Activity Graph based Convolutional Neural Network for Human Activity Recognition using Acceleration and Gyroscope Data

Article *in* IEEE Transactions on Industrial Informatics · January 2022

DOI: 10.1109/TII.2022.3142315

4 authors, including:

Po Yang
The University of Sheffield
153 PUBLICATIONS **3,013** CITATIONS

SEE PROFILE

Vitaveska Lanfranchi
The University of Sheffield
63 PUBLICATIONS **656** CITATIONS

SEE PROFILE

Fabio Ciravegna
The University of Sheffield
10 PUBLICATIONS **55** CITATIONS

SEE PROFILE

Activity Graph based Convolutional Neural Network for Human Activity Recognition using Acceleration and Gyroscope Data

***Abstract*— Human activity recognition (HAR) using smartphone sensors have been recently studied in various applications including healthcare, fitness, smart home, etc. Their recognition accuracy often depends on high-quality feature design and effectiveness of classification algorithms, where existing work mostly replies on laborious hand-crafted design and shallow feature learning architecture. Recent deep learning techniques demonstrate outstanding effectiveness in performing automatic feature learning and outperform traditional models in terms of accuracy. But their performance is limited by the quality and volumes of available labelled data. It is challenging to achieve accurate multi-subject HAR with only smartphone sensing data. This paper proposes a novel optimal activity graph generation model incorporating a deep learning framework for automatic and accurate HAR with multiple subjects using only acceleration and gyroscope data. The activity graph generation model presents a multisensory integration mechanism with three-steps sorting algorithms for producing optimal activity graphs containing alignments of neighbored signals in their width and height. Then, we propose a deep convolutional neural network to automatically learn distinguishable features from the graphs for HAR. By leveraging superior presentation of correlations between human activities and neighbored signals alignments via optimal activity graphs, the learned features are endowed with more discriminative power. The experimental evaluation was carried out on several benchmark datasets (i.e., UCI, USCHAD and UTD-MHAD). The results showed that our approach improved the average recognition accuracy by about 5% when compared with other state-of-the-art HAR methods. Particularly towards multi-subject HAR cases (UTD-MHAD dataset with 21 subjects), it achieved up to 10% accuracy gain over other methods. These improvements show the advantage and potential of our method dealing with complex HAR problems with multiple subjects using limited sensing data.**

***Index Terms*— Human activity recognition, deep learning, activity graph**

I. INTRODUCTION

With the rapid development of microelectronics and pervasive computing in the past decade, human activity recognition (HAR) using wearable and mobile computing technologies has been playing an increasingly important role in many fields from personlised healthcare to behavior analysis [1-2]. Particularly towards treatment and long-term care of many physical inactivity associated diseases like Parkinson's disease and diabetes, effectively monitoring and accurately recognising patients' daily physical activity (PA) using cost-effective mobile devices is helpful to achieve identification of abnormal activities [3] and prevention of serious consequences [4]. Also, HAR-related mobile applications enable providing reasonable exercise advice and fitness level reports [5]. Design and development of innovative and cost-effective mobile HAR approaches have much significance many health-related fields.

In earlier studies of HAR, optical sensing solutions [6-8] like using camera or depth sensors are one of the most popular cost-effective technologies to monitor and recognise human activity in many applications. Comparing with other HAR solutions [9-10], optical sensing approaches only require a small amount of low-cost camera to acquire video sources, and use advanced video analysis techniques for robust and accurate performance. But their usage suffers from many social limitations and technical challenges such as personal privacy, environmental illumination, video resolution, and cost and complexity of video processing algorithm. These limitations drive research and development of new cost-effective HAR technologies.

Mobile sensing based HAR approaches rely on three steps: 1) data acquisition using smartphone sensors; 2) distinguished feature extraction; and 3) feature learning and classification. Most previously existing works need to design laborious hand-crafted features (i.e., time, frequency or hybrid domains) [11] and classification algorithms (i.e., SVM, Random Forest Tree, ANN, weighted support tensor machines) [12-13] [31-32]. While these approaches show excellent accuracy in many well-calibrated lab-setting scenarios, their performance are often constrained by quality of hand-crafted feature, expensive data labelling process and strenuous experimental protocols for data collection. Recently, deep learning techniques [14-16] demonstrate outstanding abilities of modelling high-level abstractions of data and achieving automatic and robust feature learning. Typically, a deep neural network architecture with multiple layers is built up for automating feature design. Each layer in the deep architecture performs a non-linear transformation on the outputs of the previous layer; the data are represented by a hierarchy of features from low-level to high-level through the deep learning models. An effective mode could first represent raw signal data as 2D image or graphs containing visual features [17], and then apply well-known deep learning models like Convolutional Neural Networks (CNN) [14-15] and Long Short-Term Memory (LSTM) [18] for processing these data. But in this mode, their performance is limited by the types, quality and volumes of

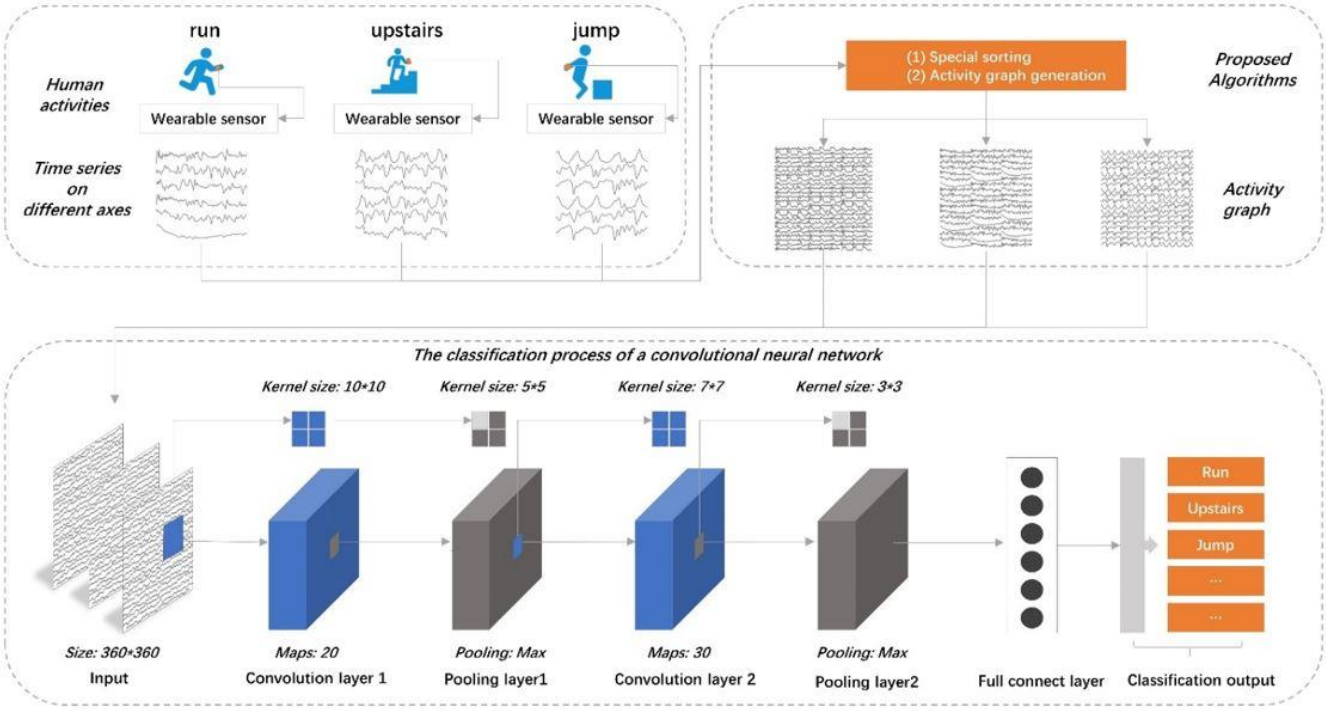


Fig.1. Technical pipeline of our approach architecture.

available labelled sensing data. Importantly, these 1D time series data captured by smartphone sensors require further processing for better presenting human activities exist in the three-dimensional space, such as a critical process of converting multiple 1D time series signal into 2D activity images or graphs. How to represent the correlation between activity subjects and signals alignments in 2D data will affect recognition accuracy of deep learning models. So far, there has been little attention to the issue of finding ways for deep learning to reach high accuracy of multi-subject HAR using only smartphone sensing data.

Targeting at above issues, this paper focuses on studying and developing novel activity graph generation mechanisms with superior presentation of correlations between human activities and neighbored signals alignments. It also incorporates a deep convolutional neural network (CNN) to improve multi-subject HAR performance. In a typical way [19], activity graphs were generated by simply placing and fusing all axis signals in one batch, where it possibly ignores some latent 'correlation'. Our idea is based on an assumption that there should be some latent '*correlations*' between human activity subjects and alignments of multiple sensing signals, where each activity subject should have some patterns simultaneously reflecting into individual axis of sensor in smartphones. Thus, following the technical pipeline shown in Fig.1, we aim to design a multisensory integration mechanism with sorting algorithms for producing optimal activity graphs containing alignments of neighbored signals in their width and height. Then, a deep convolutional neural network is proposed to enable automatic learning of distinguishable features from optimal activity graphs for HAR. By leveraging superior presentation of correlations between human activities and neighbored signals alignments via optimal activity graphs, the learned features are endowed with more discriminative power for achieving high accuracy and robustness of multi-subject HAR. Our key contributions are summarized below:

• A novel optimal activity graph generation approach with multisensory integration mechanisms and three-steps sorting algorithms is proposed to better present correlations between multiple human activity subjects and multiple sensing signals alignments. The generated optimal activity graphs contain rich and distinguished correlation information for learning features.

• A deep convolutional neural network with optimised parameters is designed for processing activity graphs with high accuracy of human activity recognition on smartphone sensors data. This network utilizes CNN to explore important latent features along both width and height of activity graphs simultaneously, further improving classification performance.

• A comprehensive experimental evaluation and analysis is given on three benchmark datasets (i.e., UCI [21], USCHAD [22] and UTD-MHAD [23]). The results show our proposed approach can averagely improve recognition accuracy of 3-5% compared with other state-of-the-art approaches. Towards some complex type HAR cases (UTD-MHAD dataset with 21 activity types), it achieves up to 10% accuracy gain over other methods. These improvements show potential of our method dealing with complex HAR problems with multiple subjects using limited sensing data.

The rest of the paper is organized as follows. Section II presents related work. Section III gives an overview to our approach; the technical details of our system are introduced in Section IV and V. Section VI describes and discusses the experimental results. Section VII gives conclusions.

II. Related Work

Recent deep learning techniques [14-16] in HAR studies mainly focus on two issues: fusion strategies of multi-sensor data, and optimisation of network architecture. In the first part,

some well-known models like Convolutional Neural Networks (CNN) [14-15] and Long Short-Term Memory (LSTM) [18] were used for processing multi-sensor data fusion for activity recognition. The work [19] suggested a classification method that fused different axis signals into an activity graph, and took it as input information of a deep convolutional neural network for recognising human activities. In [24], researchers used the Gramian Angular Fields (GAF) to encode one time series into a two-channel image, and applied a fusion ResNet framework for HAR. It is worth noting that some similar studies suggested some feature images generation approaches, where encoders were used to extract arrays or vectors similar to local images, and corresponding neural network classifiers were constructed to carry out the task of HAR. Ronao and Cho [25] proposed a deep convolutional neural network based HAR system. They constructed the original signal into arrays of six channels and used deep convolutional neural networks to extract relevant features from raw data for classification. Also, a temporal fast Fourier transform was applied in their approach to process original sensor data for enhancing the performance of neural networks. These methods prove that it is feasible and efficient to extract the information of the original data from the image for activity classification.

In the second section, many HAR studies attempted to study how to optimise the structures of deep networks to identify features and automatically complete the classification method. Advanced CNN structures such as GoogleNet [26], ResNet [27] and ZFNet [28] are all capable of achieving outstanding results in the HAR fields when given large volume of data. These methods usually designed special structures to solve the most common problem of gradient disappearance or explosion during model training. Additionally, due to a strong ability of processing time series data, LSTM networks were also widely studied in the HAR. Tao et al. [29] presented an improved LSTM method, called bidirectional long short-term memory (BLSTM). They first converted the raw data from the sensor into the norm of the horizontal component and vertical component, and then applied a multicolumn BLSTM with different signals to improve the performance of the classifier. Their work shows that utilising common wearable sensors and simpler architecture of neural networks can potentially achieve better generalization in the HAR.

III. Proposed Approach

A. *Brief summary of the model*

As mentioned before, our key idea is based on an assumption that there should be some latent '*correlations*' between human activity subjects and alignments of multiple sensing signals. As shown in Fig.1, we will first get raw accelerometer and gyroscope sensing data from smartphone for preprocessing; where they are presented as time series data of different axes of X, Y and Z of smartphone sensors. Then, we will use a series of sorting algorithms to generate a baseline of activity graph and an optimal activity graph with special sorting and stacking operations. The optimal activity graphs will be taken as the input of our optimized CNN approach. The classification result of HAR will be finally obtained.

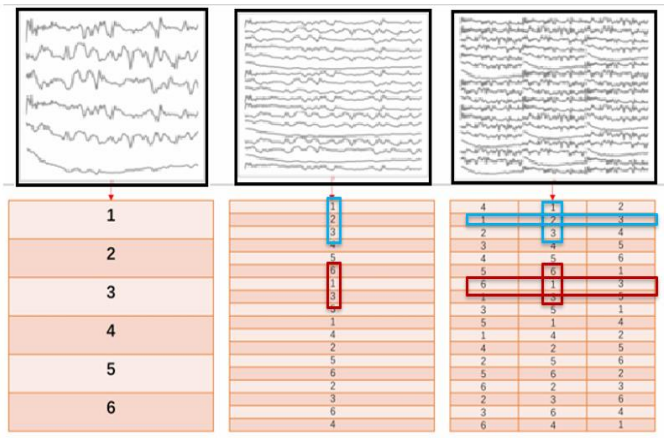


Figure 2. Our idea of activity graph generation model.

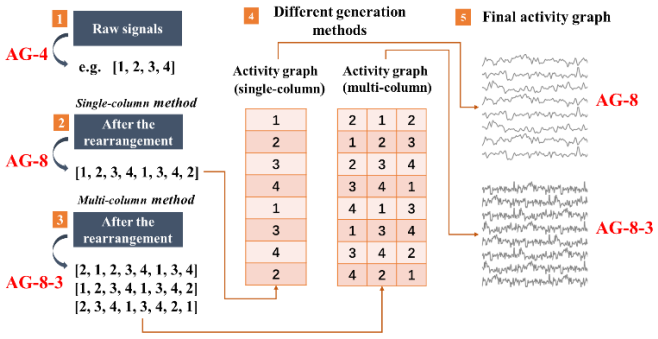


Figure 3. Demonstration of generating an Activity Graph with inputting 4 raw signals (AG-4: Activity Graph (1 x 4), AG-8: Activity Graph (1 x 8), AG-8-3: Activity Graph (3 x 8)).

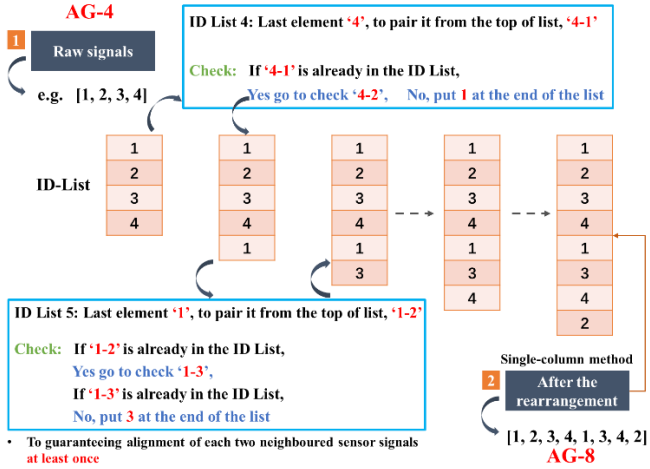


Fig.4. Activity Graph Generation with single-column method.

B. *Activity Graph Generation*

As for activity graph generation, the first issue is to determine its dimension of height. This paper only considers smartphone sensors, thus we can get six original data sequences from both accelerometer and gyroscope sensors in the X, Y, Z axis, as same as the datasets of USCHAD and UTD-MHAD shown in Fig2. However, the UCI dataset provides data after eliminating gravity component on the axis of the three components of the accelerometer, so each sample of UCI has nine data sequences.

A baseline activity graph generation method with a sorting algorithm was proposed by Jiang and Yin [19]. They highlight

certain latent 'correlation' between activity subjects and neighbored signals alignments in the height dimension of activity graphs. But their method only supports UCI dataset with 9 input sequences. Thus, as shown in Fig2, we improve their method as a baseline solution by taking any number of input sequences with a sorting algorithm named *single-column activity graph method*. Also, for producing optimal activity graphs containing alignments of neighbored signals in their width and height, another activity graph generation algorithm containing three columns in each activity graph is proposed and named *multi-column activity graph method*. Fig.2 shows a sample of taking 6 input sequences for generating baseline and optimal activity graphs. Also, another demonstration of generating an Activity Graph with inputting 4 raw signals is shown in Fig.3.

C. *Single-column activity graph method*

Jiang and Yin [19]'s algorithm is limited to the number of input sequences of 9 in UCI dataset, in other cases, the algorithm will give unexpected results. An activity graph stack algorithm that can accommodate any number of inputs is necessary, as different datasets are likely to use a variable number of sensors to collect data. For example, if we use a dataset containing signal data from one accelerometer and one gyroscope, then we use 6 different signals data (X, Y, and Z axis of accelerometer and the gyroscope) as raw input information, the resulting sequence by [19] is not completely sorted.

For solving this problem, we improved single-column method algorithm consisting of *Algorithm 1* and *2*. Algorithm1 outputs a specific data stack order based on the input original signal, Algorithm2 stacks the original signal sequentially into a graph based on the output of Algorithm1. The core idea is to stack the original data sequence row-by-row and make sure that each sequence is adjacent to any other sequence at least once. Note that the activity graph obtained using Algorithm2 is an internal distribution of single columns and multiple rows. Following the example in Fig.3, when giving 4 raw signals with AG-4, the activity graph generated by single-column method should guarantee alignment of each two neighbors sensor signals at least once. Finally, it will output an activity graph with AG-8. The procedure is shown in Fig.4.

D. *Multi-column activity graph method*

As single-column method only duplicates data along height of an activity graph, we further proposed another algorithm that generates a new activity graph to guarantee alignment of each two neighbored sensor signals at least once in its width and height. We called it as a multi-column method consisting of Algorithm-1 and Algorithm-3. Following the example in Fig.3 and 4, when giving an output activity graph AG-8 from Fig.4, the activity graph generated by multiple-column method should guarantee alignment of each two neighbors sensor signals at least once in its width and height. It will output an activity graph with AG-8-3. The procedure is shown in Fig.5.

Note that the activity graph obtained using Algorithm3 is an internal distribution of three columns and multiple rows. The core design idea of Algorithm3 is that the distribution of data sequences should be distributed as far as possible in higher dimensions, that is, not limited to a single column like Algorithm2, i.e., not only by stack signal sequence row-by-row, for each row of a single signal sequence, in its left and

Algorithm 1: Raw signals → Signal index sequence

```
Input: Id-list, Output-string;
/* Id-list is a list of the raw signal ids.  Output-string
   is a sequence of signal stacks. If there are six raw
   signals, Id-list is initialized to [1,2,3,4,5,6], then
   Output-string is initialized to '1,2,3,4,5,6'.          */
Output: Output-string;
Function IfComplate(Id-list,Current-string):
    // set is a null list;
    for i ∈ Id-list do
        for j ∈ Id-list do
            if i≠ j then
                Append 'i, j' to set
    return-value = 0;
    for k ∈ set do
        k1 = The first element of k;
        k2 = The second element of k;
        if k ∉ Current-string and 'k2, k1' ∉ Current-string then
            return-value = -1
    return return-value;
idx = the last item in the Id-list;
add = 1;
while IfComplate(Id-list,Output-string) = -1 do
    if idx = add then
        add = add +1;
        if add > length of Id-list then
            add = add - 2;
            Append ', add' to Output-string;
            idx = add;
            add =1;
    if 'idx, add' ∈ Output-string or 'add, idx' ∈ Output-string then
        add = add +1;
        if add > length of Id-list then
            idx = idx + 1;
            add =1;
            while 'idx, add' ∈ Output-string or 'add, idx' ∈
             Output-string do
                add = add + 1;
                if idx = add then
                    add = add + 1
                if add > length of Id-list then
                    idx = idx + 1;
                    break;
            add = 1;
            Append ', idx' to Output-string;
    else
        Append ', add' to Output-string;
        idx = add;
        add = 1;
return Output-string;
```

Algorithm 2: Signal index sequence → single-column activity graph

```
Input: Id-list
/* Id-list is the output Algorithm 1.  It indicates the
   order in which signals are stacked.                     */
Output: Activity graph
begin
    Activity graph is a blank graph.
    for i ∈ Id-list do
        the time series waveform of the i_th signal sequence is stacked to
         the bottom of the activity graph
/* Activity graph is a composite graph of one-column,
   multi-rows.                                             */
```

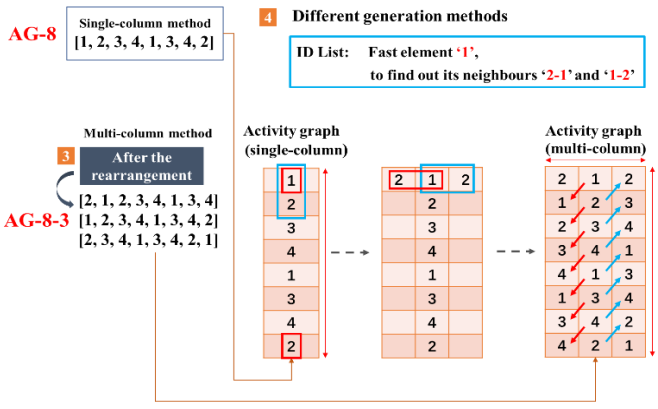


Fig.5. Activity Graph Generation with multi-column method.

Algorithm 3: Signal index sequence → multi-column activity graph

Input: Id-list

/* Id-list is the output of Algorithm 1. It indicates the order in which signals are stacked. */

Output: Activity graph

begin

Activity graph is a blank graph and be divided into three columns.

// for the first column.

for $i \in$ *Id-list* **do**

j = the previous id of i in the Id-list.

// when i is the first id in the Id-list, j is the last id

the time series waveform of the j_{th} signal sequence is stacked to the bottom of the first column of activity graph.

// for the second column.

for $i \in$ *Id-list* **do**

the time series waveform of the i_{th} signal sequence is stacked to the bottom of the second column of activity graph.

// for the third column.

for $i \in$ *Id-list* **do**

j = the next id of i in the Id-list.

// when i is the last id in the Id-list, j is the first id

the time series waveform of the j_{th} signal sequence is stacked to the bottom of the third column of activity graph

/* Activity graph is a composite graph of multi-columns, multi-rows. */

right, respectively inserted into the signal sequence. At the same time, for each data sequence, not only keeping it on row direction and adjacent to other sequences at least once, but also on the direction of the column to guarantee with other sequences adjacent at least once.

IV. Principle of Activity Graph Design

As shown in Fig.2, the design principle of optimal activity graph is to enable containing more latent correlation presenting between human activity subjects and neighbored signals alignments. To better explain the concept of the" correlation", we used the simplest activity graph generated by the original unordered method, the activity graph generated by single-column method, compared with the activity graph generated by multi-column method, as shown in Fig.6.

The most important part of using a CNN for activity graph operation is to use the convolution kernel for convolution operation. Specifically, a fixed size convolution kernel moves the image horizontally and vertically until it reaches the end point. In this process, the information contained in the activity graph is aggregated into the final convolutional layer through corresponding matrix operations. The selection of convolution kernel will directly affect the results of information extraction. In our experiment, we use a convolution kernel of size 10x10 for the convolution operation of the active graph. We used the activities in the USCHAD data set to obtain the final activity graph through three different generation algorithms. The size of the graph remains consistent 360x360. Moreover, we extracted the number of the real signal sequence in the activity graph to construct a corresponding order graph for each activity graph, as shown in the Fig.2. Then, when the convolution kernel with a size of 10x10 moves over these images, it shows that the" correlation" obtained by convolution kernel is different, as shown in Fig.6 to make a specific comparison. For the activity graph from original unordered method, no matter where the convolution kernel moves to, the maximum number of original signals that a single convolution kernel can recognize at the same time is 2, as shown in Fig.6.

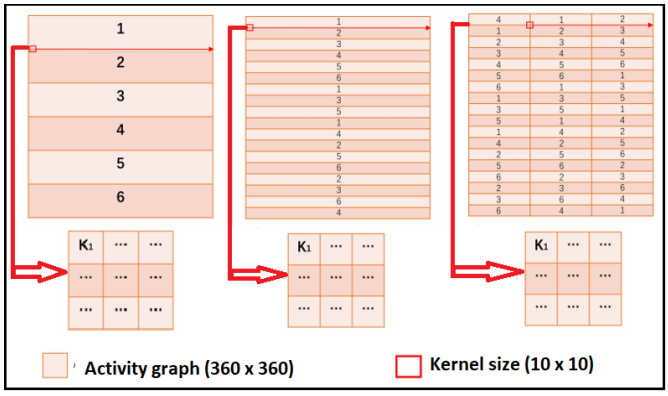


Figure 6. Convolution operation on different activity graphs

As all signals are not extended sorted, many correlations between signals will be lost, such as signals 1 and 3, signals 2 and 4, etc. This method contains the least "correlation" information, and the recognition effect is theoretically the worst.

For the single-column method, the maximum number of original signals that a single convolution kernel can recognize at the same time is 2, as shown in Fig.6. However, due to the extended sorting, all signals are adjacent to any other signals at least once, which solves the problem of large amount of information loss in the unsorted method. This method contains more" correlation" information, and the recognition effect can be improved theoretically. In the multi-column method, the maximum number of original signals that a single convolution kernel can recognize at the same time is improved, we can see in Fig.6 when the 10x10 convolution kernel moves to the signal junction, the convolution operation can be performed on up to four signals (containing three different signals, such as signal 4,1,1,2) at the same time, and our method remains the signal extend sorting. Our proposed method can contain more "correlation" information than the single-column method, which further improves the theoretical recognition effect.

It is worth noting that, in theory, if the size of the convolution kernel is increased, more different signals can be recognised simultaneously. But the "correlation" information does not necessarily become richer, as too large a convolution kernel will aggregate too much original information of the image during the convolution operation, resulting in the loss of information. So, a balance needs to be reached between the size of the activity graph and the size of the convolution kernel so that more" correlation information" can be contained and better recognition effect can be achieved. According to our experimental tests, for the activity graphs (default width and height are equal, both are m), the most appropriate convolution kernel size interval is *[m/50, m/30]*. At the same time, based on our theoretical explanation of the proposed method, for more complex activities, the more complex the activity pattern on each decomposition axis is, the more" correlation" needs to be recognized to better avoid false recognition. It means that our method should be able to deal with some complex cases with many activity subjects.

V. Design and Optimisation of Deep Network

We use CNN in deep learning technology to automatically extract potential features in activity graphs. CNN is mainly composed of convolution layer, activation function and

pooling layer. And the dimension of our input (activity graph) is $W \times H \times C$, where W, H, and C are width, height and the number of channels, respectively.

Convolutional layer: The typical convolutional layer uses the kernel (also called filter) to perform mathematical processing. The kernel size is usually $f \times f$, the other four important parameters are the number of input feature map, the number of output feature map, padding P and stride S. Note that the number of channels C must be equal to that of input feature map so the convolution operation be performed correctly. The output size $(W^{l_n} \times H^{l_n})$ of convolution layer l_n is shown in Eq.(1, 2).

$$W^{l_n} = \frac{W^{l_{n-1}} + 2P^{l_n} - f^{l_n}}{S^{l_n}} + 1 \qquad (1)$$

$$H^{l_n} = \frac{H^{l_{n-1}} + 2P^{l_n} - f^{l_n}}{S^{l_n}} + 1 \qquad (2)$$

The final output of the convolutional layer l_n is shown in Eq.(3).

$$X_j^{l_n} = \varphi\left(\sum_{i=1}^{M} X_i^{l_{n-1}} * f_{ij}^{l_n} + b_j^{l_n}\right) \qquad (3)$$

In Eq.(3), b is bias term, i and j are indexes of input and output feature maps of convolutional layer. M means the range of filter values.

Activation function: Rectified linear activation function (ReLU) is usually selected as the activation function after the convolution operation, the purpose of using ReLU is to introduce non-linearity because the CNN needs to learn nonnegative linear values. In the Eq.(3), φ is the ReLU function, is shown in Eq.(4).

$$\varphi(x) = max(0, x) \qquad (4)$$

In addition to this, finally, all data is entered into a full connection layer and applying the final activation function (typically Sigmod in Eq.(5) or Softmax in Eq.(6)) to get the prediction results, C represents the number of classes in a multi-classification problem.

$$\varphi(x) = \frac{1}{1 + e^{-x}} \qquad (5)$$

$$\varphi(x_i) = \frac{e^{x_i}}{\sum_{c=1}^{C} e^{x_c}} \qquad (6)$$

Pooling layer: CNN uses pooling layers to reduce the number of parameters significantly. The most commonly used pooling layers include average pooling and max pooling. Here, we use max pooling as pooling layer. The output size $(W^{l_n} \times H^{l_n})$ of max pooling layer l_n is equal to the Eq.(1, 2). Specifically, the Max Pooling layer preserves the maximum value within each kernel range and discards other values as the final output.

In order to select an appropriate CNN as our baseline classifier, we compared three advanced CNNs: ResNet, GoogleNet and LeNet, the comparing result is shown in Table. I. We can see that although ResNet and GoogleNet are more complex, to do better in some image classification problems, but in our experiment, due to the sample of dataset is less (from 5000 to 14000, small amounts of data are also a common feature of HAR data sets), for ResNet and GoogleNet, which are attempting to solve the problem of large-scale data recognition, unable to play to their advantages, and their training time is longer. So we finally used LeNet as the baseline classifier in subsequent experiments due to its good performance and lower training consumption. And we finally used the structure of CNN is shown in the bottom of Fig.1, the detailed parameters are shown in the Table II.

TABLE I PERFORMANCE COMPARISON OF DIFFERENT CNNS

CNN architecture	UCI	USCHAD	UTD1
LeNet	86.2%	83.1%	54.2%
GoogleNet	85.7%	81.9%	53.5%
ResNet	84.5%	82.5%	52.8%

TABLE II THE PARAMETERS AND HYPERPARAMETERS OF OUR CNN

Parameters	Value
Input layer size	360 x 360
The kernel size of convolutional layer 1	10
The kernel size of convolutional layer 2	7
The number of output maps of convolutional layer 1	20
The number of output maps of convolutional layer 2	30
The kernel size of convolutional layer 2	7
The type of subsampling layer	Max-pooling
The kernel size of subsampling layer 1	5
The kernel size of subsampling layer 2	3
Learning rate	0.0001
Optimizer type	Adam
Batch size	256
The number of epochs	1000
The dropout rate	0.1

TABLE III DATASET DESCRIPTION

Dataset	Sensors	Position	Data Subjects	Sampling rate	Number of activities
UCI [21]	2 (Acc, Gyro)	Waist	30	50HZ	6
USCHAD [22]	2 (Acc, Gyro)	Hip	15	100HZ	12
UTDI [23]	2 (Acc, Gyro)	Wrist	8	50HZ	21

TABLE IV PROCESSING RESULTS AND PARAMETER SELECTION

Dataset	Sliding window overlap	Sampling time	Training set samples	Test set samples
UCI [21]	50%	2.5s	7352	2947
USCHAD [22]	50%	2s	18557	7954
UTDI [23]	50%	1s	3014	1293

VI. EXPERIMENTS EVALUATION AND RESULTS

A. *Experimental Settings*

We used three public datasets to validate the proposed method. The information of these datasets is summarized in Table.III. UCI dataset was collected from 30 volunteers within an age bracket of 19-48 years [21]. Each volunteer wore a Samsung Galaxy S II smartphone on the waist and performed six different activities. The researchers collected data from accelerometers and gyroscopes embedded in the smartphone, and videotaped the experiments so that activity types are manually labeled. The original sampling frequency is 50HZ. Extra, the sensor acceleration signal was separated into gravity and body acceleration parts based on a Butterworth low-pass filter. For the USCHAD [22] dataset, a sensing platform called MotionNode is used to collect data and this platform integrates a 3-axis accelerometer, 3-axis gyroscope, and a 3-axis magnetometer with the sampling frequency of 100HZ. The researchers selected 14 subjects within an age bracket of 21-49 years to collect data from 12 different activities, and they used observers to manually record and label these activities. In the UTD [23] dataset, the researchers used one Kinect camera and one wearable inertial sensor to collect data. The sampling rate of the wearable sensor is 50 HZ. The 8 objects are required to perform 27 different activities, it's worth noting that, for actions 1 through 21, the inertial sensor was placed on the wrist of subjects but for actions 22 through 27, the inertial sensor was placed on the subject's right thigh. In this paper, we only take the first 21 class activities of inertial sensor data for research, and we also do not use the camera data because we don't study optical sensor data, as called UTD1.

B. *Evaluation Metrics*

In order to accurately evaluate our model, Mean Average Precision (mAP) is used as metric for evaluation that takes mean of Average Precision (AP) value among classes. Given an IoU threshold, AP value used to fuse the precision and recall together and defined as the area under Precision-Recall (PR) curve:

$$AP(c) = \int PR(c) \tag{7}$$

where c denotes the class and PR is calculated by:

$$Precision(c) = \frac{\#TP(c)}{\#TP(c)+\#FP(c)} \tag{8}$$

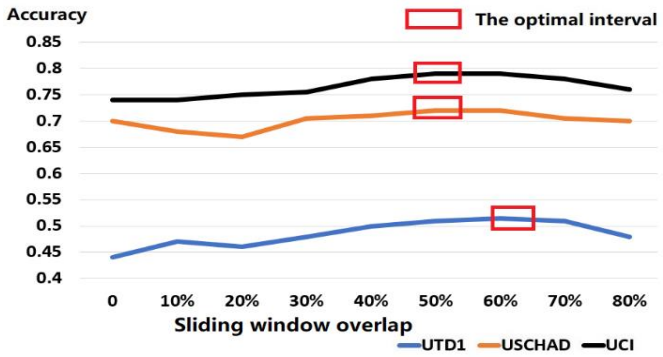
$$Recall(c) = \frac{\#TP(c)}{\#TP(c)+\#FN(c)} \tag{9}$$

in which TP, FP and FN represent True Positive, False Positive and False Negative samples respectively so the Precision measures the samples that are incorrectly detected while Recall measures those misdetection samples. Then the mAP could be obtained by taking mean:

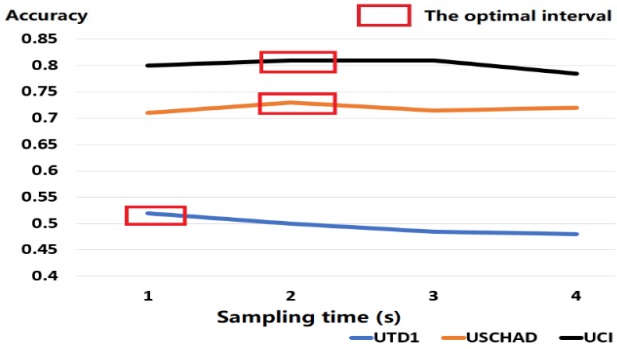
$$mAP = \frac{1}{|C|}\sum_{c \in C} AP(c) \tag{10}$$

C. *Parameter Optimisation*

In the process of data preprocessing, generating the activity graph, we respectively show the relevant parameter



a) accuracy of differnet sliding window overalp rate

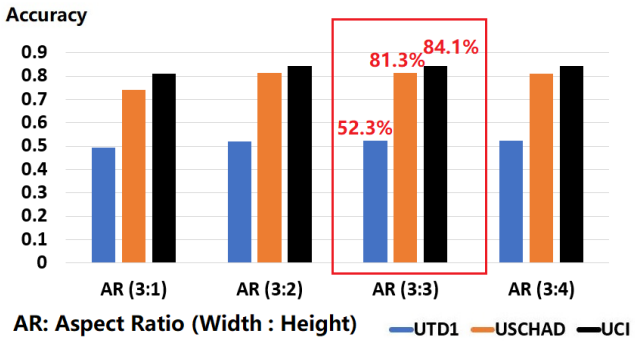


b) accuracy of differnet sampling time

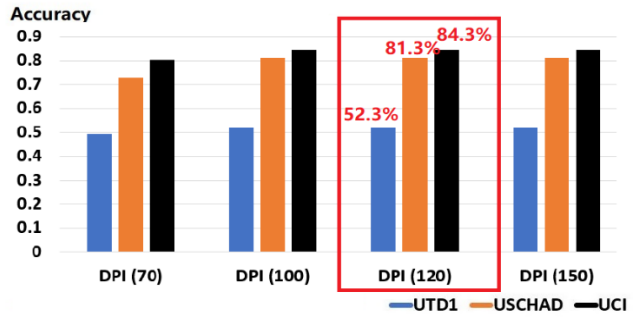
Figure 7. Comparison of accuracy with varied sliding window size and sampling time over three datasets.

optimization process and the corresponding results of these three parts. For all datasets, LeNet as a baseline classifier is used to get the classification accuracy with single-column method as a baseline. For data preprocessing, Fig.7 shows the different results obtained from the different selection of sliding window overlap and sampling time. For sliding window overlap in Fig.7a, the greater the overlap means that finally can use the sample will be more. It is of great value to the original data set sample amount is small, in UTD1 data set, due to its original sample amount relative to the other two data sets is less, so we can see the overlap take a larger value (65%), eventually the highest classification accuracy, for the other two data sets, one of the most commonly used overlap value (50%), can obtain the good effect. The length of sampling time also has a similar effect on the final number of samples. For UCI and USCHAD datasets, the sampling time around two seconds could achieve a good classification effect, as shown in Fig.7b. However, more than two seconds is too long for the UTD1 dataset, this results in too few samples available for training, one second is the relative optimal choice of UTD1 when both the number of samples available and the length of a single sample need to be considered. The final data preprocessing result is shown in Table IV.

For all datasets, there are two most important parameters, aspect ratio and dots per inch (DPI), to determine the final size and quality of the image when generating the activity graphs., Fig.8 show the different results obtained from the different selection of aspect ratio and DPI. For aspect ratio, we find that the ratio is equal to 3:3 or 3:4 with the highest accuracy. This is due to compression problems when data sequences are arranged into images. For images that have the same content and format except for the different proportions of width and height, we can see when aspect ratio is 3:2 (Width: Height is



a) accuracy of using differnet aspect ratio of AG generation



b) accuracy of using differnet DPI of AG generation

Figure 8. Comparison of accuracy with varied aspect ratio and DPI over three datasets.

3:2), the data sequence is stacked in rows, the height of the data sequence graph in each row is compressed because the number of rows is large, this could cause fluctuations in the original data to be partially obscured (i.e., locally lost information). In contrast, if we increase the height so that the height is suitable for the arrangement of multiple rows of data sequence, this situation will be greatly improved, conducive to the subsequent recognition. But blindly increasing the height is not a good choice because it makes the activity graph file too large, considering that we need to save and read a lot of images later, the ratio of 3:3 was used as our final choice in this paper (the actual pixels used are 360:360). Moreover, when data sets need to generate activity graphs with more rows, we recommend using larger heights such as 3:4 or 3:5 as appropriate. In total, for HAR, too small aspect ratio will cause loss of details of picture information, while too large aspect ratio will cause unnecessary waste of computing resources. The appropriate aspect ratio should be 3:3 or 3:4.

Similarly, we also tested the choice of different DPI, the result was shown in Fig.8b. The value of DPI will directly affect the picture quality of the activity graphs, and too much DPI will cause unnecessary space resource occupancy, we finally choose DPI equal to 120 for the subsequent experiments. Too small DPI will cause loss of details of picture information, while too large DPI will cause unnecessary waste of computing resources. We recommended DPI is 120 as the generally choice.

D. *Comparison of the Single-column method and the proposed method*

The classification performance of the three public datasets using single-column activity graphs and multi-column activity graphs (our proposed method) is shown in the Table V. For each dataset, the method we proposed has achieved better classification accuracy compared with the single-column activity graphs, the classification accuracy was improved by 3.96%, 4.56% and 9.93% respectively. It proves the effectiveness of the algorithm we designed, in other words, for different activity types and different number of original signal sources, our method can generate activity graphs containing more potential features, thus effectively improving activity recognition accuracy.

Moreover, we can see with the improvement of the number of activities on these datasets (21 activities >12 activities >6 activities), the degree of improvement of classification accuracy also increases (9.93% >4.56% >3.96%), these results suggest a possibility, that is, our proposed method has greater potential to recognize data with more activity types. To verify this hypothesis, we randomly selected different types of activities in the UTD1 dataset to generate sub-datasets, and used two feature graph generation methods respectively to obtain classification accuracy data, which were then compared with the original UTD1 recognition data.

We also found that when the number of activities decreases to 12, the accuracy of Multi-column activity method increase to 72.02% and the accuracy of single-column activity method increase to 64.57%, but the improvement of Multi-column methods decrease to 7.45%, a similar test result was presented when the number of activity types was equal to 16 (the improvement of Multi-column methods decrease to 8.18%), in contrast, the number is 9.93% when the number of activity is 21 on UTD1 dataset. This verifies our hypothesis that the performance improvement of our proposed method is more significant on datasets with more subject activities.

In comparison with two state-of-the-art deep learning techniques [35][36], we found that our proposed multi-column method performs not good as [35][36] in UCI datasets with 6 activity subjects. It is because the two state-of-the-art deep learning techniques have proposed new SE blocks and SK convolution to optimise the kernel of CNNs, so that it could achieve up to 96.60% accuracy in UCI dataset.

TABLE V PERFORMANCE COMPARISON OF OUR PROPOSED METHOD

Activity Graph generation method	UCI [21] (6 activities)	USCHAD [22] (12 activities)	UTD1[23] (21 activities)
Single-column	86.21%	83.14%	54.19%
Multi-column	**90.17%**	**87.70%**	**64.12%**

TABLE VI PERFORMANCE COMPARISON WITH OTHER STATE-OF-THE-ART METHODS

	UCI [21] (6 activities)	USCHAD [22] (12 activities)	UTD1[23] (21 activities)
Our multi-column method	90.17%	87.70%	**64.12%**
	RF 91.31%	LR 76.08%	LR 15.54%
	SVM **96.47%**	**J48DT 91.37%**	J48DT 48.57%
	ABDT 91.31%	ABDT 90.21%	ABDT 51.42%
[35]	94.51%	87.36%	61.53%
[36]	96.60%	86.70%	60.12%

TABLE VII PERFORMANCE COMPARISON OF COMPUTATIONAL COST

Dataset	UCI [21] (6 activities)		USCHAD [22] (12 activities)		UTD1[23] (21 activities)	
Methods	**Single column**	**Multi-column**	**Single column**	**Multi-column**	**Single column**	**Multi-column**
Accuracy	86.21%	90.17%	83.14%	87.70%	54.19%	64.12%
Precision	86.01%	87.62%	81.19%	85.59%	53.94%	63.69%
Recall	84.64%	88.15%	81.14%	85.78%	52.88%	63.31%
Computational Cost (ms)	0.52	0.54	0.54	0.54	0.52	0.54

However, as for the datasets [22][23] with more activity subjects, our proposed algorithms perform better than these two state-of-the-art algorithms, particularly in UTDI with 21 activities, our proposed method has outperformed them with 3% accuracy gain. This verifies our hypothesis that the performance improvement of our proposed method is more significant on datasets with more subject activities. Therefore, the above results show that our proposed approach performs better than these state-of-the-art deep learning approaches with selected kernel convolution, due to our optimised activity graph generation model.

E. *Comparison of other state-of-the-art methods*

We also compared the classification performance of the proposed method with that of other state-of-art methods on the three datasets. Most of these methods use manual feature extraction methods and variants or improvements based on traditional classifiers (such as SVM, random forest, etc.) for activity recognition. The result is shown in the Table VI. RF: Random forest tree; SVM: Support Vector Machine, ABDT: AdaBoost decision tree; J48DT: J48 Decision Tree; LR: logical regression;

Since most of state-of-art methods improve the traditional classifiers, so the first thing we evaluate the proposed method and the original traditional classifier performance difference, we used the UCI dataset have done manually extract with 561 features with four traditional classifiers (Bayesian, Random Forest, GBDT, SVM,) carried out the experiment, the 561 features include the characteristics of the time domain and frequency domain, is constructed from the original dataset author through expert knowledge and proved the validity of them. We compared our method with traditional feature extraction methods and traditional classifiers. The results show that our proposed methods can approach or exceed the performance of some traditional classifiers (our method: 90.17% > Bayesian: 85.00%). At the same time, the traditional manual feature extraction method can obtain the optimal classification result (SVM: 96.47%), moreover, by comparison with other state-of-art methods, the effect of our proposed method (90.17%) is also close to that of Casale et al. 's method based on the random forest classifier (91.31%) and Anguita et al. 's method based on the multiclass SVM classifier gets the best result (96.47%), which indicates that the proposed method is effective but not yet optimal. However, these analysis conclusions above just from UCI dataset contains only six common kinds of activities and the activities themselves are not complicated, not all of the datasets could be extracted specification and reasonable characteristics of as many as 561, in more cases, it is limited by the highly complex and more categories of the classification activity itself, from Table VI for USCHAD and UTD1 tests, we can get more information. In addition to accuracy, other important experimental results such as precision and recall are shown in Table VII.

F. *Computational cost*

All our experiments are conducted on a normal computer with a 2.7GHZ CPU and 8GB memory. When training the convolutional neural network, we used the Tesla P100 GPU for acceleration. When testing the test set to calculate various metrics and computational Cost, we did not invoke the GPU. Our computational cost average is 0.54ms per test sample, which is a very low time resource consumption and helps to perform real-time human physical activity recognition on low-power devices, especially mobile devices. The result is shown in the Table VII.

VII. DISCUSSION

While our proposed activity graph generation approach with multi-column sorting algorithms demonstrates a superior performance than existing state-of the-art deep learning algorithm [19] on the most complex UTD1 dataset, there are still some further issues requiring discussion and future study.

One main issue is the difference on how to transfer signals into pixel value between our method and [19]. In [19], their method is to directly map the original signal into pixel value and then generate activity graph through Discrete Fourier Transform (DFT). But our method including single-column and three-column algorithms is to generate activity graph by stacking the waveform of the original signal. We have re-produced the method in [19] with DFT pre-processing in a variety of experiments and tested the use of DFT for our proposed method, but the results show that the DFT does not improve (or even decrease) accuracy. It is probably because [19]'s method is a direct mapping of the original sensor readings to pixels, and the images generated by this method are more densely arranged, which is more suitable for the use of DFT. However, our method could better extract the 'correlation information' on different coordinate axes without DFT to extract the frequency domain information. So the method we proposed refers to idea from [19]: "*Every two signals must be adjacent once*", but has some significantly difference with their method. These two approaches also have different advantages. For data sets with fewer activity categories and uncomplicated activities, [19]'s method is more

effective. For data sets with more complex activities and more categories, for example in the dataset of UTD1, [19]'s method contain less information than our method. Our method is inspired by [19]'s idea, but towards more complex datasets with multiple subjects' activities.

Another notable issue is whether we could have potentially optimised the soring algorithms, as it is key to the success of the output of activity graphs. Current multi-column sorting algorithm contains duplicated and redundant information of activity graph. For instance, Figure.2 shows a case of activity graph with only 6 axis input. When the axis input is increased to 9, 12, or even more, we will have some more duplicated and redundant information in activity graph, further potentially affecting efficiency of our CNN solution. Also, our sorting algorithm is not the only unique solution for optimal activity graph, as it is dependent on initialization of axis signals input.

Lastly, one important issue is about selecting parameters of CNN such as kernel size. We find out that the 5x5 convolution kernel processes frequency domain data rather than direct correlation of signals on different axes in [19]. Catching the "right" amount of information with the adopted 10x10 convolution kernel is applicable to the method we proposed, where this kernel size could better extract the presentation of correlations between multiple activity subjects and sensor signals alignments. Our key contribution of this paper is to seek an importantly ignored problem in existing literatures of HAR, that generating optimal activity image can present correlations between multiple human activity subjects and sensor signals alignments, further improving its accuracy when applying deep learning techniques into activity images.

Notably, recent literatures [31-34] on HAR indicates some interesting new research progress, such as ambient sensing technologies [33] for smart home care, Kinect based human affection recognition technologies [34] for remote healthcare. Their practical applications in free-living environment are still limited, as using smartphone data is easier and more accessible to end-users. Thus, our proposed solution has importance in this field. To our best knowledge, it is the first time in the literature to point out the issue of activity graph generation using "latent 'correlation' between human activity subjects and neighboured signals alignments", also we prove its effectiveness in complex datasets of HAR with 21 subjects of activities, with up to 10% accuracy improvement.

VIII. CONCLUSION

This paper designed a novel optimal activity graph generation model by incorporating deep learning frameworks for automatic and accurate HAR with multiple subjects using only acceleration and gyroscope data. Specifically, through a comprehensive comparison, we confirm that the classification effect of our proposed multicolumn activity graph is better than other deep learning or traditional supervised learning HARs. The results showed that our approach averagely improved recognition accuracy about 5% compared with other state-of-the-art HAR methods. Particularly towards multi-type HAR cases, it achieved up to 10% accuracy gain over other methods. These improvements show the advantage and potential of our method dealing with complex HAR problems with multiple subjects using limited sensing data.

REFERENCES

[1] F. Lin, A. Wang, Y. Zhuang, M. R. Tomita and W. Xu, "Smart Insole: A wearable sensor device for unobtrusive gait monitoring in daily life", *IEEE Trans on Industrial Informatics*, Vol 12, Issue 6, pp.2281-2291, Jan 2016.

[2] L. Qi, C. Hu, X. Zhang, M. R. Khosravi, S. Sharma, S. Pang and T. Wang, "Privacy-aware data fusion and prediction with spatial-temporal context for smart city industrial environment", *IEEE Trans on Industrial Informatics*, Early Access, July 2020.

[3] G. Zhao, Y. Liu and Y. Shi, "Real-time assessment of the cross-task mental workload using physiological measures during anomaly detection", *IEEE Trans on Human-Machine Systems*, Vol 48, Issue 2, pp.149-160, Jan 2018.

[4] A. H. Kronbauer, H. C. Da Luz and J. Campos, "Mobile Security Monitor: a wearable computing platform to detect and notify falls", *IEEE Latin America Transactions*, Vol 16, Issue 3, pp.957-965, May 2018.

[5] Z. Li, S. Das, J. Codella, T. Hao, K. Lin, C. Maduri and C. H. Chen, "An adaptive, data-driven personalized advisor for increasing physical activity", *IEEE Journal of Biomedical and Health Informatics*, Vol 23, Issue 3, pp.999-1010, May 2019.

[6] X. Wang, K. Tieu and E. L. Grimson, "Correspondence-free activity analysis and scene modelling in multiple camera views", *IEEE Trans on Pattern Analysis and Machine Intelligence*, Vol 32, Issue 1, pp.56 -71, Jan 2010.

[7] A. Kamel, B. Sheng, P. Yang, P. Li, R. Shen and D.D. Feng, "Deep convolutional neural networks for human action recognition using depth maps and postures", *IEEE Trans on Systems, Man, and Cybernetics: Systems*, Vol 49, Issue 9, pp.1806-1819, July 2018.

[8] C. T. Chu and J. N. Hwang, "Fully unsupervised learning of camera link models for tracking humans across nonoverlapping cameras", *IEEE Trans on Circuits and Systems for Video Technology*, Vol 24, Issue 6, pp.979-994, Jan 2014.

[9] J. Qi, P. Yang, A. Waraich, Z. Deng, Y. Zhao and Y. Yang, "Examining sensor-based physical activity recognition and monitoring for healthcare using internet of things: a systematic review", *Journal of Biomedical Informatics*, Vol 87, pp.138-153, Nov 2018.

[10] J. Qi, P. Yang, L. Newcombe, X. Peng, Y. Yang and Z. Zhao, "An overview of data fusion techniques for internet of things enabled physical activity recognition and measure", *Information Fusion*, Vol 55, pp.269-280, March 2020.

[11] A. Mannini and S. S. Intille, "Classifier personalisation for activity recognition using wrist accelerometers", *IEEE Journal of Biomedical and Health Informatics*, Vol 23, Issue 4, pp.1585-1594, July 2019.

[12] Z. H. Chen, Q. C. Zhu, Y. C. Soh and L. Zhang, "Robust human activity recognition using smartphone sensors via CT-PCA and online SVM", *IEEE Trans on Industrial Informatics*, vol 13, issue 6, pp.3070-3080, Dec 2017.

[13] 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 Journal of Biomedical and Health Informatics*, Vol 22, Issue 4, pp.979-988, July 2018.

[14] J. H. Huang, S. S. Lin, N. Wang, G. H. Dai, Y. X. Xie and J. Zhou, "TSE-CNN: A two-stage end-to-end CNN for human activity recognition", *IEEE Journal of Biomedical and Health Informatics*, Vol 24, Issue 1, pp.292-299, Jan 2020.

[15] E. Kim, "Interpretable and accurate convolutional neural networks for human activity recognition", *IEEE Trans on Industrial Informatics*, Vol 16, Issue 11, pp.7190-7198, Nov 2020.

[16] D. Ravi, C. Wong, B. Lo and G. Z. Yang, "A deep learning approach to no-Node sensor data analytics for mobile or wearable devices", *IEEE Journal of Biomedical and Health Informatics*, Vol 21, Issue 1, pp.56-64, Jan 2017.

[17] D. F. Silva, V. M. De Souza, G. E. Batista, Time series classification using compression distance of recurrence plots, *in: 2013 IEEE 13th International Conference on Data Mining, IEEE, 2013*, pp. 687–696.

[18] F. J. Ord´o˜nez, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, *Sensors* 16 (1) (2016) 115.

[19] W. Jiang, Z. Yin, Human activity recognition using wearable sensors by deep convolutional neural networks, *in: Proceedings of the 23rd ACM international conference on Multimedia, 2015*, pp. 1307–1310.

[20] Z. Chen, L. Zhang, Z. Cao, J. Guo, Distilling the knowledge from handcrafted features for human activity recognition, *IEEE Transactions on Industrial Informatics* 14 (10) (2018) 4334–4342.

[21] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz, A public domain dataset for human activity recognition using smartphones., in:

Esann, Vol. 3, 2013, p. 3.
[22] M. Zhang, A. A. Sawchuk, Usc-had: a daily activity dataset for ubiquitous activity recognition using wearable sensors, in: *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 1036–1043.
[23] C. Chen, R. Jafari, N. Kehtarnavaz, Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor, in: *2015 IEEE International conference on image processing (ICIP)*, *IEEE*, 2015, pp. 168–172.
[24] Z. Qin, Y. Zhang, S. Meng, Z. Qin, K.-K. R. Choo, Imaging and fusing time series for wearable sensor-based human activity recognition, *Information Fusion* 53 (2020) 80–87.
[25] C. A. Ronao, S.-B. Cho, Human activity recognition with smartphone sensors using deep learning neural networks, Expert systems with applications 59 (2016) 235–244.
[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.
[27] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[28] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer, 2014, pp. 818–833.
[29] D. Tao, Y. Wen, R. Hong, Multicolumn bidirectional long short-term memory for mobile devices-based human activity recognition, *IEEE Internet of Things Journal* 3 (6) (2016) 1124–1134.
[30] A. Jordao, A. C. Nazare Jr, J. Sena, W. R. Schwartz, Human activity recognition based on wearable sensor data: A standardization of the stateof-the-art, arXiv preprint arXiv:1806.05226 (2018).
[31] Z. Ma, L. Yang, M. Lin, Q. Zhang and C. Dai, "Weighted Support Tensor Machines for Human Activity Recognition with Smartphone Sensors", *IEEE Trans on Industrial Informatics*, early access, 2021.
[32] Z. H. Chen, C. Y. Jiang and L. Xie, "A Novel Ensemble ELM for Human Activity Recognition Using Smartphone Sensors", *IEEE Trans on Industrial Informatics*, vol 15, issue 5, pp.2691-2699, May 2019.
[33] M. Kaur, G. Kaur, K. Sharma, A. Jolfaei and D. Singh, "Binary cuckoo search metaheuristic-based supercomputing framework for human behavior analysis in smart home", The Journal of Supercomputing, vol 76, pp.2479-2502, 2020
[34] U. Tripathi, R. S. J, V. Chamola, A. Jolfaei and A. Chintanpalli, "Advancing remote healthcare using humanoid and affective systems", IEEE Sensors Journals, early access, Jan. 2021.
[35] R. Abdel-Salam, R. Mostafa, and M. Hadhood, "Human activity recognition using Wearable Sensors: Review, Challenges, Evaluation Benchmark", the 2nd International Workshop on Deep Learning for Human Activity Recognition, Held in conjunction with IJCAI-PRICAI 2020 , Jan. 2021.
[36] W. Gao, L. Zhang, W. Huang, F. Min, J. He and A. Song, "Deep Neural Networks for Sensor-based Human Activity Recognition using Selective Kernel Convolution", IEEE Trans. Instrum. Meas, Vol 70. 2021.