IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 29, 2020, accepted March 17, 2020, date of publication March 20, 2020, date of current version March 31, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2982225

LSTM-CNN Architecture for Human Activity Recognition

KUN XIA, JIANGUANG HUANG, AND HANYU WANG
University of Shanghai for Science and Technology, Shanghai 200093, China

Corresponding author: Jianguang Huang (hjg.123@foxmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 512070901, and in part by the Key Project of National Natural Science Foundation of China under Grant 51637001.

ABSTRACT In the past years, traditional pattern recognition methods have made great progress. However, these methods rely heavily on manual feature extraction, which may hinder the generalization model performance. With the increasing popularity and success of deep learning methods, using these techniques to recognize human actions in mobile and wearable computing scenarios has attracted widespread attention. In this paper, a deep neural network that combines convolutional layers with long short-term memory (LSTM) was proposed. This model could extract activity features automatically and classify them with a few model parameters. LSTM is a variant of the recurrent neural network (RNN), which is more suitable for processing temporal sequences. In the proposed architecture, the raw data collected by mobile sensors was fed into a two-layer LSTM followed by convolutional layers. In addition, a global average pooling layer (GAP) was applied to replace the fully connected layer after convolution for reducing model parameters. Moreover, a batch normalization layer (BN) was added after the GAP layer to speed up the convergence, and obvious results were achieved. The model performance was evaluated on three public datasets (UCI, WISDM, and OPPORTUNITY). Finally, the overall accuracy of the model in the UCI-HAR dataset is 95.78%, in the WISDM dataset is 95.85%, and in the OPPORTUNITY dataset is 92.63%. The results show that the proposed model has higher robustness and better activity detection capability than some of the reported results. It can not only adaptively extract activity features, but also has fewer parameters and higher accuracy.

INDEX TERMS Human activity recognition, convolution, long short-term memory, mobile sensors.

I. INTRODUCTION

Human activity recognition (HAR) plays an important role in people's daily lives because it has the ability to learn profound advanced knowledge about human activities from raw sensor data [1]. With the development of human-computer interaction applications, the technology of HAR has become a popular research direction at home and abroad. People could automatically classify the type of human motion and obtain the information that the human body needs to convey by extracting features from daily activities, which in turn provides a basis for other intelligent applications. Hitherto, this technology has been widely used in the fields of home behavior analysis [2], video surveillance [3], gait analysis [4], and gesture recognition [5], etc.

Due to the rapid development of sensor technology and ubiquitous computing technology, sensor-based HAR has become more and more popular, and it is widely used with privacy being well protected. Researchers have explored the role of different types of sensing technology in activity recognition to improve recognition accuracy. According to the manner in which sensors are employed in an environment, the technologies of human activity recognition could be widely divided into two categories: approaches based on fixed sensors and approaches based on mobile sensors [6].

The methods based on fixed sensors mean that the information is obtained from sensors mounted at a fixed position, involving acoustic sensors [7], radars [8], static cameras [9], and other ambient-based sensors. Among them, camera-based methods are the most popular methods, among which background subtraction method, optical flow method and energy-based segmentation method are usually applied

The associate editor coordinating the review of this manuscript and approving it for publication was Yongping Pan.

to extract features [10]–[13]. Representative is an image processing method based on Kinect sensors which could acquire the depth image features of moving targets whereby Jun Liu *et al.* [10] proposed a space-time short-term memory (ST-LSTM) network to recognize activities. Kitani *et al.* [11] presented a sparse optical flow algorithm to acquire the histogram of human motion features and proposed an unsupervised Dirichley hybrid model to classify 11 human activities. Although these activity monitoring methods can provide better recognition accuracy, they are not suitable in many indoor environments, especially where privacy is a concern. Furthermore, the results of vision-based approaches are easily affected by illumination variations, ambient occlusion, and background change. This greatly limits their practical use.

The other methods of activity recognition are to use mobile sensors. In these methods, the information from different kinds of behaviors is usually collected from a set of dedicated body-worn motion sensors, such as accelerometers, gyroscopes, and magnetometers. Acceleration and angular velocity data would change according to human motion. Therefore, they could be used to infer human activities. The miniaturization and flexibility of sensors allow individuals to wear or carry mobile devices embedded with various sensing units. This is different from fixed sensor-based approaches [14]. Moreover, these sensors have the characteristics of low cost, low power consumption, high capacity, miniaturization, and less dependence on surroundings [15]. Therefore, activity recognition based on mobile sensors has received widespread attention because of its portability and high acceptance in daily life. Correspondingly, a large number of researches have been carried out to explore the potential of mobile sensors for activity recognition in a ubiquitous and pervasive way. Margarito *et al.* [16] put accelerometers on the wrist of subjects to collect acceleration data and then used template matching algorithm to classify 8 common sports activities. In [17], a smart life assistant system (SAIL) for the elderly and disabled was proposed. Zhu *et al.* [17] collected the features by the way of multi-sensor fusion strategy and achieved the target of recognizing 13 kinds of daily activities.

The rest of this paper is organized as follows. Section II presents some current sensor-based activity recognition researches that using machine learning methods and deep learning methods. Section III presents the description of three public datasets and data pre-processing for the implemented network. Section IV gives details on the proposed LSTM-CNN architecture. Section V shows the experimental results and compares them with some of the previously reported works. Moreover, the impact of network structure and hyper-parameters on model performance is discussed. Finally, the last section summarizes this research with a brief summary.

II. RELATED WORK

In recent years, an enormous amount of researches has been conducted by researchers in exploring different sensing technologies and a number of methods have been proposed for modeling and recognizing human activities [18]. Early researches mainly used decision tree, support vector machine (SVM), naïve Bayes and other traditional machine learning methods to classify the data collected by sensors [19]–[22]. In [19], gradient histogram and Fourier descriptor based on centroid feature were used to extract the features of acceleration and angular velocity data. Then Jain *et al.* [19] used two classifiers, support vector machine and k-nearest neighbor (KNN), to recognize the activities of two public datasets. Jalloul *et al.* [20] used six inertial measurement units to construct a monitoring system. After performing network analysis, a number of network measures that satisfy the statistical test were selected to form a feature set, and then the authors used the random forest (RF) classifier to classify the activities. Finally, an overall accuracy of 84.6% was achieved. The paper [21] presented a wearable wireless accelerometer-based activity recognition system and its application in medical detection. Relief-F and sequential forward floating search (SFFS) were combined for feature selection. Finally, Naïve Bayesian and k-nearest neighbor (KNN) were used for activity classification and comparative analysis.

Machine learning methods may rely heavily on heuristic manual feature extraction in most daily human activity recognition tasks. It is usually limited by human domain knowledge [23]. To address this problem, researchers have turned to deep learning methods that could automatically extract appropriate features from raw sensor data during the training phase and present the low-level original temporal features with high-level abstract sequences. In view of the successful application of deep learning models in image classification, voice recognition, natural language processing, and other fields, it is a new research direction in pattern recognition to transfer it to the field of human activity recognition [24]–[27]. In [24], authors proposed to convert the data acquired by three-axis accelerometers into an ‘‘image’’ format, and then they used CNN with three convolutional layers and one fully-connected layer to identify human activities. Ordóñez and Roggen. [25] proposed an activity recognition classifier, which combined deep CNN and LSTM to classify 27 hand gestures and five movements. Finally, simulation results showed that the F_1 score on the two classifiers were 0.93 and 0.958, respectively. Lin *et al.* [26] presented a novel iterative CNN strategy with autocorrelation pre-processing capability, instead of traditional micro-Doppler image pre-processing, which can accurately classify seven activities or five subjects. And this strategy used an iterative deep learning framework to automatically define and extract features. Finally, traditional supervised learning classifiers were used to mark different activities based on the captured radar signals.

Although the above models could generally recognize human activities, the overall network structure is relatively complex. In addition, these models have a large number of parameters, which results in high computational cost. It is difficult to be used in occasions that require high real-time performance. Many researchers have made great efforts in this regard. Agarwal *et al.* [28] proposed a lightweight deep

TABLE 1. Information of three public datasets.

Datasets	Activities	Sensors	S. Rate	Volunteers	Samples
UCI-HAR	6	A, G	50Hz	30	748,406
WISDM	6	A	20 Hz	36	1,098,209
OPPORTUNITY	17	A, G, M, O, AM	30 Hz	4	701,366

A = accelerometer, G = gyroscope, M = magnetometer, O = object sensor, AM = ambient sensor

learning model for HAR and deployed it on Raspberry Pi3. This model was developed using a shallow RNN in combination with the LSTM algorithm, and its overall accuracy on the WISDM dataset achieved 95.78%. Although the proposed model has high accuracy and brief architecture, it was only evaluated on one dataset which has just six activities, which does not prove that the proposed model has good generalization ability. The paper [29] proposed a deep learning model (InnoHAR) based on the combination of inception neural network and recurrent neural network to classify activities. The authors used separate convolution to replace the traditional convolution, which achieved the goal of reducing model parameters. The results showed an excellent effect, but the model converged hardly, causing a lot of time to be wasted in the training stage.

To address the shortcomings of the above methods, a novel deep neural network for human activity recognition was proposed, which we referred to as LSTM-CNN. The model could extract activity features automatically and classify them with few parameters. In addition, it was evaluated on three of the most widely used public datasets. The results show that the proposed model not only has high accuracy but also has good generalization ability and fast convergence speed.

III. DATASET DESCRIPTION

The information of three public information was summarized in Table 1. It can be seen that there are some differences between them. The UCI-HAR dataset has the largest number of volunteers, which means that this dataset was constructed from the recordings of 30 subjects. The WISDM dataset consists of 6 activities as same as the UCI-HAR dataset, but it has the largest number of samples. And it is an unbalanced dataset, which would be mentioned later. The OPPORTUNITY dataset consists of 17 activities. It was collected by 5 types of sensors, namely accelerometers, gyroscopes, magnetometers, object sensors, and ambient sensors.

A. UCI-HAR

The UCI-HAR dataset [30] was built from the recordings of 30 subjects aged 19-48 years. During the recording, all subjects were instructed to follow an activity protocol. And they wore a smartphone (Samsung Galaxy S II) with embedded inertial sensors around their waist. The six activities of daily living are *standing* (*Std*), *laying* (*Lay*), *walking* (*Walk*), *walking downstairs* (*Down*) and *walking upstairs* (*Up*). In addition, this dataset also includes postural transitions that occur between the static postures: *standing* to *sitting*, *sitting* to *standing*, *sitting* to *laying*, *laying* to *sitting*, *standing* to *laying*, *laying* to *standing*. Specifically, in this paper, only six basic activities were selected as input samples due to the percentage of postural transitions is small. The experiments had been video-recorded to manually label the data. Finally, the researchers captured 3-axial acceleration and 3-axial angular velocity data at a constant rate of 50Hz. According to statistics, the number of samples in this dataset is 748406, and the detailed information was shown in Table 2.

TABLE 2. Activities of UCI-HAR.

Activities	Samples	Percentage
Walk	122,091	16.3%
Up	116,707	15.6%
Down	107,961	14.4%
Sit	126,677	16.9%
Std	138,105	18.5%
Lay	136,865	18.3%

TABLE 3. Activities of WISDM.

Activities	Samples	Percentage
Walk	424,400	38.6%
Jog	342,177	31.2%
Up	122,869	11.2%
Down	100,427	9.1%
Sit	59,939	5.5%
Std	48,397	4.4%

B. WISDM

The WISDM dataset [31] has a total of 1098209 samples, and the percentage of the total samples associated with each activity was shown in Table 3. It can be seen that WISDM is an unbalanced dataset. Activity *walking* takes up the most, reaching 38.6% while *standing* only accounts for 4.4%. Its experimental object consists of 36 subjects. These subjects performed certain daily activities with an Android phone in their front leg pockets. The sensor used is an accelerometer with a sampling frequency of 20 Hz. It is also a built-in motion sensor of the smartphone. Six activities were recorded: *standing* (*Std*), *sitting* (*Sit*), *walking* (*Walk*), *upstairs* (*Up*), *downstairs* (*Down*), and *jogging* (*Jog*). The data collection was supervised by a dedicated person to ensure the quality of data. Fig. 1 shows the acceleration waveform of 2.56 seconds (128 points in total) of each activity with the aim of visualizing the characteristics of the raw data on each axis.

C. OPPORTUNITY

The OPPORTUNITY dataset [32], [33] was collected in a sensor-rich environment, which includes 17 complex gestures and modes of locomotion. Overall, it contains recordings of

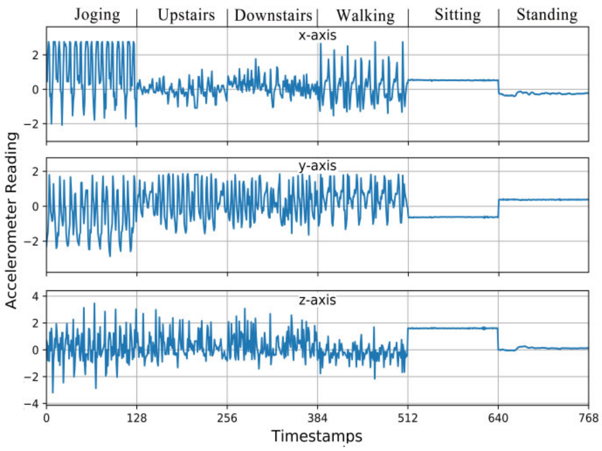

FIGURE 1. Acceleration waveform of 2.56 seconds of each activity.

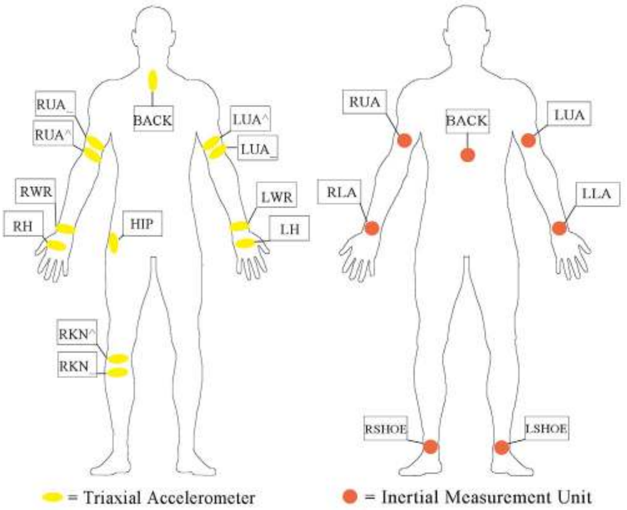

FIGURE 2. Placement of on-body sensors used in the OPPORTUNITY dataset.

four subjects who perform morning activities in daily life scenes. Different modalities of sensors had been integrated into the environment, objects and on the body. In terms of the sensor setting, the OPPORTUNITY challenge guidelines [33] were adopted. We only considered the sensors on the body, including 5 inertial measurement units on the sports jacket, 2 InertiaCube3 sensors on the feet and 12 Bluetooth 3-axis acceleration sensors. As shown in Fig. 2, the yellow oval blocks denote 3-axis accelerometers and red round blocks represent inertial measurement units, where "RSHOE" and "LSHOE" are two InertiaCube3 sensors. During the recording, five activities of daily living (ADL) sessions and one drill session were conducted for each subject. Each sensor axis is considered as a separate channel, resulting in an input space of 113 channels in size. Specifically, these sensors have a sampling rate of 30 Hz. In this paper, we focused only on the recognition of sporadic gestures. Thus, this is an 18-class (including the *Null* class) segmentation and classification problem. The gestures included in this dataset were summarized in Table 4 and the characters in parentheses denote the symbols of gestures.

TABLE 4. Activities of OPPORTUNITY.

Gestures	
Open Door1 (ODo1)	*Open Drawer1* (ODr1)
Open Door2 (ODo2)	*Close Drawer1* (CDr1)
Close Door1 (CDo1)	*Open Drawer2* (ODr2)
Close Door2 (CDo2)	*Close Drawer2* (CDr2)
Open Fridge (OF)	*Open Drawer3* (ODr3)
Close Fridge (CF)	*Close Drawer3* (CDr3)
Clean Table (CT)	*Toggle Switch* (TS)
Drink from Cup (DfC)	*Open Dishwasher* (OD)
Close Dishwasher (CD)	*Null*

D. DATA PRE-PROCESSING

In order to feed the proposed network with a certain data dimension and improve the accuracy of the model, the raw data collected by motion sensors need to be pre-processed as follows.

1) LINEAR INTERPOLATION

The datasets mentioned above are realistic and the sensors worn on the subjects are wireless. Therefore, some data may be lost during the collection process, and the lost data is usually indicated with *NaN*/0. To overcome this problem, the linear interpolation algorithm was used to fill the missing values in this paper.

2) SCALING AND NORMALIZATION

Using large values from channels directly to train models may lead to training bais, So it is necessary to normalize the input data to the range of 0 to 1, as shown in (1):

$$X_{\mathrm{i}} = \frac{X_i - \mathrm{x}_{i\,\min}}{\mathrm{x}_{i\,\max} - \mathrm{x}_{i\,\min}} \quad (i = 1, 2, \cdots, n) \tag{1}$$

where n denotes the number of channels, and $\mathrm{x}_{i\,\max}$, $\mathrm{x}_{i\,\min}$ are the maximum and minimum values of the $i - th$channel, respectively.

3) SEGMENTATION

In this paper, an end-to-end human activity recognition model was implemented. The input to the model consists of a data sequence. The sequence is short time series extracted from the raw sensor data. In the process of data collection, the data were recorded continuously. In order to preserve the temporal relationship between the data points in an activity, a sliding window with an overlap rate of 50% was used to segment the data collected by motion sensors. For the WISDM and UCI-HAR dataset, the length of the sliding window is 128. For the OPPORTUNITY dataset, the recordings of each activity only last for a short period of time, and a short sliding window is needed to segment the data to obtain more samples.

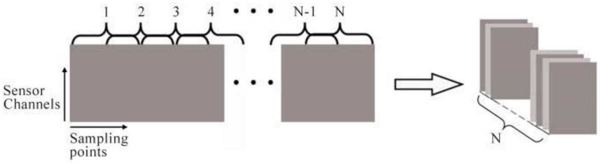

FIGURE 3. Segmentation of sensor data.

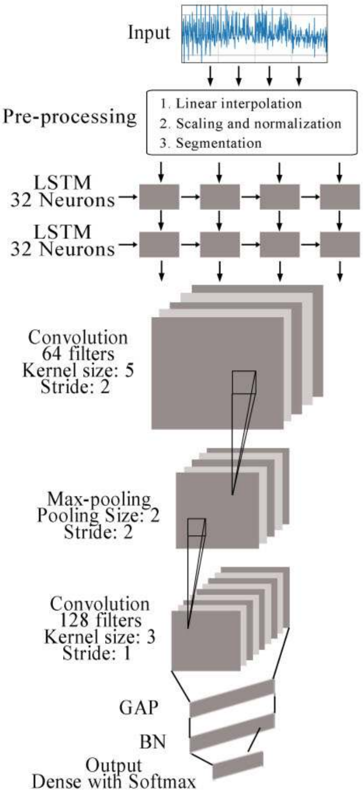
Input
Pre-processing
1. Linear interpolation
2. Scaling and normalization
3. Segmentation
LSTM 32 Neurons
LSTM 32 Neurons
Convolution 64 filters Kernel size: 5 Stride: 2
Max-pooling Pooling Size: 2 Stride: 2
Convolution 128 filters Kernel size: 3 Stride: 1
GAP
BN
Output Dense with Softmax

FIGURE 4. Frame diagram of the LSTM-CNN model.

In this paper, the length of the sliding window for the OPPORTUNITY dataset is chosen to be 24. It is worth noting that our choice on the optimal window size was made in an adaptive and empirical manner [29] to generate good segments for all the activities considered. Fig. 3 shows the details of the segmentation. The horizontal data represents the sampling points and the vertical data represents the sensor channels.

IV. PROPOSED ARCHITECTURE

The network structure of the LSTM-CNN model is as shown in Fig. 4. It consists of eight layers. Firstly, the preprocessed data is fed into a two-layer LSTM with 64 neurons in total. It is used for the extraction of temporal features. Following LSTMs are two other convolutional layers, and it is used for extracting spatial features. The first convolution layer has 64 filters while the other has 128. And between the two convolutional layers is the max-pooling layer. At the end of the model, there is a global average pooling layer (GAP) followed with a batch normalization layer (BN). Finally, the output of the model is obtained from an Output layer (a dense layer with a Softmax classifier), yielding a probability distribution over classes.

A. LSTM LAYERS

RNN could take advantage of the chronological relationship between sensor readings. Although RNN has the ability to capture temporal information from sequential data, it has the problem of gradient vanishing, which hinders the ability of the network to model between raw sensor data and human activities in a long context window. LSTM is a variety of RNN, which could eliminate this limitation. LSTM has great advantages in feature extraction of sequence data than convolutional neural networks due to its special memory cells. In this paper, the input data first passes through two layers of LSTMs to better extract the temporal features in the sequence data. Each layer of LSTMs has 32 memory cells. The inputs are sent to different gates, including input gates, forgetting gates and output gates, to control the behavior of each memory cell. The activation of each LSTM unit is calculated by the following formula:

$$\mathrm{h}_t = \sigma(w_{i,h} \cdot x_t + w_{h,h} \cdot \mathrm{h}_{t-1} + b) \tag{2}$$

where h_t and h_{t-1} represent the activation at time t and $\mathrm{t} - 1$, respectively, σ is a non-linear activation function, $w_{i,h}$ is the input-hidden weight matrix, and $w_{h,h}$ is the hidden-hidden weight matrix, and b is the hidden bias vector.

The output of the LSTM layer has three dimensions (samples, time steps, input dimension), while the size of the input sample of CNN needs four. In order to adapt to the input shape of the convolutional layer, the output of the second layer of LSTM is dimensionally expanded, which could be presented as (samples, 1, time steps, input dimension).

B. CONVOLUTIONAL AND POOLING LAYERS

CNN has gained increasing popularity because of its ability to learn unique representations from images or speech [34]. And the convolutional layer is the most important unit in CNN, which uses convolution kernels to convolve the inputs. It works as a filter and is then activated by a non-linear activation function, as follows:

$$\mathrm{a}_{i,j} = f(\sum_{m=1}^{M}\sum_{n=1}^{N} w_{m,n} \cdot x_{i+m,j+n} + b) \tag{3}$$

where $\mathrm{a}_{i,j}$ is the corresponding activation, $w_{m,n}$ denotes the $\mathrm{m} \times n$ weight matrix of convolution kernel, $x_{i+m,j+n}$ indicates the activation of the upper neurons connected to the neuron (i, j), b is the bias value, and f is a non-linear function.

In this paper, the convolutional layers employ rectified linear units (ReLU) to calculate the feature maps, and its non-linear function is defined as:

$$\sigma(x) = \max(0, x) \tag{4}$$

Generally speaking, the more convolution kernels are used, the more hidden features could be mined in the input samples.

There are two convolutional layers in the LSTM-CNN model. In the first convolutional layer, 64 convolution kernels are used for feature extraction and the size of each convolution kernel is 1 × 5. The sliding step of the convolution window is 2. In the second, 128 convolution kernels are used to perform a deeper feature extraction operation on the features output from the upper layer. Each convolution kernel has a size of 1 × 3 and the convolution window in this layer has a sliding step size of 1. There is a max-pooling layer between the two convolutional layers for performing the downsampling operation. It serves two purposes. One is to reduce the parameters while maintaining dominant features, the other is to filter the interference noise caused by the unconscious jitter of the human body.

C. GLOBAL AVERAGE POOLING LAYER

Different from classical CNN, the model mentioned in this paper used a global average pooling layer (GAP) to replace the fully-connected layer behind the convolutional layer. At the end of CNN, there would usually be one or more fully-connected layers, which could convert multi-D feature maps into a 1D feature vector. Each node of the fully-connected layer is connected with the nodes of the upper layer, thus the weight parameters of the fully-connected layer may occupy the most. For instance, in the model Krizhevsky [35], the first fully-connected layer FC1 has 4096 nodes, and the output of the upper pooling layer MaxPool3 has 9216 nodes. Thus, there would be more than 37 million weight parameters between the MaxPool3 layer and the FC1 layer, which would consume a lot of memory and computational cost. Unlike the fully-connected layer, the GAP layer performs a global averaging pooling operation on each feature map. There is no parameter to optimize in the GAP layer. Thus, it achieves the goal of reducing global model parameters. Furthermore, GAP sums out the spatial information, so it is more robust to the spatial transformation of the input.

D. BATCH NORMALIZATION LAYER

During the training process, the distribution of input data of each layer would continuously change due to the weight parameters of the upper layer are constantly updated. Therefore, it is necessary to change the weight parameters to adapt to this new distribution, which leads to difficulty in network training and slows down the convergence speed. To address this problem, a batch normalization layer (BN) is added after the GAP layer to accelerate the convergence of the model. The BN layer normalizes and reconstructs the input data on each batch of training samples to ensure the stability of the output of the previous layer, so as to improve the training speed and accuracy.

E. OUTPUT LAYER

In the LSTM-CNN model, the output layer consists of a fully-connected layer and a Softmax classifier. There is an important benefit to adding the fully-connected layer at the end of the model. Each node of the fully-connected layer is connected to the nodes of the upper layer so that the features extracted from the upper layer could be merged. It makes up for the shortcomings of the GAP layer in this regard.

Behind the fully-connected layer is the Softmax classifier which converts the output of the upper layer into a probability vector whose value represents the probability of classes to which the current sample belongs. The expression formula is as follows:

$$S_j = \frac{e^{a_j}}{\sum_{k=1}^{N} e^{a_k}} \tag{5}$$

where N is the number of classes, a is the output vector of the fully-connected layer, and a_j is the $j - th$ value of the output vector.

TABLE 5. Instances of three public datasets.

	HAR-UCI	WISDM	OPPORTUNITY
Training set	7,319	13,654	43,412
Test set	3,069	3,036	9,308

V. EXPERIMENTAL RESULTS

In this paper, three widely used public datasets were used to evaluate the generalization ability and the accuracy of the LSTM-CNN model. They were all recorded continuously and a common method is to use a fixed-length sliding window to segment the sensor data. Here, the length of the window is 128, with a step size of 64. But for the OPPORTUNITY dataset, the length of the window is 24. To be specific, a subset of the dataset was used to construct the test set that is separated entirely from the training set to better evaluate the model performance. For the UCI-HAR dataset, the database was built from the recordings of 30 subjects who performed 6 activities. Among them, the recordings of 22 subjects were used to build the training set, and the rest was used to build the test set. The WISDM dataset consists of 6 activities performed by 36 subjects. The training set is composed of the recordings of 30 subjects and the remaining recordings of 6 subjects are used to build the test set. The two parts are completely separate. For the OPPORTUNITY dataset, the same subset employed in the OPPORTUNITY challenge [33] was used to train and test our models. The training set includes the full recordings of Subject 1, as well as for three ADLs and drill sessions of subjects 2 and 3. And the test set composes of ADL4 and ADL5 for Subjects 2 and 3. Table 5 details the number of instances of the test set and the training set obtained on each dataset after segmentation.

A. MODEL IMPLEMENTATION

Keras was used to build the proposed network structure, which is a high-level neural networks API written in Python capable of running on top of TensorFlow, CNTK, or Theano. In the experiments, TensorFlow was used as

TABLE 6. List of selected hyper-parameters.

Stage	Hyper-parameters		Selected Values
Data Preprocessing	Window Size		128/24
	Step Size		64/12
Architecture	LSTM_1 Neurons		32
	LSTM_2 Neurons		32
	Convolution_1	Kernel Size	5
		Stride	2
		Filters	64
	Max-pooling	Pooling Size	2
		Stride	2
	Convolution_2	Kernel Size	3
		Stride	1
		Filters	128
Training	Optimizer		Adam
	Batch Size		192
	Learning Rate		0.001
	Number of Epochs		200

the backend. The model training and classification were on a PC that has an E5-2620 Xeon CPU with 2.10 GHz, 64GB RAM and an NVIDIA QUADRO P5000 graphics card with 16 GB memory. And the PC is equipped with an Ubuntu operating system with 64 bits.

The model was trained in a fully-supervised manner, and the gradient was back-propagated from the Softmax layer to the LSTM layer. The weights and biases of each layer were initialized by randomly selected values. Cross entropy is used to evaluate the difference between the real distribution and the probability distribution. In this paper, the cross-entropy loss function was used to measure the error between the prediction and the true values. Adam [36] is a stochastic optimization algorithm based on the first-order gradient, here it was selected as the optimizer. For the sake of efficiency, in the training stage, the batch size was set to 192 and the number of epochs was 200. Furthermore, a small learning rate of 0.001 was used to enhance the fitting ability, and the order of the training set was randomly shuffled to improve the robustness of the model. The selected hyper-parameters were listed in Table 6.

B. PERFORMANCE MEASURE

When collecting human activity data in natural environments, imbalances often occur [37]. The WISDM and OPPORTUNITY mentioned above are both imbalanced datasets. If the classifier predicts each instance as a majority class and uses the overall classification accuracy to evaluate the model performance, the results could achieve high accuracy. Therefore the overall classification accuracy is not an appropriate measure of performance. F-measure (F_1 score) takes both false positives and false negatives into account and it combines two measures defined based on the total number of correctly recognized samples, which is known in the information retrieval community as "precision" and "recall". Thus, the F_1 score is usually a more useful performance indicator than accuracy. Precision corresponds to $\frac{TP}{TP+FP}$, and recall is defined as $\frac{TP}{TP+FN}$, where TP, FP are the number of true and false positives, respectively, and FN corresponds to the number of false negatives. F_1 score offsets imbalances in classes by weighting classes based on their proportion of samples. The formula of the F_1 score is as follows:

$$F_1 = \sum_{\mathrm{i}} 2 * w_i \frac{precision_i \cdot recall_i}{precision_i + recall_i} \tag{6}$$

where $\mathrm{w_i} = n_i/N$ is the proportion of samples of class i, with n_i being the number of samples of the $i-th$ class and N being the total number of samples.

C. EVALUATION ON THREE PUBLIC DATASETS

In order to comprehensively verify the performance of the proposed model, three public datasets were used for testing. Table 7, 8, and 9 show the classification confusion matrices obtained when the model was predicted with the test set of the UCI-HAR, WISDM, and OPPORTUNITY datasets, respectively. For the UCI-HAR dataset, there were 2940 instances that have been correctly classified, and the overall accuracy reached 95.80%. There was relatively poor discrimination between *sitting* and *standing*. The recall and precision were in the range of 92%∼93%. The main reason may be that the two activities are similar from the perspective of motion sensors. It is difficult to mine deeper information only by acceleration and angular velocity data. When the trained model was exposed to the test set that contains approximately 3036 new instances, the overall accuracy of the WISDM dataset (an unbalanced dataset) reached 95.75%. The OPPORTUNITY dataset is just as unbalanced as the WISDM and it contains 17 activities in the gesture recognition case. Finally, an overall accuracy of 92.63% was achieved. In addition, when the *Null* class is removed from the classification task (see Table 10), our method achieved an overall accuracy of 87.58% in the gesture recognition task.

In order to further verify the performance of the model, LSTM-CNN was compared with CNN of Yang *et al.* [38], and DeepConvLSTM [25] under the same experiment scenario. All the results were verified by the F_1 score to ensure the fairness and consistency of the following comparison results. Fig. 5 shows the evaluation results of the deep models mentioned above. Compared with the CNN model of Yang *et al.*, LSTM-CNN has a significant increase of about 7% for the OPPORTUNITY dataset and is superior to the DeepConvLSTM model. It can also be seen that LSTM-CNN outperforms the other two models on the UCI-HAR and WISDM datasets, with the best-reported result increasing by an average of 3%. It should be noted that the model parameters have been greatly reduced under adding the GAP layer to the network. These results confirm our findings that supporting the use of the GAP layer instead of a fully-connected layer brings significant advantages in HAR tasks. It also proves that the proposed method has superior performance on different public datasets.

TABLE 7. Classification confusion matrix on the UCI-HAR.

	Activities	Predicted Label						
		Walk	*Up*	*Down*	*Sit*	*Std*	*Lay*	Recall (%)
True Label	*Walk*	478	0	0	0	0	0	100
	Up	3	440	0	0	0	0	99.32
	Down	21	23	371	0	0	0	89.40
	Sit	0	0	0	517	37	0	93.32
	Std	3	0	0	43	542	0	92.18
	Lay	0	0	0	0	0	592	100
	Precision (%)	94.65	95.03	100	92.32	93.61	100	**95.80**

TABLE 8. Classification confusion matrix on the WISDM.

	Activities	Predicted Label						
		Down	*Jog*	*Sit*	*Std*	*Up*	*Walk*	Recall (%)
True Label	*Down*	256	0	0	0	11	1	95.52
	Jog	18	950	0	0	7	26	94.91
	Sit	0	0	211	0	0	0	100
	Std	0	0	0	146	6	0	96.05
	Up	36	0	0	0	269	1	87.91
	Walk	17	0	0	0	6	1075	97.91
	Precision (%)	78.29	100	100	100	89.97	97.46	**95.75**

TABLE 9. Classification confusion matrix on the OPPORTUNITY.

	Activities	Predicted Label																	
		Null	*CD*	*CDr3*	*CDr3*	*CDo1*	*CDo2*	*CDr1*	*CF*	*TS*	*OD*	*ODr3*	*ODr2*	*ODo1*	*ODo2*	*ODr1*	*OF*	*DfC*	*CT*	Recall (%)
True Label	*Null*	6250	10	5	13	7	7	8	17	13	6	6	7	4	6	15	11	79	9	96.55
	CD	18	131	0	6	0	0	0	0	0	0	0	0	0	0	0	0	3	0	82.91
	CDr3	31	0	128	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	77.58
	CDr2	9	14	0	127	0	0	0	0	0	0	0	0	0	0	0	0	0	0	84.67
	CDo1	10	0	2	0	134	0	0	0	0	0	0	0	0	0	0	0	0	0	91.78
	CDo2	45	0	0	0	0	124	2	3	3	0	0	2	0	0	0	0	0	0	69.27
	CDr1	33	0	0	0	0	11	120	0	1	0	0	1	0	0	0	0	0	1	71.86
	CF	6	0	0	0	0	0	0	99	5	0	0	0	0	0	0	0	1	0	89.19
	TS	15	0	0	0	0	1	0	5	89	0	0	0	1	0	1	0	0	0	79.46
	OD	10	0	0	0	0	0	0	0	0	78	0	3	0	0	0	0	0	0	85.71
	ODr3	3	0	0	0	0	0	0	0	0	10	39	4	4	0	0	0	0	0	65.00
	ODr2	2	0	0	0	0	0	0	0	0	0	1	83	4	1	0	0	0	0	91.21
	ODo1	6	0	0	0	0	0	0	0	0	0	1	10	56	0	0	0	0	0	76.71
	ODo2	5	0	0	0	0	0	0	0	0	0	1	0	1	95	7	0	0	0	87.16
	ODr1	3	0	0	0	0	0	2	0	0	0	0	2	2	3	88	0	0	0	88.00
	OF	12	0	0	0	2	0	0	0	0	0	0	0	0	0	0	150	0	0	91.46
	DfC	97	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	718	0	87.78
	CT	15	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	120	85.11
	Precision (%)	95.13	84.52	94.81	86.99	89.93	86.71	90.91	77.95	80.18	78.00	81.25	74.11	77.78	90.48	79.28	93.17	89.64	92.31	**92.71**

D. IMPACT OF NETWORK STRUCTURE ON MODEL PERFORMANCE

In this section, we explored the impact of several network structures on model performance. As shown in Table 11, five kinds of model architectures (A, B, C, D, and LSTM-CNN) were constructed respectively for experimental comparison, and the classification results were evaluated by the number of model parameters and the F_1 score on the test set. Furthermore, in terms of training iterations, the computation speed in the forward phase was given. The experiments were implemented based on the UCI-HAR dataset.

The structure of model A belongs to the classical convolutional neural network structure, in which the number of nodes in the fully-connected layer is 128. In the classical CNN structure, the last convolutional layer is usually followed by a fully-connected layer to synthesize the features extracted from previous layers. Although this could improve the accuracy of the model, it also brings a huge number of

TABLE 10. Classification confusion matrix on the OPPORTUNITY (without the null class).

Activities		Predicted Label																	
		CD	*CDr3*	*CDr3*	*CDo1*	*CDo2*	*CDr1*	*CF*	*TS*	*OD*	*ODr3*	*ODr2*	*ODo1*	*ODo2*	*ODr1*	*OF*	*DfC*	*CT*	Recall (%)
True Label	*CD*	133	1	24	1	0	1	0	0	0	0	0	0	0	0	1	1	0	82.1
	CDr3	0	122	8	7	2	0	0	0	0	0	0	0	0	0	0	0	0	87.77
	CDr2	14	4	108	3	0	0	0	0	0	0	0	0	0	0	0	0	0	83.72
	CDo1	0	11	1	144	0	0	0	0	0	0	0	0	0	0	1	0	0	91.72
	CDo2	1	0	0	3	131	4	1	2	0	0	0	0	0	0	0	4	0	89.73
	CDr1	0	2	4	0	2	168	2	1	0	0	0	1	0	0	0	3	0	91.8
	CF	0	0	0	0	2	1	127	6	0	0	0	0	0	1	11	5	0	83.01
	TS	0	0	0	0	0	3	6	112	3	0	0	0	1	3	1	6	1	82.35
	OD	0	0	0	0	0	0	0	0	94	11	3	1	0	0	0	0	4	83.19
	ODr3	0	0	0	0	0	2	0	1	16	63	2	2	0	0	0	1	2	70.79
	ODr2	0	0	0	0	0	0	0	0	3	5	69	8	4	0	0	0	0	77.53
	ODo1	0	0	0	0	0	0	0	0	0	7	5	51	5	4	0	0	0	70.83
	ODo2	0	0	0	0	0	0	0	1	0	0	5	3	78	16	0	0	0	75.73
	ODr1	0	0	0	0	0	2	0	1	0	0	1	2	13	66	0	0	0	77.65
	OF	1	0	3	0	0	0	15	2	0	0	0	0	0	0	148	6	0	84.57
	DfC	6	1	4	0	1	3	5	2	0	0	0	1	0	0	7	753	0	96.17
	CT	0	0	0	0	0	1	0	0	3	1	0	0	0	0	0	0	116	95.87
	Precision (%)	85.81	86.52	71.05	91.14	94.93	90.81	81.41	87.5	78.99	72.41	81.18	73.91	77.23	73.33	87.57	96.66	94.31	**87.58**

TABLE 11. Experiments on different network architectures.

Models	Structures									Total Parameters	F_1 Score	Computation Speed (ms/epoch)
A			Conv_1	Max-pooling	Conv_2	Flatten	Dense	Output		502,726	91.88%	1681
B			Conv_1	Max-pooling	Conv_2	GAP		Output		27,462	91.78%	1202
C			Conv_1	Max-pooling	Conv_2	GAP	BN	Output		27,974	93.35%	1323
D	LSTM_1		Conv_1	Max-pooling	Conv_2	GAP	BN	Output		41,286	94.64%	5156
LSTM-CNN	LSTM_1	LSTM_2	Conv_1	Max-pooling	Conv_2	GAP	BN	Output		49,606	95.78%	9416

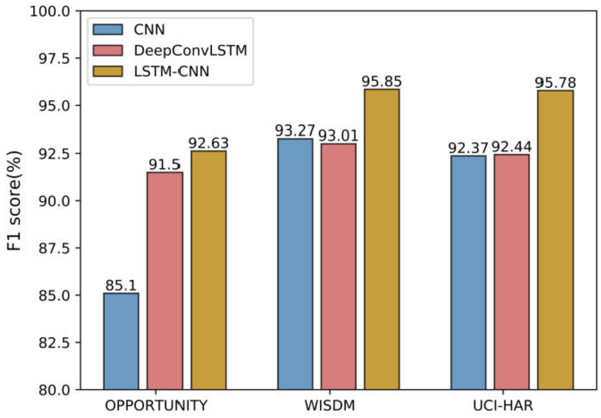

FIGURE 5. Performance of three models on three public datasets.

parameters. As can be seen that the F_1 score of model A reaches 91.88% when the trained model is exposed to the test set of the UCI-HAR. Accordingly, there are more than 502 thousand model parameters, of which the parameters of fully-connected layers occupy the most. And it takes 1681 milliseconds per epoch in the training stage. In model B, the GAP layer is used to replace the fully-connected layer behind the convolutional layer to perform a global averaging pooling operation on each feature map output from the upper layer, which structurally regularizes the entire network to reduce the over-fitting problem. The parameters of the model B are only 27462, which is about 94% less than that of the model A while the performance remains almost the same. It proves the feasibility of replacing the fully-connected layer with the GAP layer. Accordingly, the computation speed has also been improved, with an average of 1202 milliseconds per epoch. However, the use of the GAP layer would focus the training pressure of the model on the convolutional layers, which would cause the model to converge slowly. The model C adds a BN layer after the GAP layer to stabilize the output of the upper layer. It speeds up the convergence of the model and improves accuracy. Finally, the F_1 score of this model reaches 93.35%. The recordings of activities based on mobile sensors are temporal sequences and LSTM has the ability to capture temporal information from sequential data. In model D, the data captured from mobile sensors are firstly fed into two layers of LSTMs and then transmitted to convolutional layers for feature extraction. Finally, it outperforms

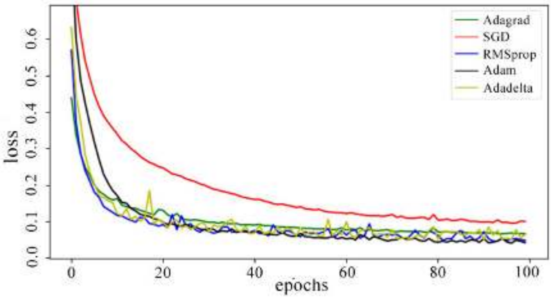

FIGURE 6. Impact of optimizer on model performance.

the model C by 1% on average. In our work, we added another layer of LSTM, in a total of 2 layers, on the basis of model D to further improve the model performance. Eventually, the F_1 score on the test set reached the expected 95.78%. It could be seen that the computation speed of the model D and LSTM-CNN is greatly reduced. They are 5156 milliseconds per epoch and 9416 milliseconds per epoch, respectively. This is due to the LSTM layers added to the model. It is because of its special network structure that LSTM could extract temporal information effectively. However, every coin has two sides. When training LSTM layers, the calculation of each time step depends on the output of the previous time step. As a result, it could not compute in parallel, which slows down the computation speed of the model.

To sums up, the strategy of using the global average pooling layer and batch normalization layer to replace the fully-connected layer is effective. Moreover, the method of using LSTM to extract the temporal information to improve model performance is favorable. The model proposed in this paper not only could achieve high recognition accuracy but also greatly simplify the model structure.

E. IMPACT OF HYPER-PARAMETERS ON MODEL PERFORMANCE

Hyper-parameters have a great impact on the classification model performance. This section presents the impact of the important hyper-parameters such as the number of convolution filters, the batch size and the type of optimizer on model performance. The experiments were implemented on the UCI-HAR dataset and the model performance was evaluated by varying a number of model parameters. F1 score was used as the measurement criteria.

1) EFFECT OF OPTIMIZER

Optimizer is used to update and calculate network parameters that affect model training and the output, so as to approximate or reach the optimal value, thereby minimizing the loss function. It is the essence of neural network training. Thus it is important to choose a suitable optimizer to train deep models. Several common optimizers such as SGD, Adagrad, Adadelta, Adam, and RMSprop were experimentally verified, as shown in Fig. 6. It can be seen that the model trained by Adam optimizer has the best fitting effect and the fluctuation of gradient descent curve is the most stable. Here, Adam was used as the optimizer when training the model LSTM-CNN.

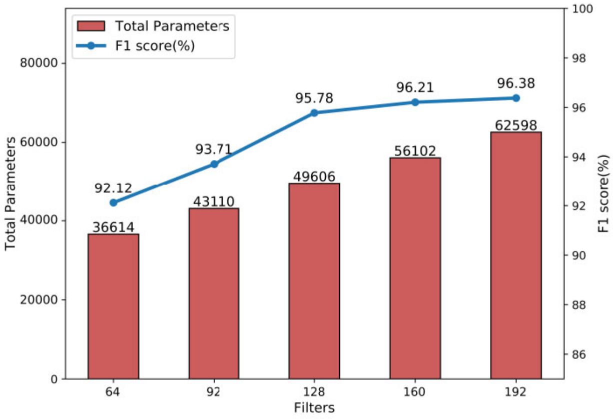

FIGURE 7. Impact of increasing number of filters of the second convolutional layer on model performance.

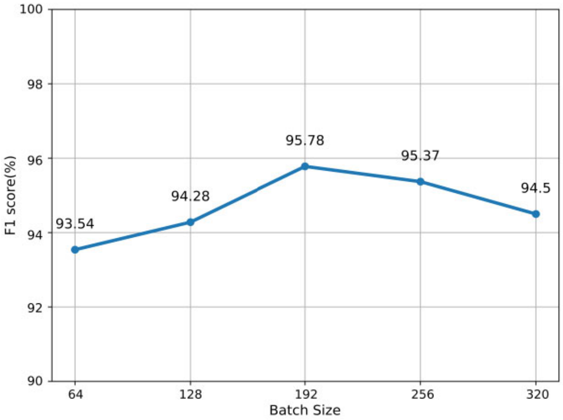

FIGURE 8. Impact of batch size on model performance.

2) EFFECT OF NUMBER OF FILTERS

The more filters (namely convolution kernels), the more complex and deeper features the model may learn. But it also increases the model parameters, which may lead to overfitting. Thus, how to choose the number of filters is of critical importance. Fig. 7 shows the accuracy and parameters of the model LSTM-CNN with a varying number of filters of the second convolutional layer. With an increasing number of filters, the network parameters increase from 36614 to 62598. Absolutely, the accuracy of the model does increase correspondingly. F_1 score reaches 96.38% when the number of filters is selected as 192, which outperforms when the number of filters is 64 by 4%. However, the model parameters increase by more than 70%.

3) EFFECT OF BATCH SIZE

Mini-batch processing is a common method in deep learning when training neural networks. Optimizing the cumulative error over the entire training set would make the gradient descent slowly, also may lead the model into local optimum. If the error of only one sample is optimized in one iteration, the gradient descent could fluctuate drastically, which would eventually lead to difficulty in training. Fig. 8 presents the accuracy varying with 5 different batch sizes. It can be seen that the accuracy reaches the highest when the batch size is selected as 192.

VI. CONCLUSION

A novel deep neural network that combines convolutional layers with LSTM for human activity recognition was proposed in this paper. The weight parameters of CNN mainly concentrate on the fully-connected layer. In response to this characteristic, a GAP layer is used to replace the fully-connected layer behind the convolutional layer, which greatly reduces the model parameters while maintaining a high recognition rate. Moreover, a BN layer is added after the GAP layer to speed up the convergence of the model and obvious effect was obtained. In the proposed architecture, the raw data collected by mobile sensors is fed into a two-layer LSTMs followed by convolutional layers, which makes it capable of learning the temporal dynamics on various time scales according to the learned parameters of LSTMs so as to obtain better accuracy. In order to prove the generalization ability and effectiveness of the proposed model, the three public datasets, UC-HAR, WISDM, and OPPORTUNITY, were used for the experiment. Considering that the accuracy is not an appropriate and comprehensive measure of performance, the F_1 score was used to evaluate the model performance. Eventually, the F_1 score reached 95.78%, 95.85% and 92.63% on the UCI-HAR, WISDM and OPPORTUNITY datasets, respectively. Furthermore, we also explored the impact of some hyper-parameters on model performance such as the number of filters, the type of optimizers and batch size. Finally, the optimal hyper-parameters for the final design were selected to train the model. To sum up, compared with the methods proposed in other literatures, the LSTM-CNN model shows consistent superior performance and has good generalization. It can not only avoid complex feature extraction but also has high recognition accuracy under the premise of a few model parameters.

REFERENCES

[1] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognit. Lett.*, vol. 119, pp. 3–11, Mar. 2019.

[2] P. Vepakomma, D. De, S. K. Das, and S. Bhansali, "A-wristocracy: Deep learning on wrist-worn sensing for recognition of user complex activities," in *Proc. IEEE 12th Int. Conf. Wearable Implant. Body Sensor Netw. (BSN)*, Jun. 2015, pp. 1–6.

[3] J. Qin, L. Liu, Z. Zhang, Y. Wang, and L. Shao, "Compressive sequential learning for action similarity labeling," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 756–769, Feb. 2016.

[4] N. Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," 2016, *arXiv:1604.08880*. [Online]. Available: http://arxiv.org/abs/1604.08880

[5] Y. Kim and B. Toomajian, "Hand gesture recognition using micro-Doppler signatures with convolutional neural network," *IEEE Access*, vol. 4, pp. 7125–7130, 2016.

[6] M. Cornacchia, K. Ozcan, Y. Zheng, and S. Velipasalar, "A survey on activity detection and classification using wearable sensors," *IEEE Sensors J.*, vol. 17, no. 2, pp. 386–403, Jan. 2017.

[7] K. Yatani and K. N. Truong, "BodyScope: A wearable acoustic sensor for activity recognition," in *Proc. ACM Conf. Ubiquitous Comput. (UbiComp)*, 2012, pp. 341–350.

[8] B. Cagliyan, C. Karabacak, and S. Z. Gurbuz, "Human activity recognition using a low cost, COTS radar network," in *Proc. IEEE Radar Conf.*, May 2014, pp. 1223–1228.

[9] X. Yang and Y. Tian, "Super normal vector for human activity recognition with depth cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 5, pp. 1028–1039, May 2017.

[10] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3007–3021, Dec. 2018.

[11] K. M. Kitani, T. Okabe, Y. Sato, and A. Sugimoto, "Fast unsupervised ego-action learning for first-person sports videos," in *Proc. CVPR*, Jun. 2011, pp. 3241–3248.

[12] M. R. Amer and S. Todorovic, "Sum product networks for activity recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 800–813, Apr. 2016.

[13] W. Lin, S. Xing, J. Nan, L. Wenyuan, and L. Binbin, "Concurrent recognition of cross-scale activities via sensorless sensing," *IEEE Sensors J.*, vol. 19, no. 2, pp. 658–669, Jan. 2019.

[14] I. H. Lopez-Nava and A. Munoz-Melendez, "Wearable inertial sensors for human motion analysis: A review," *IEEE Sensors J.*, vol. 16, no. 22, pp. 7821–7834, Nov. 2016.

[15] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 42, no. 6, pp. 790–808, Nov. 2012.

[16] J. Margarito, R. Helaoui, and A. M. Bianchi, "User-independent recognition of sports activities from a single wrist-worn accelerometer: A template-matching-based approach," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 4, pp. 788–796, Apr. 2016.

[17] C. Zhu and W. Sheng, "Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 41, no. 3, pp. 569–573, May 2011.

[18] L. Chen, C. D. Nugent, and H. Wang, "A knowledge-driven approach to activity recognition in smart homes," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 6, pp. 961–974, Jun. 2012.

[19] A. Jain and V. Kanhangad, "Human activity classification in smartphones using accelerometer and gyroscope sensors," *IEEE Sensors J.*, vol. 18, no. 3, pp. 1169–1177, Feb. 2018.

[20] N. Jalloul, F. Poree, G. Viardot, P. L'Hostis, and G. Carrault, "Activity recognition using complex network analysis," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 4, pp. 989–1000, Jul. 2018.

[21] P. Gupta and T. Dallas, "Feature selection and activity recognition system using a single triaxial accelerometer," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 6, pp. 1780–1786, Jun. 2014.

[22] E. Fullerton, B. Heller, and M. Munoz-Organero, "Recognizing human activity in free-living using multiple body-worn accelerometers," *IEEE Sensors J.*, vol. 17, no. 16, pp. 5290–5297, Aug. 2017.

[23] Y. Bengio, "Deep learning of representations: Looking forward," in *Proc. Int. Conf. Stat. Lang. Speech Process.* Berlin, Germany: Springer, 2013, pp. 1–37.

[24] Y. Zheng, Q. Liu, and E. Chen, "Time series classification using multi-channels deep convolutional neural networks," in *Proc. Int. Conf. Web-Age Inf. Manage.* Cham, Switzerland: Springer, 2014, pp. 298–310.

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

[26] Y. Lin, J. Le Kernec, S. Yang, F. Fioranelli, O. Romain, and Z. Zhao, "Human activity classification with radar: Optimization and noise robustness with iterative convolutional neural networks followed with random forests," *IEEE Sensors J.*, vol. 18, no. 23, pp. 9669–9681, Dec. 2018.

[27] M.-O. Mario, "Human activity recognition based on single sensor square HV acceleration images and convolutional neural networks," *IEEE Sensors J.*, vol. 19, no. 4, pp. 1487–1498, Feb. 2019.

[28] P. Agarwal and M. Alam, "A lightweight deep learning model for human activity recognition on edge devices," 2019, *arXiv:1909.12917*. [Online]. Available: https://arxiv.org/abs/1909.12917

[29] C. Xu, D. Chai, J. He, X. Zhang, and S. Duan, "InnoHAR: A deep neural network for complex human activity recognition," *IEEE Access*, vol. 7, pp. 9893–9902, 2019.

[30] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, Jan. 2016.

[31] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Mar. 2011.

[32] D. Roggen, A. Calatroni, M. Rossi, T. Holleczek, K. Forster, G. Troster, P. Lukowicz, D. Bannach, G. Pirkl, A. Ferscha, J. Doppler, C. Holzmann, M. Kurz, G. Holl, R. Chavarriaga, H. Sagha, H. Bayati, M. Creatura, and J. D. R. Millan, "Collecting complex activity datasets in highly rich networked sensor environments," in *Proc. 7th Int. Conf. Networked Sens. Syst. (INSS)*, Jun. 2010, pp. 233–240.

[33] R. Chavarriaga, H. Sagha, A. Calatroni, S. T. Digumarti, G. Tröster, J. D. R. Millán, and D. Roggen, "The opportunity challenge: A benchmark database for on-body sensor-based activity recognition," *Pattern Recognit. Lett.*, vol. 34, no. 15, pp. 2033–2042, Nov. 2013.

[34] C. A. Ronaoo and S. B. Cho, "Evaluation of deep convolutional neural network architectures for human activity recognition with smartphone sensors," in *Proc. KIISE Korea Comput. Congr.*, 2015, pp. 858–860.

[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[37] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Syst. Appl.*, vol. 59, pp. 235–244, Oct. 2016.

[38] J. Yang, M. N. Nguyen, X. L. Li, and P. P. San, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jun. 2015.

KUN XIA was born in China, in 1980. He received the B.Eng. degree in industrial automation and the Ph.D. degree in power electronics and power drives from the Hefei University of Technology (HFUT), Hefei, China, in 2002 and 2007, respectively. He was a Visiting Scholar with the Electrical and Computer Engineering Department, National University of Singapore, Singapore, in 2015. From 2007 to 2011, he was a Lecturer with the University of Shanghai for Science and Technology (USST), Shanghai, China, where he has been an Associate Professor and the Department Head of the Electrical Engineering Department, since 2011. His current research interests include motor control and deep learning.

JIANGUANG HUANG was born in China, in 1996. He received the B.Eng. degree from the Department of Electrical Engineering, University of Shanghai for Science and Technology (USST), Shanghai, China, in 2018, where he is currently pursuing the M.Eng. degree. His current research interests include motor control and deep learning.

HANYU WANG was born in China, in 1995. She received the B.Eng. degree from the Department of Electrical Engineering, University of Shanghai for Science and Technology (USST), Shanghai, China, in 2018, where she is currently pursuing the M.Eng. degree with the Electrical Engineering Department. Her current research interests include motor control and deep learning.