

Research Article

Towards Activity Recognition through Multidimensional Mobile Data Fusion with a Smartphone and Deep Learning

Junkuo Cao,¹ Mingcai Lin,² Han Wang,³ Jiacheng Fang,³ and Yueshen Xu ³

¹Network and Data Center, Hainan Normal University, Haikou, China

²Haikou National Science Park, Hainan Normal University, Haikou, China

³School of Computer Science and Technology, Xidian University, Xi'an, China

Correspondence should be addressed to Yueshen Xu; ysxu@xidian.edu.cn

Received 29 October 2020; Revised 4 March 2021; Accepted 13 April 2021; Published 20 April 2021

Academic Editor: Xiaoxian Yang

Copyright © 2021 Junkuo Cao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The field of activity recognition has evolved relatively early and has attracted countless researchers. With the continuous development of science and technology, people's research on human activity recognition is also deepening and becoming richer. Nowadays, whether it is medicine, education, sports, or smart home, various fields have developed a strong interest in activity recognition, and a series of research results have also been put into people's real production and life. Nowadays, smart phones have become quite popular, and the technology is becoming more and more mature, and various sensors have emerged at the historic moment, so the related research on activity recognition based on mobile phone sensors has its necessity and possibility. This article will use an Android smartphone to collect the data of six basic behaviors of human, which are walking, running, standing, sitting, going upstairs, and going downstairs, through its acceleration sensor, and use the classic model of deep learning CNN (convolutional neural network) to fuse those multidimensional mobile data, using TensorFlow for model training and test evaluation. The generated model is finally transplanted to an Android phone to complete the mobile-end activity recognition system.

1. Introduction

Human activity recognition belongs to a branch of pattern recognition, and its related research can be traced back to the 1980s. Because it can provide personalized support for many different applications and has connections with many different subject areas, such as medicine, human-computer interaction, and sociology [1, 2]. The research on human activity recognition has never stopped and has always been a research hot topic for researchers. Numerous researchers have tried to find a method that can efficiently and accurately identify human activities [3, 4]. Based on the support of various software and hardware, there have been many good research results in the field of human activity recognition, but the recognition effect still needs to be improved, and with the continuous development of technology and continuous research of various theories, it is necessary to continuously carry out new exploration and research in the field of human activity recognition in order to propose an

efficient and accurate method of human activity recognition in the future [5, 6].

At present, researches in the field of human activity recognition can be divided into two categories, one of which is based on the analysis of video images, and the other is based on various motion sensors, such as inertial navigation modules and acceleration sensors [7, 8]. The image-based research method can be more intuitive and accurate and can better identify the complex motion state. At present, the research on human activity recognition in China is mainly based on image analysis [9, 10]. This method has its advantages, but its disadvantages are also obvious. It requires higher data acquisition equipment, and its costs are higher; it can only be used in characteristic venues [11, 12]. Therefore, this method is not very popular and is only applied in some specific fields. This research method does not belong to the research content of this article. It will not be described in detail. Interested readers can inquire related information by themselves.

Another type of activity recognition is based on motion sensors. In recent years, with the rapid development of technology, the accuracy of sensors has increased, and the cost of production has been reduced [13, 14]. Particularly in recent years, smartphone users have continued to increase at an almost explosive speed. The growth of sensor devices in smart phones is becoming more sophisticated, and wearable devices are spreading at an alarming rate [15, 16]. These have led many researchers to see the research prospects of the sensor-based activity recognition. Therefore, in recent years, research in this field has attracted more and more researchers to participate in it and has created many gratifying researches results [17, 18]. Many related researches have been applied to the daily lives of people such as medicine and sports. In China, many researchers have also proposed many cutting-edge research results. However, as the research continues and the accuracy increases, the increase in the types of identification activities has become a major problem. There is no particularly good solution, so research in this direction still requires the continuous efforts of many researchers.

The research on human activity recognition can be divided into two general directions [19]. It is divided into research based on video images, or research based on motion sensors. I will not go into details here, but the general research methods and steps are similar and are generally divided into the following steps: data collection, feature extraction, model building, and model evaluation. The research in this paper starts with data acquisition, that is, acceleration sensor data and uses TensorFlow¹ (Google open-source system for artificial intelligence) to build a CNN (convolutional neural network). We further use the collected data for model training and test and evaluate and develop the model. In the end, we transplant the developed model to the mobile computing platform to implement the mobile-end activity recognition system.

The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 elaborates the data collection work in real-world environment. Section 4 explains the proposed convolutional neural network-based model. Section 5 depicts the implementation and running of our platform. Section 6 presents the experiments and analysis. Section 7 concludes the whole paper.

2. Related Work

As early as 2010, the method of using signal strength descriptor to detect indoor movement was proposed. This method can also be applied to the detection of virtual objects near the transmitter and the movement of people moving in the room [20, 21].

Later, the method of using WiFi signal to track human body and recognize simple gestures has been developed one after another [22, 23], and the corresponding tools have also been released, such as tools that can record the detailed measurement of wireless channel and the tracking of received 802.11 packets [24]. Wireless network can realize device-free fall detection [25], which breaks through the limitation of conventional fall detection system without

external modification or additional environment settings. WiFi signal can also be used to detect smoking behavior, and a passive smoking detection system based on foreground detection is realized [26]. The latest related papers also focus on multitarget tracking in mobile environment without equipment. This paper proposes an antinoise, unobtrusive, and no-equipment tracking framework [27].

In addition, the fine-grained activity recognition can be realized by RGB-D camera, which improves the intelligence of pervasive and interactive system to a new level [28, 29]. All kinds of new types of small sensors shine brilliantly in the application of human activity recognition. Through wearable sensors or even sensors based on smart phones, we can track human activities and provide health care support [30, 31]. For example, through wearable acoustic sensors, we can analyze the voice generated in the throat area of the user and accurately identify the user's activities [32]. In Implantable Medical Devices (IMD), wireless communication is used to deal with the interference attack of others and improve the security of IMD [33]. Through WiFi signal, we can not only recognize human activity, but also hear our speech by detecting the mouth and analyzing the fine-grained radio reflection from the mouth action [34]. When analyzing and calculating the collected data, edge computing can be used to improve resource utilization and execution efficiency [35].

The application of neural network in the field of activity recognition is more and more extensive. The image classification method based on deep convolution neural network promotes the development of neural information processing system [36]. On the basis of CNN, the ability of activity recognition is improved by using data enhancement and transfer learning [37]. The framework of activity recognition based on CNN is developed [38]. When training human activity data, the activity recognition model trained on one person may not work well when it is applied to predict another person's activity [39, 40]. In order to meet this challenge, data enhancement for human activity recognition is also a research hotspot.

3. Data Collection

3.1. Sensors of Smartphone. The data collected by the sensors of the smartphone has its own set of coordinate systems (the natural coordinate system of the mobile phone). As shown in Figure 1, the phone is positive in the x -axis direction, the y -axis is in the upward direction, and the z -axis is in the positive direction perpendicular to the mobile phone screen. The built system can monitor the change of acceleration value of the smart phone device in the three axes of the corresponding x -axis, y -axis, and z -axis.

Looking at the official Google Android developer website², we can find that the Android platform provides thirteen sensors for use. Some of these sensors are hardware-based and some are software-based. However, not all Android sensors have all of these sensors. Different Android devices integrate different sensors and already support different devices. The acceleration sensor used in this article is based on hardware. It monitors the mobile phone's x -, y -,

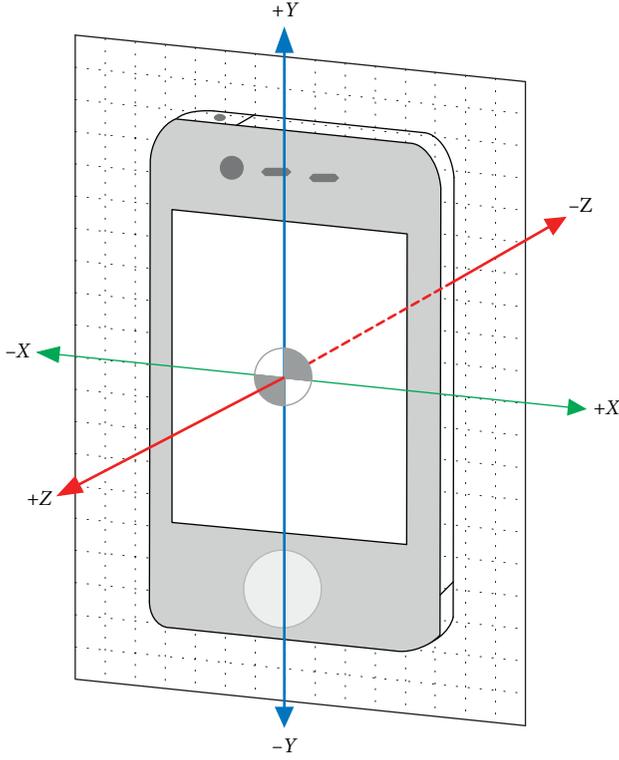


FIGURE 1: Mobile phone natural coordinate system.

and z -axis acceleration in m/s^2 (including gravity, and the x -, y -, and z -axes are shown in Figure 1). This sensor is integrated in most mobile phones and tablets, and the Android platform has supported it since version 1.5 (Table 1), so using this sensor to design the activity recognition system discussed in this article can be applied to almost all the current market Android phone devices. The acceleration sensor measures the force exerted on the sensor and detects the acceleration of the equipment according to the following formula:

$$A_d = -\frac{\sum F_s}{\text{mass}} \quad (1)$$

Gravity always affects the accuracy of measurement, which is computed according to the following relationship:

$$A_d = -g - \frac{\sum F}{\text{mass}} \quad (2)$$

The Android platform provides a complete sensor framework, including a series of sensor classes and interfaces. Using the corresponding API allows us to easily use the functions of the corresponding sensor. The main classes used are `SensorManager`, `Sensor`, `SensorEvent`, and `SensorListener`. Each category is specifically explained with the acceleration sensor studied in this paper.

- (1) First, we obtain an instance of the `SensorManager` class and undertake the management of each sensor, such as creating a sensor instance, setting the sensor sampling frequency, and registering/deregistering sensor event monitoring.

- (2) Second, we create an instance of the `Sensor` class (acceleration sensor) through the `SensorManager` instance.
- (3) We use `SensorManager` to set the sampling frequency of the acceleration sensor and event monitoring.
- (4) We rewrite the listener method, `onSensorChanged` (`SensorEvent` event), where the accelerations on the corresponding coordinate axes x , y , and z can be obtained through `event.values [0]`, `event.values [1]`, and `event.values [2]`, respectively. We can obtain the timestamp through `event.timestamp`.

3.2. Design and Implementation of Data Collector. Due to the timeliness of experimental purposes, the data is stored locally in txt text, one sample of data occupies one line, and each sample of data uses the following format:

Data format: user ID, time stamp, x -axis acceleration, y -axis acceleration, z -axis acceleration. Example: 1, sitting, 288956018483233, -6.3601966 , -1.3551182 , and 7.7943234 . Register and log in. After entering the main page of the collector, select the corresponding state switch button on the interface to start collecting sensor data. Every 200 pieces of data are collected and written to the log file. Discard the data collected for the first time and start recording from the second set of data. The design of the interface of the collector is shown in Figure 2.

Due to the limitation of experiment time, environment, personnel, and other factors, the personnel involved in the sampling were only themselves. Such a sampling way is for improving the ability to train a better model with less data. In this paper, the mobile phone is sampled with the screen facing forward and the head down in the right front pocket of the pants. And the total number of samples finally obtained is 153,000. The amount of samples per activity is shown in Table 2.

The specific pie chart distribution is shown in Figure 3.

The sensor sampling frequency is 50 HZ; that is, one piece of data is collected every 0.02 s. The data are collected in 200 pieces. Here, waveform analysis of the first 200 pieces of data collected for each activity is performed, where the data are collected by the sensor in 4 seconds. The horizontal axis of the waveform graph is a time stamp, and the vertical axis is the x , y , and z accelerations and the true acceleration after normalization. The computation is as follows:

$$\text{Acc} = \sqrt{x^2 + y^2 + z^2}, \quad (3)$$

where `Acc` denotes the total acceleration.

Each activity corresponds to 4 waveforms. By observing the waveform diagram of the sensor data corresponding to human activities, we can intuitively feel that the sensor data generated by different activities are different, and they have a certain regularity. Therefore, we can use deep learning to let machines learn this law and generate corresponding models to achieve the purpose of identifying related human activities. The 4 s sampling waveform of *downstairs* is given in Figure 4, the 4 s sampling waveform diagram of *running* is

TABLE 1: Support for acceleration sensors in each version of the Android platform.

Sensor	Android 4.0 (APL level 14)	Android 2.3 (APL level 9)	Android 2.2 (APL Level8)	Android 1.5 (APL level 3)
TYPE_ACCELEROMETER	Yes	Yes	Yes	Yes

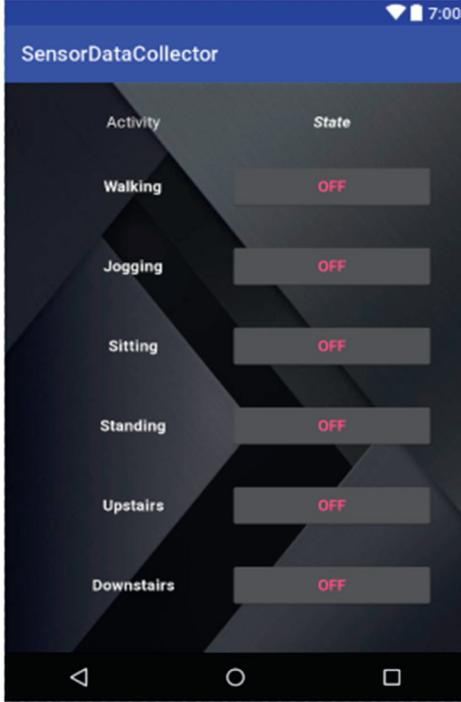


FIGURE 2: Design of the main interface of the collector.

TABLE 2: The amount of sampling data per activity.

Activity	Number of samples	Proportion (%)
Walking	21,600	14.1
Running	20,200	13.2
Standing	33,600	22.0
Sitting	27,000	17.6
Upstairs	26,600	17.4
Downstairs	24,000	15.7

given in Figure 5, and the 4 s sampling waveform of *sitting* is given in Figure 6. The 4s sampling waveform of *standing* is shown in Figure 7, the 4 s sampling waveform of *upstairs* is shown in Figure 8, and the 4 s sampling waveform of *walking* is shown in Figure 9.

4. Convolutional Neural Network-Based Model

Convolutional neural network (CNN) belongs to a type of feedforward neural network. Convolutional neural network has been widely used in the field of image and speech recognition because of its better test results. The most commonly applied field of CNN is in the field of pattern recognition, especially for large-scale image processing. It has its extraordinary performance. Because it can make images directly input to the network, it can avoid

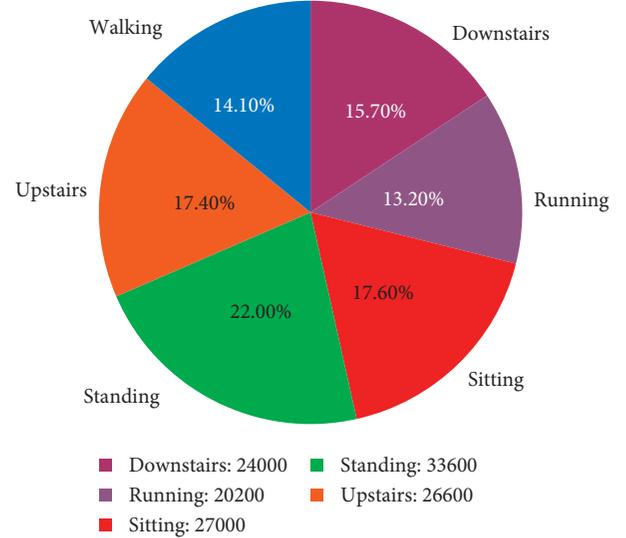


FIGURE 3: Sample data distribution.

complicated feature extraction processes and data reconstruction process. However, due to its continuous innovation, convolutional neural networks (CNN) are now also used in the fields of video analysis, intelligent language processing, and drug discovery. And at present, it has become one of the hotspots in many scientific fields. Various fields are trying to use convolutional neural network technology to add new vitality to their fields.

Since the successful release of the AlexNet architecture in 2012, a series of classic architectures such as VGG, GoogLeNet, and ResNet have appeared successively. In recent years, researchers have continued to design many new methods to improve CNNs. Therefore, many variants of CNN architectures have been proposed. Therefore, in different literatures, some detailed description about convolutional neural networks (CNN) may be biased in some places. However, no matter how the variants are, the basic concepts and principles of the CNN architecture will not change, and their various components are also very close. We adopt LeNet-5 in this paper and adjust the parameters in order to meet our demand. LeNet-5 can be divided into 6 layers in addition to the input layer and output layer. Each layer contains a different number of training parameters (connection weights), as shown in Figure 10. The specific structure is convolution layer, pooling layer, volume layer, pooling layer, fully connected layer, and fully connected layer.

- (1) Convolutional layer: Convolutional layer is used for feature extraction. In convolutional neural networks, we often use multiple layers of convolutional layers to get deeper feature maps.
- (2) Pooling layer (lower sampling layer): The main work of the pooling layer is to compress the input feature

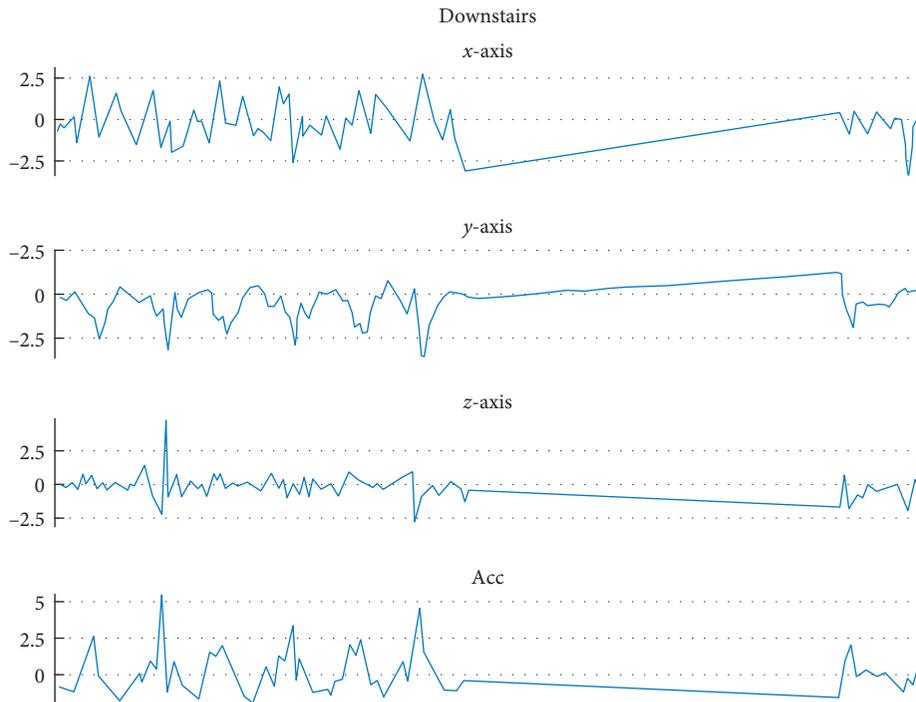


FIGURE 4: 4 s sampling waveform of *downstairs*.

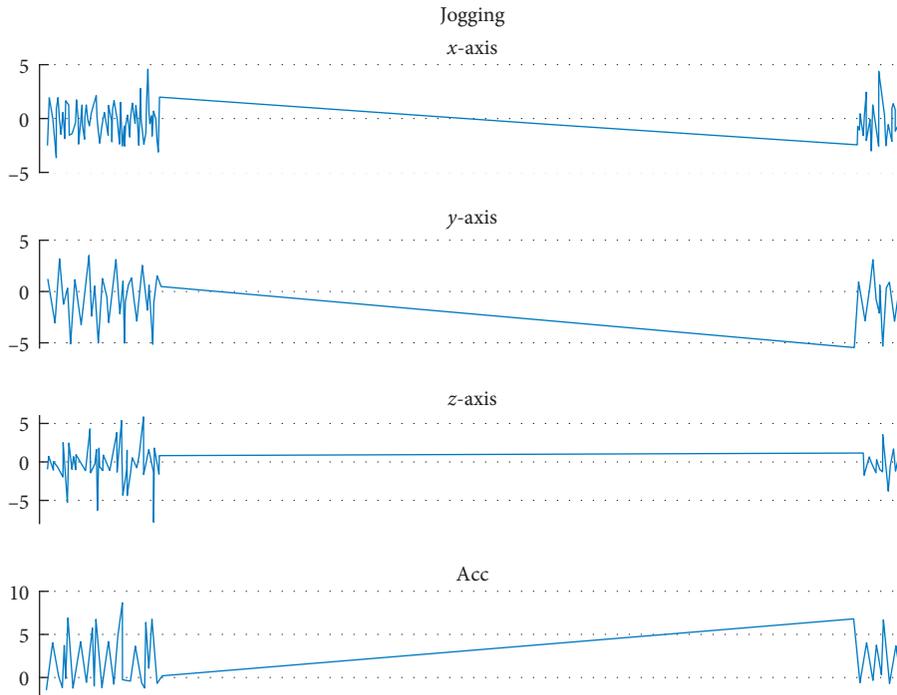
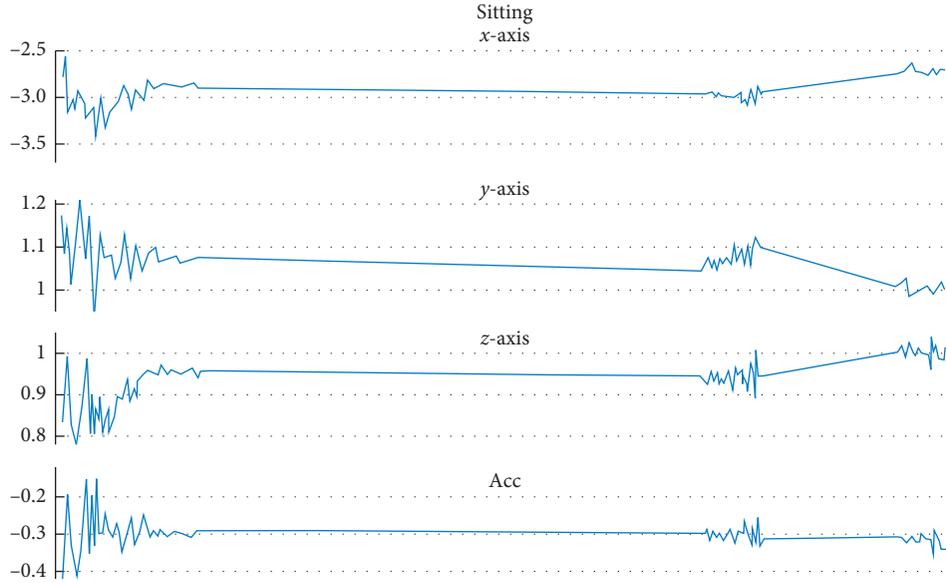
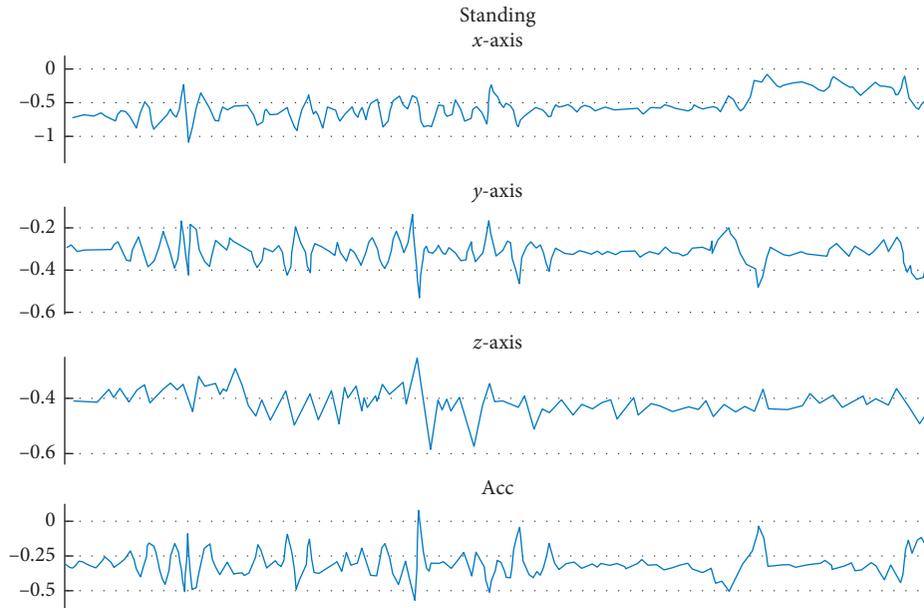


FIGURE 5: *Running* sampling waveform diagram 4 s.

map along the spatial dimension (height and width). The pooling layer can compress the feature map output by the convolutional layer to extract the main features, thereby reducing the number of parameters and accelerating the neural network. And pooling has translation invariance, which enables us to

extract the feature maps unchanged after the image is panned and scaled, helping us to make correct and identical recognition results for images that have been panned and scaled.

(3) Fully connected layer: The work of the fully connected layer is relatively simple. It connects all the

FIGURE 6: *Sitting* 4 s sampling waveform.FIGURE 7: *Standing* 4 s sampling waveform.

features and transmits these output values to the (i.e., SVM and Softmax) classifier for final classification and judgment.

The computation procedure of LeNet-5 employed in this paper is as follows. The weights that are generated from the previous layer are represented as w_i for the neural unit i . We use sigmoid function to generate the state z_i , which is computed as follows:

$$z_i = \text{sigmoid}(w_i), \quad (4)$$

where $\text{sigmoid}(\cdot)$ refers to the sigmoid function. The output layer is computed using RBF function (radial basic function)

to compute the result over each class. RBF computes the final result y_i over each class as

$$y_i = \sum_k (z_k - q_{ij})^2, \quad (5)$$

where q_{ij} represents the ground truth of states.

The overall structure is similar to LeNet-5 in our paper, except for input and output, and it can be divided into six layers. We use the visualization module that comes with TensorFlow to visualize the specific neural structure in front of us. The overall neural network structure of this paper is shown in Figure 10. Except for the input and output, the specific neural network structure is convolution layer-

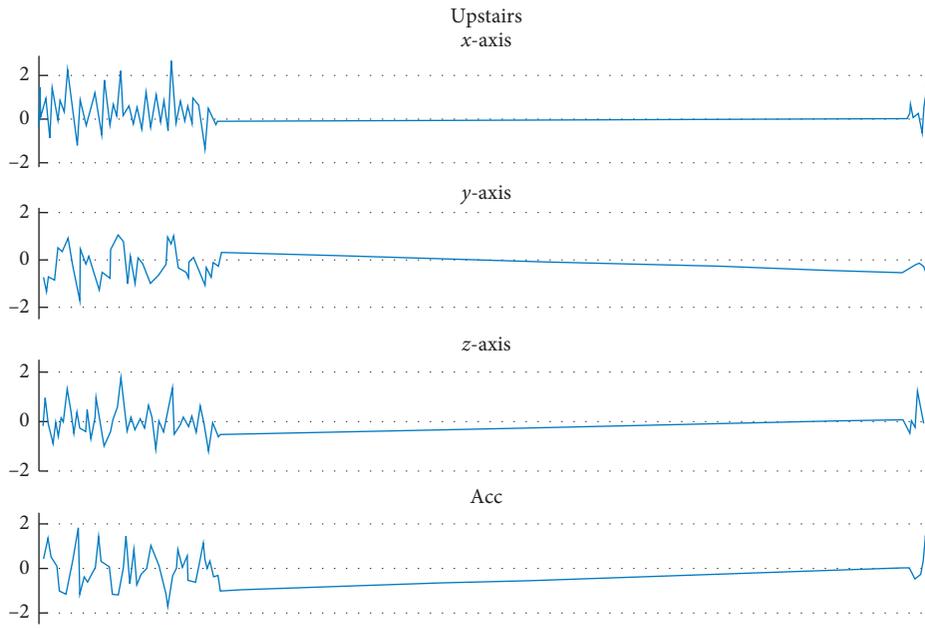


FIGURE 8: Upstairs 4 s sampling waveform.

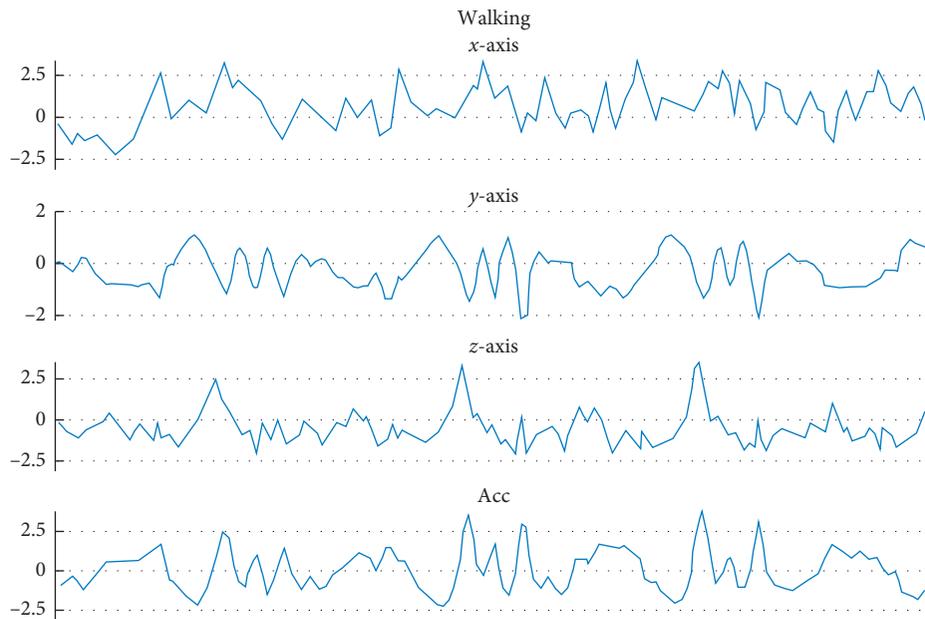


FIGURE 9: Walking 4 s sampling waveform.

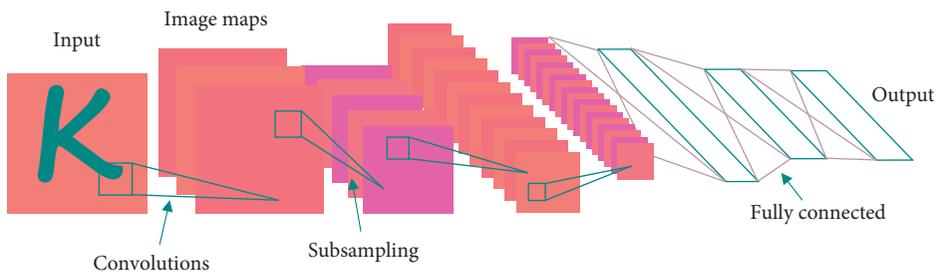


FIGURE 10: LeNet-5 network structure.

This paper will use Pandas and NumPy for related data processing; use some functions under the scikit-learn package for machine learning analysis; and Matplotlib is the suite we use to plot. Therefore, the reader can install these separately. Of course, for convenience, we can directly install Anaconda like this article, which integrates many third-party libraries related to scientific computing, such as NumPy and Pandas. The whole process includes four components, which are data preprocessing, data normalization, data sampling, and saving of data.

6. Experiment and Evaluation

According to the convolutional neural network model designed in Section 3, we use the function method provided by TensorFlow to build the network and configure the corresponding parameters.

6.1. Training Model. In order to train the model, we need to define an index to evaluate the quality of the model. In general, we define a loss index, the so-called loss to indicate that the model is bad, and then try to minimize this level of index. This paper uses cross entropy as the loss function. We do not make a detailed derivation of cross entropy but give its definition as follows:

$$H_{y'}(y) = - \sum_i y'_i \log(y_i). \quad (6)$$

Among them, y_i is the predicted probability distribution, and y_i is the actual distribution. In actual implementation, we compute the code as follows:

$$\text{cross_entropy} = -\text{tf.reduce_sum}(Y \times \text{tf. log}(y_i)). \quad (7)$$

Among them, $y_$ corresponds to y_i in the above formula, and Y corresponds to y_i in the formula, and cross_entropy is the cross entropy loss we define. Then, we choose the shaving descent algorithm to continuously back-propagate and modify the variable values to reduce costs. In this paper, the batch size is set to 50, the learning rate is 0.0001, and the number of iterations is 4 times. Finally, the test result with a correct rate of 98.24% is obtained. The reason why we only need to iterate 4 times to achieve convergence, this article believes that because of our previous expansion of the data, a lot of similar data appeared in the training set, which accelerated the convergence of the network. Imagine if batch training is performed in the same two batches, the training effect of one iteration can be equivalent to two.

6.2. Evaluation Model and Parameter Tuning. This paper uses the TensorBoard tool to track the changes in the loss and accuracy values during the training process. From the change of the two values of our final model, it can be seen that the loss value we finally got to the model quickly went to 0 during the training process, and the accuracy value was in the training process quickly reached a level close to 1. Therefore, it can be judged that the effect of the generated model is still relatively good.

The loss in this article uses the cross-entropy loss mentioned above, which is not repeated here. We introduce how to obtain accuracy. We can use the `tensorflow.argmax` function to give the index value of the maximum data value of a tensor object in a certain dimension. And labels we previously changed to one-hot encoding through the `pandas.get_dummies` function, so the label values are all composed of 0 and 1. Therefore, it can be known whether the prediction is correct by comparing whether the index of the predicted label value and the actual correct label value are the same. Then calculate the proportion of the number of correct predictions, which is the correct rate of prediction. This value is important for model evaluation.

In addition, this paper also uses 3 indicators to evaluate the quality of the model: F1-score, recall, and precision. The higher the recall score, the stronger the model's ability to recognize positive samples. The higher the precision score, the stronger the model's ability to distinguish negative samples. This shows that the model is more robust. The calculation formula of the three indicators is as follows:

$$\begin{aligned} P &= \frac{TP}{TP + FP}, \\ R &= \frac{TP}{TP + FN}, \\ F1 - \text{score} &= \frac{2P \times R}{P + R}, \end{aligned} \quad (8)$$

where P denotes precision and R denotes recall. TP represents true positive samples, TN represents true negative samples, FP represents the false positive samples, and FN represents the false positive samples.

In order to improve the scores of these indicators, this paper uses the control variable method to compare multiple groups of experiments and adjust the model parameters. In the end, after continuous optimization, the three indicators scored by the model generated by the test using the test set are that precision is 0.9825, recall is 0.9824, and F1-score is 0.9823. Finally, we applied the model to output the corresponding confusion matrix on the test set. The so-called confusion matrix is to use columns to represent the predicted value that is given by the model according to the input, and the rows correspond to the ground-truth categories. Through the confusion matrix, we can intuitively see that based on the predictions made by the model, we can make a better decision on the status of our model. The confusion matrix for the test set application to the final model is given in Table 3.

From the analysis of the confusion matrix, we can conclude that we made 10608 predictions, of which 187 were wrong. There are 72 errors on the upstairs; the error rate is 4.55%. It can be seen that going upstairs is more likely to be confused with going downstairs and walking. There are 52 errors on the walk; the error rate is 3%. It is shown that walking is more likely to be confused with going upstairs. There are 51 errors on the running; the error rate is 4.35%. Running is more likely to be confused with going downstairs and upstairs. There are 12 errors on the downstairs; the error

TABLE 3: Prediction confusion matrix.

	Downstairs	Running	Sitting	Standing	Upstairs	Walking
Downstairs	1773	2	0	0	6	4
Running	20	1122	0	0	31	0
Sitting	0	0	1836	0	0	0
Standing	0	0	0	2499	0	0
Upstairs	45	0	0	0	1509	27
Walking	9	9	0	10	24	1682

rate is 0.67%. This activity is more likely to be confused with going upstairs and walk. The accuracy of sit and stand is 100%. Therefore, it can be concluded that the model is more accurate in the recognition of standing, sitting, and going down, while the recognition of going up, running, and walking is slightly inferior.

7. Conclusion

This paper starts with data collection, uses convolutional neural network modeling, and finally transplants the model back to the mobile phone to complete the activity recognition system. After a series of experiments and tests, we found that the method of collecting data based on mobile phone sensors and then training the model through a convolutional neural network can well complete the task of activity recognition. In the absence of data, this paper can finally train a model with an accuracy rate of more than 98% through some operations on the data, model optimization, and parameter adjustment.

We verify the practical feasibility of the model by transplanting it to a real machine test. Therefore, this article concludes that it is completely feasible to train a convolutional neural network model based on mobile phone sensor data to complete an activity recognition system, and this method has great potential.

Data Availability

The underlying data supporting the results of this paper are generated during the study.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Major Special Science and Technology Project of Hainan Province (Grant no. ZDKJ2017012), the National Key R&D Program of China (no. 2020YFB2104004), and the Qinghai Key R&D and Transformation Project (no. 2021-GX-112).

References

- [1] F. Xiao, J. Chen, X. Xie, L. Gui, L. Sun, and R. Wang, "SEARE: a system for exercise activity recognition and quality

- evaluation based on green sensing," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 3, pp. 752–761, 2018.
- [2] H. Gao, W. Huang, and Y. Duan, "The cloud-edge-based dynamic reconfiguration to service workflow for mobile ecommerce environments: a QoS prediction perspective," *ACM Transactions on Internet Technology*, vol. 21, no. 1, 2020.
- [3] M. Saadat, S. Sur, S. Nelakuditi, and P. Ramanathan, "MilliCam: hand-held millimeter-wave imaging," in *Proceedings of 29th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–9, Honolulu, HI, USA, August 2020.
- [4] X. Yang, S. Zhou, and M. Cao, "An approach to alleviate the sparsity problem of hybrid collaborative filtering based recommendations: the product-attribute perspective from user reviews," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 376–390, 2020.
- [5] H. Gao, L. Kuang, Y. Yin, B. Guo, and K. Dou, "Mining consuming behaviors with temporal evolution for personalized recommendation in mobile marketing apps," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1233–1248, 2020.
- [6] R. Zhang, X. Jing, S. Wu, C. Jiang, J. Mu, and F. Yu, "Device-free wireless sensing for human detection: the deep learning perspective," *IEEE Internet of Things Journal*, vol. 8, no. 4, pp. 2517–2539, 2020.
- [7] F. Wang, W. Gong, J. Liu, and K. Wu, "Channel selective activity recognition with WiFi: a deep learning approach exploring wideband information," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 181–192, 2018.
- [8] H. Xue, J. Yu, F. Lyu, and M. Li, "Push the limit of multipath profiling using commodity WiFi devices with limited bandwidth," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4142–4154, 2020.
- [9] F. Zhang, Z. Chang, K. Niu et al., "Exploring LoRa for long-range through-wall sensing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–27, 2020.
- [10] P. Wijewardena, A. Bhaskara, S. Kasera, S. Mahmud, and N. Patwari, "A plug-n-play game theoretic framework for defending against radio window attacks," in *Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 284–294, Linz, Austria, July 2020.
- [11] J. Liu, H. Liu, Y. Chen, Y. Wang, and C. Wang, "Wireless sensing for human activity: a survey," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1629–1645, 2019.
- [12] H. Gao, C. Liu, Y. Li, and X. Yang, "V2VR: reliable hybrid-network-oriented V2V data transmission and routing considering RSUs and connectivity probability," *Proceedings of IEEE Transactions on Intelligent Transportation Systems (T-ITS)*, early access, 2020.
- [13] Y. He, Y. Chen, Y. Hu, and B. Zeng, "WiFi vision: sensing, recognition, and detection with commodity MIMO-OFDM WiFi," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 8296–8317, 2020.

- [14] Z. Han, Z. Lu, X. Wen, J. Zhao, L. Guo, and Y. Liu, "In-air handwriting by passive gesture tracking using commodity WiFi," *IEEE Communications Letters*, vol. 24, no. 11, pp. 2652–2656, 2020.
- [15] Y. Yin, Q. Huang, H. Gao, and Y. Xu, "Personalized APIs recommendation with cognitive knowledge mining for industrial systems," *IEEE Transactions on Industrial Informatics*, early access, 2020.
- [16] H. Yin, A. Zhou, G. Su, B. Chen, L. Liu, and H. Ma, "Learning to recognize handwriting input with acoustic features," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–26, 2020.
- [17] D. Y. Huang, N. Aphorpe, F. Li, G. Acar, N. Feamster, and IoT. Inspector, "IoT inspector," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–21, 2020.
- [18] S. Yue, Y. Yang, H. Wang, H. Rahul, and D. Katabi, "BodyCompass," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 2, pp. 1–25, 2020.
- [19] J. Chen, F. Li, H. Chen, S. Yang, and Y. Wang, "Dynamic gesture recognition using wireless signals with less disturbance," *Personal and Ubiquitous Computing*, vol. 23, no. 1, pp. 17–27, 2019.
- [20] R. W. Heath, "Communications and sensing: an opportunity for automotive systems [from the editor]," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 3–13, 2020.
- [21] K. Kleisouris, B. Firner, R. Howard, Y. Zhang, and R. P. Martin, "Detecting intra-room mobility with signal strength descriptors," in *Proceedings of ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 71–80, Chicago, IL, USA, September 2010.
- [22] B. Wei, W. Hu, M. Yang, and C. T. Chou, "From real to complex," *ACM Transactions on Sensor Networks*, vol. 15, no. 3, pp. 1–32, 2019.
- [23] F. Adib and D. Katabi, "See through walls with wifi!" in *Proceedings of ACM SIGCOMM Conference*, pp. 75–86, Hong Kong, China, August 2013.
- [24] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: gathering 802.11n traces with channel state information," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, p. 53, 2011.
- [25] Y. Wang, K. Wu, L. Ni, and M. Wifall, "Device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, 2016.
- [26] X. Zheng, J. Wang, and L. Shangguan, "Ubiquitous smoking detection with commercial wifi infrastructures. Computer Communications," in *Proceedings of the 35th IEEE International Conference on Computer Communications (INFOCOM)*, pp. 1–9, San Francisco, CA, USA, April 2016.
- [27] R. Li, Z. Jiang, Y. Xu, H. Gao, F. Chen, and J. Du, "Device-free indoor multi-target tracking in mobile environment," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1195–1207, 2020.
- [28] J. Lei, X. Ren, and D. Fox, "Fine-grained kitchen activity recognition using rgb-d," in *Proceedings of ACM Conference on Ubiquitous Computing (UbiComp)*, pp. 208–211, Pittsburgh, PA, USA, September 2012.
- [29] C. Karanam, B. Korany, and Y. Mostofi, "Magnitude-based angle-of-arrival estimation, localization, and target tracking," in *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pp. 254–265, Porto, Portugal, April 2018.
- [30] M. Keally, G. Zhou, G. Xing, J. Wu, and A. J. Pyles, "PBN: towards practical activity recognition using smartphone-based body sensor networks," in *Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pp. 246–259, Washington, DC, USA, November 2011.
- [31] J. Lu, X. Zheng, and M. Sheng, "Efficient human activity recognition using a single wearable sensor," *IEEE Internet of Things Journal*, vol. 7, no. 11, pp. 1137–1146, 2020.
- [32] K. Yatani and K. N. Truong, "Bodyscope. A wearable acoustic sensor for activity recognition," in *Proceedings of the ACM Conference on Ubiquitous Computing (UbiComp)*, pp. 341–350, Pittsburgh, PA, USA, September 2012.
- [33] S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, and K. Fu, "They can hear your heartbeats," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4, pp. 2–13, 2011.
- [34] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We can hear you with wi-fi!" *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2907–2920, 2016.
- [35] L. Kuang, T. Gong, S. OuYang, H. Gao, and S. Deng, "Off-loading decision methods for multiple users with structured tasks in edge computing for smart cities," *Future Generation Computer Systems*, vol. 105, pp. 717–729, 2020.
- [36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deepconvolutional neural networks," in *Proceedings of Conference on Advances in Neural Information Processing Systems (NeurIPS)*, pp. 1097–1105, Lake Tahoe, NV, USA, December 2012.
- [37] G. Kalouris, E. I. Zacharaki, and V. Megalooikonomou, "Improving CNN-based activity recognition by data augmentation and transfer learning," in *Proceedings of IEEE International Conference on Industrial Informatics (ICII)*, pp. 1387–1394, Xi'an, China, June 2019.
- [38] P. Khaire, P. Kumar, and J. Imran, "Combining CNN streams of RGB-D and skeletal data for human activity recognition," *Pattern Recognition Letters*, vol. 115, pp. 107–116, 2018.
- [39] J. Zhang, F. Wu, and B. Wei, "Data augmentation and dense-LSTM for human activity recognition using WiFi signal," *IEEE Internet of Things Journal*, early access, 2020.
- [40] Z. Wang, Z. Yu, X. Lou, B. Guo, and L. Chen, "Gesture-Radar: a dual Doppler radar based system for robust recognition and quantitative profiling of human gestures," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 1, pp. 32–43, 2020.