

Research Article

Exploiting Serialized Fine-Grained Action Recognition Using WiFi Sensing

Weiyan Tong ¹, Rong Li ¹, Xiaoqing Gong ¹, Shuangjiao Zhai ¹, Xia Zheng ²,
and Guixin Ye ¹

¹School of Information Sciences and Technology, Northwest University, Xi'an 710127, Shaanxi, China

²School of Art and Archaeology, Zhejiang University, Hangzhou 310028, Zhejiang, China

Correspondence should be addressed to Xiaoqing Gong; gxq@nwu.edu.cn

Received 29 July 2021; Revised 30 September 2021; Accepted 15 October 2021; Published 8 November 2021

Academic Editor: Jose M. Barcelo-Ordinas

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

Gestures serve an important role in enabling natural interactions with computing devices, and they form an important part of everyday nonverbal communication. In increasingly many application scenarios of gesture interaction, such as gesture-based authentication, calligraphy, sketching, and even artistic expression, not only are the underlying gestures complex and consist of multiple strokes but also the correctness of the gestures depends on the order at which the strokes are performed. In this paper, we present WiCG, an innovative and novel WiFi sensing approach for capturing and providing feedback on stroke order. Our approach tracks the user's hand movement during writing and exploits this information in combination with statistical methods and machine learning techniques to infer what characters have been written and at which stroke order. We consider Chinese calligraphy as our use case as the resulting gestures are highly complex, and their assessment depends on the correct stroke order. We develop a set of analyses and algorithms to overcome many issues of this challenging task. We have conducted extensive experiments and user studies to evaluate our approach. Experimental results show that our approach is highly effective in identifying the written characters and their written stroke order. We show that our approach can adapt to different deployment environments and user patterns.

1. Introduction

Hand gestures are an important communication channel for humans [1] and provide a natural way to support human-computer interactions [2]. Recent years have witnessed increased interest in exploiting wireless signals such as WiFi [3–7] or RFID [8] for gesture recognition. Compared to other technologies, there are several advantages to wireless signals. First, solutions based on wireless signals have low deployment overhead as only Commercial Off-The-Shelf (COTS) wireless devices are required. Second, wireless signals allow users to perform gestures naturally without any instrumentation of the user. Indeed, previous work on nonwireless modalities has focused either on instrumenting the environment with video cameras [9] or depth cameras [10] or indirectly instrumenting the user by performing the recognition on smartphones, smartwatches, or other wearables, e.g., taking advantage of inertial sensors or audio

[11]. The former suffers from being highly privacy-intrusive, whereas the latter requires users to hold (or wear) a device, decreasing the naturalness of the gestures.

A significant drawback of current wireless gesture recognition systems is that they are predominantly designed for supporting a (relatively small set of) simple, unambitious gestures, such as interface commands [12]. In many application scenarios, more complex gestures that consist of several strokes and that potentially contain a high degree of ambiguity need to be supported. Indeed, domains requiring support for complex gestures range from authentication [13] to interactive learning [14], sketching [15], and even artistic expression [16]. In these domains, recognizing the strokes constituting the gesture alone is not sufficient, but also their *order* must be known.

Taking Chinese calligraphy as an example, it is composed of different strokes in a certain order. Most Chinese characters are formed by dozens of strokes in different

arrangements. Therefore, to recognize Chinese characters, it is first necessary to identify their strokes. However, the position and size of the same stroke may be different in different Chinese characters. The existing wireless sensing methods cannot solve this difference, and there is no applicable and generalizable method for identifying fine-grained complex continuous actions (such as strokes in Chinese characters). Therefore, it is necessary to provide a general and effective method for such complex and continuous motion recognition.

In this paper, we seek to extend the scope of wireless gesture recognition systems to support complex *multistroke* gestures and to recognize the user's *stroke order*. To achieve these goals, we introduce WiCG, a novel WiFi-based approach for capturing and providing feedback on fine-grained gestures. WiCG uses two COTS WiFi devices as illustrated in Figure 1 to track the user's hand movement during writing and does not require the user to carry any device. As a case study, we consider an interactive learning scenario for Chinese calligraphy that aims at providing feedback on appropriate stroke order. Chinese calligraphy serves as a representative example of application domains that benefit from our approach as individual characters consist of several complex primitives (see Figure 2) and as optimal stroke order has a strong influence on the final output (see Figure 3). Indeed, mastering stroke order is generally perceived as difficult for calligraphy learners because many Chinese characters do not follow a general rule.

WiCG exploits WiFi channel state information (CSI) to first track the user's hand movement and then uses this information to identify which strokes have been written to infer the corresponding characters and stroke order. It then provides the user with the correct stroke order for incorrectly written characters, by comparing the detected stroke order against a database of standardized order. We design a range of new analysis and models based on statistical methods and machine learning to achieve our objectives: (1) to enable WiCG to adapt to different deployment sites and user patterns, (2) to map the tracked CSI signal to specific strokes, and (3) to exploit contextual information to infer what characters are most likely to be written when ambiguity arises. The result is a new way to use CSI to track fine-grained gesture movements and in which order the movements were performed. We believe the proposed study can encourage a new line of research on fine-grained gesture recognition using CSI.

We have implemented a working prototype of WiCG and conducted extensive experiments to evaluate its performance. Experimental results show that WiCG can successfully identify about 80% of the characters in our experiments. In terms of strokes, WiCG achieves 93% accuracy for identifying simple strokes and 79% accuracy for complex ones.

Summary of contributions:

- (i) We demonstrate, for the first time, how WiFi signals can be used to track fine-grained gesture movements.
- (ii) Our work is the first to exploit CSI to automatically track written characters and their stroke order. The



FIGURE 1: The deployment of WiCG.

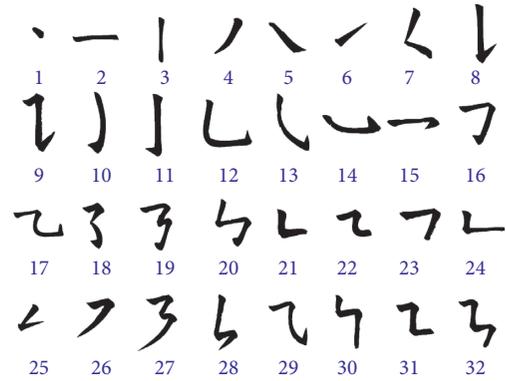


FIGURE 2: 32 basic strokes.

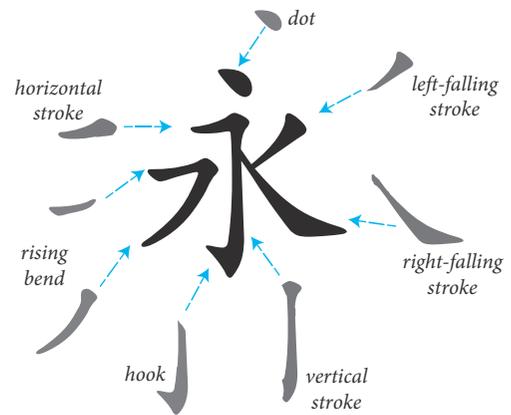


FIGURE 3: Character 永.

proposed techniques are generally applicable and can be applied to many other gesture recognition domains.

- (iii) We show how to combine statistical methods and machine learning techniques to enable the developed system to adapt to various environments.

2. Related Work

2.1. WiFi Signal-Based Applications. WiFi-based sensing attracts considerable attention because of the wide

deployment of WiFi infrastructures. Much of the previous studies develop a device-free solution by utilizing commercial, off-the-shelf wireless devices to sense the environment or object movements. In particular, there are many research results in positioning and target detection [17, 18]. However, these works are limited to coarse-grained tracking and detection (e.g., detecting whether there is a human or a pet in the monitoring area), which cannot meet the needs of fine-grained human activity detection and human-computer interaction. Beyond the target detection and localization functions of WiFi systems, many WiFi-based applications and systems have been proposed in recent research, including gesture recognition [5, 19], user identification [4, 6], health and risk assessment [20], and activity detection [21]. These works prove that WiFi has emerged as a powerful medium for sensing information in performing complex tasks. These works inspire us to exploit serialized fine-grained action recognition leveraging WiFi signals.

2.2. Fine-Grained Human Activities. Due to the ability of capturing subtle movement with CSI of WiFi signals, more recent works propose to recognize fine-grained human activity, through WiFi signals. Wang et al. [22] present a system to detect tiny breath of human by using a common WiFi device and a Fresnel zone model. Melgarejo et al. [23] leverage a directional antenna to recognize the fine-grained gestures. Inspired by that, Ali et al. [24] use existing commercial wireless devices and CSI values to recognize keyboard keys, such as 26 letters and numeric keys. WiFinger [25] makes finger and device interact through WiFi signals. Li et al. [26] study how CSI of keystrokes gesture can be leveraged to crack the digital passwords of Alipay. However, no work so far has exploited wireless signals to track and identify fine-grained gesture movements on a continual basis. Our work develops a novel framework to do so. We evaluate our approach by applying it to identify stroke order for Chinese calligraphy, a challenging task for gesture recognition.

2.3. Handwritten Character Recognition. Handwritten Chinese character recognition is one of the most important research fields in pattern recognition. Most of research works using image processing techniques for handwritten characters recognizing [27], but it is difficult for Image-based Chinese character recognition to determine whether the stroke order is correct. RF-Copybook [28] uses wireless signals for Chinese calligraphy recognition, but it has several limitations. RF-Copybook uses two RFID tags attached to the ink brush; i.e., it is not device-free. RF-Copybook can track the hand movements for writing, but it requires richer information like what character is being written and assumes the character is written with the correct stroke order. While having such information can simplify fine-grained gesture recognition, it would be highly inconvenient to supply this information in practice. Furthermore, RF-Copybook can only track one character at a time and requires training for individual users and font sizes. WiCG employs a set of novel

techniques to avoid these drawbacks to offer a device-free solution.

Unlike existing works, our work aims to focus more on how characters have been written than just what has been written. We utilize wireless signals to capture continuous fine-grained writing gestures for handwritten calligraphy character recognition, which is nonintrusive and device-free and does not require changing writing tool and habit.

3. Background

WiCG augments existing WiFi gesture recognition systems to support *multistroke* gestures and to capture *stroke order* for handwritten character recognition. As a case study for WiCG, we consider Chinese calligraphy since the resulting characters are composed of several strokes, and the appropriate ordering of strokes is nontrivial for learners. Another challenge in Chinese calligraphy is that the strokes forming characters contain a high degree of ambiguity, with only subtle differences separating primitives. This challenge makes Chinese calligraphy difficult for *any* gesture recognition system, let alone one operating using wireless channel information. In the following we describe our case study system and the use of CSI for WiFi sensing.

3.1. Description of Case Study. As a case study, we build a prototype system to track the hand gestures of calligraphy learners for writing Chinese characters. Writing a Chinese character involves a sequence of fine-grained hand movements for writing a set of strokes. Our system serves as a good use case for fine-grained gesture recognition as tracking strokes mimics many real-world applications where we need to track the order of sketching lines during drawing or detecting continuous user gestures in human-computer interactions.

For this specific case study, our task is to identify what characters have been written and whether they are written in the correct order. We note that drawing on a tablet is often not an alternative, as doing this would affect user experience. This is because the subtle nuances of execution, where a stroke was made swiftly or slowly, whether the brush was put to the paper with great delicacy or force, would significantly affect the aesthetics of the writing or drawing. One of the advantages of our approach is that it does not require users to change their writing tool and habits.

3.2. Channel State Information. Our technique to detect fine-grained hand movements based on the CSI of WiFi signals. CSI has been proven to be useful in prior wireless sensing tasks [26]. Our key observation is that the hand movement during writing will introduce a multipath effect to the WiFi signal, and this interference will result in a specific CSI pattern. This means that we can find a unique mapping from a CSI pattern to a stroke. With this mapping mechanism in place, we can then build a system to infer what strokes have been written and in which order automatically, by analyzing the CSI data collected during writing.

As an example, consider Figure 3, which shows character 永 (eternity) in regular script. This character can be broken down into eight basic strokes illustrated in the figure. Figure 4 shows the measured CSI amplitudes for the eight basic strokes of character 永. In this example, we asked a user to write each stroke using an ink brush five times, and the five measurements are shown in each subgraph. It can be seen from the diagram that the measured CSI amplitudes are more or less consistent over time for the same stroke and are sufficiently different across strokes.

4. WiCG Design and Overview

4.1. Design Goals. WiCG has been designed as a novel WiFi-based system to track and identify strokes during drawing or sketching. WiCG has been designed to meet three design constraints that are fundamental to ensure a good level of usability and acceptability of the resulting system. First, the system should not require the user to carry any device (i.e., device-free) because it will reduce its usability. Second, our solution should be low-cost and do not require specialized hardware. Third, the system should collect as little privacy-sensitive information as possible. In view of the application context and our design goal, we find that the WiFi signal is a good fit for the problem we target. This is because there are many low-cost WiFi devices available, and our initial experiment suggests that CSI can precisely capture the written strokes (see Figure 4).

4.2. Overview of WiCG. We present a user-friendly Chinese character recognition system, WiCG, which leverages commodity WiFi devices to infer what characters have been written and at which stroke order based on user’s hand movements with real-time response during writing. Figure 5 depicts the workflow of WiCG, which includes the following steps:

Data collection: WiCG collects CSI data when the user writes strokes or characters. WiCG currently supports Chinese characters written in regular script, a standard style usually used by calligraphy beginners.

Data preprocessing: WiCG uses Butterworth low-pass filter to remove noises. Then, WiCG conducts segmentation scheme to obtain stroke-level CSI segments. After that, WiCG uses the mapping function to transform CSI segments to eliminate the differences from different environments.

Feature extraction: WiCG uses a range of statistical features that were found to be helpful in prior work [29] of WiFi sensing to capture the essential feature of stroke-level CSI segments. Then, WiCG normalizes each feature to the same scale and reduces the dimension.

Identification: in this stage, the extracted features are fed into the Random Forest classifier for stroke identification. Using the identified strokes, WiCG infers what characters have been written by exploiting the unique stroke combinations of characters. To

disambiguate characters with similar stroke structures, a Long Short-Term Memory (LSTM) network is employed to exploit the context (e.g., what characters are likely to appear together to form a word or a sentence) to infer which character is most likely to be written. In the last step, WiCG identifies whether the characters are written in the correct stroke order by comparing the detected stroke order against a database of standardized order.

5. System Design

5.1. Data Preprocessing

5.1.1. Signal Denoising. CSI is inherently noisy due to the multipath effect of the surrounding environment. We remove noise using a classical Butterworth low-pass filter to remove any signal with a higher frequency than 30 Hz. This is based on the following observation: CSI change caused by the hand movement during writing is often within a frequency domain of 2–30 Hz [7]. Our experimental evaluation suggests that this assumption works well.

Figure 6(a) shows the raw, noisy CSI amplitudes collected when writing character 永. Figure 6(b) gives the CSI amplitudes after applying our denoising method. A close look at the diagram shows that the CSI after removing noise is a combination of some of the basic strokes shown in Figure 4. This example shows that our method can cancel the multipath effect of the physical environment to capture the subtle CSI patterns for character strokes effectively.

5.1.2. Signal Segmentation. We propose two segmentation schemes, namely, character-level segmentation and stroke-level segmentation.

(1) Character-Level CSI Segmentation. The first challenge of WiCG is to identify when the user started and finished writing a character. This information is essential for inferring what specific characters have been written to discover incorrect stroke order. To do so, we first remove the noise of the CSI data and then divide the CSI data into *character-level* segments where each segment corresponds to a character. Note that we do not concern what specific characters and we just mark the start and end points of each segment at this stage.

Our hypothesis for character-level segmentation is that users typically pause for a few seconds before writing the next character. This is because learners need either to have a close look, from the copybook, on the structure of the next character to write, or to dip the brush in ink. To test this hypothesis, we invite 20 participants to write down a sentence with 22 Chinese characters. We record the writing process using a video camera and measure the gap between two consecutive characters by counting the number of video frames. We find that, on average, our participants pause for 2.1 seconds between characters. The minimum break time is 1.2 seconds, and the maximum is 3.5 seconds. This experiment confirms our hypothesis.

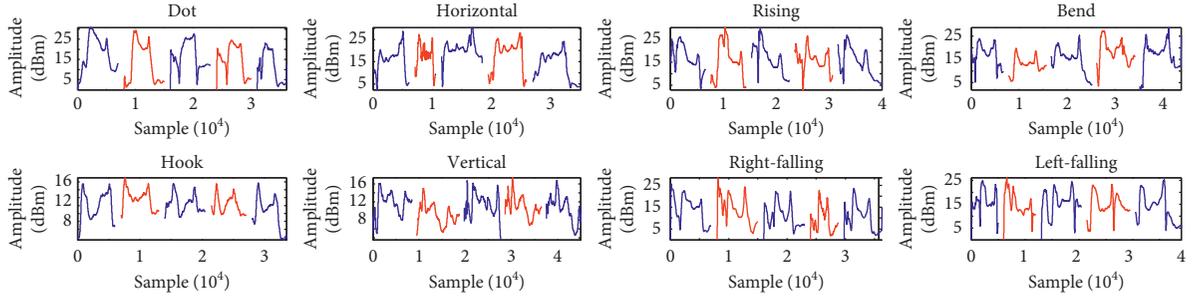


FIGURE 4: Corresponding CSI amplitudes for writing the eight Chinese character strokes illustrated in Figure 3.

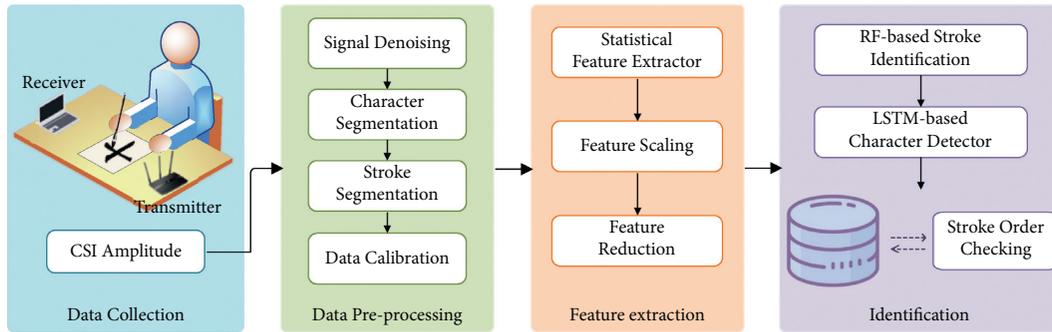


FIGURE 5: System overview.

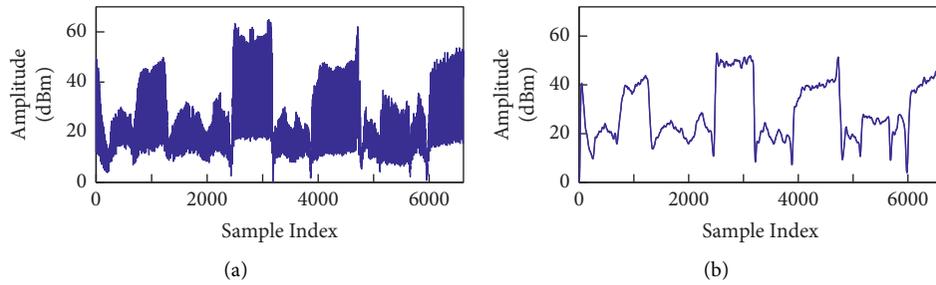


FIGURE 6: Illustration of signal denoising. (a) Raw CSI amplitude. (b) Denoised CSI amplitude.

Hence, we present a sliding window segmentation method using the cumulative difference in amplitude, based on the fact that the CSI amplitude change will be much more prominent when the user performs a writing gesture. To depict the change of CSI amplitude, we define cumulative amplitude difference D as follows:

$$D_i = \sum_{i=1}^{N-L} |(C_{i+L} - C_i)|, \quad (1)$$

$$C_i = \sum_{t=1}^i A_t,$$

where A_t denotes a CSI amplitude value at time t , and C_i denotes the cumulative value of the first i amplitudes. L is the length of a sliding window, which is set according to the sampling rate, and N is the number of samples. Figure 7 gives the CSI measurement collected when a user was writing the

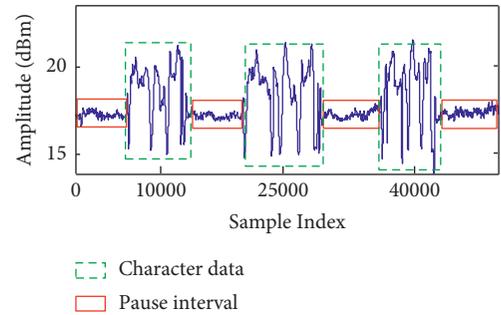


FIGURE 7: A pause between writing two characters.

first three Chinese characters, (eternity), (harmony), and (nine), from *The Orchid pavilion* (i.e.), a famous Chinese calligraphy work. As can be seen from this diagram, the pause between two characters leads to a consistent CSI

pattern. Thus, we can utilize the cumulative amplitude difference for character segmentation. Specifically, WiCG first calculates the minimum and maximum values of the accumulated amplitude difference (D_i) to determine a rough single character data segment and then takes two-thirds of the difference between the maximum and minimum values as the dynamic threshold p . Finally, WiCG utilizes sliding windows to compare the cumulative amplitude difference with p for locating the starting and ending point of a single character.

(2) *Stroke-Level CSI Segmentation.* The regular script has obvious characteristics with square shapes, straight strokes, and no continuous writing phenomenon. In calligraphy practice, we found that there will be lifting and falling movements when writing each stroke since the user needs to put the ink brush down before writing a stroke and then lift it after writing. Figure 8(a), respectively, shows the ten times lifting and falling movements when the user is in stroke writing. We can observe that apparent increases of the CSI amplitudes appear when the user performs the gesture of lifting a pen and the CSI amplitudes show obvious decreases when the user performs the gesture of falling a pen. Moreover, the amplitude changes of the lifting and falling movements are more compared with writing strokes.

Similar to our observations for the character-level segmentation, the lifting and falling movements lead to unique CSI patterns which can be used to segment strokes. As shown in Figure 8(b), the CSI data of the Chinese character 永 is divided into five segments, corresponding to five strokes: (1) dot, (2) horizontal turning and hook, (3) horizontal and left-falling, (4) left-falling, and (5) right-falling. We train a backpropagation neural network (BP) model based on lifting and falling movements to obtain a single stroke segment. BP neural network, a kind of multilayer forward neural network based on error backpropagation algorithm, can learn and store a large amount of input-output models mapping relationship. We choose BP neural network due to its powerful ability of nonlinear fitting [30].

5.1.3. *Calibration.* The first step of our approach is to calibrate the impact caused by the physical environment and the user's writing pattern (e.g., writing speed and font size). Our solution is to learn a mapping function to translate the CSI obtained in the user's environment to one collected in our training environment. Specifically, we wish to learn a mapping function, f , to take in the CSI signal vector, \bar{x}_u , collected in the user's environment, and translate it to another CSI signal vector, \bar{x}_t , i.e., $\bar{x}_t = f(\bar{x}_u)$. We want to ensure that, for a given stroke i , the function output, $\bar{x}_{t,i}$, is close to the *standard* CSI pattern (we refer to a standard CSI pattern as the CSI data collected for a specific stroke from our training data collection environment) collected in our training environment. If we are able to learn such a function, we can then use the transformed vector, \bar{x}_t , to match our database to infer what strokes have been written.

We have also tested a range of linear and nonlinear methods. We find that a simple linear regression function,

$f(\bar{x}_u) = w \cdot \bar{x}_u + b$, is enough for our purpose. Figure 9 shows the calibration result of the user writing pattern of the vertical stroke to the standard pattern. It can be seen that the mapping function can well eliminate the difference in user pattern. We thus decided not to use more sophisticated models like neural networks because these methods typically require having a large number of samples to learn an effective model.

We ask the user to write multiple times a specific Chinese character during the calibration step, 永 with a similar font size that the user will write later for other characters. We choose this character because it is often used to illustrate the basic strokes of Chinese calligraphy and thus many learners are familiar with it. However, other characters can be used. The number of times the user needs to write is determined by making sure the 95% confidence interval across collected data is narrower than a threshold (10% in our case). In our experiments, we find that our participants need to write the character 3.5 times on average.

To determine the weights (w and b) of the mapping function, we follow several steps. First, we split the CSI data into segments, one segment per stroke. The stroke-level segmentation heuristic is described in the following subsection. Then, we scale the collected CSI data to the same time frame of the *standard* CSI for each stroke. The scaled data will be organized as a vector of real values.

Furthermore, during the scaling process, we also make sure that both the collected and the *standard* CSI vectors have the same number of data points. Finally, we find the set of parameters for w and b , which minimizes the mean squared error between the scaled CSI and the *standard* CSI data across stroke samples. This calibration process is only performed once before each writing.

5.2. *Feature Extraction.* To recognize a character, WiCG first needs to identify each character's strokes and then infer what writing characters. One of the critical aspects of building a successful machine learning model is finding the right features to characterize the input data. In this subsection, our goal is to extract reliable features for stroke identification.

5.2.1. *Raw Features.* To capture the essential characteristics of stroke-level CSI, we consider 24 raw features as shown in Table 1. Each WiFi channel has 30 subcarriers, and we have 720 (24×30) raw features in total. The features are obtained from the time (T) and the frequency (F) domains of the CSI data. Some of these features are selected based on information gain (IG), while others are chosen based on prior work [31]. Information gain is a feature selection algorithm, which is used to estimate how useful a feature is. Different features have different information gains, and features with large information gains have stronger classification capabilities. Information gain can be calculated by the following formula:

$$\text{Gain}(S, A) = E(S) - E(S|A),$$

$$E(S) = - \sum_{i=1}^n p_i \log_2 p_i, \quad (2)$$

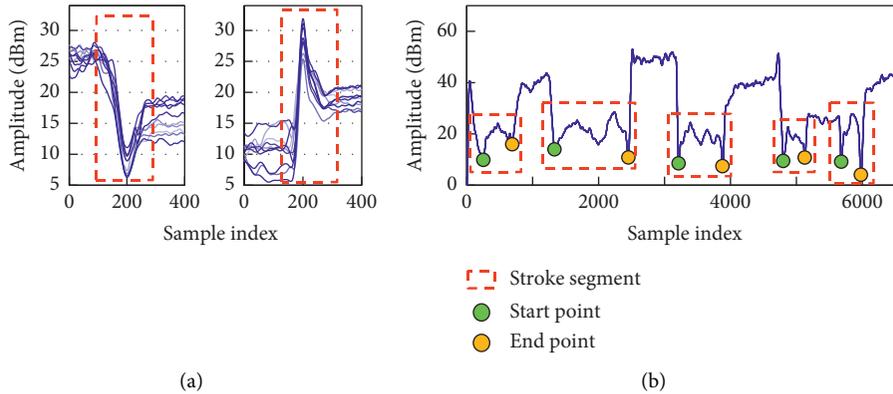


FIGURE 8: Stroke segmentation of character 永 leveraging the CSI change caused by a lifting and falling movement. (a) Hand movements. (b) Stroke segmentation.

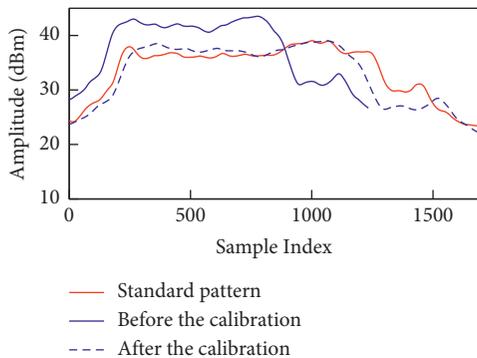


FIGURE 9: Calibration of user writing pattern.

TABLE 1: Time (T) and frequent (F) domain features used for stroke recognition.

ID	Features	IG
1	Mean absolute deviation (T)	2.161
2	RMS amplitude (T)	2.110
3	Mean (T)	2.071
4	Median (T)	2.060
5	Maximum (T)	1.856
6	Minimum (T)	1.839
7	Standard deviation frequency (F)	1.782
8	Mean frequency (F)	1.086
9	Kurtosis coefficient (T)	0.825
10	Magnitude margin factor (T)	0.783
11	Peak factor (T)	0.737
12	Impulse factor (T)	0.731
13	Variance (T)	0.649
14	Standard deviation (T)	0.643
15	Form factor (T)	0.628
16	Peak-to-peak value (T)	0.585
17	Kurtosis factor (T)	0.534
18	RMS frequency (F)	0.507
19	Skew coefficient (T)	0.446
20	Skewness factor (T)	0.403
21	Variance frequency (F)	0.403
22	Energy (F)	0.393
23	Central frequency (F)	0.391
24	Quartile deviation (T)	0.385

where $E(S)$ is the information entropy of the sample set S where S is the sample set, and $E(S|A)$ represent the information entropy after the sample set S is divided by the feature A . p_i is the proportion of each category in the sample set S , and n represents the number of categories.

5.2.2. *Feature Scaling.* Supervised learning will work better if the feature values lie in a certain range. Therefore, we scale the value for each of our features between the range of 0 and 1. We record the maximum and minimum value of each feature found at the training phase and use these values to scale CSI features obtained from the real user’s environment.

5.2.3. *Feature Reduction.* Given our relatively small number of training samples (32 strokes \times 5 users involved in training data collection), we need to find a compact set of features in order to build an effective predictor. Feature reduction is automatically performed through applying Principal Component Analysis (PCA) on the scaled raw features. This technique removes the redundant features by linearly aggregating features that are highly correlated. After application of PCA, we use the top 7 principal components (PCs) which account for over 95% of the variance of the original feature space. We record the PCA transformation matrix and use it to transform the raw features of the stroke-level CSI collected from the deployment environment. Figure 10 illustrates how much feature variance that each component accounts for. This figure shows that prediction can accurately draw upon a subset of aggregated feature values.

5.3. *Stroke and Character Identification.* In this subsection, we first introduce how to recognize a stroke and then introduce the process of inferring Chinese characters based on identified candidate strokes.

5.3.1. *Stroke Identification.* After partitioning a character-level segment into stroke-level segments, our goal is to map each stroke-level segment to a specific stroke. In this work, we use the Random Forest (RF) algorithm to build the stroke identification model. The RF method consists of a set of

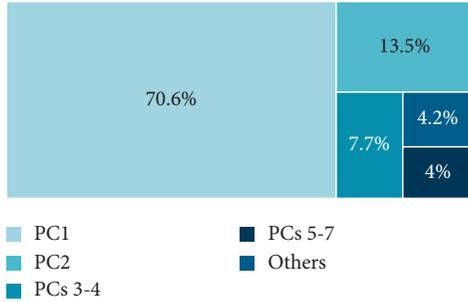


FIGURE 10: The percentage of explained variance of each principal component.

decision trees (10 in our cases). The main strength of this method is its more precision due to randomness presented in several forms, such as the random selection of the features and the random selection of the training samples. Moreover, we chose RF because it is proven to be robust to noise and can avoid overfitting [32]. In Section 7.3, we also explore the prediction accuracy of various alternative modeling techniques and conclude that RF has the best overall performance.

(1) *Building Stroke Identification Model.* The inputs to the stroke identification model are statistical feature vectors extracted from collected CSI data of 32 basic Chinese strokes, and the outputs are a set of categories obtained by a voting method indicating which of the 32 basic strokes may have been written. The set of features used is described in Section 5.2. Building and using such a model follow the 3-step process for supervised machine learning: (i) generate training data, (ii) train a model, and (iii) use the model. These steps are described as follows.

- (i) *Generate Model Training Data.* Our RF model is trained *offline* based on CSI examples collected for the 32 basic strokes of Chinese characters. To generate the training data to build the stroke identification model, we have asked five participants to write down each basic stroke multiple times. We record the raw CSI data for each stroke per user during each writing. We apply our signal denoising method to the raw CSI data and then average the CSI signals across multiple writing for each user. This allows us to build a training dataset that contains a stroke-level CSI pattern per basic stroke per user.
- (ii) *Training the Model.* The stroke labels and their corresponding CSI feature vectors are passed to our supervised learning algorithm. The learning algorithm tries to find a correlation between the feature values and the desired stroke label. The output of our learning algorithm is the RF model where the weights of the model are instantiated using the training data. Training is performed offline only once, and the learned model can be used for any unseen basic strokes without extra training.
- (iii) *Using the Model.* Once we have built and trained the stroke identification model as described above, we

can perform stroke identification using CSI data collected in the real user's environment and no further retraining is required. To make a stroke prediction, the input stroke-level CSI segments will be first transformed using the mapping function learned in the calibration phase. WiCG then extracts a feature vector of real values that describe the stroke-level CSI data and feeds the feature values to the learned model. The RF algorithm constructs ten decision trees, among which each tree randomly selects the subsets of training samples and features during training. Hence, it may produce different classification results. In order to identify which stroke was written, our RF model aggregates the results and selects the categories agreed by at least three trees as the candidate stroke set. As a result, the model can produce up to three candidate strokes, further refined in the character identification stage.

(2) *Remove Inaccurate Stroke CSI Partitioning.* As mentioned previously (see Section 5.1.2), the lifting and falling movements in stroke writing can split the stroke-level segments from the character-level CSI. However, we find that there may be more than one way to map a character-level CSI to stroke-level CSI segments (e.g., when the lifting and falling movements are part of a stroke). For example, the 17th basic stroke as shown in Figure 2 can be split into several single strokes wrongly. This invalid segmentation results in different combinations, such as (1) horizontal, vertical, and horizontal stroke; (2) horizontal turning and horizontal stroke; (3) horizontal and vertical bend hook stroke. To overcome this, we enumerate all possible sets of single strokes CSI segmentation during stroke segmentation and apply the trained stroke identification model to recognize each unknown stroke. For each stroke set, WiCG gives the average prediction probability of all strokes and then removes the stroke set with probability (or confidence) less than 0.5. We find that this simple strategy works well, giving no more than three candidate stroke sets in our experiments. These remaining stroke sets will be further refined in the character identification stage.

(3) *Training Cost.* The time consumption of WiCG is comprised of four parts: data collection, data preprocessing, feature extraction, and training the model. Data collection consumes most of the total training time, which takes around two hours. Nevertheless, the latter three components consume less than 30 minutes in our system.

5.3.2. *Character Identification.* For each character segmentation, the stroke identification step produces a set of identified strokes and their writing order. This step then tries to map these strokes to a specific character. We use the identified strokes (*ignoring the order*) of a character-level segment to infer what character may have been written. Since most Chinese characters are a unique combination of strokes, this step gives one candidate character for most stroke sets.

(1) *Multicharacter Scenarios*. There are, however, scenarios where multiple candidate characters may be generated from a single character-level CSI segment. This is because (1) a specific stroke set can be mapped to more than one character (for example, characters, (Earth) and (scholar), are composed of the same strokes, but their standard stroke order is different), or (2) the stroke identification step produces more than one stroke set. For the former, we develop a *context-aware character detector* to guess which character is most likely to be written. For the latter, we first remove stroke sets that cannot lead to a valid character. We then use the context-aware character detector to refine the result.

(2) *Context-Aware Character Detector*. To utilize the context semantic relation of Chinese characters in the corpus, we leverage Long Short-Term Memory (LSTM) architecture of Recurrent Neural Network to learn the long-range dependencies of Chinese characters (contextual features) for probability estimation of candidate characters. As illustrated in Figure 11, we construct a three-layer LSTM architecture.

LSTM is to build based on Recurrent Neural Network (RNN) and each hidden layer consists of an RNN with LSTM units, which abstracts the input as a set of feature representations. Traditional RNN suffers from exploding gradients. The RNN only retains the previous short-term information and cannot deal with the previous semantic relationships in a long sequence. Therefore, RNN cannot fully depict the unique character features based on the context. LSTM is a special kind of RNN, developed to address the exploding and vanishing gradient problem when using traditional RNN [33]. LSTM has shown its superiority in the context-based prediction problem [34–36]. Thus, we employ LSTM rather than the typical RNN for character prediction.

We used a corpus of more than 50,000 articles gathered from traditional and modern Chinese literature and calligraphy copybooks as input to the model. This corpus results in a vocabulary of over 18,000 traditional and Simplified Chinese characters. From this corpus, our model learns a probability distribution over sets of characters seen in the literature. Once the model has been trained, it can estimate the probability of character candidates based on previous characters when a character-level CSI segment is mapped to more than one candidate character. Specifically, for multiple candidate results of the currently written character, we use the model to predict the probability of all candidate characters based on the previously written characters and then select a character with the highest probability as the final prediction result.

(3) *Stroke Order Recognition*. In this step, we compare, for each detected character, the stroke order written by the user against a database of the correct order. If an incorrect stroke order is detected, our system will suggest the correct order to the user. Specifically, in cases where the detected strokes could be mapped to multiple candidate characters, WiCG checks the stroke order for each candidate character and presents the results to the user according to how likely a candidate character was written given the context. The

probability is given by a context-aware character detector. The entire process of gathering raw CSI data, character and stroke identifications, and assessment takes less than 2 minutes when processing 200 characters. This overhead can be further reduced by processing character segments in parallel.

6. Experimental Setup

6.1. *Evaluation Environments*. We evaluate WiCG in three different indoor environments: a meeting room, an office, and a hall corresponding to high, medium, and low multipath environments. Figure 12 shows the experimental deployment in three rooms.

6.2. *System Setup*. Our prototype system is using a TP-Link WDR 4300 wireless router as the transmitter and a laptop with an Intel WiFi Link 5300 NIC as the receiver. Under our evaluation setting, the transmitter and receiver are placed on the user's left and right hand sides separately.

6.3. *Data Collection*. We recruited 5 participants (three females and two males), who were postgraduate students at our institution during the experiments. As shown in Figure 1, they were sitting and writing on a specific practice calligraphy paper with a pen in their right hand. The writing instrument is an ink brush, a traditional writing instrument and painting instrument originated from China, which consists of a nib made of animal hair and a wooden penholder. The paper has standard 10 cm × 10 cm grids (i.e., mats) suitable for calligraphy beginners to practice. Each participant was allowed to practice multiple times, allowing them to follow a natural writing speed.

6.4. *Model Training*. Our stroke recognition model is trained based on the precollected CSI patterns of 32 basic strokes. The training data were collected in an indoor environment of different sizes to our evaluation environments. To provide a fair evaluation, we recruited two groups of users, one participated in training data collection and the other participated in the evaluation. Each participant was asked to write basic strokes 30 times. We collected 4800 ($32 \times 5 \times 30$) basic stroke CSI data from 5 participants in total. For the same purpose, we did not include the Multi-Treasure Pagoda Stele copybook in our training dataset when building the context-sensitive character detector.

6.5. *Pruning Script and Copybook*. We conduct experiments using the regular script, a standard writing style recommended for all Chinese calligraphy beginners. We ask our participants to use an ink brush to copy the characters from the Multi-Treasure Pagoda Stele copybook, which is one of the classical copybooks for practicing the regular script. This book consists of over 2,500 traditional Chinese characters and we evaluate WiCG using the first 200 characters. Figure 13 shows an example page of the Multi-Treasure Pagoda Stele copybook used in our experiments.

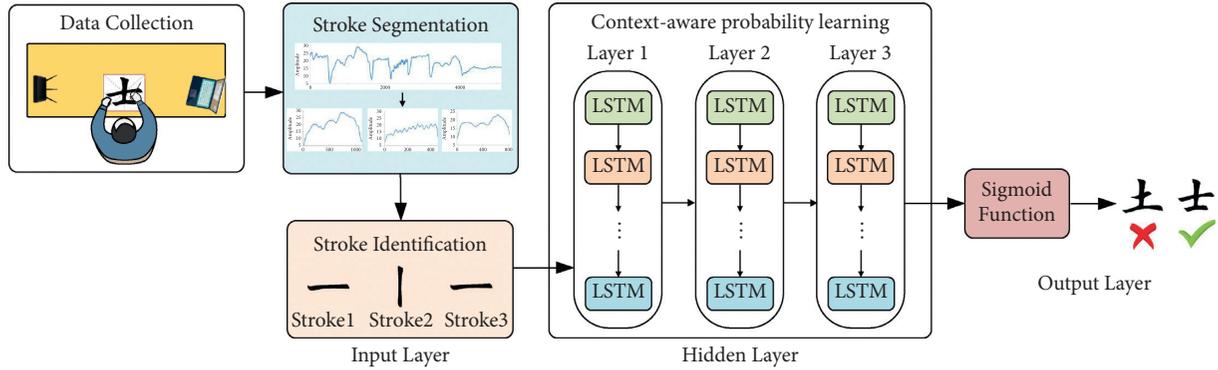


FIGURE 11: Architecture of context-aware character detector.

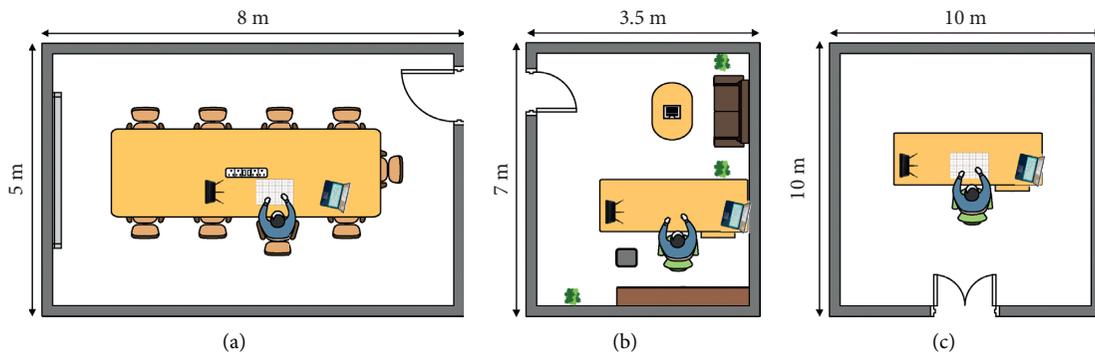


FIGURE 12: System setup and experimental environments. (a) Meeting room. (b) Office. (c) Hall.



FIGURE 13: An example page of the copybook.

7. Experimental Results

7.1. Microbenchmarks. We start with two benchmark experiments to validate the effectiveness of our proposed methods.

7.1.1. Character and Stroke Segmentation. In this experiment, we first investigate the number of strokes of 200 characters. Figure 14 shows the frequency of Chinese characters with different numbers of strokes. We can see that the Chinese characters with the number of strokes between 5 and 10 are the most. Specifically, among the 200 Chinese characters, the proportion of characters with 2–4 strokes, 5–10 strokes, and 11–19 strokes are 13.5%, 62.5%, and 24% respectively. Therefore, we divide the characters used in our evaluation into three groups according to the number of strokes of Chinese characters: *simple* (characters that have less than five strokes), *medium* (characters that have 5–10 strokes), and *complex* (characters that have more than ten strokes).

The segmentation accuracy of 200 characters and corresponding strokes is shown in Figure 15. The complexity of Chinese characters is different, and the segmentation results are also different. The segmentation accuracy rates of simple, medium, and complex Chinese characters are 99%, 98%, and 91%. The accuracy of segmentation of strokes corresponding to a single Chinese character reached 98%, 90%, and 76.5%,

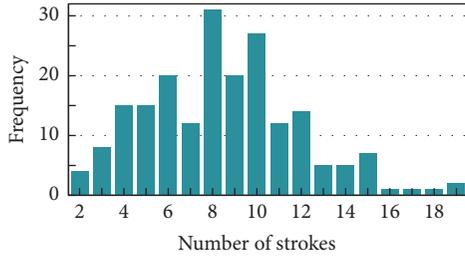


FIGURE 14: Distribution of characters.

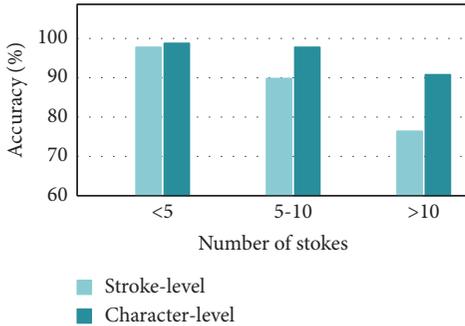


FIGURE 15: Segmentation accuracy.

respectively. We can see that the number of strokes has little effect on character segmentation, but it has a more significant impact on the stroke segmentation in Chinese characters. The more the stroke number is, the lower the accuracy of character segmentation is.

7.1.2. Data Calibration. In order to evaluate whether our calibration method is effective, we collected CSI data of 32 basic strokes five times in each of the six positions at the top, bottom, left, right, and the intersection of the mats. In addition, participants were asked to write at different writing speeds and font sizes. The purpose is to eliminate the difference between different positions, speeds, and font sizes. The conversion between test data and training can be carried out between standard data, and the transfer between different positions in the mask can be learned through BP neural network. The font size and speed transition function parameters between each position are used to calibrate between test data and training data.

Through experiments, we can conclude that the CSI amplitude of strokes with close writing positions (1-2 cm) is the most similar, and the writing data of strokes with a shorter speed time difference (1-2 seconds) are the most similar. Before calibration, the recognition rate of strokes and Chinese characters of different users or different environments is low, less than 20%. After calibration, the recognition and intensive reading are greatly improved. Specifically, even if the environment and the user are changed, the identification accuracy of strokes and Chinese characters can reach 81% and 74.5%, respectively.

7.2. Overall Performance

7.2.1. Character Recognition Accuracy. In order to evaluate the overall performance of character recognition, we asked participant to write the first 200 characters of Multi-Treasure Pagoda Stele copybook 6 times according to the standard of the stroke order. We conduct this experiment using the data collected from the 5 participants. Figure 16 shows the accuracies in three different environments. We can see that WiCG can achieve an average accuracy of 85% for character recognition.

In rare cases, WiCG fails in detecting some of the characters written by users. Close examination of the resulting CSI patterns revealed that these characters were predominantly written in a running-script-like style; i.e., the users did not pause in between the characters which makes stroke segmentation harder. This problem can be mitigated to a large degree by augmenting the model with stroke combinations integrating consecutive strokes into a single gesture. The performance of WiCG can also be improved by using the data collected in the end-user environment to continuously update the identification model, so that the model can adapt to the user's writing behavior and the environment over time.

Overall, WiCG can automatically detect over 80% of the written characters and identify all incorrectly written characters seen in the experiment. This experimental result suggests that WiCG is useful in helping calligraphy learners to detect characters that are written with the wrong stroke order.

7.2.2. 32 Basic Strokes Identification Accuracy. In this experiment, we first investigate the frequency of strokes for 200 characters used in our experiment and the results are shown in Figure 17. The statistics results show that the first four strokes have higher frequencies in 200 Chinese characters. The recognition accuracy of 32 basic strokes is shown in Figure 18. We can see that the recognition accuracies of the first four strokes are less than other basic strokes. That explains why the overall character accuracy cannot achieve higher. In fact, a character contains various strokes so that WiCG will achieve higher accuracy in practical applications.

7.3. Performance under Different Impacts

7.3.1. Impact of Different Number of Strokes. Figure 19(a) shows the recognition accuracies of 200 characters of different complexity (i.e., different number of strokes). The recognition accuracy of simple, medium, and complex Chinese characters is 93%, 89.3%, and 79%, respectively. The reason behind this result is that the complexity of Chinese characters is related to the number of strokes included. Complex Chinese characters contain more strokes. From simple Chinese characters to complex Chinese characters, the stroke number increases, which leads to an increase in the sampling length of CSI data of Chinese

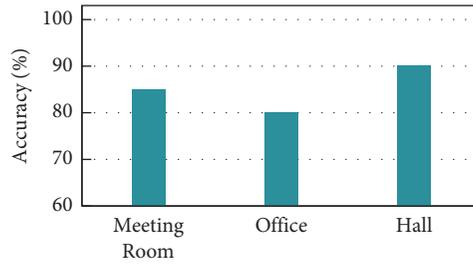


FIGURE 16: Character recognition accuracy.

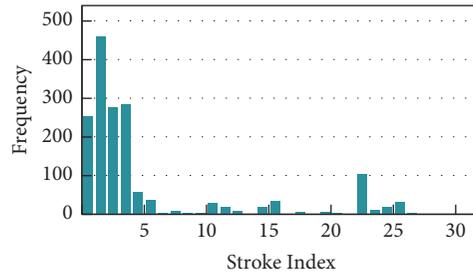


FIGURE 17: Distribution of strokes for 200 characters.

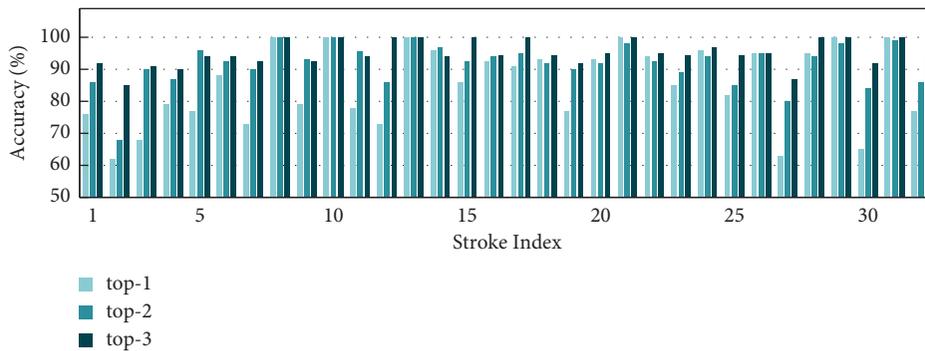


FIGURE 18: The performance of recognizing basic strokes.

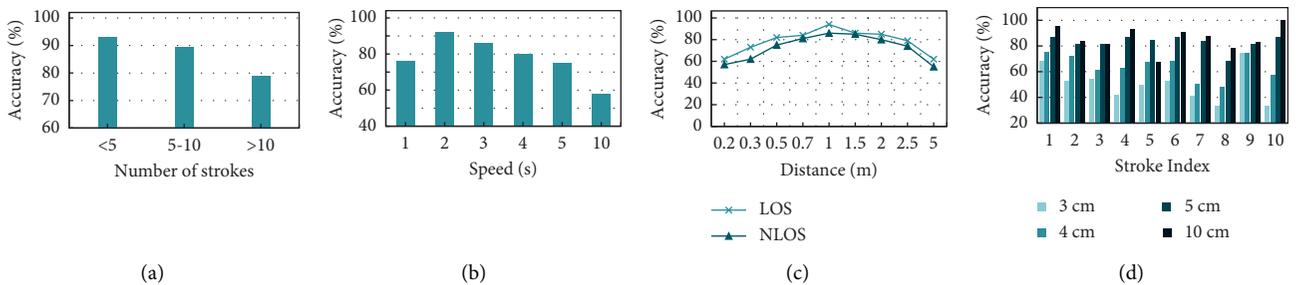


FIGURE 19: Recognition accuracy under different impacts. (a) Stroke number. (b) Writing speed. (c) Distance between devices. (d) Font size.

character and a decrease in the accuracy of Chinese character recognition.

7.3.2. *Impact of Different Writing Speeds.* In order to evaluate the influence of different writing speeds on stroke

recognition, we performed experiments with seven different writing speeds of 32 basic strokes. As shown in Figure 19(b), the influence of different writing speeds for stroke recognition is different. When the stroke writing speed is 2 seconds, the average recognition accuracy is about 93%. When the speed is too fast, the sampling points are less, the

speed is too slow, and the redundancy effect of the sampled data and the change effect of the signal data are not obvious. Therefore, the writing speed needs to be controlled within a reasonable range.

7.3.3. Impact of Device Distances and Positions. In this experiment, we evaluate how the distance and position between the two wireless devices affect the accuracy of our approach. To do so, we vary the distance between the two wireless devices. We evaluate our approach with different distance settings between 0.2 m and 5 m. For each distance setting, we report the accuracy under two scenarios: Line-of-Sight (LOS) and Non-Line-of-Sight (NLOS). In the LOS scenario, the user's hand would roughly align with the middle line between the two wireless devices, while in the NLOS scenario, the user's hand does not align with the middle line of the wireless devices. This experiment was performed in the meeting room where we invited 5 participants to write down the 32 basic strokes.

Figure 19(c) gives the stroke identification accuracy under different evaluation settings and scenarios. As can be seen from the diagram, the wireless device distance has little impact on accuracy. This means that the user can adjust the wireless device distance as needed, e.g., according to the table and paper sizes, without influencing the accuracy too much in many writing environments. Furthermore, our approach works better in the LOS scenario because this scenario allows us to obtain a more discriminative CSI pattern for stroke identification. However, the difference between the LOS and NLOS scenarios is not significant. Therefore, we do not make assumptions on where the user places the hands during writing.

7.3.4. Impact of Different Font Sizes. The font size also affects the accuracy of recognition. After preliminary experiments, we found that the strokes with fonts of 1 cm and 2 cm are very small and write hard, even indistinguishable by the naked eye. Therefore, we only consider a slightly larger font size. In this experiment, we invite participants to write with font sizes of 3 cm, 4 cm, 5 cm, and 10 cm. We use specific size mats to control the font size. We use the 10 most common strokes for evaluation experiments as shown in Figure 17. Figure 19(d) shows the recognition results of 10 strokes in different font sizes. We can see that the average recognition accuracies are 50.4%, 64.1%, 83.1%, and 86.3% when the font sizes are 3 cm, 4 cm, 5 cm, and 10 cm. By checking the original CSI data with font sizes of 3 cm and 4 cm, we observe that the data change is not obvious enough to achieve a satisfactory classification effect. The possible reason is that the magnitude of the user's writing action is small when the written font is small, and WiCG cannot capture more fine-grained features. Therefore, we recommend that users write with a font size of not less than 5 cm.

7.3.5. Alternative Stroke Identification Models. Table 2 gives the top-3 stroke prediction accuracy for characters written in the meeting room (averaged across users) of various

alternative classification techniques and our RF model. The alternative models were built using the same features and training data. Due to the high-quality features, all classifiers are highly accurate in predicting stroke. We choose RF because its accuracy is comparable to alternative techniques but can avoid overfitting [37].

7.3.6. Comparison with Prior Works. As Table 3 shows, compared with five existing systems, WiCG are available without additional hardware facilities, special hardware, or modification of communication protocols. In addition, users can use the system quickly without any training, and users in writing do not have restrictions on the gesture speed and other settings in writing gestures. The overall character identification of WiCG is higher, reaching 91%. These advantages of our system make it possible to deploy and use it in the real environment.

7.3.7. Comparison of Different Sensing Techniques. The four different ways of recognizing calligraphy and considered in our evaluation are shown in Figure 20, which are video-based, sensor-based, WiFi-based, and RFID-based technologies. Under the same scenario deployment, users are asked to write the same Chinese character.

Video-Based: Figure 21(a) is a sketch of the result of motion trajectory tracking based on video, which can roughly get the writing trajectory of Chinese character. The calligraphy writing process is dynamic, but the camera itself cannot dynamically change its distance, position, and angle, so the tracking result is poor.

Sensor-Based: Figure 21(b) is the result of tracing the movement of the Chinese character with the sensor on the wrist. Although trajectories can be obtained, the character cannot be recognized. This result occurs because the position accuracy of sensor tracking and positioning on mobile devices is limited at millimeter level, and the recognition accuracy is low in a small range and distance. So the sensor-based method is not applicable to the recognition of fine-grained calligraphy movement, and it also needs to improve the positioning and tracking accuracy of hardware facilities.

WiFi-Based: Figure 21(c) is a CSI amplitude of Chinese character based on WiFi signal action recognition. It can be seen that there are three distinct stroke segments, and the recognition accuracy is about 93%. WiFi-based action recognition is a passive action recognition method. The device is cheap and easy to obtain, and the user does not need to carry any facilities, so it is easy to use.

RFID-Based: Figure 21(d) is a phase waveform of Chinese character written three times under the RFID-based action recognition mode. It can also see three distinct stroke segments, with an average recognition accuracy of about 82%. RFID-based action recognition is a kind of active action recognition, which is

TABLE 2: Stroke recognition accuracy per model.

Classifier	Accuracy (%)	Classifier	Accuracy (%)
Naive bayes	45	SVM	54.68
MLP	80.1	KNN	38.4
Decision tree	79.2	ANN	72.3
RF	86.7		

TABLE 3: Comparing WiCG against the previous five state of the arts.

System	FEMO [38]	WiHear [7]	WiGest [3]	Wisee [39]	Wikey [24]	WiCG
Signal	RFID	RSS	SDR	SDR	CSI	CSI
Require specialized hardware	No	No	Yes	Yes	No	No
Training	Yes	No	Yes	No	Yes	No
Predefined actions	Yes	Yes	Yes	Yes	Yes	No
Accuracy	90.4%	96%	91%	94%	93.5%	91%

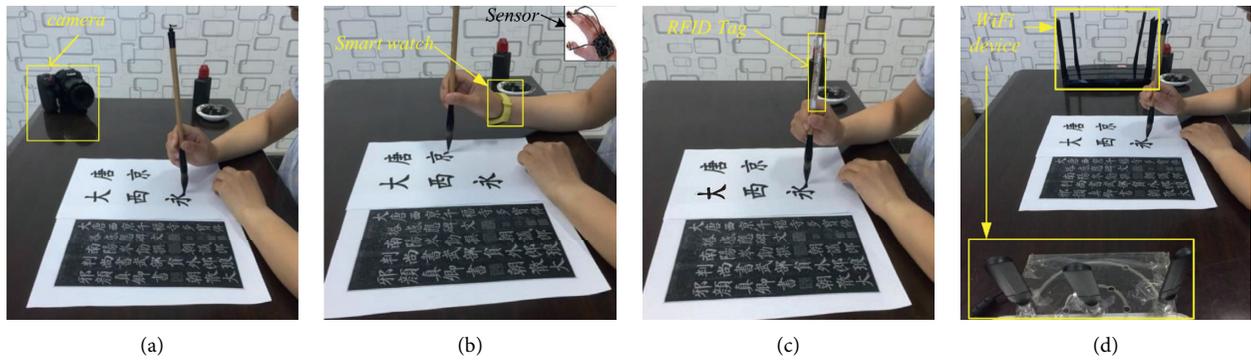


FIGURE 20: Comparison with different sensing approaches. (a) Video-based. (b) Sensor-based. (c) WiFi-based. (d) RFID-based.

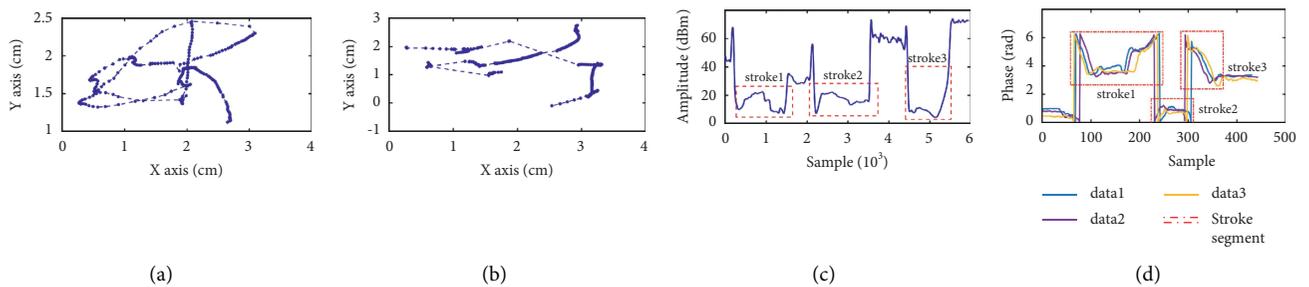


FIGURE 21: The results of four recognition methods for the Chinese character. (a) Video-based sensing. (b) Sensor-based sensing. (c) WiFi-based sensing. (d) RFID-based sensing.

vulnerable to multipath influence in the process of use. Moreover, the reader is expensive, and putting tags on the brush will affect the user experience.

7.4. User Experience Study. We have also performed a user evaluation to assess user-friendliness and overall user experience in four different ways (see Figure 20) of recognizing calligraphy through anonymous questionnaires. The participants in the questionnaire covered students with different

degrees of doctoral, master, and undergraduate students, including 50 females and 70 males. Nearly half of all participants claimed to have an interest in calligraphy. The questionnaire adopts a 10-point Likert scale to assess the user-friendliness of different interactions. In total, indicators in 3 aspects were considered:

- (i) Subjective feelings of user experience: Ease of use, satisfaction, visual perception, and willingness to recommend to others.

- (ii) Task completion degree: Indicators for user experience behavior, such as efficiency effects and error rates.
- (iii) User engagement: Comprehensive rating indicator for the depth of users' participation in the use process, usually the frequency, intensity, or depth of interaction used during a period of time.

The returned questionnaire result processing includes weighting summation statistical scores for the indicators of the four methods and then sorting by the interval. The division of 8 or more and less than ten is *A*, the division of 6 or more and less than eight is *B*, the division of three or more and 6 or less is *C*, and the division of greater than or equal to zero and less than three is *D*. With a total of four intervals, the highest user-friendliness is *A*, followed by *B* and *C* and the worst is *D*.

The results of 120 anonymous questionnaires are shown in Table 4. It can be seen that the user experience of *A*-level based on WiFi method accounts for 49.17%, which is the highest level compared with other interactive methods. Note that since the questionnaire did not involve any interactions with an actual system, these results should be considered as indicators of acceptability of the technology and prior perceptions of the user.

8. Discussions

WiCG is an attempt to exploit WiFi for fine-grained gesture recognition. Naturally, there is room for further work and improvements. We discuss a few points here.

8.1. Targeting Other Application Domains. In this paper, we considered Chinese calligraphy as our use case. However, many of the techniques developed in WiCG are generally applicable and can be used for other gesture recognition tasks. For example, the techniques for recognizing fine-grained hand movements can be used to understand how a sketch is drawn, by tracking what line segments are drawn and at which order. Currently, doing this requires the user to use a tablet, but many users prefer to use traditional tools like pens and papers. Our techniques for modeling the stroke order can also be useful for human-computer interactions to support more complex hand gesture recognition. Prior work has demonstrated the great potential for such applications [40].

8.2. Domain Adaptation. WiCG exploited the context of the written characters to improve the accuracy of character recognition (Section 5.3.2). The model is trained from a corpus of Chinese literature. This model is domain-specific, but the concept can inspire other application domains. For example, a blue plan for building structures often consists of typical building blocks; by learning how these building blocks are used together, one can learn a context-aware model to improve the accuracy for recognizing what is most likely to be drawn. We note that a different machine learning

TABLE 4: User-friendly study with different sensing approaches.

Interactive method	A (%)	B (%)	C (%)	D (%)
Video-based	9.17	25	20.83	45
Sensor-based	13.33	37.5	20	29.17
RFID-based	10	40	15.83	34.17
WiFi-based	49.17	37.5	13.33	0

model might have to be used depending on the amount of available data.

8.3. Cross-Site Recognition. WiCG employs a statistical method to learn a mapping function (Section 5.1.3) that allows adapting to different user environments. This strategy can be applied to a wider context of wireless sensing to allow the sensing models to adapt to different user environments without expensive model retraining [29]. Furthermore, our automatic approach for feature selection (Section 5.2) can be directly applied to other learning-based sensing tasks.

8.4. Subject Independent. Traditional gesture recognition systems separate between subject-dependent and subject independent recognition, i.e., whether there is information available about the person performing the gesture. Most existing WiFi-based gesture recognition work cannot eliminate the differences between different users and hence are subject independent. WiCG uses statistical models to allow the model to adapt to different environments and user behaviors (font sizes, writing habits, etc.), but it requires a small amount of training data to update a previously trained model (Section 5.1.3); i.e., the adaptation allows us to customize a model to support also subject-dependent performance. Our future work will look into how to eliminate the need for new training data. This could be achieved by, e.g., combining analytical models and a learning-based approach to use the analytical model when the learned model is out-of-date.

8.5. Writing Position Tracking. Commercial CSI transceivers are currently used to collect position-based CSI information in vertical and horizontal directions. As discussed in Section 7, even under the same experimental setup, the CSI information acquired from the commercial wireless device varies depending on the environment, distance, etc. Therefore, to use CSI measurements for sensing applications, current systems need frequent calibration, which is unpractical. As the phase information acquired from the commercial wireless device is inaccurate and difficult to calibrate, the tracking granularity of fine-grained calligraphy actions is too rough, and the centimeter-level positioning and tracking of commercial devices have not yet been achieved. Therefore, the position-based calligraphy movement stroke tracking will be further explored in future calligraphy movement recognition. There will also be new applications, such as the standardized detection of calligraphy copying [28].

We will leave the issues of improving the compatibility of Intel 5300 NIC with a wider range of mobile devices to our future work.

9. Conclusions

This paper has presented WiCG, a prototyping wireless sensing system for tracking fine-grained gestures. WiCG is designed to track and capture the hand gesture movements and use the information to identify fine-grained stroke order. To have a concrete application domain, we use Chinese calligraphy as a case study. WiCG firstly tracks the user's hand movements during the writing and then uses the tracked information to identify what character has been written and if the character is written in the correct stroke order. Specifically, WiCG achieves this goal by tracking how the channel state information of the WiFi signal is affected by a hand movement. We combine statistical methods and machine learning techniques to develop a set of methods and analyses to overcome a range of challenges brought by this new application context.

We evaluate WiCG by conducting extensive experiments and user studies. Experimental results show that WiCG can successfully identify about 80% of the written characters and detect all incorrectly written characters seen in our experiments. Given that WiCG only relies on commercial wireless and computing devices and avoids many of the privacy issues a video-based approach would have, it is highly applicable to many user scenarios. We believe WiCG represents an important first step towards exploiting wireless sensing for fine-grained gesture movements. We hope our study can provide compelling evidence to encourage further work in this promising research area.

Data Availability

The data used to support the findings of the study can be obtained according to this link: <https://github.com/NISL-twty/WICG>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) (Grant Agreement Nos. 61972314, 61872294, and 62102315), in part by the International Cooperation Project of Shaanxi Province (2020KWZ-013, 2019KW-009, and 2021KW-04), and in part by the Shaanxi Province Key R&D Projects (2018SF-369).

References

- [1] R. M. Krauss, Y. Chen, and P. Chawla, "Nonverbal behavior and nonverbal communication: what do conversational hand gestures tell us?" *Advances in Experimental Social Psychology*, vol. 28, pp. 389–450, 1996.
- [2] Y. Chen, W. Wei, and W. Xiao, "Human-computer interaction control of snake-like robot based on gesture recognition," in *Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering*, Shenzhen, China, July 2019.
- [3] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: a ubiquitous wifi-based gesture recognition system," *Computer Communications*, pp. 1472–1480, 2015.
- [4] H. Kong, L. Lu, J. Yu, Y. Chen, L. Kong, and M. Li, "Fingerpass: finger gesture-based continuous user authentication for smart homes using commodity wifi," in *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 201–210, Catania, Italy, July 2019.
- [5] C. Li, M. Liu, and Z. Cao, "Wihf: enable user identified gesture recognition with wifi," in *Proceedings of the IEEE INFOCOM 2020—IEEE Conference on Computer Communications*, pp. 586–595, Toronto, Canada, July 2020.
- [6] M. Shahzad and S. Zhang, "Augmenting user identification with wifi based gesture recognition 2," 2018.
- [7] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We can hear you with wi-fi!" in *Proceedings of the International Conference on Mobile Computing and Networking*, pp. 593–604, Maui, HI, USA, September 2014.
- [8] C. Dian, D. Wang, Q. Zhang, R. Zhao, and Y. Yu, "Towards domain-independent complex and fine-grained gesture recognition with rfid," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, 2020.
- [9] K. C. Lim, S. H. Sin, C. W. Lee et al., "Video-based skeletal feature extraction for hand gesture recognition," in *Proceedings of the 4th International Conference on Machine Learning and Soft Computing*, pp. 108–112, Haiphong City, Viet Nam, January 2020.
- [10] M. A. Rahim, J. Shin, and M. R. Islam, "Human-machine interaction based on hand gesture recognition using skeleton information of kinect sensor," in *Proceedings of the 3rd International Conference on Applications in Information Technology*, pp. 75–79, Aizu-Wakamatsu Japan, November 2018.
- [11] Y. Liu, S. Zhang, and M. Gowda, "When video meets inertial sensors: zero-shot domain adaptation for finger motion analytics with inertial sensors," in *Proceedings of the International Conference on Internet-of-Things Design and Implementation*, pp. 182–194, Charlottesville, VA, USA, May 2021.
- [12] A. Tsagaris and D. Trigkas, "Mobile gesture recognition," in *Proceedings of the International Conference on Machine Vision and Applications*, pp. 13–17, Singapore, April 2018.
- [13] A. Huang, D. Wang, R. Zhao, and Q. Zhang, "Au-id: automatic user identification and authentication through the motions captured from sequential human activities using rfid," *Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies*, vol. 3, no. 2, 2019.
- [14] H. Chao, W. Fenhua, and Z. Ran, "Sign language recognition based on cbam-resnet," in *Proceedings of the 2019 International Conference on Artificial Intelligence and Advanced Manufacturing*, Dublin, Ireland, October 2019.
- [15] J. L. Pérez-Medina and K. B. Jimenes-Vargas, "Recognizing a sketched task model by multi-stroke gesture recognition," in *Proceedings of the 12th ACM SIGCHI Symposium on Engineering Interactive Computing Systems*, Sophia Antipolis, France, June 2020.
- [16] A. Bogaers, Z. Yumak, and A. Volk, "Music-driven animation generation of expressive musical gestures," in *Proceedings of the Companion Publication of the 2020 International Conference on Multimodal Interaction*, pp. 22–26, Virtual Event, The Netherlands, October 2020.
- [17] Y. Lin, Y. Gao, B. Li, and W. Dong, "Revisiting indoor intrusion detection with WiFi signals: do not panic over a pet!"

- IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 10437–10449, 2020.
- [18] M. Zhou, Y. Lin, N. Zhao, Q. Jiang, X. Yang, and Z. Tian, “Indoor wlan intelligent target intrusion sensing using ray-aided generative adversarial network,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 1, pp. 61–73, 2020.
- [19] R. H. Venkatnarayan, G. Page, and M. Shahzad, “Multi-user gesture recognition using wifi,” in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 401–413, Munich Germany, June 2018.
- [20] Y. Zeng, Z. Liu, D. Wu, J. Liu, J. Zhang, and D. Zhang, “A multi-person respiration monitoring system using cots wifi devices,” in *Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers*, pp. 195–198, New York, NY, USA, September 2020.
- [21] Y. Ma, S. Arshad, S. Muniraju et al., “Location- and person-independent activity recognition with wifi, deep neural networks, and reinforcement learning,” *ACM Transactions on Internet Technology*, vol. 2, 2021.
- [22] H. Wang, D. Zhang, J. Ma et al., “Human respiration detection with commodity wifi devices: do user location and body orientation matter?” in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 25–36, Heidelberg Germany, September 2016.
- [23] P. Melgarejo, X. Zhang, P. Ramanathan, and D. Chu, “Leveraging directional antenna capabilities for fine-grained gesture recognition,” in *Proceedings of the Ubicomp '14*, Seattle, WA, USA, September 2014.
- [24] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, “Keystroke recognition using wifi signals,” in *Proceedings of the International Conference on Mobile Computing and Networking*, pp. 90–102, Paris, France, September 2015.
- [25] H. Li, W. Yang, J. Wang, Y. Xu, and L. Huang, “Wifinger: talk to your smart devices with finger-grained gesture,” in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 250–261, Heidelberg Germany, September 2016.
- [26] M. Li, Y. Meng, J. Liu et al., “When csi meets public wifi: inferring your mobile phone password via wifi signals,” in *Proceedings of the ACM Sigsac Conference*, pp. 1068–1079, Vienna, Austria, October 2016.
- [27] B. Hong, B. Kim, J. D. Kim, and J. Koo, “Performance analysis of Chinese cursive character recognition based on convolutional neural network,” in *Proceedings of the Conference on Research in Adaptive and Convergent Systems*, pp. 125–127, Chongqing China, September 2019.
- [28] L. Chang, J. Xiong, J. Wang et al., “Rf-copybook: a millimeter level calligraphy copybook based on commodity rfid,” *Proceedings of the ACM on Interactive Mobile Wearable and Ubiquitous Technologies*, vol. 1, no. 4, pp. 1–19, 2018.
- [29] J. Zhang, Z. Tang, M. Li, D. Fang, P. Nurmi, and Z. Wang, “Crosssense: towards cross-site and large-scale wifi sensing,” in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, New Delhi, India, October–November 2018.
- [30] G. Moroni, W. P. Syam, and S. Petró, “Performance improvement for optimization of the non-linear geometric fitting problem in manufacturing metrology,” *Measurement Science and Technology*, vol. 25, no. 8, Article ID 085008, 2014.
- [31] Y. Zeng, P. H. Pathak, and P. Mohapatra, “Wiwho: wifi-based person identification in smart spaces,” in *Proceedings of the International Conference on Information Processing in Sensor Networks*, p. 4, Vienna, Austria, April 2016.
- [32] T. K. Ho, “The random subspace method for constructing decision forests,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.
- [33] R. Jozefowicz, W. Zaremba, and I. Sutskever, “An empirical exploration of recurrent network architectures,” in *Proceedings of the 32nd International Conference on International Conference on Machine Learning—Volume 37*, pp. 2342–2350, JMLR.org, Lille, France, July 2015.
- [34] L. Aina, K. Gulordava, and G. Boleda, “Putting words in context: lstm language models and lexical ambiguity,” 2019.
- [35] J. Zeng, X. Ma, and K. Zhou, “Enhancing attention-based lstm with position context for aspect-level sentiment classification,” *IEEE Access*, vol. 7, pp. 20462–20471, 2019.
- [36] J. Zhou, Y. Lu, H.-N. Dai, H. Wang, and H. Xiao, “Sentiment analysis of Chinese microblog based on stacked bidirectional lstm,” *IEEE Access*, vol. 7, pp. 38856–38866, 2019.
- [37] A. Liaw and M. Wiener, “Classification and regression by randomforest,” *R News*, vol. 23, 2002.
- [38] H. Ding, L. Shangguan, Z. Yang et al., “Femo: a platform for free-weight exercise monitoring with rfids,” in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 141–154, Seoul, South Korea, November 2015.
- [39] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, “Whole-home gesture recognition using wireless signals,” in *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking—MobiCom'13. Gesture Recognition Using Wireless Signals*, pp. 27–38, Miami, FL, USA, September–October 2013.
- [40] L. Shangguan, Z. Zhou, and K. Jamieson, “Enabling gesture-based interactions with objects,” in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*, pp. 239–251, ACM, Niagara Falls, NY, USA, June 2017.