Fusing Feature Descriptors for Action Recognition

EECS 6890 Visual Recognition and Search, Spring 2013

Instructor: Rogerio Feris and Liangliang Cao
Columbia University

Group: Jiawei Chen jc3960
Qian Liang ql2198
Chen Xue cx2146
Applications: Surveillance

Monitor suspicious activities for real-time reactions.
Applications: Human-machine Interface

Minimize the barrier between the human's cognition and the computer's understanding
Applications: Sports Events Analysis

Analyze what play it is and players’ behavior.
How can we accomplish them?

ACTION RECOGNITION

- Feature tracking
- Motion history
- Spatio-Temporal Interest Points (STIP)
- Dense Trajectories
Different methods

- Spatio-Temporal Interest Points (STIP)

**STIP**

- **Detectors**
  - Harris3D
  - Hessian
  - Cuboids
  - Dense

- **Descriptors**
  - HOF
  - HOG
  - HOG/HOF
  - HOG3D

System Framework

1. Input video
2. Interest point detection
3. Feature representation
4. K-means clustering
5. Feature quantization
6. SVM
7. Codebook generation
8. Action classification
Standard dataset
- HMDB 51 categories of actions
- 7,000 clips

Self-recorded dataset
- 380 clips
- 12 categories of actions
  - Corresponding to 12 actions in HMDB51:
    - brush hair, catch, chew, drink, hit, jump, pick, punch, push, shake hands, smile, turn
Harris3D

The distinct interest points are detected at moments and at spatial positions where the hand changes its direction of motion.

System Framework

Input video → Interest point detection → Feature representation → K-means cluster

K-means cluster → Feature quantization → SVM → Action classification

Visual Recognition and Search - Page 10
STIP Descriptors
- Cuboid: spatial-temporal window at interest point
  - Local features
    - Histogram of Optical flow (HOF) – 90D
    - Histogram of Oriented Gradient (HOG) – 72D
    - HOG/HOF – 162D

Image descriptors extended to Spatial-temporal Domain
- HOG3D – 305D

Implementation:
- Feature descriptor
System Framework

- Input video
- Interest point detection
- Feature representation
- K-means cluster
- Feature quantization
- SVM
- Codebook generation
- Action classification
Follow the instructions on HMDB51 official website

Build a spatio-temporal bag-of-features (BoF)

Number of clusters: \( k = 2000 \)
- Faster calculation
- Consistent with static image classification

Non-linear SVM: RBF

\[
K(H_i, H_j) = \exp\left(-\frac{1}{A}D(H_i, H_j)\right),
\]
\[
D(H_i, H_j) = \frac{1}{2} \sum_{k} \frac{(H_i(k) - H_j(k))^2}{H_i(k) + H_j(k)},
\]

\( A \) is the average distance between all training examples
System Framework

1. Input video
2. Interest point detection
3. Feature representation
4. K-means cluster
5. Feature quantization
6. SVM
7. Codebook generation
8. Action classification
## Comparison with standard results on HMDB51

<table>
<thead>
<tr>
<th></th>
<th>HMDB51 website results</th>
<th>Our results</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-means</td>
<td>8000</td>
<td>2000</td>
</tr>
<tr>
<td>Codebook samples</td>
<td>100,000 STIP descriptors from training dataset</td>
<td>6656×20 STIP descriptors from all dataset</td>
</tr>
<tr>
<td>SVM kernel</td>
<td>RBF</td>
<td>RBF</td>
</tr>
<tr>
<td>SVM Parameter</td>
<td>5-folder cross-validation</td>
<td>5-folder cross-validation (train_auto)</td>
</tr>
<tr>
<td>Training accuracy</td>
<td>N/A</td>
<td>99.972%</td>
</tr>
<tr>
<td>Testing accuracy</td>
<td>20.44%</td>
<td>20.4575%</td>
</tr>
</tbody>
</table>

Previous Update2 problem solved  
Framework now works fine
Fusing Different Features
Another Method

- Dense Trajectories

Use dense sampling instead of feature detector
The novel descriptor based on motion boundary histograms (MBH)

Wang et al, CVPR 2011, IJCV 2012
- Input video: HMDB51
- Compute descriptor:
  - HOG(dense)
  - HOF(dense)
  - Trajectories
  - MBH_x
  - MBH_y

Implementation: [http://lear.inrialpes.fr/people/wang/dense_trajectories](http://lear.inrialpes.fr/people/wang/dense_trajectories)

Advantage of MBH

- **Optical flow**
  - Represents absolute motion between frames
  - Contain object motion + background camera motion
  - **Problem**: camera motion disturbs

- **Motion boundary histograms (MBH)**
  - Relative motion between pixels
  - Gradient of optical flow
  - Robust to camera motion

System framework – STIP (recap)

1. Input video
2. Interest point detection
3. Feature representation
4. K-means cluster
5. Feature quantization
6. SVM
7. Codebook generation
8. Action classification
System framework – Dense trajectories

Input video → Dense sampling → Trajectory-aligned descriptors → K-means cluster

Action classification ← SVM ← Feature quantization ← Codebook generation
How to fuse?

Early fuse
- Concatenate histograms
- Normalize kernel matrices to sum up

Late fuse
- Combine output of classifiers (linear SVM)
Fuse method

Trajectory-aligned descriptors

- HOG
- HOF
- Trajectories
- MBHx
- MBHy

Fuse descriptors

- HOGHOF(dense)
- Trajectories
- MBH

96 + 108 = 204D
96 + 96 = 192D
30D
Fuse system

- **BOW**
  - 2000-codebook for each feature

- **Train 3 classifiers**
  - SVM for each kind of histograms

- **Linear SVM**
  - SVM for output scores of three classifiers
Use the same system as STIP’s
Testing results on HMDB51 dataset:

<table>
<thead>
<tr>
<th></th>
<th>Paper result (4000 codebook)</th>
<th>Our result (2000 codebook)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOG</td>
<td>27.9%</td>
<td>HOGHOF:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>24.9%</td>
</tr>
<tr>
<td>HOF</td>
<td>31.5%</td>
<td></td>
</tr>
<tr>
<td>Trajectories</td>
<td>28.0%</td>
<td>16.8%</td>
</tr>
<tr>
<td>MBH</td>
<td>43.2%</td>
<td>34.2%</td>
</tr>
</tbody>
</table>
### Comparison of fusion results

#### Our result (2000 codebook)

<table>
<thead>
<tr>
<th>Method</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>HOGHOF</td>
<td>24.9%</td>
</tr>
<tr>
<td>Trajectories</td>
<td>16.8%</td>
</tr>
<tr>
<td>MBH</td>
<td>34.2%</td>
</tr>
</tbody>
</table>

#### Fuse method

<table>
<thead>
<tr>
<th>Fuse method</th>
<th>Our result (2000 codebook)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MBH + HOGHOF</td>
<td>29.30%</td>
</tr>
<tr>
<td>MBH + Trajectories</td>
<td>29.56%</td>
</tr>
<tr>
<td>HOGHOF + Trajectories</td>
<td>20.79%</td>
</tr>
<tr>
<td>MBH + HOGHOF + Trajectories</td>
<td>29.37%</td>
</tr>
</tbody>
</table>
Comparison of fusion results

HOGHOF 24.9%

MBH 34.2%

TRAJ 16.8%

HOGHOF + MBH + TRAJ 29.4%
Thoughts on Fusion

- Fuse complementary features
- Fusion may not necessarily improve the accuracy compared with individual features
- Dense Trajectories has better results – but too much computation
Some thoughts

- Dense trajectories compare with STIP
  - Pros:
    - Robust to camera motion
    - Relatively good accuracy
  - Cons:
    - Low efficiency (real time)
    - Relatively large for our computers:

Semantic Models

- Low level features (like histograms, STIP)
- High level: Semantic features (attribute-based, action bank, Scene Aligned Pooling)
Fusing Different Dataset
Testing on Self-recorded Dataset

- randomly select 10 videos clips in each action class to predict
- chance: 0.083333
Testing on Self-recorded Dataset: codebook: all of recorded dataset

Training: rest of recorded dataset
Will performance be better when having more training data?

1) STIP features are efficient for a new dataset
2) As the number of training samples increases
   1) training accuracy tends to drop
   2) Testing accuracy tends to increase, but not always…

Possibly kick out some outliers by accident
Testing on Self-recorded Dataset: Using HMDB51 dataset

<table>
<thead>
<tr>
<th>training codebook</th>
<th>360 HMDB12</th>
</tr>
</thead>
<tbody>
<tr>
<td>HMDB12</td>
<td>13.33%</td>
</tr>
<tr>
<td>Self-recorded</td>
<td>13.33%</td>
</tr>
<tr>
<td>HMDB 12 + self-recorded</td>
<td>16.67%</td>
</tr>
</tbody>
</table>

Fig: dropping of prediction accuracy when adding more samples to the training set from HMDB51

a) HMDB 51 and our recorded videos are quite different datasets
b) bias exists although dataset is large
c) the self-recorded dataset is “highly biased”
d) better to utilize as much as we can from the prediction dataset
e) results using HMDB dataset strange?
Confusion Matrix

Codebook: recorded dataset
Training: 260 recorded videos
Testing: 120 recorded videos
Recognition accuracy: 93.3%
Exampe of self-recorded dataset
Future work

- Train larger code-book $\rightarrow$ improvement?

- Try other feature fusion method

- Investigate better method to utilize multiple dataset