Human Activity Recognition Using Smartphones With WiFi Signals

Guiping Lin, Weiwei Jiang, Sicong Xu, Xiaobo Zhou, *Senior Member, IEEE*, Xing Guo, Yujun Zhu, and Xin He, *Member, IEEE*

***Abstract*—In this article, we present a work using a smartphone with an off-the-shelf WiFi router for human activity recognition with various scales. The router serves as a hotspot for transmitting WiFi packets. The smartphone is configured with customized firmware and developed software for capturing WiFi channel state information (CSI) data. We extract the features from the CSI data associated with specific human activities, and utilize the features to classify the activities using machine learning models. To evaluate the system performance, we test 20 types of human activities with different scales including seven small motions, four medium motions, and nine big motions. We recruit 60 participants and spend 140 hours for data collection at various experimental settings, and have 36 000 data points collected in total. Furthermore, for comparison, we adopt three distinct machine learning models, including convolutional neural networks (CNNs), decision tree, and long short-term memory. The results demonstrate that our system can predict these human activities with an overall accuracy of 97.25%. Specifically, our system achieves a mean accuracy of 97.57% for recognizing small-scale motions that are particularly useful for gesture recognition. We then consider the adaptability of the machine learning algorithms in classifying the motions, where CNN achieves the best predicting accuracy. As a result, our system enables human activity recognition in a more ubiquitous and mobile fashion that can potentially enhance a wide range of applications such as gesture control, sign language recognition, etc.**

***Index Terms*—Channel state information (CSI), human activity recognition (HAR), machine learning, smartphone, WiFi sensing.**

I. Introduction

HUMAN activity recognition (HAR) is one of the most important sensing problems in multiple fields including human–computer interaction (HCI), human sensing, and smart homes [1]–[3]. While there are various techniques developed during recent years, the WiFi-based sensing methods attract significant attention for their ubiquity, versatility, and high performance [4], [5]. WiFi-based sensing can be involved into the integrated sensing and communication since the channel information is adopted for not only communication but also sensing functionality [6]. However, existing studies focus on utilizing specific wireless communication devices, such as software-defined radio (SDR) that is primarily used for wireless communication systems for professionals [7], or a particular WiFi adapter such as the Intel 5300 network interface card (NIC) [8], [9] that has been obsolete in the latest hardware. This motivates our work on utilizing nonspecific devices for HAR.

In principle, many WiFi-based HAR methods involve extracting and processing channel state information (CSI) specified in the IEEE 802.11 protocol [10], [11]. In particular, recent progress in the extraction of channel state information from the WiFi packets using orthogonal frequency-division multiplexing (OFDM), contributes a way to elaborate the implicit patterns caused by the sensing target. The CSI represents how the WiFi signal propagates which contains the fine-grained information of both the reflected and the direct paths (multiple paths); in particular, the reflected signal by the sensing target such as humans. The CSI data are further processed to extract the features caused by the sensing target. For example, the regular respiration behavior can change the multipath propagation of the WiFi signal, and thus, the CSI exhibits a regular variation corresponding to the respiration. Therefore, by connecting the features of the variation pattern with the respiration, it is possible to identify whether the respiration behavior is normal [10]. However, the CSI data included in the OFDM packets cannot be directly obtained as restricted by the firmware and the operating system. A popular workaround tool for capturing CSI data is the 802.11n CSI tool [12] for some specific WiFi adapters such as the aforementioned Intel 5300 NIC using a laptop or PC. Alternatively, existing works have adopted a router for data collection while still limited to specific hardware [13].

In contrast, a recent work, Nexmon [14], enables the potential of using smartphones to collect CSI data in a mobile fashion. In this work, we present a novel HAR system using a smartphone

Manuscript received 25 November 2021; revised 1 March 2022 and 3 May 2022; accepted 24 June 2022. Date of publication 18 July 2022; date of current version 16 January 2023. This work was supported in part by the Natural Science Foundation of China under Grant 62072004 and Grant 61702011, in part by Anhui Provincial Natural Science Foundation under Grant 1908085MF214, and in part by the University Synergy Innovation Program of Anhui Province under Grant GXXT-2019-024. This article was recommended by Associate Editor Vinod A. Prasad. *(Guiping Lin and Weiwei Jiang are co-first authors.) (Corresponding authors: Xin He; Xing Guo.)*

Guiping Lin, Sicong Xu, Yujun Zhu, and Xin He are with the School of Computer and Information, Anhui Normal University, Wuhu 241002, China (e-mail: linguiping@ahnu.edu.cn; sicong.xu@ahnu.edu.cn; zhuyujun@ahnu.edu.cn; xin.he@ahnu.edu.cn).

Weiwei Jiang is with the University of Melbourne, Parkville, VIC 3010, Australia (e-mail: weiwei.jiang@student.unimelb.edu.au).

Xiaobo Zhou is with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: xiaobo.zhou@tju.edu.cn).

Xing Guo is with the School of Computer Science and Technology, Anhui University, Hefei 230601, China (e-mail: guox@ahu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/THMS.2022.3188726.

Digital Object Identifier 10.1109/THMS.2022.3188726

and an off-the-shelf router, which further lowers the loss and is more ubiquitous compared to the previous work. One challenge is that, however, the WiFi signals received by the smartphone are much weaker due to the limited size of the hardware, in particular the antenna. To investigate the impact of this phenomenon, we further study how the performance is affected with different movement scales using smartphones. Specifically, we define 20 types of motions ranging from a small scale to a big scale for activity recognition. Moreover, we compare several classification methods to further study the performance using smartphones, including conventional machine learning methods and deep learning methods. The following main contributions of our work are three-fold.

1) We develop an Android application (APP) to collect WiFi packet data and extract CSI based on Nexmon firmware patch [14], which further lower the cost and is able to ubiquitously deploy compared to the previous work.
2) We design and implement a motion recognition system using smartphones with WiFi signals. Our system enables the flourishing WiFi-sensing applications with greater convenience and lower cost without using modified routers or other specific WiFi adapters, etc.
3) We conduct a thorough user study with 60 participants. Each participant performs 20 different motions in two different experimental settings, resulting in 36 000 data points collected.

The rest of this article is organized as follows. Section II presents the related state-of-the-art works. In Section III, we explain the design of the human activity sensing system. The experimental protocol is detailed in Section IV. We demonstrate the experiment results in Section V, followed by the discussions in Section VI. Finally, Section VII concludes the article.

II. Related Work

We classify the state-of-the-art HAR techniques into two categories: Based on 1) dedicated devices; or 2) commercial off-the-shelf (COTS) devices. Both of them have made great contributions in the field of sensing recognition, but they also have their own advantages and disadvantages. Sensory recognition based on dedicated devices mainly realizes HAR through devices, such as SDR, radar, infrared, dedicated camera, and dedicated sensor devices, while the sensory recognition based on COTS devices mainly relies on nonlaboratory commercial devices such as WiFi routers, smart watches, and smart gloves.

A. Methods Using Dedicated Devices

Existing work using dedicated devices include leveraging dedicated cameras for computer vision [15], radar [16], infrared [17], or other specific sensors, and even higher-cost devices such as universal software radio peripheral [18]. While these methods can achieve decent performance, they have particular limitations such as light conditions, line of sight, cost, and privacy issues, which hinder real-life scenario interactions. For example, the camera-based gesture recognition system [15] used a dedicated hardware system for a single camera to recognize user gestures by capturing movement paths. However, its ethics and privacy issues are broadly concerned [19]. Beyond vision, fast-convergence distributed support vector machine (FDSVM) [20] utilized the 3-D accelerometer that had no such issue. It could recognize 4 or 12 user gestures, with an accuracy of 0.98 or 0.89, respectively. Furthermore, UWave [21] also adopted a 3-D accelerometer to recognize eight user gestures. The authors utilized a template adaption method that could achieve an accuracy of 0.98 or 0.93 without the template adaption. Other than using inertial measurement unit sensors that had to be worn by users, software-defined radio has been widely adopted without requiring wearables. For instance, WiSee [7] used SDR to measure the Doppler effect generated by different gestures. The authors successfully classified nine user gestures with the accuracy of 0.94. Nevertheless, the equipment cost is high (2600 USD for WiSee in 2021). Also, Greg Malysa *et al.* [22] developed a gesture recognition system by using 77-GHz frequency modulated continuous wave radar system. Through measuring the micro-Doppler signals of gestures, the energy distribution in the action space that changes with time is constructed. The authors then used a hidden Markov model for gesture recognition with an accuracy of 0.83. Similarly, the cost of using radar for gesture recognition is infeasible for daily scenarios. In addition to classification, the authors in [16] presented a novel system, Soli. It adopted a millimeter-wave radar-based approach for robust gesture recognition and gesture tracking with submillimeter accuracy, which achieves operational speed in excess of 10 000 frames per second on embedded hardware. Finally, the infrared-based methods are also studied, especially for dim environment. Representative products such as Leap Motion1 [17] and Microsoft Kinect2 were not limited by light conditions and could realize non-line-of-sight (NLOS) sensing. However, the infrared sensing range is limited and expensive additional equipment is needed, making it difficult to apply on a large scale.

B. Methods Using COTS Devices

With the continuous popularity of wireless technologies (such as WiFi, radio frequency identification (RFID), etc.), there are an increasing number of sensory recognition systems based on commercial devices such as routers, RFID devices, and other commercial sensor devices. In principle, the systems are mainly based on RF signals, with great interest in the received signal strength and CSI. One major advantage is that these methods do not require line-of-sight (LOS). For instance, Nandakumar *et al.* [23] proposed a gesture recognition system using Intel 5300 NIC, which could recognize four user gestures with an accuracy of 0.91 in LOS and 0.89 in NLOS. Also, He *et al.* [24] proposed WiG, using COTS WiFi cards, which could classify four gestures with an accuracy of 0.92 in LOS and 0.88 in NLOS. Furthermore, WiFinger [25] extracted the fixed pattern of gesture signal with principal component analysis as the features for gesture recognition, which achieved an accuracy of 0.93. Moreover, Shang *et al.* [13] proposed WiSign, a CSI-based sign language recognition system. Unlike other systems, WiSign used a router and two receivers (two routers) to recognize gestures, which improved the recognition performance of the system to American Sign

Language (ASL). Besides, WiGeR [26] used the CSI amplitude fluctuations caused by gestures to recognize, with an accuracy of 0.92. On top of that, WiCatch [27] combined data from different antennas to eliminate interference, thus achieved an overall recognition accuracy of 0.95, and an accuracy of 0.94 for gesture recognition of hands. In addition, there are also application-specific studies. For instance, SignFi [11] collected CSI in laboratory and home environment, and recognized 276 sign languages with an accuracy of 0.98 or higher. Similarly, WiGest [28] used the changes in WiFi signal strength caused by gestures to recognize multiple gestures around the device and used them in the control of the multimedia player application, achieved 0.875 and 0.96 accuracy in the case of one access point and three access points, respectively. However, as there were some significant changes to received signal strength indicator (RSSI) values that resulted in persistent error detection, WiGest lacked the security to operate through walls. In addition to WiFi-based systems, Wang *et al.* [29] introduced RF-IDRaw, which was designed with a commercial RFID reader and allowed users to interact with the device through the gesture with a virtual touch screen. Above all, a recent study, WiPhone [10], showed a smartphone-based respiration monitoring system and proposed an ambient-reflected signal model in the NLOS setting. It overcame the limitations of existing work in LOS blocking scenarios, and could operate with normal WiFi traffic without interruption in a mobile fashion without wearables. We are particularly interested in expanding their work into a more general HAR scenario with common gestures.

III. System Design

In this section, we present how we design the HAR system using a COTS smartphone and an unmodified COTS WiFi router.

A. Design Principles and Goals

We design a system to investigate the possibility of recognizing different scales of human activities (big motions, medium motions, and small motions or gestures) using a smartphone. In practice, the WiFi signals propagate via multiple paths in a typical indoor scenario. Such multipath effect produces different path losses in the frequency domain, and can be described by the fine-grained CSI data. As human activities may affect the propagation of WiFi signals, the CSI data change accordingly, thus can be used for activity recognition. With this phenomenon, we aim to further examine how the movement scale of human activities affect the generation of a unique pattern on the WiFi propagation, especially, of the smartphone. Then, we can extract the patterns associated with different motions as the classification features. Moreover, the WiFi propagation varies in different scenarios, which should be clarified in the smartphone case. Thus, we conduct comprehensive experiments including 20 different types of activities in two different offices with completely different layouts. For classifying the motions, we consider three commonly used machine learning methods, including: 1) convolutional neural network (CNN); 2) decision tree (DTree); and 3) long short-term memory (LSTM) to investigate which method fits well in such a scenario.

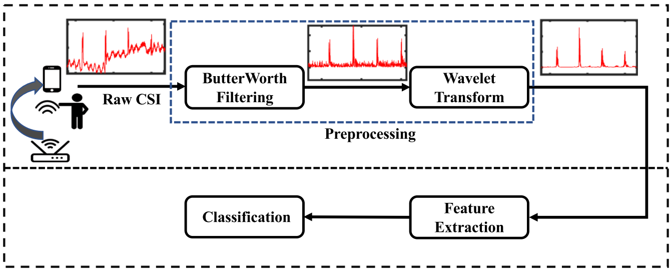


Fig. 1. Overview of the motion recognition system using the WiFi signal from a smartphone.

B. System Flow

1) Overview of System: As shown in Fig. 1, the process consists of the following four stages:

a) data collection;

b) preprocessing;

c) feature extraction;

d) classification.

The data are collected by an APP using a smartphone (Google Nexus 5), and processed using a desktop computer by our customized software developed with Python to recognize the motions.

2) Data Collection: A router and a smartphone form a pair of the transmitter and the receiver. The router operates at a central frequency of 5.805 GHz with a bandwidth of 80 MHz as an access point. The router broadcasts a beacon frame every 100 ms, as specified by the IEEE 802.11 protocols. Accordingly, we develop an APP based on Nexmon [14] firmware patch to collect the WiFi data at a sampling rate of 10 Hz. In total, we collect 140 h of WiFi data (equivalent to 5.8 d) for HAR. For each collected packet, we select 62 subcarriers (Nos. 131–192) out of 256 subcarriers, their corresponding frequency is 5746.25 MHz$\sim$ 5764.0625 MHz. The channel state information of those subcarriers are much more sensitive to human motion as shown in Fig. 2. The possible reason is that we use the amplitude of the CSI data to recognize the human activities, and the 802.11ac protocol with a bandwidth of 80 MHz performs a so-called spectrum mask technique to result in a higher spectrum around the center frequency of the channel [30]. As shown in Fig. 2, it contains the CSI amplitude values of 256 subcarriers, each of which corresponds to a different color bar. The x-axis indicates the index (ranged from 0 to 255) of each subcarrier, and the y-axis represents the CSI amplitude value. Each CSI depicts the amplitude and phase of a subcarrier

$$H\left(f_k\right) = \left\| H\left(f_k\right) \right\| e^{j \sin(\angle H)} \tag{1}$$

where $H(f_k)$ is the CSI at the subcarrier with central frequency of f_k and $\angle H$ denotes its phase. Indeed, the CSI of the subcarriers between 131 and 192 constantly vary, indicating that these subcarriers respond to the environmental change caused by the human activities, while other relatively stable subcarriers are less sensitive to motions. Therefore, we can select these subcarriers due to their high sensitivity to the motions.

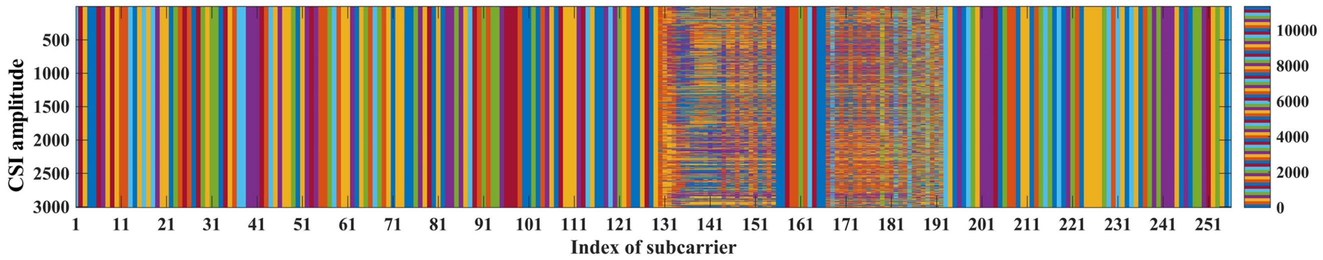


Fig. 2. CSI amplitude of all subcarriers.

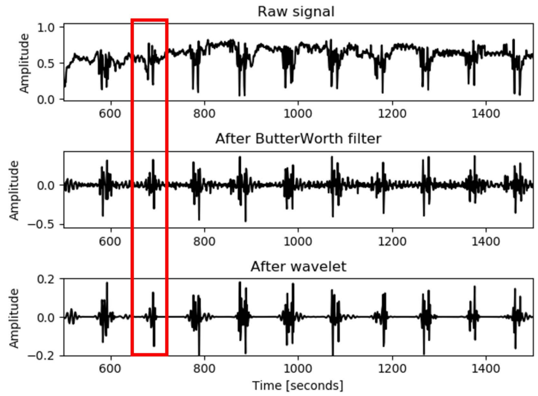
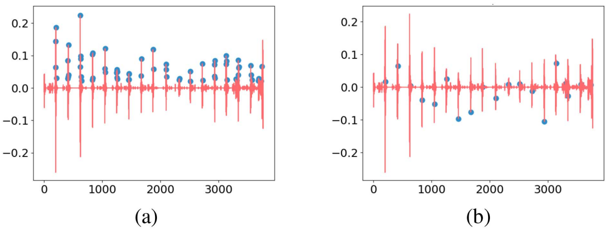


Fig. 3. CSI data waveform of "left arm flat lift" w/o preprocessing.



(a)

(b)

Fig. 4. Illustration of selecting the central peak values in the CSIdata sequence. (a) Local peak. (b) Central peak.

3) Data Preprocessing: The collected data usually contain noise which may affect the classification performance. As shown in Fig. 3, the raw CSI data have a discernible pattern corresponding to a human motion. However, it is a challenging task to determine the starting and the ending points of the pattern. Thus, it brings difficulty on the feature extraction for machine learning. Hence, we adopt the following common signal processing techniques to process the raw CSI data.

a) *Butterworth filtering:* Since the frequency of data variations caused by human activities is quite different from that caused by the external noise, we develop a Butterworth filter to denoise the data by removing the frequency components associated with the noise. The Butterworth filter is a type of filter with the frequency response in the passband as flat as possible (*i.e.*, without ripples). The transfer function of a typical low-pass Butterworth filter is

$$H(j\omega) = \frac{1}{1 + (j\omega/\omega_c)^n} \qquad (2)$$

where n denotes the order of filter, and ω_c is the cutoff frequency in radians per second. As n approaches infinity, the gain becomes a rectangle function and frequencies below ω_c are passed, while frequencies above ω_c are suppressed. However, the smaller the value of n, the less sharp the cutoff. It is worth noting that we can easily implement a low-pass Butterworth filter from (2).

In our system, the CSI data are smoothed by the Butterworth filter with an order $n = 10$, with the cutoff frequency at 2 Hz, in a high-pass mode. As shown in Fig. 3, the CSI data now have clearly associated patterns. However, both ends of each pattern are still slightly fluctuated. Therefore, we add a wavelet decomposition to further remove such fluctuations in the data.

b) *Wavelet Decomposition:* Wavelet decomposition is a method to analyze and process the signal in the time domain. Due to the fact that the wavelet coefficient generated by the signal contains the important information of the signal itself, it is usually larger after wavelet decomposition, than that of the noise. Hence, we choose an appropriate threshold to suppress the portion below the threshold value, *i.e.*, the noise is removed. We perform a level 5 wavelet decomposition of the signal using the wavelet, and then we reconstruct the approximation coefficients at level 5 from the wavelet decomposition structure and the detail coefficients at level 2. After filtering with the Butterworth filter, the wavelet denoising can further make the variation characteristics caused by the human motions more obvious, which facilitates the feature extraction process.

4) Feature Extraction: After filtering and wavelet denoising the original CSI data, we can select the subcarrier sequence with the most obvious characteristics as the segmentation standard, and obtain a subcarrier sequence $\varphi(t)$ with obvious and smooth motion. As shown in Fig. 3, we can clearly observe the motion intervals through the figure. Then, we find the local peak values in the waveform of CSI sequence [as shown in Fig. 4(a), the blue dots represent the local peak values], and we combine multiple local peak values of the same motion to get the central peak values. Finally, we use the combined central peak value as the index in the CSI sequence. The indexed point is the midpoint of the motion segment [as shown in Fig. 4(b), marked the central peak of each motion]. With this midpoint, the data fragment is intercepted from the original data with a certain length as the feature vector. The motion label is added according to the

Algorithm 1: Motion Feature Extraction of Each Subcarrier.

```
Input: CSI amplitude $\mathbf{r}_{(1\times m)}$;
Repeated times of a motion $M = 30$;
Threshold of the amplitude $\delta$;
Specific length $L$ corresponds to the duration of a motion;
Output: Motion feature matrix $\mathbf{F}_{(L\times M)}$;
for $j = 2$ to $m - 1$ do
    if $\mathbf{r}[j-1] - \mathbf{r}[j] < 0$ && $\mathbf{r}[j+1] - \mathbf{r}[j] < 0$ &&
    $\mathbf{r}[j] < \delta$ then
        add the index $j$ to the local peak set $\mathcal{P}_{index}$;
    end if
end for
$N =$ the length of $\mathcal{P}_{index}$;
for $i = 2$ to $N$ do
    if $\mathcal{P}_{index}(i) - \mathcal{P}_{index}(i-1) < L/2$ then
        the midpoint of the detected motion interval
        $mid = (\mathcal{P}_{index}(i) + \mathcal{P}_{index}(i-1))/2$;
        add the index $mid$ to the central peak set $\mathcal{M}_{index}$;
    end if
end for
for $i = 1$ to $M$ do
    $\mathbf{F}[:, i] = \mathbf{r}[\mathcal{M}_{index}(i) - L/2 : \mathcal{M}_{index}(i) + L/2]$;
end for
```

category of the motion. We also summarize the aforementioned algorithm in Algorithm 1. Algorithm 1 takes the CSI amplitude data of the sensitive subcarriers as the input. By observing the waveform of the CSI amplitudes of motions, we set a filtering threshold[1] δ and screen out all values above the threshold as the local peak values of all motions in the corresponding duration. However, the selected data of some motions may have multiple peaks, thus, we make an average of the local peaks and set the averaged value as the central peak of each motion. Finally, we extract the filtered data (above the threshold) with a specific length[2] L around the central peak as the feature pattern for further processing.

5) Machine Learning-Based Classification: After segmentation and labeling, we then train classification models for activity recognition. For crossvalidation, we adopt K-fold crossvalidation method to evaluate the performance of the classifiers. The main idea of K-fold crossvalidation is to divide the initial samples into K subsamples, of which a single subsample is used as the test data set for crossvalidation, and another $K - 1$ samples are used for training. Crossvalidation is repeated K times, thus, each subsample will be tested once. Finally, the performance is calculated by averaging the results of all K tests. The advantage of such method is that all subsamples are used for either training or crossvalidation each time, which is more representative of the total dataset. When training the model, we constantly adjust the value of parameter K to achieve a higher learning accuracy. Finally, we choose $K = 10$ as the parameter value for crossvalidation.

For the machine learning models, we choose the following three methods as they have been widely adopted in literature [31]: 1) CNN [32]; DTree [33]; and LSTM memory [34]. For the CNN model, the extracted features are fed as input. The CNN model consists of three convolutional blocks, having three Conv2D layers with 64 128 256 filters separately, and a MaxPooling2D layer of size 2 × 2 per block. It is followed by a dropout layer with $p = 0.5$, flatten layer, and a dense layer of size 256. Finally, the output dense layer is the size of the number of output classes. For the LSTM model, it consists of two hidden layers with 128 neurons, a dense layer of size 256 and a linear output layer with softmax activation to predict the human activities. For the DTree model, the feature selection method selected by the decision tree is "gini," which searches the local optimal division point in the randomly selected partial division points. The depth of the tree is 20; the minimum number of samples required for node subdivision is 2, and the minimum number of samples required for leaf nodes is 5, which can prevent overfitting.

IV. Experiment Protocol

A. Prototype

An off-the-shelf *TPLink_5G_BFA1* with no modification was used as the router, which used 5-GHz bands to transmit signals. The transmit power at the router was 500 mW. We selected *Google Nexus 5* with operating system, Android Stock 6 as the smartphone to collect data, the received power at the mobile units is around -52 dBm, and the SNR is about 18 dB by setting the noise power being -70 dBm typically. We developed an APP, namely motion recognition, in the smartphone using Android studio. Our APP is able to collect data and show the dynamic variation on the amplitude of the WiFi packets. The user interface of the APP is shown in Fig. 5(a), in which three photos represent the interface, the collection, and visual demonstration of the CSI analysis, respectively. In particular, the left one is the user interface of the APP, the button "CAPTURING" represents the collection of WiFi signals in the environment, while the button "CSI ANALYZING" represents the processing of collected WiFi data packets. The specific process is to extract the CSI data first and plot the dynamic waveform of CSI, which is shown in the middle of Fig. 5(a). At last, a plot of CSI data of five subcarriers is illustrated in the right of Fig. 5(a). By observing the dynamic waveform of CSI, we find that the value of CSI is constantly changing over time. The CSI data are also affected by human activities by the distinct propagation which results in the data change in subcarriers.

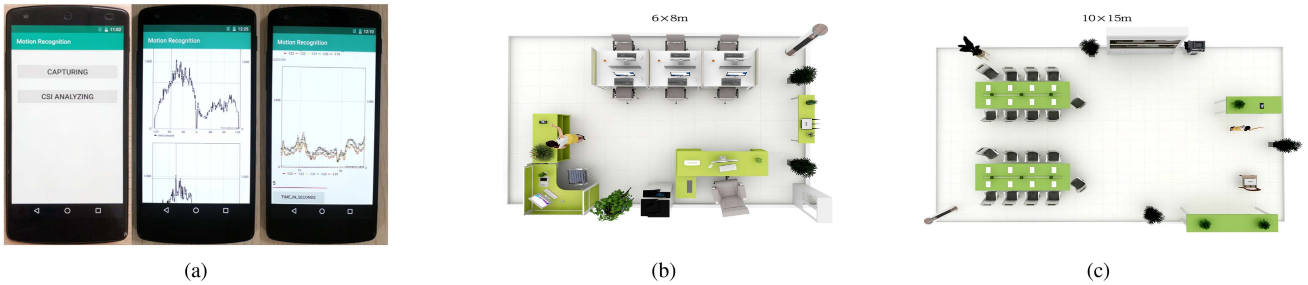
The collected data formed a . *pcap* file in the smartphone which was transmitted to the computer via USB connection. All the data were processed by a Python script with tensorflow version 2.4.0 [35] and scikit-learn [36] packages.

B. Experiments Setup

As illustrated in Fig. 6(a) and (b), each participant stood or sat between the router and the smartphone with different positions or distances. The distance from the router to the smartphone

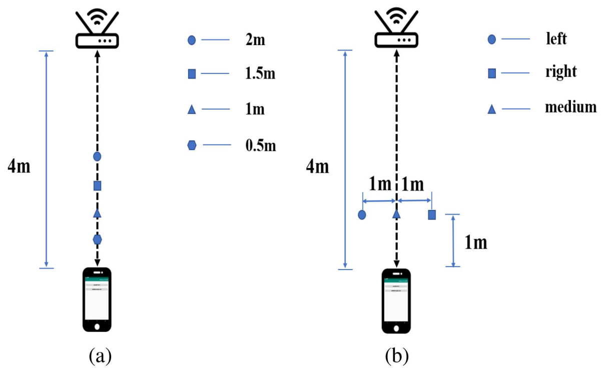
[1] We set the threshold to be the averaged value of the maximum and the minimum values of the amplitudes of the CSI.

[2] Empirically, we set $L = 40$ which makes the length of the extract features being 40 in the experiments, since each motion roughly lasts 3 s which corresponds to 30 data points at the sampling rate 10 Hz, and the length 40 (relaxed a little bit) results in a better accuracy of the recognition.



(a) (b) (c)

Fig. 5. Developed APP and the environment of labs where we conducted experiments. (a) The developed APP for human activity recognition and its user interfaces. (b) Laboratory 1. (c) Laboratory 2.



(a) (b)

Fig. 6. Settings of different distances and positions.(a) Four different distances conducted in our experiments. (b) Three different positions conducted in our experiments.

was fixed at 4 m, and the distance from the participant to the smartphone varied from 0.5 to 2 m with a step of 0.5 m. In addition, we also tested the scenario when the participants put the smartphone in their pocket as follows.

1) Experimental Environment: Furthermore, we conducted our experiments in two different rooms. Both rooms are regular laboratories located in a four-floor building. The first three floors are occupied with offices. The fourth floor has three labs which are only used if there is a scheduled course. Lab 1 (at fourth floor) is with 9 × 5 m in size; there were ten tables and desktops placed. We considered Lab 1 as a complex environment, as shown in Fig. 5(b). Lab 2 (at second floor) was used to store devices with a size 10 × 15 m, of which the layout is depicted in Fig. 5(c). However, Lab 2 was in the proximity to the department offices, thus, it had more interference due to the higher WiFi traffic, compared to that of Lab 1.

2) Type of Motions: We tested 20 different types of motions ranging from a small scale to a big scale. As shown in Fig. 7, there are nine types of big motions, four types of medium motions, and seven types of small motions. For the big motions, the participant moved one of their entire arm or leg, or both, while the participant only moved their forearm in medium motions, and their fingers in small scale motions, respectively.

3) Participants: In total, we recruited 60 participants for our study (40 males, 20 females). The participant received a gift card at the end of experiment. The ages range from 22 to 27, with heights ranging from 160 cm to 188 cm, and 62 kg is the mean weight. Furthermore, we divided the 60 participants into 4 groups, with 20 people in Group 1 and 2, 10 people in Group 3 and 4. The experiments for Group 2 were conducted in Lab 2, while other experiments were conducted in Lab 1. During one session, each participant performed multiple motions. Each motion was repeated 30 times. We conducted multiple sessions at different positions.

The experimental conditions for each group are shown in Table I. It is worth noting that the distance for Group 4 means the gap between the participant and the router, as the participants were required to take the phone. For other groups, the distance refers to the gap between the participant and the smartphone.

V. Evaluations

A. Performance Metrics

We used both the confusion matrix and the accuracy to evaluate the system performance. In the confusion matrix, each row represents the ground truth of the human activities, and each column indicates the predicted classification result of our system. The accuracy means the proportion of correct predictions in the total predictions.

B. Results

1) Overall Performance: For all the tested activities, we used three learning algorithms (CNN, DTree, and LSTM) to obtain the learning accuracy. It has been proved by practice that the accuracy obtained by CNN algorithm is higher (CNN: 0.9725, DTree: 0.9425, LSTM: 0.8421), so we adopted the CNN learning method to obtain the overall accuracy, where the confusion matrix is shown in Fig. 8. The mean accuracy of the tested 20 types of motions is 0.9725. Among these activities, the gesture "OK" achieves the best accuracy of 1.0, while the motion "jump" is the worst with the accuracy of 0.92. In general, the classification model is robust across all scales of the activities, while it may misclassify motions within the same scale. In particular, 3% of "cupping hands" are predicted as "hand heart," and 3% vice versa. This is caused by the relatively high similarity of these two gestures. In order to measure the accuracy of the model, we calculated the F1 score of the 10 crossvalidation. As shown in Fig. 9(b), in the 10 crossvalidation, the values of F1 score are all very high, indicating that our model is reasonable. As we can see in Fig. 9(a), the size of the epoch has an effect on

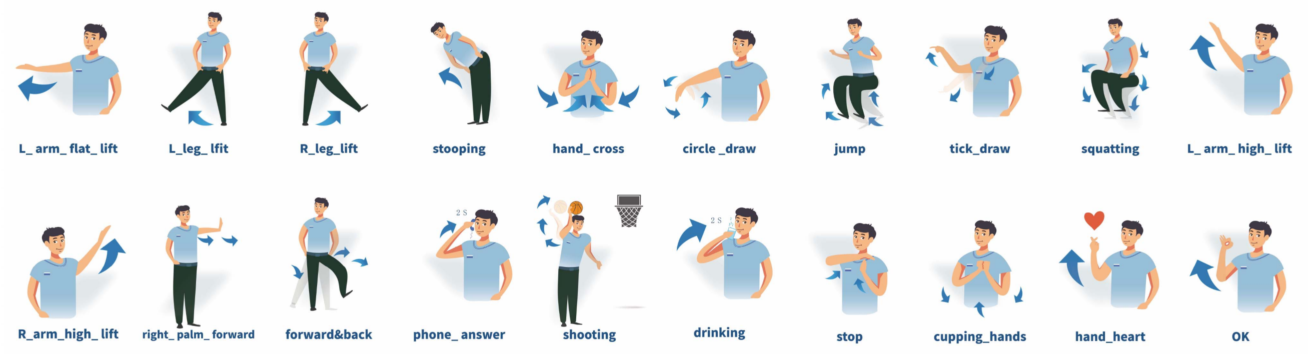


Fig. 7. Tested 20 different types of human activities, ranging from big scales to small scales.

TABLE I
CONFIGURATION OF PARTICIPANT GROUPS DURING THE EXPERIMENTS

Group ID	Number of volunteer	Motions	Take phone	Position	Distance (m)	Scenarios	Repeat times
1	20	20	No	middle	1	Lab 1	30
2	20	20	No	middle	1	Lab 2	30
3	10	5	No	middle left, right	0.5, 1, 1.5, 2 1	Lab 1	20
4	10	5	Yes	middle left, right	3.5, 3, 2.5, 2 3	Lab 1	20

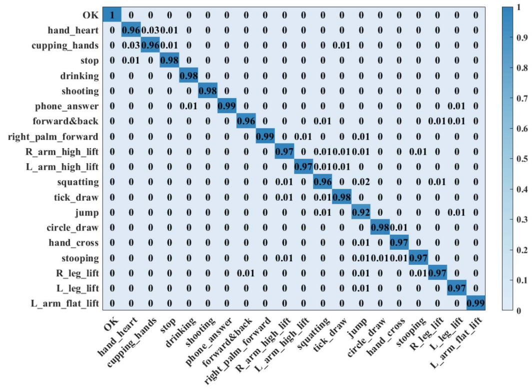
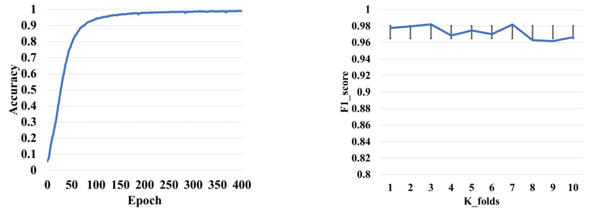


Fig. 8. Accuracy confusion matrix for all motions in Group 1 (CNN, 20 motions, total accuracy: 0.9725).



(a) (b)

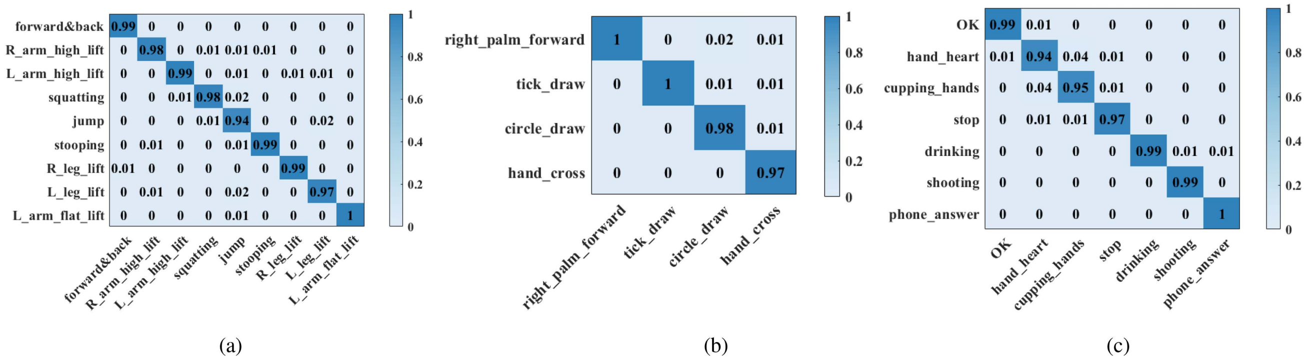
Fig. 9. Influence of epoch and F1 scores in K folds. (a) Accuracy of different epoch. (b) F1 scores in K folds.

learning accuracy, as the epoch grows, it becomes more accurate, eventually flattening out between $350 - 400$. We adopt epoch $= 400$ as the learning parameter.

2) Impact of Motion Scales: We first investigated how the motion scales impact on the performance, including seven small motions, four medium motions, and nine big motions. The accuracy of each motion is shown in Fig. 10.

Big motions: The mean accuracy of big motions is 0.981. Among the nine big motions, the "left arm flat lift" motion achieves the highest accuracy of 1.0, while "jump" achieves the lowest accuracy. The main reason is that the motion of "jump" can cause significant changes to the environment, such as floor vibrations, or to the participant, such as body tilt. Also, the landing position may be different each time, which can lead to more fluctuations of the waveform that may confuse the classifiers. In contrast, the "forward and backward," "stooping," "squatting," "left arm high lift," and "right arm high lift" are more stable and the accuracy tends to be slightly higher. In addition, we observe the "right leg lift" achieved 0.20 higher accuracy than "left leg lift." This is caused by the dissimilarity of human's left hemisphere and right hemisphere (*i.e.*, the left part and the right part of the brain). Existing studies show that human's right part of the body is mainly controlled by the left hemisphere, which is relatively more sophisticated than the right hemisphere [37]. According to Dr. Sperry, the right hemisphere was responsible for spatial memory, intuition, emotion, physical coordination, visual perception, and so on. When we lift our right leg and the left leg is on the ground, the right hemisphere is better able to control the left side of the body for balance. Consequently, the stability of lifting the right leg is better than that of lifting the left leg. Hence, the human body is easily swinging when standing with the right leg while lifting the left leg, which results in much more noise in the collected data samples.

Medium motions: The overall accuracy for four medium motions is 0.9875, as shown in Fig. 10(b). The accuracy of these four motions is similar, topped by "right palm forward" and "draw tick," followed by "draw circle," as they need only one arm to complete the motion; while "hand cross" needs to take



(a)

(b)

(c)

Fig. 10. Accuracy confusion matrix for different scales of motions. (a) Accuracy confusion matrix for big motions in Group 1 (CNN, 9 motions, mean accuracy: 0:981.) (b) Accuracy confusion matrix for medium motions in Group 1 (CNN, 4 motions, mean accuracy: 0:9875.) (c) Accuracy confusion matrix for small motions in Group 1 (CNN, 7 motions, mean accuracy: 0:9757.)

the motion with the two arms. Therefore, it causes movements in other parts of the body, then there will be more interference on the signal, as opposed to "draw circle." Although it is a similar type of motion to "draw tick," the accuracy of "draw tick" is slightly higher because it has only half the range of motion of "draw circle," causing fewer fluctuations in the signal.

Small motions: The mean accuracy for all seven small motions is 0.9757, as shown in Fig. 10(c). The highest accuracy within all the seven motions is "phone answer," with the value of 1.0. When answering the phone, our hands will rest on our ears for 1 − 2 seconds, which is more stable than other motions. While "hand heart" achieves the lowest accuracy, followed by "cupping hands" and "stop," when these motions act, most of them are in the front of chest, and part of the motions will be blocked by the human body, so the waveform changes caused by the signal are poor, and the recognition accuracy is slightly lower. Other motions including "OK," "shooting," and "drinking," they have the same accuracy: 0.99; as they both have the right arm going up over the right shoulder, the resulting waveform changes are more obvious. In general, as shown in Fig. 10, the mean accuracies for all three scales are close, with big motions worse than the medium ones. This phenomenon is mainly due to the fact that the smaller motions have less impact on the experimental setting itself (*e.g.*, less movement generated by the rest parts of the human body), resulting in less interference noise to the CSI data. When the motion is smaller, the body will block part of the motion due to the problem of our experimental setting, resulting in less obvious waveform changes. Therefore, the recognition accuracy is low combined with the influence of the environment.

3) Impact of Distances: In addition to the scale of the motion, we investigated how the distance impacts on the performance. The experimental settings are illustrated in Fig. 6(a). In this experiment, the distance between the participant and the phone varied. We randomly selected 10 motions out of 20 motions, including three big motions (squatting, leg left lift, and forward and backward), two medium motions (hand cross and draw tick), and five small motions (shooting, drinking, stop, cupping hands, and hand heart). Each participant performed the same ten motions repeated 20 times at each distance. We also compared the performance difference between a conventional machine learning method, DTree, and a deep learning method, CNN. From the results of the experiment, CNN is more robust in such a scenario compared to DTree, facilitated by higher complexity and capacity of the model itself. For DTree, the mean accuracy of all motions over the four distance was below 0.95, while for CNN, the mean accuracies of all motions over the four distances were all above 0.95, topped by the condition with distance at 1.5 m. In particular to the motions, the accuracies of the motions "cupping hands" and "hand cross" dropped below 0.9, while the others did not. In addition, the motion "squatting" had the best stability that could maintain a high recognition accuracy at all four distances, seconded by the motion "draw tick." In general, the recognition of all motions is stable and efficient.

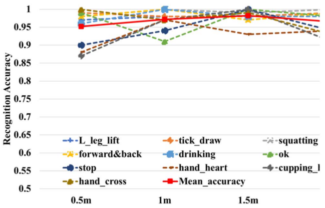


Fig. 11. Accuracy with respect to different distances using CNN.

In summary, as we can observe in Fig. 11, the recognition accuracy of each motion is different at four distances. However, in general, the recognition results of the motions at different distances are relatively reliable and stable. Thus, our system using a smartphone is relatively robust at different positions.

4) Impact of Positions: Next, we evaluated the performance at different positions. Specifically, we aimed to examine whether the system can be affected by the position shift in the LOS scenarios, *i.e.*, whether the participant is in the middle of the transmitter-receiver line, as shown in Fig. 6(b). The results are shown in Fig. 12. In general, all motions are accurately classified in all three positions using CNN. The mean accuracy of all motions in the three positions (solid red line) is high (above 0.97), topped by the position in the right. In particular, only the "ok" and "stop" motions in the position of middle did not achieve an accuracy above 0.95. Other motions are recognized

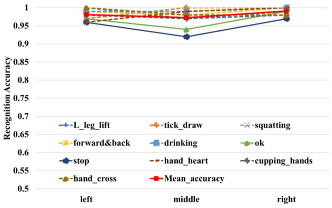


Fig. 12. Accuracy with respect to different positions using CNN.

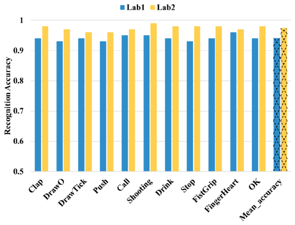


Fig. 13. Performance comparison of the obtained accuracy in different labs.

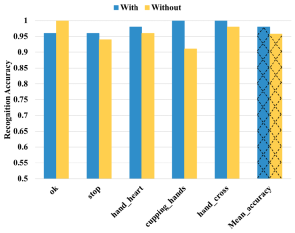


Fig. 14. Accuracy of the case whether the participant takes the smartphone or not.

with stable performance across all three positions. While for DTree, the mean accuracy of all motions is around 0.90, which is lower than the result of using CNN. In general, the results of all motions in the three positions are stable with high accuracy when using CNN.

5) Impact of Environments: Participants in the Group 2 were arranged in the Lab 2 with a different environment. Other settings were identical to that of for Group 1. Compared with Lab 2, the environment of Lab 1 is more complex. The accuracy comparison of Lab 1 and Lab 2 is shown in Fig. 13. In general, the accuracy of all motions in Lab 2 is better than that in Lab 1. This is because the smartphone is close to the participant, the signal collected is more influenced by the environment than by other interference due to the other WiFi traffic. Both of them achieved mean accuracy of around 0.95, showing a limited effect on the accuracy.

6) Impact of Whether Taking Phone: In order to compare the effect of people carrying their smartphone versus not carrying them, our participants performed the same motions with and without smartphone. The results are shown in Fig. 14, the accuracy of motion recognition is better when carrying smartphone, the average accuracy in both is above 0.95.

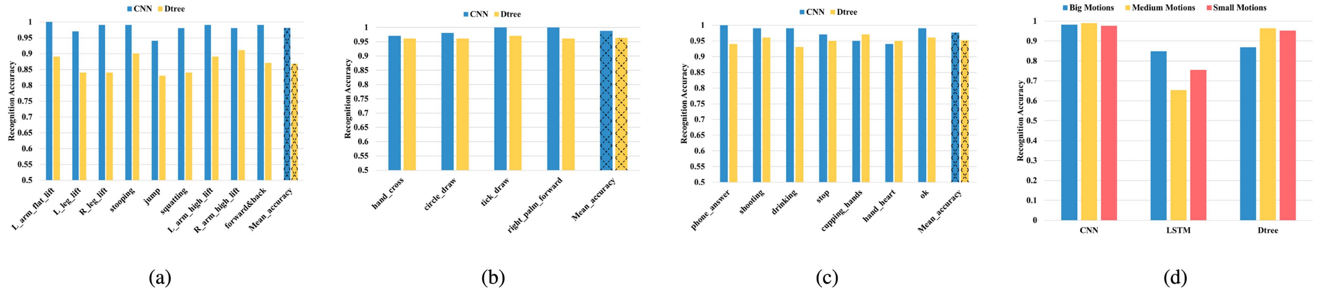
7) Machine Learning Models: Then, we offered a further comparison among different machine learning methods in general. We considered three machine learning methods: 1) CNN; 2) DTree; and 3) LSTM. We used the data collected in Group 1. The classification results are shown in Fig. 15. It can be observed that CNN algorithm is more suitable for all-sized motions. For HAR, many researchers have used various machine learning methods. For example, Sigg [38] used software-defined radios to transmit radio frequency signals, and recognized human motions according to the changes in the received signal strength. It adopted a series of statistical features, and compared various classification algorithms such as K-nearest neighbors (KNN), Bayesian, and DTree. The average recognition rate is above 85%. Xiao [39] established a CNN-based activity segmentation framework model, DeepSeg. For activity recognition based on WiFi signals, the model had good performance in both fine-grained and coarse-grained activity scenarios. M. Sulaiman [40] performed activity recognition on CSI data containing seven kinds of activities (sleeping, falling, picking up, running, sitting, standing, and walking), and the established RNN method achieved an average accuracy of 94.68%. Then, the RNN was replaced with a CNN-based model, which improved the accuracy to 95.12%. This shows the feasibility of using CSI for activity recognition in a CNN-based architecture. WiFinger [25] used the KNN algorithm to recognize nine ASL on commercial WiFi devices through a series of signal processing methods such as filtering and denoising, and realized finger-level gesture recognition with an accuracy of 90.4%. Zhang [41] proposed a novel deep neural network (DNN) model—Dense-LSTM, which was optimized for WiFi CSI data. Compared with the traditional DNN model, the Dense-LSTM model improved the accuracy by 21.2%, and achieved a stable recognition accuracy rate of about 90.0% in small data scenarios.

8) Impact of Sampling Rates: We further consider the impact on the recognition performance using the data sampling rates. We observe the following five sampling rates: 10 Hz, 5 Hz, 2.5 Hz, 1.25 Hz, and 1 Hz. We randomly select five kinds of data from the big motions in Group 1, and extract the features at different sampling rates by downsampling. The classification results are illustrated in Fig. 16(a), where the accuracy of motion recognition decreases with the reduction of sampling rates, as expected. However, it is found that the accuracy at 5 Hz is comparable with that at 10 Hz, i.e., the system complexity can be further reduced using a halved size of data.

9) Channel Coherence Time: Finally, we consider the influence of the channel coherence time. According to [42], the coherence time τ is calculated by

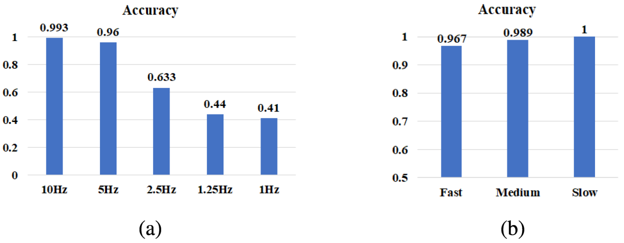
$$\tau = \frac{9}{16\pi(v/\lambda)} \tag{3}$$

with v being the maximum velocity of the object measured in m/s, and λ being the carrier wavelength. Table II summarizes the channel coherence time for three typical scenarios at the center frequency 5.805 GHz. In fact, the coherence times are shorter than the duration of our performed activities, i.e., the human motion could affect the CSI at least in one coherence time. We can then utilize the patterns in the CSI data affected by the human motions to achieve recognition, which is the main goal of this work.



(a) (b) (c) (d)

Fig. 15. Accuracy of adopting different machine learning algorithms for different scales of motions. (a) Comparison of different learning algorithms for big motions. (b) Comparison of different learning algorithms for medium motions. (c) Comparison of different learning algorithms for small motions. (d) Mean accuracy of three different scales of motions by adopting different learning algorithms.



(a) (b)

Fig. 16. Influence of sampling rates and speed. (a) Accuracy of different sampling rates. (b) Accuracy of different speed.

TABLE II
Wireless Coherence Times At 5-GHz Band for Different Motion Speed

Speed	Coherence time at 5.805 GHz
Almost stationary (0.25 m/s)	37ms
Walking (1.5 m/s)	6.2ms
Running (3.5 m/s)	2.6ms

Furthermore, we perform an experiment by doing motions at different speed. We defined the speed as fast, medium and slow, which corresponds to one motion- per second, per 3 second, and per 5 second, respectively. We follow the experimental settings, and carry out experiments in Lab 1. At each speed, three selected motions: L_arm_high_lift, R_arm_high_lift, and Stooping are performed 30 times. The recognition results are shown in Fig. 16(b), where the accuracy of the recognition increases if the motion speed is slow. The finding here is consistent with the results of different sampling rates. In general, if the motion lasts longer or has more data points, the accuracy is higher. Therefore, the motion should cover the channel coherence time and make more impact on the CSI.

VI. Discussion

A. Applications

Compared to existing works, our system takes advantage of COTS devices that are nonlaboratory. It enables HAR in a more ubiquitous and mobile fashion, which greatly enlarges the application niche toward daily use. In particular, our system demonstrated a superior advantage in the recognition of small motions, *i.e.*, gestures. Consequently, it can be adopted in various applications that have been shown useful in many scenarios in particular HCIs, including gesture-based control systems [43], sign language recognition [44], etc.

Furthermore, our system can enhance the interaction modalities in daily life. For example, our system can be integrated into a smart speaker as a compensation for user inputs to voice commands that are yet-maturing, despite the great advancements recent year on natural language processing. Also, the emerging smart projectors can be enhanced by our system. For instance, the speaker can use gesture or posture to control the slides while delivering the speech. It can also be used for motion sensing games like Nintendo Switch [45] without requiring wearables.

B. Limitations

We notice several limitations of our work. First of all, the data processing is offline. For practical use, an online system is required. However, as we successfully show the potential of using smartphones to recognize human activities, it is not difficult to migrate our software to a live system. In particular, thanks to the significant progress on lightweight machine learning models [46]–[48], we can easily deploy our recognition software to entirely perform *inference* at smartphones, then to train the model at a local computer.

Second, in our study, we only tested 20 types of motions at two locations, and generated 36 000 data samples. Although the data size is large enough for our study, it may not be sufficient for a real-life product. In particular, in practice, more types of motions should be considered, with larger training data collected from more participants, and at more scenarios. Moreover, in this work, we did not consider the case where WiFi signal is blocked by some obstacles, such as NLOS scenarios.

Besides, if there are multiple participants in one room, we currently cannot detect different kinds of motions simultaneously. However, it is possible to use transfer learning technique to solve this problem. It is worthwhile to mention that we can gain benefits by using mobile phones to collect the data for simultaneous detection. Since each participant could take a mobile phone, we can then collect the CSI data and perform the sensing tasks for future study.

In addition, this work performs the training process of the machine learning models by using the data from each lab or room and verifies the performance using the data collected from the same room. However, it is an important direction that using the trained machine models from a specific room and to make validation from the dataset from other locations. In the future study, we will solve this problem by using transfer learning.

Finally, for the classification using machine learning models, we did not tune the parameters to optimize the performance. The performance can be further improved with a thorough tuning process, as well as other optimization methods such as more feature engineering. Furthermore, we only selected three models to evaluate the performance. Although our system already achieved high performance, there might be other machine learning methods that can outperform the models we used, as the artificial intelligence field advances.

VII. Conclusion

In this article, we developed a HAR system using a COTS smartphone and a COTS WiFi router. We evaluated our system with 20 different motions in two laboratories, and recruited 60 participants in total for the experiments. Overall, our system achieved 0.9725 mean accuracy. We further show that our system is robust in different positions and environments. Our system enhances the already flourishing applications using HAR, which enables a variety of applications in daily scenarios, including gesture control, sign language recognition, etc.

References

[1] J. Wan, M. J. O'Grady, and G. M. O'Hare, "Dynamic sensor event segmentation for real-time activity recognition in a smart home context," *Pers. Ubiquitous Comput.*, vol. 19, no. 2, pp. 287–301, Feb. 2015. [Online]. Available: https://doi.org/10.1007/s00779-014-0824-x

[2] X. Hong and C. D. Nugent, "Segmenting sensor data for activity monitoring in smart environments," *Pers. Ubiquitous Comput.*, vol. 17, no. 3, pp. 545–559, Mar. 2013. [Online]. Available: https://doi.org/10.1007/s00779-012-0507-4

[3] J. Tewell, D. O'Sullivan, N. Maiden, J. Lockerbie, and S. Stumpf, "Monitoring meaningful activities using small low-cost devices in a smart home," *Pers. Ubiquitous Comput.*, vol. 23, no. 2, pp. 339–357, Apr. 2019. [Online]. Available: https://doi.org/10.1007/s00779-019-01223-2

[4] Z. Chen, L. Zhang, C. Jiang, Z. Cao, and W. Cui, "WiFi CSI based passive human activity recognition using attention based BLSTM," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2714–2724, Nov. 2019.

[5] H. Yan, Y. Zhang, Y. Wang, and K. Xu, "WiAct: A passive WiFi-based human activity recognition system," *IEEE Sensors J.*, vol. 20, no. 1, pp. 296–305, Jan. 2020.

[6] F. Liu et al., "Integrated sensing and communications: Towards dual-functional wireless networks for 6G and beyond," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 6, pp. 1728–1767, 2022.

[7] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Network.*, 2013, pp. 27–38.

[8] C. Han, K. Wu, Y. Wang, and L. M. Ni, "WiFall: Device-free fall detection by wireless networks," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, 2014, pp. 581–594.

[9] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of WiFi signal based human activity recognition," in *Proc. 21st Annu. Int. Conf. Mobile Comput. Netw.*, ser. MobiCom '15. New York, NY, USA: Association for Computing Machinery, 2015, pp. 65–76. [Online]. Available: https://doi.org/10.1145/2789168.2790093

[10] J. Liu, Y. Zeng, T. Gu, L. Wang, and D. Zhang, "Wiphone: Smartphone-based respiration monitoring using ambient reflected WiFi signals," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 5, no. 1, pp. 1–19, 2021.

[11] Y. Ma, G. Zhou, S. Wang, H. Zhao, and W. Jung, "SignFi: Sign language recognition using WiFi," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 2, no. 1, pp. 1–21, Mar. 2018. [Online]. Available: https://doi.org/10.1145/3191755

[12] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: Gathering 802.11n traces with channel state information," *Comput. Commun. Rev.*, vol. 41, pp. 53–53, 01 2011.

[13] J. Shang and J. Wu, "A robust sign language recognition system with multiple Wi-Fi devices," in *Proc. Workshop Mobility Evolving Internet Architecture*, ser. MobiArch '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 19–24. [Online]. Available: https://doi.org/10.1145/3097620.3097624

[14] M. Schulz, D. Wegemer, and M. Hollick, "Nexmon: The C-based firmware patching framework," 2017, [Online]. Available: https://nexmon.org

[15] S. Jeong, J. Jin, T. Song, K. Kwon, and J. Jeon, "Single-camera dedicated television control system using gesture drawing," *IEEE Trans. Consum. Electron.*, vol. 58, no. 4, pp. 1129–1137, Nov. 2013.

[16] J. Lien et al., "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–19, Jul. 2016. [Online]. Available: https://doi.org/10.1145/2897824.2925953

[17] L. E. Potter, J. Araullo, and L. Carter, "The leap motion controller: A view on sign language," in *Proc. 25th Australian Computer-Human Interaction Conf.: Augmentation, Appl., Innovation, Collaboration.*, 2013, pp. 175–178. [Online]. Available: https://doi.org/10.1145/2541016.2541072

[18] Ettus Research, "USRP hardware driver and USRP manual," 2022. [Online]. Available: https://files.ettus.com/manual/

[19] T. Akter, "Privacy considerations of the visually impaired with camera based assistive tools," in *Proc. Conf. Companion Publication Comput. Supported Cooperative Work Social Comput.*, 2020, pp. 69–74. [Online]. Available:https://doi.org/10.1145/3406865.3418382

[20] J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li, "Gesture recognition with a 3-D accelerometer," in *Proc. 6th Int. Conf. Ubiquitous Intell. Comput.*, ser. UIC '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 25–38. [Online]. Available: https://doi.org/10.1007/978-3-642-02830-4_4

[21] J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: Accelerometer-based personalized gesture recognition and its applications," in *Proc. IEEE Int. Conf. Pervasive Comput. Commun.*, 2009, pp. 1–9.

[22] G. Malysa, D. Wang, L. Netsch, and M. Ali, "Hidden Markov model-based gesture recognition with FMCW radar," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*2016, pp. 1017–1021.

[23] R. Nandakumar, B. Kellogg, and S. Gollakota, "Wi-Fi gesture recognition on existing devices," 2014. [Online]. Available: https://arxiv.org/abs/1411.5394

[24] W. He, K. Wu, Y. Zou, and Z. Ming, "WiG: WiFi-based gesture recognition system," in *Proc. 24th Int. Conf. Comput. Commun. Netw.*, 2015, pp. 1–7.

[25] S. Tan and J. Yang, "Fine-grained gesture recognition using WiFi," in *Proc. IEEE Conf. Comput. Commun. Workshops*, 2016, pp. 257–258.

[26] A. Q. Mohammed and F. Li, "WiGeR: WiFi-based gesture recognition system," *Int. J. Geo- Inf.*, vol. 5, no. 6, pp. 1–7, 2016.

[27] Z. Tian, J. Wang, X. Yang, and M. Zhou, "WiCatch: A Wi-Fi based hand gesture recognition system," *IEEE Access*, vol. 6, pp. 16911–16923, 2018.

[28] H. Abdelnasser, M. Youssef, and K. A. Harras, "WiGest: A ubiquitous WiFi-based gesture recognition system," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 1472–1480.

[29] J. Wang, D. Vasisht, and D. Katabi, "RF-IDraw: Virtual touch screen in the air using RF signals," in *Proc. ACM Conf. SIGCOMM*, ser. SIGCOMM '14. New York, NY, USA: Association for Computing Machinery, 2014, pp. 235–246. [Online]. Available:https://doi.org/10.1145/2619239.2626330

[30] Rohde and S. China, "802.11ac technology introduction white paper," White Paper, 2012. [Online]. Available: https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1ma192/1MA192_7e_80211ac_technology.pdf

[31] X. Li, J. Luo, and R. Younes, "ActivityGan: Generative adversarial networks for data augmentation in sensor-based human activity recognition," in *Adjunct Proc. ACM Ubicomp ISWC*, ser. UbiComp-ISWC '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 249–254. [Online]. Available: https://doi.org/10.1145/3410530.3414367

[32] J. Bouvrie, "Notes on convolutional neural networks," *Neural Nets.*, MIT CBCL Tech. Rep. Accessed: Apr. 25, 2020, 2006. [Online]. Available: http://cogprints.org/5869/1/cnn_tutorial.pdf

[33] E. B. Hunt, J. Marin, and P. J. Stone, "Experiments in induction," *Amer. J. Psychol.*, vol. 80, no. 4, 1966.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 11 1997. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[35] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, [Online]. Available: https://www.tensorflow.org/
[36] F. Pedregosa et al., "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, no. pp. 2825–2830, Oct. 2011.
[37] R. W. Sperry, "Cerebral organization and behavior," *Science*, vol. 133, no. 3466, pp. 1749–1757, 1961. [Online]. Available: https://science.sciencemag.org/content/133/3466/1749
[38] S. Sigg, S. Shi, F. Büsching, Y. Ji, and L. Wolf, "Leveraging RF-channel fluctuation for activity recognition: Active and passive systems, continuous and RSSI-based signal features," in *Proc. Int. Conf. Adv. Mobile Comput. Multimedia*, 2013, pp. 43–52.
[39] C. Xiao, Y. Lei, Y. Ma, F. Zhou, and Z. Qin, "DeepSeg: Deep-learning-based activity segmentation framework for activity recognition using WiFi," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5669–5681, Apr. 2021.
[40] M. Sulaiman, S. A. Hassan, and H. Jung, "True detect: Deep learning-based device-free activity recognition using WiFi," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops*, 2020, pp. 1–5.
[41] J. Zhang et al., "Data augmentation and dense-LSTM for human activity recognition using WiFi signal," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4628–4641, Mar. 2021.
[42] J. Xiong and K. Jamieson, "SecureArray: Improving WiFi security with fine-grained physical-layer information," in *Proc. 19th Annu. Int. Conf. Mobile Comput. Network.*, New York, NY, USA:Association for Computing Machinery, 2013, pp. 441–452. [Online]. Available: https://doi.org/10.1145/2500423.2500444
[43] S.-H. Kim, J. Ok, H. J. Kang, M.-C. Kim, and M. Kim, "An interaction and product design of gesture based TV remote control," in *Proc. Extended Abstr. Hum. Factors Comput. Syst.*, ser. CHI EA '04. New York, NY, USA: Association for Computing Machinery, 2004, Art. no. 1548. [Online]. Available: https://doi.org/10.1145/985921.986124
[44] J. Hou et al., "Signspeaker: A real-time, high-precision smartWatch-based sign language translator," in *Proc. 25th Annu. Int. Conf. Mobile Comput. Network.*, ser. MobiCom '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3300061.3300117
[45] Nintendo, "Nintendo switch," 2017. [Online]. Available: https://www.nintendo.com/switch/
[46] S. Disabato and M. Roveri, "Incremental on-device tiny machine learning," in *Proc. 2nd Int. Workshop Challenges Artif. Intell. Mach. Learn. Internet Things*, ser. AIChallengeIoT '20.New York, NY, USA: Association for Computing Machinery, 2020, pp. 7–13. [Online]. Available: https://doi.org/10.1145/3417313.3429378
[47] M. A. Merzoug, A. Mostefaoui, M. H. Kechout, and S. Tamraoui, "Deep learning for resource-limited devices," in *Proc. 16th ACM Symp. QoS Secur. Wireless Mobile Networks*ser. Q2SWinet '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 81–87.[Online]. Available: https://doi.org/10.1145/3416013.3426445
[48] S. Gopinath, N. Ghanathe, V. Seshadri, and R. Sharma, "Compiling KB-sized machine learning models to tiny IoT devices," in *Proc. 40th ACM SIGPLAN Conf. Programm. Lang. Des. Implementation*, ser. PLDI 2019. New York, NY, USA:Association for Computing Machinery, 2019, pp. 79–95. [Online]. Available: https://doi.org/10.1145/3314221.3314597

Guiping Lin received the B.Eng. degree in Internet of Things engineering and the M.Sc. degree in computer science and technology from the School of Computer and Information, Anhui Normal University, Wuhu, China, in 2019 and 2022, respectively. She is currently working toward the doctoral degree with the Harbin Institute of Technology, Shenzhen, China.

Her current research interests include wireless communication and sensing and human–computer interaction.

Weiwei Jiang received his B.Eng. degree in the internet of things from Huazhong University of Science and Technology, Wuhan, China and M.Sc. degree in information science from Japan Advanced Institute of Science and Technology, Nomi, Japan, in 2014 and 2016, respectively. He is currently working towards the Ph.D. in computer science with the University of Melbourne.

His research interests include ubiquitous computing, wireless communications and sensing, and human–computer interaction.

Sicong Xu received the B.Eng. degree in Internet of Things engineering in 2019 from the School of Computer and Information, Anhui Normal University, Wuhu, China, where he is currently working toward the M.Sc. degree in computer science and technology.

His research interests include backscatter communications and wireless sensing.

Xiaobo Zhou (Senior Member, IEEE) received the B.Sc. degree in electronic information science and technology from the University of Science and Technology of China (USTC), Hefei, China, the M.E. degree in computer application technology from the Graduate University of Chinese Academy of Science (GUCAS), Beijing, China, and the Ph.D. degree in information science from the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, in 2007, 2010, and 2013, respectively.

From April 2014 to March 2015, he was a Researcher with the Department of Communications Engineering, University of Oulu, Oulu, Finland. He is currently a Professor with the School of Computer Science and Technology, College of Intelligence and Computing, Tianjin University, Tianjin, China. His research interests include Internet of Things, cloud computing, data center networks, vehicular networks, and mobile edge computing.

Xing Guo received the B.A. degree in sociology, and the M.Sc. and Ph.D. degrees in computer science and technology from Anhui University, Hefei, China, in 2004, 2009, and 2013, respectively.

He is currently an Associate Professor with the School of Computer Science and Technology, Anhui University and a Postdoctoral Researcher with the School of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei, China. His research interests include computer graphics, backscatter communications, and human–computer interaction.

Yujun Zhu received the B.Sc. degree in computer science and technology from Qingdao University, Qingdao, China in 2006, the M.Sc. degree in computer application technology from Anhui Polytechnic University, Wuhu, China, in 2009, and the Ph.D. degree in computer software and theory from Tongji University, Shanghai, China, in 2014.

He is currently an Associate Professor with the School of Computer and Information, Anhui Normal University. His research interests include wireless sensor network and IoT applications.

Xin He (Member, IEEE) received the M.Sc. degree in information science from the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), Ishikawa, Japan, in 2013, and the Ph.D. degree from JAIST and the University of Oulu, Oulu, Finland, in 2016.

He is currently an Associate Professor with the School of Computer and Information, Anhui Normal University. He is also a Postdoctoral Researcher with the School of Computer Science and Technology, University of Science and Technology of China, Hefei, China. His research interests include joint source-channel coding, cooperative wireless communications, network information theory, energy harvesting, backscatter communications, and human–computer interaction.

Dr. He was the recipient of the Best Paper Runner up Award on the IEEE/ACM International Workshop on Quality of Service (IWQoS) 2020.