Dhananjay Kumar
Department of Information Technology
Anna University, MIT Campus
Chennai, India

Session 8: Security in industrial applications

Paper S8.2
Outlines

- Introduction
- Motivation
- System architecture
- Feature Vector Optimization
- Experimental results
- Summary
Introduction

Approaches in Sensor-based Human Activity Recognition

- Logic and reasoning
  - Inherent infeasibility to handle uncertainty
  - Limitation of learning ability with logic based techniques

- Probabilistic model
  - Generative (e.g., HMM, Bayesian Networks) or discriminative models (e.g., Conditional random fields)

- Data mining-based methods
  - Mining a set of pattern of features, activity model
  - **Skeleton featured-based**
    - Skeleton feature of human-subjects with different body positions
Motivation

Learning based Approaches for Vision-based Activity Recognition

- Before Deep Learning
  - Hand crafted features (e.g., HOG) from sparsely / densely sampled trajectories
  - Hand-crafted vs. learned features
    - Bag of words
    - Frame level processing

- Post Deep Learning Approaches
  - The fusion of spatial and temporal data across streams
  - Creation of multi-level loss to handle temporal dependencies in long term
System Architecture

**Pose Estimation**
- Video stream
  - Joints Prediction
  - Heat Map
- Estimated skeleton
- Hip Transformation
- Theta Transformation

**Preprocessing**
- Joint Vector
- Body height
- Normalized JV
- Body Disp
- Joint Disp

**Feature Extraction**
- Fisher Vector
  - Classification
    - Action Class
  - Dimensionality Reduction-PCA
- Add new person label
- Update the action of the person

**Figure 1** – The architecture of the proposed model
Skeleton Generation and Processing

The heat-map is used to represent a joint location in skeleton of human-subject in an image.

1. Hip Transformation
To make the skeletons invariant to the location of the subjects,

\[
[x'_j, y'_j] = [x_j - x_{hip\text{center}}, y_j - y_{hip\text{center}}]
\]

Where \(x_{hip\text{center}}\) and \(y_{hip\text{center}}\) represent the hip center of the input skeleton.

2. Theta Transformation
To make the poses rotation invariant, a rotation operation is applied on the joints relative to the camera view angle \(\theta\).

\[
\theta = \tan^{-1} \left( \frac{y_{right\_hip} - y_{left\_hip}}{x_{right\_hip} - x_{left\_hip}} \right)
\]

\[
\begin{bmatrix}
x' \\
y' \\
1
\end{bmatrix} = \begin{bmatrix}
\cos \theta & \sin \theta & 1 \\
-\sin \theta & \cos \theta & 1 \\
0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
x \\
y \\
1
\end{bmatrix}
\]

Figure 2 - Stacked hourglass architecture for 2D skeleton
The **Fisher Vector** (FV) and **dimensionality reduction** using **PCA** are applied for the optimization of the features.

The FV encodes the gradients of the log-likelihood of the features under the **Gaussian-Mixture-Model** (GMM), with respect to the GMM parameters.

The **PCA** is used to preserve the essential parts that have more variation of the data and remove the non-essential parts with fewer variations.

The multi-class classification is carried out by a **DNN** or **SVM**.

**Figure 3** - Feature vector optimization
Implementation Overview

The data sets used to train and evaluate the model:
- MSR Action Dataset
- NTU RGB+D 3D Skeletal Dataset

Implementation using Python programming with
- Flask framework - Web server implementation
- OpenCV library – Processing of video stream at frame level
- Keras library with Tensorflow - Design of convolutional neural networks

The multi-class classification is carried out by either
1. A one-vs-rest SVM or
2. A three-layer multi-layer perceptron (MLP) DNN

Fig: System display of the recognized action
Fig. Bounding boxes in two MSR action data sets
The SVM is trained with the help of a feature vector generated from the MSR Action Data Set.

### Table 1: Confusion matrix of the one-vs.-rest SVM

<table>
<thead>
<tr>
<th>Action label</th>
<th>Wave</th>
<th>Punch</th>
<th>Kick</th>
<th>Squat</th>
<th>Sit</th>
<th>Jump</th>
<th>Run</th>
<th>Walk</th>
<th>Stand</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wave</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>343</td>
</tr>
<tr>
<td>Punch</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>237</td>
<td>5</td>
</tr>
<tr>
<td>Kick</td>
<td>15</td>
<td>4</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>296</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Squat</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>279</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sit</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>562</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Jump</td>
<td>39</td>
<td>5</td>
<td>23</td>
<td>246</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Run</td>
<td>6</td>
<td>6</td>
<td>274</td>
<td>11</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Walk</td>
<td>26</td>
<td>327</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Stand</td>
<td>460</td>
<td>33</td>
<td>7</td>
<td>15</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2: Performance metrics of the action classifier

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1-score</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stand</td>
<td>0.83</td>
<td>0.89</td>
<td>0.86</td>
<td>519</td>
</tr>
<tr>
<td>Walk</td>
<td>0.87</td>
<td>0.91</td>
<td>0.89</td>
<td>361</td>
</tr>
<tr>
<td>Run</td>
<td>0.87</td>
<td>0.9</td>
<td>0.88</td>
<td>306</td>
</tr>
<tr>
<td>Jump</td>
<td>0.88</td>
<td>0.76</td>
<td>0.82</td>
<td>322</td>
</tr>
<tr>
<td>Sit</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>563</td>
</tr>
<tr>
<td>Squat</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>279</td>
</tr>
<tr>
<td>Kick</td>
<td>0.95</td>
<td>0.93</td>
<td>0.93</td>
<td>326</td>
</tr>
<tr>
<td>Punch</td>
<td>1</td>
<td>0.99</td>
<td>0.99</td>
<td>243</td>
</tr>
<tr>
<td>Wave</td>
<td>0.96</td>
<td>0.96</td>
<td>0.96</td>
<td>352</td>
</tr>
</tbody>
</table>

\[
\text{Precision} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalsePositives}}
\]

\[
\text{Recall} = \frac{\text{TruePositives}}{\text{TruePositives} + \text{FalseNegatives}}
\]
Figure 4 - Precision-recall plot of the proposed DNN-based classifier.

It shows the trade-off between precision, a measure of result relevancy, and recall, a measure of how many relevant results are returned.

A large area under the curve indicates high recall and corresponding precision values.

The average precision score of the proposed DNN-based classifier, micro-averaged over all the action classes, is 0.85.
Results (cont.)

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}
\]

where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

Table 3: Comparison of methods based on accuracy

<table>
<thead>
<tr>
<th>S. No.</th>
<th>Method</th>
<th>Accuracy %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Skeleton Feature + SVM</td>
<td>92.4</td>
</tr>
<tr>
<td>2</td>
<td>HGN+DNN</td>
<td>95.6</td>
</tr>
</tbody>
</table>

Figure 5: Training and test accuracy of SVM and DNN
The classification model is trained on two types of processed skeleton data.

In the first type,
- the data from each frame of the video is processed separately and
- the skeleton data is used to extract and generate the feature vector on which the classifiers are trained.

In the second type,
- five frames are taken as a sliding window and the skeleton data obtained from these are concatenated and
- used to extract the features and generate the vector.

When these concatenated frames are used, it improves the accuracy of both the SVM and DNN model.

Figure 6: Accuracy of different types of Feature vector
Summary

Conclusion

- A combination of two models HGN and DNN to capture the action performed by the human subject and to recognize the action.
- The proposed system achieved an accuracy of 95.6% in action recognition on two different standard data sets of MSR Action and NTU RGB+D 3D skeleton.
- It meets the requirements of service description for video surveillance specified in Recommendation ITU-T F.743.

Future Work

- Standardization as an extension of the intelligent visual surveillance system architecture specified in Recommendation ITU-T H.626.5.
Thank you!