# A Device-free Number Gesture Recognition Approach Based on Deep Learning

Qizhen Zhou[1], Jianchun Xing[1], Juelong Li[1,2], Qiliang Yang[1,3]
1.College of Defense Engineering, PLA University of Science and Technology, Nanjing Jiangsu 210007, China
2. Technical Management Office of Naval Defense Engineering, Beijing 100841, China
3. Research Center of Building Information Modeling, Tsinghua University, Beijing 100084, China
(E-mail: zhouqizhen2016@163.com; {xjc, yql}@893.com.cn; lijuelong@126.com)

***Abstract*—Number gestures play essential parts in our daily communication and have attracted academic interests in developing Human-Computer Interface. In this paper, we resort to the fine-grained Channel State Information (CSI) in the 802.11n standard to recognize number gestures. The intuition is that certain gestures can affect wireless environment in a specific formation and thus generate unique features. Unfortunately, the majority of CSI-based technologies only extracted coarse grained features to recognize macro-movements. Besides, it can be time-consuming to select the most discriminative feature as salient evidence. In this paper, we present a device-free number gesture recognition approach based on deep learning, named DeNum. First, we explore the sensibility of both the amplitude and phase information of de-noised CSI values to action transitions. Then the amplitude difference is utilized to detect the finishing points of actions through multiple sliding windows. To extract discriminative features from both the amplitude and phase information over three antennas, a 4-layer deep learning model is adopted after obtaining number gesture information. Finally, a Support Vector Machine (SVM) algorithm is applied for gesture classification. We conduct extensive experiments on commercial Wi-Fi devices with different experimental parameters. Experimental results demonstrate the presented approach can achieve the average accuracy of 94% in current office scenario.**

***Keywords: Number gesture recognition; Channel State Information; Deep learning; Wi-Fi***

## I. INTRODUCTION

With the rapid development of computer technologies, gesture recognition has attracted great attention in promoting Human-Computer Interface (HCI). As the most frequently used gestures in our daily life, number gestures seem to have broad application prospects due to their simplicity and intuitiveness (e.g., in a smart home, users can call telephone number, regulate the indoor temperature and enter password to control household equipments just by performing simple gestures). Number gesture recognition does set our hands free from handheld devices.

Various proposed techniques from sensor-based to vision-based methods have been successfully applied in a certain scenario. Typical sensor-based technologies utilize hands' position and fingers' bending angles to detect current gesture by wearing a data glove [1, 2]. However, it is unrealistic for users to wear such a glove all day long. The other vision-based solutions seem to be better choices for a smart home [3]. What they need are installed high-resolution cameras to capture a series of pictures. Unfortunately, the major problem is their inherent requirement for light conditions. It cannot ensure the performance of these solutions in weak illumination condition. Moreover, the privacy issues matter as well.

Due to such limitations of above techniques, we turn to widespread and cheap Wi-Fi devices for help, considering emerging Wi-Fi technologies [4-6] have been successfully applied in sensing areas. Stimulated by the observation that CSI is a more fine-grained information compared with Received Signal Strength (RSS), a large part of emerging technologies resort to extract physical layer Channel State Information (CSI) by modifying the device driver (e.g., Intel IWL 5300 card) based on the literary [7]. Previous works [8-11] have exploited the CSI properties to identify human gesture activities. The reason is that different activities will make relatively unique effects on received Wi-Fi signals. However, these feature extraction methods cannot be directly applied to recognize number gestures, because slight fingers' movements may affect the wireless environment in a similar way and thus result in similar patterns over time. Therefore, we need to further extract robust and feasible features from CSI so as to efficiently recognize slight number gestures.

In this paper, we present a robust and accurate **de**vice-free **num**ber gesture recognition approach (DeNum), by fully utilizing the extracted amplitude and phase features over three antennas based on deep learning. We make three main contributions as follows:

1) We creatively recognize ten different number gestures by utilizing a 4-layer deep learning model to extract both the CSI amplitude and phase features on commodity Wi-Fi devices.

2) We further explore the sensibility of both the amplitude and phase information in the face of action transitions, and then resort to amplitude difference through certain multiple sliding windows to detect the finishing points of action transitions.

3) Extensive experiments are conducted on commodity Wi-Fi devices and our proposed scheme is tested with different parameters. Experimental results demonstrate the proposed approach can achieve an average accuracy of 94%.

The rest of this paper is structured as follows. Section II reviews related work. Section III introduces the concept of CSI and predefines ten number gestures. We present the details and methodologies of our approach in Section IV.

978-1-5090-4840-3/16 $31.00 © 2016 IEEE
DOI 10.1109/CIS.2016.21

Then we evaluate the performance and compare the accuracy with different parameters in Section V. We discuss the limitations and potential in Section VI and draw the conclusion in Section VII.

## II. Related Work

In this section, we will briefly introduce the state-of-art works from three perspectives: number gesture recognition, CSI-based activity recognition and deep learning based feature extraction.

### A. Number Gesture Recognition

Number gestures are widely used in our daily communication and Human-Computer Interaction (HCI) areas due to its simplicity and intuitiveness. Roughly, number gesture recognition approaches can be classified into two broad categories: sensor-based approaches and vision-based approaches. Sensor-based approaches always require subjects to wear customized data gloves in order to acquire necessary experimental data. Wu et al. firstly presented the process of building a sign language approach to recognize simple words [1]. Kumar et al. used data gloves to capture current position of arms and record the angle of adjacent fingers, leveraging K-Nearest Neighbors (K-NN) algorithm to recognize the gestures [2]. The major problem with sensor-based approaches lies in the fact that data gloves are inconvenient to carry with. These approaches can only work or operate well on the premise that all these devices must be worn appropriately in the period of motions.

Vision-based approaches can be generally divided into four main steps, including the segmentation step, orientation detection step, feature extraction step and classification step. Panwar and Mehra installed a high-resolution camera to take successive motion pictures as input to their proposed algorithm, which was independent of user characteristics [3]. However, the vision-based methods fail to effectively recognize hand gestures in a dim or dark environment. In other words, these methods extremely require for light conditions (e.g., illuminant color and incident angle of light). Besides, they may lead to privacy problems if cameras are installed in some special occasions, like dressing room and bathroom.

### B. CSI-based Activity Recognition

In recent years, since CSI can be exported from off-the-shelf Network Interface Cards (NICs), CSI-based schemes have been widely applied in personnel localization [12, 13], crowd counting [14], and activity recognition [8-11]. Wu et al. proposed a fingerprinting system for localization by building a propagation model and taking advantage of frequency diversity and spatial diversity of CSI [12]. Chapre et al. improved the accuracy of localization by creating a novel signature of CSI based on magnitude and phase difference and using Multiple Input Multiple Output (MIMO) system [13]. Xi et al. proposed a device-free crowd counting system, called Electronic Frog Eye, which indicated the relationship between the number of people and variation of CSI values [14]. Wang et al. aimed to detect the fall activities by using statistical features extracted from both the amplitude and phase information through multiple optimal sliding windows [8]. Qian et al. realized passive human movement detection by utilizing the sensitive phase information [10]. Wang et al. proposed a CSI-activity model, studying the correlation between CSI value dynamics and the movement speeds of different body parts [11].

These interesting woks motivated us to explore both the amplitude and phase information of CSI so as to recognize number gestures caused by micro-movements. However, these works just extracted coarse grained features to identify macro-movements (e.g., identifying falling down activities, detecting human presence and counting crowds). Ali et al. employed CSI-waveform to recognize micro-keystrokes because each key generated relatively unique multi-path effects in received CSI samples [9]. Unfortunately, waveform profile can change in shape with days, which means it cannot be taken as salient evidence for robust gesture recognition.

In such case, deep learning methods are more suitable to explore discriminative features from these similar gestures. DeepFi [16] and PhaseFi [17] respectively utilized the amplitude and calibrated phase information of CSI as feature-based fingerprints for accurate indoor localization based on deep learning. DFLAR [18] took the de-noised RSS data as input of deep learning model and developed a device-free localization and an activity recognition approach. Inspired by above works, we expect to extract remarkable features based on deep learning.

## III. Preliminaries

### A. Channel State Information

Owning to [7], Channel State Information (CSI) can be easily obtained by accessing the device driver of NICs, such as Intel IWL 5300. CSI amply depicts the properties of the subcarrier-level channel measurements during propagation. In other words, if a wireless signal is reflected by multipath effect, fading, scattering, and power decay with distance, the variations can be revealed in the CSI values.

In frequency domain, channel model can be defined as

$$\vec{Y} = H\vec{X} + \vec{N} \tag{1}$$

where $\vec{Y}$ and $\vec{X}$ represents the received and transmitted signal vectors, respectively. $\vec{N}$ denotes the additive white Gaussian noise and $H$ denotes the channel matrix which can be estimated from $\vec{Y}$ and $\vec{X}$ .

Since the Intel IWL 5300 NICs implement an Orthogonal Frequency Division Multiplexing (OFDM) system with 48 subcarriers, a modified device driver can only export 30 subcarriers for each of three antennas. Hence, a complex value $H_i$ of $i_{th}$ subcarrier can be defined as:

$$H_i = |H_i| e^{j \sin\theta}, \ i \in [1,30] \tag{2}$$

where $|H_i|$ and $\theta$ denotes the amplitude and phase of $i_{th}$ subcarrier, respectively.

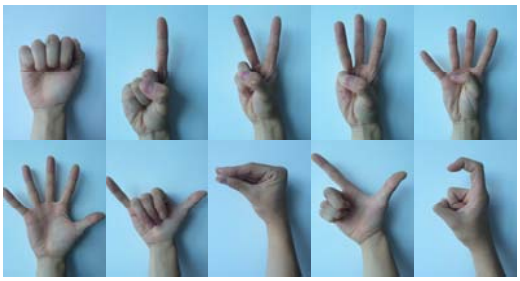

Figure 1. Ten Number Gestures Used in Our Experiments

### B. *Predefine Number Gestures*

Considering the differences existing in gesture expression among individuals, it is necessary to predefine our ten number gestures before an experimental study. Figure 1 shows ten static number gestures used in our experiments. We choose the most widely used number gestures standing for 0 to 9 operated by the right hands. Thus we can avoid semantic ambiguities and improve the accuracy as well as generalizability to some extent. In order to eliminate the effect of arm's movement, we let subjects keep arms still during the experiments.

## IV. METHODOLOGY

In this section, we elaborate the methodology of DeNum which consists of four functional steps: signal preprocessing, gesture extraction, deep learning based feature extraction and SVM classification. As shown in Figure 2, physical layer CSI is taken as the input to correctly depict the properties of the affected wireless environment. We implement a common wireless router as a transmitter and a mini PC with off-the-shelf Intel IWL 5300 NIC as receiver to collect CSI samples. In order to obtain feasible wireless signals, a band-pass filter is adopted to filter out the irrelevant measurement noises and then a linear transformation is utilized to remove the random phase offsets. We use the sensitive amplitude information over multiple sliding windows to detect the finishing points of action transitions and extract static number gesture information for further classification. As the most important part of number gesture recognition, we present a feature extraction method based on deep learning to learn discriminative features from amplitude and phase information over three antennas. The features constitute the input of SVM classifier and finally the proposed approach achieves the goal of gesture recognition. We would present the details of each step in the following sections.

### A. *Signal Preprocessing*

The collected samples provided by off-the-shelf Intel IWL 5300 NIC are extremely noisy. We adopt a band-pass filter first to remove irrelevant noises and then apply a simple linear transformation to mitigate the phase offsets.

1) *Band-pass filtering*

The majority of environment noises (e.g., electronic noises) lie in the high frequency range, and those slight body movements (e.g., chest movements and slight tremble of fingers) fall into extremely low frequency range compared

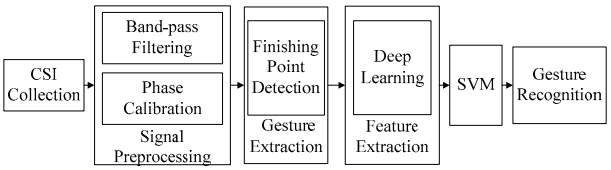


Figure 2. Overview of DeNum

with human common activities [19]. In order to smooth out those random noises with high-frequency and any other irrelevant components, a band-pass filter is a natural choice to be applied in such a case. Based on experimental observations, the frequencies of finger movements with different subjects lie in the range of [0.28Hz, 9.86Hz]. To be conservative, we adopt a band-pass filter with cut-off frequency of 0.25Hz and 10Hz.

2) *Phase calibration*

Motivated by RT-Fall [8], we try to use the state transition of both the amplitude and phase difference variance over a pair of antennas in the time domain to extract the static gestures information from dynamic action transitions. However, even through band-pass filtering, filtered phase information still distributes randomly due to unsynchronized time clock between transmitter and receiver, which make it inapplicable and infeasible in gesture recognition, just as Figure 3 shows. Therefore, we apply a simple effective phase calibration method [17] to mitigate the random phase offsets.

The measured phase $\vartheta_i$ for the $i_{th}$ subcarrier can be written as:

$$\vartheta_i = \theta_i + 2\pi \frac{m_i}{N} \Delta t + \beta + Z \tag{3}$$

where $\theta_i$ is the true phase we need, $\Delta t$ stands for the time lag at the antenna, $\beta$ is an unknown phase offset, $Z$ is the existing measurement noise, $m_i$ denotes the indices of 30 subcarriers ranging from -28 to 28 and the FFT size $N$ equals to 64 in IEEE 802.11n. As we cannot obtain available phase information directly, a linear transformation is implemented to remove the unknown terms $\Delta t$ and $\beta$ .

We define the slope of phase and offset across the entire frequency band as $a$ and $b$ respectively. When the indices of 30 subcarriers are symmetric, we can infer:

$$a = \frac{\vartheta_{30} - \vartheta_1}{m_{30} - m_1}, b = \frac{1}{30} \sum_{j=1}^{30} \vartheta_j \tag{4}$$

Subtracting the linear items $am_i + b$ from the measured $\vartheta_i$ , while the measurement noise $Z$ can be ignored, we can obtain a linear combination of true phase which is given by:

$$\hat{\theta}_i = \vartheta_i - am_i - b = \vartheta_i - \frac{\vartheta_{30} - \vartheta_1}{m_{30} - m_1} m_i - \frac{1}{30} \sum_{j=1}^{30} \vartheta_j \tag{5}$$

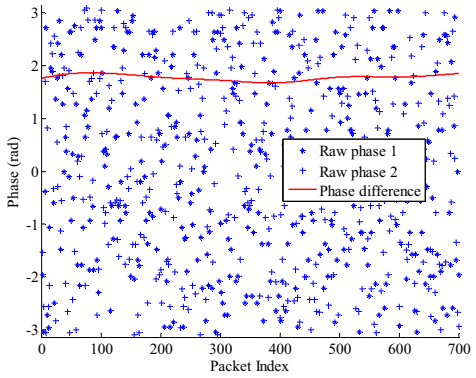


Figure 3. Raw Phase and Calibrated Phase Difference

In Figure 3, calibrated phase difference distributes relatively stable at 1.8 when there is a subject sitting still between the receiver and transmitter, while the raw phase values of antenna 1 and antenna 2 distribute randomly. We choose the 12th subcarrier in the first and second antenna to test the performance of phase calibration. We can observe the proposed phase calibration method does remove the time lags $\Delta t$ and unkown phase offset $\beta$. In the following step, we try to explore the sensitivity of both the amplitude and phase to gesture transitions.

### B. *Number Gestures Extraction*

To extract feasible number gesture information, in step 1, we intend to explore the sensitivity of both the amplitude and phase to gesture transitions and then in step 2 we extract number gesture information after detecting the finishing points of action transitions.

1) *Amplitude vs. Phase*

Through numerous experiments, we infer that the amplitude difference variance can be a robust feature to detect the finishing points of gesture transitions, rather than phase difference variance. We choose four similar activities to validate our assumption. Figure 4 shows both the amplitude and phase difference sliding variance of 4 different activities. We use the difference of amplitude and phase value over antenna 1 and antenna 2 because these information can be affected in a similar way. Furthermore, in order to detect the finishing points, we adopt a sliding window based approach with 5 seconds window size and move over CSI data at an interval of 1 second to obtain sliding variance over packets. As is observed from Figure 4, dynamic gesture transitions lead to obvious fluctuation in amplitude sliding variance, which denotes the state transition of static-dynamic-static. Besides, we can find that different actions result in the varying peak values of fluctuation, from 0.6 to 1.6. However, we fail to come up with the same conclusion in testing sliding phase difference variance, they seem relatively stable and steady around 1.7. Through extensive works, we can conclude that amplitude is more sensitive to finger movements. In other words, amplitude difference is a better base signal to identify the finishing points of gesture transitions in our experimental scenario.

2) *Detect the finishing points*

Inspired by [8], to robustly detect the finishing points, we adopt a threshold-based sliding window method by comparing the sliding mean amplitude difference with empirical thresholds calculated in stable states. We compute the thresholds as follows:

$$\mu^{i}_{stable} + 2\sigma^{i}_{stalbe} + 3\varepsilon^{i}_{stable} \le \delta^{i}_{threshold} \tag{6}$$

where $\mu^{i}_{stable}$, $\sigma^{i}_{stalbe}$ and $\varepsilon^{i}_{stable}$ denotes the mean, normalized standard deviation (STD) and median absolute deviation (MAD) of sliding amplitude difference extracted from $i_{th}$ gesture respectively, $i \in [0,9]$. Considering that different gestures correspond to different thresholds, we take the maximum value from ten thresholds as judging standard so as to correctly detect the finishing points.

Then, instead of extracting features from gesture transitions in a certain window size, we obtain the packets which contain static gesture information after detecting the finishing points. The rationale is that gesture transitions with different initial gestures and speeds may generate different statistic features (e.g., gesture 0 to 9 and gesture 5 to 9, even any other kinds of gestures) and it is time-consuming to store all kinds of gesture transitions. Therefore, we utilize the static CSI information after detecting the finishing points.

### C. *Deep Learning based Feature Extraction*

In order to achieve trade-off between accuracy and labor-saving, we firstly normalize the amplitude and phase values of three antennas and then average them as input data. We take the algorithm proposed by [20, 21], using a deep model with one input layer and three hidden layers to study essential features. Specifically, there are three steps in deep

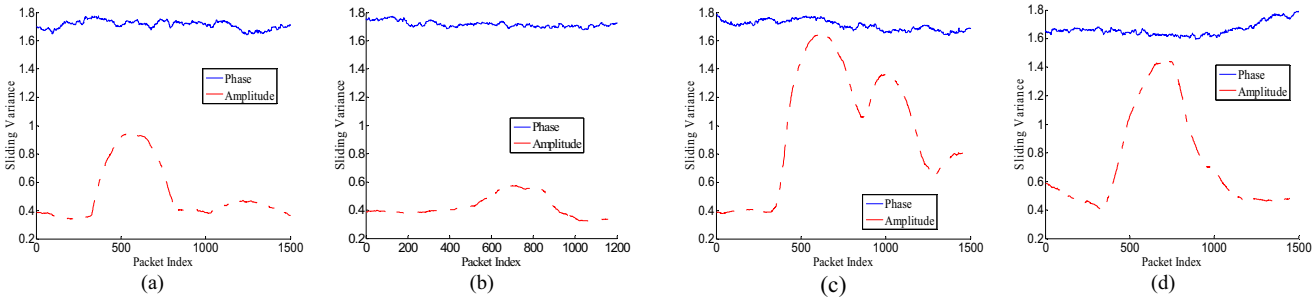


Figure 4. Amplitude Difference Variance and Phase Difference Varience of Four Different Actions : (a) 0 to 1, (b) 0 to 2, (c) 3 to 5, (4) 0 to 9.

learning procedure, i.e., pre-training, unrolling and fine-tuning. We try to obtain the near-optimal weights in the pre-training step. The learned feature activations of one Restricted Boltzmann Machine (RBM) are used as the input data for training the next RBM in the stack[20]. We define $h^0$ as the input layer, $h^1$, $h^2$, $h^3$ and $K_1$, $K_2$, $K_3$ as the first, second，third hidden layer and its hidden units number. $W_1$, $W_2$, $W_3$ as the weight between layers respectively, $V_j^i$ as the $j_{th}$ input data of the $i_{th}$ gesture, $i \in [0,9]$ and $j \in [0,90]$.

To obtain near optimal initial values, we represent the deep network with probabilistic generative model $\Pr(h^0,h^1,h^2,h^3)$ and need to maximize the marginal distribution of input which is given by:

$$\max_{\{W_1,W_2,W_3\}} \sum_{h^1}\sum_{h^2}\sum_{h^3} \Pr(h^0,h^1,h^2,h^3) \tag{7}$$

The probabilistic generative model can be formulated with the joint probability density distribution as:

$$\begin{aligned} &\Pr(h^0,h^1,h^2,h^3)=\Pr(h^0 \mid h^1)\Pr(h^1 \mid h^2)\Pr(h^2,h^3) \\ &=\prod_{j=1}^{90}\Pr(h_i^0 \mid h^1)\prod_{j=1}^{K_1}\Pr(h_i^1 \mid h^2)\Pr(h^2,h^3) \end{aligned} \tag{8}$$

where $\Pr(h^0 \mid h^1), \Pr(h^1 \mid h^2), \Pr(h^2,h^3)$ respectively represents the joint distribution. We adopt the contrastive divergence with one step iteration (CD-1) algorithm to approximate the joint distribution $\Pr(h^2,h^3)$ by sigmoid belief network as

$$\begin{aligned} \Pr(h^{i-1} \mid h^i) &= \prod_{j=1}^{K_{i-1}} \Pr(h_j^{i-1} \mid h^i) \\ &= \prod_{j=1}^{K_{i-1}} \frac{1}{1+\exp(-b_j^{i-1} - \sum_{t=1}^{K_i} W_i^{j,t} h_x^i)} \end{aligned} \tag{9}$$

$$\begin{aligned} \Pr(h^i \mid h^{i-1}) &= \prod_{j=1}^{K_i} \Pr(h_j^i \mid h^{i-1}) \\ &= \prod_{j=1}^{K_i} \frac{1}{1+\exp(-b_j^i - \sum_{t=1}^{K_{i-1}} W_i^{j,t} h_x^{i-1})} \end{aligned} \tag{10}$$

We resort to greedy layer-wise algorithm [21] to train the parameters of all weights layer by layer. First, CD-1 algorithm is used to estimate the initial parameters $b^0, b^1, W_1$, and then parameters $b^1, b^2, W_2$ of the second layer are estimated by conditional probability $\Pr(h^1 \mid h^0)$, while the initial parameters are frozen during the process, and so forth. $b^0, b^1, b^2$ represents the biases of an input layer, hidden layer 1 and hidden layer 2 respectively.

Thus, we could obtain updated parameters by CD-1 algorithm as follows:

$$\begin{cases} \Delta W_i = \lambda(h^{i-1}h^i - \hat{h}^{i-1}\hat{h}^i) \\ \Delta b^i = \lambda(h^i - \hat{h}^i) \\ \Delta b^{i-1} = \lambda(h^{i-1} - \hat{h}^{i-1}) \end{cases} \tag{11}$$

where $\Delta b^i$ denotes the updated biases for unit $i$, $\hat{h}^i$ means the "empirical" distribution sampled from $\Pr(h^{i-1} \mid h^i)$, $\lambda$ is the step size. After the pre-training step, the stacked RBMs are unrolled to create reconstruction data with forward propagation in unrolling step, and the obtained error derivatives can be utilized to adjust weights with back-propagation algorithm in fine-tuning step. We set $K_1, K_2, K_3$ as 200 and only use the output of the last hidden layer as features for effective classification.

### *D. SVM Classification*

To recognize our number gestures correctly, we utilize the Support Vector Machine (SVM) [22], a supervised binary classification algorithm, to separate input data into specific classes by mapping samples into a high dimensional space and constructing an optimal hyperplane. SVM requires a training dataset to build a classification model with labeled features and a test dataset to predict the label with given features. In our experiment, we choose LIBSVM [23] toolbox to create a $k(k\text{-}1)/2$ ($k$ is the number of classes) classification model with the radial basis function (RBF) kernel and then obtain the accuracy value by comparing the predefined labels and predicted labels. To be specific, normalized features were injected into SVM classifier with 0-9 corresponding labels. We adopt a 10-fold cross-validation method to find the optimal parameter of cost and gamma, and leave other parameters as default.

## V. EXPERIMENT EVALUATION

In this section, we first introduce the details of experimental settings, as Figure 5 shows. Then we report what extent our approach performance can achieve. Moreover, we show the accuracy of our approach with different environmental parameters.

### *A. Experiment Setup*

In our experiment, we use a mini PC equipped with an Intel IWL 5300 NIC and three antennas as receiver, operating in Ubuntu 10.04 system. A single antenna TL-WR742N wireless router is implemented, running in the 5 GHz frequency as the transmitter. We utilize a modified driver to collect CSI values from Intel IWL 5300 NIC proposed by Halperin [7].

Extensive experiments have been conducted in an office


Figure 5. Experiment Scenario

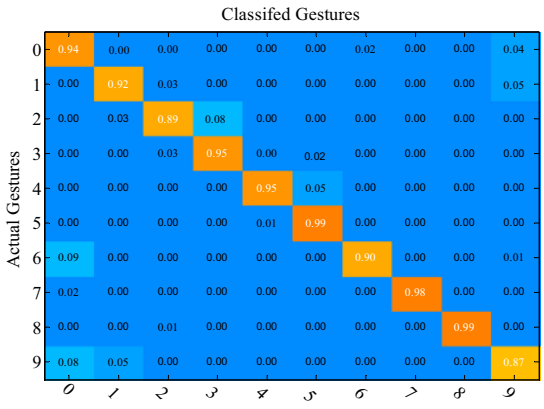


Figure 6. Confusion Matrix of Number Gestures

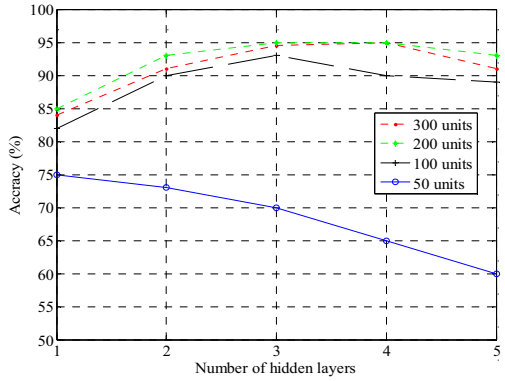


Figure 7. The Recognition Accuracy with The Number of Hidden Layers and Hidden Units

scenario with different experimental parameters to test our approach, just as Figure 5 shows. Both the receiver and transmitter are placed at the height of 1.25m. The distance from receiver to Wi-Fi router is about 3 meters. In the experiment, we choose 5 volunteers (5 males with a similar figure) to operate one number gesture at a time in the middle of line-of-sight (LOS) path between receiver and transmitter. Subjects must keep their hands stationary and in the same position so as to eliminate the disturbance of irrelevant variables. We conduct each set of experiment for 5 minutes and then change the subject. The transmitting rate is set at 100 packets per second. A deep network with a 90-200-200-200 structure is utilized in our experiment.

### *B. Experimental Results*

#### 1) *Overall Performance*

To evaluate the performance of our proposed DeNum approach, a confusion matrix of ten number gestures is used in our experiment. Figure 6 shows the confusion matrix for 10 number gestures. Y-axis denotes the actual gestures operated by our volunteers while X-axis represents the classified gestures. It can be observed that actual gestures like 5, 7, 8 achieve great recognition accuracy over 0.98 and 0, 1, 3, 4 can be recognized with considerable accuracy over 0.92. Besides, the average accuracy is around 0.94, which is sufficiently high for our approach. Though, we can't ignore the mistakes in distinguishing similar gestures (e.g., 0 and 6, 2 and 3, 3 and 4, 0 and 9) because they affect the propagation link in approximately the same way. We still get an acceptable accuracy of gesture recognition.

#### 2) *Impact of the Deep Learning Parameters*

We evaluate the performance by varying the number of hidden layers and hidden units. We set the hidden units number of each layer at 50, 100, 200, 300, respectively. In Figure 7, when the number of hidden units increases, we could generally obtain a better recognition accuracy. However, we fail to see the same trend as the number of hidden layers increases. Thus, we obtain optimal parameters with 3 hidden layers and 200 units per hidden layer.

#### 3) *Impact of Environmental Changes*

To test the approach robustness, we conduct our experiments with several common setting changes, including (a) normal surroundings, (b) turning on the fan, (c) opening windows, (d) changing the room layout and (e) arranging another subject standing nearby. As shown in Figure 8, approach accuracy still seems to be acceptable by turning on the fan or opening windows. However, the accuracy is obviously affected by changing the room layout from 94% to 76% and obviously deteriorates from 94% to 67% when another subject stands nearby.

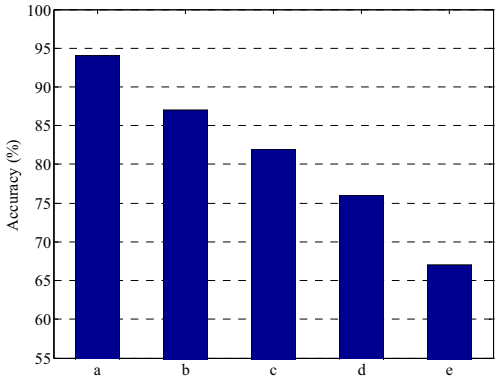


Figure 8. Mean Accuracy with Different Environmental Changes.

## VI. DISCUSSION

### *A. Controlled Making Gestures*

In order to minimize interferences of irrelevant variables, we asked volunteers remain hands stationary during data collection. They were instructed not to move their heads or any other significant movements. However, some slight natural motions were allowed such as moving lips and blinking eyes. We also required volunteers to make one predefined gesture at a time and keep inter arrival time between 2 to 3 seconds so that the finishing points can be correctly identified. In the future, we plan to recognize number gestures in successive actions.

### *B. Different Kinds of Subjects*

DeNum currently is only tested for male subjects in the 22-28 age range. Considering that the size of palms and fingers can affect the experimental results, the performance

of DeNum can be improved if we take different kinds of volunteers as subjects.

### C. The Position of Subjects

Intuitively, we instructed subjects to sit in the middle of LOS path between receiver and transmitter so as to generate relatively obvious features. Previous research [12] had exploited the relationship between the varied CSI values and the subjects' positions. In the future, we will try to find the most optimal position for gesture recognition.

## VII. Conclusion and Future work

In this paper, we present a robust and accurate device-free number gesture recognition approach, named DeNum. We exploit the amplitude difference variance through a sliding window to detect the finish points of gesture transitions and extract static gesture information. We creatively recognize number gestures via fine-grained CSI information based on deep learning. We adopt a 4-layer deep learning model to extract discriminative features from both the amplitude and phase information and then inject the output of the last hidden layer into SVM classifier for accurate classification. Through extensive experiments, we can prove DeNum approach achieves an average accuracy of 94% in office scenario and is sufficiently accurate to deal with environmental changes.

In the perspective of future research, we intend to study further in the following aspects: First, we plan to expand the scope of use in multi-person scenario. Second, we try to extract number gestures from successive actions as "gesture password" for further application. We expect to make breakthroughs in these challenging topics.

## Acknowledgment

This work is supported in part by the Project on the Natural Foundation of Jiangsu Province under grant NO. BK20151451.

## References

[1] J. Wu, W. Gao, and Y. Song et al, "A simple sign language recognition approach based on data glove," in Fourth International Conference on Signal Processing Proceedings, Beijing, 1998, pp. 125-145.

[2] P. Kumar, J. Verma and S. Prasad, "Hand data glove: a wearable real-time device for human-computer interaction," International Journal of Advanced Science & Technology, vol.43, pp.15-26, June 2012.

[3] S. S. Rautaray and A. Agrawal, "A novel human computer interface based on hand gesture recognition using computer vision techniques," in International Conference on Intelligent Interactive Technologies and Multimedia, Allahabad, 2010, pp. 292-296.

[4] A. E. Kosba and M. Youssef, "RASID demo, A robust WLAN device-free passive motion detection approach," in IEEE International Conference on Pervasive Computing and Communications Workshops, Lugano, 2012, pp. 180 - 189.

[5] C. Wu, Z. Yang, Z. Zhou, et al, "Non-Invasive Detection of Moving and Stationary Human With WiFi," IEEE Journal on Selected Areas in Communications, vol.33(11), pp. 2329-2342, October 2015.

[6] F. Adib, Z. Kabelac and D.Katabi, "Multi-person localization via RF body reflections," in Usenix Conference on Networked Approachs Design and Implementation, Santa Clara, 2015, pp.279-292.

[7] D. Halperin, W. Hu and A. Shet et al, "Predictable 802.11 packet delivery from wireless channel measurements," Acm Sigcomm Computer Communication Review, vol.40(4), pp.159-170, October 2010.

[8] H. Wang, D. Zhang and Y. Wang et al, "RT-Fall: A Real-time and Contactless Fall Detection Approach with Commodity WiFi Devices," IEEE Transactions on Mobile Computing, 2016, no. 1, pp. 1, PrePrints.

[9] K.Ali, A. Liu and W. Wang et al," Keystroke recognition using WiFi signals," in ACM MobiCom, Paris, 2015, pp.90-102.

[10] K. Qian, C. Wu and Z. Yang et al, "PADS: Passive detection of moving targets with dynamic speed using PHY layer information," in Parallel and Distributed Approachs (ICPADS), the 20th IEEE International Conference, Hsinchu, 2014, pp.1-8.

[11] W. Wang, A. Liu and M. Shahzad et al, "Understanding and modeling of wifi signal based human activity recognition," in ACM MobiCom, Paris, 2015, pp.65-76.

[12] K. Wu, J. Xiao and Y Yi et al, "CSI-based indoor localization," IEEE Transactions on Parallel & Distributed Approachs, vol.24(7), pp.1300-1309, July 2013.

[13] Y. Chapre, A. Ignjatovic and A. Seneviratne et al, "CSI-MIMO: indoor Wi-Fi fingerprinting approach," in Conference on Local Computer Networks, Edmonton, 2014, pp.202-209.

[14] W. Xi, J. Zhao and X. Li et al, "Electronic frog eye: Counting crowd using WiFi," in IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, Toronto, 2014, pp.361-369.

[15] Y. Zeng, P. H. Pathak and P. Mohapatra, "Analyzing shopper's behavior through WiFi signals," in WPA '15 Proceedings of the 2nd workshop on Workshop on Physical Analytics, Florence, 2015, pp.13-18.

[16] X. Wang, L. Gao and S. Mao et al, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in IEEE Wireles Communitacion and Network Conference, New Orleans, 2015, pp.1666-1671.

[17] X. Wang, L Gao and S. Mao, "PhaseFi: Phase fingerprinting for indoor localization with a deep learning approach," in IEEE Global Communications Conference, San Diego, 2015, pp.1-6.

[18] X. Zhang, J. Wang and Q. Gao et al. , "Device-free wireless localization and activity recognition with deep learning," in IEEE International Conference on Pervasive Computing and Communication Workshops, Sydney, 2016, pp.1-5.

[19] S. Gupta, D. Morris, S. Patel and D. Tan, "SoundWave: using the doppler effect to sense gestures," in Sigchi Conference on Human Factors in Computing Approachs, Austin, 2012, pp.1911-1914.

[20] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," Science, vol.313(5786), pp.504-507, July 2006.

[21] Y. Bengio, P. Lamblin D. Popovici and H. Larochelle, "Greedy layer-wise training of deep networks," Advances in Neural Information Processing Approachs, vol.19, pp.153-160, August 2006.

[22] B. Schölkopf, J. C.Platt, J. Shawetaylor, A. J. Smola and R. C. Williamson, "Estimating the support of a high-dimensional distribution," Neural Computation, vol.13(7), pp.1443-1471, July 2001.

[23] C. Chang and C. Lin, "LIBSVM: A library for support vector machines," Acm Transactions on Intelligent Approachs & Technology, vol.2(3), pp. 389-396, April 2011.