2018 24th International Conference on Pattern Recognition (ICPR)
Beijing, China, August 20-24, 2018

Human Activity Recognition Based On Convolutional Neural Network

Wenchao Xu, Yuxin Pang, Yanqin Yang
Shanghai Key Laboratory of Multidimensional Information Processing
East China Normal University
Shanghai, People's Republic of China
E-mail: wchxu@ce.ecnu.edu.cn

Yanbo Liu
SJTU Engineering Training Center
Shanghai Jiao Tong University
Shanghai, People's Republic of China

***Abstract*—Smartphones are ubiquitous and becoming increasingly sophisticated, with ever-growing sensing powers. Recent years, more and more applications of activity recognition based on sensors are developed for routine behavior monitoring and helping the users form a healthy habit. In this field, finding an efficient method of recognizing the physical activities (e.g., sitting, walking, jogging, etc) becomes the pivotal, core and urgent issue.**

In this study, we construct a Convolutional Neural Network (CNN) to identify human activities using the data collected from the three-axis accelerometer integrated in users' smartphones. The daily human activities that are chosen to be recognized include walking, jogging, sitting, standing, upstairs and downstairs. The three-dimensional (3D) raw accelerometer data is directly used as the input for training the CNN without any complex pretreatment. The performance of our CNN-based method for multi human activity recognition showed 91.97% accuracy, which outperformed the Support Vector Machine (SVM) approach of 82.27% trained and tested with six kinds of features extracted from the 3D raw accelerometer data. Therefore, our proposed approach achieved high recognition accuracy with low computational cost.

Keywords—convolutional neural network; human activity recognition; 3D accelerometer data; SVM

I. INTRODUCTION

With the rapid development of information technology and the popularity of smart devices, it is easier to gather the data that can describe human daily activities collected from various sensors, which are integrated in smart devices. Apparently, instead of attaching a variety of cumbersome sensors to the user's body [1], people are more willing to accept portable, wearable and multi-functional devices such as smartphones and smartwatches, which also embed various sensors, for instance, accelerometer and gyroscope. Therefore, more and more methods and software applications based on smartphone sensors are proposed and developed for human activity detection [2, 3].

One of the smartphone-based methods for human activity recognition is through the pre-installed software, such as Health app for iphone, which is implemented by the use of IOS Activity Recognition API [4]. IOS Activity API can acquire these motion types of users such as stationary, walking, running, automotive and cycling by collecing the sensor data of smartphone sensors. We used the Health app to detect three activities, walking, running and cycling, and compared the recognition results with the ground truth labels. The experiment result showed that many of the activities recognized by the API were inaccurate or identified as 'unknown', resulted in a low recognition performance. We presume that such low performance is caused by the user's individual differences in movement patterns, such as travel speed, gait and so on. That is, the recognition accuracy will be greatly improved if the activity recognition model is trained by user's personal activity data and ground truth labels before it can be used.

In order to reduce the impact of the diversity of human activity patterns in human activity recognition, we present a Convolutional Neural Network (CNN)-based method that uses the three-dimensional (3D) raw accelerometer data collected from the smartphone accelerometer sensor, which has less computation, higher recognition accuracy, and higher flexibility and robustness. We compare our method to a Support Vector Machine (SVM) approach trained and tested with six kinds of features extracted from the 3D raw accelerometer data to exhibit the effectiveness of our method. The six kinds of human activities we chose to be recognized were walking, jogging, sitting, standing, upstairs and downstairs.

The rest of the paper is organized as follows: Section II presents the related works on CNN-based human activity recognition; Section III explains our CNN-based method for human activity recognition; Section IV presents our experiment and its result. Finally, we make a conclusion of this study in Section V.

II. RELATED WORK

Recently, more and more CNN-based human activity recognition methods have been proposed, as well as the machine learning-based mobile sensing applications in the life-logging, fitness tracking and health monitoring domains with the rapid progression of artificial intelligence.

978-1-5386-3788-3/18/$31.00 ©2018 IEEE

Ha and Choi presented CNNs (CNN-pf, CNN-pff) for human activity recognition to handle multivariate time series data measured at multiple heterogeneous [5]. Partial and full weight sharing, and full weight sharing were employed in the lower layers, and the upper layers of the CNN model, respectively. They demonstrated that the CNNs achieved the high performance of 91.94% with much smaller number of parameters compared to CNNs with 1D kernels, and handled multi-modal data more efficiently compared to existing CNNs with 2D kernels on the benchmark Mhealth dataset which is about twelve daily activities recorded from four different types of sensors.

Yang et al. aimed to build a deep convolutional neural network (DCNN) to automate feature learning from the multichannel time series data for the human activity recognition (HAR) task [6], which mainly employed the convolution and pooling operations to capture the salient patterns of the sensor signals at different time scales. The accuracy of the proposed CNN method was 86.7% on the Opportunity Activity Recognition dataset, and 96.0% on the Hand Gesture dataset, outperformed other state-of-the-art methods. The proposed method was testified to be as a competitive tool of feature learning and classification for the HAR problems.

Lee, Sang and Cho utilized triaxial accelerometer data collected from users' smartphones to recognize three human activities through a 1D CNN-based method [7]. The 3-axes acceleration data were transformed into a vector magnitude data to reduce the possible rotational interference, their network achived 92.71% activity recognition accuracy, which was higher than the baseline random forest approach. In addition, they found that the activity recognition performance was improved if the input vector dimention was increased.

Sheng et al. explored a novel short-time activity recognition method based on convolutional neural network [8]. In their proposed method, an over-complete pattern library which consisted of short-time activity patterns collected by wearable sensors using sliding window method have been created, then they used the library to train a well-designed CNN in order to extract robust features and classify several kinds of short-time human activities. This method showed the recognition performance of 95.92% on WARD1.0 dataset.

Different to the approaches mentioned above which adopted a variety of sensor data, such as accelerometer data, gyroscope data, etc. [5, 6, 8], our method uses the three-dimensional (3D) raw accelerometer data directly collected from a single accelerometer sensor, thus can greatly reduce the workload of data pre-processing. Compared with [7], we add a convolutional layer in the end of our CNN structure to improve the recognition accuracy while using the raw accelerometer data to reduce computational complexity.

III. Proposed Method

In this section, we explain the architecture of the proposed CNN model for human activity classification.

As shown in Fig. 1, the network architecture of our CNN consists of one convolution layer followed by max pooling and another convolution layer, after that, the model has fully connected layer connected to softmax layer that outputs the probability of each of the six activities. Before entering the first convolution layer, the normalization processing is applied to the 3D accelerometer data.

A. Input

A fixed time-length raw accelerometer data was used as input as Fig. 1 *a. input* shows. Specifically, we segmented each of the acceleration values generated from the acceleration sensor's x, y, and z-axis using a sliding window with a size of 90 samples with 50% overlap over time domain. Then the generated segments were reshaped to have a height of 1 to implement one-dimensional (1D) convolution to improve computational efficiency. Besides, in order to eliminate the dimensional impact of the indicators, data normalization was applied first (as shown in Fig. 1 *b. normalization*), and the data normalization method we utilized was the Z-score standardization, its formular is:

$$Z' = \frac{Z - \mu}{\sigma} \tag{1}$$

The standard values of the raw x, y, and z acceleration data can be calculated by equation (1), μ, and σ are the mean value, and the standard deviation of Z, respectively. After such simple data preprocessing, the dataset that subjected to normal distribution was obtained.

B. Convolution

There were two convolution layers in our CNN model. In the first convolution layer (as shown in Fig. 1 *c. convolution*), the convolution operations were performed using a filter with size of 60 and stride size of 1, yielding a feature map size of 31, and a total of 60 filters were used on each input x, y, and z acceleration data which were normalized. In the second convolution layer (as Fig. 1 *e. convolution* shows), it took an input of max-pooling layer and applied the filter with size and depth of 6, and stride size of 1 outputting a unit feature map as the input to the next fully connected layer. Note that the convolution and max-pool layers were one-dimensional (1D).

C. Max-pooling

As shown in Fig. 1 *d. max-pooling*, after the first convolution layer, a max-pooling was performed. The purpose of this layer is to select the largest feature value mainly to achieve two goals, one is to introduce invariance for the following processing, including translation invariance, rotation invariance and scale invariance; the other is to reduce parameters and calculations while retaining the key features of the feature map. The pooling layer's filter size was set to 21 and with a stride of 2, thus can generate the max-pooled feature vectors with size of 6, which was then going to be the input to the second convolution layer.

D. Dropout

After the processing of second convolution layer, the convolved and max-pooled result was then flattened out as shown in Fig. 1 *reshape* to create a long 1080-length feature

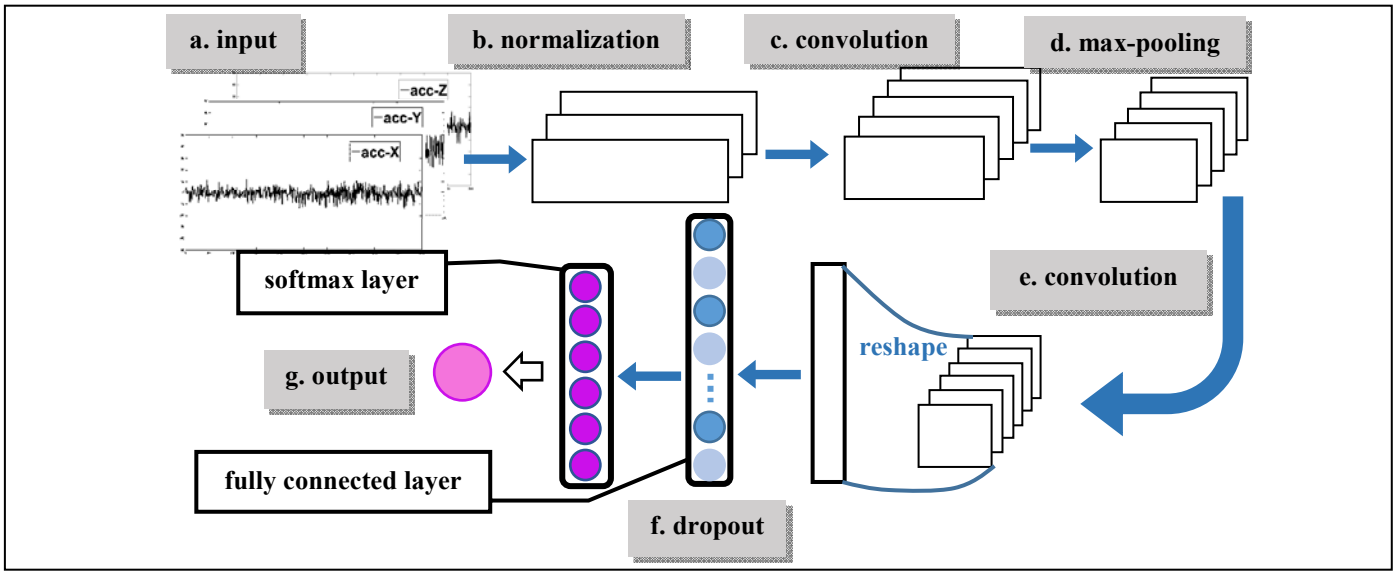

Fig. 1. The network architecture of proposed CNN-based activity recognition method

vector for every input data, which was used as the input to the fully connected layer with dropout applied.

As a common method of regularization, the dropout was applied to prevent the neural network from overfitting and make the model have effective generalization performance [9], the default value of the percentage of dropout was 0.5. In the fully connected layer, there were 1000 neurones and the activation function of the network chosen here was the *tanh* function. The *tanh* function typically performs better than the logistic sigmoid as a neuron activation function because it resembles the identity function more closely.

E. Output

At the end of this network structure, the softmax layer was defined as an output layer of the fully connected layer as shown in Fig. 1 *softmax layer*. Each node in the softmax layer calculated the probability of each activity (i.e., walking, jogging, sitting, standing, upstairs, and downstairs) given by the long feature vector. Then the activity with the highest probability was chosen to be the predicted (or recognized) activity and its label was outputted as the final identification result of the CNN model as shown in Fig. 1 *g. output*.

IV. Experiments

In this part, we introduce the dataset used in our experiment; describe experimental settings including hyper-parameter settings in CNN model and feature extraction method in Support Vector Machine method, finally the comparision of the results between our method and the SVM activity recognition approach is given.

A. Dataset

The dataset we used was Actitracker dataset released by Wireless Sensor Data Mining (WISDM) lab (Kwapisz, Weiss, and Moore 2012) [10]. This dataset contains six human daily activities including walking, jogging, sitting, standing, upstairs,

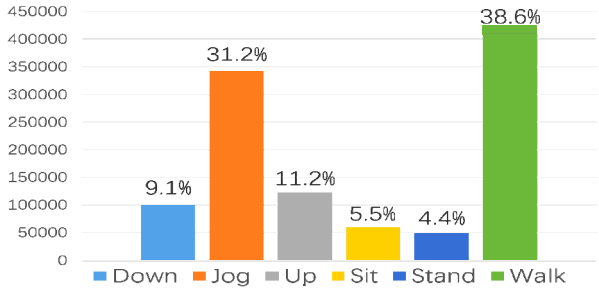

Fig. 2. Data distribution of WISDM dataset

and downstairs, which were collected through a single accelerometer sensor in a controlled laboratory environment.

Specifically, the data was collected from thirty-six users placing a smartphone in their pocket with the 20Hz sampling rate (20 values per second). The dataset contains 1,098,207 samples in total and its distribution with respect to activities (class labels) is shown in Fig. 2.

In our experiment, first, the dataset was segmented into 2,440,4 segments with 50% overlap using a sliding window as described in Section III *A. input*. Then we divided the segments into training set and testing set randomly for the network training and testing experiments according to the principle of 70/30. Table I summarizes the size of train and test data.

B. Experimental Settings

The network architecture of our CNN-based human activity recognition method is shown in Fig. 1, which was implemented by using the open source software TensorFlow [11], and the neural network was trained and tested in the Python environment. The following hyper-parameter values were selected in the experiment: *convolution window size* = 60,

TABLE I. TRAIN & TEST DATA

Train	Test	Total
1,707,9	7,325	2,440,4

6, *number of convolution filters* = 60, 6 (in the first, and the second convolution layer, respectively), *window stride size* = 1; *pooling window size* = 21, *window stride size* = 2; *training batch size* = 16; *training epoch size* = 20; *number of hidden neurons* = 1000. As for the calculation and optimization of the loss function in our CNN model, we used the Adaptive Moment Estimation (ADAM) optimization method [12] to minimise the negative log-likelihood cost function.

The explanations of some important hyper-parameter settings in our CNN model are as follows:

- **Training epoch size:** When we set the training epoch size to 20, the difference between training loss (6.71%) and testing loss (8.03%) was very little, so it could be considered that the current training epoch size was appropriate.
- **Training batch size:** Generally, in the training process of the convolution neural network, small batch of training data can avoid falling into local minimun value and improve the generalization ability of the network, so we chose a relatively small value of 16 as the training batch size.
- **Weight initialization:** In the convolution layer, small random numbers are usually used to initialize the weight of the convolution kernels, such as the data set subjected to Gaussian distribution with standard deviation of 0.1 ($\sigma = 0.1$) and mean value of 0 ($\mu = 0$). However, not all the data obeying the Gaussian distrubution is useful for weight initialization because too small numbers of the data may generate zero gradient networks. To get around this, we used an optimized Gaussian function, the truncated Gaussian distribution function, with this function, the too small data outside the interval ($\mu - 2\sigma$, $\mu + 2\sigma$) were abandoned.

For evaluation, we compared the proposed method with the SVM-based activity recognition approach. Support vector machine is a classification method mainly based on feature extraction [13], in our evaluation experiment, six kinds of features extracted from the WISDM dataset (introduced later) were used as input of the SVM-based activity recognition model, the model was implemented by employing the open source LIBSVM [14, 15], which ran under the Disk Operating System (DOS). In order to extract the eigenvalues, first, the *x*, *y*, and *z* acceleration data need to be transformed into a vector magnitude data by calculating the Euclideam norm of them as equation (2) shows:

$$\|a\| = \sqrt{x^2 + y^2 + z^2} \tag{2}$$

TABLE II. FEATURES EXTRACTED FROM WISDM DATASET

Feature	Description
Mean	Average value of samples in window
Max	Maximum
Min	Minimum
STD	Standard deviation
Range	Maximum minus minimum
Mean crossing rate	Rate of signal crossing mean value

Then, the six features extracted from the vector magnitude accelerometer data calculated above are simply introduced in Table II.

All the features in Table II were obtained from the vector magnitude accelerometer data using the sliding window described in Section III *A. Input*.

Eventually, after the above processes, the amount of data that would be used for training and testing the SVM model was much larger than the amount of data used in the CNN method (in our case, it was twice as much), in addition, the computational complexity of data preprocessing in SVM activity recognition method was also higher.

C. Experimental Results

The results of the proposed CNN-based activity recognition method and the SVM-based method on WISDM dataset are presented in the form of confusion matrix as shown in Table III. In the field of machine learning, specifically the problem of statistical classification, a confusion matrix is known to be a specific table layout that allows visualization of the performance of an algorithm. In the confusion matrix, each row represents the instances in an actual class while each column represents the instances in an predicted class, so the diagonal entries show the number of correctly classified test data. Benefit from the confusion matrix, the overall accuracies of human activity recognition for our CNN-based method and SVM-based method can be calculated easily as well as the precision and recall for each activity.

The bottom right corner of each confusion matrix displays the overall accuracy of each method. We see that our CNN method (91.97%) outperforms the SVM approach (82.27%). Note that the recognition accuracy of SVM method is not very high, we speculate the reason for this is that the features extracted for SVM model learning are not enough as well as the six data processing methods described in Table II are relatively simple.

Indeed, as Tran and Phan have found [16], the accuracy of SVM-based activity recognition system depends on the selected features. In their system, data was collected from 3 types of sensor and after the feature extraction processing, a total of 248 features (much more than we have extracted) in the time domain and the frequency domain were obtained, and finally the SVM-based recognition system achieved 89.59% accurate rate. This shows that the SVM-based human acitivity recognition system can reach a high accuracy only at the cost of very high computational complexity of feature extraction.

TABLE III. CONFUSION MATRIX FOR PROPOSED CNN & SVM METHOD

(ACTIVITY NAMES ON TOP ARE ABBREVIATED)

CNN	DS	JG	ST	SD	US	WK	Recall (%)
Down	**543**	26	0	0	86	83	73.58
Jog	2	**2195**	0	0	0	22	98.92
Sit	2	0	**390**	47	0	0	88.84
Stand	0	0	16	**272**	3	0	93.47
Up	81	57	0	3	**702**	89	75.32
Walk	39	6	0	0	29	**2632**	97.27
Accur. (%)	81.4	96.1	96.1	84.5	85.6	93.1	**91.97**

SVM	DS	JG	ST	SD	US	WK	Recall (%)
Down	**486**	28	0	0	111	113	65.85
Jog	73	**1964**	0	0	15	167	88.51
Sit	22	0	**349**	68	0	0	79.50
Stand	6	0	25	**243**	17	0	83.51
Up	95	82	0	0	**628**	127	67.38
Walk	145	96	0	0	109	**2356**	87.07
Accur. (%)	58.8	90.5	93.3	78.1	71.4	85.3	**82.27**

Unlike the SVM-based method, our CNN-based human activity recognition system can achieve high accuracy with low computational cost as is expected.

Table III demonstrates that we can achieve high levels of accuracy in most cases, especially for the two most common activities, walking and jogging, we achieve accuracies nearly 90% in SVM method and above 90% in our proposed CNN method.

In these two activities, jogging appears easier to identify than walking, which seems to make sense, since jogging involves more extreme changes in acceleration. For both proposed CNN method and SVM method, it appears that it is much more difficult to identify the two stair climbing activities, which is because those two similar activities are easily confused with one another.

Note that although there are very few samples of sitting (5.5%) and standing (4.4%) in WISDM dataset as shown in Fig. 2, we can still identify these activities quite well in both methods, because the two activities cause the smartphone to change orientation and this is easily detected from the accelerometer data.

On the whole, the human activity recognition accuracy of our prooposed CNN method outperformes the Support Vector Machine approach, because, as noted earlier, the feature extraction method we adopted cannot fully describe the characteristics of activity data. On the other side, as the types of daily activities to be identified increases, more complex feature extraction methods may be used for SVM-based model learning, and the complexity of data preprocessing is also increasing, but on the contrary, the convolutional neural network can solve this problem easily.

V. CONCLUSION

In this paper, we designed and constructed a CNN-based human activity recognition method uses raw triaxial accelerometer data gathered from the user's smartphone. We found that our proposed method achieved high recognition accuracy with low computational cost of data preprocessing, which outperformed the SVM-based method in the multi human activity classification. In addition, this characteristic makes it possible to apply the activity recognition method to portable smart devices with limited computing power such as smartphones.

However, it is undeniable that the proposed method still has some certain restrictions; percentage of recognition is low in some action such as downstairs and upstairs. So in the future, more research should be done to improve the performance and increase the detection capabilities of the model.

ACKNOWLEDGMENT

The research work was supported by the National Nature Science Foundation of China under grant nos. 61300043, 61373156 and 91438121, and the Science and Tecnology Commission of Shanghai Municipality under grant no. 14DZ2260800.

REFERENCES

[1] Zubair, Muhammad, K. Song, and C. Yoon. "Human activity recognition using wearable accelerometer sensors." *IEEE International Conference on Consumer Electronics-Asia* IEEE, 2016:1-5.

[2] Yang, Hua Cong, et al. "HARLib: A human activity recognition library on Android." *International Computer Conference on Wavelet Active Media Technology and Information Processing*IEEE, 2015:313-315.

[3] Siirtola, Pekka, and Juha Röning. "Recognizing Human Activities User-independently on Smartphones Based on Accelerometer Data." *International Journal of Interactive Multimedia & Artificial Intelligence* 1.5(2012):38-45.

[4] IOS API for Activity Recognition: https://developer.apple.com/documentation/coremotion/cmmotionactivitymanager (retrieved: May. 3, 2017)

[5] Ha, Sojeong, and S. Choi. "Convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors." *International Joint Conference on Neural Networks* IEEE, 2016:381-388.

[6] Yang, Jian Bo, et al. "Deep convolutional neural networks on multichannel time series for human activity recognition."*International Conference on Artificial Intelligence* AAAI Press, 2015:3995-4001.

[7] Lee, Song Mi, M. Y. Sang, and H. Cho. "Human activity recognition from accelerometer data using Convolutional Neural Network." *IEEE International Conference on Big Data and Smart Computing* IEEE, 2017:131-134.

[8] Sheng, Min, et al. "Short-time activity recognition with wearable sensors using convolutional neural network." *ACM SIGGRAPH Conference on Virtual-Reality Continuum and ITS Applications in Industry* ACM, 2016:413-416.

[9] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research* 15.1(2014):1929-1958.

[10] Kwapisz, Jennifer R., G. M. Weiss, and S. A. Moore. "Activity recognition using cell phone accelerometers." *Acm Sigkdd Explorations Newsletter* 12.2(2011):74-82.

[11] Google TensorFlow, https://www.tensorflow.org/ (retrieved: Mar. 27, 2017)

[12] Kingma, Diederik P, and J. Ba. "Adam: A Method for Stochastic Optimization." *Computer Science* (2014).

[13] Hsu, C. W., C. C. Chang, and C. J. Lin. "A Practical Guide to Support Vector Classication." *Taipei: National Taiwan University, Department of Information Engineering* 67.5(2016).

[14] LIBSVM, http://www.csie.ntu.edu.tw/~cjlin/libsvm/ (retrieved: Dec. 22,2016)

[15] Chang, Chih Chung, and C. J. Lin. *LIBSVM: A library for support vector machines*. ACM, 2011.

[16] Tran, Duc Ngoc, and D. D. Phan. "Human Activities Recognition in Android Smartphone Using Support Vector Machine." *International Conference on Intelligent Systems, Modelling and Simulation* IEEE, 2017:64-68.