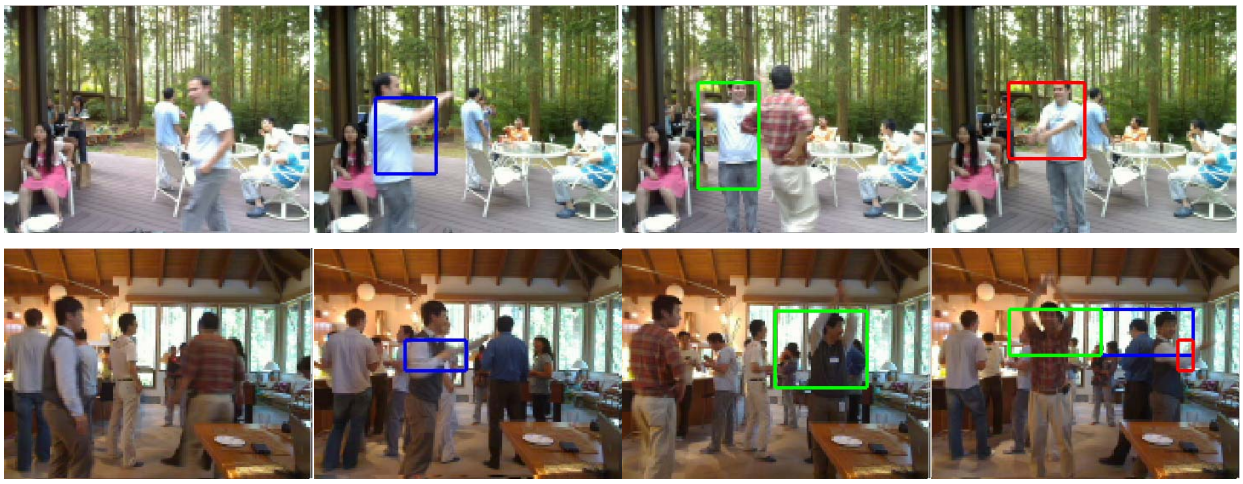

Figure 4. Action detection results on MSR dataset. The located actions are bounded by boxes with different colors: red for hand clapping, green for hand waving, and blue for boxing. Note that our approach works when there are cluttered background and occasional occlusions, and even when more than one actors performing different actions.

	clapping	waving	boxing	jogging	running	walking
clapping	144					
waving	7	137				
boxing	1		143			
jogging				137	3	4
running				28	116	
walking						144

Figure 3. Comparing the results on KTH

tions in complex scenes. It includes 54 video sequences, each of which contains several different actions, e.g., hand waving, clapping, and boxing. These videos are taken with the background of parties, outdoor traffic, and walking people. Actors are asked to walk into the scene, perform one of the three kinds of actions, and then walk out of the scene with these backgrounds. Each video clip is around 1 minute, while most action instances finish in 10 seconds. Through all the videos, people in the background keep talking with each other and walking around as they want. As shown in Figure 2, MSR dataset is different from KTH in that there are a lot of people in the scene and we need to locate the action from cluttered background. To give a concrete idea, of all the STIPs found by Laptev's detector, only 18.73% correspond to three actions of interests (we use the ground truth boxes to determine whether or not a STIP belongs to an action instance), while the rest of the STIPs (81.27%) belong to the background. We use KTH videos with the three actions as source dataset and apply Algorithm 1 to detect the actions from MSR dataset. In the updating stage, we search for the best region in each video using (12), and use the union of these regions as $\mathbf{Q}$.

To evaluate the detection results of our model, we manually labeled the MSR dataset with bounding subvolumes and action types. By denoting the subvolumes ground truth as $\mathbf{Q}^g = \{Q_1^g, Q_2^g, \ldots, Q_m^g\}$, and the detected subvolumes as $\mathbf{Q}^d = \{Q_1^d, Q_2^d, \ldots, Q_n^d\}$, we use $HG(Q_i^g)$ to denote whether a groundtruth subvolume Q_i^g is detected, and $TD(Q_j^d)$ to denote whether a detected subvolume makes sense or not.

$$HG(Q_i^g) = \begin{cases} 1, & \text{if } \exists Q_k^d, \quad s.t. \frac{|Q_k^d \cap Q_i^g|}{|Q_i^g|} > \delta_1 \\ 0, & \text{otherwise,} \end{cases} \tag{13}$$

$$TD(Q_j^d) = \begin{cases} 1, & \text{if } \exists Q_k^g, \quad s.t. \frac{|Q_k^g \cap Q_j^d|}{|Q_j^d|} > \delta_2 \\ 0, & \text{otherwise,} \end{cases}$$

where $|\cdot|$ denotes for the area of the subvolume , and δ_1, δ_2 are parameters to judge the overlapping ratio. In this paper, δ_1 and δ_2 are set as $1/8$.

Based on HG and TD, precision and recall can be defined as

$$\text{Precision} = \frac{\sum_{i=1}^{m} HG(Q_i^g)}{m} \tag{14}$$

$$\text{Recall} = \frac{\sum_{j=1}^{n} TD(Q_j^d)}{n}$$

Given a collection of detected subvolumes, we can compute the precision recall based on (14). By using different threshold of the region scores $\sum_{q \in Q} f(q)$, we can apply branch-and-bound algorithm and thus obtain the Precision-Recall curves for three actions in MSR dataset. Figure 5 compares

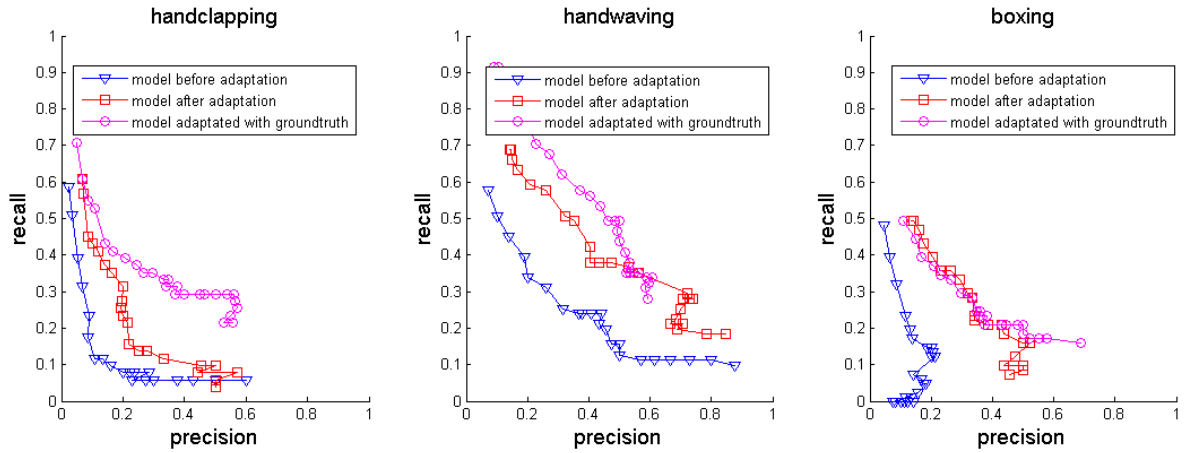

Figure 5. Adaptive action detection on MSR dataset. We compare two kinds of adaptations methods: Adaptation using cross-dataset and adaptation using a small amount of groundtruth labels. Both approaches outperform the original method significantly.

the adaptive action detector with the action model trained from KTH directly. The results show that our adaptive action detector significantly outperforms the original model for all the three actions.

We also try to build the action detection model using a small amount of labeled ground truth. We randomly select four video clips and update the action model θ_c based on the groundtruth. Since the groundtruth is known, we need not update $U_{\mathbf{Q}}$. The results are shown in Figure 5. It is easy to see that our algorithm works well even with a small number of labeled regions. In most of the case, adaptation using groundtruth obtains better performance than the adaptation using cross-domain data. However, as shown in Figure 5, both methods outperforms the original action model.

At last, we test our approach using the TRECVID 2008 dataset. TRECVID surveillance event detection is one of the most challenging action detection task in surveillance [13]. The data used in 2008 challenge is taken by the surveillance cameras in an airport. There are a lot of actions defined in TRECVID 2008, however, we only use the "running" since it overlap with the KTH dataset. We select a portion of TRECVID data by using the data taken by the 2nd camera (total five cameras) and obtain 111 video sequences of the running actions. Each video sequence is elongated to twice of the length of action duration, which incorporates the "non-running" sequence and makes the temporal detection task meaningful. The labels for original TRECVID data are mostly in the format of frame intervals, however, we are more interested in the spatial locations. We manually labeled each running action with a bounding 3D subvolume. Note that it is very time-consuming to label accurate locations frame by frame. In this work we aim to test how the labels from KTH can be used for TRECVID and test the detection results on a portion of the whole dataset with the groundtruth labeled by ourselves. The experiment

Figure 6. Action detection results on TRECVID subdataset. The running person is bounded by a red box. In the first row, one person was running in the crowd with a baby cart. In the second row, another person first ran and then walked away. Our approach detects the running action only.

in a larger scale will be our future work.

The precision-recall curves on TRECVID subdataset are shown in Figure 7. If we use the model from KTH directly, the adaptation results are very poor. However, by using Algorithm 1, we treat KTH dataset as source dataset and train adapted action detector which is significantly better. Figure 6 shows some detection results on TRECVID data using our approach.

5. Conclusion

This paper proposes a new framework for cross-dataset action detection. We build a novel framework which combines GMM-based representation of STIPs and branch-and-

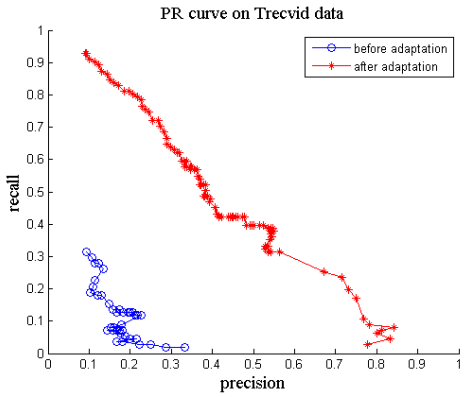

Figure 7. Precision-recall curves on TRECVID subdataset using our adaptive action detection .

bound based detection through MAP estimation. By introducing the prior of GMM parameters, we are able to seamlessly incorporate the information from a source dataset to a target dataset. By performing model adaptation and action detection simutaneously, our technique provides an effective solution for handling data mismatches. Our technique significantly improves detection results on two challenging datasets MSR and TRECVID, both of which consist of videos with crowded scenes. Our future work include improving current method by considering branch-and-bound search method with more flexible parallelepiped instead of axis-aligned 3D cubes, and also employing multiple visual features for action detection.

Acknowledgement

We would like thank Norberto Goussies, Ying-Li Tian, Zhengyou Zhang, Ming Liu, Xi Zhou, Jui-Ting Huang for their discussions and comments. We also thank Philip Chou and other MSR colleagues for the help of collecting MSR Action dataset II. Thanks to Mert Dikmen for the help of building TRECVID sub-dataset.

References

[1] S. An, P. Peursum, W. Liu, and S. Venkatesh. Efficient algorithms for subwindow search in object detection and localization. *CVPR*, 2009.

[2] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. *IEEE International Workshop on VS-PETS*, 2005.

[3] I. Haritaoglu, D. Harwood, and L. Davis. W^4: Real-time surveillance of people and their activities. *PAMI*, 22(8):809–830, 2000.

[4] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang. Action detection in complex scenes with spatial and temporal ambiguities. *ICCV*, 2009.

[5] N. Ikizler and D. Forsyth. Searching video for complex activities with finite state models. *CVPR*, 2007.

[6] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. *ICCV*, 2007.

[7] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. *ICCV*, 2007.

[8] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*.

[9] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, pages 951–958, 2009.

[10] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. *IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[11] J. Liu, J. Luo, and M. Shah. Recognizing realistic actions from videos "in the wild". *CVPR*, 2009.

[12] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, 2008.

[13] M. Dikmen *et. al.* Surveillance event detection. In *TRECVID Video Evaluation Workshop*, 2008.

[14] M. Marszałek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, 2009.

[15] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *British Machine Vision Conference*, 2006.

[16] D. Reynolds, T. Quatieri, and R. Dunn. Speaker verification using adapted Gaussian mixture models. *Digital Signal Processing*, 10(1-3):19–41, 2000.

[17] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. *ICPR*, 2004.

[18] C. Stauffer and W. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, volume 2, pages 246–252, 1999.

[19] TREC Video Retrieval Evaluation. Surveillance event detection, http://www-nlpir.nist.gov/projects/tv2008/, 2008.

[20] S.-F. Wong, T.-K. Kim, and R. Cipolla. Learning motion categories using both semantic and structural information. *CVPR*, 2007.

[21] J. Yang, R. Yan, and A. Hauptmann. Cross-domain video concept detection using adaptive SVMs. In *ACM Multimedia*, 2007.

[22] J. Yuan, Z. Liu, and Y. Wu. Discriminative subvolume search for efficient action detection. *CVPR*, 2009.

[23] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *CVPR*, 2005.

[24] X. Zhou, X. Zhuang, S. Yan, S.-F. Chang, M. Hasegawa-Johnson, and T. S. Huang. Sift-bag kernel for video event analysis. In *ACM International conference on Multimedia*, pages 229–238, 2008.