Using Convolutional Layer Features for Indoor Human Activity Recognition based on Spatial Location Information

Jun Li, Jiaxiang Zhao, Jing Li and Yingdong Ma*

IT Academy, Inner Mongolia University, 010021 Inner Mongolia, China

ABSTRACT

Identifying human actions has great importance for various applications, especially in the smart home, fitness tracking and health monitoring domains. However, human activity recognition still remains a challenging task. This is mainly due to the broad range of human activities as well as the rich variation of a given activity can be performed. In this paper, we dealt with the problem by making use of spatial location information of three different parts of a human body, which are derived via three UWB (ultrawide band) tags and an Ubisense positioning system. In order to improve the accuracy, we proposed a recognition method: convolutional layer features plus SVM (Support Vector Machine). We pre-process the raw spatial location data and transfer them into motion feature, frequency feature and statistic feature. These features are input into the CNN (Convolutional Neural Network) to generate the convolutional layer features, and then we use SVM to classify these features. By comparing the experimental results, the best recognition rate of different experimenters is 89.75%, which shows its feasibility.

INTRODUCTION

In recent years, the rapid spread of devices with sensing capabilities has created a huge demand for human activity recognition (AR). Applications that can benefit from AR include smart home, smart healthcare, senior care, personal fitness, etc. [1, 2]. Nowadays, many approaches with different features were proposed to recognize a wide range of activities. Most of these approaches rely on the statistic features, such as mean, variance, entropy or correlation coefficient. In general, existing approaches can be divided into two classes: vision-based methods and sensor-based methods [3]. With the rapid development of MEMS (Micro Electro Mechanical Systems) and sensing technologies, the sensor-based category is becoming more popular, and the method proposed in this work belongs to the second category. Different from the existing sensor-based approaches that strongly depend on various sensors, this paper focuses on the issue of recognizing indoor human actions via spatial location information.

In this work, Ubisense system [4] was chosen to collect the data. Ubisense has developed a platform for building Smart Space applications. The platform addresses the key requirements for building Smart Spaces accurate 3D positioning, scalable real-time performance, and development and deployment tools. What is
more, the Ubisense system presents the following advantages: firstly, it is a
non-line-of-sight technology with a range in free space of a few tens of meters,
which makes it practical to cover large indoor areas; secondly, it is easy to filter the
signal to minimize the multipath distortions which are the main cause of
inaccuracy in RF indoor location systems; and finally, it has low power
consumption, fine safety, high reliability and utility value. Then, we use the
convolutional layer features of Convolutional Neural Networks (CNN) [5] to
recognize activities over the Ubisense system.

RELATED WORK

AR can be considered as a classification problem, where the input is time
series signal and the output is corresponding class label of the activity. In general,
the process over activity recognition is divided into two phases: to begin with, we
extract features from the raw time series data; next these features with the labels
are used to train a classification model.

Feature extraction for AR is one of the key components. Generally, statistic
features such as mean, standard deviation, entropy and correlation coefficients are
the traditional hand-deigned features in AR. In addition, Fourier transforms [6],
wavelet transforms [7] and discrete cosine transforms (DCT) [8, 9] are the other
three commonly used hand-crafted features.

In recent years, some approaches such as principal component analysis (PCA)
[10] and restricted Boltzmann machine (RBM) [11] were also applied to adapt the
feature extraction over the dataset. The mapping from raw signals to features is not
predefined, but automatically learned from the training dataset. In addition, with
the rising popularity of machine learning DNN (Deep Neural Network)[12] has
made some breakthroughs in many research areas as well. The deep architectures
can represent complex function compactly, which have been demonstrated to
outperform most of the algorithms in many applications.

However, the fully connected DNN models, they do not capture local
dependencies of the time series signals. Thus, some methods utilized the
Convolution Neural Network (CNN) [13, 14, 15] which consists of one or more
pairs of convolution and pooling layers. Since the convolutional layer with a set of
local filters are able to capture the small local parts of the input, and the pooling
layer can preserve the invariant features. This hierarchical organization often
presents competitive results in image processing and speech recognition tasks.

METHOD

In this work, an indoor human action recognition system is developed to
recognize the action of a human from six actions: walking, sitting, lying, standing,
jogging and jumping. The innovation of this work is we use a trained CNN model
to generate the feature vectors then classify these features vector via SVM instead
of CNN.
Data Collection

The UWB tags, part of the Ubisense positioning platform, are used to collecting the spatial location information when the experimenters are performing actions. In the scene of our experiment, three UWB-RFID tags are attached in the right wrist, right waist and right ankle of every experimenter. The Ubisense positioning platform continuously reports location information of each UWB-RFID tags at the frequency of 10Hz and then synchronously returns the data structure in the format of (ID, T, X, Y, Z), where ID denotes the unique tag ID, T denotes the positioning time, and X, Y, Z are the three-dimensional coordinate of the tag location.

Data Preprocessing

The data preprocessing is routinely including two steps: firstly, all the location data sequences have to be segmented into a certain number of samples by one time sliding window which is set as 0.972 second, 1.118 second, 1.404 second and 1.620 second. In our experiment, the best result, 89.75% recognition rate, was obtained when the time sliding window is set as 1.620 second. Secondly, a median filter with window size of 3 is applied to weaken the effect of environment noise.

Feature Extraction

Ideally, features that input to the system for learning should be independent of the location. As a result, three kinds of features: motion features, frequency features and statistic features are considered in this work.

Firstly, the motion features consist of two parts: the first part includes speed, acceleration, displacement distance, and height of one UWB-RFID tag. The second part involves the relative speed, relative acceleration, and relative displacement distance of any two UWB-RFID tags in one data sequence segment. Secondly, the frequency features was obtained by performing the fast Fourier transform method on the speed, acceleration and the height. Finally, the statistic feature was calculated by the standard deviation and mean value of each tag’s displacement distance and the relative displacement distance of two tags for each data sequence segment motion feature. Consequently, for the 0.972-second setting, the dimension of the feature is 255. The dimension increases to 303 over the 1.118-second setting. For the 1.404-second setting and the 1.620-second setting, the dimensions are 339 and 375, respectively.

CNN-SVM Model

As shown in Figure 1, our model is composed by concatenating two individual components, the pre-trained CNN [16] feature extraction part and the SVM part.

We compare various feature choice among our pre-trained CNN models, and the construction of this CNN model is shown in Figure 2. This CNN model was
pre-trained by the same training dataset as we used in our model. As is shown in the figure, the pre-trained model consists of two pairs of convolutional layer with pooling layer and the kernel size over each convolutional layer is 3*3 which is small enough to capture the more distinguish features over the time series signal. The more details about the pre-trained CNN are described in [16]. In our paper, we try to collect convolutional layer features from both of the convolutional layers for comparisons.

Figure 1. The pipeline of our model, which is a concatenation of low-level feature extraction of pre-trained CNN model and SVM classifier learned with respect to diverse task.

Figure 2. The structure of our trained CNN model.

RESULTS

Dataset

We collect data from ten different experimenters in our lab. Every
experimenter is required to repeat one single action within 2 minutes. Since the time sliding windows are set into 4 scales, the training dataset and the testing data set are divided into 4 groups. One group is for one time sliding window, and for each time sliding window, the training data and the testing data was randomly selected, the details are listed in TABLE I and TABLE II.

<table>
<thead>
<tr>
<th>Time sliding window(s)</th>
<th>Group</th>
<th>Walking</th>
<th>Sitting</th>
<th>Lying</th>
<th>Standing</th>
<th>Jumping</th>
<th>Jogging</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.972</td>
<td>1</td>
<td>1041</td>
<td>1102</td>
<td>1206</td>
<td>1133</td>
<td>1084</td>
<td>1070</td>
<td>6636</td>
</tr>
<tr>
<td>1.118</td>
<td>2</td>
<td>853</td>
<td>904</td>
<td>984</td>
<td>924</td>
<td>906</td>
<td>869</td>
<td>5440</td>
</tr>
<tr>
<td>1.404</td>
<td>3</td>
<td>717</td>
<td>753</td>
<td>843</td>
<td>795</td>
<td>747</td>
<td>732</td>
<td>4587</td>
</tr>
<tr>
<td>1.620</td>
<td>4</td>
<td>649</td>
<td>668</td>
<td>744</td>
<td>687</td>
<td>664</td>
<td>636</td>
<td>4048</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Time sliding window(s)</th>
<th>Group</th>
<th>Walking</th>
<th>Sitting</th>
<th>Lying</th>
<th>Standing</th>
<th>Jumping</th>
<th>Jogging</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.972</td>
<td>1</td>
<td>86</td>
<td>67</td>
<td>90</td>
<td>87</td>
<td>90</td>
<td>80</td>
<td>500</td>
</tr>
<tr>
<td>1.118</td>
<td>2</td>
<td>86</td>
<td>70</td>
<td>93</td>
<td>91</td>
<td>70</td>
<td>90</td>
<td>500</td>
</tr>
<tr>
<td>1.404</td>
<td>3</td>
<td>86</td>
<td>80</td>
<td>82</td>
<td>75</td>
<td>88</td>
<td>89</td>
<td>500</td>
</tr>
<tr>
<td>1.620</td>
<td>4</td>
<td>54</td>
<td>60</td>
<td>89</td>
<td>75</td>
<td>94</td>
<td>81</td>
<td>453</td>
</tr>
</tbody>
</table>

**FEATURE ANALYSIS**

In this section we want to analyze the correlation between the convolutional layer features and the performance. We use a straightforward way to determine how much significance each convolutional layer feature accounts for the SVM model. The TABLE III presents the final accuracy of each convolutional layer feature over the Data group 1 of testing data set. For we just intend to verify the relative performance of the convolutional features over each layer on the pre-trained CNN model, we just run the experiments on the Data group 1 of testing and training dataset.

<table>
<thead>
<tr>
<th>CNN net</th>
<th>Convolutional feature layer</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our trained CNN</td>
<td>Conv1</td>
<td>70%</td>
</tr>
<tr>
<td>Our trained CNN</td>
<td>Conv2</td>
<td>87%</td>
</tr>
</tbody>
</table>

See from TABLE III, we get an interesting observation. Over our pre-trained CNN model, the deep convolutional layer feature would have a stronger representation than the sallow convolutional layer feature. Thus, the accuracy over
the Conv2 is much better than that over Con1. Thus, we choose the Conv2 over our trained CNN model as one of the key component of our final model while the other important component is SVM.

**Experiment Results**

Finally, to validate the efficiency of the proposed method, we conduct experiments over our model as shown in Figure 1 and all the experiments are conducted on our final model. We also compare the results with the other traditional methods: RBM, DNN and the stand-alone CNN model. The testing results are listed in **TABLE IV**. As we can see, the new approach outperforms the rest of the models, and the best result was obtained on group-4. Especially, our model is better than the stand-alone CNN model. In my view, the main reason is as below: firstly, the top fully connected layer in stand-alone CNN model would compromise the distinguish ability of the feature generated by the convolutional layer in our experiments while our model just did not take account into the effect caused by the top fully connected layer. Secondly, it seems that the SVM classifier is more appropriate than the CNN classifier for our topic. The further reason will be discussed in future work.

**TABLE IV. TESTING ACCURACY OVER EACH CONVOLUTIONAL LAYER FEATURE.**

<table>
<thead>
<tr>
<th>Group</th>
<th>input dimension</th>
<th>RBM</th>
<th>DNN</th>
<th>CNN accuracy(%)</th>
<th>Our accuracy(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16*16</td>
<td>79.4</td>
<td>80.6</td>
<td>84.6</td>
<td>86.9</td>
</tr>
<tr>
<td>2</td>
<td>18*18</td>
<td>81.3</td>
<td>82.1</td>
<td>85.2</td>
<td>87.2</td>
</tr>
<tr>
<td>3</td>
<td>19*19</td>
<td>78.4</td>
<td>80.6</td>
<td>82.6</td>
<td>85.7</td>
</tr>
<tr>
<td>4</td>
<td>20*20</td>
<td>82.4</td>
<td>82.4</td>
<td>87.6</td>
<td>89.75</td>
</tr>
</tbody>
</table>

**TABLE V. THE CONFUSION MATRIX OF THE TEST RESULT OF GROUP-4 ON OUR MODEL.**

<table>
<thead>
<tr>
<th>Actual Class</th>
<th>Predict Class</th>
<th>Running</th>
<th>Jumping</th>
<th>Standing</th>
<th>Sitting</th>
<th>Jogging</th>
<th>Lying</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Running</td>
<td>66</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>9</td>
<td>4</td>
<td></td>
<td>82.7</td>
</tr>
<tr>
<td>Jumping</td>
<td>2</td>
<td>60</td>
<td>4</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td>89.5</td>
</tr>
<tr>
<td>Standing</td>
<td>1</td>
<td>3</td>
<td>70</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td>93.3</td>
</tr>
<tr>
<td>Sitting</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>58</td>
<td>0</td>
<td>0</td>
<td></td>
<td>96.7</td>
</tr>
<tr>
<td>Jogging</td>
<td>9</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>41</td>
<td>2</td>
<td></td>
<td>75.9</td>
</tr>
<tr>
<td>Lying</td>
<td>7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>77</td>
<td></td>
<td>86.5</td>
</tr>
</tbody>
</table>

The details of the test result over group-4 are shown in **TABLE V**. The table illustrates that the action of standing and sitting have the best accuracy while the walking has the worst accuracy. The accuracy is different from the data groups, because each data group is divided by different time sliding window, and each time sliding window denotes one complete action. In addition, the intensity and duration
of every action are different. Our experiment shows that the best result was achieved when the time sliding window is set as 1.62 second. It demonstrates the feasibility of our model.

CONCLUSIONS

In this paper, we presented the system for recognizing six ordinary human actions by using spatial information obtained from the Ubisense positioning system. This system is based on the CNN model and SVM that extracts the local dependency and scale invariant features of the raw data, and the experimental results confirmed the efficiency of the proposed system.

For future work, experiments with larger dataset are needed to further study the robustness of our model. Further improvements may be achieved by using unsupervised pre-training and then extracting the features over them.

ACKNOWLEDGEMENT

This research was supported by the National Natural Science Foundation of China under Grant 61461039 and 2017 innovation program in Inner Mongolia autonomous region under Grant 10000-16010109.

REFERENCES