Hindawi
Journal of Sensors
Volume 2018, Article ID 8580959, 10 pages
https://doi.org/10.1155/2018/8580959

Research Article

Sequential Human Activity Recognition Based on Deep Convolutional Network and Extreme Learning Machine Using Wearable Sensors

Jian Sun,[1,2] **Yongling Fu**,[1] **Shengguang Li**,[2] **Jie He**,[3] **Cheng Xu**,[3] **and Lin Tan**[2]

[1]*School of Mechanical Engineering & Automation, Beihang University, Beijing 100191, China*
[2]*The F.R.I. of Ministry of Public Security, Beijing 100048, China*
[3]*School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China*

Correspondence should be addressed to Shengguang Li; shijsun@163.com

Received 1 January 2018; Revised 29 July 2018; Accepted 6 August 2018; Published 27 September 2018

Academic Editor: Jaime Lloret

Human activity recognition (HAR) problems have traditionally been solved by using engineered features obtained by heuristic methods. These methods ignore the time information of the streaming sensor data and cannot achieve sequential human activity recognition. With the use of traditional statistical learning methods, results could easily plunge into the local minimum other than the global optimal and also face the problem of low efficiency. Therefore, we propose a hybrid deep framework based on convolution operations, LSTM recurrent units, and ELM classifier; the advantages are as follows: (1) does not require expert knowledge in extracting features; (2) models temporal dynamics of features; and (3) is more suitable to classify the extracted features and shortens the runtime. All of these unique advantages make it superior to other HAR algorithms. We evaluate our framework on OPPORTUNITY dataset which has been used in OPPORTUNITY challenge. Results show that our proposed method outperforms deep nonrecurrent networks by 6%, outperforming the previous reported best result by 8%. When compared with neural network using BP algorithm, testing time reduced by 38%.

1. Introduction

Human activity recognition (HAR) is a new technology that can recognize human activities or gestures through computer system. Identified signals can be obtained from different types of detectors, such as audio sensors, image sensors, barometers, and accelerometers. With the rapid development of human-computer interaction (HCI) and wireless body area networks (WBANs), more and more technologies and methods have been applied to the sensor-based human activity recognition. Meanwhile, the growing maturity of ubiquitous computing [1] and machine learning algorithms has made human activity recognition widely used in athletic competition [2], medical care [3], smart home [4], and health care for the old people [5].

There are two methods of human activity recognition: human activity recognition based on visual images [6, 7] and based on wearable sensors [8]. Human motion analysis in computer vision involves object detection, tracking, and human motion recognition [6]. Computer vision-based human activity recognition method has many limitations. For example, the difficulty of motion detection will be greatly improved under unconstrained conditions, occlusion of the object, and video data acquisition problems for a long time. In addition, the camera needs to be deployed in advance, which cannot be used in some special scenarios, such as emergency rescue. Compared with computer vision, it is more advantageous to obtain signals from wearable sensors than video cameras, due to the following reasons: (1) wearable sensors alleviate the limitations of environmental constraints

and fixed scenes that cameras often suffer from [9, 10]; (2) wearable sensors can better protect the privacy of users, as they can acquire signals for a specific target; and (3) multiple sensors can be deployed more accurately and efficiently on the body for signal acquisition.

In this paper, we study activity recognition based on wearable sensors. This work is motivated by requirements of activity recognition: decreasing dependence on engineered features to address increasingly complex recognition problems, improving recognition accuracy, and improving recognition efficiency. Human activity recognition is challenging due to the large variability of the given action. In order to obtain high accuracy, a large number of data are required. For example, the OPPORTUNITY Activity Recognition Challenge that was organized in 2011, which aims at recognizing activities and gestures in a complex home environment, showed that recognition accuracy of 17 gestures could not exceed 88% [11]. Therefore, addressing the recognition problem in complex scenes will require further improving recognition performance to face a wider set of activities.

Statistical learning methods have been widely used to solve activity recognition problems [12]. Gupta and Dallas used a Naïve Bayes (NB) and a K-nearest neighbor (KNN) classifier to recognize seven motions, such as walking, running, and jumping [13]. However, they relied on handcrafted features and could not find discriminative features to accurately distinguish different activities. The feature extraction methods such as symbolic representation [14], statistics of raw data [15], and transform coding [16] are widely applied in human activity recognition, but they are heuristic and require expert knowledge to design features.

Some methods can automatically extract features without requiring expert knowledge such as convolutional neural network (CNN). Yang et al. [17] used deep convolutional neural networks to automatically learn features from the original inputs. Through the deep structure, the learning features are considered as higher-level abstract representations of low-level raw time series signals. However, this structure ignored the temporal dependencies on the features and was not suitable to recognize real-time sensor signals. Applying the time dependence to the features obtained from the original sensors is a key factor for the success of sequential human activity recognition.

HAR also faces many challenges, such as large variability of a given action, similarity between classes, time consumption, and the high proportion of Null class. All of these challenges have led researchers to develop representation methods of systematic features and efficient recognition methods to effectively solve these problems. Ordóñez and Roggen [18] proposed deep convolutional network with utilization of CNN and LSTM. This paper took advantage of LSTM to solve sequential human activity recognition problem and achieved a good precision. But the complex network framework suffered from low efficiency and can hardly meet real-time requirements in practice applications.

For the above reasons, we creatively integrate extreme learning machine (ELM) [19] into deep convolutional network. Different from [18], we adopt ELM to improve the real-time performance of our framework. In ELM, the parameters of the hidden-layer nodes were chosen randomly and the output weights were solved by the least squares method. It was training-efficient and could be having a very good classification performance.

The contributions of this paper are as follows:

(i) We propose a hybrid deep structure called CNN-LSTM-ELM, which solves the problem of sequential activity recognition

(ii) The framework is composed of convolutional layers, LSTM recurrent layers, and ELM classifier, which can automatically learn feature representations and model the temporal dependencies between features, and improved the real-time capability with ELM

(iii) We show that the proposed deep framework using the ELM classifier is superior to the one using fully connected layer in running time, namely, with high efficiency

The rest of this paper are organized as follows. In Section 2, we provide a primer on the relevant background in deep learning for human activity recognition. A detailed description is presented in Section 3, illustrating the structure of our proposed CnvLSTM-ELM model. Experiments are demonstrated in Section 4, and the results show the superiorities of our proposed model. Then, conclusions come in Section 5.

2. Related Work

Continuous human action recognition is a challenging issue in machine learning with some difficulties to determine the parameters and sizing required because it highly depended upon some issues like feature selection in continuously training data streaming and the typical of classifier methods and we have less prior knowledge to determine the final size of training data and the size of machine learning architectures. In human action data features, we have to deal with a variance problem as explained in the survey paper [6, 7, 15]. In the typical of classifier methods, it makes uneasy requirements and conditions for any machine learning methods [16], such as neural networks [17], dynamic Bayesian networks [4], extreme learning machine [19], and deep learning [12] that may not give good generalization accuracy and processing speed for all human action recognition cases. Here we, summarize that state of the art for e-health monitoring and other proposals for human activity recognition.

2.1. Convolutional Neural Network. The input of neural network is generally the original signal; however, applying features extracted from the original signal to the neural network tends to improve performance. Extracting more useful features from the original signal requires sufficient expert knowledge, which inevitably limits a systematic exploration of the feature space [20]. Convolutional neural networks have been proposed to address this problem. Generally, CNN can be considered to comprise two parts. The first part is the hierarchical feature extractor, which contains

convolutional layers and pooling layers. The input of each layer is the output of its previous layer. As a result, the original signal is mapped into feature vectors. The second part is a fully connected layer, and the feature vectors are classified by the fully connected layer.

The most widely used deep learning approach in the ubiquitous computing field in general and in human activity recognition using wearables in particular employs CNNs. CNNs typically contain multiple hidden layers that implement convolutional filters that extract abstract representations of input data. Combined with pooling and/or subsampling layers and fully connected special layers, CNNs are able to learn hierarchical data representations and classifiers that lead to extremely effective analysis systems. A multitude of applications are based on CNNs, including but not limited to [21–24].

Recently, sophisticated model optimization techniques have been introduced that actually allow for the implementation of deep CNNs in resource-constrained scenarios, most prominently for real-time sensor data analysis on smartphones and even smart watches [25].

2.2. Long Short-Term Memory (LSTM) Network. The de facto standard workflow for activity recognition in ubiquitous and wearable computing [26] treats individual frames of sensor data as statistically independent, that is, isolated portions of data are converted into feature vectors that are then presented to a classifier without further temporal context. However, ignoring the temporal context beyond frame boundaries during modelling may limit the recognition performance for more challenging tasks. Instead, approaches that specifically incorporate temporal dependencies of sensor data streams seem more appropriate for human activity recognition. In response to this, recurrent deep learning methods have now gained popularity in the field. Most prominent models based on so-called LSTM units [27] have been used very successfully. In [18], deep recurrent neural networks have been used for activity recognition on the OPPORTUNITY benchmark dataset. The LSTM model was combined with a number of preceding CNN layers in a deep network that learned rich, abstract sensor representations and very effectively could cope with the nontrivial recognition task. Through large-scale experimentation in [28], appropriate training procedures have been analyzed for a number of deep learning approaches to HAR including deep LSTM networks. In all of the previous work, single LSTM models have been used and standard training procedures have been employed for parameter estimation. The majority of existing methods [18, 27, 28] is based on (variants of) sliding-window procedures for frame extraction. The focus of this paper is on capturing diversity of the data during training and to incorporate diverse models into ensemble classifiers.

2.3. Extreme Learning Machine (ELM). The sequential concept in neural networks was mainly introduced by Yingwei et al. [29] for function approximation and time series prediction using a minimal radial basis function neural network (RBFNN), named minimal resource allocation neural network (MRAN). MRAN with sequential learning algorithms then become popular for feedforward networks with RBF nodes. Not like MRAN, OS-ELM (online sequential ELM) can handle both additive and RBF nodes in a unified framework and can learn the training data not only one-by-one but also chunk-by-chunk (with fixed or varying length). OS-ELM originates from the batch learning extreme learning machine (ELM) that was developed by Huang et al. [19].

OS-ELM problem cannot automatically determine the optimal number of hidden nodes in a hidden layer. In most of the cases, searching for the suitable number of hidden nodes relies on the trial and error method. There are two disadvantages: (1) it wastes lots of time and (2) it cannot always guarantee to get the optimal solution, because too small a network cannot learn the problem well and too large a network may lead to overfitting and poor generalization performance. According to Jun and Er [30], OSELM as well as the ELM requires much more hidden nodes compared with MRAN that would increase the processing time in real-time applications. The EOS-ELM algorithm adapts the node location, adjustment, and pruning method of the minimal resource allocation neural network (MRAN), so that the number of hidden nodes used in the OS-ELM can be modified. As a further improvement by Lan et al. [31], CEOS-ELM was proposed with searching capability for the optimal network architecture, and it can handle both additive and radial basis function (RBF) hidden nodes. The optimal number of hidden nodes can be obtained automatically in the sequential training process.

3. The Proposed Architecture

Current popular algorithms mainly use CNN for feature extraction and classification. But these methods are often not ideal in dealing with time series problems. We propose a hybrid activity recognition architecture; the flow chart of the architecture is depicted in Figure 1. We introduce the LSTM units to model temporal dependencies on the features extracted by CNN, which can be used to deal with time series problems. In addition, we use the ELM classifier with better generalization performance to classify the features that contain time information, which can improve the classification performance and shorten the running time. The structure combines convolutional layers, LSTM layers, and ELM classifier.

3.1. Feature Extraction Using CNN-LSTM. The convolutional neural network has great potential to identify various salient features in human activity recognition signals. In particular, the processing unit of the lower layers obtains local features of the signal (to represent the nature of each elementary motion in human activity). Higher-layer processing units represent the data in a more abstract way (to characterize the saliency of several basic motion combinations). Note that, as described below, each layer can have multiple convolution operations and pooling operations (specified by different parameters), so the CNN takes the features contained in the data into consideration from different aspects, and the

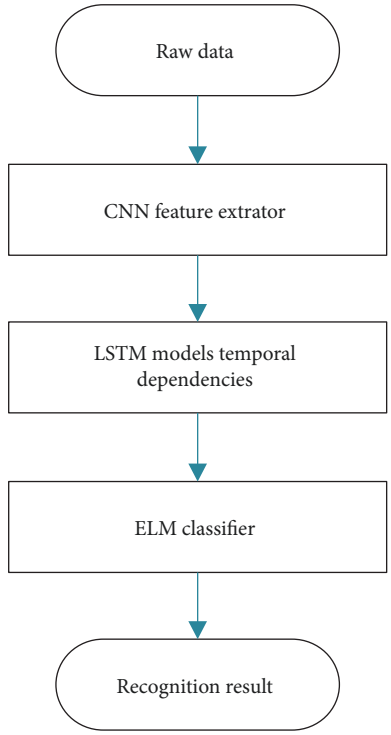

Figure 1: Hybrid activity recognition architecture: the structure combines convolutional layers, LSTM layers, and ELM classifier.

learned features are more comprehensive than those of artificial extraction.

When these operations with the same parameters are applied to sequence signals (or their mappings) at different time periods, the property of translation invariance is obtained. Therefore, what matters is the salient mode of the signal, not its position or scale. However, in human activity recognition, we are faced with multiple time series signal channels, and traditional CNN cannot be used directly. Our challenges include that: (1) the processing unit in the CNN needs to be applied along the time dimension, (2) sharing or unifying the units in the CNN between multiple sensors. Next, we will define the convolution operation and the pooling operation along the time dimension, and then present the entire architecture of the CNN used in human activity recognition.

We use a sliding window strategy to decompose the time series signal into a collection of short signals. Specifically, an example used by CNN is a two-dimensional matrix containing r original samples (each sample contains D attributes). Here, r is selected as the sampling rate (for example, 30 is used in the experiment), and the sliding step size of the window is selected as $r/2$. We can choose a smaller step size to increase the number of instances, but it may result in higher computational cost. For training data, the actual label of the matrix instance is determined by the most frequently occurring labels of the r original records. For the jth feature map in the ith layer of the CNN, it is also a matrix. For convenience, the value of the Xth row of the sensor D is expressed as $v_{ij}^{x,d}$.

The convolutional neural network assumes that the inputs and outputs of the model are independent from each other. However, the collected data is time dependent; thus, some time information must be included in the input data. Long short-term memory (LSTM) network has been proposed to solve this problem. The LSTM is an extension of the recurrent neural network that uses memory cells instead of loop units to store and output information. In this study, we used LSTM to model the features extracted by CNN and output the feature vectors containing time relationships.

The feature extraction network proposed in this paper combines four convolutional layers, two LSTM layers, and a fully connected (FC) layer, as shown in Figure 2. The convolution layer acts as a feature extractor and provides an abstract representation of the original data in the form of a feature graph. The LSTM layer builds the time dynamics for the feature graph. In order to illustrate the advantages of the proposed ConvLSTM-FC network, we compare it with the nonrecurrent deep convolution neural network (namely, baseline CNN). They all use four convolutional layers. The input is processed in a hierarchical manner, with each layer handling the input and then passing it to the next layer. Under these two frameworks, the number of convolution kernels in the convolution layer is the same as that in the dense layer. The only difference is that the ConvLSTM-FC uses a loop unit, while the baseline CNN is nonrecurrent and fully connected.

The input of the network is sequential data. These sequences are extracted from the sensor data by using a fixed-length sliding window. The convolution layer performs convolution operations on these data sequences and represents the data in forms of feature graphs. The LSTM layer constructs the time correlation of the feature graph and extracts the features containing time information. When the whole ConvLSTM-FC training is completed, we only reserve the convolution layer and the LSTM layer.

3.2. Classification Using ELM. As we all know, the more the discriminative features are and the more the powerful classifier is, the higher the recognition rate will be. The full-connected layer is equivalent to a general single hidden-layer feedforward neural network classifier, which is trained by backpropagation (BP). On one hand, the BP algorithm can make the weight converge to a certain value, but it is not guaranteed to be the global minimum of the error surface [32]. On the other hand, the network may be overtrained and obtain nonideal generalization performance when BP learning is used. In other words, the full-connected layer is not suitable for classifying the discriminative deep convolutional features.

Given a training set of instance-label pairs (X_i, y_i) (X_i stands for features, and y_i is labels), where $X_i = [x_{i1}, x_{i2}, \cdots, x_{\text{in}}]^T \in R^n$ and $y_i = [y_{i1}, y_{i2}, \cdots, y_{\text{in}}] \in R^m$, for a single hidden-layer neural network with M hidden nodes, the output of the network (denoted as o_j) is calculated by the following equation:

$$\sum_{i=1}^{M} \beta_i f\left(W_i \cdot X_j + b_i\right) = o_j, \quad j = 1, \ldots, N, \tag{1}$$

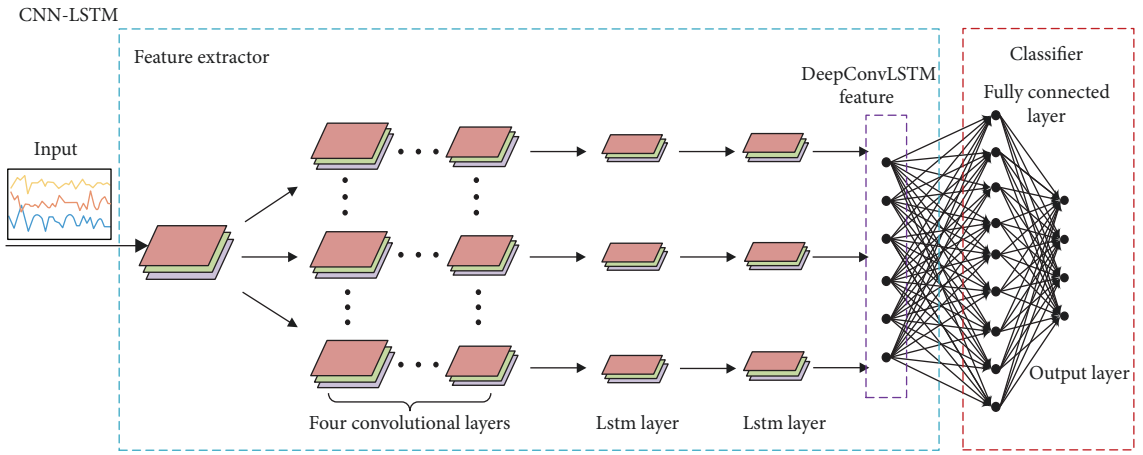

Figure 2: Our proposed CnvLSTM-FC model, combining convolutional layers, LSTM layers, and ELM classifier.

where $f(x)$ is an activation function, $W_i = [w_{i1}, w_{i2}, \cdots, w_{\text{in}}]$ is the input weight matrix, β_i is the output weight vector, and b_i is the bias of the ith hidden node.

As shown in Figure 3, the learning goal of ELM is to minimize the output error (see (2)), that is, there exist β_i, W_i, and b_i that make (3) hold true.

$$\sum_{J=1}^{M} \left\| o_j - y_j \right\| = 0, \tag{2}$$

$$\sum_{i=1}^{M} \beta_i f\left(W_i \cdot X_j + b_i\right) = y_j, \quad j = 1, \ldots, N. \tag{3}$$

Equation (4) is the compact version of (3), where H is the hidden-layer output matrix (see (4)), β is the output weight vector, and Y is the desired output.

$$H\beta = Y,$$

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_M^T \end{bmatrix}_{M\times m}, \quad Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix}_{N\times m}, \tag{4}$$

$$H_{w,b,X} = \begin{bmatrix} f(W_1 \cdot X_1 + b_1) & \cdots & f(W_M \cdot X_1 + b_M) \\ \vdots & \cdots & \vdots \\ f(W_1 \cdot X_N + b_1) & \cdots & f(W_M \cdot X_N + b_M) \end{bmatrix}_{N\times M}. \tag{5}$$

The traditional gradient-based learning algorithm requires that all parameters be adjusted during the iteration. However, the output weight of ELM is solved in a noniterative way and there is no dependency between the input weights and the output weights. A noniterative solution of

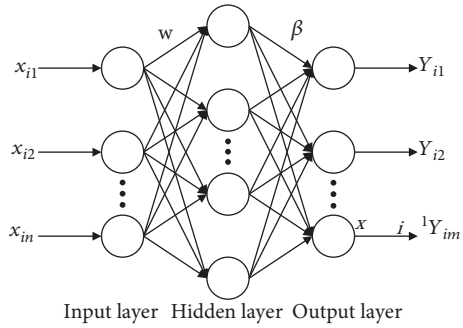

Figure 3: ELM classifier architecture.

ELM provides a speedup of 5 orders of magnitude compared to multilayer perceptron (MLP) [33] or 6 orders of magnitude compared to support vector machines (SVM) [34].

Using ELM to classify deep convolutional features can get a good recognition result and has a good generalization ability. Unlike BP, the weights between the input layer and the hidden layer of ELM are randomly set, and the output weights are solved through the method of least squares [19]. Therefore, the training speed of ELM is very fast.

3.3. Data Preprocessing. In this paper, we use the OPPORTUNITY dataset to evaluate the ConvLSTM-ELM model and compare the performance with baseline CNN and other literatures using some other machine learning algorithms used in the dataset experiment. The baseline CNN provides a performance reference for deep networks.

In this paper, we have used the OPPORTUNITY dataset to train and test our model. OPPORTUNITY activity recognition dataset is composed of a set of complex human natural activities collected in an environment where rich sensors are installed [35]. We only consider the on-body sensors, including inertial measurement units and 3-axis accelerometers. The wearing position of the sensors is shown in Figure 4. Each sensor channel is treated as an individual channel, a total of 113 channels. The sampling frequency of these sensors is 30 Hz. OPPORTUNITY dataset contains several

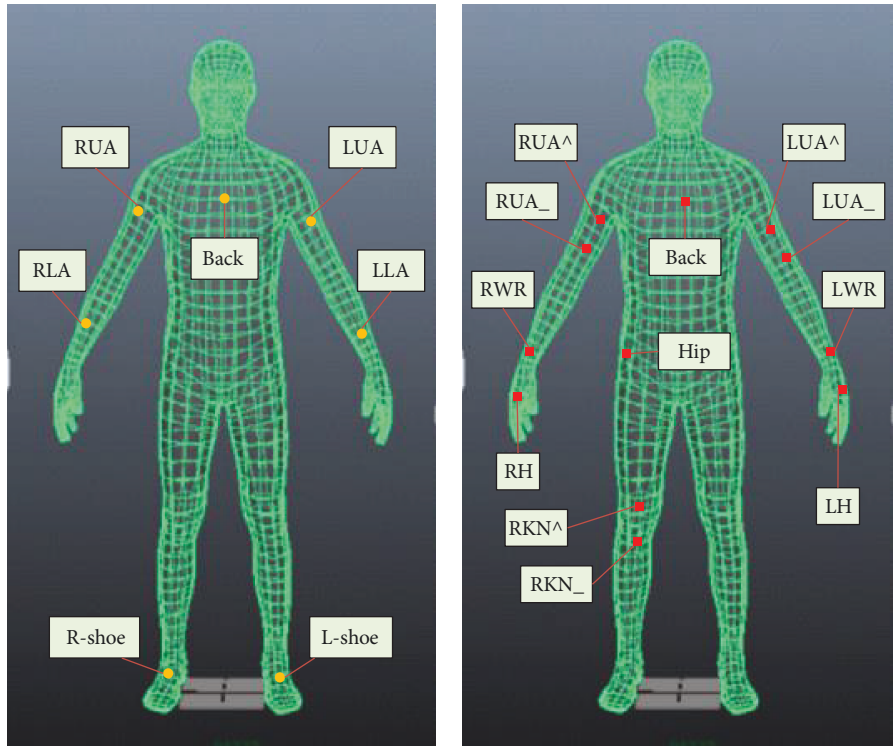

Figure 4: Position of on-body sensors used in the OPPORTUNITY dataset ((a) IMU sensors; (b) 3-axis accelerometers) [11].

gestures and postures, and we mainly realize the recognition of gestures either including or ignoring the Null class. This is an 18-class classification problem; the gestures in the dataset are summarized in Table 1.

We used 0 to fill in missing values of the sensor data, and each sensor channel was normalized to interval $[0, 1]$. We used a fixed-length sliding window to segment data; each segmentation of data was called a sequence. The length of the window is 500 ms and the step size is 250 ms. The model is trained with a learning rate of 0.001 and a decay factor of 0.9. CNN-LSTM works as the feature extractor. The whole setting of the CNN-LSTM architecture will be detailed later in Section 4, which is shown in Table 2. When the training is completed, parameters of convolutional layers and LSTM layers are reserved, and the full-connected layer is removed. Then, ELM is fed with features extracted by CNN-LSTM and trained as classifier.

Corresponding to the serialized window data is the pose observed during the time window. Given a sliding window with a length of T, we choose the label of the $t = T$ moment as the class of the sequence; in other words, we choose the label of the last data in the sequence as the label of the sequence, as shown in Figure 5.

3.4. Model Implementation. We build and train the neural network in Theano using Lasagne [36]. The model runs on a 1536 core, 1050 MHz clock speed, and 8 GB RAM GPU. The detailed software and hardware parameters are shown in Tables 3 and 4, respectively.

Table 1: Class labels for the mode of gesture recognition.

Gestures	
Open dishwasher	Close dishwasher
Open fridge	Close fridge
Open drawer 1	Close drawer 1
Open drawer 2	Close drawer 2
Open drawer 3	Close drawer 3
Open door 1	Close door 1
Open door 2	Close door 2
Drink from cup	Clean table
Toggle switch	Null

Table 2: CNN-LSTM architecture parameters.

Layer	Type	Number of neurons	Kernel size	Stride
1	Input	24×113	—	—
2	Convolutional	20×113	5×1	1
3	Convolutional	16×113	5×1	1
4	Convolutional	12×113	5×1	1
5	Convolutional	8×113	5×1	1
6-7	LSTM	128	—	—
8	Fully connected	n	—	—

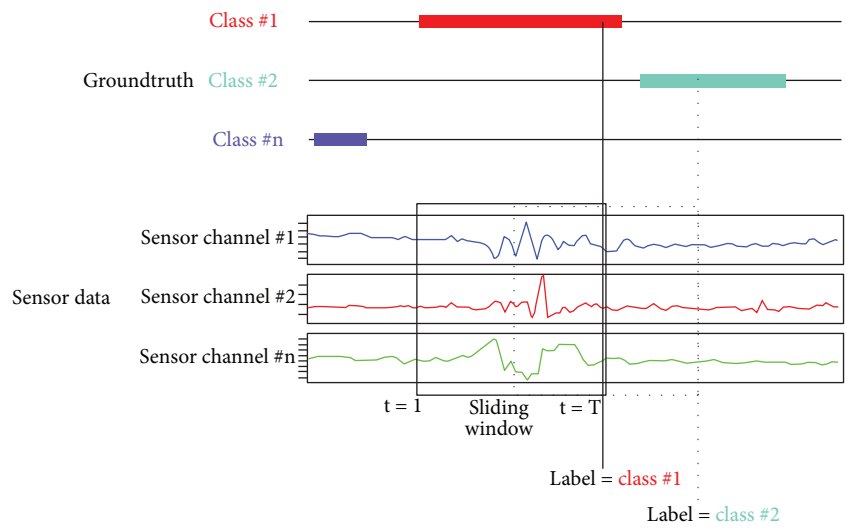

Figure 5: Sequence tags after splitting data with sliding windows.

Table 3: Software parameter configuration.

Operating system	Ubuntu 14.04
Programming language	Python 2.7.11
IDE	Spyder 3.1.4

Table 4: Hardware parameter configuration.

CPU	Intel Xeon E3-1505M v5
CPU dominant frequency	2.8 GHz
Core/thread count	4 cores/8 threads
RAM volume	64 GB
Graphics card	NVIDIA Quadro M5000M
Graphic memory	8 GB
Clock frequency	1050 MHz
CUDA cores	1536

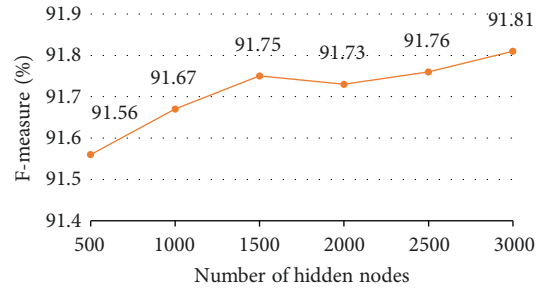

Figure 6: CNN-LSTM-ELM performance with different numbers of hidden nodes.

Table 5: The comparison of runtime.

	Training time (s)	Testing time (s)
CNN-LSTM [18]	12154.5	8.025
CNN-LSTM-ELM	11456.124	4.901

4. Result and Analysis

In this section, we test our model on the public OPPORTUNITY dataset. We will give a brief introduction to the dataset used in the experiment and the indicators to evaluate the performance of proposed models. Discussion on the experimental results is detailed. We demonstrated the performance of the proposed method and evaluated the impact of some key parameters on the methods.

4.1. Performance Measure. Human activity datasets collected in natural scenes are often imbalanced between classes [37]. Some classes may contain a large number of samples while other classes have only a few samples. The gestures of OPPORTUNITY dataset are extremely imbalanced; the Null class accounts for more than 70% of all the data. The classifier predicts the classification accuracy of each class; the Null class can achieve very high accuracy. The overall classification accuracy is not an appropriate index for performance evaluation. F-measure (F_1) considers the correct classification of each class as equally important. It takes into account both the precision and the recall of each class to compute the score and can evaluate the model better than the precision. Precision is defined as $P - (\mathrm{TP}/(\mathrm{TP} + \mathrm{FP}))$, and recall corresponds to $R - (\mathrm{TP}/(\mathrm{TP} + \mathrm{FN}))$, where TP and FP are the number of true and false positives, respectively, and FN corresponds to the number of false negatives. Class imbalance is countered by weighting classes according to their sample proportion:

$$F_1 = \sum_i 2^* w_i \frac{\mathrm{precision}_i \cdot \mathrm{recall}_i}{\mathrm{precision}_i + \mathrm{recall}_i}, \quad (6)$$

where $w_i = n_i/N$ is the proportion of samples of the ith class, with n_i being the number of samples of the ith class and N being the total number of samples.

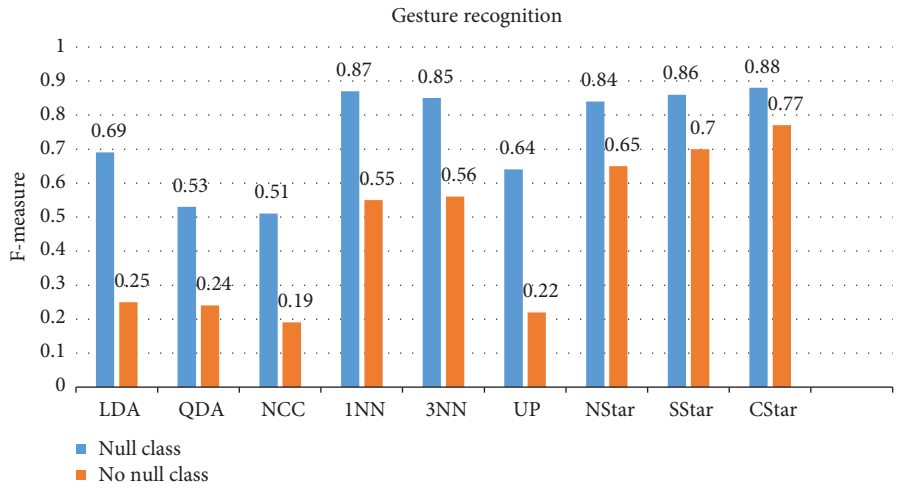

Figure 7: F1 score comparison of the proposed architecture and the OPPORTUNITY challenge models, either including or ignoring the Null class.

4.2. Parameter Evaluation. We describe the impact of key parameters of the architecture. We evaluate the influence of ELM hidden-layer nodes. For ELM, it is fed with discriminative features and a different number of hidden nodes are chosen, and the corresponding recognition performance is recorded in Figure 6.

Apparently, the recognition performance increases as the number of hidden nodes increases. When there are 500 hidden-layer nodes, the recognition performance is up to 91.56%, which is already better than that of many methods. It reaches 91.81% with 3000 hidden-layer nodes. The results show that the performance may be further improved if the number of hidden nodes is increased.

4.3. Runtime Analysis of CNN-LSTM-ELM. The runtime analysis of the CNN-LSTM-ELM is done on OPPORTUNITY dataset. It has 46495 training samples and 9894 testing samples, and each sample is a sequence. The model training and testing are run on a GPU with 1536 cores, 1050 MHz clock speed, and 8 GB RAM.

Because the traditional feedforward neural network adopts the iterative algorithm of gradient descent to update the weight parameters, it has obvious defects: the learning speed is slow, so the time cost is unacceptable; the learning rate is difficult to be determined and the network is easy to fall into the local minimum; it would probably be overtrained; and its generalization performance is not guaranteed to be optimal. These defects become a bottleneck for the wide application of feedforward neural network using iterative algorithms. In order to solve these problems, ELM is utilized instead of the traditional algorithm with gradient descent. The algorithm can calculate the output weights of the learning network by one step. Compared with the iterative algorithm, the ELM greatly improves generalization ability and the learning speed of the network. Therefore, for our network compared with the traditional network, the runtime mainly depends on the use of ELM classifier.

The CNN-LSTM with the fully connected layer is used to compare the runtime with CNN-LSTM-LSTM; training time and testing time are shown in Table 5.

As can be seen from Table 4, when compared to the CNN-LSTM, training time is reduced by 5% and testing time is reduced by 38%. The length of the sliding window is 500 ms and the testing samples are composed of 9894 windows, so the testing samples contain the original data of 4947 s. CNN-LSTM-ELM recognizes the test samples and only takes 4.901 s, indicating that our framework can recognize real-time sensor data.

4.4. Performance Comparison. The classification results of the proposed deep methods on the OPPORTUNITY dataset are shown in Figure 7 and Table 2. We report classification performance either including or ignoring the Null class. Including the Null class may lead to a much higher recognition rates for Null class than rare classes. In other words, some samples belonging to rare classes are often wrongly recognized as the Null class. By providing both results, we can better understand the type of errors caused by the model.

Figure 7 includes the classification performance of past published classification techniques implemented on OPPORTUNITY dataset. All of these methods are based on sliding window and only the feature extraction and classifier are different. Compared to the best approach of the OPPORTUNITY challenge, our method improves the performance by 8% on average. There is more than 13% improvement in the gesture recognition task including the Null class when compared to the OPPORTUNITY challenge methods.

From the results in Table 6, CNN-LSTM-ELM has better performance than other deep models either including or ignoring the Null class. CNN-LSTM-ELM improves by 6% compared to CNN used by Yang et al. [17]. The Baseline CNN has the same number of convolutional layers as the CNN-LSTM-ELM, but it uses the traditional classifier based on the gradient descent algorithm. When compared with

Table 6: F_1 score comparison of the proposed architecture and deep architecture, either including or ignoring the Null class.

	Deep architectures	
	Gesture recognition	Gesture recognition (no Null class)
CNN [17]	0.851	
Baseline CNN	0.883	0.783
CNN-LSTM [18]	0.897	0.802
CNN-LSTM-ELM	**0.918**	**0.906**

baseline CNN, CNN-LSTM-ELM improves by 7.5% on average. For the recognition of these similar gestures, CNN-LSTM-ELM can get better classification performance because LSTM cells have the ability to model time dynamics of the data sequences. However, the baseline CNN ignores the temporal dependencies between the data sequences.

5. Conclusions

In this article, we demonstrated the advantages of a deep architecture based on the combination of convolutional layers, LSTM recurrent layers, and ELM classifier for wearable activity recognition. This combined structure is used to study multichannel time series data. It mainly utilizes convolution operations and LSTM cells to capture significant features of sensor signals with different time scales. Then, all the extracted features are classified by the ELM classifier. The main advantages of this framework are as follows: (i) feature extraction is automatically performed without manual intervention; (ii) LSTM cells can capture the temporal dependencies on features extracted by convolution operations; and (iii) ELM classifier has outstanding generalization ability and fast learning speed. This framework outperformed best result of the OPPORTUNITY challenge by 13% on average in an 18-class gesture recognition task. In the experiment, we illustrate that the proposed method is superior to other best methods, so we believe that the proposed method can be used as a powerful tool for human activity recognition problems.

As for future work, we aim to improve sequential learning adaptive capability; thus, the machine learning has capability to continuously learn while at the same time doing verification (learning by doing). Transfer learning approach based on existing models to perform activity recognition on large-scale data may be a potential working direction.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by Projects of International Cooperation and Exchanges NSFC (Key Program) no. 61327807.

References

[1] T. Plötz, N. Y. Hammerla, and P. Olivier, "Feature learning for activity recognition in ubiquitous computing," *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, 2011, pp. 1729–1734, Barcelona, Catalonia, Spain, July 2011.

[2] J. Margarito, R. Helaoui, A. M. Bianchi, F. Sartor, and A. Bonomi, "Userindependent recognition of sports activities from a single wrist-worn accelerometer: a template-matching-based approach," *IEEE transactions on bio-medical engineering*, vol. 63, no. 4, pp. 1–796, 2015.

[3] S. Sendra, L. Parra, J. Lloret, and J. Tomás, "Smart system for children's chronic illness monitoring," *Information Fusion*, vol. 40, pp. 76–86, 2018.

[4] P. C. Roy, S. Giroux, B. Bouchard et al.L. Chen, C. Nugent, J. Biswas et al., "A possibilistic approach for activity recognition in smart homes for cognitive assistance to Alzheimer's patients," in *Activity Recognition in Pervasive Intelligent Environments*, Atlantis Ambient and Pervasive Intelligence, pp. 33–58, Atlantis Press, 2011.

[5] C. Xu, J. He, X. Zhang, C. Wang, and S. Duan, "Detection of freezing of gait using template-matching-based approaches," *Journal of Sensors*, vol. 2017, Article ID 1260734, 8 pages, 2017.

[6] A. Bux, P. Angelov, and Z. Habib, "Vision based human activity recognition: a review," in *Advances in Computational Intelligence Systems*, P. Angelov, A. Gegov, C. Jayne, and Q. Shen, Eds., vol. 513 of Advances in Intelligent Systems and Computing, Springer, Cham, 2017.

[7] S. R. Ke, H. L. U. Thuc, Y. J. Lee, J.-N. Hwang, J.-H. Yoo, and K.-H. Choi, "A review on video-based human activity recognition," *Computers*, vol. 2, no. 2, pp. 88–131, 2013.

[8] C. Xu, J. He, X. Zhang, C. Yao, and P.-H. Tseng, "Geometrical kinematic modeling on human motion using method of multisensor fusion," *Information Fusion*, vol. 41, 2017.

[9] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *CVPR 2011*, pp. 3361–3368, Providence, RI, USA, June 2011.

[10] S. Ji, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, 2013.

[11] R. Chavarriaga, H. Sagha, A. Calatroni et al., "The opportunity challenge: a benchmark database for on-body sensor-based activity recognition," *Pattern Recognition Letters*, vol. 34, no. 15, pp. 2033–2042, 2013.

[12] W. Liu, J. Yang, L. Wang, C. Wu, and R. Zhang, "Movement behavior recognition based on statistical mobility sensing," *Adhoc & Sensor Wireless Networks*, vol. 25, no. 3-4, pp. 323–340, 2015.

[13] P. Gupta and T. Dallas, "Feature selection and activity recognition system using a single triaxial accelerometer," *IEEE transactions on bio-medical engineering*, vol. 61, no. 6, pp. 1780–1786, 2014.

[14] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming

algorithms," in *DMKD '03 Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pp. 2–11, San Diego, CA, USA, June 2003.

[15] C. Xu, J. He, X. Zhang, P. H. Tseng, and S. Duan, "Toward near-ground localization: modeling and applications for TOA ranging error," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 10, pp. 5658–5662, 2017.

[16] A. Rghioui, S. Sendra, J. Lloret, and A. Oumnad, "Internet of things for measuring human activities in ambient assisted living and e-health," *Network Protocols and Algorithms*, vol. 8, no. 3, pp. 15–28, 2016.

[17] M. Kahng, P. Y. Andrews, A. Kalro, and D. H. Chau, "ActiVis: visual exploration of industry-scale deep neural network models," *IEEE Transactions on Visualization and Computer Graphics*, vol. 24, no. 1, pp. 88–97, 2018.

[18] F. J. Ordóñez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[19] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.

[20] C. Xu, J. He, and X. Zhang, "DFSA: a classification capability quantification method for human action recognition," in *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, San Francisco, CA, USA, August 2017.

[21] C. Xu, J. He, X. Zhang et al., "Recurrent transformation of prior knowledge based model for human motion recognition," *Computational Intelligence and Neuroscience*, vol. 2018, Article ID 4160652, 12 pages, 2018.

[22] C. A. Ronao and S. B. Cho, "Deep convolutional neural networks for human activity recognition with smartphone sensors," in *Neural Information Processing*, pp. 46–53, Springer International Publishing, 2015.

[23] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence*, pp. 3995–4001, Buenos Aires, Argentina, July 2015.

[24] N. M. Rad, A. Bizzego, S. M. Kia, G. Jurman, P. Venuti, and C. Furlanello, "Convolutional neural network for stereotypical motor movement detection in autism," 2015, https://arxiv.org/abs/1511.01865.

[25] S. Bhattacharya and N. D. Lane, "Sparsification and separation of deep learning layers for constrained resource inference on wearables," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM - SenSys '16*, pp. 176–189, Stanford, CA, USA, November 2016.

[26] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, pp. 1–33, 2014.

[27] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[28] N. Y. Hammerla, J. M. Fisher, P. Andras, L. Rochester, R. Walker, and T. Plotz, "PD disease state assessment in naturalistic environments using deep learning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 1742–1748, Austin, TX, USA, January 2015.

[29] L. Yingwei, N. Sundararajan, and P. Saratchandran, "A sequential learning scheme for function approximation using minimal radial basis function neural networks," *Neural Computation*, vol. 9, no. 2, pp. 461–478, 1997.

[30] Y. Jun and M.-J. Er, "An enhanced online sequential extreme learning machine algorithm," in *2008 Chinese Control and Decision Conference*, pp. 2902–2907, Yantai, Shandong, China, July 2008.

[31] Y. Lan, Y. C. Soh, and G.-B. Huang, "A constructive enhancement for online sequential extreme learning machine," in *2009 International Joint Conference on Neural Networks*, pp. 1708–1713, Atlanta, GA, USA, June 2009.

[32] B. F. Books and S. Haykin, *Neural Networks a Comprehensive Foundation*, Pearson Education, Singapore, 2010.

[33] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, no. 6, pp. 386–408, 1958.

[34] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[35] D. Roggen, A. Calatroni, M. Rossi et al., "Collecting complex activity datasets in highly rich networked sensor environments," in *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pp. 233–240, Kassel, Germany, June 2010.

[36] S. Dieleman, J. Schlüter, C. Raffel et al., *Lasagne: First Release*, Zenodo, Geneva, Switzerland, 2015.

[37] C. X. Ling and V. S. Sheng, *Class Imbalance Problem*, Springer US, 2011.