PAPER

The Effect of Axis-Wise Triaxial Acceleration Data Fusion in CNN-Based Human Activity Recognition

Xinxin HAN[†,††], **Jian YE**[††,†††], ***Nonmembers***, **Jia LUO**[††††a)], ***Member***, ***and*** **Haiying ZHOU**[†], ***Nonmember***

SUMMARY The triaxial accelerometer is one of the most important sensors for human activity recognition (HAR). It has been observed that the relations between the axes of a triaxial accelerometer plays a significant role in improving the accuracy of activity recognition. However, the existing research rarely focuses on these relations, but rather on the fusion of multiple sensors. In this paper, we propose a data fusion-based convolutional neural network (CNN) approach to effectively use the relations between the axes. We design a single-channel data fusion method and multichannel data fusion method in consideration of the diversified formats of sensor data. After obtaining the fused data, a CNN is used to extract the features and perform classification. The experiments show that the proposed approach has an advantage over the CNN in accuracy. Moreover, the single-channel model achieves an accuracy of 98.83% with the WISDM dataset, which is higher than that of state-of-the-art methods.

key words: *triaxial accelerometer, human activity recognition, data fusion, convolutional neural network*

1. Introduction

Human activity recognition (HAR) is widely used in security, medical, smart homes and entertainment. HAR detects and recognizes human activities in the real environment [1] by learning useful information from raw sensor data or videos containing human motions. In the past, machine learning algorithms that relied on manual design features were used to solve HAR tasks [2]. However, the performance of the model is limited because machine learning algorithms heavily rely on manual design features and can only extract shallow features.

In recent years, an increasing number of mobile devices have been embedded with sensors such as accelerometers and gyroscopes, which has greatly promoted the development of sensor-based HAR [3]. A sensor-based model reduces the computational complexity and extends the application scenarios compared to that of a video-based model, which makes the sensor-based model increasingly popular in HAR. For example, Chen and Shen analyzed the performance of smartphone-sensor behavior [4]. In [5], the authors believe that not only accelerometer plays an important role in solving HAR problems, but also the role of gyroscopes is worth exploring. In [6], experiments show that accelerometers contain more discriminant information than gyroscopes, while using both will improve classification performance. With the impressive performance of deep learning in image recognition and speech recognition, an increasing number of researchers have attempted to use deep learning methods to solve sensor-based HAR and has achieved good performance [7], [8]. Deep neural networks can obtain higher accuracy than traditional machine learning methods and can be used in different application scenarios [9]. Gjoreski et al. demonstrated that deep learning has the ability to automatically extract features, while at the same time pointing out that it performs poorly in solving HAR problem with small amount of data [10]. Compared with traditional machine learning methods, deep learning automatically extracts features by training an end-to-end network instead of relying on experience to manually extract features [11], [12].

The triaxial accelerometer is one of the most important sensors used in sensor-based HAR [13]. The relations between axes are called hidden relations in [14] and correlation in [15]–[17]. Hidden relations between axes can help the deep activity recognition model increase its accuracy [14], [16], especially for discriminating between activities that involve translation in just one dimension [18]. The relations between the axes in three directions is considered in [19]. In [20], [21], the authors consider the relations between every two axes. For instance, using synthetic acceleration to reduce the rotational interference caused by different placement positions of mobile phones and arranging sensor data to generate an activity image consider the relations between axes to some extent. However, the design of these methods depends largely on the experience of the researchers and has a greater impact on the classification accuracy. These methods have certain limitations.

To solve this problem, we propose a data fusion-based convolution neural network (CNN) approach. Single-channel data fusion method and multichannel data fusion method are designed in consideration of the diversified formats of sensor data. After using the data fusion method to obtain fused data, a CNN is used for the feature extraction and classification. The experimental results indicate that our approach has a higher classification accuracy than that of the CNN. When the single-channel data fusion method is used, the accuracy exceeds that of the state-of-the-art

Manuscript received December 6, 2018.
Manuscript revised August 6, 2019.
Manuscript publicized January 14, 2020.

[†]The authors are with School of Data Science, North University of China, TaiYuan 030051, China.
[††]The authors are with Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China.
[†††]The author is with Beijing Key Laboratory of Mobile Computing and Pervasive Device, Beijing 100000, China.
[††††]The author is with College of Economics and Management, Beijing University of Technology, Beijing 100124, China.
a) E-mail: jia.luo.bjut@hotmail.com (Corresponding author)

DOI: 10.1587/transinf.2018EDP7409

methods. The contributions of this paper are as follows:

1. We consider the hidden relations between axes and propose a data fusion-based convolutional neural network approach to explore the impact of the hidden relations on classification accuracy.
2. We design single-channel and multichannel data fusion method to discuss the influence of different data formats of sensor on classification accuracy.
3. We show that the proposed approach is suitable for sensor-based human activity recognition, and is especially suitable for solving the problem of activity recognition with large differences in triaxial data.
4. We demonstrate that our approach has an advantage over the CNN and state-of-the-art methods.

The remainder of this paper is organized as follows. In Sect. 2, we review sensor-based HAR. The proposed methods and model architecture are introduced in Sect. 3. Section 4 shows the dataset and the specific information of the experiments. Section 5 covers the conclusion of this paper.

2. Related Work

Deep learning has the ability to find intricate structures and is good at dealing with high-dimensional data [22], which makes it widely used in HAR.

The sensor data here refer to the data segmented using a sliding window. In [23], a deep CNN was proposed to perform activity recognition, and the author explored the influence of different hyperparameter settings on the classification accuracy. Using a deep CNN to handle multichannel time-series signals was presented in [11]. Hammerla et al. [24] provided the first systematic exploration of the performance of deep, convolutional, and recurrent models. Ordóñez and Roggen [25] combined convolutional and LSTM recurrent units to propose a generic framework for activity recognition. The results showed that a deep architecture based on the combination of convolutional and LSTM recurrent layers outperformed previous results and could enhance the ability to recognize similar activities. These works take the sensor data directly as the input of the model and then use the deep learning approaches to extract the features and perform classification.

Different data preprocessing methods are proposed to eliminate the noise of the sensor data and to increase the classification accuracy. Panwar et al. [26] used a CNN to recognize elementary arm movements. Considering the noise and artifacts of the sensor data, different data preprocessing techniques were used. Alsheikh et al. [27] used the spectrogram representation as the input for the deep activity recognition model, as it can reduce computational complexity by reducing the data dimension. The authors also proposed a new model based on deep belief networks, which includes two stages: an unsupervised pre-training step and a supervised fine-tuning step. In [28], the authors proposed user-independent and user-dependent models. The data preprocessing stages of the two models include the following steps: converting sample rate, calculating composite value, using window function and Fourier transform to calculate power spectrum. The author in [29] employed Extensible receptor Stream Processing (ESP) to preprocess sensor data.

HAR usually depends on data from diversified sensors. How to fuse the data of these sensors is also worth exploring. Münzner et al. [30] used a CNN and proposed four multimodal sensor fusion approaches, which are early fusion, sensor-based late fusion, channel-based late fusion and shared filters hybrid fusion. Radu et al. [31] used the restricted Boltzmann machine (RMB) and proposed a multimodal RBM (MM-RBM) which is a variant of a RBM. The experimental results showed that the performance of the MM-RBM outperforms C4.5, SVM and Random Forest. This outcome means that the deep learning can better extract the discriminative information from the multiple sensor data than can the shallow methods.

When multiple sensors are used for activity recognition, different data fusion methods lead to different results, indicating that there is a certain relation between multiple sensors. Since there are relations between sensors, are there hidden relations between axes? The x-axis, y-axis and z-axis directions of the triaxial accelerometer are based on the screen of the mobile phone. Different positions of the mobile phone will cause the direction of the axis to change. In [32] and [33], the authors both eliminate the possible rotational interference by synthesizing the acceleration without considering the direction of the axis, which partly takes into account the relations between the three axes. By converting the time-series signal of the sensor into an active image containing the hidden relations between axes, the recognition accuracy of the model is obviously improved [14]. However, these methods have some limitations because they are largely dependent on human domain knowledge. In this paper, we propose a data fusion-based CNN approach, which allows the network to acquire hidden relations between axes through training.

3. Methodology

The principle of our approach is shown in Fig. 1. To make use of the relations between the three axes of an accelerometer, we use convolution to fuse the triaxial data.

The sensor data have two formats. One is a multichannel format where each axis is a single channel [12], [27],

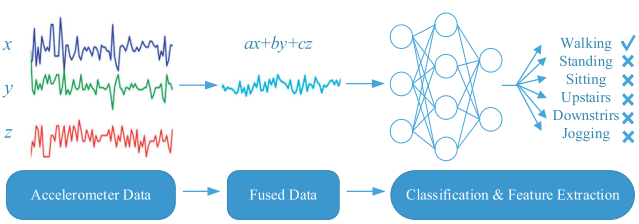


Fig. 1 Overall flow of our data fusion-based convolutional neural network approach; $ax + by + cz$ refers to the linear combination of the triaxial acceleration data.

Table 1 Input description of the model established by the baseline algorithm.

Model	Data description	Data format
MC-BA	Raw sensor data	(1, 200, 3)
SC-BA	Raw sensor data	(3, 200, 1)

[30], and the other is a single-channel format where the $x-$, $y-$, and z-axis data are arranged in rows [30], [35], [36]. For the two formats, we designed a single-channel data fusion method and a multichannel data fusion method. For each data format of sensor, there is a convolution kernel. We only want a linear combination of data, so we do not use the activation function when performing the data fusion. If an activation function such as the rectified linear unit (ReLU) is used, part of the data will become zero, meaning that the fusion result will be partially lost. Baseline algorithms were chosen for comparison and analysis.

3.1 Baseline Algorithm

For a baseline algorithm, we used the CNN to compare with our approach. The CNN is a supervised learning algorithm that relies on the labels of the data. A generic CNN structure is composed of an input layer, convolutional layers, pooling layers, fully connected layers, normalization layers and an output layer. The network can automatically extract features from the sensor data without manually designing the features.

For the two data formats of sensor, we established two baseline algorithms accordingly. One is a MC-BA (multichannel baseline algorithm), the other is a SC-BA (single-channel baseline algorithm). Detailed information is shown in Table 1.

3.2 Deep Learning Architecture

Data preprocessing - We use normalization techniques to preprocess the raw sensor data. Z-normalization (zNorm) is a widely used normalization technique that uses the mean and standard deviation of the raw data. The normalized output is calculated by

$$x_{ij}{}' = \frac{x_{ij} - mean(x_i)}{std(x_i)} \tag{1}$$

The x_{ij} is jth element on channel i. $mean(x_i)$ and $std(x_i)$ refer to the mean and the standard deviation over all data for channel i.

Convolutional Layer – Each layer in a CNN consists of many neurons. A convolutional layer is used to extract features with local connectivity and shared weights characteristics.

In the convolutional layer, a convolution operation is performed on the input data, and its output forms the next layer of the network. Performing a dot product operation on the convolution kernel and input local area produces an output value that is one neuron in the next layer. The input data have three dimensions: width, height and depth. When the depth of the input data is 1, we can obtain a 2-dimensional output by moving the convolution kernel according to the stride and performing dot product operation on the width and height of the input data. This output is called a feature map. When the depth of the input data is not 1, the number of 2-dimensional outputs is the same as the depth, and these 2-dimensional outputs are added to generate a feature map. The feature maps generated by different convolution kernels are stacked along the depth dimension to form the entire output of the convolutional layer. The output of the convolutional layer can be calculated by

$$x_j^{l+1,k} = \sum_{m=0}^{M-1} w_m^{l+1,k} \cdot x_{j+m-1}^{l,k} + b^{l+1,k} \tag{2}$$

$$x_j^{l+2,k} = \sigma(x_j^{l+1,k}) \tag{3}$$

M is the length of the convolutional kernel. $w_m^{l+1,k}$ is the value at the position m of the convolutional kernel. $b^{l+1,k}$ is the bias term for the kth feature map in the $(l+1)$th layer. The output of Eq. (2) represents the jth neuron of the kth feature map in the $(l+1)$th layer. If we want to add some nonlinearity to the network, we can use Eq. (3) in which σ is the activation function.

Pooling Layer – Unlike the convolutional layer that explores the local correlation of the previous layer of features, the pooling layer combines the semantically similar features to generate a single neuron in the next layer [22]. The average pooling calculates the mean value of each local area and outputs the mean value to the next layer. The max pooling selects the maximum value of each local area and passes the output of this area to the next layer.

Softmax Layer – In a classification task, we can use the softmax layer as the last layer of the model to obtain a final classification result. The final classification result can be calculated by

$$f(x) = \arg\max_c p(y = c|x) = \arg\max_c \frac{e^{XW_j}}{\sum_n^N e^{XW_n}} \tag{4}$$

Learning Rate Schedule – Learning rate is an important hyperparameter that affects the final classification result. If the learning rate is too large, the loss curve may fluctuate or even rise. If the learning rate is too small, it will lead to an increase in the number of iterations. In this paper, we use a learning rate that dynamically changes according to the number of iterations. Setting a relatively large learning rate at the beginning of the training can quickly converge the model to an ideal accuracy, but there will be large fluctuations in the loss and accuracy curves. Therefore, setting a relatively small learning rate in the middle stage allows the model to continue learning while fluctuating less. In the late stages of training, the learning rate will continue to decrease, and fluctuations can be further reduced. This method can reduce the overfitting and the number of iterations to some extent.

Table 2 Data description of the multichannel model.

Model	Data description	Data format
MCM-FRD	Raw sensor data and fused data	$(1, 200, n+3)$
MCM-FD	Fused data	$(1, 200, n)$

Algorithm 1 Multichannel Data Fusion Algorithm.

Input: Labeled dataset $\{(x_i, y_i, z_i), a_i\}$, unlabeled dataset $\{(x_i, y_i, z_i)\}$.
Output: Activity Labels of unlabeled dataset

```
repeat
Forward Propagation:
for each labeled data (x, y, z) do
    Use (2) to perform data fusion operation on the input data
    Concatenate fused data and input data along channel direction
    Use (2) (3) to perform convolution operation
    Perform max pooling operation on the output of convolutional layer
end for
Use (4) to perform classification
Back propagation:
Use stochastic gradient descent algorithm to perform back propagation
until weight convergences;
for each unlabeled data (x, y, z) do
    Use the trained network to predict activity labels
end for
```

3.3 Multichannel Data Fusion Method

For the multichannel format of sensor data, the number of channels is 3, and the number of rows is 1. The multichannel data fusion method adopts a 1×1 convolution to fuse the triaxial data and then uses a CNN for the feature extraction and classification. To explore whether the fused data can completely replace the raw sensor data, we established two models: the model that concatenates the raw sensor data with the fused data and the model that only uses the fused data. Table 2 shows a brief description of the two models. Here, we set the number of fusion data to be the same as the number of filters in the data fusion operation.

For the MCM-FRD (multichannel model based on fused data and raw data), we use a 1×1 convolution to fuse the x-, y- and z-axis data, and then we concatenate the fused data with the raw x-, y- and z-axis sensor data along the channel direction so that the data are changed from 3 channels to $n + 3$ channels. n is the number of filters in 1×1 convolution, and 3 is the number of channels of raw sensor data. Next, we perform a convolution operation with a 1×7 convolution kernel and pooling operation. Finally, the softmax classifier is used to obtain the output of the model.

The pseudo-code for the MCM-FRD is described in Algorithm 1. The schematic illustration of the MCM-FRD is shown by the solid line in Fig. 2.

For the MCM-FD (multichannel model based on fused data), we only use the fused data generated by using a 1×1 convolution, and does not concatenate the raw sensor data with the fused data. The format of the data is $(1, 200, n)$, where n represents the number of filters in a 1×1 convolution. Then, we perform the convolution operations and pooling operations on the fused data directly. Finally, the softmax layer is used for the activities classification. The schematic illustration of the MCM-FD is shown by the dashed line in Fig. 2.

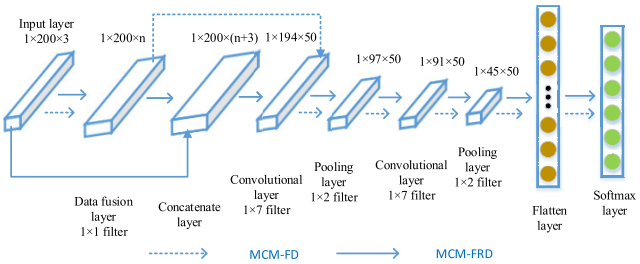


Fig. 2 Schematic illustration of the multichannel model.

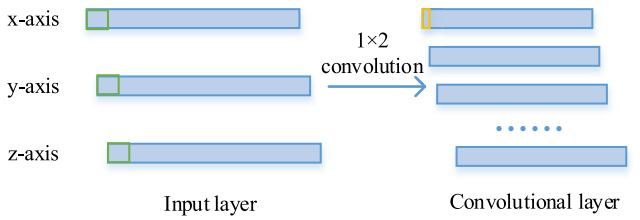


Fig. 3 Input layer and convolutional layer of MC-BA.

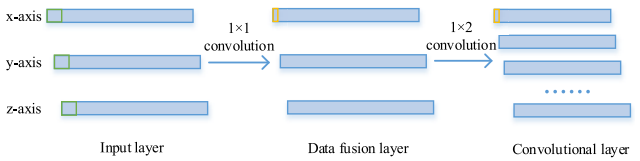


Fig. 4 Input layer, data fusion layer and convolutional layer of MCM-FD.

Compared with the baseline algorithm, our MCM-FD is equivalent to adding a data fusion layer before the first convolutional layer in the baseline algorithm. Data fusion is achieved by convolution operations. Let's compare the difference between the values of first convolutional layer in MC-BA and MCM-FD. In this section, we do not consider bias terms in order to simplify the equation. In Figs. 3 and 4, the green box represents the convolution kernel, and the orange box represents the output value of the convolution operation.

For convenience of comparison, the convolution kernel size is set to 1×2. We use the input value as $[[x_1, x_2], [y_1, y_2], [z_1, z_2]]$, the corresponding convolution kernel uses $[[k_1, k_2], [k_3, k_4], [k_5, k_6]]$ indicated. The output value of Fig. 3 can be calculated by

$$\text{output} = \sigma(x_1k_1 + x_2k_2 + y_1k_3 + y_2k_4 + z_1k_5 + z_2k_6) \quad (5)$$

In order to obtain the fused data, we use 1×1 convolution to fuse the three channels of the input data. Assuming that the number of 1×1 convolution kernel is three, the data fusion layer has three channels. The three convolution kernels are represented by $[[a_1], [b_1], [c_1]]$, $[[a_2], [b_2], [c_2]]$, $[[a_3], [b_3], [c_3]]$, respectively. The 1×2 convolution kernel is represented by

$[[k_1, k_2], [k_3, k_4], [k_5, k_6]]$. The value represented by the orange box of the data fusion layer in the Fig. 4 can be calculated as follows

$$\text{DF\_Output} = \sigma(a_1 x_1 + b_1 y_1 + c_1 z_1) \tag{6}$$

The output value represented by the orange box in the convolutional layer can be calculated by Eq. (7). Comparing Eqs. (5) and (7), it can be see that the output of Eq. (7) has more possibilities, which means that the CNN can extract more useful information than the baseline algorithm after adding the multichannel data fusion method.

$$\begin{aligned}\text{MCM-FD\_Output} = \sigma((&a_1 x_1 + b_1 y_1 + c_1 z_1)k_1 \\ &+ (a_1 x_2 + b_1 y_2 + c_1 z_2)k_2 \\ &+ (a_2 x_1 + b_2 y_1 + c_2 z_1)k_3 \\ &+ (a_2 x_2 + b_2 y_2 + c_2 z_2)k_4 \\ &+ (a_3 x_1 + b_3 y_1 + c_3 z_1)k_5 \\ &+ (a_3 x_2 + b_3 y_2 + c_3 z_2)k_6)\end{aligned} \tag{7}$$

3.4 Single-Channel Data Fusion Method

For the single-channel format of the sensor data, the number of channels is 1, and the number of rows is 3. Due to the change in the sensor data format, we change the convolution kernel size from 1×1 to 3×1 when performing the data fusion operations. The single-channel data fusion method adopts a 3×1 convolution to fuse the triaxial data. We establish a model to make use of the relations between axes. The 2-dimensional convolution kernel of the shape $n \times 1$ was used to extract the features, where n is equal to the number of rows of the data.

For the SCM-FD (single-channel model based on fused data), we adopt a 3×1 convolution to fuse the x-, y- and z-axis data, which can be called a data fusion operation. A convolution operation with a convolution kernel size of 3×1 is performed on the raw sensor data; then, a fused data of shape $(1, 200, 1)$ can be obtained. When multiple fused data are required, we repeat the convolution operation with a convolution kernel size of 3×1 on the raw sensor data. Next, we concatenate the fused data by row to obtain the input data of the next layer. If the number of fused data is n, we perform the 3×1 convolution operation n times. The pseudo-code for the SCM-FD is described in Algorithm 2. Here, we set the number of fused data to be the same as the number of convolutional layers that perform the data fusion operation.

The schematic illustration for the SCM-FD is shown in Fig. 5. We first perform data fusion operations to obtain fused data. The number of fused data is n, and each shape is $(1, 200, 1)$. Then, we concatenate each fused data by row to form the entire fused data, and its number of channels is 1. The shape of entire fused data is $(n, 200, 1)$. CNN with a 1-dimensional kernel only can capture local dependency, while the CNN with a 2-dimensional kernel can capture local dependency and spatial dependency [34]. Hence,

Algorithm 2 Single-Channel Data Fusion Algorithm.

```
Notations:
N_c the number of fused data
Input: Labeled dataset {(x_i, y_i, z_i), a_i}, unlabeled dataset {(x_i, y_i, z_i)}
Output: Activity Labels of unlabeled dataset
repeat
Forward Propagation:
i ← 1
for each labeled data (x, y, z) do
    while i < N_c do
        Use (2) to perform data fusion operation on input data
        i = i + 1
    end while
    Concatenate fused data by row
    Use (2), (3) to perform convolution operation on fused data
    Perform max pooling operation with the output of convolutional layer
end for
Use (4) to perform classification
Back propagation:
Use stochastic gradient descent algorithm to perform back propagation until weight convergences;
for each unlabeled data (x, y, z) do
    Use the trained network to predict activity labels
end for
```

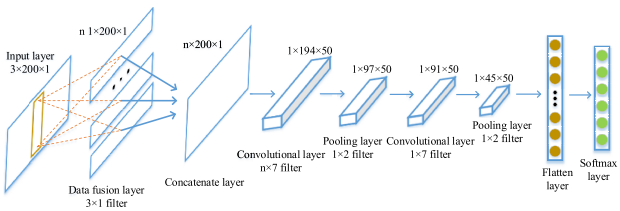


Fig. 5 Schematic illustration of the single-channel model.

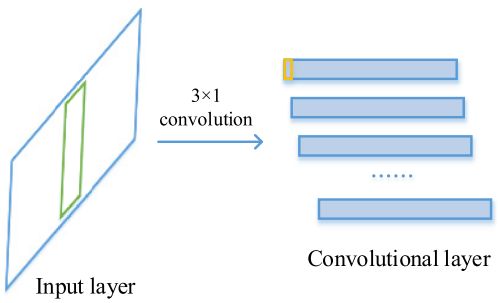


Fig. 6 Input layer and convolutional layer of SC-BA.

the convolution kernel size of the next layer is set to $n \times 7$. Next, we perform convolution operations whose convolution kernel size is 1×7 and the pooling operations. The last layer of the model is a softmax layer, and the output of the model is obtained by the softmax layer.

Let's compare the difference between the values of first convolutional layer in SC-BA and SCM-FD. We do not consider bias terms in order to simplify the equation. In Figs. 6 and 7, the green box represents the convolution kernel, and the orange box represents the output value of the convolution operation.

As for Fig. 6, we use the input value as $[[x], [y], [z]]$.

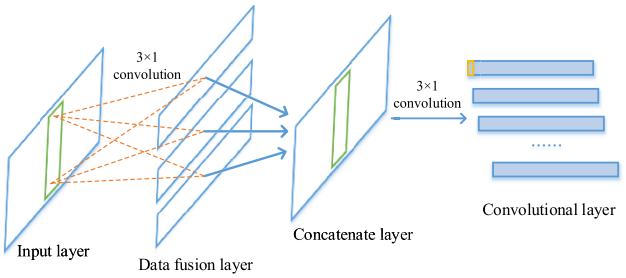


Fig. 7 Input layer, data fusion layer, concatenate layer and convolutional layer of SCM-FD.

The corresponding convolution kernel uses $[[f_1], [f_2], [f_3]]$ indicated. The output value can be calculated by

$$\text{Output} = \sigma(xf_1 + yf_2 + zf_3) \tag{8}$$

As for Fig. 7, 3×1 convolution is used to fuse triaxial data. Each execution of a 3×1 convolution will generate 1-dimensional data, and execution of three times will generate three 1-dimensional data. The corresponding convolution kernel uses $[[a_1], [b_1], [c_1]], [[a_2], [b_2], [c_2]], [[a_3], [b_3], [c_3]]$ indicated. The output value of data fusion layer can be calculated by Eq. (9), where $i = 1, 2, 3$.

$$\text{DF\_Output} = \sigma(xa_i + yb_i + zc_i) \tag{9}$$

Concatenate the data in the data fusion layer by row. Then do the convolution operation to get the value represented by the orange box. This value can be obtained by Eq. (10).

$$\begin{aligned}\text{SCM-FD\_Output} = \sigma((&xa_1 + yb_1 + zc_1)f_1 \\ &+ (xa_2 + yb_2 + zc_2)f_2 \\ &+ (xa_3 + yb_3 + zc_3)f_3)\end{aligned} \tag{10}$$

Comparing Eqs. (8) and (10), it can be seen that x in Eq. (8) corresponds to $xa + yb + zc$ in Eq. (10). That is, the raw sensor data corresponds to the fused data containing the hidden relations between the axes. Therefore, the output of Eq. (10) has more possibilities, which means that the CNN can extract more useful information than the baseline algorithm after adding the single-channel data fusion method.

4. Experiments

For experiments using the WISDM dataset [37], a stochastic gradient descent is used to train the deep activity recognition models. A 10-fold cross-validation (CV) method is used to evaluate our approach. If the number of iterations is less than 1000, the learning rate is 0.005. If the number of iterations is greater than 3000, the learning rate is 0.0005. Otherwise, the learning rate is 0.001. The specific parameters of the experiments are shown in Table 3.

For experiments using the UCI dataset [38], we take the average of 10 experiments as the final result. If the number of iterations is less than 180, the learning rate is 0.0005. Otherwise, the learning rate is 0.0001. The specific parameters of the experiments are shown in Table 4.

Table 3 The detailed information of parameter setting of WISDM experiments.

Parameter	Value
The size of input vector	200
The number of convolutional layers	2
Number of filter	50
Pooling size	1×2
The probability of dropout	0.9
Momentum	0.9
Weight decay	1e-6
L2 norm regularizations	0.0015
The size of minibatches	128
Maximum epochs	4000
Activation function	ReLU (rectified linear units)

Table 4 The detailed information of parameter setting of UCI experiments.

Parameter	Value
The size of input vector	200
The number of convolutional layers	2
Number of filter	100
Pooling size	1×2
The probability of dropout	0.9
L2 norm regularizations	0.01
The size of minibatches	32
Maximum epochs	200
Optimizer	Adam
Activation function	ReLU (rectified linear units)

4.1 Dataset Description

In this paper, we use the WISDM Actitracker dataset and UCI dataset to evaluate the efficiency of our approach. A total of 1,098,213 samples for 29 users are included in WISDM dataset. This dataset contains 6 types of human activities, which are walking, jogging, upstairs, downstairs, sitting and standing. The sampling rate is 20 Hz. We divide the samples by sliding a window of size 200 on the data. Figure 8 shows the acceleration data for each activity after regularization, which is used to train the model. The UCI dataset contains triaxial acceleration, estimated body acceleration and triaxial angular velocity. Since the goal of this paper is to explore the hidden relations between axes inside the sensor, the data from a single sensor is used as the input of the experiment. The triaxial acceleration in the UCI dataset is used to verify the proposed methods.

4.2 Influence of the Number of Fused Data

In this section, to fairly compare the accuracy of the model, the number of filters is set to 50, and all experiments were performed under this condition. The number of fused data is an important factor which will affect the final classification accuracy. If the number of fused data is too small, it may result in not containing enough information. In contrast, if the number is too large, it may cause part of the data to be

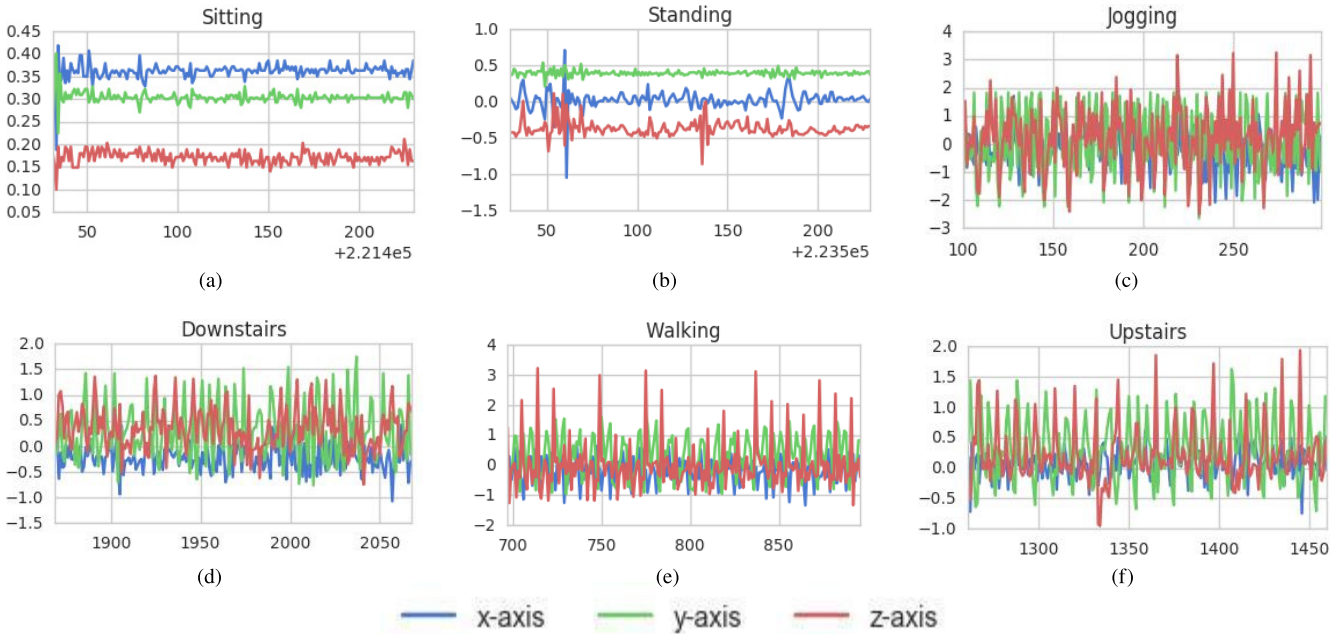


Fig. 8 Triaxial acceleration data for each activity after normalization.

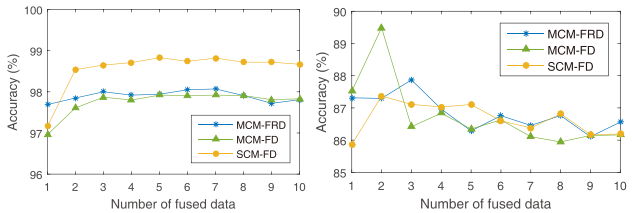


Fig. 9 Comparison of the classification accuracy of model with different numbers of fused data.

redundant and increase the number of parameters, even resulting in overfitting or a decrease in accuracy.

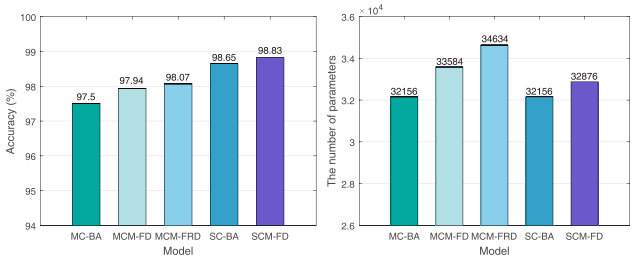
Figure 9 shows the effect of increasing the number of fused data on the accuracy of the MCM-FRD, MCM-FD and SCM-FD. It can be seen from Fig. 9 (a) that the curve of the SCM-FD is above the curves of other two models except when the number of the fused data is 1. After the MCM-FD and SCM-FD generate the fused data, they only extract the features of the fused data instead of the fused data and the raw sensor data. Therefore, they cannot extract sufficient information to compare with the MCM-FRD when the number of fused data is 1.

When the data are in a multichannel format, we observe the two curves of the MCM-FRD and MCM-FD. We found that the two curves as a whole show a trend of rising first and then decreasing. The accuracy of the MCM-FRD is lower than that of the MCM-FD, except that the number of the fused data is 8, 9 and 10. In the other cases, the curve of the MCM-FRD is either higher than the curve of the MCM-FD or intersects it. When the number of fused data is 7, the accuracy of the MCM-FRD reaches a maximum of 98.07%. When the number of the fused data is 5, the accuracy of the MCM-FD reaches a maximum of 97.92%. The highest accuracy of the MCM-FRD exceeds the highest accuracy of the MCM-FD. It is proved that the MCM-FRD with raw data outperforms the MCM-FD without raw data.

When the number is 3, 6 or 7, the accuracy of the MCM-FRD is 98%, 98.05% and 98.07%, respectively. In fact, there is not much difference in accuracy. If we only consider the accuracy, we can choose the number of fused data to be 7. If we consider both accuracy and the number of parameters, the number of fused data can be chosen to be 3. In this section, we mainly consider the accuracy. Therefore, we choose the number of fused data for the MCM-FRD to be 7.

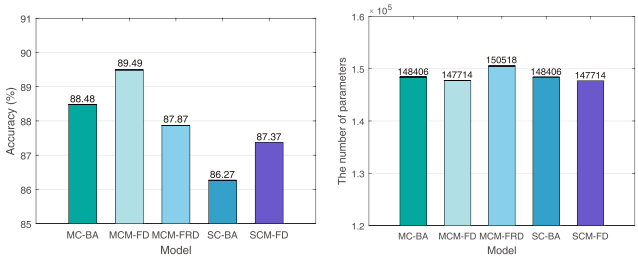
For the single-channel format of the sensor data, when the number of the fused data is 1, the curve reaches the lowest point and its value is 97.16%, which means that the fused data do not contain sufficient useful information. When the number of the fused data is between 2 and 5, the accuracy of the model still slightly increases from 98.54% to 98.83%. When the number of the fused data is between 4 and 9, the accuracy of the model fluctuates slightly. From Fig. 9 (a) we can see that when the number of the fused data is 5, the SCM-FD reaches the highest accuracy of 98.83%. However, a continuous increase in the number of the fused data will not cause an increase in accuracy but will increase the number of parameters, which may lead to overfitting.

Figure 9 (b) shows the experimental results when increasing the number of fused data on the UCI dataset. It can be seen from Fig. 9 (b) that the three curves on the whole show at first a ascending trend and then a descending one. There are fluctuations during the decreasing process. When the number of fused data is less than 3, the curve of



(a) Accuracy (b) Parameters

Fig. 10 Comparison of the accuracy and the number of parameters of five models on the WISDM dataset.



(a) Accuracy (b) Parameters

Fig. 11 Comparison of the accuracy and the number of parameters of five models on the UCI dataset.

MCM-FD is above the curve of MCM-FRD and SCM-FD. When the number of fused data is 2, the accuracy of MCM-FD reaches a maximum of 89.49%. The curve of MCM-FRD reaches the highest point when the number of fused data is 3. The accuracy of the highest point is 87.87%. For the single-channel format of the sensor data, when the number of the fused data is 2, the curve reaches the highest point and its value is 87.37%. By comparing the highest points of the three curves, it can be found that MCM-FD works best on the UCI dataset.

4.3 Comparison with Baseline Algorithm

In this section, the performance of the model is considered from three aspects: accuracy, the number of parameters and the dependency on the amount of training data. We have two purposes: (1) Determine whether the data fusion method is effective by comparing the accuracy of the five models; and (2) Determine which data fusion method works better by considering the dependency on the amount of training data and the number of parameters.

Figure 10 (a) shows the comparison of the accuracy of the five models on the WISDM dataset. The MC-BA and SC-BA are baseline algorithms; the former's input data are expressed as a multichannel format, and the latter's input data are expressed as a single-channel format. The MCM-FRD and the MCM-FD use a multichannel data fusion method, and the SCM-FD uses a single-channel data fusion method. First, we compare the MC-BA and SC-BA with different input data formats. The MC-BA uses a 1-dimensional convolution kernel, while the SC-BA uses a 2-dimensional convolution kernel. We find that the accuracy of the SC-BA is 1.15 percentage points higher than that of the MC-BA, indicating that the effect of the SC-BA is better than that of the MC-BA.

When the data are in a multichannel format, the accuracy of the MCM-FRD and the MCM-FD are 0.57 and 0.44 percentage points higher than that of the MC-BA, respectively. When the data are in a single-channel format, the accuracy of the SCM-FD using a single-channel data fusion method achieves 98.83%, which is 0.18 percentage points higher than that of the SC-BA.

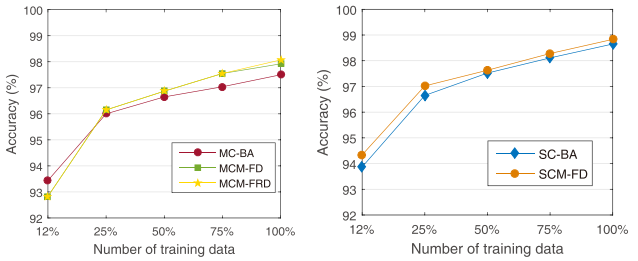
According to the analysis above, we know that both the single-channel data fusion method and the multichannel data fusion method can increase the accuracy of the model. Both data fusion methods are effective.

The number of parameters in the five models is shown in Fig. 10 (b). We can see from Fig. 10 (b) that the MC-BA and the SC-BA have the same parameters, even though they have different numbers of rows and channels. The number of parameters is 32,156. Considering Fig. 10 (a) and Fig. 10 (b) comprehensively, we know that the MCM-FRD, MCM-FD and SCM-FD have higher accuracy and more parameters than do the MC-BA and SC-BA.

When the data are in multichannel format, the accuracy of the MC-BA is 97.50% and the number of parameters is 32,156. We compare the MCM-FRD and MCM-FD with the MC-BA. The number of parameters in the MCM-FRD and MCM-FD are 2,478 and 1,428 more than that in the MC-BA, respectively. The MCM-FRD has the highest accuracy of 98.07%, but it also has the largest number of parameters, which is 34,634. When the data are in a single-channel format, we compare the SC-BA and SCM-FD. We can find that the SCM-FD has 720 more parameters than the SC-BA does but that the accuracy has increased by 0.18 percentage points. For the MCM-FRD, MCM-FD and SCM-FD, the SCM-FD has the least number of parameters but the highest accuracy, and the accuracy of the SCM-FD is up to 98.83%.

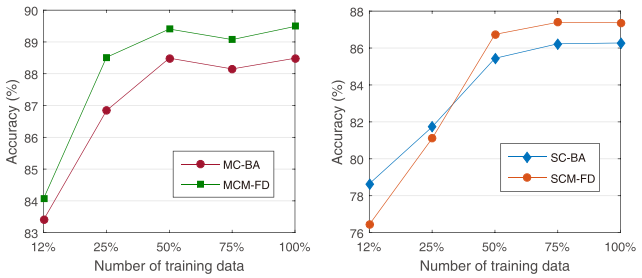
Figure 11 (a) shows the comparison of the accuracy of the five models on the UCI dataset. When the data are in multichannel format, the accuracy of MCM-FD reaches 89.49%, which is 1.01 percentage points higher than that of MC-BA. The accuracy of MCM-FRD is lower than that of MC-BA. However, the accuracy of MCM-FRD is the highest on the WISDM dataset, indicating that the effects of MCM-FD and MCM-FRD are related to the dataset. When the data are in a single-channel format, the accuracy of SCM-FD is 1.1 percentage points higher than the accuracy of SC-BA.

Figure 11 (b) shows the number of parameters of the five models on the UCI dataset. The number of parameters in MC-BA and SC-BA is kept the same as 148,406. The number of parameters in MCM-FD is 692 less than that in MC-BA. The number of parameters in SCM-FD is the same as that in MCM-FD. Considering Fig. 11 (a) and Fig. 11 (b) comprehensively, it can be seen that MCM-FD and SCM-FD have fewer parameters and higher accuracy than MC-BA and SC-BA.



(a) Multichannel (b) Single Channel

Fig. 12 Comparison of the accuracy of five models under different amounts of training data on the WISDM dataset.



(a) Multichannel (b) Single Channel

Fig. 13 Comparison of the accuracy of four models under different amounts of training data on the UCI dataset.

The effect of the different amounts of training data on model accuracy is shown in Fig. 12. In Fig. 12 (a), when the amount of training data is greater than or equal to 25%, the accuracy of the model using a multichannel data fusion method is higher than that of the baseline algorithm. In Fig. 12 (b), the curve of the SCM-FD stays above the curve of the SC-BA. The result shows that the data fusion method is able to improve the accuracy. Overall, the SCM-FD achieved the highest accuracy, regardless of the amount of training data.

Based on the above analysis of single-channel and multichannel scenarios on the WISDM dataset, we know that the accuracy of the SCM-FD is the highest, but the added parameters are the fewest in the three models. With only a few parameters added, a high accuracy can be achieved, which proves that our single-channel data fusion method works better than the multichannel data fusion method do.

Since the accuracy of MCM-FRD is lower than that of MC-BA, only four models are compared in this section. Figure 13 shows the accuracy of the four models under different numbers of training data. In Fig. 13 (a), it can be clearly seen that the curve of the MCM-FD is above the curve of the MC-BA. In Fig. 13 (b), when the number of training data is between 25% and 50%, the accuracy of SCM-FD exceeds that of SC-BA. As the amount of training data increases, the SCM-FD curve is above the SC-BA curve.

Based on the above analysis of single-channel and multichannel scenarios on the UCI dataset, we know that the accuracy of the MCM-FD is the highest. The number of parameters decreases but the accuracy increases, which proves that the proposed multichannel data fusion method and the single-channel data fusion method are effective.

Table 5 Comparison of our proposed method against existing methods.

Method ID	Reference	Method	Accuracy (%)
1	Kwapisz, Weiss, and Moore [37]	C4.5	85.10
2	Kwapisz, Weiss, and Moore [37]	Logistic regression	78.10
3	Kwapisz, Weiss, and Moore [37]	MLPs	91.70
4	Catal et al. [39]	Ensemble learning	94.30
5	Alsheikh et al. [27]	Deep learning	98.23
6	Ravì et al. [40]	Shallow-features-only	97.40
7	Ravì et al. [41]	Deep learning	98.20
8	Ravì et al. [40]	Deep learning	98.60
9	Ours	Single-channel data fusionbased CNN	98.83

4.4 Comparison with Existing Methods

As shown in Sect. 4.3, the highest accuracy among the five models on the WISDM dataset occurs with the SCM-FD. To prove the effectiveness of our proposed approach, we compare it with some existing methods. The results are presented in Table 5. The author in [39] combined the J48, logistic regression and multi-layer perceptron algorithms used in [14] to propose an ensemble of classifiers approach. Experiments show that the ensemble of classifiers approach achieves better performance than that of the above three standalone algorithms. The result in [27] proved that the deep model can effectively improve the accuracy and does not require hand-engineering of the features. The author in [40] found that the features derived from a deep learning method are sometimes less discriminatory than those of shallow features, which may be due to resource constraints and simple design methods. Therefore, they combined the shallow features and deep learnt features. In comparison with the model using only features extracted by deep learning [41], the accuracy was improved to 98.6%. The SCM-FD does not manually extract features with human domain knowledge but uses a CNN for feature extraction and classification, which is different from the methods above.

From Table 5, it is evident that the single-channel data fusion-based CNN approach improves the accuracy. Compared with the existing methods, our approach improves the accuracy by 0.23 percentage points. Our approach is relatively uncomplicated but achieves the highest accuracy, which indicates that it is an effective approach for HAR.

Figure 14 is the confusion matrix of the method we proposed. The proposed method is compared with the method of obtaining the highest accuracy in Table 5. Table 6 shows the classification accuracy of each activity under the two methods. The last line of the Table 6 is the difference in accuracy between the two methods, clearly indicating which

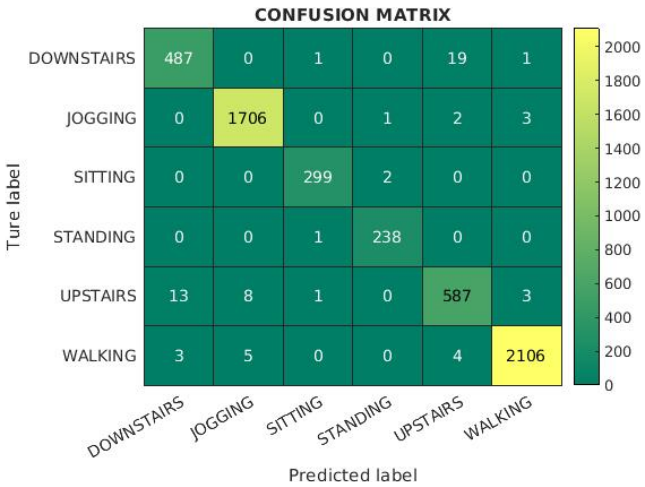


Fig. 14 Confusion matrix of proposed method.

Table 6 Comparison of our proposed method against state-of-the-art method. The last line is the difference of accuracy between the two methods.

Method	Classification accuracy (%)					
	Downstairs	Jogging	Sitting	Standing	Upstairs	Walking
Ravì et al. [40]	94.44	99.64	97.85	98.15	95.52	99.37
ours	95.87	99.65	99.34	99.58	95.92	99.43
	1.43	0.01	1.49	1.43	0.40	0.06

method achieves higher accuracy and which activity improves recognition accuracy more. From Table 6, we can find that the accuracy of the three activities of Downstairs, Sitting and Standing has increased by more than 1.4%, while the accuracy of Jogging and Walking has only slightly improved. As can be seen from Fig. 8, there are many places where the x-, y-, z-axis curves of Jogging and Walking intersect. On the contrary, the x-, y-, z-axis curves of Standing and Sitting intersect only a few places. When the triaxial data differs greatly, the recognition accuracy of the activity is increased more. When the triaxial data differs little, the accuracy of the activity changes very little. We can conclude that the proposed method is more suitable for solving the problem of activity recognition with a large difference in triaxial data.

5. Conclusions

In this paper, we proposed a data fusion-based CNN approach that uses convolution to fuse sensor triaxial data. Two data fusion methods are proposed due to the differences in the data format of sensor. One is a single-channel data fusion method, and the other is a multichannel data fusion method. The proposed approach is designed to make good use of the hidden relations between axes inside the sensor. The results show that our approach is effective and that our approach have an advantage over the baseline algorithm. At the same time, when the single-channel data fusion method is used, the accuracy of our approach even exceeds that of the state-of-the-art methods. The proposed method is more suitable for solving the problem of activity recognition with large difference in triaxial data. For using a single sensor, we have demonstrated that making good use of the hidden relations between axes can improve the accuracy of the model. With multiple sensors, how to use the relations between axes is still a problem worth exploring.

Acknowledgments

This research was supported by National Key R&D Program of China, grant number 2017YFB1302400, and by the Open Project of the Beijing Key Laboratory of Mobile Computing and Pervasive Device.

References

[1] D.J. Cook, N.C. Krishnan, and P. Rashidi, "Activity discovery and activity recognition: A new partnership," IEEE Trans. Cybern., vol.43, no.3, pp.820–828, 2013.
[2] S. Wan, Y. Liang, Y. Zhang, and M. Guizani, "Deep multilayer perceptron classifier for behavior analysis to estimate parkinson's disease severity using smartphones," IEEE Access, vol.6, pp.36825–36833, 2018.
[3] 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, 2012.
[4] Y. Chen and C. Shen, "Performance analysis of smartphone-sensor behavior for human activity recognition," Ieee Access, vol.5, pp.3095–3110, 2017.
[5] N. Hnoohom, S. Mekruksavanich, and A. Jitpattanakul, "Human activity recognition using triaxial acceleration data from smartphone and ensemble learning," 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), pp.408–412, IEEE, 2017.
[6] A. Wang, G. Chen, J. Yang, S. Zhao, and C.-Y. Chang, "A comparative study on human activity recognition using inertial sensors in a smartphone," IEEE Sensors J., vol.16, no.11, pp.4566–4578, 2016.
[7] D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist, and A. Holzinger, "Human activity recognition using recurrent neural networks," International Cross-Domain Conference for Machine Learning and Knowledge Extraction, pp.267–274, Springer, 2017.
[8] L. Zhang, X. Wu, and D. Luo, "Real-time activity recognition on smartphones using deep neural networks," 2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), pp.1236–1242, IEEE, 2015.
[9] L. Zhang, X. Wu, and D. Luo, "Recognizing human activities from raw accelerometer data using deep neural networks," 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp.865–870, IEEE, 2015.
[10] H. Gjoreski, J. Bizjak, M. Gjoreski, and M. Gams, "Comparing deep and classical machine learning methods for human activity recognition using wrist accelerometer," Proc. IJCAI 2016 Workshop on Deep Learning for Artificial Intelligence, New York, NY, USA, 2016.
[11] J. Yang, M.N. Nguyen, P.P. San, X. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," IJCAI, pp.3995–4001, 2015.
[12] M. Zeng, L.T. Nguyen, B. Yu, O.J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," 2014 6th International Conference

on Mobile Computing, Applications and Services (MobiCASE), pp.197–205, IEEE, 2014.
[13] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," Pattern Recognition Letters, 2018.
[14] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," Proc. 23rd ACM international conference on Multimedia, pp.1307–1310, Acm, 2015.
[15] S. Wang, J. Yang, N. Chen, X. Chen, and Q. Zhang, "Human activity recognition with user-free accelerometers in the sensor networks," 2005 International Conference on Neural Networks and Brain, pp.1212–1217, IEEE, 2005.
[16] L. Bao and S.S. Intille, "Activity recognition from user-annotated acceleration data," International conference on pervasive computing, pp.1–17, Springer, 2004.
[17] N. Ravi, N. Dandekar, P. Mysore, and M.L. Littman, "Activity recognition from accelerometer data," AAAI, pp.1541–1546, 2005.
[18] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," Pattern recognition letters, vol.29, no.16, pp.2213–2220, 2008.
[19] Z.-Y. He and L.-W. Jin, "Activity recognition from acceleration data using ar model representation and svm," 2008 international conference on machine learning and cybernetics, pp.2245–2250, IEEE, 2008.
[20] T. Gu, L. Wang, Z. Wu, X. Tao, and J. Lu, "A pattern mining approach to sensor-based human activity recognition," IEEE Trans. Knowl. Data Eng., vol.23, no.9, pp.1359–1372, 2010.
[21] Y. Lu, Y. Wei, L. Liu, J. Zhong, L. Sun, and Y. Liu, "Towards unsupervised physical activity recognition using smartphone accelerometers," Multimedia Tools and Applications, vol.76, no.8, pp.10701–10719, 2017.
[22] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol.521, no.7553, pp.436–444, 2015.
[23] C.A. Ronao and S.B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," Expert Systems with Applications, vol.59, pp.235–244, 2016.
[24] N.Y. Hammerla, S. Halloran, and T. Ploetz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," arXiv preprint arXiv:1604.08880, 2016.
[25] 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.
[26] M. Panwar, S.R. Dyuthi, K.C. Prakash, D. Biswas, A. Acharyya, K. Maharatna, A. Gautam, and G.R. Naik, "Cnn based approach for activity recognition using a wrist-worn accelerometer," 2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp.2438–2441, IEEE, 2017.
[27] M.A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.P. Tan, "Deep activity recognition models with triaxial accelerometers," AAAI Workshop: Artificial Intelligence Applied to Assistive Technologies and Smart Environments, 2016.
[28] S. Matsui, N. Inoue, Y. Akagi, G. Nagino, and K. Shinoda, "User adaptation of convolutional neural network for human activity recognition," 2017 25th European Signal Processing Conference (EUSIPCO), pp.753–757, IEEE, 2017.
[29] L. Liu, Y. Peng, S. Wang, M. Liu, and Z. Huang, "Complex activity recognition using time series pattern dictionary learned from ubiquitous sensors," Information Sciences, vol.340, pp.41–57, 2016.
[30] S. Münzner, P. Schmidt, A. Reiss, M. Hanselmann, R. Stiefelhagen, and R. Dürichen, "Cnn-based sensor fusion techniques for multimodal human activity recognition," Proc. 2017 ACM International Symposium on Wearable Computers, pp.158–165, ACM, 2017.
[31] V. Radu, N.D. Lane, S. Bhattacharya, C. Mascolo, M.K. Marina, and F. Kawsar, "Towards multimodal deep learning for activity recognition on mobile devices," Proc. 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, pp.185–188, ACM, 2016.
[32] S.-M. Lee, S.M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," 2017 IEEE International Conference on Big Data and Smart Computing (BigComp), pp.131–134, IEEE, 2017.
[33] Y. Li, D. Shi, B. Ding, and D. Liu, "Unsupervised feature learning for human activity recognition susing smartphone sensors," in Mining Intelligence and Knowledge Exploration, pp.99–107, Springer, 2014.
[34] F.J.O. Morales and D. Roggen, "Deep convolutional feature transfer across mobile activity recognition domains, sensor modalities and locations," Proc. 2016 ACM International Symposium on Wearable Computers, pp.92–99, ACM, 2016.
[35] S. Ha, J.-M. Yun, and S. Choi, "Multi-modal convolutional neural networks for activity recognition," 2015 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp.3017–3022, IEEE, 2015.
[36] Y. Chen and Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," 2015 ieee international conference on systems, man, and cybernetics (smc), pp.1488–1492, IEEE, 2015.
[37] J.R. Kwapisz, G.M. Weiss, and S.A. Moore, "Activity recognition using cell phone accelerometers," ACM SigKDD Explorations Newsletter, vol.12, no.2, pp.74–82, 2011.
[38] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J.L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," ESANN, 2013.
[39] C. Catal, S. Tufekci, E. Pirmit, and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition," Applied Soft Computing, vol.37, pp.1018–1022, 2015.
[40] D. Ravì, C. Wong, B. Lo, and G.-Z. Yang, "A deep learning approach to on-node sensor data analytics for mobile or wearable devices," IEEE J. Biomed. Health Inform., vol.21, no.1, pp.56–64, 2017.
[41] D. Ravì, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," 2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp.71–76, IEEE, 2016.

Xinxin Han received the B.E. degree in software engineering from North University of China, Taiyuan, China in 2016. She is currently pursuing her M.E. degree in computer technology from North University of China, China. Her current research interests include human activity recognition and machine learning.

Jian Ye received the Ph.D. degree in computer science and technology from Chinese Academy of Sciences, China in 2010. Since 2008 he has been an associate professor in the Institute of Comoputing Technology, Chinese Academy of Sciences. His research interests include ubiquitous computing, context awareness, data mining and machine learning. He served as a leader of multiple research projects of China. He is a member of the Chinese Computer Federation.

Jia Luo received the bachelor's degree from Shanghai University, China, in 2011 and the master's degree from Waseda University, Japan, in 2015. Afterwards, she received her Ph. D, degree from Université de Toulouse, France, in 2019. Currently, she is a lecturer at College of Economics and Management, Beijing University of Technology, China. Her research interests include shop scheduling, parallel algorithms, pattern recognition and GPU computing.

Haiying Zhou received the Ph.D. degree in control science and engineering from University of Science and Technology Beijing, China in 2015, his M.E. degree in computer application from North University of China, China in 2005, and his B.S. degree in computational mathematics from Shanxi University, China in 1985. Now he is a professor in the School of Data Science, North University of China, Taiyuan, China. His current research interests include cognitive science and image pattern recognition.