Towards Location Free Movement Recognition with Channel State Information

by

CHUNHAI FENG

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2019

To my family

ACKNOWLEDGEMENTS

It has been over twenty-three years since the first day I went to elementary school. From a naive Chinese countryside boy who did not like school at all to an independent and open-minded person with a doctoral degree in the United States, I have had such a wonderful journey with different people at different time and places. When burying myself in these memories, I can not feel anything but lucky and grateful.

Firstly, I would like to express my deepest gratitude to my supervisor, Dr. Yonghe Liu, for his tutoring and guidance towards the completion of my PhD degree. I am deeply grateful of the opportunity to pursue the doctoral degree with scholarship in UTA.

Secondly, I would like to express my sincere appreciation to Dr. Jianling Hu, who was my Master supervisor in Soochow University. I am truly grateful of the enlightening discussions we had and the inspirational suggestions he gave about the research and life plans. The three years I spent with all other labmates in 331 is one of the most indelible memories in my life.

Thirdly, I would also like to thank my committee members, Dr. Gergely Zaruba, Dr. Hao Che and Dr. Junzhou Huang, for all the assistance and suggestions on this dissertation. Besides, I would also like to thank all my previous teachers for their patience and instructions.

Lastly but most importantly, my parents, Mr. Weihong Feng and Mrs. Koufang Zhu. Similar as many Chinese parents, they do not talk about thanks and love very much, but I am so grateful to have them in my life. I would like to thank them for all the supports, company and love. Especially, I appreciate their respects of my decisions and choices, even though sometimes the outcomes are not in their favor. Besides, I would also like to

thank my grandparents and other relatives, who have been always supportive and caring. In addition, I am also indebted to all my friends, no matter it is in or out of the country, on or off campus, online or offline, for their ungrudging company.

One more thing, I would like to express my love to all my beloved friends, family and relatives. I also hope, one day, I would have the courage to tell the world of my love because love is love and it always wins.

November 18, 2019

ABSTRACT

Towards Location Free Movement Recognition with Channel State Information

CHUNHAI FENG, Ph.D.

The University of Texas at Arlington, 2019

Supervising Professor: Yonghe Liu

Channel state information based movement recognition has gathered immense attention over recent years. Different from traditional systems which usually require wearable sensors or surveillance cameras, many existing works achieved desirable performance with only wireless signals in various applications, including healthcare, security and Internet of Things, with different machine learning algorithms. However, it still remains many challenges to be solved. Particularly, the location dependent nature of channel state information is one of the most significant challenges remaining. Firstly, many previous researchers deploy and evaluate their systems with employing machine learning or deep neural networks. Because of the aforementioned challenge, the models would need to be retrained with the dataset collected from new locations. However, they usually fail to consider the availability of enough samples to be trained. In other words, it generally requires a large number of samples to train a robust model, which is challenging especially at the early stage of system deployment. Therefore, it is significant to develop a system that is able to accommodate the size of available samples in the profile. Secondly, as the location dependent features are interleaved with movement dependent features, how to separate them effectively becomes

the main challenge in order to correctly identify the activities at different locations without training new models.

In order to address the first challenge, we propose a three-phase system Wi-multi that targets at recognizing multiple human movements in a wireless environment. Different system phases are applied according to the size of available collected samples. Specifically, distance-based classification using Dynamic Time Warping is applied when there are few samples in the profile. Then, Support Vector Machine is employed when representative features can be extracted from training samples. Lastly, recurrent neural networks is exploited when a large number of samples are available. In addition, an effective movement sample extraction algorithm is also proposed to identify the start and end points of multiple subject movements. A diverse dataset of multiple human activities is also built in order to evaluate the performance of this system. Extensive experiments results show that Wi-multi achieves an accuracy of 96.1% on average. It is also able to achieve a desirable tradeoff between accuracy and efficiency in different phases.

In order to solve the second challenge, a deep neural network system, consisted of feature extraction, feature separation, gesture recognition and location identification modules. The key idea in designing this system is to separate movement dependent features from location dependent features. Specifically, a feature extraction module that consisted of three long short-term memory layers network is employed to select representative features. Afterwards, the first half features are fed into the gesture recognition module while the second half is passed to location identification module. During the training process, the network will learn to cluster the first half features as gesture dependent features while the second half as location dependent features by minimizing the total loss of gesture recognition and location identification modules. The system is evaluated with a dataset collected from various subjects performing 4 different gestures in 2 rooms and 6 locations. The

results show that the proposed location independent gesture recognition system is able to achieve 85.42% accuracy on average in new locations.

Keyword: Channel State Information; Activity Recognition; Multiple Subjects; Location Independent

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

Table Page

CHAPTER 1

Introduction

Given the ubiquitous presence of WiFi signals, numerous research efforts have been devoted to fully unfold their potentials in real life applications. Among them, channel state information (CSI) of WiFi signals have recently been extensively exploited in various scenarios. These include indoor navigation [1], gesture recognition [2], and human body activity identification [3]. For instance, house monitoring in case of an intrusion and emergency care for elders are very common contexts of use in modern society. Obviously, many traditional approaches with carry-on sensors [4] or cameras [5] have achieved desirable performance in movement recognition. However, it is now considered as inconvenience to ask users to wear sensors all the time because of the battery charge issues. Besides, it also raises many privacy security concerns on camera based systems. On the contrary, CSI can be extracted from commercial wireless devices instead of requiring extra costs of equipments as in traditional approaches. Therefore, it has attracted broad interests in recent years.

Because of the multipath propagation phenomenon, human presence or body movement around the wireless devices can affect the strength quality of Wi-Fi signal [6, 7]. CSI can record this detailed physical layer information from different subcarriers of the channel. By modifying the driver of Intel 5300 network interface card (NIC) [8], many existing papers proposed various systems to detect human activities, such as keystroke [9], gestures [10] and breathing [11]. However, it is well known [12, 13, 14, 15] that different movements can lead to diverse CSI variations. Owing to the effects of multipath propagation, movements performed in different locations can also cause different reflections on differ-

ent signal paths. As a result, CSI fluctuations introduced by the same movements can show very distinct patterns in different locations, even though it is due to the same movement performed by the same subject. On one hand, many existing works would require to train new models with dataset collected from new locations. They usually assume that enough samples can be provided for model training [16, 17]. However, it is generally unlikely to have hundreds or even more samples in reality, especially at the early stage of system deployment. Since machine learning or deep neural networks usually require adequate samples, it becomes infeasible to apply them in activity detection if there are few samples available. In addition, signal processing methodologies[9, 18, 19] take longer time if size of dataset increases. Therefore, it is crucial to design a system that is able to accommodate the size of available samples in the profile. One the other hand, it is evident that raw CSI data contains both movement and location dependent features. For identifying movements in different locations without training new models, the main challenge becomes *effectively separating movement dependent features from location dependent features.*

In the first part of this paper, we propose a three-phase system that targets at multiple human activity recognition according to the size of available samples in the profile. In the first phase, PCA that reduces dimensions of features and Dynamic Time Warping (DTW) that calculates the distance between various length signals so as to measure similarities are employed at the beginning stage of building the system when limited training samples are available. In the second phase, when more samples are available, support vector machine (SVM) is exploited for model training and testing where a number of representative features can be extracted from both time and frequency domain. In this case, a model can be pre-trained and hence it is not necessary to compare the similarities among all samples as in the first phase of the system, which may largely reduce time cost as sample size increases. In the third phase, we propose a deep learning system structure based on Long Short Term Memory (LSTM) unit if a large number of samples are available in the profile. LSTM,

as one type of recurrent neural network (RNN) units, is capable of remembering and filtering the past information in the input sequence during training process. The proposed deep learning network is able to automatically select high level features without any preprocessing modules. The evaluation results demonstrate that our proposed system achieves a desirable tradeoff performance between accuracy and efficiency in different phases. In general, Wi-multi can achieve 96.1% accuracy on average.

Before activity classification, we also have developed an effective activity sample extraction algorithm to identify the start and end points of multiple subject activities. Firstly, we apply outlier filtering and differential algorithms to the variance of CSI values among different subcarriers. Secondly, we calculate the largest eigenvalues from both amplitude and calibrated phase correlation matrices so as to eliminate potential false detection. The extracted samples are then presented to different system phases for further analysis. It is shown that our algorithm can extract activity samples in both noisy and non-noisy environments with multiple subjects.

The key contributions of this part about three phase multiple human activity recognition system can be summarized as following.

- We propose a three-phase system that can recognize multiple human activities, where each phase is designed according to the size of available dataset in the profile during different stages of a system deployment.
- We evaluate the system in terms of various aspects. Extensive results show that Wi-multi is able to achieve desirable tradeoff between accuracy and efficiency in different phases.
- We propose a novel activity extraction algorithm that is able to identify the start and end point of an activity even in noisy environment with multiple subjects.

In the second part of this paper, we develop a location independent gesture detection system by taking advantage of channel state information from commercially available WiFi

devices. Unlike previous works that require training new machine learning models for different locations, our system is able to recognize gestures performed in new locations with no data therein being trained. We propose a deep learning network that is capable of extracting gesture and location dependent features simultaneously and separating them from each other during the training process, which lead to the location independent recognition capability. Specifically, the designed network contains four components, which are feature extraction, feature separation, gesture recognition and location identification modules. Feature extraction that consists of three long short-term memory (LSTM) layers is employed to select high level representative features. Afterwards, the first half extracted features are fed into the gesture recognition module while the second half is passed to the location identification module. Both gesture recognition and location identification modules are composed of two fully connected layers in order to map the features to latent space for classification. By minimizing the total loss of the gesture recognition and location identification modules during the training process, the network is able to gradually cluster the first half extracted features as highly gesture dependent representations while the second half as intensively location dependent representations. The result is a system that can successfully separate gesture and location related features and address the challenge as discussed above.

In order to evaluate our proposed methodology, we collect CSI data by asking various subjects to perform 4 different gestures (boxing, swipe, punch and handwave) at 6 different locations in 2 rooms. Experiments show that our system is able to achieve an average of 85.42% accuracy of gesture detection in new locations.

The key contributions of this part about location independent gesture detection can be summarized as following.

- We propose a location free gesture detection system that is able to recognize gestures performed in different locations without training new models.

- We design a multi-layers neural network structure that can extract both gesture related and location related features independently.
- We separate gesture related features from location related features by reducing the overall loss of both gesture recognition and location identification module.
- We will evaluate our system with extensive experiments in different locations.

In the remaining sections, we briefly introduce the related works in Chapter 2, and presents channel state information in the Chapter 3. Chapter 4 introduces the design of activity extraction algorithm, different system phases for multiple human activity recognition and evaluation results. Chapter 5 presents the system overview of location free gesture identification. The conclusion is made in Chapter 6.

CHAPTER 2

Related Works

In this section, we briefly describe recent works on movement recognition. They can generally be divided into two categories, traditional and WiFi based approaches. Traditional systems are usually based on carry-on sensors or computer vision, which may raise concerns on inconvenience or privacy. With the availability of channel state information from commercial wireless devices, many recent works developed movement recognition systems by analyzing the correlations of CSI and movement. However, many of these approaches fail to consider the efficiency of collecting large number of samples for training with machine learning alorithms in each new location. Instead, the first part of our proposed systems is able to accommodate different sizes of available samples in the profile.

Traditional Approaches: Before channel state information becomes available, traditional movement recognition systems are usually designed with wearable sensors or computer vision methods. For example, the author in [20] proposed a system to detect human eating and drinking gestures with dedicated sensors. Besides, a framework designed by Zhang [21] employed a three-axis accelerometer and multichannel electromyography sensors to detect sign language hand gestures. In [22], the authors developed a system to distinguish among various hand shapes and orientations by taking advantage of the information provided by depth sensors. Moreover, recent smart devices like Apple Watch [23] and Fitbit [24] are able to detect different arm movements for healthcare purposes.

In addition, many works achieved desirable performance using various computer vision algorithms. For instance, TAFFI in [25] designed a robust algorithm to detect pinch gestures with camera images. Besides, the authors in [26] utilizes multiclass support vector

machine to build a model with dedicated features for real-time hand gesture identification. Another system in [27] exploited principal component analysis to recognize various gestures with a webcam. Similarly, [28] designed a two level approach to address the challenges of real-time hand gesture recognition with Haar-like features.

Wi-Fi based Approaches: Since CSI can be obtained from off-the-shelf network interface card [8], it has been exploited in a variety of applications. There are usually two types of activity identification, coarse-grained and fine-grained, with channel state information.

Coarse-grained activity usually refers to activities performed at macro levels. For example, many previous works [29, 18, 30, 31, 32] focus on daily activity recognition (including walking and running) and achieve desirable performance in both Line-of-Sight and Non-Line-of-Sight circumstances by the support of two mathematical model in [31]. Both WiFall [33, 34] and RT-Fall [35] build an alarm system to detect human falls in realtime. The authors in [36] develop a Smokey system that can detect the smokers behavior without deploying special devices. Moreover, the authors in [14] are able to detect different humans based on the features of their behaviors. However, many of these works only target at single subject environment. This limits the potentials of applications according to a national survey [37], which indicates that there are usually around 2 to 3 people in each household in 2015.

Fine-grained activity refers to activities performed at micro levels. For instance, the authors in [38] employ CSI to detect the breath rate and monitor sleep quality with predeployed antennas and transceivers. Besides, keystroke systems implemented on either laptop [9] or smartphone [39] yield desirable accuracy in keystroke recognition. In addition, WiHear in [40] can analyze and detect people speech based on the disturbance of CSI caused by lip movement. Furthermore, many recent works have developed gesture recognition systems [10, 41, 42, 43, 44] based on channel state information (CSI) from commer-

cially available Wi-Fi devices. For example, WiFinger [12] captures the subtle movement of fingers and recognizes fine-grained finger gestures with wireless devices. The authors in [45] utilized deep learning approaches to extract dedicated features and employed support vector machine for classification. Moreover, researchers in [46, 47] have tried to investigate the impact of gesture position and orientation on CSI. Indeed, the authors in [48] designed a neural network structure and tried to eliminate location correlated information with adversarial learning. However, their system has quite low accuracy when small number of locations data are used for training. Moreover, their system requires to preprocess raw CSI data and also has several empirical parameters, which makes it difficult to deploy in different scenarios. The biggest difference between our proposed system and their system is that our system is able to separate gesture dependent features from location dependent features.

CHAPTER 3

Channel State Information

WiFi signals arriving at a receiver usually come from different paths [49]. This multipath effect often causes interference, phase shifting and fading of the signal. Compared with the environment with only one person, multiple humans can cause even more disturbances in wireless channels.

There are usually two ways to evaluate the channel condition without chip level access. On one hand, Radio Signal Strength Indicator (RSSI) is the most common way because of its accessibility. However, it only provides limited amplitude information due to its low resolution. On the other hand, CSI, which represents the channel frequency response (CFR), can capture both the amplitude and phase variance for each OFDM subcarrier.

By modifying the driver of Intel 5300 NIC in 802.11n network [8], we are able to get CSI values of 30 subcarriers between one pair of transmit-receive antennas. Let T and R represent the WiFi signals in the frequency domain from the transmitter and receiver respectively. The wireless channel model can be modeled as

$$R = H \times T + \mathcal{N}_o, \tag{3.1}$$

where H represents the CSI estimation of channel frequency response. Note that H can then be approximated as

$$\widehat{H} = \frac{R}{T}, \tag{3.2}$$

assuming noise $\mathcal{N}_o$ follows zero mean complex normal distribution of circularly-symmetric, i.e., $\mathcal{N}_o \sim N_c(0, \Gamma)$.

$\widehat{H}$ is a matrix consisted of channel state information from all subcarriers. $\widehat{h}$, CSI of one subcarrier, can also be represented as

$$\widehat{h} = ||\widehat{h}||e^{j\angle\widehat{h}}, \tag{3.3}$$

where $||\widehat{h}||$ and $\angle\widehat{h}$ represent the amplitude and phase information respectively.

CHAPTER 4

Three Phase System for Multiple Human Activity Recognition

4.1 Activity Extraction

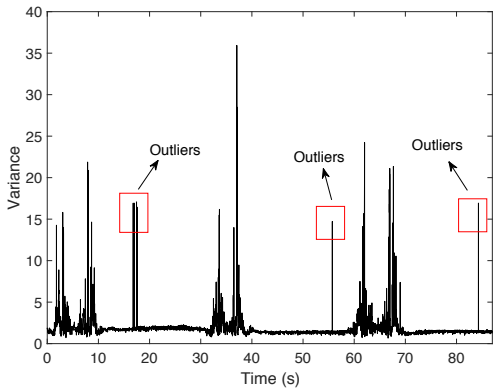
In order to extract the activity sample, it is necessary to detect the period of time where activity occurs. We firstly observe that the variances of CSI amplitudes among different subcarriers can be used as an indicator for activity presence. An example of CSI amplitude variance among 30 subcarriers can be found in Fig.4.1a. It can be easily observed that variance when no activity presents is much more stable than that when activities occur. In this section, we design a two-step algorithm to extract the start and end points of each activity. Note that only one CSI stream of 30 subcarriers is required for activity extraction.

4.1.1 Step one: Differential Threshold Estimation

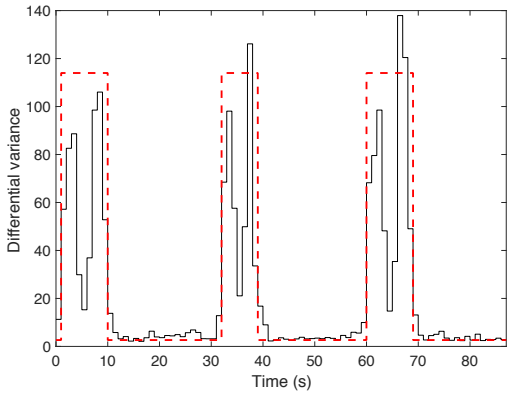
As shown in Fig. 4.1a, the outliers caused by the internal hardware errors improve the difficulty for activity extraction. In order to address this challenge, we compare the CSI value at the m_{th} time point with a threshold defined as $\delta_v = \lambda|V(m+1) - V(m-1)|$, where λ is an empirical coefficient. Assume the amplitude variance of 30 subcarriers is V, then we remove the outlier by setting it as the average of the values at prior and posterior packets.

$$V(m) = \begin{cases} (V(m-1) + V(m+1))/2, & \text{if } V(m) > \delta_v \\ V(m), & \text{otherwise} \end{cases} \tag{4.1}$$

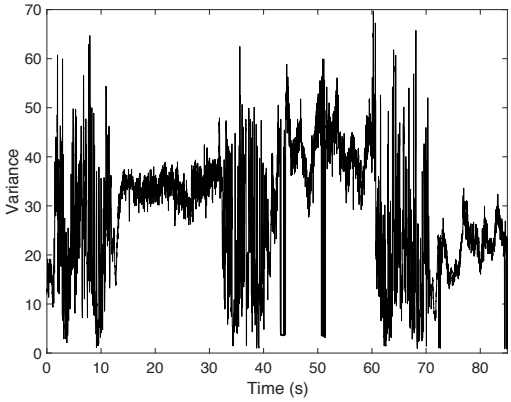
Next, we split the signal into even slots in time domain and determine if there is activity presence in each slot. Here we assume that the length of the activity is longer than



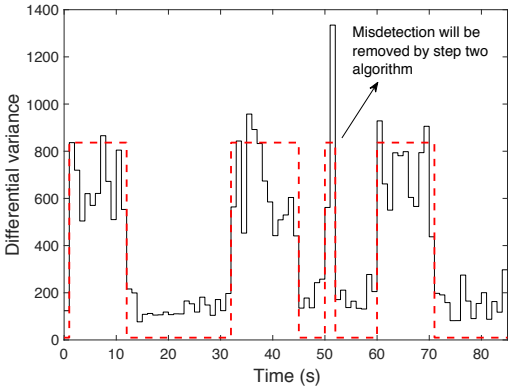
(a) Variance under non-noisy environment



(b) Differential variance and extraction result



(c) Variance under noisy environment



(d) Differential variance and extraction result

Figure 4.1: Performance of activity extraction algorithm in non-noisy and noisy environment

one time slot. In our case, we set the length of each time slot as 1 second. It can be changed to larger or smaller values according to different scenarios. We then consider continuous time slots with activity presence as one activity sample.

Denote the sampling rate of CSI as R_s , the variances of CSI values in the i_{th} slot can then be described as $V(j)$, where $j = 1, 2, ..., R_s$. In ascending order, they can be

represented as $V'(1) < V'(2) < ... < V'(R_s)$. We then compute the difference between the sum of largest half values and the sum of smallest half values as

$$D_i = \sum_{j=1}^{R_s/2} (V'(j + R_s/2) - V'(j)). \tag{4.2}$$

Fig.4.1b depicts the corresponding results of Fig.4.1a. By comparing D_i with a self adaptive threshold δ_D, we can obtain an initial result that determines whether there is an activity in this slot.

$$I_i = \begin{cases} 1 & \text{if } D_i > \delta_D \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

Here I is the indicator of the presence of activity, $\delta_D = (\sum_{i=1}^{L/R_s} D_i)/(L/R_s)$ and L here represents the total length of the signal. Afterwards, we consider continuous time slots where I_i is 1 as one activity sample. In non-noisy environment, it can usually detect the boundary of activity correctly as shown in Fig.4.1b. However, it may also result in detection errors in some cases, especially in extremely noisy environments or/and multiple subjects activities. Fig.4.1c and Fig. 4.1d show an example that the first step algorithm misdetects an activity. In order to address this challenge, our algorithm uses a second step to double check the results and remove potential, although rare, misdetections.

4.1.2 Step two: Eigenvalues Comparison

As discussed above, the CSI values in stationary environment tends to be more stable than that with human presence. Therefore, the correlation between consecutive CSI values can be much higher in stationary environment. In this case, we build the correlation matrices of both amplitude and calibrated phase [29] within a time window. Assume the size of window is W, the CSI measurements in this window can be described as

$$H(i) = [H_1(i), H_2(i), ..., H_K(i)]$$

where $i = 1, 2, \ldots, W$ and K is the total number of subcarriers. Thus the covariance matrices of amplitude and phase can be computed as

$$\mathbf{A} = \begin{bmatrix} cov(|H(1)|, |H(1)|) & \cdots & cov(|H(1)|, |H(W)|) \\ \vdots & \ddots & \vdots \\ cov(|H(W)|, |H(1)|) & \cdots & cov(|H(W)|, |H(W)|) \end{bmatrix}$$

and

$$\mathbf{P} = \begin{bmatrix} cov(\angle\widetilde{H(1)}, \angle\widetilde{H(1)}) & \cdots & cov(\angle\widetilde{H(1)}, \angle\widetilde{H(W)}) \\ \vdots & \ddots & \vdots \\ cov(\angle\widetilde{H(W)}, \angle\widetilde{H(1)}) & \cdots & cov(\angle\widetilde{H(W)}, \angle\widetilde{H(W)}) \end{bmatrix}.$$

Afterwards, the largest normalized eigenvalues of **A** and **P** can be computed as

$$\alpha_A = max(norm(eigen(\mathbf{A})))$$

and

$$\alpha_P = max(norm(eigen(\mathbf{P})))$$

respectively. After conducting several experiments, we observe that α_A and α_P tend to be larger in stationary environment. As depicted in Fig. 4.2, scenarios with activities can be easily separated from stationary scenarios. Moreover, this threshold is independent from different environmental background since eigenvalues are power independent. As shown in Fig. 4.1d, the misdetection shown in the result of step one can then be removed. Therefore, step two can further improve the accuracy based step one result.

4.2 Three-phase System Design

In this section, we describe the system structure of Wi-multi based on a three phase design, corresponding to different phases of system deployment. Phase one is applied when only few samples are available in the profile. Phase two is employed only when

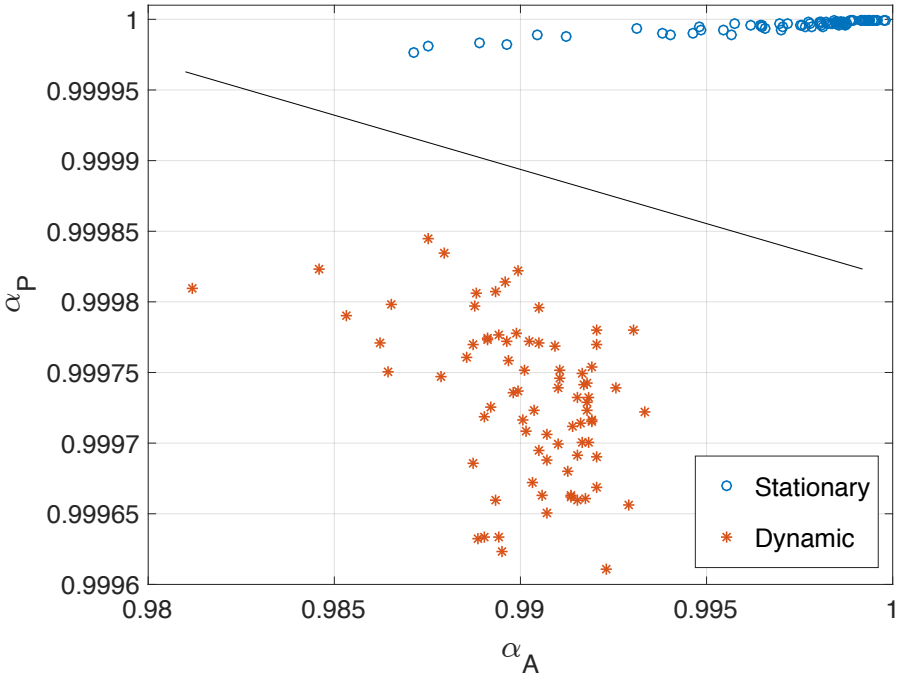


Figure 4.2: Eigenvalues comparison between stationary and dynamic cases

the effective features can be extracted and trained with SVM. Lastly, phase three based on LSTM is employed when there are a large number of samples for deep learning networks. An overview of the system structure is shown in Fig. 4.3.

Before applying phase one and phase two, we first apply interpolation to the raw data as data points can be missing because of errors in the system or the collecting tool [11]. Afterwards, we also apply a low pass Butterworth filter and phase calibration on amplitude and phase in order to remove outliers and random noises [29]. Note that phase three does not need any preprocessing as it can automatically extract representative features.

4.2.1 Phase One

In this phase, we assume very limited availability of samples in the profile, i.e., at the early stage of system deployment. Firstly, PCA is exploited to remove correlated information and reduce feature dimensions among the 30 subcarriers. Secondly, DWT is

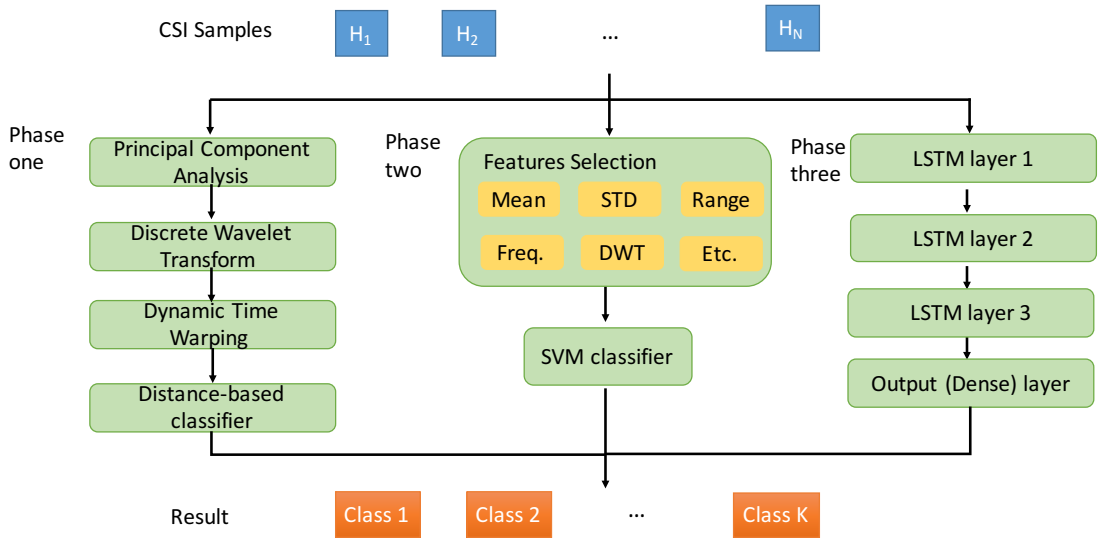


Figure 4.3: Three-phase system overview

employed in order to compress the length of signal data without losing much representative information. Lastly, a distance based method is applied to identify the label of activity.

4.2.1.1 Principal Component Analysis

As discussed earlier, PCA is used to remove correlated information among 30 sub-carriers. The detailed PCA implementation is as following.

Training Samples Combination: Denote CSI matrix of each training sample as H_i, and the size of each matrix is $L_i \times 30$, where $i = 1, 2, ..., N$. Thus the combined matrix of training data can be represented as H, whose size is $\sum_i^N L_i \times 30$.

Static Component Removal: In this step, the static component is calculated by the average of the signal in each subcarrier. Denote it as avg_j, where $j = 1, 2, ..., 30$ represents subcarrier index. By subtracting avg_j from each column of H, we can get a centered matrix H_D.

Covariance Matrix Computation: The covariance matrix is then computed as $H_D^T \times H_D$.

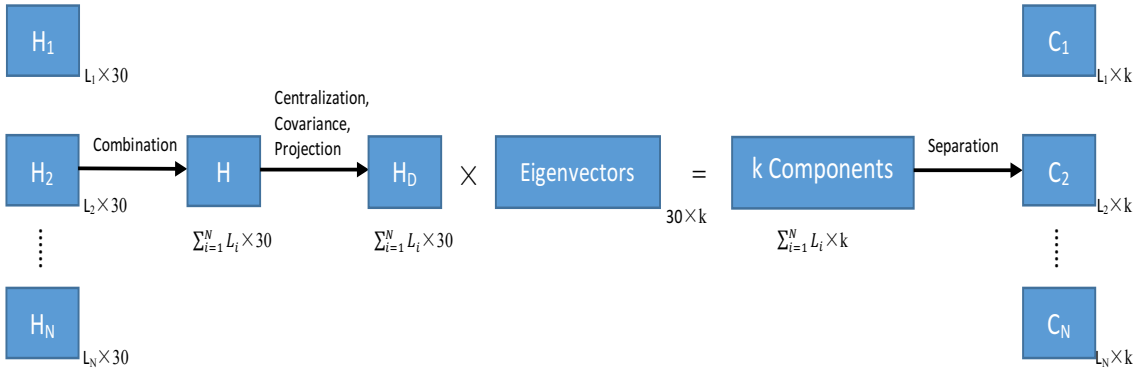


Figure 4.4: PCA implementation

Eigenvectors Calculation: The eigenvectors corresponding to the covariance matrix can be calculated as $q_1, q_2, ...q_n$ respectively.

Training Samples Projection: In order to project training samples to the eigenspace, the first k components can then be computed as $[c_1, c_2, ...c_k] = H_D \times [q_1, q_2, ..., q_k]$, where c_i represents the i_{th} component.

Testing Samples Projection: Similarly, denote the centered CSI matrix of the testing samples as T_D, then the first k components of the testing data can be calculated as $Z = T_D \times [q_1, q_2, ..., q_k]$ by projecting testing samples to eigenspace towards the same direction as training data.

Matrix Separation: Let $C = [c_1, c_2, ...c_k]$, whose size is $\sum_i^N L_i \times k$. Therefore, the first k components of each training sample can be obtained by splitting it into N separate matrices. Similar separation process is applied to matrix Z computed in the last step.

Note that this approach requires the computation of the eigenvectors only once, meaning that both the training data and testing data are projected to the eigenspace in the same direction. A flow chart of PCA implementations is presented in Fig.4.4.

4.2.1.2 Discrete Wavelet Transform

Because of the high computation cost often associated with longer sample activity data, we further employ discrete wavelet transform (DWT) to compress the signal. It is able to reduce the length of the signal without losing much representative information. Denote the measured discrete signal as

$$s[n] = \frac{1}{\sqrt{M}} \sum_{i} \alpha[j_0, i]\phi_{j_0,i}[n] + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{\infty} \sum_{i} \beta[j, i]\psi_{j,i}[n], \tag{4.4}$$

where M is the length of the signal. $\phi_{j_0,i}[n]$ and $\psi_{j,i}[n]$ are defined as orthogonal to each other and they represent scaling functions and wavelet functions respectively. Similarly, $\alpha[j_0, i]$ and $\beta[j, i]$ are termed as approximation coefficients and detail coefficients respectively. They can be modeled as

$$\alpha[j_0, i] =< s[n], \phi_{j_0+1,i}[n] >= \frac{1}{\sqrt{M}} \sum_{n} s[n]\phi_{j_0+1,i}[n] \tag{4.5}$$

$$\beta[j, i] =< s[n], \psi_{j+1,i}[n] >= \frac{1}{\sqrt{M}} \sum_{n} s[n]\psi_{j+1,i}[n] \tag{4.6}$$

in the j_{th} level.

In this paper, we adopt approximation coefficients in order to reduce computation cost. Note that different levels of computation of DWT will lead to various compression lengths of the signal. The higher level of DWT, the shorter length the signal can be compressed. We will discuss the impact of different DWT levels in the next section.

4.2.1.3 Distance-based classification

In order to compare the similarities of different waveforms, DTW is employed to calculate the Euclidean distances between signals. By aligning the waveforms, DTW yields the addition of Euclidean distances between their corresponding points. Smaller distance

usually represents higher similarity of waveforms. After the construction of different profiles, the label of the new test sample is predicted by comparing its distances from different activities.

Denote the Euclidean distances between a test sample and each sample in the profile as D_i, where $i = 1, 2, ..., N$ represents N different activity samples in the profile. Denote the increasingly ordered K distances for test sample as D'_i, where $D'_1 < D'_2 < ... < D'_K$ and K is chosen empirically. Assume that there are n kinds of different activities in the profile and the corresponding label of sample that is associated with D'_i is B_i. The final predicted label of the test sample can be presented as

$$F = \begin{cases} 1 & \text{if } \sum_{i=1}^{K}(B_i == 1) \geq \sum_{i=1}^{K}(B_i \neq 1) \\ 2 & \text{if } \sum_{i=1}^{K}(B_i == 2) \geq \sum_{i=1}^{K}(B_i \neq 2) \\ \vdots & \\ n & \text{if } \sum_{i=1}^{K}(B_i == n) \geq \sum_{i=1}^{K}(B_i \neq n) \end{cases} \tag{4.7}$$

Note that our scheme predicts labels based on the similarity between test signal and sample signals in the profile. Therefore, it still works well even if only a few number samples are available in the profile.

4.2.2 Phase Two

When more samples become available as time progresses, it may become difficult to still use the system of phase one for activity recognition. Distance-based classification requires calculation with every activity sample in the dataset for each test data. It leads to inefficiency especially with longer signals and larger sample sizes. In this case, we firstly extract representative features from time and frequency domain, then utilize SVM to train a

model for classification. Since the model can be pre-trained, it can achieve higher efficiency with larger sample size as compared with phase one.

4.2.2.1 Feature Extraction

We manually select representative features from both time and frequency domains. As stated in [29], CFR power can be considered as an indicator of the speed of paths length change caused by multiple human activities. Therefore, we employ six different features, including the standard deviation, median absolute deviation, max, mean, first and third quartile of the filtered CFR power respectively. Besides, same features from calibrated phase [17] are also extracted. In addition, we also select six frequency domain features from different energy levels of DWT [31] which indicate the intensity of movement in each speed range. In general, we can get a group of 18 features for each sample in total.

4.2.2.2 Support Vector Machine

Let $x_i = \{f_1, f_2, \ldots, f_{N_f}\}$ be the i_{th} sample and y_i be the corresponding label in feature space, where N_f is the number of extracted features. In other words, the training dataset can be expressed as $T = (x_i, y_i)$ with uncertain distribution. By applying Gaussian kernel, it converts features to a higher-dimensional feature space where classifier hyper-plane can be computed by solving quadratic function as following.

$$\underset{maximum}{q(c_1, c_2, \ldots, c_n)} = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i c_i k(x_i, x_j) y_j c_j \tag{4.8}$$

$$subject\ to \quad \sum_{i=1}^{n} c_i y_i = 0, and \quad 0 \leq c_i \leq \frac{1}{2n\lambda} \tag{4.9}$$

Here, $k(x_i, x_j)$ represents kernel function that satisfies $k(x_i, x_j) = \varphi(x_i)\varphi(x_j)$. The wights ω and bias b can then be calculated as $\omega = \sum_{i=1}^{n} c_i y_i \varphi(x_i)$ and $b = \omega\dot{\varphi}(x_i) - y_i$ respectively.

In this phase, all subcarriers are used independently for classification since they show similar fluctuations for the same activity [29]. In other words, we can get predicted labels from each subcarrier CSI series. Assume the predicted result of one activity sample is $g = [g(1), g(2), \ldots, g(30 \times N_s)]$, where N_s is the number of CSI streams used for recognition. The final predicted label can be computed with majority voting as following:

$$L = \max_{j \in [1,2,\ldots,n]} \left(\frac{\sum_{i=1}^{30 \times N_s} (g(i) == j)}{30}\right). \tag{4.10}$$

4.2.3 Phase Three

As abundant samples are available in the profile, we propose a deep learning network structure based on LSTM for activity recognition. It is able to automatically extract effective features from raw signals rather than manually selecting as in phase two, which can potentially be subjective in choosing different features [18].

4.2.3.1 Long Short Term Model

LSTM, as one type of recurrent neural network (RNN) units, is able to remember and filter the past information in the input sequence during training process. It is proposed to solve the problem of exploding and vanishing gradient during learning long-term dependencies with back propagation in traditional RNNs. Therefore, it becomes one of the most popular system structure in many areas, including speech identification and sequence classification.

An LSTM block consists of three gates that can be configured to control information through the cell state. The first one is termed forget gate, which decides to how much previous information is removed from the memory. The forget gate vector can be denoted as

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f), \tag{4.11}$$

where σ_g is a sigmoid function, W and U are input and forget weight matrixes, x_t is input vector, h_{t-1} is output vector, and b is bias vector. The black block in the figure represents time delay of the self-loop. Besides, input gate decides the amount of new information allowed to flow into the memory. Input vector can be presented as

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i). \tag{4.12}$$

In addition, output gate decides how much information is filtered to produce the output. Similarly the output gate vector is

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o). \tag{4.13}$$

Then the cell state vector is computed by

$$c_t = f_t c_{t-1} + i_t \sigma_c(W_c x_t + U_c h_{t-1} + b_c). \tag{4.14}$$

Therefore, output vector can be derived as

$$h_t = o_t \sigma_h(c_t), \tag{4.15}$$

where both σ_c and σ_h are hyperbolic tangents.

4.2.3.2 Deep Learning Structure

Based on the LSTM unit as discussed above, we propose to obtain representative features from recorded CSI samples by a multi-layers neural network. As shown on the right side of in Fig.4.3, it is composed of three LSTM hidden layers and one fully connected dense layer. During training process, each LSTM layer learns to filter the information from CSI input samples or the output from last layer. Denote the input CSI sample as X, the output of three LSTM layers can be represented as

$$Y_1 = LSTM(X, \Omega_1), \tag{4.16}$$

$$Y_2 = LSTM(Y_1, \Omega_2), \tag{4.17}$$

$$Y_3 = LSTM(Y_2, \Omega_3), \tag{4.18}$$

where Ω_1, Ω_2, and Ω_3 are set of LSTM parameters in different layers. In this paper, the number of LSTM cells are configured as 128, 64 and 32 in three layers. In this case, the shape of Y_1 and Y_2 will be $(timesteps) \times 128$ and $(timesteps) \times 64$ respectively, where timesteps indicate the length of an activity sample. Differently, the output of the third LSTM layer is the hidden states of the last timestep, which length will be 32. By concatenating the LSTM layers one by one together, the network is capable of learning representatively high level features from collected CSI samples. In order to project the extracted features from the third LSTM layer into activity label probability distribution, a fully connected layer with softmax activation function is added as the output layer as follows.:

$$\widehat{Y} = Softmax(W_l Y_3 + b_l), \tag{4.19}$$

where W_l and b_l are trainable parameters. The predicted label is computed as the corresponding index with the largest probability.

Similar as in phase two, each CSI series from different subcarriers are considered as independent samples in order to enlarge the dataset for training purpose. It is reasonable as we observe that CSI on different subcarriers show similar fluctuation patterns after normalization [29, 32]. Besides, majority voting is applied afterwards the same as depicted in Equation (13).

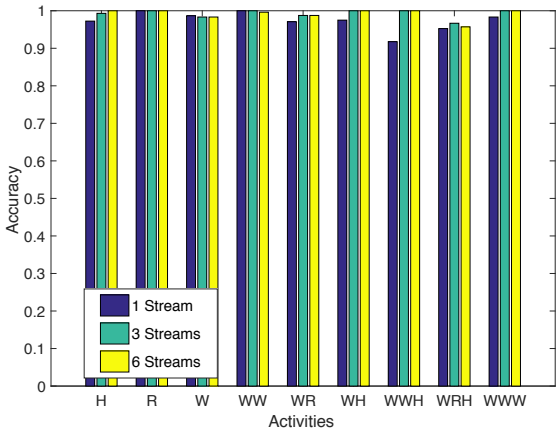
4.3 Experiments and Evaluation

In this section, we evaluate the performance of the proposed three phase design, namely Wi-multi, from a variety of aspects. In order to collect CSI of different activities, we deployed two off-the-shelf wireless devices in our lab, which size is around 6×8 meters. We use a Linksys EA4500 router equipped with 3 antennas as transmitter and a Sony laptop equipped with 2 antennas as receiver. By installing the tool on the Intel 5300 NIC on the laptop, we are able to record 6 CSI streams between different pair of transmit-receive antennas. Considering that the frequency of most human activities is below $10HZ$[50], we configure the CSI sampling rate as $80pkts/s$. Afterwards, we ask 10 volunteers, with different body shape, age and sex, to perform different activities in the lab. Different number of people, from 1 to 3, may be asked to perform activities at the same time. In summary, we collect 936 samples in total. They are composed of 9 different combinations, including Walk (W), Run(R) , Hand Movement (H), W&R, W&W, W&H, W&W&W, W&W&H and W&R&H. Unless otherwise specifically mentioned, half of the samples are used for training and the other half for testing. The detailed experimental results are shown as below.

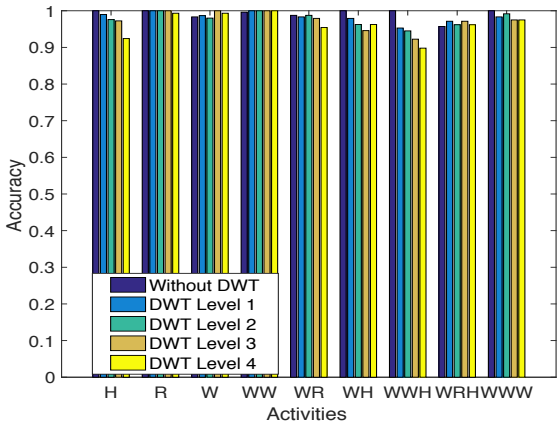
4.3.1 Evaluation of Phase One

Firstly, we compare the accuracies of activity recognition using different numbers of streams. As shown in Fig.4.5a, 1 to 6 streams CSI are exploited to evaluate the impact on accuracies. It is observed that all activities achieve an accuracy above 90% even with only one stream employed. Besides, it is also found that the accuracy slightly increases with more number of streams. For example, the average accuracy rises from 97.43% at 1 stream to 99.23% at 6 streams. This is reasonable since the space diversity of different antennas can provide more diverse information.

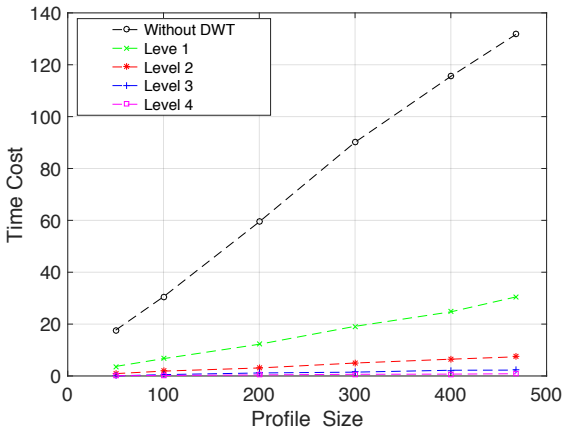
Secondly, we evaluate the impact of different DWT levels on recognition accuracy and efficiency. As shown in Fig.4.5b, the overall accuracy has a slight fall with larger DWT



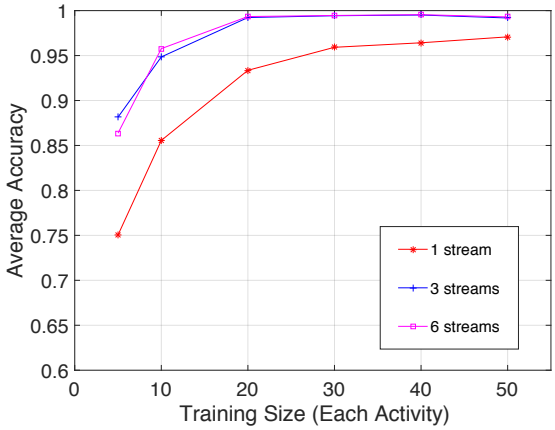
(a) Accuracy comparison with one stream CSI



(b) Accuracy of various DWT levels



(c) Time cost of various DWT levels



(d) Accuracy with different number of profile samples

Figure 4.5: Evaluation of phase one

levels. We can also observe from Fig.4.5c that the time cost of predicting new samples dramatically decreases with larger DWT levels. This is reasonable since DWT can reduce the length of the signal while keeping most of the features and thus reducing time needed for DTW computation. Indeed, the time for predicting new sample is more than 20s without DWT compression, making it nearly impossible for real time applications. On the contrary, the time is dramatically reduced to as low as 0.14s with a 4 level DWT. To achieve a tradeoff between accuracy and efficiency, DWT level 3 seems to be the ideal choice as it achieves

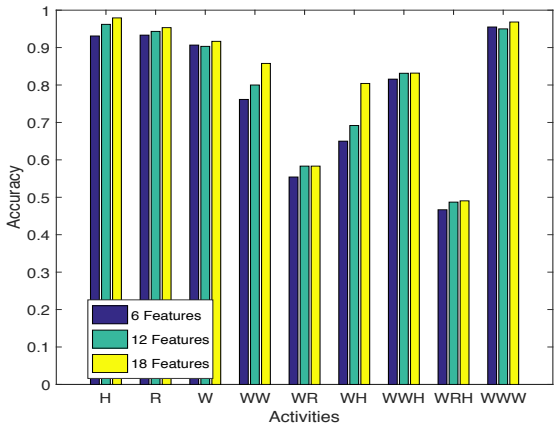
an average accuracy of 97.5% while the time needed for predicting a new sample is only around 0.38s with one stream CSI.

Thirdly, we evaluate the impact of training sample size (each activity) on recognition accuracy. As shown in Fig.4.5d, our scheme achieves desirable accuracy even with only a few number of samples available in the profile. Here the training size refers to the number of samples of each activity in the profile and the result is achieved with 3 level of DWT. As shown in this figure, it achieves an average accuracy of around 85% with only 10 training samples and 1 CSI stream. Moreover, the corresponding prediction time drops to as low 0.078s with the decreasing DWT computation cost. Therefore, we conclude that phase one is able to achieve desirable performance without requiring a large number of training samples. It is suitable at the beginning stage of establishing a system when only few samples can be provided in the profile.

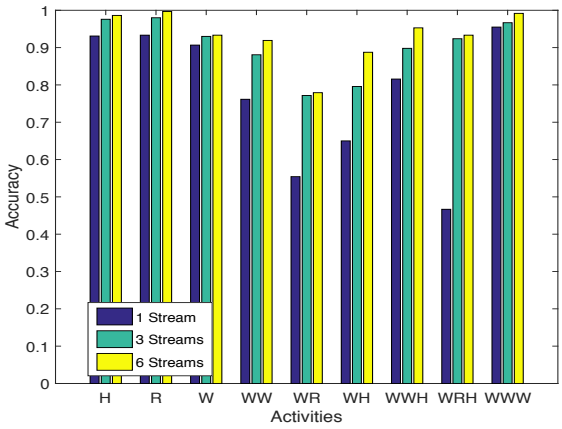
4.3.2 Evaluation of Phase Two

As presented earlier, it is impractical to apply phase one system when there are more samples (say 200 in total) available in the profile, since the computation time cost is over 1s for predicting test data even with one CSI stream. In this case, phase two design can be employed. Since the SVM model can be pre-trained with representative features, the time cost for predicting test samples is usually constant. We evaluate the system of phase two from following aspects.

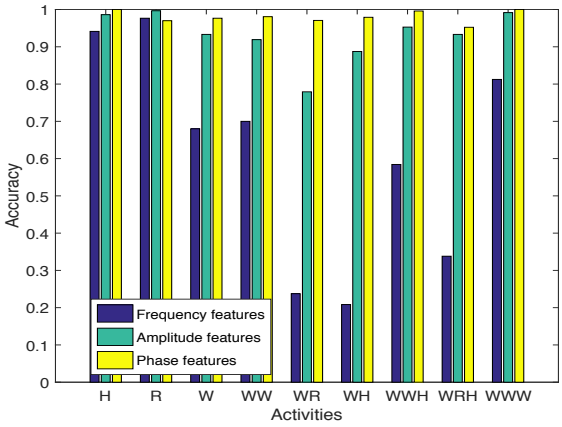
Firstly, we evaluate the impact of features numbers. In this paper, we compare the results of 6, 12, and 18 features respectively. These amplitude features include standard deviation, median absolute deviation, max, mean, first and third quartile of the filtered CFR power with different cut-off frequencies. As shown in Fig. 4.6a, it demonstrates the accuracy results under different number of features. For instance, the accuracy of recognizing W&H rises from 65% to 80.42% when the number of features increases from 6 to 18. In



(a) Impact of features numbers on accuracy



(b) Impact of stream numbers on accuracy



(c) Impact of feature metrics on accuracy

Figure 4.6: Evaluation of phase two

general, the average accuracy increases from 79.19% with 6 amplitude features to 82.87% with 18 amplitude features. Because of the limited space of this paper, we only present the result using one CSI stream here. Similar results can be observed with more CSI streams.

Secondly, we evaluate the impact of CSI stream numbers. As shown in Fig.4.6b, we compare the results of activity recognition accuracy with different number of CSI streams. Similar as shown in Section 4.3.1, it is observed that more CSI streams can be utilized in order to achieve higher accuracy. For example, the accuracy of recognizing W&R&H climbs from 46.67% to 93.33% when the number of CSI streams increases from 1 to 6. Be-

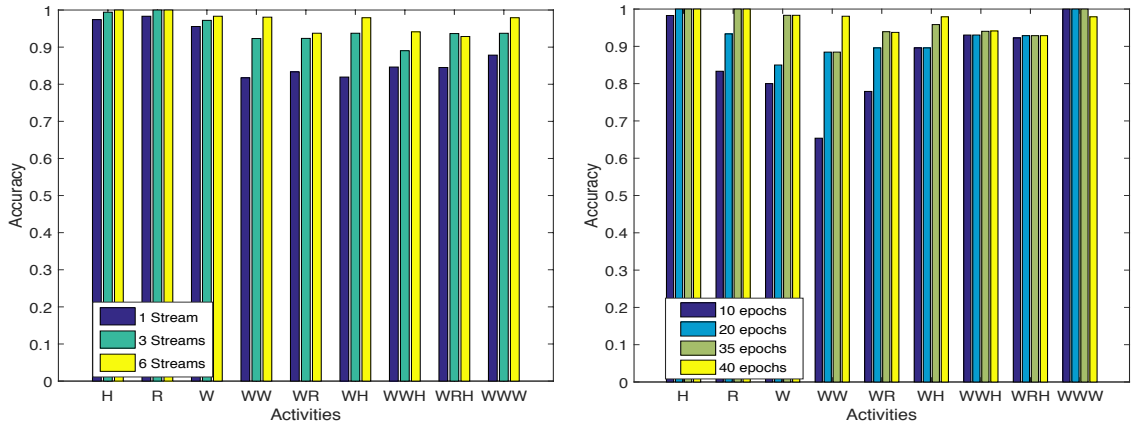
sides, the average accuracy of all activities is 79.19% with 1 CSI stream, which is 14.31% less than the average accuracy with 6 CSI streams. As we discussed in Section 3, each CSI stream is collected from 30 subcarriers between one pair of transmit-receive antennas. Since different pairs of antennas provide spatial diversities, it is reasonable that higher accuracy can be achieved with more CSI streams.

Thirdly, we evaluate the impact of different feature metrics. As discussed in Section 4.2.2, we select 6 different features from amplitude, phase and frequency domains respectively. As shown in Fig.4.6c, it compares the accuracy results achieved by different feature metrics. Besides, we observe that results are similar using different number of CSI streams. Because of space limitation, we only present the results using 6 CSI streams in this paper. It is shown from Fig.4.6c that the accuracies achieved by frequency domain features are much lower than the other two. For instance, the accuracy of recognizing W&R using frequency domain features is 23.75%, which is much lower compared with 77.92% of amplitude features and 97.08% of phase features. In general, the average accuracy with frequency domain features is 63.13%, which is 30.27% and 34.99% less than accuracy with amplitude and phase features respectively. This is reasonable considering that some activities (such as W&R and W&R&H) may have similar intensity of movement, which results in alike patterns of DWT energy level.

4.3.3 Evaluation of Phase Three

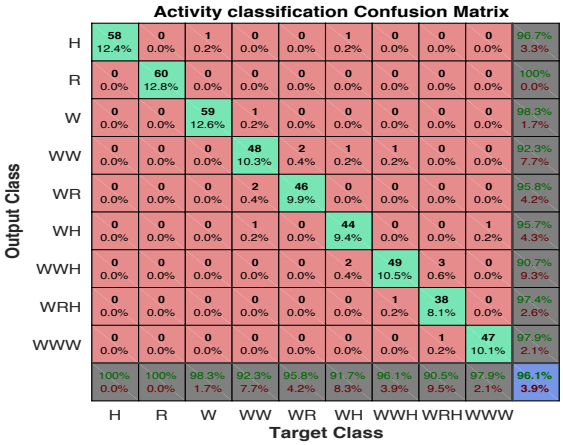
As depicted in Section 4.3.2, it can be subjective to select different features. With increasing numbers of samples in the profile, phase three based on deep learning network is exploited by automatically extracting representative features. The experimental results are shown as below.

Firstly, we evaluate the impact of different stream numbers. Similar as shown in Section 4.3.1 and Section 4.3.2, we observe that the accuracies with more number of streams



(a) Impact of stream numbers on accuracy

(b) Impact of iteration numbers



(c) Performance of deep learning structure

Figure 4.7: Evaluation of deep learning system

are much higher. As shown in Fig.4.7a, the accuracy of recognizing W&W with 6 CSI streams is 98.08%, which is 16.35% and 5.98% higher than accuracy with 1 CSI stream and 3 CSI streams respectively. Overall, the average accuracy increases from 88.36% to 97.22% when the number of CSI streams increases from 1 to 6.

Secondly, we evaluate the impact of epoch numbers. Because of the limitation of hardware, the input training samples are usually divided into small batches, which go through the network one by one. One epoch is the process of passing the entire training dataset (all batches) once through the neural network. It is known that different num-

ber of epochs may cause overfitting and underfitting of the trained model. As shown in Fig.4.7b, we compare the classification accuracies of models trained with different number of epochs. It is observed that the accuracy increases with more epochs and becomes stable after 35 epochs. This is reasonable since it requires enough epochs to update the network parameters and train a robust model.

Lastly, we present the detailed evaluation result of phase three in Fig.4.7c. We configure the number of CSI streams as 6, the number of epochs as 35 and the batch size as 64. From this confusion matrix, it is observed that the overall accuracy of phase three is 96.1%, which is higher than 95.18% of phase one and 93.4% of phase two. Compared with phase one, it requires no additional time cost when more training samples are available in the profile. This is because the model can be pre-trained and does not require to compute similarities between test sample and all training samples. Moreover, compared with phase two, it is able to automatically extract effective features. In conclusion, phase three of deep learning networks achieves better performance when abundant samples are collected.

4.4 Discussion

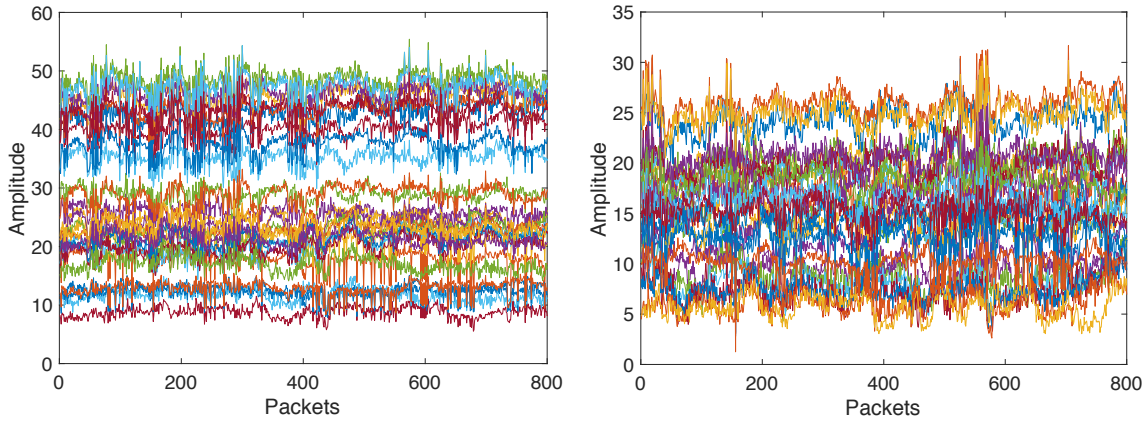
Targeting at multiple human activity recognition, in this Chapter we propose Wi-multi, a three-phase system using channel state information. At the initial stage of system deployment, it is infeasible to apply machine learning algorithms as there are usually few samples available in the profile. In this case, our designed phase one of the system that utilizes distance-based classification is exploited. As more samples become available for training, phase two of our design that employs SVM with representative features from both time and frequency domain is applied. It dramatically reduces computation costs as compared with phase one which requires computing similarities between the test sample and all samples in the profile. Finally, when we have enough samples for deep learning

networks, phase three based on LSTM is proposed. It can achieve higher accuracy and efficiency since it can automatically choose representative features and pre-train the model. Given the availability of samples, each phase of our design achieves a desirable tradeoff between accuracy and efficiency.

CHAPTER 5

Location Independent Gesture Identification

5.1 Challenges



(a) Push in location A

(b) Push in location B

Figure 5.1: CSI of push gesture in different locations

Due to multipath phenomenon, gestures performed in different locations can lead to distinct reflections on various paths. As shown in Fig. 5.1, we present the comparison of one CSI stream (30 subcarriers) between one pair of transmit-receive antennas. It can be observed that the same push gesture in two locations exhibits significantly different CSI fluctuation patterns. This presents significantly challenges to apply the machine learning model trained with gestures performed in one location to different locations.

Existing works have achieved desirable performance for gesture detection if the subject remain at the same location [12, 10]. It indicates that CSI of the same gesture show similar patterns at the same location, showing that the collected CSI contains gesture de-

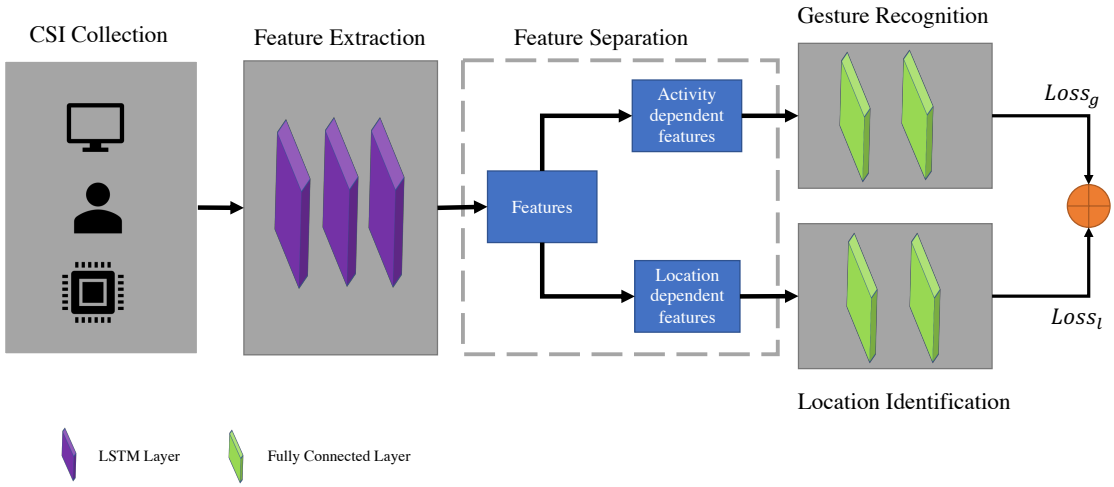


Figure 5.2: System structure

pendent features. At the same time, as we discussed earlier, gestures performed at different locations can exhibit distinct CSI fluctuations, showing that raw CSI also contains location dependent features. As the location dependent features are interleaved with gesture dependent features, the challenge becomes separating them effectively in order to correctly identify the gestures at different locations. Different from traditional deep learning systems that extract high level mixed representative features for gesture recognition, we propose a deep learning network that can extract both gesture and location dependent features, while capable of separating them effectively and independently by applying gesture recognition and location identification modules simultaneously.

5.2 System Overview

In this paper, our goal is to separate gesture dependent features from location dependent features. In other words, the system should be able to extract gesture dependent features only at a new location. Towards this goal, we propose a system that consists of five modules, including CSI collection, feature extraction, feature separation, gesture recognition and location identification. Fig. 5.2 depicts the overview of the system structure.

CSI Collection: As discussed earlier, CSI is affected by both gesture movements and the locations. In order to cluster these two kinds of features separately, the proposed deep learning network requires adequate samples for training purpose. Therefore, we collect CSI of various gestures from different locations by leveraging tools on wireless devices [8] so as to feed them into the network.

Feature Extraction: In this component, a three LSTM layers network is designed to obtain high level feature representations. The input of this component is the collected CSI samples from various gestures performed by different subjects at multiple locations (not necessarily at the targeted location). Besides, the extracted features from the hidden states of last timestep at the last layer contain both gesture and location dependent feature representations.

Feature Separation: After obtaining the extracted features, we divide them into two halves and feed them into the gesture recognition and location identification module respectively.

Gesture Recognition: In order to recognize gestures based on the first half extracted features, we employ two fully connected layers to map feature representations to a new latent space for classification. By minimizing the loss of this module, the first half features will be gradually clustered as gesture dependent.

Location Identification: Similarly, two fully connected layers are utilized to map the second half extracted features to a new space for location classification. During the training process, the network will cluster the second half feature representations as location dependent.

In general, the overall loss function used for training is computed as the sum of the losses of gesture recognition and location identification. By decreasing the overall loss, the proposed network system will learn to gradually separate the first half features as gesture dependent and the second half as location dependent.

5.2.1 CSI Collection

In oder to provide enough samples for system training, CSI of various gestures performed by different subjects are collected from multiple locations. Denote the input CSI sample as $\boldsymbol{X}_i \in \{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N\}$, where N is the number of collected CSI samples. Each CSI sample is associated with a gesture label and a location label. The corresponding gesture label can be represented as $L_i^g \in \{0, \ldots, n-1\}$, where n is the number of gesture types. The corresponding location label can be represented as $L_i^l \in \{0, 1, \ldots, m-1\}$, where m is the number of locations. During the training process, CSI samples with corresponding gesture and location labels are then fed into the deep learning network. After the model is trained, CSI samples with ground truth gesture labels from new locations are provided to evaluate the performance.

5.2.2 Feature Extraction

5.2.2.1 Long short-term memory

LSTM [51] is one of the recurrent neural network (RNN) structures that is widely employed in time series data classification and prediction. It has the ability to remember and filter information over time through three gates. The first gate, known as forget gate, is designed to filter past information. It determines the amount of information kept in the cell state. The second gate, known as input gate, controls rgw amount of new information from current timestep to be added into the cell state. The third gate, known as output gate, determines the amount of information used for output. In addition, LSTM also solves the long-term dependency problem [52], where gradient usually either explodes or vanishes in conventional RNNs.

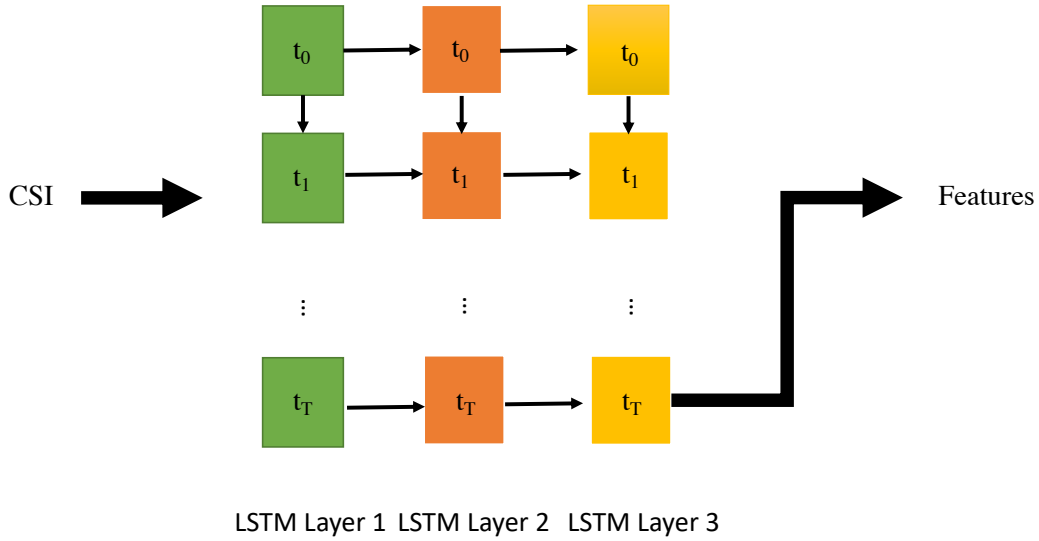


Figure 5.3: Three LSTM layers network

In this paper, we endeavor to recognize gestures performed at different locations with CSI. As CSI is highly correlated time series signal, we employ LSTM to extract representative features.

5.2.2.2 Three LSTM layers network

Fig. 5.3 depicts the three LSTM layers network structure used to extract feature representations in this paper. Denote the size of each CSI sample as $T \times S$, where T is the length of CSI and $S = (P * 30)$, where P is the number of antenna pairs used to collect CSI and 30 is the number of subcarriers in each stream. In the first LSTM layer, there are one cell for states initialization and T other cells, each of which is corresponding to a timestep CSI data. Each cell delivers its current cell states and hidden states to the next cell while also outputs the hidden states at every timestep. Similarly, the second and third LSTM layers are also equipped with $T + 1$ cells, each of which takes the corresponding output from the previous layer as input. Let C_i be the size of hidden states in each cell at the i_{th}

layer. The size of the output at the last LSTM layer can be described as $T \times C_3$. Afterwards, we employ the hidden states in the last timestep as the extracted feature representations. Given the input CSI as $\boldsymbol{X}_i$, the corresponding extracted features can be represented as

$$\boldsymbol{F}_i = \boldsymbol{LSTM}_3(\boldsymbol{LSTM}_2(\boldsymbol{LSTM}_1(\boldsymbol{X}_i, \boldsymbol{\zeta}))), \tag{5.1}$$

where $\boldsymbol{\zeta}$ denotes the parameters of the LSTM network.

5.2.3 Feature Separation

After the feature representations are extracted from the LSTM network, we try to separate the gesture dependent features from location dependent features. In order to achieve this, we divide the extracted features into two halves. Assume that the extracted features are $\boldsymbol{F} = [f_1, f_2, \ldots, f_{C_3}]$. The first half of the features can be denoted as $\boldsymbol{F^g} = [f_1, f_2, \ldots, f_{C_3/2}]$ while the second half can be represented as $\boldsymbol{F^l} = [f_{(C_3/2+1)}, f_{(C_3/2+2)}, \ldots, f_{C_3}]$. The first half features are delivered to the gesture recognition module for classification while the second half is fed into the location identification module.

5.2.4 Gesture Recognition

As shown in Fig. 5.4, we employ two fully connected layers in order to classify different gestures and gradually cluster the first half extracted features as gesture dependent. Specifically, the first layer followed by an activation function is used to forward and map the first half features into a new feature space. Given the CSI input as X_i and the corresponding first half features as $\boldsymbol{F_i^g}$, the output can then be represented as

$$\boldsymbol{R_i} = softplus(\boldsymbol{W_g}\boldsymbol{F_i^g} + \boldsymbol{b_g}), \tag{5.2}$$

where $softplus$ is the activation function and $\boldsymbol{W_g}$ and $\boldsymbol{b_g}$ are the weights and bias respectively.

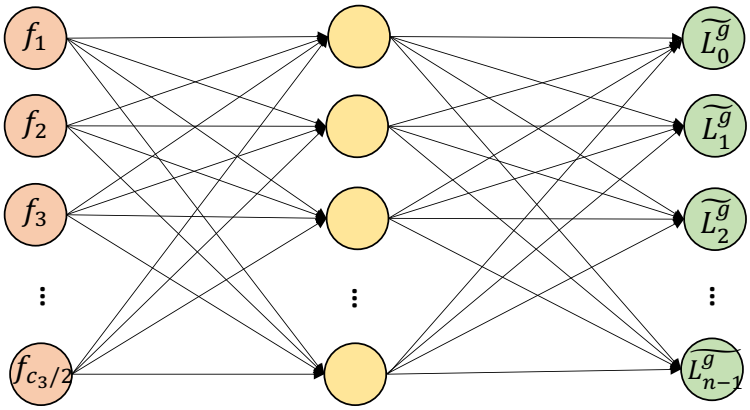


Figure 5.4: Two fully connected layers in gesture recognition

Assuming that there are n types of gestures, we map R_i into an n dimensional space for classification by introducing another fully connected layer. Therefore, the output can be described as

$$\widetilde{\boldsymbol{L_i^g}} = softmax(\boldsymbol{W_h R_i} + \boldsymbol{b_h}), \tag{5.3}$$

where $softmax$ is an activation function that computes the probability distribution for each class and $\boldsymbol{W_h}$ and $\boldsymbol{b_h}$ are the weights and bias respectively. The final predict label is usually the index whose corresponding probability is the largest in $\widetilde{\boldsymbol{L_i^g}}$.

In addition, cross entropy is used as the loss function in this module. It can be computed as

$$Loss_g = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{n-1} L_{ij}^g log(\widetilde{L_{ij}^g}), \tag{5.4}$$

where N is the number of CSI samples and n is the number of gesture types. By reducing $Loss_g$ during the training process, the network will gradually learn to cluster the first half features as gesture related.

5.2.5 Location Identifcation

Similar to 5.2.4, two fully connected layers are exploited so as to classify locations and cluster the second half extracted features as location dependent. Provided that the second half extracted features corresponding to input CSI X_i is F_i^l, the outputs of the first and second layer can be computed as

$$\boldsymbol{O_i} = softplus(\boldsymbol{W_l F_i^l} + \boldsymbol{b_l}), \tag{5.5}$$

$$\widetilde{\boldsymbol{L_i^l}} = softmax(\boldsymbol{W_z R_i} + \boldsymbol{b_z}), \tag{5.6}$$

where $\boldsymbol{W_l}$ and $\boldsymbol{b_l}$ are weights and bias in the first layer, $\boldsymbol{W_l}$ and $\boldsymbol{b_l}$ are weights and bias in the second layer respectively.

Assuming there are m locations, the cross entropy loss function in this module can be defined as

$$Loss_l = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=0}^{m-1} L_{ij}^l log(\widetilde{L_{ij}^l}). \tag{5.7}$$

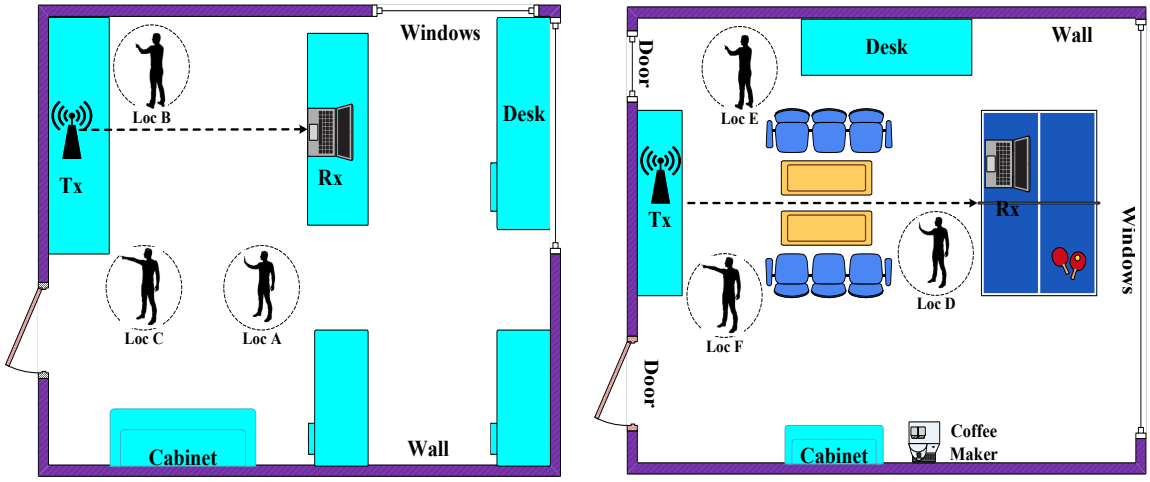
By decreasing $Loss_l$ during the training process, the network will learn to cluster the second half features as highly location related.

5.2.6 Loss Optimization

Overall, our goal is to separate gesture dependent features from location dependent features. Based on equations 5.4 and 5.7, we define the overall loss as

$$Loss = Loss_g + \lambda Loss_l, \tag{5.8}$$

where λ is the coefficient that controls the balance between gesture recognition and location identification losses. By optimizing $Loss$, the proposed deep learning network will learn to classify gestures and locations simultaneously with different halves of the extracted



(a) Room 1 (b) Room 2

Figure 5.5: Layout of experiment environments

features. On one hand, in order to classify gestures, the first half features will be gradually clustered as gesture dependent features. On the other hand, the second half extracted features will be clustered as location dependent so as to classify locations. As a result, the system will be able to learn to separate gesture dependent features from location dependent features by reducing the overall loss function.

5.3 Experiments and Evaluation

We have implemented our system on commercially available devices. CSI of different gestures performed by various subjects in different locations are collected. We then implement the proposed deep learning network with Tensorflow [53] on Google Cloud Platform [54]. The details of the system implementation and results are described below.

5.3.1 Data Collection

We collect CSI samples from off-the-shelf Wi-Fi devices. One router that acts as the transmitter and one laptop that acts as the receiver are deployed in different environments.

Table 5.1: Number of samples collected

Gestures / ***Locations***	*Bounce*	*Handwave*	*Push*	*Swipe*
A	251	298	242	292
B	239	298	346	288
C	249	246	286	261
D	265	275	245	250
E	287	298	241	253
F	272	289	262	262

The router is equipped with 3 antennas while the laptop has 1 antenna. By implementing the tool [8] on Intel 5300 NIC, we can obtain $3 * 30$ CSI values at each time point. We ask multiple subjects to perform gestures in 6 different locations that are in 2 rooms with different layouts and furniture decorations. Fig 5.5 depicts the layout of the two rooms and different CSI collection locations. Each subject is asked to perform 4 gestures, including bounce, handwave, push and swipe, at each location. Each gesture sample lasts 4 seconds and the CSI sampling rate is set to 200 $packets/second$, thus each CSI gesture sample has 90×800 values in total. In summary, we collect 6495 CSI samples from various locations. Detailed number of collected samples can be found in Table 5.1.

5.3.2 Evaluation

5.3.2.1 Accuracy using CSIs from all locations for training

We first present the results achieved by using data from all locations for training. In other words, we split the collected samples from all locations into two halves, one of which is used for training while the other is used for evaluation. During the training process, each provided CSI gesture sample is associated with one gesture label and one location label. Note that we only evaluate the performance of gesture recognition. As shown in Fig. 5.6, the system achieves 95.11% accuracy on average. Specifically, it achieves an

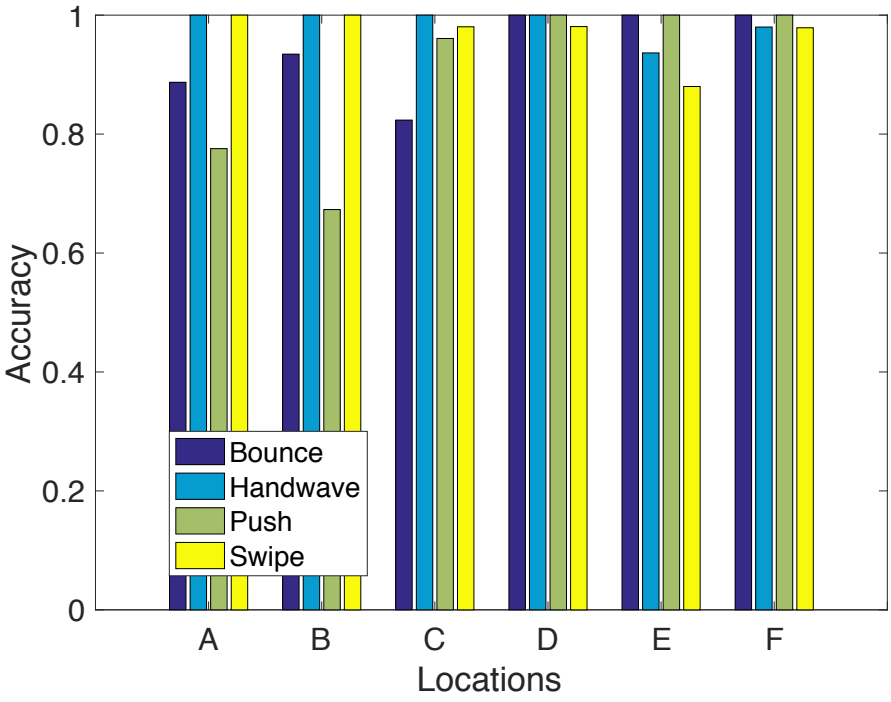


Figure 5.6: Accuracy using all location samples for training

accuracy of 90.99%, 90.00%, 93.99%, 99.50%, 95.50% and 99.00% at locations from A to F respectively. In general, it achieves above 90.00% accuracy at all locations. In addition, as shown in Fig. 5.7, the testing loss of gesture recognition show similar patterns as the training losses of gesture recognition and location identification. As a result, our system is able to achieve desirable performance in recognizing gestures performed at different locations.

5.3.2.2 Accuracy using different locations for training

Here, we evaluate the proposed system's performance for identifying gestures in locations with no training data employed a prior. Fig. 5.8 shows that the gesture recognition accuracy in each location when CSI samples from other locations are used for training. Note that the CSI samples from the targeted location are never used for training. For

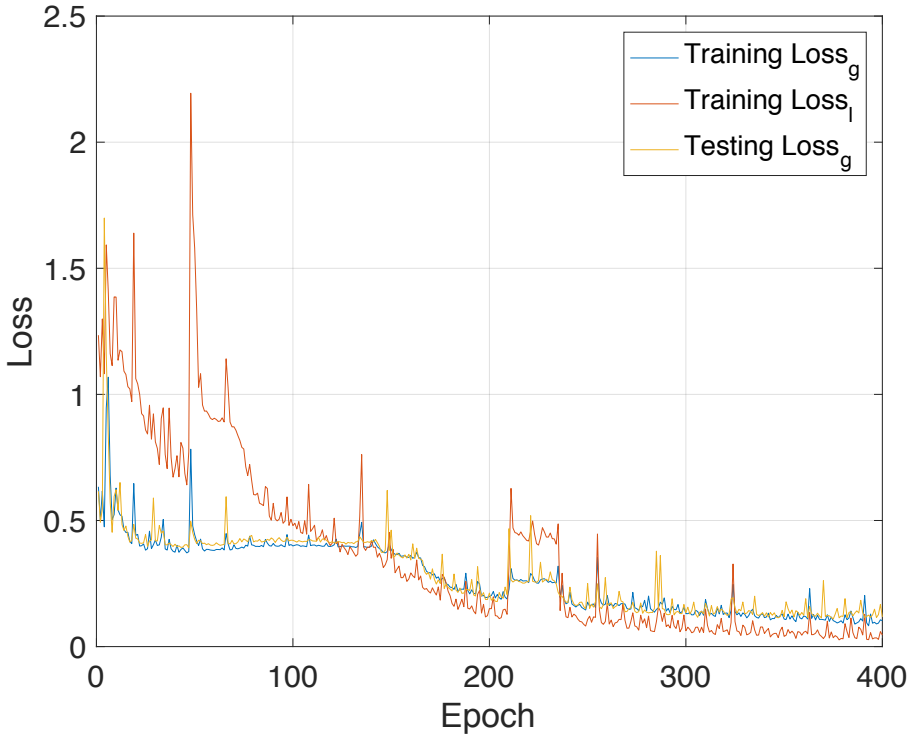


Figure 5.7: Losses during training and testing process

example, the results for location A denotes the accuracy when the model is trained using CSI samples from the other 5 locations.

As shown in the figure, the system achieves desirable average accuracy in different locations. For instance, it achieves 79.20% accuracy on average for location A. Specifically, it achieves 52.08% accuracy in recognizing bounce, 100.00% in recognizing handwave, 66.94% in recognizing push and 100.00% in recognizing swipe. The average accuracy for location B is higher as illustrated in Fig. 5.8. It is able to recognize bounce with an accuracy of 72.12%, handwave with an accuracy of 100%, push with an accuracy of 91.29% and swipe with an accuracy of 100.00% while the overall average accuracy is 90.38%. Similar results can be found for identifying the gestures performed in the other room. For example, the system achieves 84.67% accuracy on average in location D while the average accuracy in location E is around 87.39%. Similar results can also be observed for other locations.

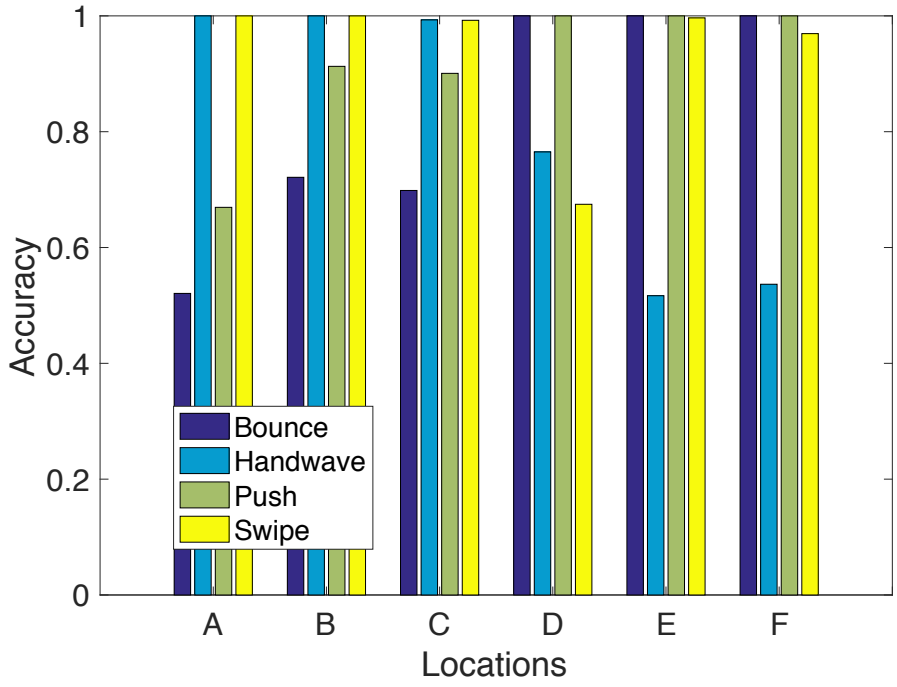


Figure 5.8: Accuracies using other locations for training

In summary, our proposed system is able to recognize gestures performed in new locations without training new models.

5.3.2.3 Different number of locations used for training

Here, we randomly select $k \in \{2, 3, 4, 5\}$ locations from collected CSI samples for training and use the rest for evaluation. Fig.5.9 illustrates the average accuracy with different number of locations used for training. Each average accuracy corresponding to one k value is obtained by 5-fold cross validation. It is observed, from Fig.5.9, that higher average accuracy can be achieved with more location data used for training. Specifically, the average accuracy increases from 58.67% with 2 training locations to 85.42% with 5 training locations. The average accuracy using 3 and 4 locations for training are 67.84% and 75.09% respectively. This is reasonable as more training locations can provide more diversities to the network and consequently gain better evaluation performance. It is also

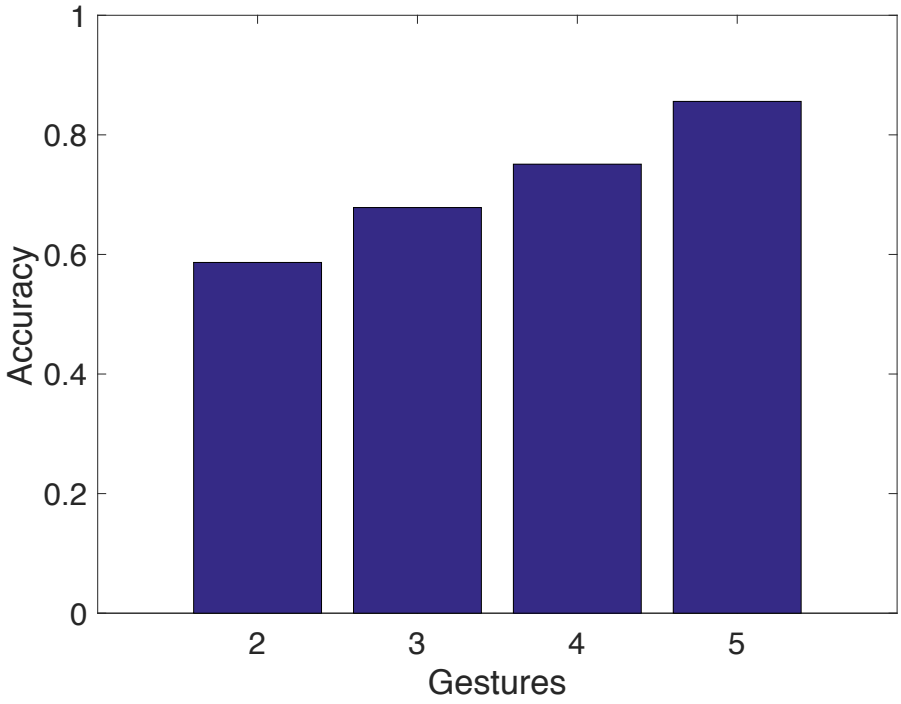


Figure 5.9: Average accuracy using different number of locations for training

observed that the accuracy is undesirable in some cases, where we use samples collected in one room for training and samples of the other room for testing. Therefore, our proposed system will benefit from training at more locations in order to achieve desirable performance.

5.4 Discussion

Although many existing works have achieved desirable performance in gesture recognition with channel state information, they generally cannot be directly used to recognize gestures performed in new locations without training new models. In order to address this, we design a deep learning neural network that can achieve location independent gesture recognition. The system includes feature extraction, feature separation, gesture recognition and location identification modules. Specifically, a three LSTM layers network is designed

to extract both gesture and location dependent features. Afterwards, we split them into two halves where the first half is delivered to gesture recognition module and the second half is passed to the location identification module. By optimizing the total loss of both modules, the network will learn to gradually cluster and separate these features as gesture dependent and location dependent. We evaluate the proposed system by collecting the CSI samples from various subjects who are asked to perform 4 gestures in 6 locations of 2 rooms. The proposed system achieves 85.42% accuracy on average in recognizing gestures performed in new locations.

While our proposed network can achieve desirable performance, the following challenges remain and we consider them our future work. It is well known that current off-the-shelf WiFi devices can cover more than 3000 square feet in an indoor environment. It is likely, in practice, that people in the same environment other than the subject can also potentially affect the multipath propagations. It will be significantly helpful if one can eliminate the effects owing to other people while identifying the gestures performed by the targeted subject. Related, it will be ideal to be able to recognize multiple human gestures performed at the same locations at the same time and identify individual gesture embedded therein. Indeed, the authors in [43] attempt to address this by creating virtual combination samples from a single user. However, it still unclear on how to identify each of the individuals from the combined gestures.

CHAPTER 6

Conclusion

Many challenges in movement recognition with channel state information remain unsolved, although it has attracted large attentions in recent years and many existing works have achieved desirable performance in different applications. Specifically, previous researches generally ignore the fact of location dependent nature of channel state information. Therefore, many existing systems trained with machine learning algorithms would require to be retrained in each new location. However, they usually fail to consider the availability of enough samples, especially at the early stage of system deployment in new locations. Additionally, since location dependent features and movement dependent features are interleaved with each other in CSI, how to effectively separate them becomes the main challenge in order to recognize activities correctly at different locations without training new models.

In this paper, we firstly propose a three phase system that targets at multiple human activity recognition with channel state information. At the early stage of system deployment where only few samples are available in the profile, our designed phase one of the system that utilizes distance-based classification is exploited. As more samples become available for training, phase two of our design that employs SVM with representative features from both time and frequency domain is applied. It dramatically reduces computation costs as compared with phase one which requires computing similarities between the test sample and all samples in the profile. Finally, when we have enough samples for deep learning networks, phase three based on LSTM is proposed. It can achieve higher accuracy and efficiency since it can automatically choose representative features and pre-train the

model. Given the availability of samples, the experiments results illustrate that each phase of our design achieves a desirable tradeoff between accuracy and efficiency. In general, Wi-multi can achieve 96.1% accuracy on average.

In addition, we also propose a neural network that can recognize gestures performed in new locations without traning new models. In other words, the proposed system is able to separate gesture relevant features from location relevant features. The system consists of four modules, including feature extraction, feature separation, gesture recognition and location identification. In details, feature extraction selects representative features by taking advantage of a three layers LSTM network. Next, the feature separation moudle delivers half of the features to gesture recognition and the other half to location identification. By minimizing the overall loss of the gesture recognition and location identification modules during the training process, the network learns to cluster the first half extracted features as highly gesture related representations while the second half as intensively location related representations. Our evalutions show that the proposed system achieves an average of 85.42% accuracy of gesture recognition in new locations.

REFERENCES

[1] K. Wu, J. Xiao, Y. Yi, M. Gao, and L. M. Ni, "Fila: Fine-grained indoor localization," in *2012 Proceedings IEEE INFOCOM*. IEEE, 2012, pp. 2210–2218.

[2] B. Kellogg, V. Talla, and S. Gollakota, "Bringing gesture recognition to all devices," in *11th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 14)*, 2014, pp. 303–316.

[3] X. Huang and M. Dai, "Indoor device-free activity recognition based on radio signal," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5316–5329, 2016.

[4] K. Yatani and K. N. Truong, "Bodyscope: a wearable acoustic sensor for activity recognition," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, 2012, pp. 341–350.

[5] R. Bodor, B. Jackson, and N. Papanikolopoulos, "Vision-based human tracking and activity recognition," in *Proc. of the 11th Mediterranean Conf. on Control and Automation*, vol. 1. Citeseer, 2003.

[6] S. Arshad, C. Feng, I. Elujide, S. Zhou, and Y. Liu, "Safedrive-fi: A multimodal and device free dangerous driving recognition system using wifi," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[7] S. Arshad, C. Feng, R. Yu, and Y. Liu, "Leveraging transfer learning in multiple human activity recognition using wifi signal," in *2019 IEEE 20th International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*. IEEE, 2019, pp. 1–10.

[8] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, "Tool release: gathering 802.11 n traces with channel state information," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 1, pp. 53–53, 2011.

[9] K. Ali, A. X. Liu, W. Wang, and M. Shahzad, "Keystroke recognition using wifi signals," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 90–102.

[10] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 1472–1480.

[11] X. Liu, J. Cao, S. Tang, and J. Wen, "Wi-sleep: Contactless sleep monitoring via wifi signals," in *Real-Time Systems Symposium (RTSS), 2014 IEEE*. IEEE, 2014, pp. 346–355.

[12] S. Tan and J. Yang, "Wifinger: leveraging commodity wifi for fine-grained finger gesture recognition," in *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing*. ACM, 2016, pp. 201–210.

[13] Z. Tian, J. Wang, X. Yang, and M. Zhou, "Wicatch: A wi-fi based hand gesture recognition system," *IEEE Access*, vol. 6, pp. 16 911–16 923, 2018.

[14] W. Wang, A. X. Liu, and M. Shahzad, "Gait recognition using wifi signals," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 2016, pp. 363–373.

[15] H. Lee, C. R. Ahn, N. Choi, T. Kim, and H. Lee, "The effects of housing environments on the performance of activity-recognition systems using wi-fi channel state information: An exploratory study," *Sensors*, vol. 19, no. 5, p. 983, 2019.

[16] J. Wang, L. Zhang, Q. Gao, M. Pan, and H. Wang, "Device-free wireless sensing in complex scenarios using spatial structural information," *IEEE Transactions on Wireless Communications*, 2018.

[17] X. Huang, S. Guo, Y. Wu, and Y. Yang, "A fine-grained indoor fingerprinting localization based on magnetic field strength and channel state information," *Pervasive and Mobile Computing*, vol. 41, pp. 150–165, 2017.

[18] C. Feng, S. Arshad, R. Yu, and Y. Liu, "Evaluation and improvement of activity detection systems with recurrent neural network," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[19] C. Feng, S. Arshad, S. Zhou, D. Cao, and Y. Liu, "Wi-multi: A three-phase system for multiple human activity recognition with commercial wifi devices," *IEEE Internet of Things Journal*, 2019.

[20] O. Amft, H. Junker, and G. Troster, "Detection of eating and drinking arm gestures using inertial body-worn sensors," in *Wearable computers, 2005. proceedings. ninth ieee international symposium on*. IEEE, 2005, pp. 160–163.

[21] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, "A framework for hand gesture recognition based on accelerometer and emg sensors," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 41, no. 6, pp. 1064–1076, 2011.

[22] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using kinect sensor," *IEEE transactions on multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.

[23] "Apple watch." https://www.apple.com/watch/.

[24] "Fitbit." https://www.fitbit.com/home.

[25] A. D. Wilson, "Robust computer vision-based detection of pinching for one and two-handed gesture input," in *Proceedings of the 19th annual ACM symposium on User interface software and technology*. ACM, 2006, pp. 255–258.

[26] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 11, pp. 3592–3607, 2011.

[27] N. H. Dardas and E. M. Petriu, "Hand gesture detection and recognition using principal component analysis," in *Computational Intelligence for Measurement Systems and Applications (CIMSA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 1–6.

[28] Q. Chen, N. D. Georganas, E. M. Petriu *et al.*, "Real-time vision-based hand gesture recognition using haar-like features," in *Instrumentation and Measurement Technology Conference Proceedings*. Citeseer, 2007, pp. 1–6.

[29] C. Feng, S. Arshad, and Y. Liu, "Mais: Multiple activity identification system using channel state information of wifi signals," in *International Conference on Wireless Algorithms, Systems, and Applications*. Springer, 2017, pp. 419–432.

[30] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: device-free location-oriented activity identification using fine-grained wifi signatures," in *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM, 2014, pp. 617–628.

[31] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 2015, pp. 65–76.

[32] S. Arshad, C. Feng, Y. Liu, Y. Hu, R. Yu, S. Zhou, and H. Li, "Wi-chase: A wifi based human activity recognition system for sensorless environments," in *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. IEEE, 2017, pp. 1–6.

[33] C. Han, K. Wu, Y. Wang, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 271–279.

[34] Y. Wang, K. Wu, and L. M. Ni, "Wifall: Device-free fall detection by wireless networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 581–594, 2017.

[35] H. Wang, D. Zhang, Y. Wang, J. Ma, Y. Wang, and S. Li, "Rt-fall: a real-time and contactless fall detection system with commodity wifi devices," *IEEE Transactions on Mobile Computing*, vol. 16, no. 2, pp. 511–526, 2017.

[36] X. Zheng, J. Wang, L. Shangguan, Z. Zhou, and Y. Liu, "Smokey: Ubiquitous smoking detection with commercial wifi infrastructures," in *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*. IEEE, 2016, pp. 1–9.

[37] "Average size of households in the u.s. 1960-2015," https://www.statista.com/statistics/183648/average-size-of-households-in-the-us/.

[38] O. J. Kaltiokallio, H. Yigitler, R. Jäntti, and N. Patwari, "Non-invasive respiration rate monitoring using a single cots tx-rx pair," in *Proceedings of the 13th international symposium on Information processing in sensor networks*. IEEE Press, 2014, pp. 59–70.

[39] M. Li, Y. Meng, J. Liu, H. Zhu, X. Liang, Y. Liu, and N. Ruan, "When csi meets public wifi: Inferring your mobile phone password via wifi signals," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1068–1079.

[40] G. Wang, Y. Zou, Z. Zhou, K. Wu, and L. M. Ni, "We can hear you with wi-fi!" *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2907–2920, 2016.

[41] H. Abdelnasser, K. A. Harras, and M. Youssef, "A ubiquitous wifi-based fine-grained gesture recognition system," *IEEE Transactions on Mobile Computing*, 2018.

[42] Q. Zhou, J. Xing, W. Chen, X. Zhang, and Q. Yang, "From signal to image: Enabling fine-grained gesture recognition with commercial wi-fi devices," *Sensors*, vol. 18, no. 9, p. 3142, 2018.

[43] 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*. ACM, 2018, pp. 401–413.

[44] A. Virmani and M. Shahzad, "Position and orientation agnostic gesture recognition using wifi," in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 2017, pp. 252–264.

[45] Q. Zhou, J. Xing, J. Li, and Q. Yang, "A device-free number gesture recognition approach based on deep learning," in *Computational Intelligence and Security (CIS), 2016 12th International Conference on*. IEEE, 2016, pp. 57–63.

[46] Y. Shu, Y. Huang, J. Zhang, P. Coué, P. Cheng, J. Chen, and K. G. Shin, "Gradient-based fingerprinting for indoor localization and tracking," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2424–2433, 2016.

[47] D. Zhang, J. Ma, Q. Chen, and L. M. Ni, "An rf-based system for tracking transceiver-free objects," in *Fifth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'07)*. IEEE, 2007, pp. 135–144.

[48] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas *et al.*, "Towards environment independent device free human activity recognition," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 289–304.

[49] Z. Zhou, Z. Yang, C. Wu, W. Sun, and Y. Liu, "Lifi: Line-of-sight identification with wifi," in *IEEE INFOCOM 2014-IEEE Conference on Computer Communications*. IEEE, 2014, pp. 2688–2696.

[50] Y. Zeng, P. H. Pathak, and P. Mohapatra, "Wiwho: wifi-based person identification in smart spaces," in *Proceedings of the 15th International Conference on Information Processing in Sensor Networks*. IEEE Press, 2016, p. 4.

[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[52] Y. Bengio, P. Simard, P. Frasconi *et al.*, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[53] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[54] "Google cloud platform." https://cloud.google.com/.