Adaptive Action Detection

Illinois Vision Workshop
Dec. 1, 2009

Liangliang Cao
Dept. ECE, UIUC

Zicheng Liu
Microsoft Research

Thomas Huang
Dept. ECE, UIUC
Motivation

- Action recognition is important in surveillance, however, it is an extremely tedious work to label the action locations frame by frame.
Motivation

• Action recognition is important in surveillance, however, it is an extremely tedious work to label the action locations frame by frame.

• Is it possible to use the existing labels from another dataset?
Difficulties in cross-dataset action recognition

No computer vision algorithm works well on two different datasets.

---- Ram Nevatia

Given two datasets containing the same kind of action, they usually endure

• different background
• various performer
• different lighting condition, scale, action speed
Difficulties in cross-dataset action recognition

- If we train a classifier on the source dataset and test it on the target dataset, the performance usually drops significantly.

Source dataset (with labels)

Target dataset (without labels)

boxing  waving  clapping  running
Goal: How to recognize action across different datasets?

• There are no or minimum labels in target dataset.
• There are two sources of information we can use to alleviate the requirement of labels
  • Spatial-temporal coherence of actions
  • The labels from source dataset can be treated as “prior” for the MAP estimation
Our approach: Adaptive Action Detection

Features: Laptev’s spatial-temporal interest points (STIPs)
Feature models: 512-component Gaussian Mixture Models (GMMs)
Related works

Our approach is motivated mainly by two works

• Adaptive tracking [Stuffer&Grimson, CVPR’99]
  – Similarity: both use GMM adaptation
  – Differences:
    • Stuffer aims to keep up with the slow but continuous change over time, while we aim to adapt the model to a very different dataset.
    • Our approach combines action detection and model adaptation into a single framework, while [Stuffer99] only considers the adaptation issue.
Our approach is motivated mainly by two works

- Adaptive tracking [Stuffer&Grimson, CVPR’99]
- Discriminative action detection [Yuan, Liu & Wu, CVPR’09].

  - **Similarity**: both employ branch-and-bound approach to locate the action
  - **Differences**:
    - [Yuan 09] models STIPs with the mutual information scores based nearest-neighbor search, while this paper employs GMM to represent STIP whose posterior can be effectively estimated from the source dataset.
    - [Yuan 09] does 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.
Adaptive Action Detection

\[ \mathcal{L} = \log Pr(U_Q; \theta_c) + \log Pr(\overline{U}_Q; \theta_b) \]
\[ = \sum_{q \in U_Q} \log Pr(q|\theta_c) Pr(\theta_c) + \sum_{q \in \overline{U}_Q} \log Pr(q|\theta_b) Pr(\theta_b) \]

**Notation**

- \( Q \): set of subvolumes containing the actions
- \( U_Q \): union of \( Q \)
- \( \overline{U}_Q \): complement of \( U_Q \)
- \( q \): spatial-temporal interest point feature
- \( \theta_c \): parameters modeling the action of interest
- \( \theta_b \): parameters modeling the background
Adaptive Action Detection

\[ \mathcal{L} = \sum_{q \in U_Q} \log Pr(q|\theta_c) Pr(\theta_c) + \sum_{q \in \overline{U}_Q} \log Pr(q|\theta_b) Pr(\theta_b) \]

Combining two tasks:

1. Model adaptation:
   \[ \text{given } Q, \theta_b, \quad \theta^*_c = \arg \max_{\theta_c} \mathcal{L}(Q, \theta_c) \]

2. Action localization:
   \[ \text{given } \theta_c, \theta_b, \quad Q^* = \arg \max_Q \mathcal{L}(Q, \theta_c), \]
target dataset

(2): Action localization

Distribution of STIPs

source dataset

(1): model adaptation

Distribution of STIPs

- boxing action
- background

Distribution of STIPs

- background
Adaptive Action Detection

\[ \mathcal{L} = \sum_{q \in U_Q} \log Pr(q|\theta_c)Pr(\theta_c) + \sum_{q \in \overline{U_Q}} \log Pr(q|\theta_b)Pr(\theta_b) \]

- The distribution of \( q \): Gaussian mixture model
  \[ \theta_b = \{\mu_b^k, \Sigma_b^k, w_b^k\} \quad \theta_c = \{\mu_c^k, \Sigma_c^k, w_c^k\} \]
  \[ \sum_{k=1}^{K} w^k N(q; \mu^k, \Sigma^k) \]

- The prior: normal-wishart for each component (Conjugated distribution of Gaussian)
  \[ Pr(\theta_c) = \prod_k Pr(\mu_c^k, \Sigma_c^k) \]
Adaptive Action Detection

\[ \mathcal{L} = \sum_{q \in U_Q} \log Pr(q|\theta_c) Pr(\theta_c) + \sum_{q \in \overline{U}_Q} \log Pr(q|\theta_b) Pr(\theta_b) \]

- Model updating:
  - Only update foreground models
  - Do not update background models.
  - Background model is trained on target dataset without label.
Adaptive Action Detection

\[ \mathcal{L} = \sum_{q \in U_Q} \log Pr(q|\theta_c) Pr(\theta_c) + \sum_{q \in U_Q} \log Pr(q|\theta_b) Pr(\theta_b) \]

• Task I: Model adaptation
  – The second part is not related to \( \theta_c \)
  \[ \theta_c^* = \arg \max_{\theta_c} \sum_{q \in U_Q} \log Pr(q|\theta_c) Pr(\theta_c) \]
  – Solutions:
  \[ \mu_c^k = \alpha^k E^k_c(x) + (1 - \alpha^k)\mu_c^k \]
  \[ \Sigma_c^k = \beta^k E^k_c(x^2) + (1 - \beta^k)(\Sigma_c^k + \mu_c^k \mu_c^k) - \mu_c^k \mu_c^k \]

Do not update \( \Sigma_c^k \) for faster speed and better robustness
Adaptive Action Detection

$$\mathcal{L} = \sum_{q \in U_Q} \log Pr(q|\theta_c) Pr(\theta_c) + \sum_{q \in \overline{U}_Q} \log Pr(q|\theta_b) Pr(\theta_b)$$

- Task I: Model adaptation
  - The second part is not related to $\theta_c$

$$\theta^*_c = \arg \max_{\theta_c} \sum_{q \in U_Q} \log Pr(q|\theta_c) Pr(\theta_c)$$

- Solutions:

$$\mu^k_c = \alpha^k E^k_c(x) + (1 - \alpha^k)\mu^k_c$$

$$E^k_c = \frac{1}{\sum_j p_{kj}} \sum_{q_j \in U_Q} p_{kj} q_j$$

$$p_{kj} = \frac{\omega_k \mathcal{N}(q_j|\mu_k, \Sigma_k)}{\sum_k \omega_k \mathcal{N}(q_j|\mu_k, \Sigma_k)}$$

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

\[ \mathcal{L} = \sum_{q \in U_Q} \log Pr(q|\theta_c) Pr(\theta_c) + \sum_{q \in \overline{U}_Q} \log Pr(q|\theta_b) Pr(\theta_b) \]

- Task II: Subvolume detection

\[ \mathcal{L} = \sum_{q \in U_Q} \log Pr(q|\theta_c) Pr(\theta_c) + \sum_{q \in \overline{U}_Q} \log Pr(q|\theta_b) \]

\[ = \sum_{q \in U_Q} [\log Pr(q|\theta_c) Pr(\theta_c) - \log Pr(q|\theta_b) Pr(\theta_b)] \]

\[ + \sum_{q \in U_Q \cup \overline{U}_Q} \log Pr(q|\theta_b) Pr(\theta_b) \]

not related to \( U_Q \)
Adaptive Action Detection

$$\mathcal{L} = \sum_{q \in U_Q} \log Pr(q|\theta_c) Pr(\theta_c) + \sum_{q \in \overline{U}_Q} \log Pr(q|\theta_b) Pr(\theta_b)$$

- Task II: Subvolume detection

$$\mathcal{L} = \sum_{a \in U \cap \Theta} [\log Pr(q|\theta_c) Pr(\theta_c) - \log Pr(q|\theta_b) Pr(\theta_b)]$$

$$Q^* = \arg \max_Q \sum_{q \in U_Q} \log \frac{Pr(q|\theta_c) Pr(\theta_c)}{Pr(q|\theta_b) Pr(\theta_b)}$$

by assigning each $q$ a score,

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

we can find $Q^*$ using Branch and Bound
Branch-Bound

• A bounding box is represented by \((l, r, t, b)\)
Branch-Bound

- A bounding box is represented by \((l, r, t, b)\)
- A searching space is represented by \((L, R, T, B)\)

\[
L = [l_{lo}, l_{hi}]
\]
\[
T = [t_{lo}, t_{hi}]
\]
\[
R = [r_{lo}, r_{hi}]
\]
\[
B = [b_{lo}, b_{hi}]
\]
Branch-Step: Splitting Boxes

rectangle set \([L, R, T, B]\)

\([L, R_1, T, B]\) with \(R_1 := [r_{lo}, \lceil \frac{r_{lo} + r_{hi}}{2} \rceil] \)

\([L, R_2, T, B]\) with \(R_2 := \lceil \frac{r_{lo} + r_{hi}}{2} \rceil + 1, r_{hi} \)
Compute the upper-bound function

\[ f^+(Q) = \sum_{q \in Q} \max(f(q), 0) \]

\[ f^-(Q) = \sum_{q \in Q} \min(f(q), 0), \]

\[ f(Q) \leq f_0(Q) = f^+(Q_{\text{max}}) + f^-(Q_{\text{min}}) \]

• With the help of integral image or integral video, the upper-bound can be computed in \( O(1) \) time
Maintaining the priority queue

• Three actions for the priority queue:
  – Insert a searching region
  – Access the min value
  – Delete the min value

• To speed up, employ Yuan and Liu’s tighter bound

<table>
<thead>
<tr>
<th></th>
<th>Min-Heap</th>
<th>Fibonacci Heap</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert a region</td>
<td>$O(\log(n))$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Access the min value</td>
<td>$O(1)$</td>
<td>$O(1)$</td>
</tr>
<tr>
<td>Delete the min value</td>
<td>$O(\log(n))$</td>
<td>$O(\log(n))$</td>
</tr>
</tbody>
</table>

\[
F_1(W) = F(W_{\text{min}}) + \sum_{\substack{q \in Q_{\text{max}}, q \notin Q_{\text{min}}}} \lceil \max_f(q) \rceil_+ \\
F_2(W) = F(W_{\text{max}}) - \sum_{\substack{q \in Q_{\text{max}}, q \notin Q_{\text{min}}}} \lfloor \min_f(q) \rfloor_-
\]

\[
F(W) \leq \min(f_1(W), F_2(W))
\]
Study I: Action classification on KTH
<table>
<thead>
<tr>
<th>Work</th>
<th>Accuracy</th>
<th>Num of training</th>
</tr>
</thead>
<tbody>
<tr>
<td>Schuldt et al. [14]</td>
<td>71.71%</td>
<td>16 person</td>
</tr>
<tr>
<td>Dollar et al. [2]</td>
<td>80.66%</td>
<td>16 person</td>
</tr>
<tr>
<td>Niebles and Fei-Fei [12]</td>
<td>83.92%</td>
<td>16 person</td>
</tr>
<tr>
<td>Huang et al. [6]</td>
<td>91.6%</td>
<td>16 person</td>
</tr>
<tr>
<td>Laptev et al. [9]</td>
<td>91.8%</td>
<td>16 person</td>
</tr>
<tr>
<td>Yuan et al. [19]</td>
<td>93.3%</td>
<td>16 person</td>
</tr>
<tr>
<td>Liu and Shah [10]</td>
<td>94.16%</td>
<td>16 person</td>
</tr>
<tr>
<td><strong>Our work</strong></td>
<td>95.02%</td>
<td>16 person</td>
</tr>
<tr>
<td><strong>Our work</strong></td>
<td>94.01%</td>
<td>8 person</td>
</tr>
<tr>
<td><strong>Our work</strong></td>
<td>90.63%</td>
<td>4 person</td>
</tr>
</tbody>
</table>
Are running and jogging so difficult to be distinguished?
Study II: Cross dataset action detection on UD action detection
Experimental Settings

• Adapted models train in KTH to UD (Upperbody action Dataset)
• UD dataset
  – 54 videos, each video contain multiple actions
  – 3 actions: hand waving, clapping, and boxing
  – Actors walk into the scene, perform one of the three kinds of action, and then walk out of the scenes.
  – Background of parties, outdoor traffic, and walking people.
Detection on UD dataset

• Videos
Detection on UD dataset

• Numerical evaluation
Study II: Cross dataset action detection on TRECVID subdataset
Dataset

• TRECVID 2008 surveillance: taken by airport surveillance cameras.

• Sub-dataset used in our experiments
  – Only “running” action
  – Videos from 2th camera (total 5 cameras)
  – running into the scene, or walking then running.
  – Train on KTH, and test on 111 TRECVID video sequences.
Detection on TRECVID sub-dataset
Detection on TRECVID sub-dataset

- Precision-Recall curve

**PR curve on Trecvid data**

- before adaptation
- after adaptation
Challenging Instance from TRECVID

• A failure example: running kid with clothes similar with background color.
Messages from this project

• Our approach can help with
  – Effectively leveraging labels from cross dataset
  – Save human labeling efforts

• New dataset to be release soon:
  – UD dataset: 54 videos, three actions
  – TRECVID dataset: Original dataset mainly provide frame-level labels. We are labeling more detailed spatial locations. Working in progress.
Room to improve

• Better features in additional STIPs
  – One paper under review on new descriptors
  – Future work: multiple feature detection

• Improving “running” action detection
  – Current branch –and-bound approach finds a 3D cube for moving actions.
  – A better is to find parallelepiped, or more general 3D shape.

• Interface which allows flexible human interaction
  – When the user is not satisfied with the results, he might label instances, which improves the detection.
Acknowledgment

• Helps and Discussions from Norberto Goussies, Ying-Li Tian, Ming Liu, Xi Zhou, Jui-Ting Huang and Zhengyou Zhang

• Thanks to Philip Chou for the help of collecting UD dataset. Thanks to Mert Dikmen for the help of building TRECVID sub-dataset.

Questions and comments?