Device-Free In-Air Gesture Recognition Based on RFID Tag Array

WU Jiaying, WANG Chuyu, XIE Lei
(State Key Laboratory for Novel Software Technology, Nanjing 210023, China)

Abstract: Due to the function of gestures to convey information, gesture recognition plays a more and more important part in human-computer interaction. Traditional methods to recognize gestures are mostly device-based, which means users need to contact the devices. To overcome the inconvenience of the device-based methods, studies on device-free gesture recognition have been conducted. However, computer vision methods bring privacy issues and light interference problems. Therefore, we turn to wireless technology. In this paper, we propose a device-free in-air gesture recognition method based on radio frequency identification (RFID) tag array. By capturing the signals reflected by gestures, we can extract the gesture features. For dynamic gestures, both temporal and spatial features need to be considered. For static gestures, spatial feature is the key, for which a neural network is adopted to recognize the gestures. Experiments show that the accuracy of dynamic gesture recognition on the test set is 92.17%, while the accuracy of static ones is 91.67%.

Keywords: gesture recognition; RFID tag array; neural network

DOI: 10.12142/ZTECOM.202103003

https://kns.cnki.net/kcms/detail/34.1294.TN.20210816.1023.002.html, published online August 16, 2021

Manuscript received: 2021-06-10

Citation (IEEE Format): J. Y. Wu, C. Y. Wang, and L. Xie, "Device-free in-air gesture recognition based on RFID tag array," *ZTE Communications*, vol. 19, no. 3, pp. 13 - 21, Sept. 2021. doi: 10.12142/ZTECOM.202103003.

1 Introduction

Recent years have seen the rapid growth of human-computer interaction and its applications range from somatosensory games to smart screens. In these applications, gestures are an important way to convey information. How to perceive gestures through the device, so as to realize accurate recognition and natural human-computer interaction, is a research hotspot.

Methods of gesture recognition include device-based ones and device-free ones. The device-based methods need users to wear or touch the device, so as to perceive human action[1-2]. However, wearing a device is not natural for users and the battery is a big headache. On the contrary, the device-free ones do not need users to touch the devices. They use computer vision or wireless technologies instead[3-4]. Though accurate, computer vision brings privacy concerns and is sensitive to light interference. Therefore, we turn to radio frequency identification (RFID), which is a kind of wireless technology. Based on RFID, users only need to perform

This work was supported by National Natural Science Foundation of China under Grant Nos. 61902175, 61872174 and 61832008 and Natural Science Foundation of China under Grant No. BK20190293.

gestures in the air naturally and do no need to care about the privacy issues.

Here comes the question: how to recognize gestures based on RFID? With regard to hand gestures, through a passive RFID tag, we can get the signal reflected by the hand. However, one tag is far from enough to accurately recognize gestures. Therefore, we use tag array to sense gestures. When performing a gesture, the hand has different influences on different parts of the tag array, which provides more diverse information for recognition. Features extracted from the signal along time can be regarded as an image sequence, which contains both spatial features and temporal features. For dynamic gestures, we should take both the spatial and temporal features into consideration. Here, a combined convolutional neural network and Long Short-Term Memory (CNN-LSTM) system is proposed to process spatial features from the tag array by the CNN and process temporal features in the image sequence by the LSTM. For static gestures, we can get the final feature image from the image sequence as the snapshot of the gesture, and use CNN to recognize it.

There exist some challenges, however. First, it is important to improve the robustness of recognition. When different users perform gestures at different speeds, the system need to be robust enough to recognize them. Second, for dynamic gestures, both spatial and temporal features need to be considered, which should be dealt with carefully. Third, for static gestures, we need to extract their features from dynamic signals. How to decide the final features for recognition is the key.

In this paper, our contributions are shown as follows:

1) We propose a device-free in-air gesture recognition method based on RFID tag array, which can recognize dynamic gestures and static gestures.

2) For dynamic gestures, we take both spatial and temporal features into account and discuss several structures for recognition. CNN and LSTM are combined to get better performance and adjustment is made to improve the robustness.

3) We implement a gesture recognition system based on our method. Experiments show that the accuracy of dynamic gesture recognition on the test set is 92.17%, and the accuracy of static ones is 91.67%.

2 Related Work

When it comes to human-computer interaction, there is no denying that action recognition is an important part. Through action, users convey order or information and interact with computer. From the perspective of the body part to be recognized, action recognition can be divided into three kinds[5]: gesture recognition, head and facial action recognition, and overall body action recognition. In this paper, we focus on gesture recognition, including dynamic gesture recognition and static gesture recognition. According to whether the user needs to wear or touch the device, action recognition can also be divided into device-based and device-free.

2.1 Device-Based Methods

Device-based methods need users to wear or touch and input the device. These methods include mechanical, tactile, ultrasonic, inertial and magnetic methods[6]. Wearable devices usually contain sensors such as accelerometers and gyroscopes, and based on these, they capture and recognize action[2, 7]. The signal returned by a sensor changes along with its moving, making it possible for us to extract action-related features. Sometimes, RFID tags can be attached to gloves or bracelets, acting as sensors[8–9]. Electromyography (EMG) and force myography (FMG) are also used to recognize action. JIANG et al.[1] propose a novel co-located approach (EMG and FMG) for capturing both sensing modalities, simultaneously, at the same location, so as to better recognize the gesture.

2.2 Device-Free Methods

Device-based methods can accurately perceive the gesture, but they are not natural for users. Besides, the battery problem is a big headache, making it more inconvenient. Therefore, researchers turn to device-free methods. Computer vision is a typical device-free method[3, 4, 10]. Structure, color and even depth information can be provided through ordinary cameras or depth cameras. However, as people pay more and more attention to privacy issues, they tend to refuse computer vision. With regard to wireless technologies, privacy is no longer a problem. These kinds of methods use wireless signals, usually electromagnetic or acoustic, to capture the action and recognize it. For examples, the Low-Latency Acoustic Phase (LLAP)[11] scheme uses ultrasonic signals to recognize character gestures. WiGest[12] uses WiFi signals. MHomeGes[13] recognizes arm gesture based on mmWave signals. As for RFID, RFIPad[14] makes use of the phase changes and received signal strength to recognize stroke movements and detect direction through, so as to recognize basic character gestures. Using hierarchical recognition, image processing and polynomial fitting, TagSheet[15] enables the recognition of sleeping postures.

3 Preliminaries

Originating from radar technology, RFID use radio frequency signals to sense and identify targets. A typical RFID system usually consists of readers, antennas and tags[16]. RFID tags include active tags, semi-active tags and passive tags. Active tags are battery-assisted, while passive ones need a reader to power them. Once powered by the signal from the reader, an RFID tag will transmit back the signal with its own information.

When a hand is put in front of a tag S, the signal returned by the tag includes the signal directly transmitted to the

tag S_{tag} and the signal reflected by the hand S_{hand}:

$$S = S_{tag} + S_{hand}. \tag{1}$$

Readings returned by a tag S include phase θ (rad) and Received Signal Strength Indication (RSSI) R (dBm). Therefore, with the hand in front of the tag, we can calculate the signal S[17]:

$$S = \sqrt{10^{\frac{R}{10}-3}}\, e^{j\theta} = \sqrt{10^{\frac{R}{10}-3}}\cos\theta + j\sqrt{10^{\frac{R}{10}-3}}\sin\theta. \tag{2}$$

Without the hand in front of the tag, S_{tag} can be calculated in the same way. Therefore, by subtracting S_{tag}, the signal reflected by the hand S_{hand} is obtained according to Eq. (1). Based on this, the actual power P_{actl} of S_{hand} and its theoretical power P_{theor} can be calculated:

$$P_{actl} = \left| S_{hand} \right|^2, \tag{3}$$

$$P_{theor} = \frac{\mathrm{C}}{d^4}, \tag{4}$$

where C is a constant and d is the distance from the hand to the tag.

According to the algorithm in Ref. [17], an actual power map and certain a theoretical power map can be got from a tag array. Based on these two maps, a possibility map can be created. A series of possibility maps along time form the feature image sequence we want.

4 Gesture Recognition System

Our device-free in-air gesture recognition system includes a feature extraction module and a gesture recognition module.

4.1 Feature Extraction

The feature image sequence can be extracted based on the preliminaries mentioned above. The extraction process includes preprocessing, gesture region segmentation and sequence generation.

1) Data preprocessing. A sliding window is used to preprocess the data. In this phase, several consecutive readings are combined into one, while the vacant readings of some tags are interpolation. The moving average filter is used to smooth the signal.

2) Gesture region segmentation. When there is a hand in front of the tag array, the signal we get will be far different from the one without a hand. Therefore, by calculating the distance to the signal without a hand, we can segment the gesture region whose distance is larger than the threshold.

3) Feature image sequence generation. Based on the former steps, an actual power map and certain a theoretical power map can be obtained. The feature image sequence will then be calculated by the algorithm in Ref. [17]. For a 5×7 RFID tag array, we can get a 15×21 cm^2 feature image and each pixel of the image measures the possibility that the hand is in front of the pixel grid. Fig. 1 shows such a feature image. The brighter a pixel grid on the map, the more likely it is that the hand will be in front of the corresponding pixel grid.

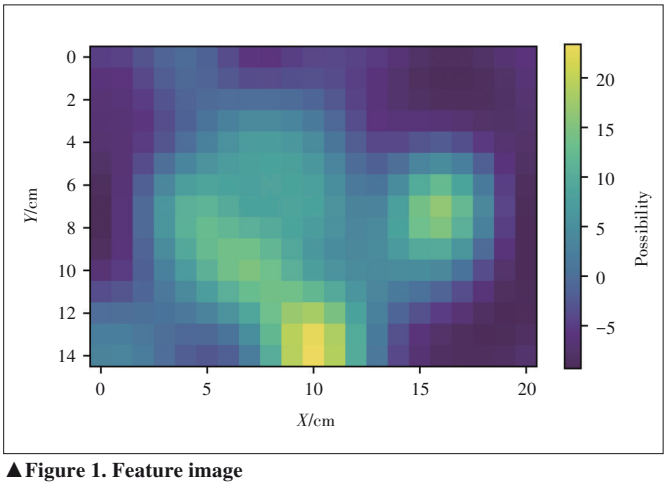
4.2 Dynamic Gesture Recognition

After a feature image sequence of a dynamic gesture is extracted, the problem of dynamic gesture recognition is transformed into a feature image sequence recognition problem. CNN has advantages in extracting spatial features from an image, while LSTM can handle the temporal feature in sequence well. In order to focus both spatial and temporal features in a feature image sequence, we can combine them as CNN-LSTM.

4.2.1 Network Structure

For efficiency, feature images in the long image sequence will be divided evenly into five groups. Images in the same group will be superimposed into one frame of the feature image. In this way, the original long image sequence is converted into a shorter sequence, which contains five frames of the feature image. The size of each image is 15×21 cm^2. Moreover, totally six kinds of gestures are needed to be recognized, so the label of each sample is coded as a six-dimensional one-hot code. The type corresponding to the highest value of the output vector is the predicted gesture type. Here, we adopt cross entropy as the loss function. To recognize the dynamic gesture from the feature sequence, we start with the CNN structure, then move to the LSTM structure, and finally discuss CNN-LSTM, the combined one.

1) CNN structure. To make it possible for CNN to learn from the data, images in the sequence are vertically stitched according to their temporal relationship. Then, the 75×21 cm^2 image is fed into the network as a sample input. The CNN structure is: the input layer, convolutional layer-1, pooling lay-



▲Figure 1. Feature image

er-1, convolutional layer-2, pooling layer-2, fully connected layer, and output layer. Activation function of the convolutional layer and fully connected layer is Rectified Linear Unit (ReLU), and one of the output layer is softmax. The kernel sizes of convolutional layer-1 and layer-2 are 3×3×*n_conv*1, 3×3×*n_conv*2, respectively. Besides, the fully connected layer maps the extracted features into an *n_fc*-dimensional vector. Here, (*n_conv*1, *n_conv*2, *n_fc*) forms the hyperparameters of the CNN structure.

2) LSTM structure. LSTM takes sequence data as input, with each element in sequence being a vector. Therefore, we flatten each image in the feature image sequence into a vector and feed them into LSTM along time. With all feature vectors fed, LSTM advances five steps in time dimension. Here, we take the output of last time dimension and map it into the output vector. As for hyperparameters, the number of hidden units *n_hd* is our target, which determines the ability of LSTM to extract temporal information along time sequence.

3) CNN-LSTM combined structure. For a dynamic gesture and its feature image sequence, CNN focuses on the spatial characteristic in image, which carry information of gesture impacts on different parts of the tag array, while LSTM focuses on the temporal characteristic in sequence, which carry information of the changes of the dynamic gesture along time. We combine them as CNN-LSTM and both spatial and temporal features are focused. Table 1 and Fig. 2 show the CNN-LSTM structure. The CNN part takes each image in feature image sequence as input and outputs a summary vector for the LSTM part. The LSTM part takes the summary vector sequence as input and extracts its temporal feature. In the last time step, the prediction vector is obtained through mapping. Here, (*n_conv*1, *n_conv*2, *n_fc*, *n_hd*) forms the hyperparameters.

4.2.2 Overfitting and Adjustment

In the process of learning, we should be vigilant against overfitting. In training, the loss on a training set declines, but the loss on the validation set tends to be flat or even starts to rise. It means that the model is still learning from the training set, but the features it learns tend to include irrelevant features, which is harmful to the generalization ability of the model.

To deal with the overfitting, we add a dropout layer between the CNN part and LSTM part. In training, it randomly drops neurons at this layer. All neurons will be used for prediction. Fig. 3 shows the loss curve with and without the dropout layer. As we can see, the validation loss begins to arise after about 130 epochs without dropout, indicating that overfitting occurs. On the contrary, this phenomenon is well-suppressed with dropout. With ten-fold cross validation, we set the dropout rate as 20%.

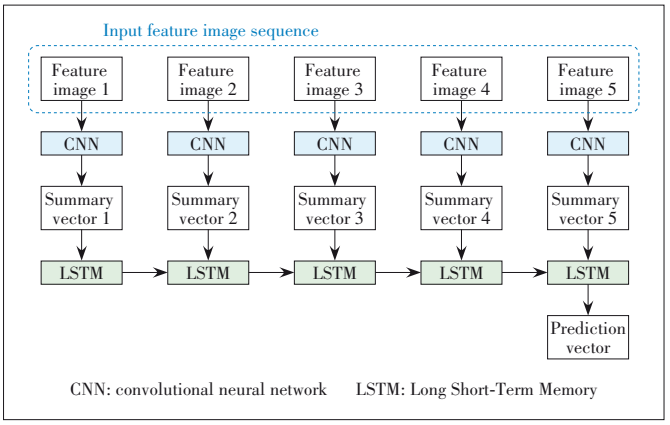
For other methods, one may think of regularization, such as L2 regularization, which adds a penalty term to loss function. L2 regularization tends to suppress excessive weight parameters. But for CNN, it doesn't make much sense. For LSTM, limitation to weight parameters will lead to rapid disappearance of the learned temporal information along time[18]. Therefore, we don't use regularization to suppress overfitting of CNN-LSTM.

Apart from the dropout layer, we adopt early stopping strategy in training to avoid serious overfitting. The red line in Fig. 3 is an example of early stopping line. If the loss reduction of the training set is less than a certain threshold or even the loss begins to rise, the training is considered to have made no progress in this epoch. If the model keeps making

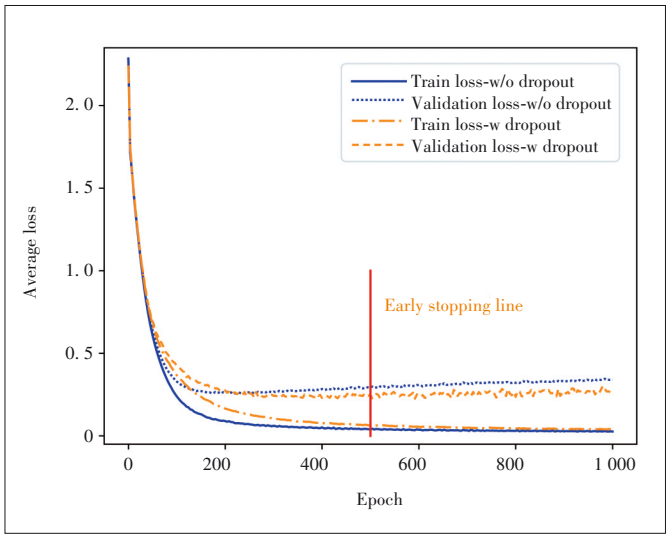
▼**Table 1. CNN-LSTM structure**

	Layer	Description
CNN part	Input layer	Input: feature image sequence Length of sequence: 5 Image size: 15×21
	Convolution layer-1	Extract *n_conv*1 features from the image Kernel size: 3×3×*n_conv*1 Step size: 1 Activation function: ReLU
	Pooling layer-1	Downsample the extracted features Pooling type: max pooling Template size: 2×2 Step size: 2
	Convolution layer-2	Extract *n_conv*2 features from the image Kernel size: 3×3×*n_conv*2 Step size: 1 Activation function: ReLU
	Pooling layer-2	Downsample the extracted features Pooling type: max pooling Template size: 2×2 Step size: 2
	Fully connected layer	Fully connect the features to *n_fc*-dimensional summary vector
LSTM part	LSTM layer	Extract features from summary vector sequence Time steps: 5 Number of hidden units: *n_hd*
	Fully connected layer	Fully connect the features to six-dimensional prediction vector

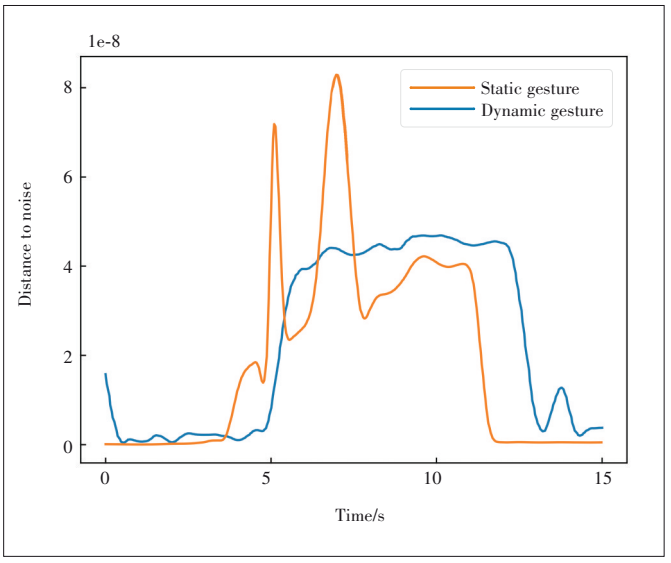
CNN: convolutional neural network LSTM: Long Short-Term Memory
ReLU: Rectified Linear Unit



▲**Figure 2. CNN-LSTM structure**



▲Figure 3. Loss curves for dynamic gesture recognition with and without dropout



▲Figure 4. Static gesture recognition by the distance to noise

no progress for certain epochs, we can stop the training in time. This prevents the model from continuing to learn even though it tends to overfit.

4.3 Static Gesture Recognition

For static gestures, when we put our hands in front of the tag array, the signal we received is still different from the signal without hands, which is defined as noise floor. Through the feature extraction module, we can still extract its feature. However, it is still difficult to distinguish between dynamic gestures and static gestures. We use the distance to noise floor to solve this problem. For a single tag, we calculate the difference between its current signal and the floor noise. And then, a module operation and a square operation are performed on this difference to get the distance to noise floor for a single tag. The sum of distances of all tags forms the distance of the tag array. As shown in Fig. 4, when a user is performing a dynamic gesture, the distance to noise floor changes a lot due to the movement of the hand. On the contrary, for a static gesture, the change is relatively small. Therefore, we can determine whether the signal is from a static gesture through the variance of the signal. If the variance is smaller than a threshold, the corresponding signal is regarded as being from a static gesture.

4.3.1 Feature Decision

With the feature image sequence extracted, how can we decide the final feature and deal with it? If we use the CNN-LSTM structure above, it makes no sense for the LSTM part to extract temporal features because a static gesture keeps unchanged when acquiring the signal, and there is almost no temporal change relation between the two adjacent feature images. Therefore, for a static gesture, the key is still the spatial characteristic—how the gesture affects different parts of the RFID tag array? Given this, there are two strategies for getting the final feature.

One is compression. The whole feature image sequence is compressed into one feature image. In other words, the final feature is the superimposition of the images in sequence. In this way, additive noise may be suppressed. It can be regarded as a smoothing method, which smooths the possible noise, but smooth the feature to a certain extent in the meantime.

The other is extracting or directly picking one feature image as the final feature. This can be regarded as a snapshot of the static gesture or an instant feature of it. Without smoothing, instant feature will remain. But at the same time, possible noise may also remain.

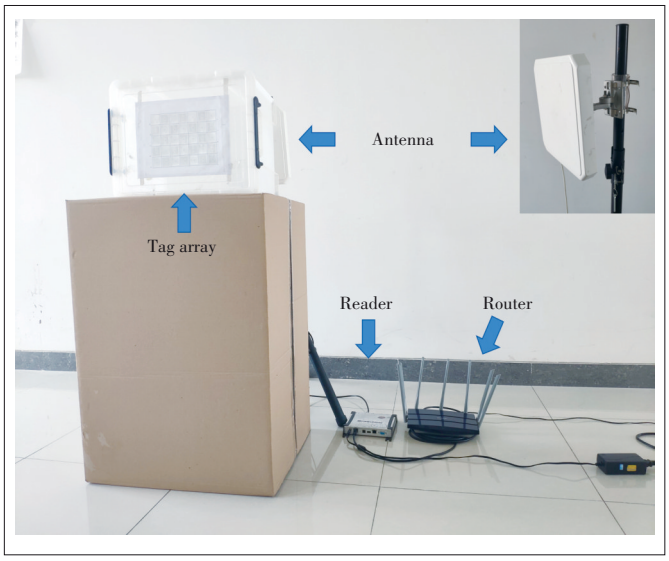
4.3.2 Network Structure

With the final feature image decided, the static gesture recognition problem is transformed into a feature image recognition problem. Here, we can adopt the CNN structure mention in Section 4.2.1 (the input layer, convolutional layer-1, pooling layer-1, convolutional layer-2, pooling layer-2, fully connected layer and output layer). The convolutional layers extract spatial features in the static gesture feature image, and the pooling layers down sample the features and reduce training overhead. The output layer outputs the final prediction vector.

5 Evaluation

5.1 Experiment Setup

Experiments are carried out in laboratory environment. As shown in Fig. 5, RF signal is transmitted through Impinj Speedway Revolution R420 UHF RFID Reader with laird s9028pcl RFID antenna. 5×7 AZ-9629 RFID tags are de-



▲Figure 5. Experiment deployment

ployed into a tag array and placed 0.5 m in front of the antenna. Users perform gestures in front of the array and PC gets signals from the reader through the router and then recognize them.

For dynamic gestures, six kinds of gestures are needed to be recognized: rotating left, rotating right, swiping left, swiping right, zooming in, and zooming out. We collect 3 600 samples, with 600 samples for each gesture. For static gestures, six kinds of gestures are needed to be recognized, including one-hand gestures (stop, victory, good, ok) and two-hand gestures (wrong, heart). We collect 600 samples, with 100 samples for each gesture. Fig. 6 shows the dynamic gestures and static gestures.

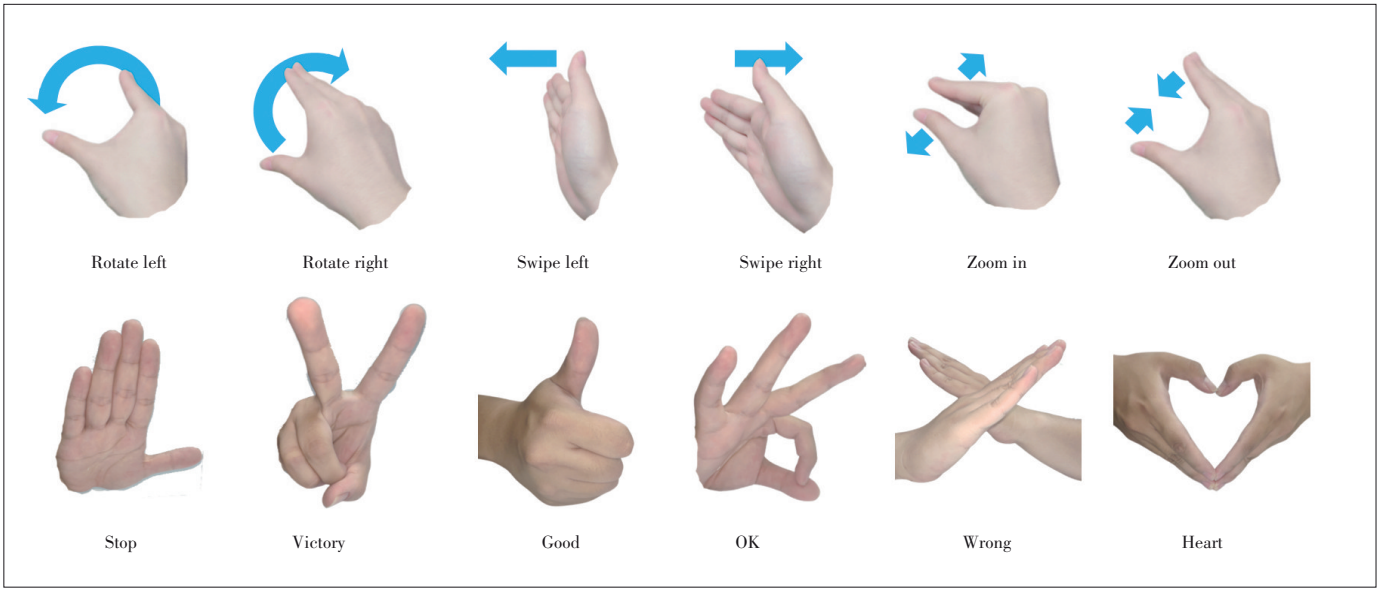
5.2 Evaluation on Model Structure

5.2.1 CNN Hyperparameter

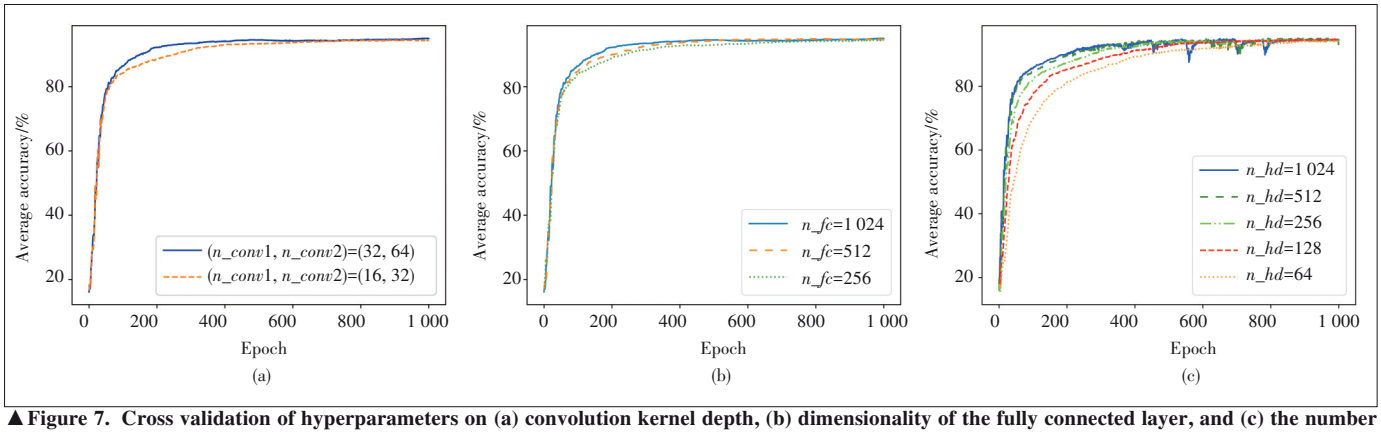
For the CNN structure mentioned in Section 4.2.1, we need to decide its hyperparameters(n_conv1, n_conv2, n_fc), that is, the kernel depths of the two convolutional layers and the feature dimensionality of the FC (fully connected) layer. To choose proper hyperparameters, we use ten-fold cross validation. The candidate values for hyperparameters (n_conv1, n_conv2) are $(32, 64)$ and $(16, 32)$. For a pure CNN structure, the average accuracy curve of different values on the validation set is shown in Fig. 7(a). Similarly, n_fc has candidate values as 1 024, 512 and 256 and the corresponding accuracy curve is shown in Fig. 7(b). The larger n_conv1 is, the less convergence time we need. Moreover, a small n_conv1 means that we can only extract a few features from the feature image sequences, limiting the capability of the model and even harming its performance. However, a too large hyperparameter also means the increased training costs and increased risk of overfitting. The same is true for the other two hyperparameters. Taking all these things into consideration, (n_conv1, n_conv2, n_fc) are set to $(32, 64, 1024)$. For CNN-LSTM structure, they are determined as $(32, 64, 256)$.

5.2.2 LSTM Hyperparameter

For the LSTM structure mentioned in Section 4.2.1, the hyperparameter is the number of hidden units n_hd. Ten-fold cross validation is used again. The average accuracy is shown in Fig. 7(c), with 1 024, 512, 256, 128 and 64 as candidate values. As we can see, if the number of hidden units is set too large, severe jitters will occur in the accuracy curve, indicating unstable performance of the model. But if it is set too small, it will take a long time to train and make the



▲Figure 6. Dynamic and static gestures used in our experiments



▲**Figure 7. Cross validation of hyperparameters on (a) convolution kernel depth, (b) dimensionality of the fully connected layer, and (c) the number of hidden units**

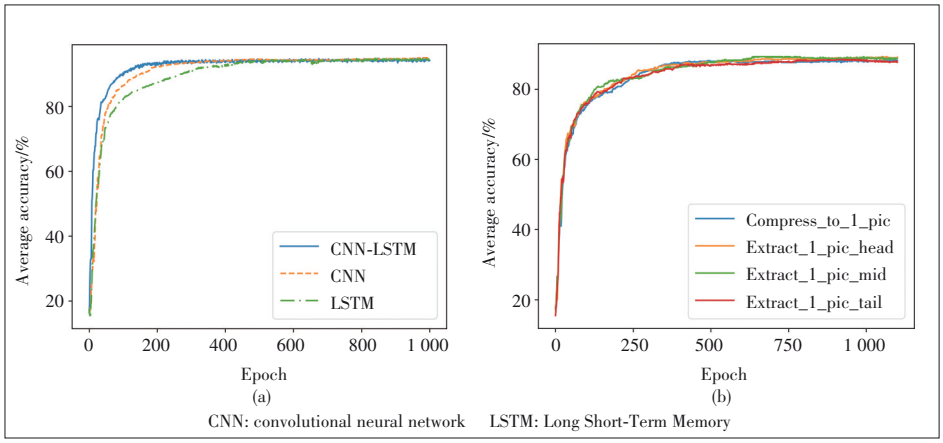
training expensive, limiting the ability of LSTM as well. Therefore, we choose 256 for *n_hd* in a pure LSTM structure and 128 in the CNN-LSTM structure.

5.2.3 Dynamic Gesture Model Selection

For dynamic gesture recognition, we also compare the three structures mentioned above through ten-fold cross validation. The average accuracy curve is shown in Fig. 8(a). In training, CNN focuses on spatial features and LSTM focuses on temporal features, while CNN-LSTM focuses on both of them. Therefore, CNN-LSTM facilitates learning more information in each epoch and costing fewer epochs to converge. That is why CNN-LSTM is chosen as the final network structure for dynamic gesture recognition.

5.2.4 Static Gesture Feature Decision

For static gesture recognition, we test the two strategies to get the final state through ten-fold cross validation. The first group use compression strategy to compress the whole feature image sequence into one picture. The second group use extracting strategy to pick feature images from the head, middle and tail of sequence respectively. From the test results shown in Fig. 8(b), four accuracy curves are close to each other, which means that both of the compression strategy and extracting strategy are feasible for recognition. Actually, these two strategies are tradeoff between smoothing noise and smoothing feature.



▲**Figure 8. Cross validation on (a) model selection for dynamic gesture and (b) feature decision for static gesture**

5.3 Evaluation of Dynamic Gesture Recognition

5.3.1 Overall Performance

With the 3 600 dynamic gesture samples, we take 600 of them as the test set and the rest as the training set. The accuracy on the test set is 92.17%, with the confusion matrix shown in Fig. 9(a). The accuracy of each gesture is above 89% and reaches up to 94%. According to the experiments, the system has accurate results in recognizing various dynamic gestures.

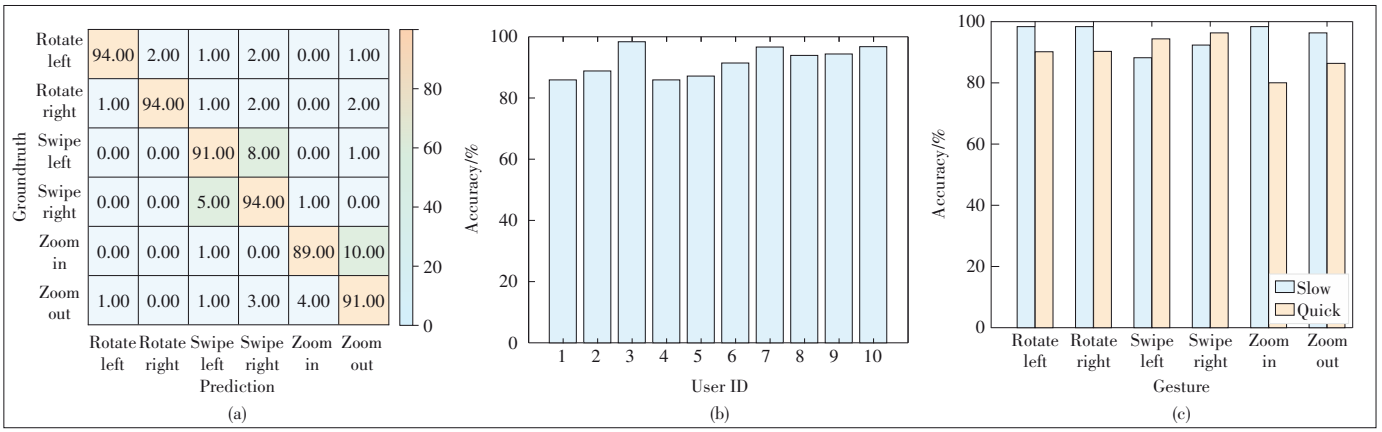
5.3.2 Evaluation on Different Users

For dynamic gestures, there are 10 users participating in the data collection of six gestures. As shown in Fig. 9(b), the accuracy of each user is above 85% and reaches up to 98.28%.

5.3.3 Evaluation at Different Speeds

Out of the 600 samples of each gesture, there are 300 fast gesture samples and 300 slow ones. The results on the test set are shown in Fig. 9(c). It can be seen that some types of gestures have higher accuracy at high speed but lower accuracy at low speed, and the other types have opposite situation. The accuracy of each gesture at different speeds is above 80% and reaches up to 96%.

The experiments show that the device-free in-air gesture recognition system based on RFID tag array can recognize several different types of



▲Figure 9. Evaluation of dynamic gesture recognition: (a) Confusion matrix, (b) accuracy of different users, and (c) accuracy at different speeds

dynamic gestures, and have high accuracy and robustness across different users at different speeds.

5.4 Evaluation of Static Gesture Recognition

5.4.1 Overall Performance

For the 600 static gesture samples, we take 60 of them as the test set and the rest as the training set. The accuracy on the test set is 91.67%. It can be seen from the confusion matrix shown in Fig. 10(a) that the accuracy of each gesture is above 80% and reaches up to 100%. The experiments show that the system has accurate results in recognizing various static gestures.

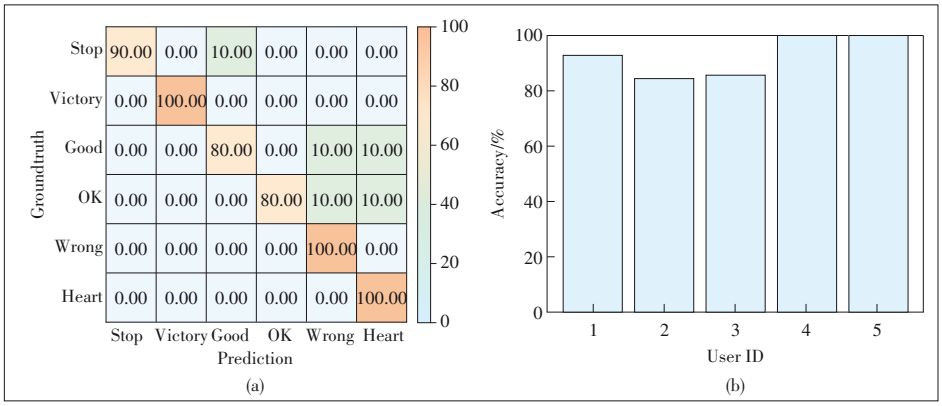
5.4.2 Evaluation on Different Users

There are five users participated in the data collection of six static gestures. The accuracy of different users is shown in Fig. 10(b). The accuracy of each user is above 84.62% and reaches up to 100%.

6 Discussion

1) Sensing distance. When performing a gesture, the recommended distance between the tag array and the hand is from 5 cm to 15 cm. If the hand is too far away from the tag array, the power of the reflected signal from the hand will be too small for us to extract the features, therefore harming the performance of the system. In this case, we need to increase the transmitting power of the reader, or even adopt beamforming technology to increase the power of the reflected signal. Besides, if the hand is too close to the tag array, it is likely to touch the tag array when performing gestures. This will change the input impedance of the tag and dramatically affect the received signal, reducing the accuracy of recognition. In this case, we can detect the touching action and require the user to repeat the gesture if we detect it. Another scheme is to build a more perfect reflection signal model that minimizes the impact of the touch action.

2) Hand size. The hand size will also affect the performance. When a user is performing gestures, signals reflected from the hand make different effects on different tags. The difference in hand sizes is not particularly significant in adults, so the effect on performance is not obvious. However, if the hand size is too small (such as a child's hand), the signal will be weak and affect the performance. To solve this problem, we may collect data of different hand sizes, thus making it possible for the model to extract size-independent features and improve the generalization ability.



▲Figure 10. Evaluation of static gesture recognition: (a) Confusion matrix and (b) accuracy at different speeds

7 Conclusions

This paper proposes a device-free method to recognize in-air dynamic and static gestures based on RFID tag array. The recognition system includes a feature extraction module and a gesture recognition module. Based on the feature image sequence extracted, we compare several structures and use CNN-LSTM to recognize dynamic gestures. For static gestures, the final feature is decid-

ed through two strategies and CNN is used for the recognition. Experiments show this method can recognize different gestures at different speeds across different users. The overall accuracy of the dynamic gesture test set is 92.17% and that of the static gesture test set is 91.67%.

References

[1] JIANG S, GAO Q, LIU H, et al. A novel, co-located EMG-FMG-sensing wearable armband for hand gesture recognition [J]. Sensors and actuators a: physical, 2020, 301: 111738. DOI: 10.1016/j.sna.2019.111738
[2] XIE R, CAO J. Accelerometer-based hand gesture recognition by neural network and similarity matching [J]. IEEE sensors journal, 2016, 16(11): 4537 – 4545. DOI: 10.1109/JSEN.2016.2546942
[3] SUN Y, WENG Y, LUO B, et al. Gesture recognition algorithm based on multi-scale feature fusion in RGB-D images [J]. IET image processing, 2020. DOI: 10.1049/iet-ipr.2020.0148
[4] WANG C, LIU Z, CHAN S C. Superpixel-based hand gesture recognition with kinect depth camera [J]. IEEE transactions on multimedia, 2014, 17(1): 29 – 39. DOI: 10.1109/TMM.2014.2374357
[5] MITRA S, ACHARYA T. Gesture recognition: A survey [J]. IEEE transactions on systems, man, and cybernetics, part C (applications and reviews), 2007, 37(3): 311 – 324. DOI: 10.1109/TSMCC.2007.893280
[6] RAUTARAY S S, AGRAWAL A. Vision based hand gesture recognition for human computer interaction: a survey [J]. Artificial intelligence review, 2015, 43(1): 1 – 54. DOI: 10.1007/s10462-012-9356-9
[7] BHASKARAN K A, NAIR A G, RAM K D, et al. Smart gloves for hand gesture recognition: sign language to speech conversion system [C]//International Conference on Robotics and Automation for Humanitarian Applications (RAHA). Coimbatore, India: IEEE, 2016: 1 – 6. DOI: 10.1109/RAHA.2016.7931887
[8] CHEN B, ZHANG Q, ZHAO R, et al. SGRS: a sequential gesture recognition system using COTS RFID [C]//IEEE Wireless Communications and Networking Conference (WCNC). Barcelona, Spain: IEEE, 2018: 1 – 6. DOI: 10.1109/WCNC.2018.8376998
[9] CHENG K, YE N, MALEKIAN R, et al. In-air gesture interaction: real time hand posture recognition using passive RFID tags [J]. IEEE access, 2019, 7: 94460 – 94472. DOI: 10.1109/ACCESS.2019.2928318
[10] HONGO H, OHYA M, YASUMOTO M, et al. Focus of attention for face and hand gesture recognition using multiple cameras [C]//Fourth IEEE International Conference on Automatic Face and Gesture Recognition Cat. No. PR00580. Grenoble, France: IEEE, 2000: 156 – 161. DOI: 10.1109/AFGR.2000.840627
[11] WANG W, LIU A X, SUN K. Device-free gesture tracking using acoustic signals [C]//22nd Annual International Conference on Mobile Computing and Networking. New York, USA: ACM, 2016: 82 – 94. DOI: 10.1145/2973750.2973764
[12] ABDELNASSER H, YOUSSEF M, HARRAS K A. Wigest: a ubiquitous wifi-based gesture recognition system [C]//IEEE Conference on Computer Communications (INFOCOM). Hong Kong, China: IEEE, 2015: 1472 – 1480. DOI: 10.1109/INFOCOM.2015.7218525
[13] LIU H, WANG Y, ZHOU A, et al. Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing [J]. Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies, 2020, 4 (4): 1 – 28. DOI: 10.1145/3432235
[14] DING H, QIAN C, HAN J, et al. Rfipad: enabling cost-efficient and device-free in-air handwriting using passive tags [C]//37th International Conference on Distributed Computing Systems (ICDCS). Atlanta, USA: IEEE, 2017: 447 – 457. DOI: 10.1109/ICDCS.2017.141
[15] LIU J, CHEN X, CHEN S, et al. TagSheet: sleeping posture recognition with an unobtrusive passive tag matrix [C]//5th IEEE Conference on Computer Communications. Chengdu, China: IEEE, 2019: 874 – 882. DOI: 10.1109/INFOCOM.2019.8737599
[16] DOBKIN D M. The RF in RFID: passive UHF RFID in practice [M]. Oxford, UK: Newnes, 2007: 1 – 493. DOI: 10.1016/B978-0-7506-8209-1.X5001-3
[17] WANG C, LIU J, CHEN Y, et al. Multi-touch in the air: device-free finger tracking and gesture recognition via cots RFID [C]//IEEE Conference on Computer Communications. Honolulu, USA: IEEE, 2018: 1691 – 1699. DOI: 10.1109/INFOCOM.2018.8486346
[18] PASCANU R, MIKOLOV T, BENGIO Y. On the difficulty of training recurrent neural networks [C]//International Conference on Machine Learning. Atlanta, USA: PMLR, 2013: 1310 – 1318

Biographies

WU Jiaying is a Ph.D. student in the Department of Computer Science and Technology, Nanjing University, China, supervised by Prof. XIE Lei and WANG Chuyu. Her research interests include smart sensing and RFID.

WANG Chuyu (chuyu@nju.edu.cn) received his Ph.D. degree in computer science from Nanjing University, China in 2018. He is an assistant professor in the Department of Computer Science and Technology, Nanjing University, China. His research interests include RFID systems, software-defined radio, activity sensing, indoor localization, etc. WANG Chuyu is the corresponding author.

XIE Lei received his B.S. and Ph.D. degrees from Nanjing University, China in 2004 and 2010, respectively, all in computer science. He is a full professor in the Department of Computer Science and Technology, Nanjing University, China. He has published over 100 papers in *IEEE Transactions on Mobile Computing*, *IEEE Transactions on Parallel and Distributed Systems*, *ACM Transactions on Sensor Networks*, ACM MOBICOM, ACM UbiComp, ACM MobiHoc, IEEE INFOCOM, IEEE ICNP, IEEE ICDCS, etc.