IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 June 2022, accepted 12 June 2022, date of publication 21 June 2022, date of current version 27 June 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3185112

A Multichannel CNN-GRU Model for Human Activity Recognition

LIMENG LU[1], **CHUANLIN ZHANG**[2], **KAI CAO**[1], **TAO DENG**[1,2,3], **AND QIANQIAN YANG**[4]
[1]Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education, Northwest Minzu University, Lanzhou 730030, China
[2]School of Mathematics and Computer Science, Northwest Minzu University, Lanzhou 730030, China
[3]Key Laboratory of Streaming Data Computing Technologies and Application, Northwest Minzu University, Lanzhou 730030, China
[4]Faculty of Nursing, Kunming Medical University, Kunming 650000, China

Corresponding author: Tao Deng (dttom@lzu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China, China, under Grant 32060234; in part by the Natural Science Foundation of Gansu Province, China, under Grant 21JR7RA164; and in part by the Fundamental Research Funds for the Central Universities, China, under Grant 31920210084.

ABSTRACT Human activity recognition (HAR) is one of the important research areas in pervasive computing. Among HAR, sensor-based activity recognition refers to acquiring a high-level knowledge about human activities from readings of many low-level sensors. In recent years, although the existing methods of deep learning (DL) have been widely used for sensor-based HAR with some good performance, they still face such challenges as feature extraction and characterization, continuous action segmentation in dealing with time series problems. In this study, a multichannel fusion model is proposed with the idea of dividing. In this proposed architecture, a multichannel convolutional neural network (CNN) is used to enhance the ability to extract features at different scales, and then the fused features are fed into the gated recurrent unit (GRU) for feature labeling and enhanced feature representation, through the learning of temporal relationships. Finally, the multichannel CNN-GRU model is designed using global average pooling (GAP) to connect the feature maps with the final classification. The model performance was conducted on three benchmark datasets of WISDM, UCI-HAR, and PAMAP2 with the accuracy of 96.41%, 96.67%, and 96.25% respectively. The results show that the proposed model demonstrates better activity detection capability than some of the reported results.

INDEX TERMS Human activity recognition, feature extraction, multichannel CNN, GRU.

I. INTRODUCTION

Human activity recognition (HAR) refers to inferring the current action and predicting the following action from a series of observations and analysis of human behavior and the environment [1]. There are two mainstream techniques for HAR: video-based [2] and sensor-based systems [3]. Video-based system classifies video clips containing various types of human actions [4]. This way is very intrusive to the life of the target individual and difficult to ensure his/her privacy. Besides, the quality of the video captured by the camera is also affected by complex environment, such as lighting, background noise, and the target object occlusion [5], leading to performance degradation. Moreover, the recognition of video images faces more difficulties and more expensive costs [6]. Sensor-based HAR extracts the features of human activity details from the raw data of the sensor and recognizes the human activity [7]. Sensors have a wider range of application scenarios such as healthcare, sports, smart home, and human-computer interaction due to their stability, non-intrusive nature, and excellent ability to protect privacy [8]. Smartphones and smartwatches, a range of wearable devices, have inertial sensors such as gyroscope, accelerometer, and magnetometer embedded in them. These increasing computational devices make it possible to collect time series data efficiently and infer details of human activities [9], and serve as very useful monitoring tools in smart homes.

In recent years, sensor-based HAR has become a popular research area, with researchers first using traditional

The associate editor coordinating the review of this manuscript and approving it for publication was Jon Atli Benediktsson.

machine learning (ML) methods for HAR task. The general process of HAR includes [10]: Collecting motion data using sensors, pre-processing the data, action segmentation, extracting features, and action classification. Fig. 1 shows the whole process of HAR task. Traditional ML, including SVM [11], decision trees [12], Bayes [13], and random forest [14], has seen excellent performance in classifying action. However, ML has many limitations and relies heavily on manual feature extraction due to its shallow learning process. Manual feature extraction, such as statistical and frequency domain features, always depends on elaborate features selection of human experience and domain knowledge [15]. Besides, the hand-crafted features can only characterize some simple human activities, but not the complex ones. As a result, shallow ML algorithms find it difficult to adapt to new complex HAR scenarios [16].

DL has achieved automatic feature extraction by end-to-end neural networks, largely reducing the time-consuming and labor-intensive manual extraction of features and simplifying the huge feature engineering. Meanwhile, the features extracted by DL are deep [17], [18]. Currently, DL methods, with higher efficiency and higher classification accuracy, have found wide use in HAR, and become effective methods for HAR. CNN and Recurrent neural network (RNN) are two typical neural networks. CNN evolved from multilayer perceptron and has features such as weight sharing, local connectivity and down-sampling [19], which has excellent performance in the field of computer vision. RNN is a DL neural network used to model sequence data [20], connects neurons which saves the previous input sequence-information to abstractly characterize the whole sequence, and it generates a new sequence in the end. RNN solves the intractable problems of variable-length sequences and long-distance dependencies in sequences that exist in feedforward neural network (FNN), and is widely used in the fields of sequence annotation, image annotation, etc. Long short-term memory (LSTM) [21] and GRU [22], two variants of RNNs, are used to solve the gradient disappearance and gradient explosion problems of RNNs. Compared with LSTM, GRU has one less control gate inside, fewer parameters, and easier training, but can get similar results.

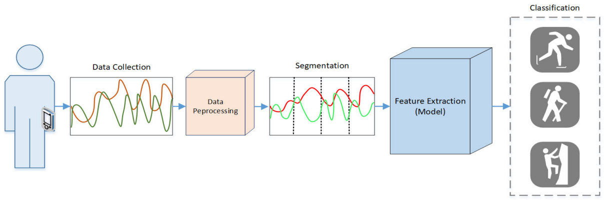


FIGURE 1. **HAR framework.**

Good results of DL networks have also been achieved in the other fields, Liu *et al.* optimized the structure of the GRU network and proposed a new modulation recognition method based on feature extraction and a DL algorithm [23], Hartpence and Kwasinski utilize ensembles to defend against data poisoning attacks attempting to create classification errors [24]. However, HAR using sensor-based DL methods still faces some problems. The first is the extraction, characterization, and classification accuracy of features [25]. Despite the advantage of DL in extracting data features automatically, different network structures have high and low characterization ability of features. Besides, time series of HAR activity has backward and forward relevance, and has difficulties in labeling the sequences. Thus, the performance of feature extraction will directly affect the accuracy of classification; The second is the computational cost, i.e., the number of parameters [26]. These lightweight wearable devices, despite the improvement of chip arithmetic power, still have high requirements on the computational cost of the model, requiring the model to be relatively lightweight and fast to response to real-time data in practical application [27].

For better feature extraction and limitation on computational cost, we propose a three-channel CNN structure for feature extraction for the input samples. The features at different scales extracted from these three channels are connected and fed into GRU for the sequence features. The use of GAP instead of fully connected (FC) layer improves the training speed of the model and get more accurate the classification. Our model achieves higher performance on the benchmark HAR dataset.

The main contributions of the proposed model are:

- To begin, a multichannel convolutional neural network is used for initial feature extraction at different scales before connection and fusion. Next, a GRU neural network is used for sequence labeling to further extract sample features. The mapping between feature maps and final classification using the GAP makes the transformation smoother, ensuring the model is more robust against fitting.
- Compared to other fusion models as well as similar multichannel models, the multichannel CNN-GRU model we proposed has fewer parameters and higher accuracy on WISDM, UCI-HAR, and PAMAP2.

The rest of this paper is structured as follows: The second section introduces the related work of HAR, especially DL methods; The third section contains the methodology used in our proposed multichannel model; the fourth section describes the experiments and the experimental procedure, and the fifth section summarizes the work of this paper and gives the areas for improvement.

II. RELATED WORK

As we know sensors could be easily built into smartphones, smartwatches, and other wearable products. For its high portability and accurate and rapid collection of motion data, sensor-based HAR has many application scenarios [28], such as motion classification, fall detection, human-computer interaction.

A. MACHINE LEARNING METHOD

Researchers have conducted a lot of research work in the area of traditional ML. Initially, researchers used traditional ML methods for the action classification task of HAR with some success. Bao and Intille [29] presented the earliest HAR system that used five wearable dual-axis accelerometers, machine learning classifiers. It could identify 20 categories of activities of daily living, achieving 84% classification accuracy, and this result is fairly good for its relatively large number of activities. However, traditional ML requires manual feature extraction from the raw data, which is a very huge project, and the effectiveness of the extracted features is affected by domain knowledge, which makes it difficult to improve the accuracy of the action classification results.

B. DEEP LEARNING METHOD

In recent years, DL methods have been used in HAR with impressive performance. Zeng et al developed a method based on CNN, which can capture local dependency and scale invariance of a signal. They also proposed a partial weight sharing approach and applied it to accelerometer signals to obtain further improvements [30]. Yang *et al.* [31] further used 1-dimensional convolution (Conv1D) in the same time window to unify and share the weights of time series data from multiple sensors.

Ronao and Cho [10] proposed a model consisting of alternating convolutional and pooling layers, the extracted features are passed to the FC and Softmax layers to predict human activities. CNN and statistical learning were combined to implement a real-time classification framework by Andrey [32]. In [25], the authors designed a cell phone sensor-based HAR model using CNNs. Wang and Liu [33] proposed a hierarchical LSTM approach to identify human activities. CNNs were also used in HAR task to extract temporal features, and achieved significant performance improvements. Bianchi *et al.* [34] proposed a CNN model consisting of four convolutional layers and one FC layer for human activity recognition, which achieved good results on a small training set. A hybrid CNN-LSTM model is proposed in [35] for multi-mode wearable sensor devices. In [36] the authors designed a LSTM-RNN architecture model for HAR.

C. SIMILAR FUSION MODEL

Recently, there have been some new studies using fusion models for sensor-based HAR. Dua *et al.* [37] used CNN and GRU in 3-head module to extract features and FC connection for classification, and they achieved satisfactory results on several datasets, but the overly complex head causes the increase of parameters for the model, and it fails to meet the HAR requirement of the lightweight. Hamza *et al.* [38] and Ronald *et al.* [39] utilized the Inception module from the Inception-Resnet model [40] in the HAR DL model to perform the HAR classification task, in [38] the authors used inception modules consisting of 1D convolutional layers and DenseNet network to design HHARNet model. This model used three ''InceptionDense module'' with which to group features together according to depth. Ronald *et al.* constructed the iSPLIception model based on the Inception-Resnet by using the Inception module directly in the HAR model for extracting more features in terms of depth and width. Table 1 lists the related works regarding HAR briefly.

TABLE 1. The related works regarding HAR.

Type	Methods	Description	Proposers
Tradition ML	ML classifier	Getting 84% accuracy rate for 20 types of activities	Bao and Intille 2004
Deep Learning	CNN, Concatenate	Each dimension is considered as a channel	Zeng et al. 2014
	CNN	Processing time series with Conv1D	Yang et al. 2015
	CNN, Pooling	Models with alternating CNN and Pooling	Ronao and Cho 2016
	CNN, Statistical Learning	Real-time classification model	Ignatov 2018
	CNN, FC	4 CNN layers and 1 FC layer Fusion model	Bianchi et al. 2019
	LSTM-RNN	Fusion model	Pienaar et al. 2019
	CNN	Basing phone sensors	Wan et al. 2020
	LSTM	Hierarchical LSTM applied to HAR	Wang and Liu 2020
Fusion Model	3-input CNN-GRU	Focusing on 3-head module to extract features	Dua et al. 2021
	HHARNet	Transfer Inception module	Imran and Latif 2021
	iSPLIception	Transfer Inception module	Ronald et al. 2021

Inception module is essentially to extract features of different dimensions to enhance the computer vision and to increase the depth of model. However, the improvement obtained by directly applying the Inception module or modifying the convolutional kernel size of the Inception module to the HAR model is not obvious enough. We continue to extend the idea of inception by optimizing the network structure and parameters of each channel. Specifically, following the input this model connects multiple channels of CNN neural networks with different convolutional kernel sizes, while the batch normalization (Batch Norm) layer is added between the two convolutional layers of a single channel to speed up the network convergence, and a max pooling layer added at the end of each channel. In the end, the output features at different scales of each channel are connected using Concatenation layer similarly.

The fused features are fed into the GRU neural network for feature labeling, and the using GAP instead of FC layers could make the transformation smoother and enable the model to have stronger anti-fitting ability, reducing the number of parameters significantly. The excessive number of parameters will limit the application of DL models of sensor-based HAR to real-world environments. Although deeper models have the ability to express features more richly, the pursuit of complexity can lead to huge system overhead, making it difficult to be applied in real world. Our model achieves better results with fewer parameters and is more adaptable to practical applications.

III. METHODS

A. MULTICHANNEL CNN

CNN has been widely used in DL and derived many classical structures, such as FCN, Res-Net. These methods play important roles in classification tasks of HAR. Fig. 2 depicts the process of extracting time series features using Conv1D. The convolution kernel is convolved with a window of medium length of the sample to obtain the corresponding

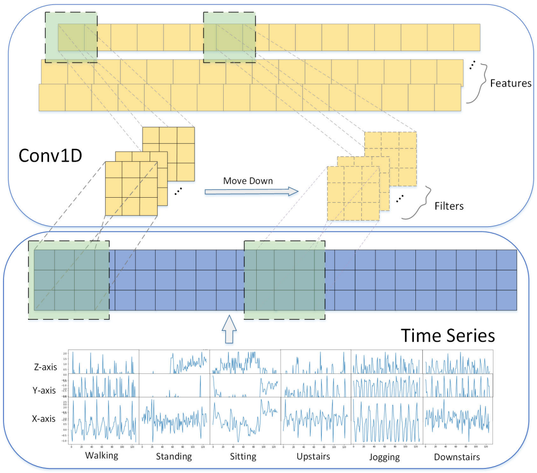


FIGURE 2. Extraction of time series features using Conv1D.

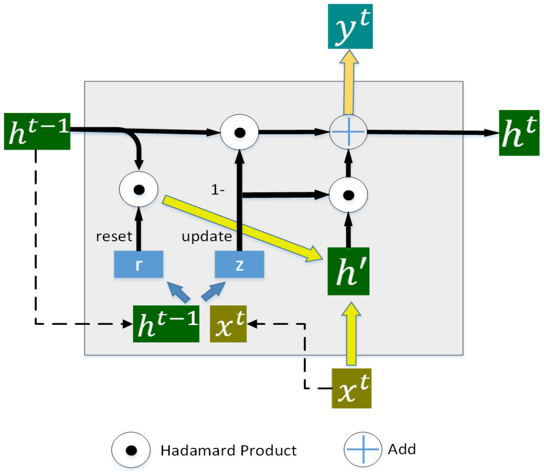


FIGURE 3. The principle of GRU.

features, and it is shifted down to convolve with the data behind. The dimension of resulting features equals the number of convolution kernels. We apply the structural features of Inception module to our model, in which different channels obtain features at different scales. This can enhance the receptive field of the computer to perform HAR tasks. Three channels at different convolutional scales are designed, with Batch Norm layer added between the convolutional layers for normalization, and the sample data go through each channel and the output features are connected. CNN extracts the features at multiple scales and makes the model obtain stronger feature representation, which will effectively improve the accuracy of classification.

B. GRU

GRU solves the problem of gradient disappearance and explosion of general RNN. Fig. 3 depicts the principle of GRU, which has the same structure of input and output as the RNN. The current input x^t and the hidden state h^{t-1} passed down from the previous node are passed through GRU to get the output y^t and the hidden state h^t passed to the next node. It only needs one unit to complete two operations of forgetting and selecting memory (LSTM needs multiple units to complete this function), and the Formula 1 is the updated expression of GRU unit.

$$h^t = (1 - z) \odot h^{t-1} + z \odot h' \tag{1}$$

GRU has better performance in longer sequence data compared to RNN. GRU controls the transmission state with the state of gates, remembering the critical information that needs to be kept for a long time and forgetting the unimportant ones. Compared with LSTM, GRU has a smaller number of parameters. The features at different scales extracted from multiple channels are fused and put into the GRU layer for labeling the time-dependent sequences, enhancing the feature representation. A model consisting of CNN networks only cannot solve the problem of error tolerance, with wrong data or illegal data increasing the recognition rate of CNN decreases. This is due to that this model fails to filter dirty data in the input samples. Instead, GRU network enables the model to have fault-tolerant capability. The input samples correspond to several feature maps at several consecutive moments, even a wrong channel occurs in the corresponding feature map in a certain moment, GRU would predict and erase the errored channel according to the other features for there is a time dependency in each feature map.

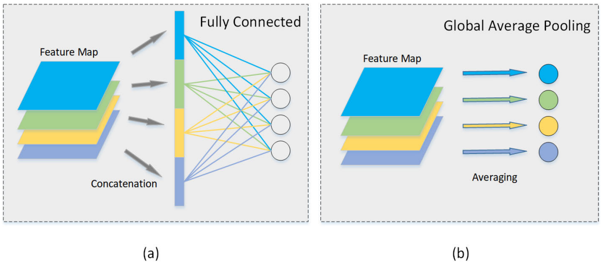


FIGURE 4. The conversion process of features using: (a) FC; (b) GAP.

C. GAP

The FC layer connects the convolutional layer and the normal layer, takes the data from the previous layer, and puts the result into the normal layer through nonlinear transformation, its conversion process is shown in Fig. 4 (a); the GAP layer averages the feature data in both height and width dimensions, while the FC layer is prone to overfitting when training too many parameters, its conversion process is shown in Fig. 4 (b). Thus, the GAP layer has a more stable performance. There are two advantages of using GAP instead of FC: First, the transformation between feature map and final classification is simpler and more natural in GAP; Second, it does not need a large number of training tuning parameters like FC layer, which reduces the number of spatial parameters and makes the model more stable.

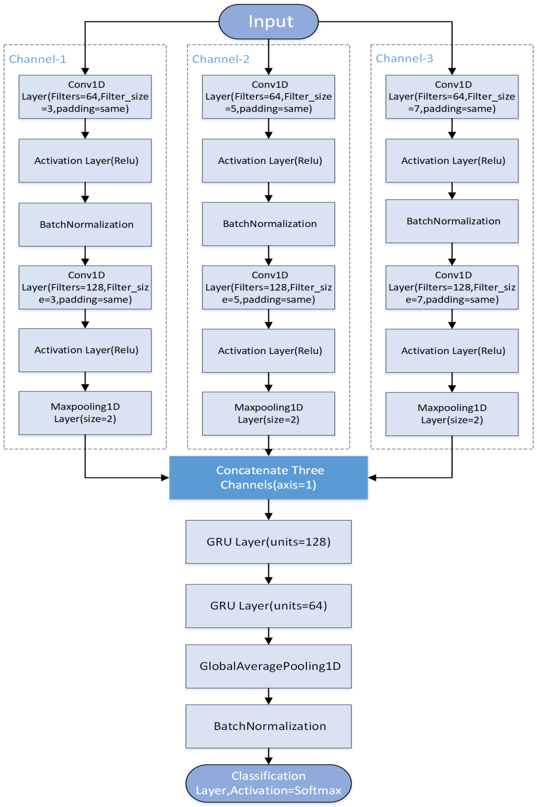

Input
Channel-1
Channel-2
Channel-3
Conv1D Layer(Filters=64,Filter_size=3,padding=same)
Conv1D Layer(Filters=64,Filter_size=5,padding=same)
Conv1D Layer(Filters=64,Filter_size=7,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=128,Filter_size=3,padding=same)
Conv1D Layer(Filters=128,Filter_size=5,padding=same)
Conv1D Layer(Filters=128,Filter_size=7,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Concatenate Three Channels(axis=1)
GRU Layer(units=128)
GRU Layer(units=64)
GlobalAveragePooling1D
BatchNormalization
Classification Layer,Activation=Softmax

FIGURE 5. Multichannel CNN-GRU model.

D. PROPOSED MODEL

In this study, a multichannel CNN-GRU model is proposed for HAR. After input of the samples, they are fed into three channels with convolutional kernels of different scales, then into two GRU layers after feature fusion, and sent to the final classification layer through GAP and Batch Norm layers. The structure of the model is shown in Fig. 5. Three channels are similar in structure except for the convolutional kernels at different size scales in the convolutional layer; the convolutional kernels at different scales obtain features of different scales from the samples and possess the capability of enhancing the vision of the neural network.

The samples first pass through a Conv1D of the channel, which accepts input data in three dimensions. The first dimension – the number of samples, the second dimension – the size of the sliding window is 128, and the third dimension – the original number of features (3 for the WISDM dataset and 9 for the UCI-HAR dataset). Then the data passes through the activation layer with an activation function of rectified linear unit (ReLU), followed by Batch Norm layer, which converts the sample to data with a mean of 0 and standard deviation of 1. This can speed up the training and convergence of the model, control the gradient explosion, prevent the gradient from disappearing, and reduce overfitting. Then the data go through a Conv1D layer and an activation layer with the same activation function as ReLU, and finally the 1-dimensional max pooling (MaxPooling1D) layer with a size of 2 and a step size of 1. The number of convolution kernels in the first Conv1D layer is 64 for all channels, and the second one is 128. The structure of the three channels is the same except that the size of the convolution kernels is 3, 5, and 7, respectively. These extracted features are concatenated in the concatenation layer and fed into the GRU layer with the number of neurons of 128 and 64, respectively. Then, they go through the GAP layer and the Batch Norm layer to realize normalization, and then the Dense layer with softmax activation function as the classification function to obtain the normalized output.

The spatial complexity of CNN networks is low, and the number of its parameters is related to the feature dimension and the number of convolutional kernels, etc. Conv1D is used in our model, with two input feature dimension of sliding window size and features, and its number of parameters is low. The number of GRU parameters is the sum of updated unit parameters and reset unit parameters, with its size related to the input dimension and gate units. In our experiment, the number of parameters of the proposed model with the comparison model is compared, which is an important evaluation of the HAR framework.

IV. EXPERIMENTS AND RESULTS

A. DATASETS

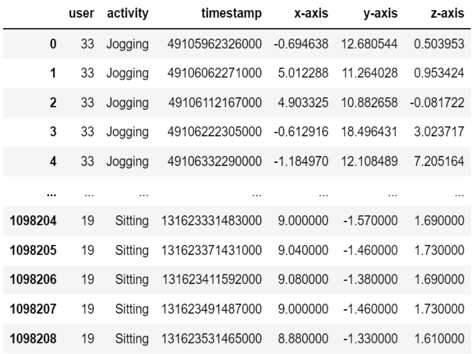
To verify the validity of the model, experiments were conducted using the WISDM dataset (single sensor), the UCI-HAR dataset (multi sensors), and the PAMAP2 dataset (multi sensors). The basics of the three datasets are described below.

1) WISDM DATASET

WISDM is a benchmark HAR dataset provided by the Wireless Sensor Data Mining (WISDM) Lab research team. 36 participants, with Android smartphones in their front leg pocket, conducted specific activities in a controlled environment [41]. A total of 1,098,207 samples (sampled at 20 Hz) are obtained using a three-axial acceleration (implanted in an Android phone). Participants were asked to perform six activities: sitting, standing, walking, walking up, down stairs, and jogging. Each sample consists of six attributes: user ID, activity, timestamp, x-acceleration, y-acceleration, and z-acceleration. Some of the data in WISDM are displayed in Fig. 6.

2) UCI-HAR DATASET

UCI-HAR was collected from 30 volunteers between the ages of 19 and 48 wearing a smartphone (Samsung Galaxy SII) on the waist [42]. Eatch Each individual performed six activities-three static activities: walking, walking upstairs, and walking downstairs, and three dynamic ones: sitting, standing, and laying. These data were recorded by the developed software. Two three-axial linear acceleration and a three-axial angular velocity captured the data at a constant rate of 50 Hz using the built-in gyroscope and accelerometer



	user	activity	timestamp	x-axis	y-axis	z-axis
0	33	Jogging	49105962326000	-0.694638	12.680544	0.503953
1	33	Jogging	49106062271000	5.012288	11.264028	0.953424
2	33	Jogging	49106112167000	4.903325	10.882658	-0.081722
3	33	Jogging	49106222305000	-0.612916	18.496431	3.023717
4	33	Jogging	49106332290000	-1.184970	12.108489	7.205164
...	...	...	...	...	...	...
1098204	19	Sitting	131623331483000	9.000000	-1.570000	1.690000
1098205	19	Sitting	131623371431000	9.040000	-1.460000	1.730000
1098206	19	Sitting	131623411592000	9.080000	-1.380000	1.690000
1098207	19	Sitting	131623491487000	9.000000	-1.460000	1.730000
1098208	19	Sitting	131623531465000	8.880000	-1.330000	1.610000

FIGURE 6. Part of WISDM dataset.

TABLE 2. A brief description of UCI-HAR dataset.

Attribute	Description
Sensors	three-axial total and body acceleration, three-axial angular velocity of gyroscope
Dividing	7,352 samples as training set, 2,947 samples as test set
Sliding window	2.56sec×50Hz = 128cycles
Activities	6

TABLE 3. A brief description of PAMAP2 dataset.

Attribute	Description
Sensors	Gyroscope, accelerometer, heart rate monitor
Sampling frequency	100 Hz
Activities	18
Volunteers	9

of the smartphone. The training and test set have been divided and its pre-processing has also been completed in UCI-HAR. So we can just use it. A brief description of UCI-HAR is shown in Table 2.

3) PAMAP2 DATASET

PAMAP2 — recorded from 18 activities performed by 9 subjects, wearing 3 IMUs and a HR-monitor — is created and made publicly available by Reiss et al [43]. Three inertial measurement units (IMUs) and a heart rate monitor were used as sensors during the data collection. These relatively lightweight and small IMUs contain 3-axis MEMS sensors, including two accelerometers, a gyroscope and a magnetometer, all sampled at 100 Hz. Participants followed a protocol of 12 activities (lie, sit, stand, walk, run, cycle, Nordic walk, iron, vacuum clean, rope jump, ascend and descend stairs) and 6 optional activities (watch TV, computer work, drive car, fold laundry, clean house, play soccer). The data are from a total of 9 volunteers, aging from 24 to 32, and each performs some of these activities. The description of this dataset is presented in Table 3.

B. DATASET PREPROCESSING

The original data needs to be pre-processed due to their unbalance distribution. By normalization, we make the data have a mean of 0 and standard deviation of 1. To better evaluate the effectiveness of the proposed model, special attention is given to divide the dataset. The original data consists of time series of different activities by user ID, and the data of the user to be predicted is completely unknown when the model is applied to reality. With a sliding window splitting the original data, the dataset is randomly divided into training set and the test set according to a certain ratio. This would lead to that some samples of the same user's activity may appear in both training set and test set. Dividing the dataset in this way may improve the accuracy of the proposed model, but does not reflect its true validity.

The reality is that the user data to be tested is completely unknown when the model is applied. Thus, we divide the training and test sets by user IDs to ensure that the samples from the same ID could only exist in one of the two sets. The size of the sliding window and the overlap have a great impact on the partitioning of time-series data. The sliding window size of 128 and the overlap rate of 50% are set to all WISDM, UCI-HAR, and PAMAP2 according to the sampling frequency and human activity habits.

C. EVALUATION METRICS

Commonly used evaluation metrics for classification models includes: precision, recall, and F1 score. These metrics will be used to evaluate the proposed model.

Accuracy: For a given test dataset, the ratio of the number of samples correctly classified by the classifier to the total number of samples is the correct rate for the identified samples.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{2}$$

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

Precision: The ratio of the number of correctly identified positive samples to the total number of samples identified as positive in the identified sample.

$$Precision = \frac{TP}{TP + FP} \tag{3}$$

Recall: Also known as sensitivity, is the fraction of examples classified as positive, among the total number of positive examples.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

F1-score: It is a measure of a model's accuracy on a dataset, used to evaluate binary classification systems, which is the harmonic mean of the precision and recall.

$$F1 - score = 2 \times \frac{precision \times recall}{precision + recall} \tag{5}$$

Confusion matrix (CM): It is a square matrix that gives the full performance of the classification model. rows of the CM represent instances of the true class labels and columns represent the predicted class labels. The diagonal elements of

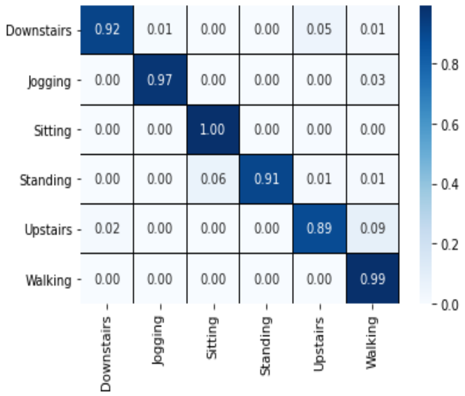


FIGURE 7. Confusion matrix of the proposed model on the WISDM test sets.

this matrix define the number of points where the predicted labels are equal to the true labels.

Parameters: The amount of data to be trained in the model, measuring the spatial complexity of the model.

D. RESULTS AND DISCUSSION

In this section we test the proposed model on three benchmark datasets to evaluate its effectiveness. We carry out four experiments: the first is the performance of our proposed model on three datasets, the second is comparison of the three-channel model with other numbers of channels model, the third is comparing GRU with LSTM, and the fourth is comparing the model connected with GAP with the one connected with FC layer. The model is built and trained based on DL framework of Keras and TensorFlow-gpu 2.6.0. The labels are transformed into One-Hot encoding and trained using Adam optimizer with a learning rate of 0.001 and categorical cross-entropy serves as the loss function of the model. The Batch size is 96 and the number of training steps is 100. All the experiments in this study are performed on Windows 10 system, and the computer's CPU is R9-5900HX, memory is 16GB, and GPU is NVIDIA GeForce
RTX3060.

1) RESULT ON WISDM DATASET

The samples in the WISDM dataset were divided according to user IDs. The first 30 users (ID: 1-30) were used as the training set and the last 6 users (ID: 31-36) were used as the test set. The training set had a total of 14,035 samples and the test set had a total of 3,121 ones. Fig. 7 shows the confusion matrix obtained from the trained model on the test set. The experimental results show that the model achieved accuracy over 97% for four action categories (walking, jogging, standing, sitting), with upstairs and downstairs lower than others due to the similarity of the two actions.

Table 4 shows the evaluation metrics of the proposed model on the WISDM dataset, the accuracy and F1-score of reaching 96.41% and 96.39%, respectively.

TABLE 4. Evaluation metrics of the proposed model on the WISDM dataset.

Model	Precision	Recall	F1-score	Accuracy
Proposed	0.9643	0.9642	0.9639	0.9641

TABLE 5. Evaluation metrics comparison of the proposed model with other models on WISDM dataset.

Model	F1-score	Accuracy
CNN [32]	–	0.9332
LSTM-CNN [27]	–	0.9585
LSTM-RNN [36]	0.9573	0.9578
Proposed	**0.9639**	**0.9641**

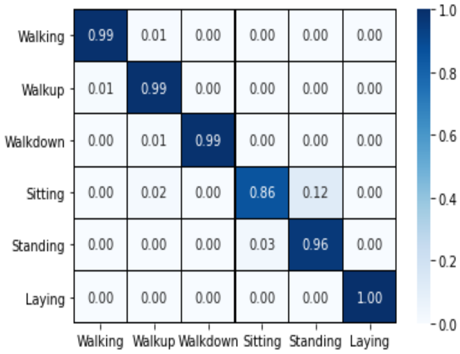


FIGURE 8. Confusion matrix of the proposed model on the UCI-HAR test sets.

Our model is compared with the existing models, as shown in Table 5. It demonstrates that the F1-score and Accuracy of this model against other models, showing that this model outperforms other compared methods for HAR.

2) RESULT ON UCI-HAR DATASET

In UCI-HAR dataset, 7352 samples are used as the training set and 2947 samples as the test set. Fig. 8 shows the confusion matrix obtained by evaluating the trained model on the test set. The results show that the model achieved over 95% accuracy for five action categories (walking, walkup, walkdown, standing, laying).

Table 6 shows the evaluation metrics of the model on the UCI-HAR dataset, with its accuracy reaching 96.67% and F1-score reaching 96.72%.

This model is also compared with the existing models. Table 7 compares the F1-score and accuracy of the proposed model with other models, showing that this model outperforms compared methods.

3) RESULT ON PAMAP2 DATASET

In this dataset, 11 protocol activities are chosen to perform classification. Note that the 24th activities of rope jumping is not chosen because it has very little recording time, and even some users did not perform this activity. The other activities are more balanced categories. The data of No. 6 and

TABLE 6. Evaluation metrics of the proposed model on the UCI-HAR dataset.

Model	Precision	Recall	F1-score	Accuracy
Proposed	0.9659	0.9657	0.9672	0.9667

TABLE 7. Evaluation metrics comparison of the proposed model with other models on UCI-HAR dataset.

Model	F1-score	Accuracy
CNN [25]	0.9293	0.9271
CNN-LSTM [44]	–	0.9213
LSTM-CNN [27]	–	0.9578
Res-LSTM [45]	0.9150	0.9160
Stacked-LSTM [45]	–	0.9313
Proposed	**0.9672**	**0.9667**

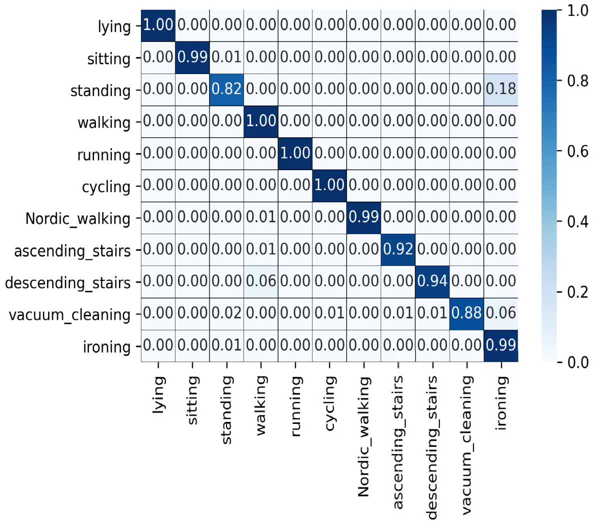


FIGURE 9. Confusion matrix of the proposed model on the PAMAP2 test sets.

No. 7 of nine users were selected as the test set, and we performed a linear interpolation of the missing values in the corresponding activities for the selected users. Meanwhile, the data of first 10 seconds and the last 10 seconds of each activity are deleted to reduce the mislabeling. All the 52 features are selected, and 19,700 training set samples and 6727 test set samples were obtained. Fig. 9 shows the confusion matrix obtained by evaluating the trained model on the test set. The proposed model has a lower recognition rate on sitting and vacuum cleaning, but has a better performance on other activities. Both standing and vacuum cleaning are easily misclassified as ironing, due to their similar activity characteristics.

Table 8 shows the evaluation metrics of the model on the UCI-HAR dataset, with its accuracy reaching 96.25% and F1-score reaching 96.59%.

In Table 9, the F1-score and accuracy of proposed model are compared with other models, and the results show that this model outperforms other comparison methods.

TABLE 8. Evaluation metrics of the proposed model on the PAMAP2 dataset.

Model	Precision	Recall	F1-score	Accuracy
Proposed	0.9719	0.9625	0.9659	0.9625

TABLE 9. Evaluation metrics comparison of the proposed model with other models on PAMAP2 dataset.

Model	F1-score	Accuracy
BiLSTM [25]	0.894	0.8952
CNN [25]	0.9116	0.91
Proposed	**0.9659**	**0.9625**

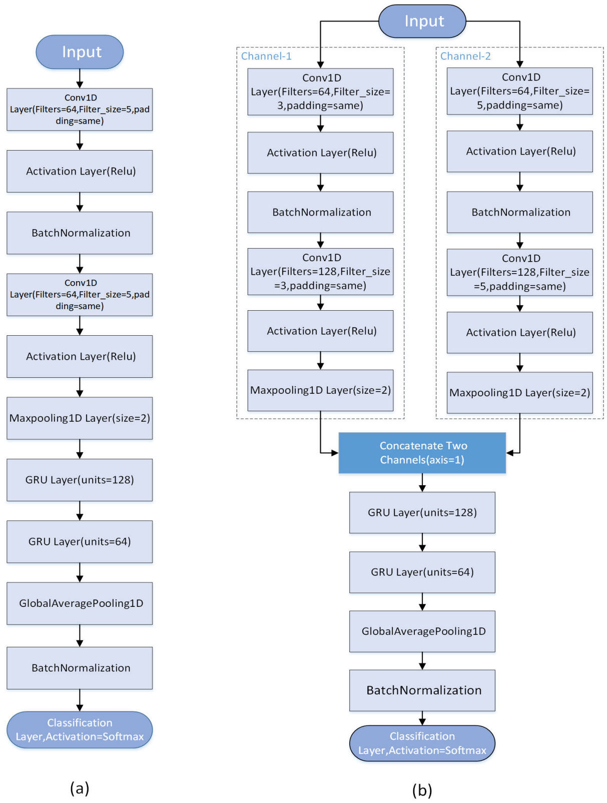

Input
Conv1D Layer(Filters=64,Filter_size=5,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=64,Filter_size=5,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
GRU Layer(units=128)
GRU Layer(units=64)
GlobalAveragePooling1D
BatchNormalization
Classification Layer,Activation=Softmax
(a)
Input
Channel-1
Conv1D Layer(Filters=64,Filter_size=3,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=128,Filter_size=3,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Channel-2
Conv1D Layer(Filters=64,Filter_size=5,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=128,Filter_size=5,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Concatenate Two Channels(axis=1)
GRU Layer(units=128)
GRU Layer(units=64)
GlobalAveragePooling1D
BatchNormalization
Classification Layer,Activation=Softmax
(b)

FIGURE 10. Model structure: (a) 1-channel; (b) 2-channel.

4) COMPARISON OF MULTICHANNEL CNN-GRU MODEL WITH FUSION MODEL

Dua *et al.* performed sequence-based convolution. Then, the samples are through pooling and flattening operations, inputted to GRU, and then concatenate was used to connect the features from multi-head module. The final classification was obtained by connecting them with a FC layer. Each head contains two layers of GRU, this will make the head heavy and result in a large number of parameters. Thus, it is not reasonable to perform classification directly after fusing features.

Some researchers have incorporated Inception module into HAR DL models, where they use one or more Inception

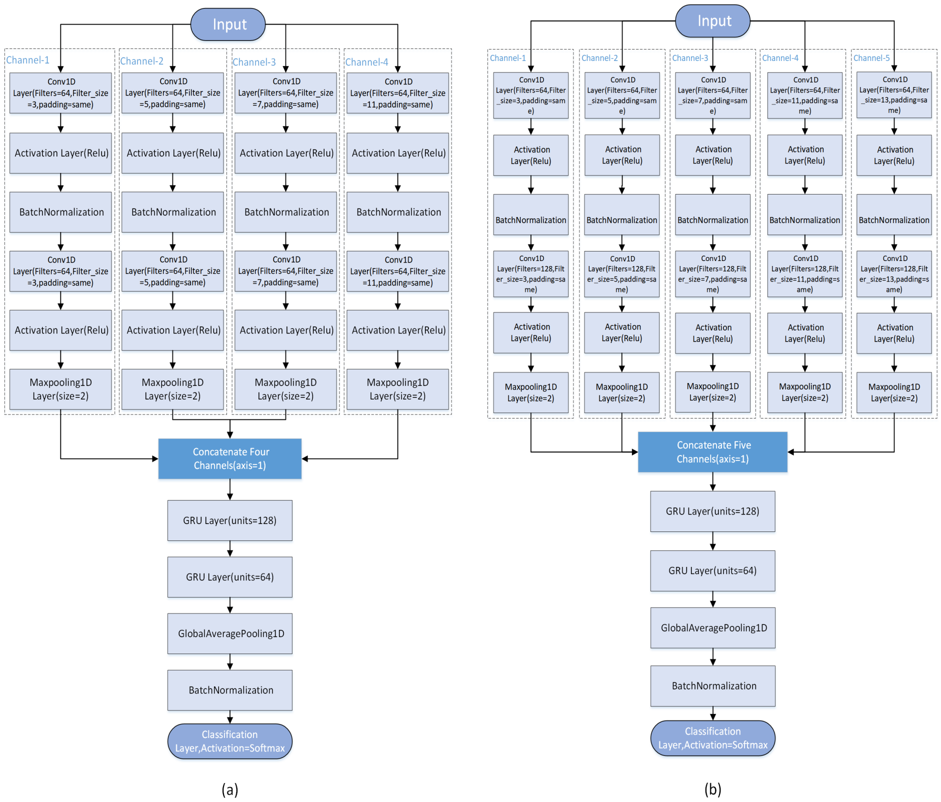

Input
Channel-1
Conv1D Layer(Filters=64,Filter_size=3,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=64,Filter_size=3,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Channel-2
Conv1D Layer(Filters=64,Filter_size=5,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=64,Filter_size=5,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Channel-3
Conv1D Layer(Filters=64,Filter_size=7,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=64,Filter_size=7,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Channel-4
Conv1D Layer(Filters=64,Filter_size=11,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=64,Filter_size=11,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Concatenate Four Channels(axis=1)
GRU Layer(units=128)
GRU Layer(units=64)
GlobalAveragePooling1D
BatchNormalization
Classification Layer,Activation=Softmax
(a)
Input
Channel-1
Conv1D Layer(Filters=64,Filter_size=3,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=128,Filter_size=3,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Channel-2
Conv1D Layer(Filters=64,Filter_size=5,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=128,Filter_size=5,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Channel-3
Conv1D Layer(Filters=64,Filter_size=7,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=128,Filter_size=7,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Channel-4
Conv1D Layer(Filters=64,Filter_size=11,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=128,Filter_size=11,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Channel-5
Conv1D Layer(Filters=64,Filter_size=13,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=128,Filter_size=13,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Concatenate Five Channels(axis=1)
GRU Layer(units=128)
GRU Layer(units=64)
GlobalAveragePooling1D
BatchNormalization
Classification Layer,Activation=Softmax
(b)

FIGURE 11. Model structure: (a) 4-channel; (b) 5-channel.

TABLE 10. Performance comparison of the proposed model with the Inception fusion model on WISDM, UCI-HAR, and PAMAP2.

Dataset	WISDM			UCI-HAR			PAMAP2		
Metric	F1-Score	Accuracy	Parameters	F1-Score	Accuracy	Parameters	F1-Score	Accuracy	Parameters
Multi-input CNN-GRU [37]	0.9722	0.9721	622,950	0.9619	0.962	631,014	0.9524	0.9527	647,916
HHARNet [38]	0.95	0.9526	–	–	–	–	–	–	–
iSPLIception [39]	–	–	–	0.95	0.9509	1,327,745	–	–	–
Proposed	**0.9639**	**0.9641**	**264,070**	**0.9672**	**0.9667**	**269,830**	**0.9659**	**0.9625**	**311,435**

modules in the hope that the model will have depth and width to extract more comprehensive and effective features. In this study, we deepen the depth and widen the width of the model, taking into account the parameter number, to optimize both the structure of the model and the network layer. Our design allows the proposed model to have good performance. Table 10 compares the F1 score, accuracy, and parameter number of similar multi-channel models fusing the inception module with the proposed model on the WISDM, UCI-HAR, and PAMAP2 datasets.

It is clear that the model in this study is better than other two similar inception fusion multichannel models in the accuracy and the parameter number. Our framework has better performance than multi-input GRU-CNN on UCI-HAR and PAMAP2, and the number of parameters is much smaller.

5) PERFORMANCE COMPARISON OF MODELS WITH DIFFERENT NUMBER OF CHANNELS

More number of channels means more convolutional kernels of different sizes can be involved for feature extraction, so is it true that more channels will have higher classification accuracy? We designed 1-channel, 2-channel, 4-channel, and 5-channel models for comparison, and the 1-channel model structure and 2-channel model structure are illustrated in Fig. 10 (a) and Fig. 10 (b) respectively, the 4-channel model structure and 5-channel model structure are illustrated in Fig. 11 (a) and Fig. 11 (b) respectively. These models have the same layers and parameters as the proposed model except the number of channels and the size of the convolutional kernel. The convolutional kernel size of the two convolutional layers of the 1-channel model is 5, and the rest of the settings are the same as the 3-channel CNN-GRU model. In the 2-channel model, the size of the convolution kernel of the first channel convolution layer is 3, the size of the convolution kernel of the second channel convolution layer is 5, and the rest of the settings are as above. The first three paths of the 4-channel model are the same as the proposed 3-channel CNN-GRU model, the fourth channel is 11, and the rest of the settings are the same as above. The first four channels of the 5-channel model have the same settings as the 4-channel model, the fifth channel is 13, and the rest of the settings are the same.

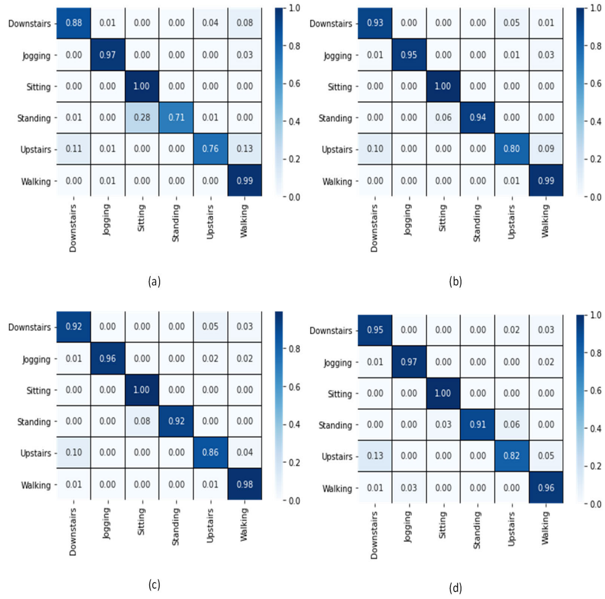


FIGURE 12. Confusion matrix for models with different channels on WISDM: (a) 1-channel model; (b) 2-channel model; (c) 4-channel model; (d) 5-channel model.

The confusion matrices of 1-, 2-, 4- and 5-channel models on WISDM are shown in Fig. 12 (a), Fig. 12 (b), Fig. 12 (c) and Fig. 12 (d) respectively, the confusion matrices of 1-, 2-, 4- and 5-channel models on UCI-HAR are shown in Fig. 13 (a), Fig. 13 (b), Fig. 13 (c) and Fig. 13 (d) respectively, the confusion matrices of 1-, 2-, 4- and 5-channel models on PAMAP2 are shown in Fig. 14 (a), Fig. 14 (b), Fig. 14 (c) and Fig. 14 (d) respectively, which were obtained from the test sets of the three datasets with different channel models separately, and it can be seen that different channels have different effects for different action recognition. On the WISDM dataset, the 1-channel CNN-GRU is prone to identify standing as sitting and upstairs as walking, and the 2-, 4- and 5-channel models are prone to identify upstairs as downstairs. On the UCI-HAR dataset, all the channel models have the problem of identifying sitting as standing, but this situation slightly improves as the number of channels increases. On the PAMAP2 dataset, both standing and vacuum cleaning are easily misclassified as ironing.

Table 11 records the accuracy, F1-score and parameter number of the models with different number of channels on the WISDM, UCI-HAR, and PAMAP2. The parameter number measures the lightweight of a model, and we can see from the table that the proposed model proposed is higher in accuracy than other models with different number of channels, and also has a reasonable number of parameters.

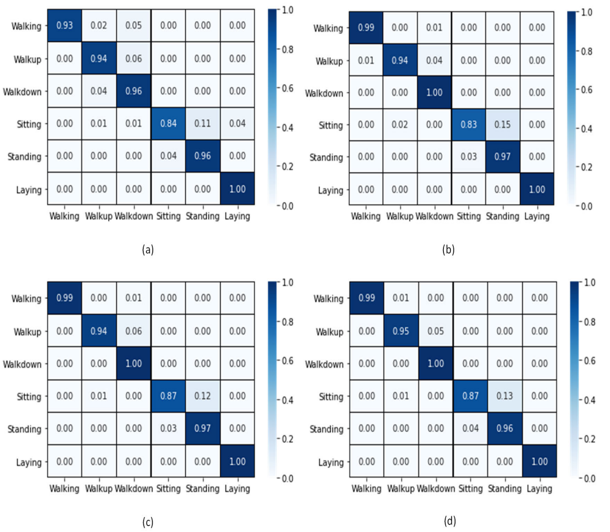


FIGURE 13. Confusion matrix for models with different channels on UCI-HAR: (a) 1-channel model; (b) 2-channel model; (c) 4-channel model; (d) 5-channel model.

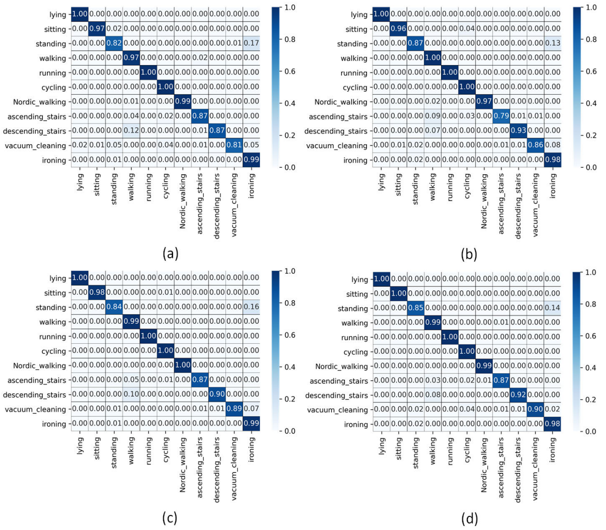


FIGURE 14. Confusion matrix for models with different channels on PAMAP2: (a) 1-channel model; (b) 2-channel model; (c) 4-channel model; (d) 5-channel model.

TABLE 11. Evaluation metrics of different number of channels models on WISDM, UCI-HAR, and PAMAP2.

Dataset	WISDM			UCI-HAR			PAMAP2		
Metric	F1-Score	Accuracy	Parameters	F1-Score	Accuracy	Parameters	F1-Score	Accuracy	Parameters
Channel-1	0.9330	0.9346	179,334	0.9380	0.9382	181,254	0.9471	0.9474	195,339
Channel-2	0.9511	0.9513	204,934	0.9541	0.9545	208,006	0.9537	0.9538	230,022
Channel-3	**0.9639**	**0.9641**	**264,070**	**0.9672**	**0.9667**	**269,830**	**0.9659**	**0.9625**	**311,435**
Channel-4	0.9554	0.9551	356,742	0.9601	0.9603	366,726	0.9618	0.9612	438,603
Channel-5	0.9512	0.9513	466,182	0.9601	0.9603	473,350	0.9643	0.9634	588,811

6) COMPARISON OF MODEL USING GRU WITH LSTM LAYER

The GRU layers has similar effect with LSTM, but has less parameters, making the model more lightweight. Fig. 15 shows the structure of the multichannel CNN-LSTM model, in which two layers after feature fusion differ from the proposed model, and the rest of the settings are the same.

The confusion matrices of multichannel CNN-LSTM model on WISDM, UCI-HAR, and PAMAP2 are shown

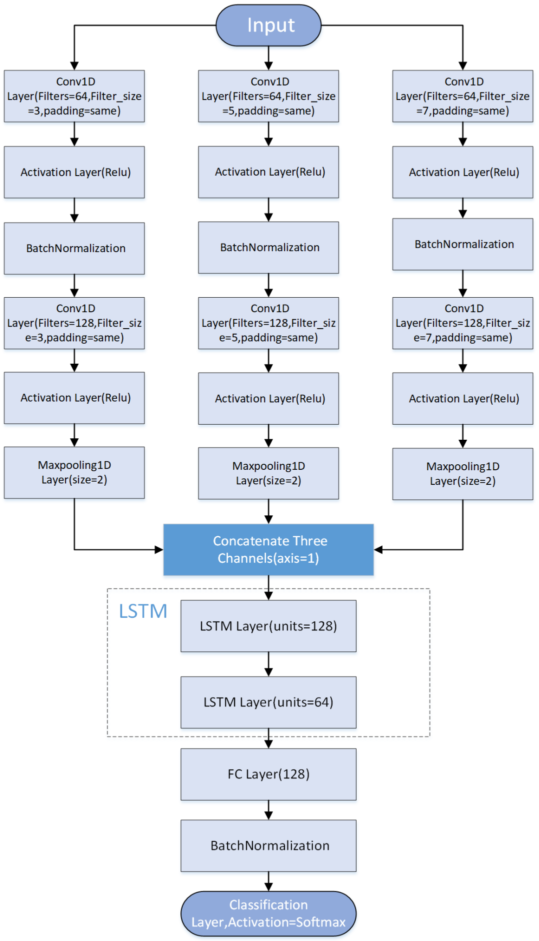

Input
Conv1D Layer(Filters=64,Filter_size =3,padding=same)
Conv1D Layer(Filters=64,Filter_size =5,padding=same)
Conv1D Layer(Filters=64,Filter_size =7,padding=same)
Activation Layer(Relu)
BatchNormalization
Conv1D Layer(Filters=128,Filter_size=3,padding=same)
Conv1D Layer(Filters=128,Filter_size=5,padding=same)
Conv1D Layer(Filters=128,Filter_size=7,padding=same)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Concatenate Three Channels(axis=1)
LSTM
LSTM Layer(units=128)
LSTM Layer(units=64)
FC Layer(128)
BatchNormalization
Classification Layer,Activation=Softmax

FIGURE 15. Multichannel CNN-LSTM model structure.

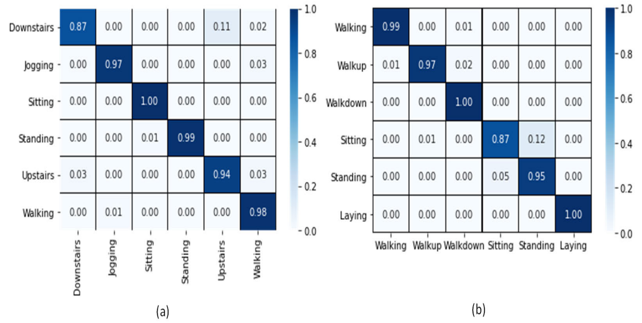


FIGURE 16. Confusion matrix of multichannel CNN-LSTM model on: (a) WISDM; (b) UCI-HAR.

in Fig. 16 (a) and Fig. 16 (b), and Fig. 17 respectively, which were obtained under the same experimental environment and settings. On the WISDM dataset, the upstairs recognition accuracy of the multichannel CNN-LSTM model is higher than that of the multichannel CNN-GRU model, but the downstairs recognition rate is lower than that of the multichannel CNN-GRU model; on the UCI-HAR and PAMAP2 dataset, the two models have similar results.

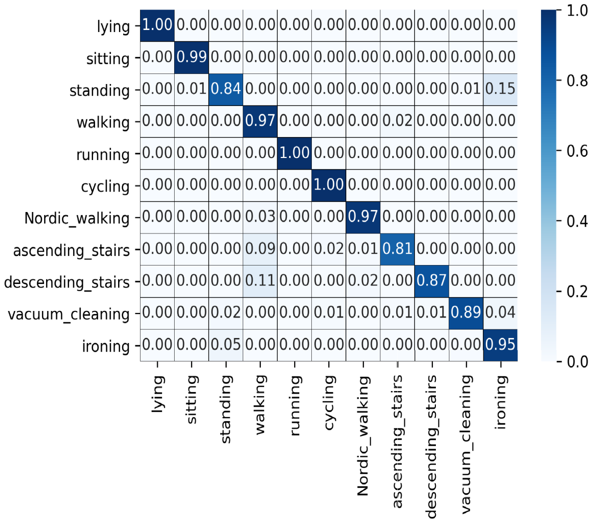


FIGURE 17. Confusion matrix of multichannel CNN-LSTM model on PAMAP2.

TABLE 12. Performance comparison of multichannel CNN-LSTM and multichannel CNN-GRU.

Dataset	WISDM			UCI-HAR			PAMAP2		
Metric	F1-Score	Accuracy	Parameters	F1-Score	Accuracy	Parameters	F1-Score	Accuracy	Parameters
Multichannel CNN-LSTM	0.9653	0.9654	308,742	0.9642	0.9637	314,502	0.9462	0.9462	356,107
Multichannel CNN-GRU (Proposed)	**0.9639**	**0.9641**	**264,070**	**0.9672**	**0.9667**	**269,830**	**0.9659**	**0.9625**	**311,435**

TABLE 13. Comparison of multichannel CNN-GRU-FC with multichannel CNN-GRU-GAP.

Dataset	WISDM			UCI-HAR			PAMAP2		
Metric	F1-Score	Accuracy	Parameters	F1-Score	Accuracy	Parameters	F1-Score	Accuracy	Parameters
Multichannel CNN-GRU-FC	0.9546	0.9542	273,030	0.9620	0.9620	278,790	0.9484	0.9416	320,715
Multichannel CNN-GRU-GAP (Proposed)	**0.9639**	**0.9641**	**264,070**	**0.9672**	**0.9667**	**269,830**	**0.9659**	**0.9625**	**311,435**

Table 12 compares evaluation metrics of the two models. They are similar in terms of accuracy and F1-score, but the multichannel CNN-GRU has a smaller parameter number. Lightweight is desirable in condition of accuracy.

7) COMPARISON OF MODELS USING THE GAP LAYER WITH THE FC LAYER

The model in this study uses a GAP layer instead of a FC layer to connect the feature maps from the GRU layer with final classification output. The two models have the same settings except for the connection layer. The structure of the multichannel CNN-GRU-FC model is shown in Fig. 18.

The trained model predicts the test set of both datasets to obtain the confusion matrices on WISDM, UCI-HAR, and PAMAP2 as shown in Fig. 19 (a), Fig. 19 (b), and Fig. 20 respectively.

Connecting the feature maps with the final classification using the fully connected layer is prone to overfitting.

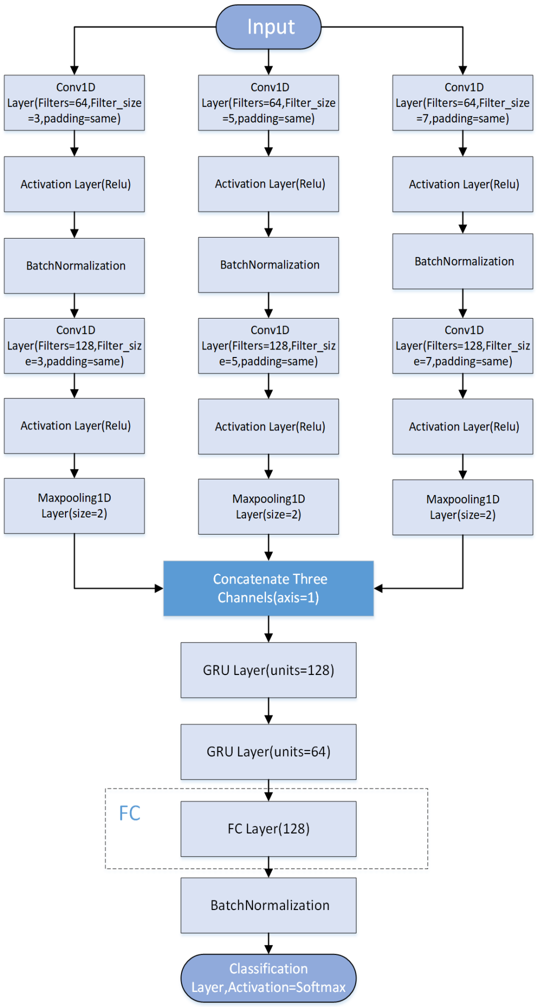

Input
Conv1D Layer(Filters=64,Filter_size =3,padding=same)
Conv1D Layer(Filters=64,Filter_size =5,padding=same)
Conv1D Layer(Filters=64,Filter_size =7,padding=same)
Activation Layer(Relu)
Activation Layer(Relu)
Activation Layer(Relu)
BatchNormalization
BatchNormalization
BatchNormalization
Conv1D Layer(Filters=128,Filter_size=3,padding=same)
Conv1D Layer(Filters=128,Filter_size=5,padding=same)
Conv1D Layer(Filters=128,Filter_size=7,padding=same)
Activation Layer(Relu)
Activation Layer(Relu)
Activation Layer(Relu)
Maxpooling1D Layer(size=2)
Maxpooling1D Layer(size=2)
Maxpooling1D Layer(size=2)
Concatenate Three Channels(axis=1)
GRU Layer(units=128)
GRU Layer(units=64)
FC
FC Layer(128)
BatchNormalization
Classification Layer,Activation=Softmax

FIGURE 18. Multichannel CNN-GRU-FC model structure.

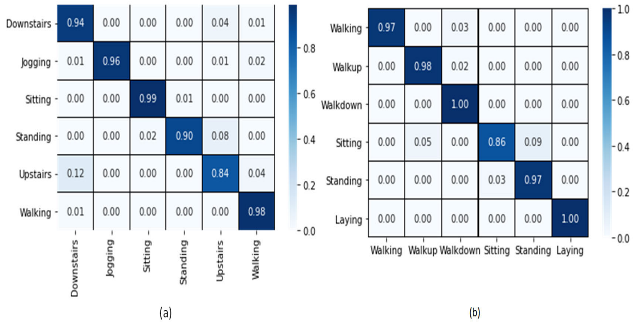


FIGURE 19. Confusion matrix of multichannel CNN-GRU-FC model on: (a) WISDM; (b) UCI-HAR.

As seen in the confusion matrix on the WISDM dataset, the multichannel CNN-GRU-FC model identifies quite a few of the standing actions as upstairs. Replacing the FC layer with the GAP layer can effectively suppress this phenomenon. Table 13 compares the F1-score, accuracy and parameter number of the multichannel CNN-GRU-GAP model with the multichannel CNN-GRU-FC model on the WISDM, UCI-HAR, and PAMAP2 datasets. It is clear that the proposed model with the GAP layer outperforms the FC-connected model in all evaluation metrics.

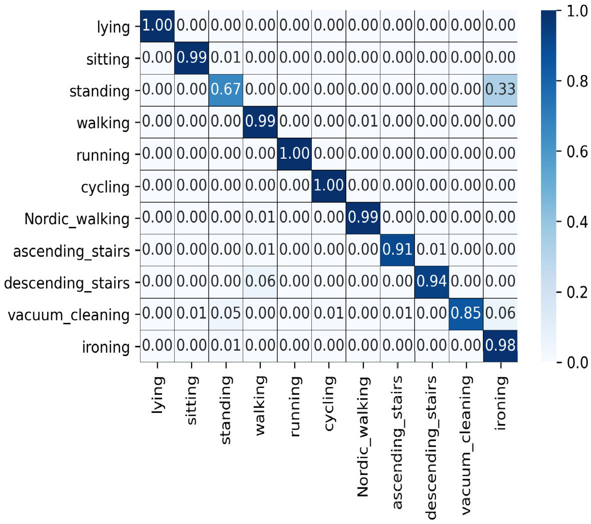


FIGURE 20. Confusion matrix of multichannel CNN-GRU-FC model on PAMAP2.

V. CONCLUSION

In this study, the proposed multichannel CNN-GRU model can identify user activity more accurately from raw data obtained from sensors. The multichannel CNN structure is able to extract different-scale features, GRU can extract time-dependent features, and the GAP layer allows to have a smaller parameter number. These advantages make the model identify human activity categories accurately and quickly. We demonstrate that the three-channel CNN-GRU model could balance both the number of parameters and the accuracy by comparing models with different channels. Experiments show that our proposed model has good performance on all datasets and outperforms other compared HAR models. Meanwhile, we observe the impact of data pre-processing on the classification results. In spite of fault tolerance of the model, the noise such as illegal and wrong data in raw data from the sensors still affects the classification result. It is not enough to use normalized pre-processing for the data in dataset, and for future work we intend to process the data more effectively. Besides, even though we tested this model on three benchmark datasets, the training samples are still relatively small, and in the future we will train our model on larger benchmark datasets or our own collected activity data to verify its generality for sensor-based HAR.

REFERENCES

[1] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Syst. Appl.*, vol. 105, pp. 233–261, Sep. 2018.

[2] D. Madhuranga, R. Madushan, C. Siriwardane, and K. Gunasekera, "Real-time multimodal ADL recognition using convolution neural networks," *Vis. Comput.*, vol. 37, no. 6, pp. 1263–1276, Jun. 2021.

[3] Z. Chen, C. Jiang, S. Xiang, J. Ding, M. Wu, and X. Li, "Smartphone sensor-based human activity recognition using feature fusion and maximum full *a posteriori*," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 3992–4001, Jul. 2020.

[4] Z. A. Khan and W. Sohn, "A hierarchical abnormal human activity recognition system based on R-transform and kernel discriminant analysis for elderly health care," *Computing*, vol. 95, no. 2, pp. 109–127, Feb. 2013.

[5] O. C. Ann and L. B. Theng, "Human activity recognition: A review," in *Proc. IEEE Int. Conf. Control Syst., Comput. Eng. (ICCSCE)*, Nov. 2014, pp. 389–393.

[6] Z. Hussain, Q. Z. Sheng, and W. E. Zhang, "A review and categorization of techniques on device-free human activity recognition," *J. Netw. Comput. Appl.*, vol. 167, Oct. 2020, Art. no. 102738.

[7] N. Hegde, M. Bries, T. Swibas, E. Melanson, and E. Sazonov, "Automatic recognition of activities of daily living utilizing insole-based and wrist-worn wearable sensors," *IEEE J. Biomed. Health Informat.*, vol. 22, no. 4, pp. 979–988, Jul. 2018.

[8] M. Vrigkas, C. Nikou, and I. A. Kakadiaris, "A review of human activity recognition methods," *Frontiers Robot. AI*, vol. 2, p. 28, Nov. 2015, doi: 10.3389/frobt.2015.00028.

[9] F. Demrozi, G. Pravadelli, A. Bihorac, and P. Rashidi, "Human activity recognition using inertial, physiological and environmental sensors: A comprehensive survey," *IEEE Access*, vol. 8, pp. 210816–210836, 2020.

[10] 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.

[11] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *Proc. 4th Int. Conf. Ambient Assist. Living Home Care*. Berlin, Germany: Springer-Verlag, 2012, pp. 216–223.

[12] W. Jiang and Z. Yin, "Human activity recognition using wearable sensors by deep convolutional neural networks," in *Proc. 23rd ACM Int. Conf. Multimedia*, Oct. 2015, pp. 1307–1310.

[13] L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, and W. Stork, "Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity," in *Proc. 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2008, pp. 5250–5253.

[14] C. Strobl, A. Boulesteix, A. Zeileis, and T. Hothorn, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinf.*, vol. 8, no. 1, p. 25, 2007.

[15] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Surv.*, vol. 46, no. 3, pp. 1–33, 2014, doi: 10.1145/2499621.

[16] 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.

[17] M. Gil-Martín, R. San-Segundo, F. Fernández-Martínez, and J. Ferreiros-López, "Improving physical activity recognition using a new deep learning architecture and post-processing techniques," *Eng. Appl. Artif. Intell.*, vol. 92, Jun. 2020, Art. no. 103679.

[18] D. K. Dewangan and S. P. Sahu, "RCNet: Road classification convolutional neural networks for intelligent vehicle system," *Intell. Service Robot.*, vol. 14, no. 2, pp. 199–214, Apr. 2021.

[19] Y. Lecun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jacke, "Handwritten digit recognition with a back-propagation network," in *Proc. Neural Inf. Process. Syst.*, vol. 2, 1997, pp. 1–9.

[20] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[21] Y. Guan and T. Plötz, "Ensembles of deep LSTM learners for activity recognition using wearables," *Proc. ACM Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 1, no. 2, pp. 1–28, 2017, doi: 10.1145/3090076.

[22] R. Dey and F. M. Salem, "Gate-variants of gated recurrent unit (GRU) neural networks," in *Proc. IEEE 60th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Aug. 2017, pp. 1597–1600.

[23] F. Liu, Z. Zhang, and R. Zhou, "Automatic modulation recognition based on CNN and GRU," *Tsinghua Sci. Technol.*, vol. 27, no. 2, pp. 422–431, Apr. 2022.

[24] B. Hartpence and A. Kwasinski, "CNN and MLP neural network ensembles for packet classification and adversary defense," *Intell. Converged Netw.*, vol. 2, no. 1, pp. 66–82, Mar. 2021.

[25] S. Wan, L. Qi, X. Xu, C. Tong, and Z. Gu, "Deep learning models for real-time human activity recognition with smartphones," *Mobile Netw. Appl.*, vol. 25, no. 2, pp. 743–755, 2020.

[26] M. B. Abidine, B. Fergani, and I. Menhour, "Activity recognition from smartphones using hybrid classifier PCA-SVM-HMM," in *Proc. Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Oct. 2019, pp. 1–5.

[27] K. Xia, J. Huang, and H. Wang, "LSTM-CNN architecture for human activity recognition," *IEEE Access*, vol. 8, pp. 56855–56866, 2020.

[28] 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.

[29] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *Proc. 2nd Int. Conf. Pervasive Comput.*, Vienna, Austria, 2004, pp. 1–17.

[30] 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," in *Proc. 6th Int. Conf. Mobile Comput., Appl. Services*, 2014, pp. 197–205.

[31] J. Yang, N. Nhut, P. San, X. Li, and P. Shonali, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. 24th Int. Joint Conf. Artif. Intell. (IJCAI)*, 2015, pp. 1–7.

[32] I. Andrey, "Real-time human activity recognition from accelerometer data using convolutional neural networks," *Appl. Soft Comput.*, vol. 62, pp. 915–922, Jan. 2018.

[33] L. Wang and R. Liu, "Human activity recognition based on wearable sensor using hierarchical deep LSTM networks," *Circuits, Syst., Signal Process.*, vol. 39, no. 2, pp. 837–856, 2020.

[34] V. Bianchi, M. Bassoli, G. Lombardo, P. Fornacciari, M. Mordonini, and I. De Munari, "IoT wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment," *IEEE Internet Things J.*, vol. 6, no. 5, pp. 8553–8562, Oct. 2019.

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

[36] S. W. Pienaar and R. Malekian, "Human activity recognition using LSTM-RNN deep neural network architecture," in *Proc. IEEE 2nd Wireless Afr. Conf. (WAC)*, Aug. 2019, pp. 1–5.

[37] N. Dua, S. N. Singh, and V. B. Semwal, "Multi-input CNN-GRU based human activity recognition using wearable sensors," *Computing*, vol. 103, no. 7, pp. 1461–1478, Mar. 2021.

[38] H. A. Imran and U. Latif, "HHARNet: Taking inspiration from inception and dense networks for human activity recognition using inertial sensors," in *Proc. IEEE 17th Int. Conf. Smart Commun., Improving Quality Life Using ICT, IoT AI (HONET)*, Dec. 2020, pp. 24–27.

[39] M. Ronald, A. Poulose, and D. S. Han, "ISPLInception: An inception-ResNet deep learning architecture for human activity recognition," *IEEE Access*, vol. 9, pp. 68985–69001, 2021.

[40] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

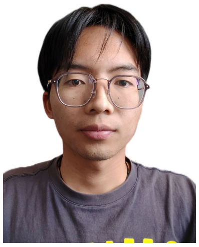
[41] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, 2011, doi: 10.1145/1964897.1964918.

[42] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones," in *Proc. 21st Eur. Symp. Artif. Neural Netw. (ESANN)*, 2013, pp. 24–26.

[43] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *Proc. 16th Int. Symp. Wearable Comput.*, Jun. 2012, pp. 108–109.

[44] R. Mutegeki and D. S. Han, "A CNN-LSTM approach to human activity recognition," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Feb. 2020, pp. 362–366.

[45] A. Gupta and V. B. Semwal, "Multiple task human gait analysis and identification: Ensemble learning approach," in *Emotion and Information Processing: A Practical Approach*, S. N. Mohanty, Ed. Cham, Switzerland: Springer, 2020, pp. 185–197.



LIMENG LU received the B.Eng. degree from Southwest University, Chongqing, China, in 2020. He is currently pursuing the M.Eng. degree with Northwest Minzu University, Lanzhou, China. His research interests include pervasive computing and deep learning.

CHUANLIN ZHANG was born in China, in 1998. He received the B.Eng. degree from the Qingdao University of Technology, Qingdao, China, in 2020. He is currently pursuing the M.Eng. degree with Northwest Minzu University, Lanzhou, China. His research interests include pattern recognition and deep learning.

KAI CAO was born in China, in 1998. He received the B.Eng. degree from Inner Mongolia University For Nationalities, Tongliao, China, in 2020. He is currently pursuing the M.Eng. degree with Northwest Minzu University, Lanzhou, China. His research interests include medical image processing and deep learning.

TAO DENG received the B.Eng. and M.S. degrees in computer science from the Lanzhou University of Technology, China, in 2007 and 2010, respectively, and the Ph.D. degree from Lanzhou University, China, in 2013. He is currently a Professor with the School of Mathematics and Computer Science, Northwest Minzu University, Lanzhou, China. His research interests include interdisciplinary research of computational biology, intelligent computing, and streaming data analysis.

QIANQIAN YANG is currently pursuing the B.S. degree with Kunming Medical University, Kunming, China. Her research interest includes information rehabilitation care.