WiFi Fingerprinting based Indoor Localization: When CSI Tensor meets Deep Residual Sharing Learning

by

Xiangyu Wang

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
Dec 16, 2017

Keywords: Phase measurement, Antenna measurements, OFDM, Neural networks

Approved by

Shiwen Mao, Samuel Ginn Endowed Professor of Electrical and Computer Engineering
Thaddeus Roppel, Associate Professor of Electrical and Computer Engineering
Xiaowen Gong, Assistant Professor of Electrical and Computer Engineering

Abstract

Location-based services (LBS) nowadays have several consumer applications such as indoor localization. Wi-Fi based indoor localization has attracted interest due to the ubiquitous access in indoor environments. In this paper, we propose ResLoc, a deep residual sharing learning based system for indoor localization with channel state information (CSI) tensor data. First, we introduce CSI data in wireless systems and discuss how to build CSI tensor data for indoor localization. Then, we design the ResLoc system, which employs two channels CSI tensor data to train the deep network by using the proposed deep residual sharing learning in the offline phase. For online test phase, we use newly received CSI tensor data to estimate the location of the mobile device based on the enhanced probabilistic method. Finally, the experimental results show the proposed ResLoc system can obtain the decimeter level localization accuracy.

Acknowledgments

I would first like to thank my advisor Prof. Shiwen Mao for his patient guidance and enthusiastic help, without which this thesis would not have been possible. I would also like to thank my committee members, professor Xiaowen Gong and professor Thaddeus Roppel, for serving as my committee members even at hardship. I also want to thank you for letting my defense be an enjoyable moment, and for your brilliant comments and suggestions, thanks to you.

I would especially like to thank my team member, Xuyu Wang, as well as other friends, who gave lots of assistance throughout my thesis work. Xuyu, who took me into this field, taught me a lot about deep learning. Thank Chao Yang, Runze Huang and Zhitao Yu for their cooperative support and continual assistance through this research.

I would also like to thank my colleagues in the RFID Lab, Justin Patton, Dr.Jian Zhang and Yibo Lyu.They were always open to discussions about my ongoing research. I learned a lot from them about how to undertake a research.

Finally, I must express my very profound gratitude to my parents and to my girlfriend for providing me with unfailing support and continuous encouragement throughout my years of study and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Problem

With the remarkable development in mobile devices and wireless techniques [1–18], location-based services for internet of things, like activity recognition and health sensing, has been enhanced significantly [19–21]. To fulfill the requirement of these applications, a high precision location information is indispensable. Considering the fact that the wireless propagation is much more complex in the indoor environment, indoor localization with wireless signals faces lots of unsolved issues, which draw so much attention from researchers. Lately, fingerprinting-based indoor localization has become a research hot-spot, which builds a database with a large amount of Wi-Fi measurements in the offline phase, and then computes the position of a mobile device by comparing the newly received Wi-Fi data with that in the database.

Due to the low requirement on hardware and the easiness to obtaining (e.g. from laptops smartphones or smartwatches), the received signal strength (RSS) is used as the fingerprint in many Wi-Fi based fingerprinting systems. Radar, the first RSS-based fingerprinting scheme, combines a deterministic method and RSS-based fingerprinting for location estimation [22]. Horus is another RSS-based fingerprinting scheme, which utilizes a probabilistic method with K-nearest-neighbor(KNN) [23]. Moreover, various machine learning methods are applied for promoting localization performance, such as neural networks, support vector machine and compressive sensing [24]. Two principle disadvantages inhibit their performance [19]. First, affected by the complex propagation environment, a stable RSS value could not be received for a given location. Second, RSS value is the coarse channel information.

Recently, several 802.11n measurement and experimentation tools are released, such as Intel Wi-Fi Link 5300 NIC [25] and the Atheros AR9580 chipset [26], which can extract channel state information (CSI) from received packets by the modified firmware. Comparing with RSS, CSI is the fine-grained channel information, including subcarrier-level channel measurements in orthogonal frequency division multiplexing (OFDM) systems. Moreover, CSI is much more stable than RSS for a given location [27]. Based on the CSI information, several fingerprinting systems exhibit better localization performance. FIFS exploits the weighted average of CSI amplitudes over three antennas to achieve fine-grained localization [28]. In addition, CSI amplitudes and calibrated phases information are leveraged by DeepFi [29] and PhaseFi [30], respectively. These two schemes collect CSI from all the subcarriers at all the three antennas and generate fingerprinting with deep autoencoder networks. Moreover, to improve the localization accuracy, BiLoc system is proposed based on average CSI amplitude and phase difference information for indoor localization by using bimodal deep autoencoder network [31]. Although these three localization systems based on deep network can obtain better localization performance, they need to build a database to store training feature as fingerprints for every training location, which increases the training time and storage space.

1.2 Approach

In this paper, we consider bimodal CSI tensor data including estimated angles of arrival (AOA) and amplitude information that are obtained from the 5GHz band. Firstly, AOA and amplitude information are stable, which can be effective features for fingerprinting based indoor localization. Moreover, AOA and amplitude information are complementary to each other under different indoor environments. For example, when light of sight (LOS) component for wireless signal is weaker than other AOAs, the amplitude information is useful for improving the localization performance. On the other hand, once the signal is blocked by objects such as wall, the estimated AOA values will help to strengthen the localization accuracy because the amplitude information is greatly weakened. Moreover, we present a new deep residual sharing learning for improving the training capacity with two channels CSI tensor data. The proposed method is different from the original deep residual unit without sharing the residual function. Moreover,

we can stack many residual blocks for adding the depth of the deep network, thus achieving higher learning and representation ability. The residual learning method has been successfully applied for image recognition [32–34]. The proposed method only requires for training one group of weights in deep residual network for all training locations as a classification problem in statistical learning, thus significantly reducing the amount of the stored data.

In particular, we present ResLoc, a deep **Res**idual sharing learning for indoor **Loc**alization with CSI Tensor. In ResLoc, we first construct a CSI tensor including three images, each of which has the same size with 30×30 based on the estimated AoA values and the CSI amplitude values. For CSI tensor, we consider two images from estimated AoA values between antenna 1 and 2, and antenna 2 and 3. Another image is from the amplitude values from one antenna. Thus, by using 990 packets, we can obtain 33 CSI tensor data for one training location. Moreover, we consider two channels CSI tensor data, where the difference between two CSI tensor data is that they have different amplitude information from different antennas for creating the third image in CSI tensor. In ResLoc system, we consider the amplitude information from antenna 1 and antenna 2 for two channel CSI tensor data. For offline training phase, all the constructed two channels CSI tensor data from all training locations are leveraged to train the weights of the deep network based on the proposed deep residual sharing learning, which includes the input block, the residual block and the output block. The new idea for the residual block is that the proposed scheme shares the residual functions for two channels, which can effectively exploit the CSI tensor data. Moreover, we also analyze the proposed deep residual sharing learning for forwarding and backward propagation. For the online stage, we use newly received CSI tensor data to compute the location of the mobile device based on the probabilistic method.

The main contributions of this paper are summarized below.

- This is the first work to use CSI tensor data for indoor localization, which can exploit the rich frequency and time features of the CSI data including the amplitude and phase difference information.

- We propose deep residual sharing learning for training two channels CSI tensor data. Moreover, we can stack many residual sharing blocks for adding the depth of the deep network, thus achieving higher learning and representation ability for CSI tensor data. Moreover, the proposed scheme is analyzed for forwarding and backward propagation. In the online test, we consider a probability method for location prediction.

- We implement the proposed ResLoc system with commodity 5 GHz Wi-Fi in two representative indoor environments with extensive experiments. The results show that ResLoc achieves decimeter level location accuracy, which is better than other deep learning methods.

1.3 Layout

In the remainder of the paper, we introduce the preliminaries and CSI tensor in Chapter 3. We design the ResLoc design in Chapter 4 and performance evaluation in Chapter 5. Chapter 6 summaries this paper.

Chapter 2

Background

RF sensing technique attracts researchers' attention because of its contact-free, easy deployment, and low-cost. In recent years, enormous literature are published to address advanced techniques about RF sensing [35]. Formerly, most of RF sensing techniques are related to signal processing. For example, MUltiple SIgnal Classification(MUSIC) algorithm is utilized in SpotFi [36] and it can achieve a good performance for indoor localization. However, there are still some disadvantages about RF sensing technology based on signal processing. Time consumption is always a key problem for real time system. Signal processing algorithms, like MUSIC, take several seconds, even minutes, to estimate the frequency content of a signal or autocorrelation matrix with an eigenspace method. This drawback limits the implementation of RF sensing in our daily life and pushes researchers to find faster algorithms. Because of AlphaGO's [37] dominant victory, deep learning has become one of hottest techniques in 2017. Moreover, due to the increase of computation capability in mobile devices and computers, more and more RF sensing applications try to solve practical problems with deep learning methods. Even though the training phase of deep learning methods always takes hours, even days, to reach the convergence of neural network weights, deep learning has been hot in several research fields because of its versatile model that could adapt to new problems easily and its outstanding performance that significantly outperforms other solutions in multiple domains. In this chapter, recent works will be discussed in two categories: The application of RF sensing and the deep learning algorithms that are leveraged in RF sensing.

2.1 RF Sensing

The applications of RF sensing are deployed in many areas, like indoor localization, healthcare, activity recognition and virtual reality entertainment. Recently, many distinguished project are constantly emerging in these areas.

2.1.1 Indoor Localization

Localization-based services attract a lot of attention because of their fast growing demands. DeepFi [29] presents a deep learning based indoor fingerprinting scheme using channel state information (CSI). Normally, the algorithm of fingerprint localization consists of two parts, including offline establishment of location fingerprint database and online positioning. To build fingerprint database, WiFi CSI values are collected as fingerprints at each position. At the stage of online positioning, fingerprints that are collected for the test the position are compared with fingerprints in the database. The test position whose fingerprint attains the best matching in the database is chosen as the final estimated position. The difference between traditional fingerprinting methods and DeepFi is that a large number of weights trained by a stack of Restricted Bolzmann Machine (RBM) are utilized as fingerprints to denote different positions. It is a much more effective way to describe the characteristics of CSI values for every location. In other word, deep learning techniques extract more useful and abstract information from CSI, so that DeepFi obtains a better performance compared with other fingerprint based indoor localization systems.

2.1.2 Health Sensing

Health is central to human happiness and well-being. In 2015, U.S health care costs were $3.2 trillion,which makes health care become one of the country's largest industries, equaling to 17.8 percent of gross domestic product. The demand for health monitoring is increasing instantly. Thus, the health sensing application becomes one of the most sought-after application [38–40]. Many researchers also focus on this area. PhaseBeat [41] system is a representation project in this area. It can monitor breathing and heart beats with commodity WiFi by utilizing the

CSI phase difference data. Because of the popularity of commodity WiFi, PhaseBeat is able to provide a contact-free, easy deployment, low-cost, and long-term vital sign monitoring for patience. In PhaseBeat, several techniques are implemented to extract breathing and heart beats. Discrete Wavelet Transform (DWT) is leveraged to remove high frequency noise from the phase difference data; Peak detection and Root-MUSIC are used to estimate breathing rate for single person and multiple person respectively. The PhaseBeat achieves an astonished performance, where 70% test data have an estimated error under 0.5 bpm. In future, more and more vital signals will be monitored by RF sensing in this convenient way.

2.1.3 Activity Recognition

Activity recognition is another important content of RF sensing. It could be exploited for great societal benefits, especially in security applications, such as fall detection and security surveillance. Recently, a deep-learning architecture that uses only RFID data for detection of process phases and activities during trauma is proposed by researchers from Drexel University and Childrens National Medical Center [42]. This project leverages RFID signals as sensors to capture human activities. 3D matrices are generated by processed RFID signals as inputs of the network that is composed with 3 convolutional layers and 3 fully (dense) connected layers. A convolutional neural network is built to classify the activities into different categories. This system achieves the average accuracy of 80.40% for recognition of 11 activities and the average accuracy of 72.03% for detection of the five phases.

2.1.4 Object Tracking

Recently people always dream to control their computer with their hands. RF sensing is making this dream become true. MIT researchers develop a system, as known as RF-IDraw [43], that could accurately trace the trajectory of an RF source. Users are allowed to control their mobile device by writing in the air any word or command. RF-IDraw is a system based on the Angle of Arrival(AoA), which is computed by the phase differences from multiple antennas of the reader. However, there is an ambiguity in computed AOA values. If the distance between two antennas is larger than $\lambda/2$, multiple beams are exhibited even though there is only one true

direction. However, a high resolution beam could not be obtained when the distance between two antennas is smaller than $\lambda/2$. To solve this ambiguity, two antenna pairs are implemented. A widely spaced antenna pair is used to provide high resolution, while tightly spaced antenna pairs are utilized to reduce the ambiguity. The median point-by-point error distance for RF-IDraw is 3.7 cm and 4.9 cm in line-of-sight and non line-of-sight, respectively. This multi-resolution approach makes RF-IDraw exceed the capability of antenna array with the same number of antennas.

2.2 Deep Learning

Recently, deep learning techniques have been undergoing a rapid expansion. Many corporate titans such as Google, Amazon, are devoting into deep learning related researches with their powerful computing resources. On May 25 2017, AlphaGo [37], an artificial intelligence program developed by Google's DeepMind lab, defeated the world best human go player, Ke Jie. The victory is a huge moment in the history of artificial intelligence. What makes AlphaGo so powerful is deep reinforcement learning based system, which makes AlphaGo a lot more human-like and artificially intelligent than something like IBMs Deep Blue, which beats chess grandmaster Garry Kasparov by using brute force computing power to search for the best moving. Moreover, Deep learning also enhances computers better than human at recognizing and sorting images. The deep learning algorithms from Microsoft and Google have reached the error rates of 4.94% and 4.8%, respectively, for sorting images into predefined categories. However, the error rate for the typical human is about 5%. Computers know images, which is much better than human. These two examples show the capability that deep learning algorithm could find or recognize the rules and patterns that human does not know. This great potential draws much attention from researchers and industries. Deep learning plays important roles in more and more areas. Specially, increasing number of wireless sensing applications leverage deep learning techniques to achieve better performance.

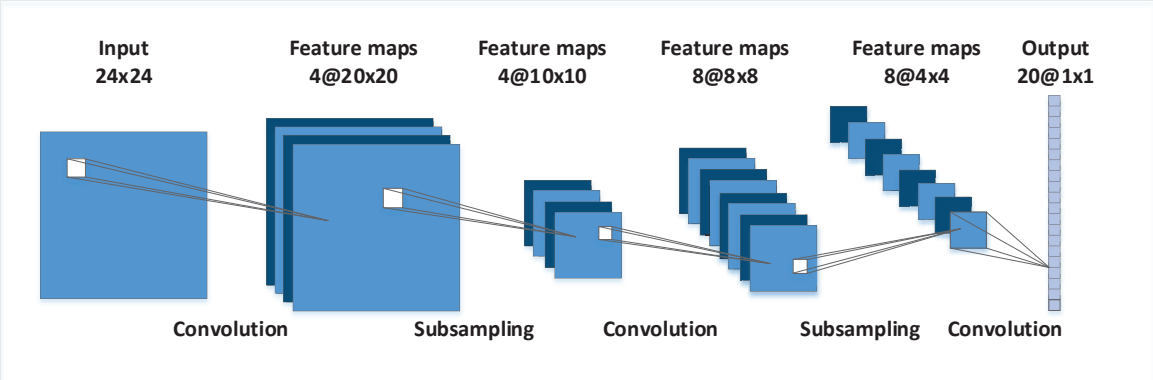

Figure 2.1: Architecture of convolutioanl neural network .

2.2.1 Convolutional Neural Network(CNN)

Convolutional neural network is a well-known deep learning architecture, which is inspired by emulating the natural visual perception mechanism of the living creatures. Thus, convolutional neural network shows great potential in computer vision. In 1998, LeCun proposed LeNet-5 [44], which is one of the first initial architecture of CNN. According to LeNet-5,shown in Figure 2.1, convolution operation and subsampling operation are first applied to the input data with computation units called convolutional layers and subsampling layers, respectively, by the network. After two groups of such computation, the output of the higher layers is processed by a fully connected traditional neural networks, where the final classification results are improved. After that, incresing number of models based on convolutional neural network are proposed, such as AlexNet [45] and ResNet [33]. AlexNet achieved top-1 and top-5 error rates of 37.5% and 17.0% in the ImageNet LSVRC-2010 contest. Compared with LeNet-5, AlexNet is a bigger and more complex model. Max pooling and ReLU nonlinear activation function are leveraged in the model. And the dropout regularization is utilized to solve overfitting problem. From AlexNet, people realize that increasing the depth of the network increases the accuracy of the network. In 2015, a residual learning framework, also called ResNet, is proposed by Microsoft Research. A 152-layer residual network achieves the error rate of 3.57% on the ImageNet test set, and the performance is 1st place on the ILSVRC 2015 classification task. To solve the vanishing gradient problem resulted by the depth of the network, residual module creates a direct path between the input and output to the module implying an identity mapping.

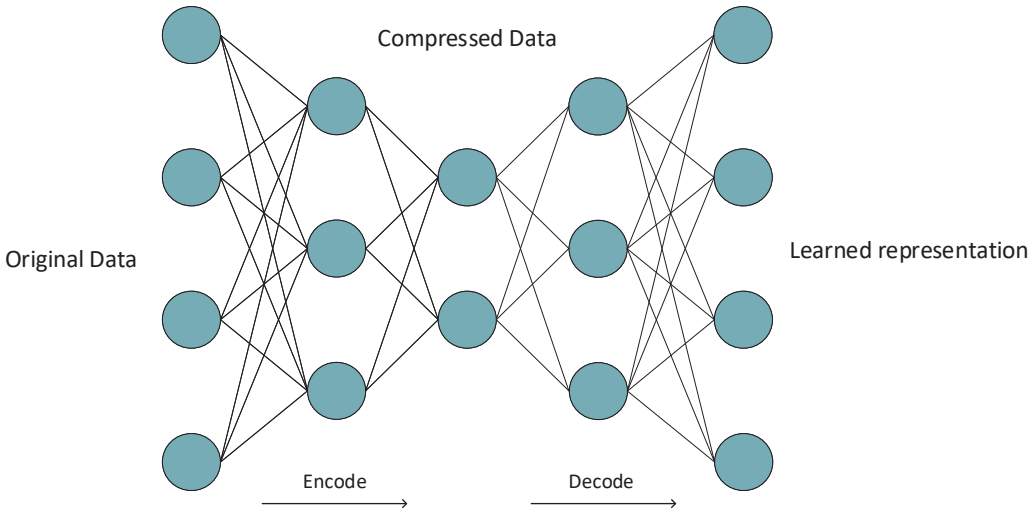

Figure 2.2: Architecture of autoencoder.

The great performance of convolutional neural networks also attracts RF sensing researchers' attention. For example, CiFi [46], an indoor localization system with commodity 5GHz WiFi, uses generated images with estimated AOA values for each received packet from every channel as the input of the network to train a deep convolutional neural network. This system has been demonstrated that the superior performance of the localization has outperformed existing schemes, like FIFS [47] and Horus [48]. Moreover, the proposed ResLoc system in this thesis is also motivated by the above deep residual learning [33].

2.2.2 Autoencoder

Unlike convolutional neural networks, an autoencoder neural network [49] is an unsupervised learning algorithm. Its aim is to generate the output that is an approximation of the input. The architecture of autoencoder is as shown in Figure 2.2. Generally, an autoencoder is composed with three parts, an input layer, one or more hidden layer and an output layer. To reconstruct its own input, the output layer has identical number of nodes as the input layer. On the other hand, the number of nodes in the hidden layers is always smaller than the number of nodes in the input layer, so that a compressed representation is extracted from the input data. There are three stages in the training process, including pretraining, unrolling, and fine-tuning. In the

pretraining stage, each neighboring set of two layer is considered as an restricted boltzmann machines(RBM) for pretraining to approximate a good solution. Then the deep network is enrolled to obtain the reconstruction data using the input data with forward propagation. After that, a backpropagation technique is used to fine-tune the results. Similar to principal components analysis(PCA), the purpose of the autoencoder is to find low-dimensional representation of the input data. Thus, autoencoder neural network is widely used in data compression and signal denoising. The first application that utilizing autoencoder in RF sensing is DeepFi [50], a deep learning based indoor fingerprinting scheme using CSI. The channel state information values are collected at every training location as inputs of the deep autoencoder. For every training location, the deep autoencoder is trained to obtain a set of weights and biases, which are leveraged as fingerprints for corresponding locations. To estimate location, the fingerprints that are gathered during the online phase are compared with fingerprints in the database. The position whose fingerprint attains the best matching is chosen as the final estimated position. The experience shows that the mean distance error in the living room and the laboratory is about 1.2 and 2.3 meters, respectively.

2.2.3 Long Short-term Memory(LSTM)

To process variable-length sequence inputs, recurrent neural networks (RNN) are proposed, which are originated from conventional feedforward neural networks. With the help of the feedback loop in the recurrent layer, long-term dependencies could be handled. However, the dependencies also result into that the recurrent neural network could not be trained successfully, because the gradient of the loss function tends to either diminish or explode, which makes gradient-based optimization methods become ineffective. In order to solve this problem, long short-term memory architecture [51] is proposed. Unlike the traditional RNN that inputs at each time-step affect every feedback loop, a LSTM unit, in Figure 2.3, contains three gates to control the dataflow in it. An input gate decides if a new value could flow into the memory. A forget gate controls if a value could remain in memory. And an output gate determines if the value in memory could be used to compute the output of the unit. All these gates make sure that gradient-based optimization methods could be used in LSTM units. The LSTM network has

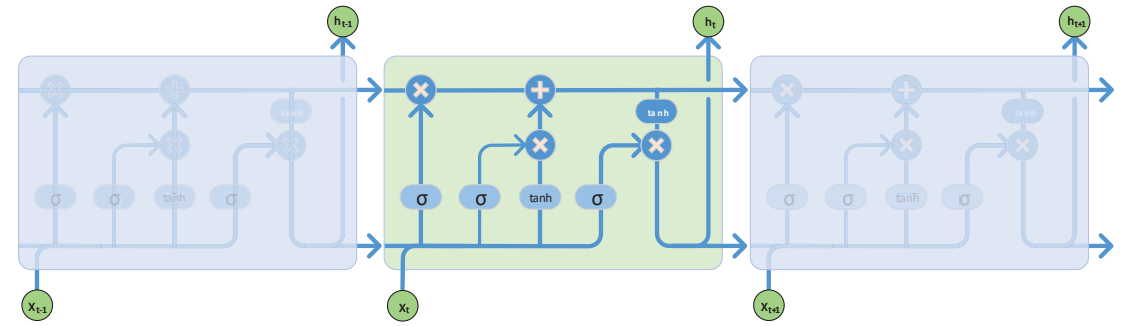

Figure 2.3: Architecture of LSTM.

been used widely for machine translation, speech recognition and time-series prediction. More and more applications based on LSTM have shown in the area of RF sensing. A system, called DeepConvLSTM [52], is presented to recognize human activities from wearable sensor data. Considering the input of the network is a short time series extracted from the sensor, the fully connected layer of the convolutional neural network is replaced by LSTM units in this system. Compared with other non-recurrent approaches, DeepConvLSTM improves the best reported result by 6% on the Skoda dataset. This result shows a great potential of LSTM in the area of RF sensing.

Chapter 3

Preliminaries and CSI Tensor

3.1 Channel State Information Preliminaries

OFDM technique is implemented to handle frequency selective channels and acquire high data rate in many wireless network standards (such as LTE, WiMAX, WiFi, and 5G) [14]. In OFDM system, the spectrum is composed by multiple orthogonal subcarriers. To alleviate channel fading and large delay spreads, wireless data could be transmitted with different modulation and coding schemes. Also, to reduce the complexity of FFT processing, cyclic redundancy is utilized at the receiver. Recently, several 802.11n measurement and experimentation tools are released, which are convenient to extract CSI data from several off-the-shelf NICs, such as Intel WiFi Link 5300 NIC [25] and the Atheros AR9580 chipsets [26]. ResLoc exploits the Intel 5300 NIC to collect CSI data, which contains 30 out of the 56 subcarriers at the WiFi receiver for a 20Mhz or 40Mhz channel.

For a Wi-Fi network with OFDM, we denote H_i as the CSI value of subcarrier i, which is a complex value expressed by

$$H_i = |H_i| \exp(j \angle H_i), \tag{3.1}$$

where $|H_i|$ and $\angle H_i$ are the amplitude response and phase response of subcarrier i, respectively. Besides amplitude information, ResLoc system also uses phase difference information between two adjacent antennas to build CSI tensor as the input data for indoor localization, where phase difference information is stable [31, 53, 54].

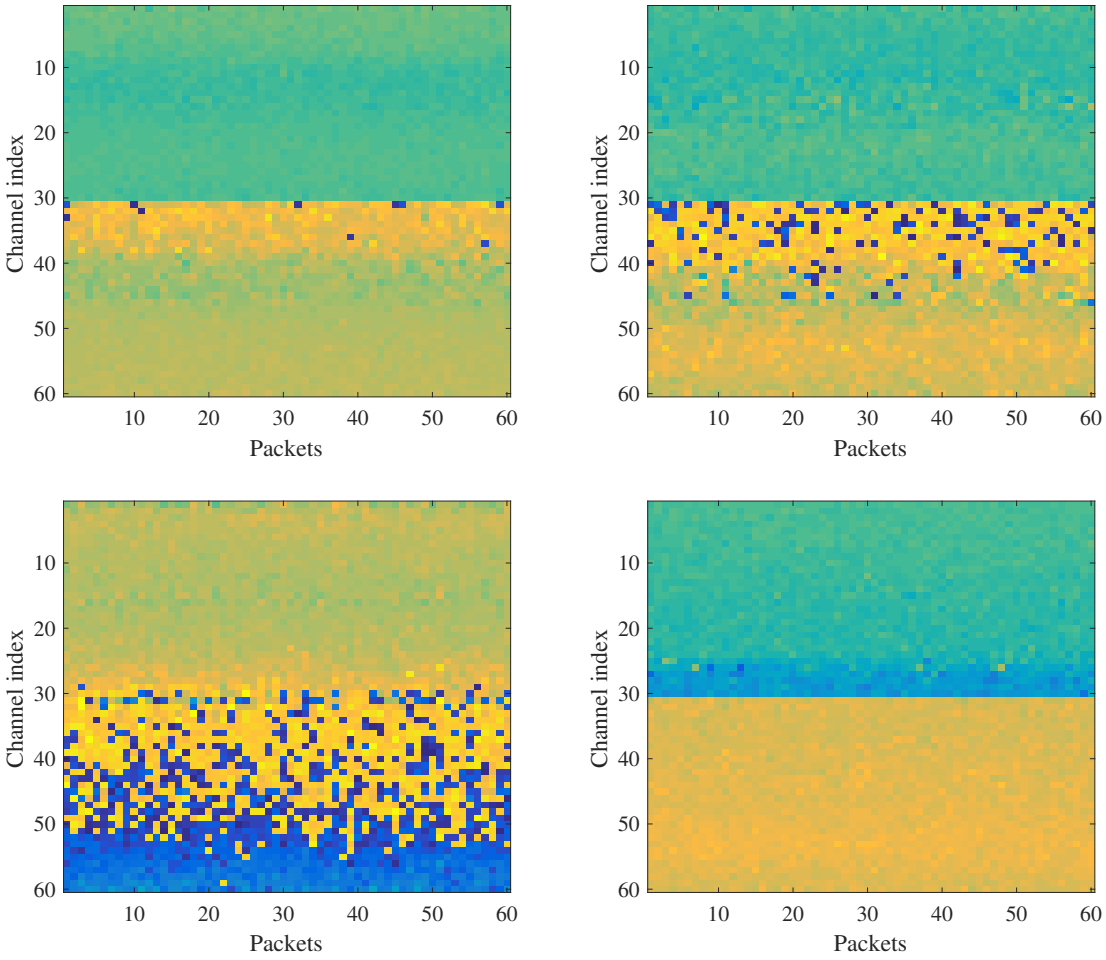

Figure 3.1: Images generated by estimated AoA values.

3.2 CSI Tensor Construction

To build CSI tensor as the input of the ResLoc system, we compute bimodal CSI data including estimated AOAs and amplitude information that are obtained from the 5GHz band. Besides the amplitude information from three antennas, it is also easy to estimate the corresponding AoA values between two adjacent antennas from each subcarrier and each received packet by using the same method in BiLoc system [31]. Then, for every 30 packets(x-axis) with 30 subcarriers (y-axis), we can construct a CSI tensor including three images, each of which has the same size with 30×30 based on the estimated AoA values and the CSI amplitude values. For CSI tensor, we consider two images from estimated AoA values between antenna 1 and 2, and antenna 2 and 3,shown in Figure 3.1; Another image is formed from extracted amplitude values from one antenna. Thus, by using 1500 packets, we can obtain 50 CSI tensor data

for one training location or one test location. For ResLoc system, we consider two channels CSI tensor data, where the difference between two CSI tensor data is that they have different amplitude information from different antennas for creating the third image in CSI tensor. In ResLoc system, we consider the amplitude information from antenna 1 and antenna 2 for two channel CSI tensor data.

There are three reasons for using the CSI tensor as the input of ResLoc. First, by using CSI tensor with three dimensional data it can strengthen the performance of deep network for classification problem with indoor localization. Moreover, all subcarrier information from all packet sample are exploited by three images in CSI tensor, which contains rich frequency and time features of the CSI data. We can thus extract more effective features from CSI tensor. Third, we leverage bimodal CSI data including the estimated AoA values and the CSI amplitude values for indoor localization, which are complementary to each other under different indoor environments [55].

Chapter 4

The ResLoc System

4.1 ResLoc System Architecture

In Figure 4.1, the ResLoc system is composed by one transmitter, which is a mobile device, and one receiver, which is an access point. Both devices are equipped with the Intel 5300 NIC. To collect the CSI data, the transmitter is set to the injection mode, and the receiver works in the monitor mode. The Intel 5300 NIC reports CSI from 30 groups of subcarriers from each antenna. After CSI data collection, we can build two channels CSI tensor based on estimated AOAs and amplitude information. ResLoc system employs the fingerprinting based method, which includes the offline training and online location prediction. For training data with two channels CSI tensor in the offline phase, we propose a deep residual sharing learning for obtaining the optimal weights of deep residual network. For online location prediction, we utilize newly received CSI tensor data to compute the location of the mobile device based on an enhanced probabilistic approach. Our ResLoc system is totally different from traditional fingerprinting based methods, which build the database for every training location based on raw data or training features as the fingerprints. In fact, ResLoc system only requires for training one group of weights in deep residual network for all training locations like the regression or classification problem in statistical learning. Apparently, this method reduces the amount of the stored data significantly. On the other hand, it also contributes to the improvement of the robustness for indoor localization based on the proposed deep residual sharing learning, which can effectively represent the features of CSI data.

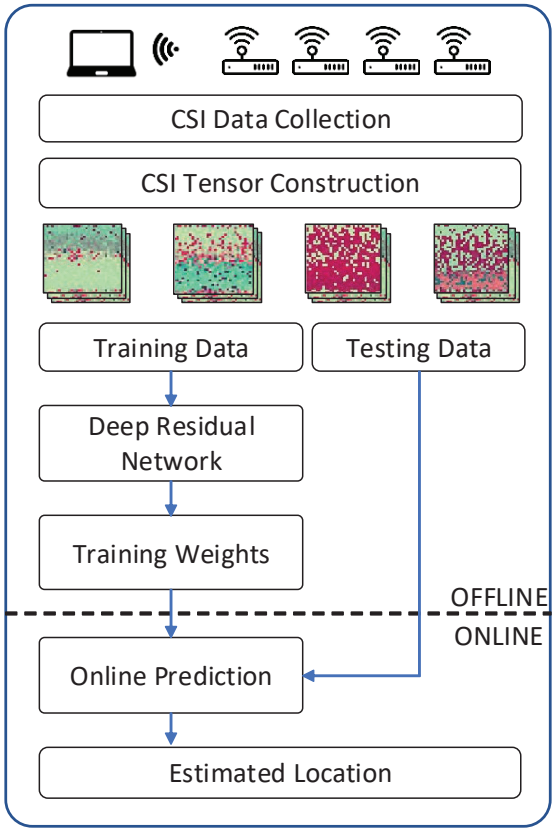

Figure 4.1: The ResLoc system architecture.

4.2 Offline Training

We propose a deep residual shared learning for training the deep network with bi-modal CSI tensor, which includes the input block, the residual block, and the output block in Fig. 4.2.

4.2.1 Input Block

For the input block, the bi-modal CSI tensor data can be trained by four different layers with the Convolution2D layer, batch normalization layer, activation layer and max pooling layer, respectively. It can obtain the local dependency and scale invariant feature from bi-modal CSI tensor. Furthermore, Input block can exploit more abstract representation of the input CSI tensor data from the lower layers to the higher layers, which can improve the feature extraction of CSI tensor data for indoor localization. We discuss four different layers for the input block.

The Convolution2D layer is to obtain feature maps within local regions in input CSI tensor or the previous layer's feature maps with several convolution kernels. In fact, each data of a

feature map is connected with the local data in the previous layer. Moreover, by using different convolution kernels we can obtain all produced feature maps. Let θ_i^l denote as the ith feature map in layer l, which is defined as

$$\theta_i^l = \sum_{m \in S_{l-1}} w_{im}^l * \theta_m^{l-1} + b_i^l, \tag{4.1}$$

where w_{im}^l is the convolutional kernel to generate the ith feature map in layer l, b_i^l is the bias of the ith feature map in layer l, S_{l-1} is the set of feature maps in layer $(l-1)$ connected to the current feature map, which is the same for different m due to local weights sharing. The convolution operation with weights sharing scheme can improve the efficiency for training deep network.

The batch normalization layer can adjust the input distribution for different layers and thus alleviate the problem of Internal Covariance Shift that is as the data flow propagates for different layers in deep network, the distribution of input will be shifted, thus reducing the learning capacity [56]. In batch normalization layer, the input data are normalized such that it can satisfy a zero mean and a unit standard deviation, where the estimation of mean and variance are obtained by each mini-batch. Then, to improve the representation ability in deep network, the normalized data is shifted and scaled. Thus, the batch normalization for the k_{th} input data x_k is formulated by

$$y_k = \gamma \frac{x_k - u_B}{\sqrt{\sigma_B^2 + \epsilon}} + \beta \tag{4.2}$$

where u_B and σ_B^2 are the the mean and variance of mini-batch, respectively, ϵ is the small constant value to avoid numerical problems in batch normalization, γ and β are the scaled and shifted parameters, which are learned from training. By using batch normalization, it can instead of Dropout for avoiding overfitting in training.

The activation layer can be employed to avoid obtaining trivial linear combinations of input data, which can detect nonlinear features. Traditional nonlinear activation functions mainly exploit sigmoid $\sigma(x) = \frac{1}{1+\exp(-x)}$ and tanh $\tanh(x) = 2\sigma(x) - 1$ functions in neural networks. In ResLoc system, we leverage rectified linear unit (ReLU) as the activation function with the

expression $ReLu(x) = max(x, 0)$, which can stay the positive part and suppress the total negative part to zero [57]. For training in deep convolution neural networks, ReLU function has faster training than that for traditional sigmoid and tanh functions. Moreover, it can also exploit the sparse representations in the hidden units and can have effectively training without pre-training.

The max pooling layer can reduce the resolution of the feature maps by downsampling over a local neighborhood in the feature maps of the previous layer. It is invariant to distortions and small shifts on the inputs. Moreover, it also improves the robustness of the deep network. The feature maps in the previous layer are pooled over a local temporal neighborhood by the max pooling function, as

$$\theta_{ij}^{l+1} = \max_{k \in G_j^l} \theta_{ik}^l, \tag{4.3}$$

where G_j^l is the set of pooling region for the jth value in the ith feature map in layer l, θ_{ik}^l is the kth value of the ith feature map in layer l. Other methods such as the mean or sum pooling function can be also used in this stage for reducing the training time.

4.2.2 Residual Block

For the residual block, we propose a new deep residual sharing learning for improving the training capacity with two channels CSI tensor data. The proposed method is different from the original deep residual unit without sharing the residual function. Moreover, we can stack many residual blocks for adding the depth of the deep network, thus achieving higher learning and representation ability. For residual learning [33, 34], the idea is that instead of learning the underlying mapping $H(x)$ by using a few stacked layers, we can learn the residual function $F(x) = H(x) - x$. Thus, the original mapping can become $F(x) + x$, where x is implemented by identity mapping with the shortcut connection. Thus, it is easy for training very deep network by using residual learning. Moreover, we implement the proposed deep residual sharing learning by sharing the residual functions for two channels input data in Figure 4.2. On the

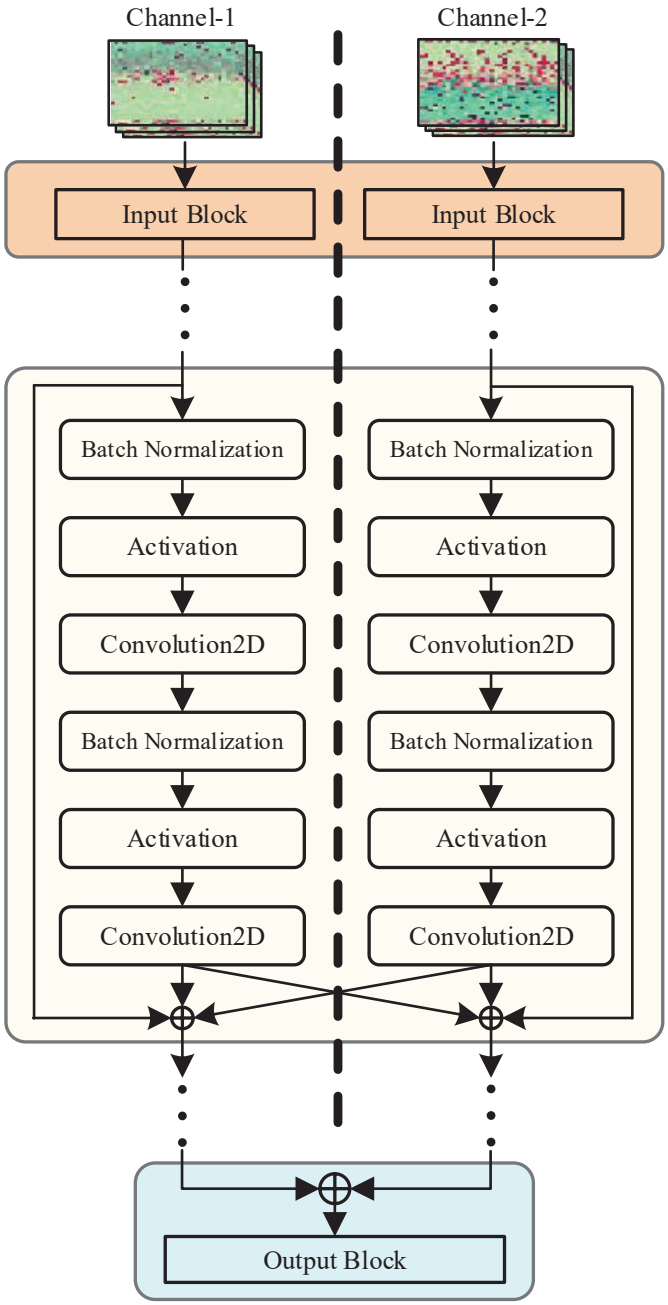

Figure 4.2: Deep residual sharing learning for offline training

other hand, the residual function includes two layers convolution operations, each of which includes the batch normalization layer, the activation layer, and the convolution2D layers. They are implemented as the same as the input block.

For analyzing deep residual sharing learning for forwarding and backward propagation, we denote the x_k^1 and x_k^2 as the input data with channel 1 and channel 2 for the k_{th} residual block, respectively. Let R denote the residual function with two 3×3 convolution layers.

Based on Figure 4.2, we have $x_{k+1}^1 = x_k^1 + R(x_k^1) + R(x_k^2)$ and $x_{k+1}^2 = x_k^2 + R(x_k^2) + R(x_k^1)$ for the k_{th} residual block at channel 1 and channel 2, respectively. Thus, we can recursively obtain the x_K^1 and x_K^2 for the K_{th} residual block at channel 1 and channel 2, which is formulated by

$$\begin{aligned} x_K^1 &= x_k^1 + \textstyle\sum_{i=k}^{K-1} R(x_i^1) + \sum_{i=k}^{K-1} R(x_i^2) \\ x_K^2 &= x_k^2 + \textstyle\sum_{i=k}^{K-1} R(x_i^2) + \sum_{i=k}^{K-1} R(x_i^1) \end{aligned} \tag{4.4}$$

Based on the above equations about forward propagation, we can find that the output x_K^1 and x_K^2 shares the same residual function, which can be represented by the summation of preceding residual functions adding input x_i^1 or x_i^2, respectively. This reduces the error of the gradient propagation. Moreover, it is easier to train the sharing residual functions, which are pushed into zeros when the identity mapping are optimal. On the other hand, we consider the loss function as L for the backward propagation. Based on the chain rule of backpropagation, we can obtain:

$$\begin{aligned} \frac{\partial L}{\partial x_k^1} &= \frac{\partial L}{\partial x_K^1}(1 + \frac{\partial}{\partial x_k^1}(\textstyle\sum_{i=k}^{K-1}(R(x_i^1) + R(x_i^2)))) \\ \frac{\partial L}{\partial x_k^2} &= \frac{\partial L}{\partial x_K^2}(1 + \frac{\partial}{\partial x_k^2}(\textstyle\sum_{i=k}^{K-1}(R(x_i^2) + R(x_i^1)))) \end{aligned} \tag{4.5}$$

By the the above equations about backward propagation, we can see that the gradients $\frac{\partial L}{\partial x_K^1}$ and $\frac{\partial L}{\partial x_K^2}$ are directly propagated back to the any shallower input x_k^1 and x_k^2, respectively. Moreover, because the gradients for the sharing residual functions $\frac{\partial}{\partial x_k^1}(\sum_{i=k}^{K-1}(R(x_i^1) + R(x_i^2)))$ and $\frac{\partial}{\partial x_k^2}(\sum_{i=k}^{K-1}(R(x_i^2) + R(x_i^1)))$ are not always -1, the gradients $\frac{\partial L}{\partial x_K^1}$ and $\frac{\partial L}{\partial x_K^2}$ cannot be canceled for the mini-batch with SGD to avoid the problem of the vanishing of gradient. Thus, the proposed deep residual sharing learning can increase the learning capacity and leverage two channels CSI tensor data.

4.2.3 Output Block

For the output block, we first merge two channel data into single channel. Then, we implement basic data operations for the merged data including batch normalization, activation with ReLU, and max pooling. Moreover, the main operation in the output block is the fully-connected layer, which employs a basic neural network with one hidden layer to train the output data based on softmax classifier. We consider the input data for the softmax function as the R dimensional

vector $z = [z_1, z_2, ..., z_R]$, where R is the number of clusters. Then, the softmax function can map the R dimensional vector to the normalized data $p = [p_1, p_2, ..., p_R]$, that is

$$p_i = \frac{e^{z_i}}{\sum_{r=1}^{R} e^{z_r}} \text{for } i = 1, 2, ...R. \tag{4.6}$$

In addition, we define the loss function as the cross-entropy to measure the difference between the output normalized data and the true label data, that is

$$E = -\sum_{r=1}^{R} y(r) log(p_r), \tag{4.7}$$

where $y(r)$ means the true label data for the r_{th} location. Then, we can train the parameters in deep network with the stochastic gradient descent method by minimizing the values of the loss function.

Weight Training with Deep Learning

The pseudocode for offline training with two input tensors is presented in Algorithm1 and Algorithm2. The inputs to the algorithm1 are two bimodal CSI tensors, For one input tensor, it includes two phase difference slices and an amplitude slice. Each of slices has the same size with 30 times 30 based on the estimated AoA values and the CSI amplitude values for every 30 packets(x-axis) with 30 subcarriers (y-axis). The input datasets are spit into mini batches to train the network. First, batches are processed by the input block, which consists of a convolution layer, a batch normalization layer, an activation layer and a pooling layer. To obtain the output of the input block, batches are dealt by the layers sequentially (lines 10-18). Because of our two-channel framework, two input tensors pass two channels parallel based on tesorflow. The outputs of the input block are processed by residual blocks.Then the outputs of residual blocks are delivered into the output block. Similarly, the output block consists of a batch normalization layer, an activation layer, a convolution layer and a pooling layer as well. What the difference in the output block is two special layers, a merge layer and a fully-connected layer. Before the input tensors pass the merge layer, they are passed

Algorithm 1: Weights Training

Input: Input tensor dataset T_1 and input tensor dataset T_2, number of repetitions for residual blocks K ;
Output: : Trained weight W, b;
Divide input datasets T_1 and T_1 into a batches that contains q CSI tensors;
c denotes as channel index;
while $epoch < 50$ **do**
  for $d = 1 : a$ **do**
    $\theta_1 = M_1^a$;
    $\theta_2 = M_2^a$;
    //M denotes as CSI tensor batch;
    for $c = 1 : 2$ **do**
      $\theta_c = Convolution(\theta_c)$;
      //Calculate outputs of the convolution layer
      $\theta_c = \gamma_c \frac{\theta_c - u_{Bc}}{\sqrt{\sigma_{Bc}^2 + \epsilon_c}} + \beta_c$;
      //Calculate outputs of the batch normalization layer
      $\theta_c = ReLU(\theta_c)$;
      //Calculate outputs of the activation layer
      $\theta_c = pool(\theta_c)$;
      //Calculate outputs of the pooling layer
    end
    do Residual blocks;
    for $c = 1 : 2$ **do**
      $\theta_c = X_c$;
      //X_c is the output of the residual block
      $\theta_c = \gamma_c \frac{\theta_c - u_{Bc}}{\sqrt{\sigma_{Bc}^2 + \epsilon_c}} + \beta_c$;
      //Calculate outputs of the batch normalization layer
      $\theta_c = ReLU(\theta_c)$;
      //Calculate outputs of the activation layer
      $\theta_c = Convolution(\theta_c)$;
      //Calculate outputs of the convolution layer
    end
    $S = \theta_1 + \theta_2$; //Calculate outputs of the merge layer
    $S = \gamma \frac{S - u_{Bs}}{\sqrt{\sigma_{Bs}^2 + \epsilon_s}} + \beta_s$; //Calculate outputs of the batch normalization layer
    $S = ReLU(S)$; //Calculate outputs of the activation layer
    $S = pool(S)$; //Calculate outputs of the pooling layer
    $q = softmax(W * flattened(S) + b)$;
    //Fully-connected Layer
    Loss function $L = -\sum_r y_r \log(q_r)$;
    Update weights and bias using the error with back-propagation;
  end
end

Algorithm 2: Pseudocode for residual blocks

Input: two outputs of input blocks, I_1 and I_2, and number of repetitions for residual blocks K;
Output: two outputs of residual blocks,X_1 and X_2;
c denotes as channel index;
for $k = 1 : K$ **do**
    if $k == 1$ **then**
        $X_1 = I_1$;
        $X_2 = I_2$;
    end
    for $c = 1 : 2$ **do**
        $\theta_c = Convolution(X_c)$;
        //Calculate outputs of the convolution layer
        $\theta_c = \gamma_c \frac{\theta_c - u_{Bc}}{\sqrt{\sigma_{Bc}^2 + \epsilon_c}} + \beta_c$;
        //Calculate outputs of the batch normalization layer
        $\theta_c = ReLU(\theta_c)$;
        //Calculate outputs of the activation layer
        $\theta_c = Convolution(\theta_c)$;
        //Calculate outputs of the convolution layer
    end
    $X_1 = \theta_1 + \theta_2 + X_1$;
    $X_2 = \theta_1 + \theta_2 + X_2$;
    //Calculate outputs of residual blocks
end

into two-channel framework parallel. Namely, two inputs of the output block are dealt by the batch normalization layer, the activation layer and the convolution layer parallel (line22-29). After the two-channel framework merges together at the merge layer, the output of the merge layer is processed by a batch normalization layer, an activation layer, a pooling layer and a fully connected layer sequentially (line 32-45). Once the output of the fully-connected layer is obtained, we could compute cross entropy between the prediction result of the network and the desired labels. Then, the weights and biases are updated using the error with back-propagation algorithm. Finally, we need to update all batches, which is implemented for 50 epochs in the offline training algorithm.

The pseudocode for residual blocks is given in Algorithm 2. The inputs to the algorithm are the number of repetitions for residual blocks $K$and two output tensors of input blocks I_1 and I_2. The outputs to the algorithm includes two tensors, X_1 and X_2.The repetition defines the number of residual blocks that are stacked to form residual blocks. The basic residual block is composed by a two-channel framework, which includes two convolution layers, a batch normalization layer and an activation layer in each channel. The stacking sequence of these layers is as shown in Algorithm 2 (line10-17). It is noteworthy that there is a sharing layer at the end of residual block. For each channel, the input of the current block and the outputs of residual paths from both channels are sum up in the sharing layer.

4.3 Test Phase

For the test phase, a probabilistic method is leveraged to estimate the position of the mobile device by using the newly received CSI tensor data from the test points based on the trained deep network. Let T denote the number of CSI tensor from one position, and p_{ij} denote as the output result of the deep network with the jth CSI tensor for the ith location. The matrix $\boldsymbol{P}$ as the prediction output of the deep network with T CSI tensor data for R training locations, that is

$$\boldsymbol{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1T} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2T} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{R1} & p_{R2} & p_{R3} & \cdots & p_{RT} \end{bmatrix}. \tag{4.8}$$

To reduce the variance of the output results, T output data for every location are averaged. Thus, we can obtain the vector $\bar{P} = [\bar{p}_1, \bar{p}_2, ..., \bar{p}_R]$, where $\bar{p}_i$ is the mean for the output vector $[p_{i1}, p_{i2}, ..., p_{iT}]$ in the i_{th} row.

Finally, we can compute the location of the mobile device as a weighted average of all R locations, that is

$$\hat{L} = \sum_{i=1}^{R} l_i \times \bar{p}_i, \tag{4.9}$$

where l_i is the ith training location.

Chapter 5

Experimental Study

5.1 Experiment Configuration

To evaluate the performance of the ResLoc, we implement it with 5GHz WiFi devices. In order to collect CSI data, a desktop computer and a Dell laptop are used as access point and mobile device. Both computers are equipped with an Intel 5300 network card, running Ubuntu desktop 14.04 LTS system. To transmit the data to the desktop, the Dell laptop with one antenna works in the injection mode. The monitor mode is executed in the desktop to receive data. The distance between two adjacent antennas on the Desktop is set as 2.68cm, which is a half wave length for 5.58GHz WiFi signal. Moreover, the PHY is the IEEE 802.11n OFDM system with QPSK modulation and 1/2 coding rate. To accelerate the training process, we employ the offline stage of the ResLoc in Keras with tensorflow backend on a PC with Intel(R) Core(TM) i7-6700K CPU, and an Nvidia GTX1070 GPU [58]. ResLoc are compared with two typical deep learning localization approaches, BiLoc [31] and DeepFi [29], to evaluate its performance. Moreover, we also consider the localization performance for ResLoc with the single channel. For the sake of fairness, the same CSI training dataset and testing dataset are leveraged in all four approaches. We examine them in two experimental environments including a computer laboratory and a corridor.

5.1.1 Computer Laboratory

Computer Laboratory: We set up the first testbed in a 6×9 m^2 computer laboratory in the Borun Hall in the Auburn University campus. This laboratory is a crammed environment. The

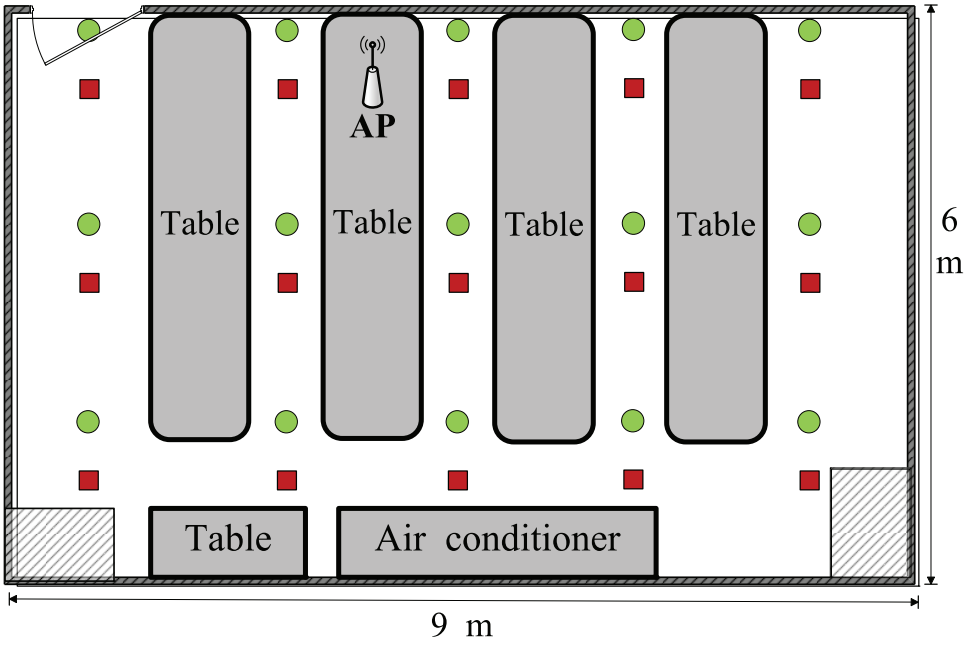

Figure 5.1: Layout of the computer laboratory: training locations are marked as red squares and testing locations are marked as green dots.

furniture and appliances block the most of LOS paths. 15 training locations are shown as red squares in Figure 5.1, while the other 15 green dots are testing locations. The distance between two adjacent training locations is 1.8m. Our receiver is fixed on the table. We collect 1000 CSI packets from every training location and testing location to accumulate CSI data. Moreover, we set the number of layers for the proposed deep network as 34, which has higher localization accuracy and smaller training time.

Corridor:

5.1.2 Corridor in Broun Hall

We set up the second testbed in a long corridor in Borun hall, which is 9×25 m^2. filled with no furniture and appliance. In this scenario, LOS path is majority. We employ 15 training location and 15 testing location in a straight line. The distance between two adjacent training locations is 1m. The red squares are training locations and the rest green dots are testing locations. We set the receiver in the middle of the corridor. 1000 packets are obtained from every training location and testing location to collect 5GHz CSI data. The number of layers in the deep network in the corridor is the same as that in the computer laboratory.

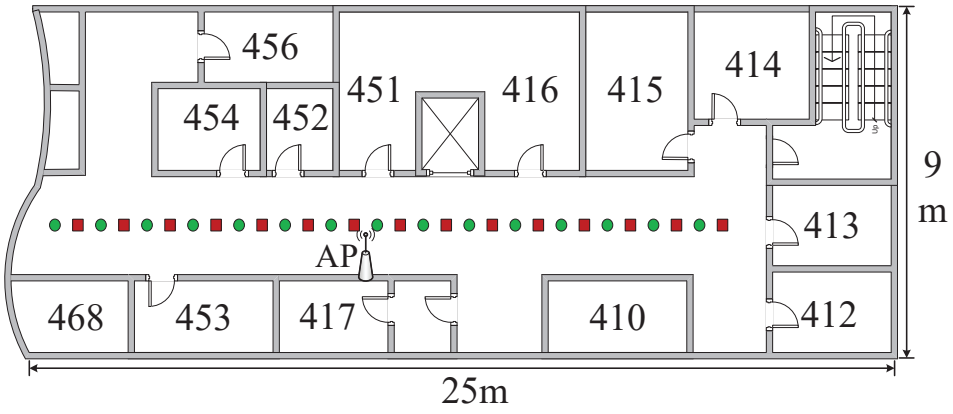

Figure 5.2: Layout of the corridor: training locations are marked as red squares and testing locations are marked as green dots.

5.1.3 Accuracy of Location Estimation

Figure 5.3 depicts the training loss over epoches of the ResLoc for the laboratory and corridor scenarios. To prevent overfitting for the training CSI tensor data set and reduce training time, the epoch is set as 50. As illustrated in Fig.3, the train loss for the corridor curve reaches about 0.3 and the training loss for the lab scenario stops at about 0.5. Moreover, based on Nvidia GTX1070 GPU, we can obtain the smaller training time for the laboratory and corridor scenarios, which are 608.14s and 619.35s, respectively. Also, the test time for the laboratory and corridor scenarios are 0.587s and 0.647s, which can be accepted for indoor localization.

Figure 5.4 shows the cumulative distribution function(CDF) of distance error across the 15 positions in the laboratory. Unlike the corridor scenario that the LOS is majority, the furniture and appliances block most of LOS paths in this environment. As we can see, the maximum distance errors for ResLoc with two channels and single channel are about 2.5m, which is less than DeepFi and BiLoc. In addition, the median of distance errors for ResLoc with two channels and single channel are about 0.89m, which also outperforms BiLoc and DeepFi by 0.51m and 0.89m, respectively. For ResLoc with two channels, the distance error of over 30% testing data is less than 0.3m. However, there is no data falling within this error range for DeepFi and BiLoc. In summary, based on the proposed deep residual sharing learning, ResLoc with two channels exhibits the best performance in this rich multipath scenario.

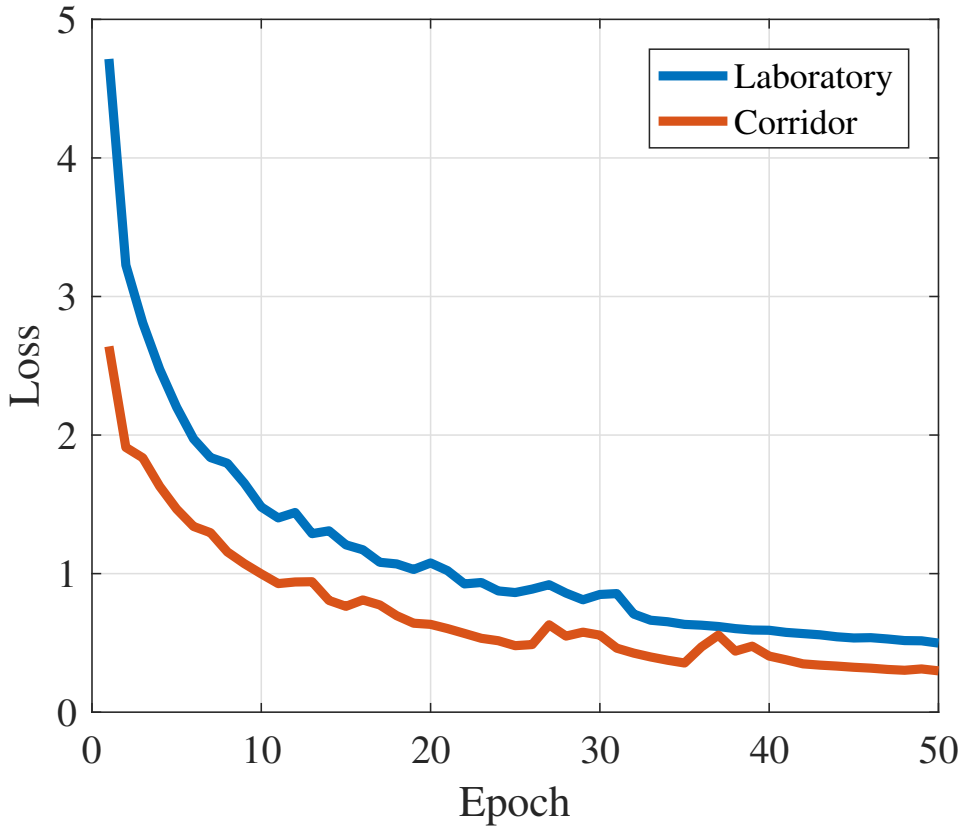

Figure 5.3: Training errors for the laboratory and corridor experiments.

Figure 5.5 plots the CDF of localization error in the corridor scenario. As shown in Figure 5.5, the maximum distance error for ResLoc with two channels and single channel are 3.14m and 3.95m, respectively, which are significantly less than that of other two schemes, DeepFi and BiLoc. It shows that the ResLoc has a better stability than DeepFi and BiLoc. In addition, the median of distance errors for ResLoc with two channels and single channel, BiLoc and DeepFi are about 0.98m, 1.24m, 1.68m, and 1.75m respectively. Thus, ResLoc with two channels achieves the best performance in this scenario. Besides the better performance, the proposed ResLoc system only requires one set of weights for all training locations to achieve localization, which means that it is not necessary for ResLoc to store fingerprints for every training location like BiLoc and DeepFi. Furthermore, ResLoc does not need a ratio for the bi-modal data to obtain a better localization performance.

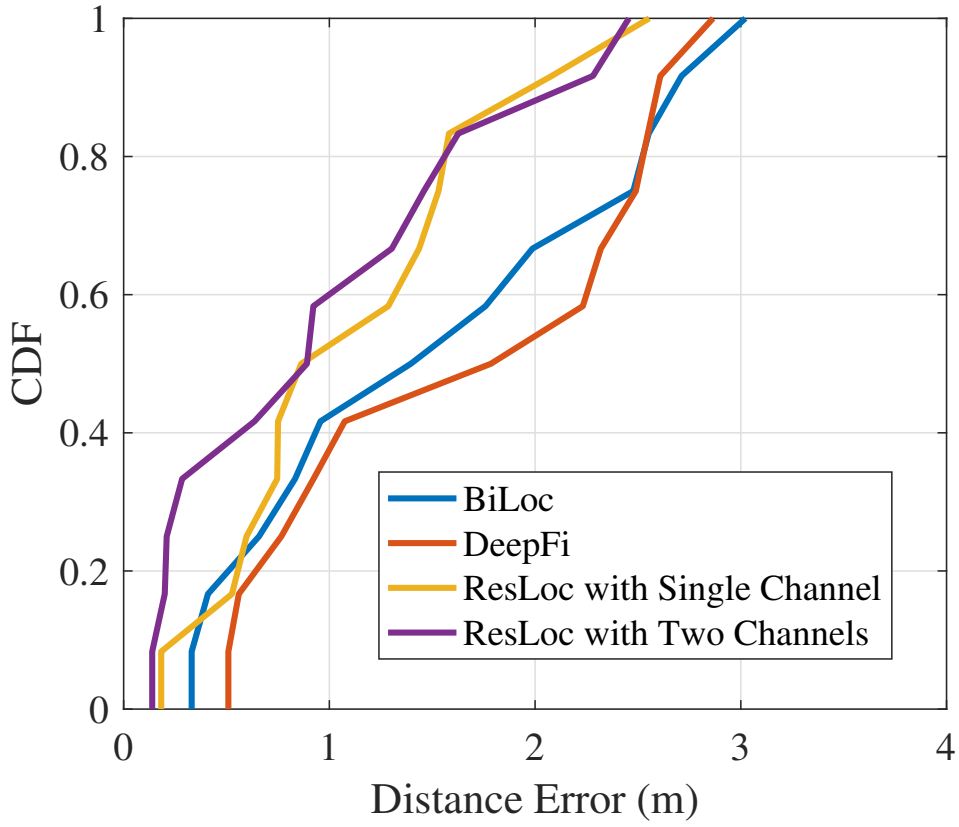

Figure 5.4: CDF of localization errors for the laboratory experiment.

5.1.4 Effect of Different Parameters

Impact of image size

To determine how image size impacts the accuracy of the indoor localization, we test ResLoc with images sized to 30x20, 30x25, 30x30, 30x35 and 30x40. For fairness, 50 images are constructed for every training position. Epoch and batch size are set to 50 and 10, respectively. As is shown in Figure 5.6, distance errors for both scenarios decrease slightly as the image size increases from 30x20 to 30x30, then rise as the image size increases from 30x30 to 30x40. However, the distance errors are stable in these two scenarios. The errors in the lobby and the lab are about 1.2 meters and 1.13 meters, respectively. This result indicates that the localization performance of Resloc is robust enough to the image size. Even though the size of images is changed, Resloc could achieve localization with high precision.

As discussed previously, the image size does not show a significant effect on localization accuracy. To select the best size for training and testing the network, we also compare the

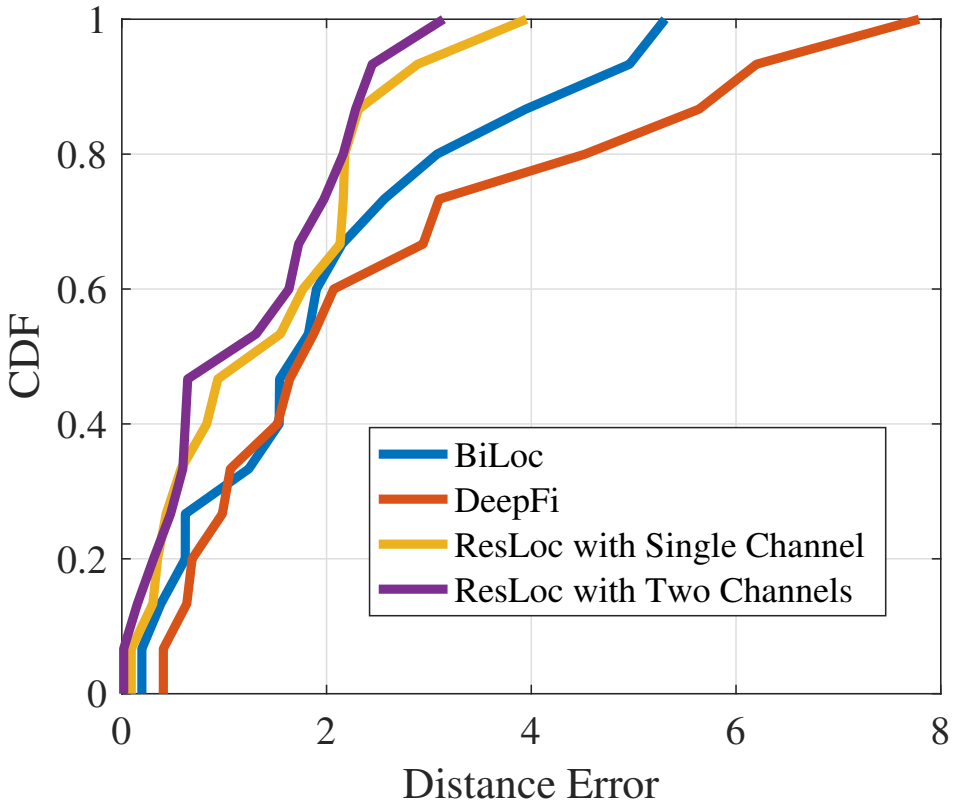

Figure 5.5: CDF of localization errors for the corridor experiment.

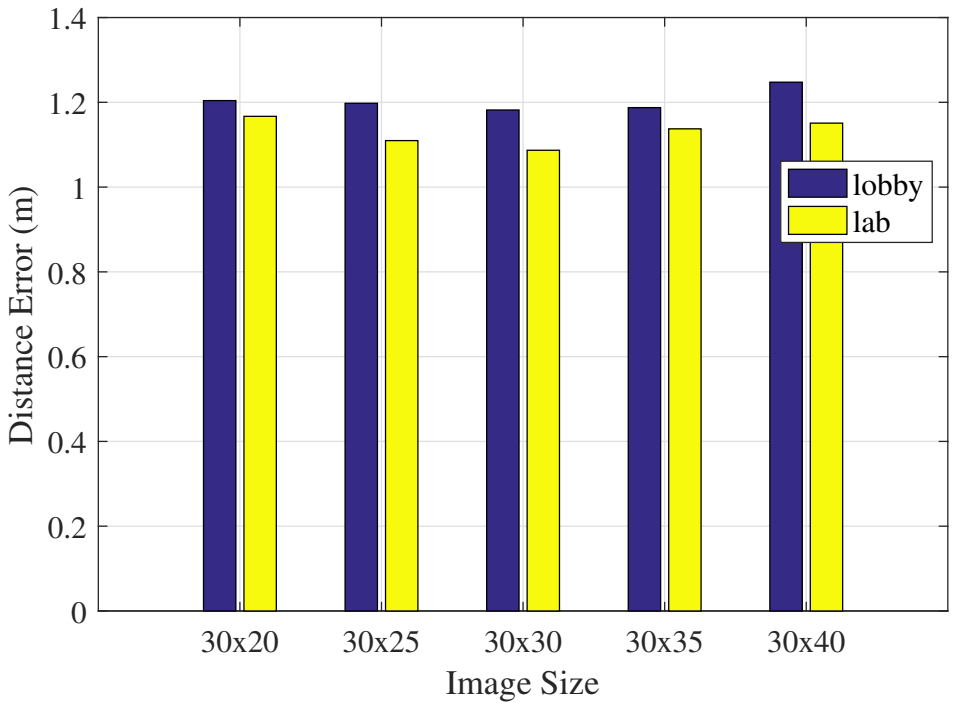

Figure 5.6: The average distance error for different size of pictures.

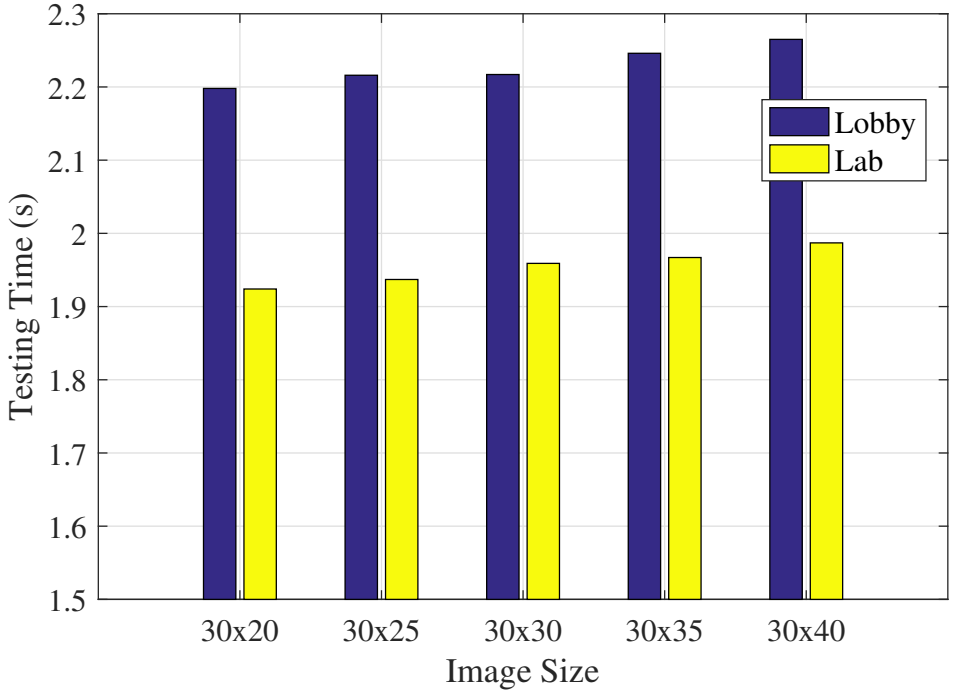

Figure 5.7: The average testing time for different size of pictures.

testing time with different image sizes.As is shown in Figure 5.7,it is apparent for both scenarios that the testing time rises with the increment of the image size. Theoretically, with the same image size, the testing time should be identical in these two scenarios. However, there is a 0.1 second gap between the lab and the lobby. We find that this gap is resulted from the computer performance. Considering the testing time and the distance error, the image size of 30x30 is the best choice for training and testing because of its lowest distance error and acceptable testing time in two scenarios.

Impact of the number of images

To further explore how many the number of pictures affects the distance error, we build 5 datasets with different number of pictures in every position. As is shown in Figure 5.8, the distance error declines with the increase of the number of pictures. The lowest distance errors, 1.0869 meters for the lab and 1.1819 meters for the lobby, are obtained when the number of pictures is 130. This result indicates that the number of pictures is related to the localization accuracy positively. Furthermore, the distance error in the lobby is more sensitive to the number of pictures. We also notice that all distance errors for the lab are smaller than 1.2 meter and

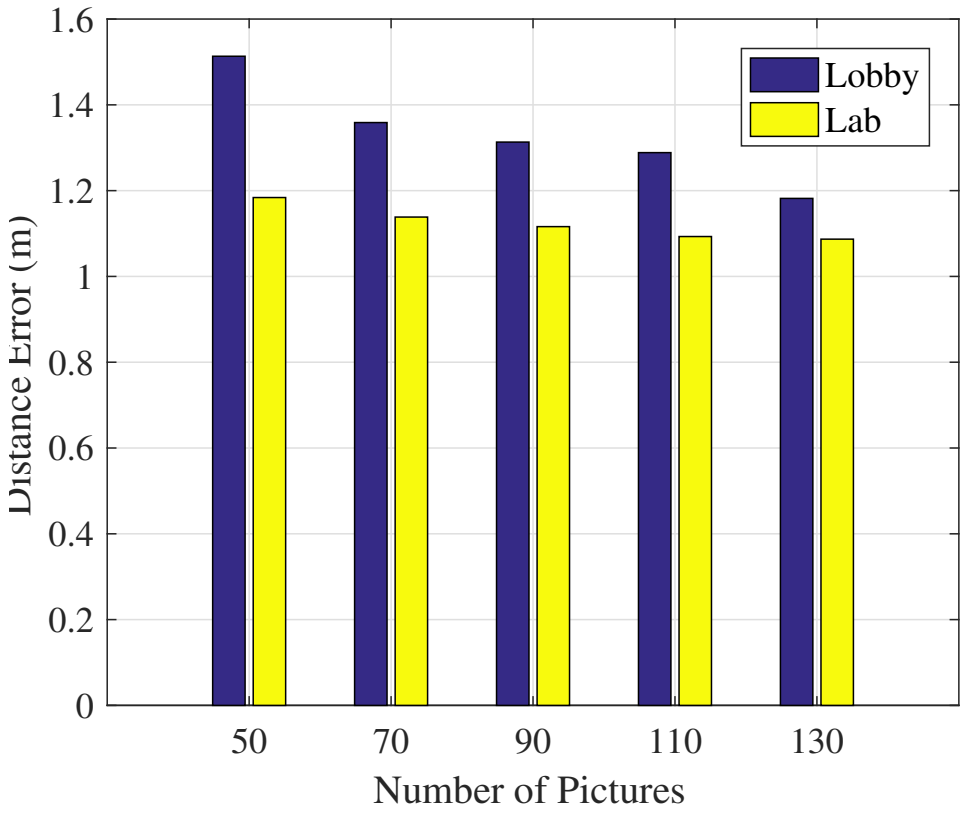

Figure 5.8: The average distance error for different number of pictures.

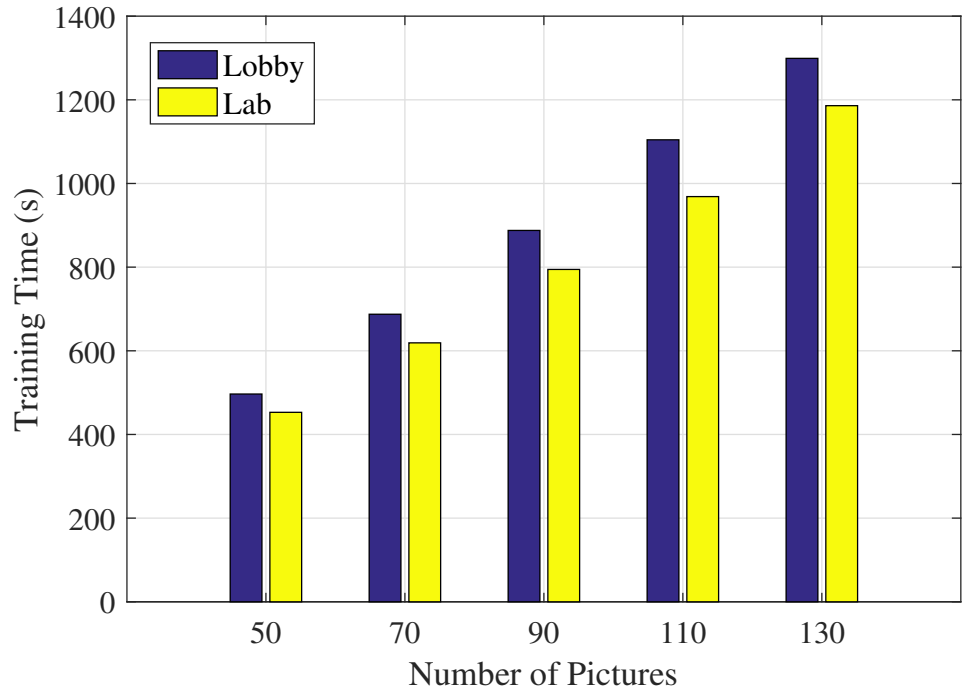

Figure 5.9: The average training time for different number of pictures.

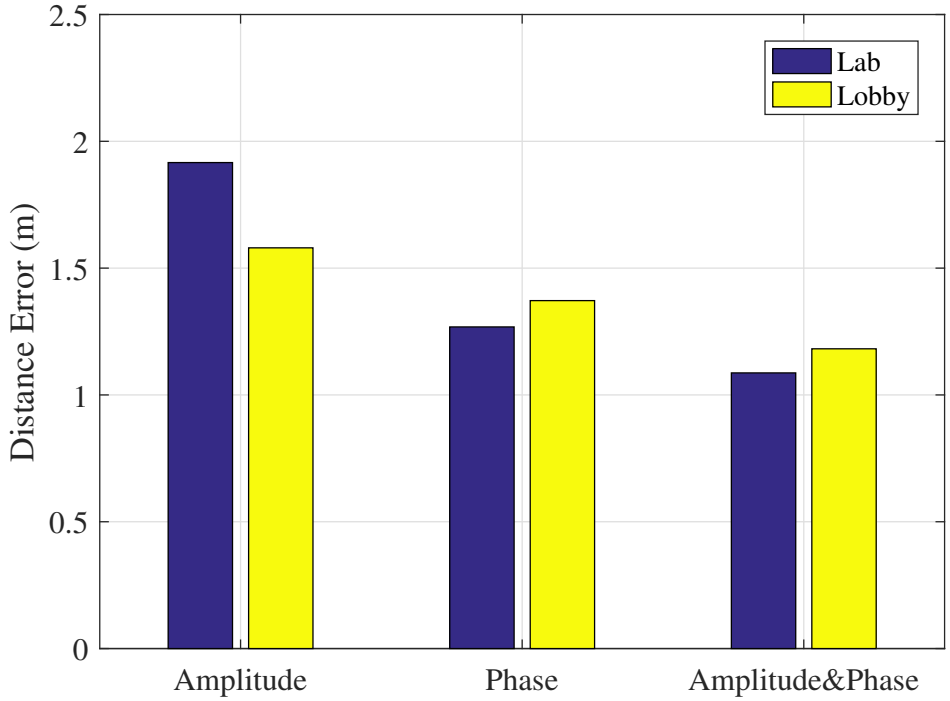

Figure 5.10: The average distance error for different input dataset with two channel model.

distance errors for the lobby are lower than 1.3 meter when the number of pictures is greater than 90. It also shows that the performance of ResLoc is robust to the change of the number of pictures.

Figure 5.9 shows the training time across all datasets with different number of pictures. It is intuitive to show that the training time is directly proportional to the number of pictures. For the same number of pictures, the training time for the lobby is slightly longer than the training time for the lab. Considering that the training process is a part of offline stage, namely the training time does not compromise user experience. Thus, we choose the dataset with 130 pictures in every training point as the input of ResLoc because of the lowest distance error.

Impact of Bimodality

To evaluate the performance of our proposed bimodal input, we also deploy our ResLoc model with different kind of input datasets, the amplitude dataset, the phase difference dataset and the bimodal dataset. We compare the performance of these three datasets in two indoor environments, a computer laboratory and a long corridor. We know that CSI amplitude values reflects channel frequency responses with abundant multipath components and channel fading.

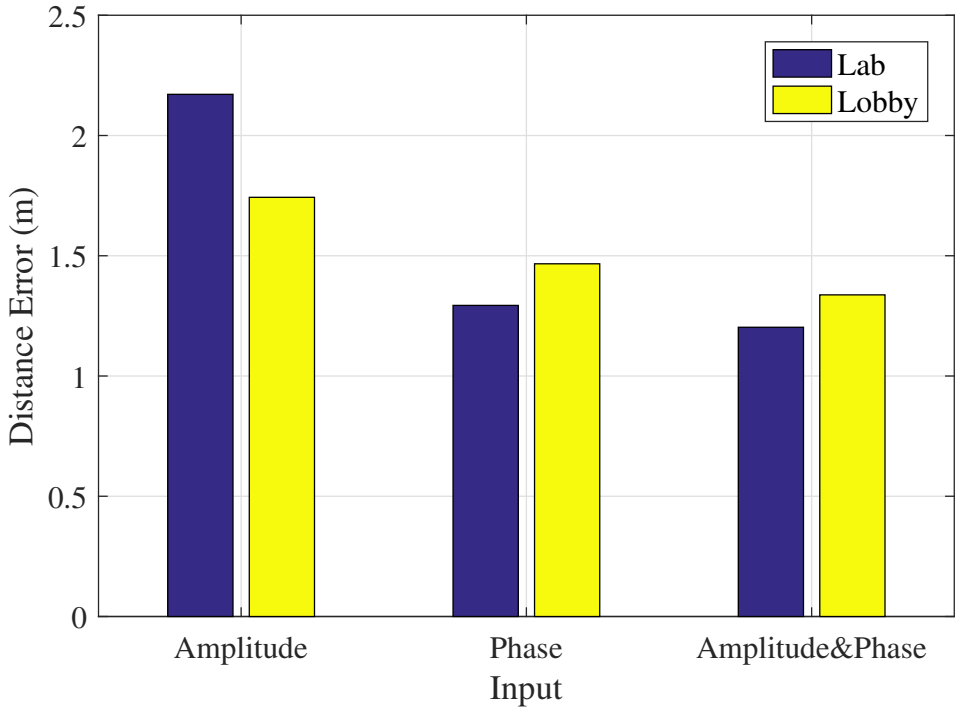

Figure 5.11: The average distance error for different input dataset with one channel model.

In other words, the performance of amplitude dataset is degraded by the indoor environments. The computer lab is a cluttered environment. The furniture, computers, and appliances block most of the LOS paths and generate lots of multipath. As is shown in Figure 5.10, the worst performance is achieved by the amplitude dataset in the lab. Comparing with the amplitude values, the phase values of the signal with the periodical change over the propagation distance is relatively more robust. According to Figure 5.10, we have a lower distance error with the phase difference dataset. The bimodal tensor shows the lowest distance error among three datasets. Due to the use of bi-modal CSI tensor, the phase difference values can be utilized to mitigate the influence of the complex indoor environment. The mean distance errors are 1.0869 meters and 1.819 meters in the lab and the corridor, respectively.

Impact of Channels

We also implement the single channel version ResLoc on these three datasets to investigate the performance difference between the two channel ResLoc and the single channel ResLoc. Figure 5.11 shows a similar trend to previous two channel ResLoc, which means that the complexity of the indoor environment is a dominant effect to the localization performance. However,

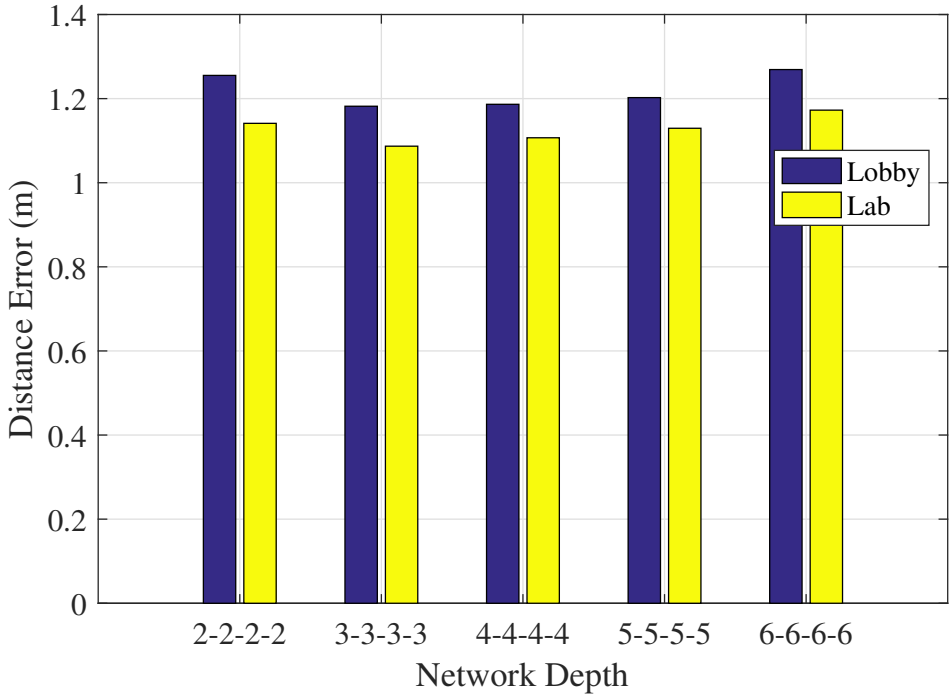

Figure 5.12: The average distance error for different network depth.

it is noticed that all distance errors showed in Fig. 5.11 are larger than corresponding distance errors slightly. The distance error that is obtained by the amplitude dataset in the lab is over 2 meters. For the single channel ResLoc, the lowest distance error, 1.2027 meter, that is obtained by bimodal dataset is still higher than the corresponding distance error, 1.0869 meters, which is the best result of the two channel ResLoc. According to Figure 5.10 and Figure 5.11, it is obvious to reveal that the two-channel architecture enhances the performance of the ResLoc system.

Impact of the number of layers

We now show the impact of different number of layers on the proposed ResLoc system. There are four sizes of basic residual blocks in the ResLoc system. Each residual block is formed by two stacked convolutional layers. In different residual blocks, the number of the feature maps in convolutional layers are different. We set the number of feature maps to 64, 128, 256 and 512 for convolutional layers in different residual blocks. To evaluate how the depth of network affects the performance of the network, we repeat the basic residual block twice, three times, four times, five times and six times, respectively.(For example, 2-2-2-2 means that four kinds

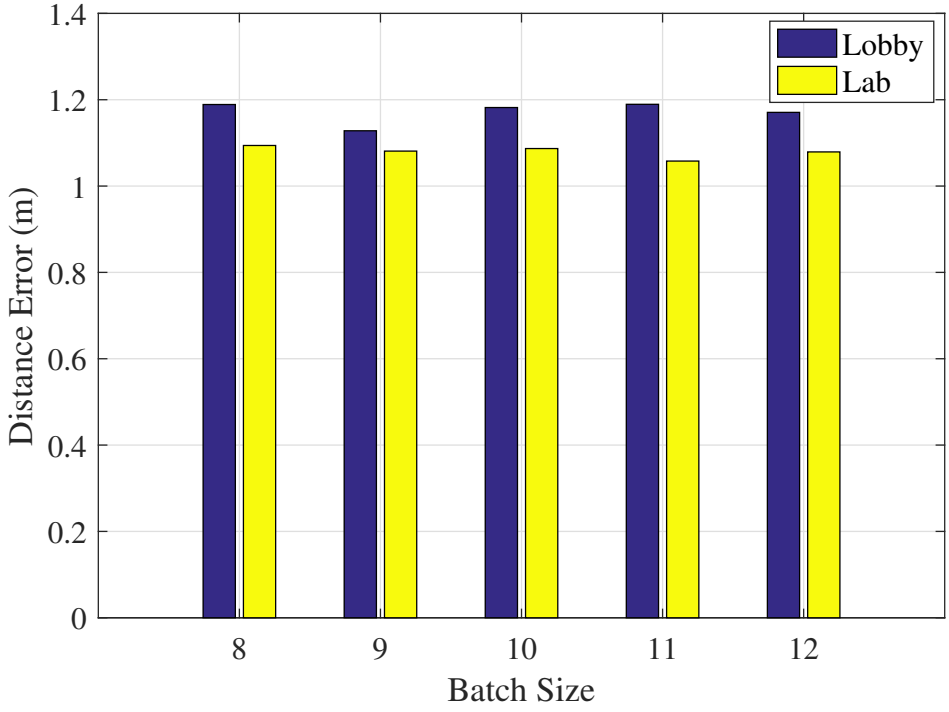

Figure 5.13: The average distance error for different batch size.

residual blocks are repeated twice respectively.) Theoretically, increasing layers may reduce the distance error. However, Figure 5.12 shows that the distance error reaches the lowest point when the network scheme is 3-3-3-3. After that, the distance error rises slightly as the network goes deep. According to Figure 5.12, all distance errors are about 1.2m, which means the distance error is robust to network depth. Thus, we choose 3-3-3-3 scheme as the best scheme to train the network, because of the lowest distance error and a relatively simple scheme.

Impact of batch size

Batch size defines number of samples that can be propagated through the network. We study the impact of batch size on localization accuracy under the two environments. Figure 5.13 illustrates the mean distance errors for increasing batch size in the lab and lobby scenarios. As we can see, there is no relation between the value of batch sizes and the mean distance error. For the lab scenario, the highest mean error is 1.0869 meters and the lowest mean error is 1.0579 meters. The difference between the maximum and minimum of mean error is 0.029 meters, which means the mean distance error is robust enough to the change of the batch size. Similarly, the difference between the maximum and minimum of the mean error is only 0.0613

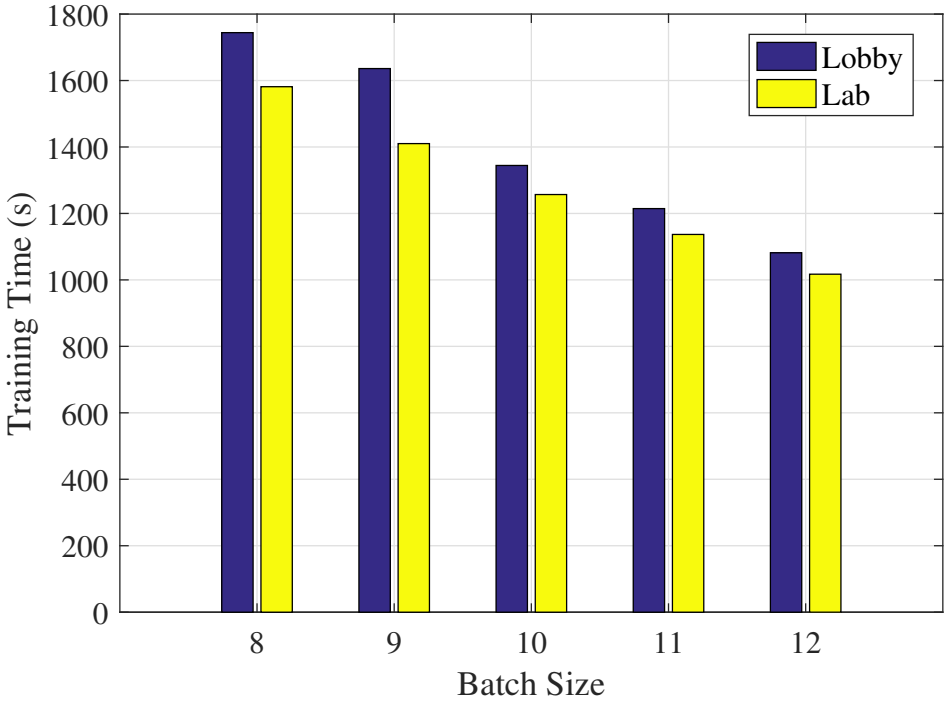

Figure 5.14: The average training time for different batch size.

meters for the lobby scenario. It also shows that the mean distance error is independent to the value of batch size.

Figure 5.14 depicts the training time for different batch sizes. Typically, networks trains faster with mini-batches. We observe that the training time gets shorter with increasing batch size. According to the Figure 5.14, we know the longest training times are 1744 seconds and 1581 seconds in the lobby and the lab, and the shortest training times are 1081 seconds and 1017 seconds in the lobby and the lab.

Impact of batch epoch

To improve the accuracy of ResLoc, we adjust the value of epoch. The impact of epoch on localization accuracy is shown in Figure 5.15. In both indoor environments, the lab and the lobby, the highest distance error is obtained when the value of epoch is 30. Along with the growth of the value of epoch, the distance error keeps decreasing. And it maintains at about 1.1 meters from 50 epochs. Intuitively, the network does not converge before 50 epochs. When the network reaches convergence, the distance error remains at same level. It is noticed that the lowest distance errors in the lab and the lobby are obtained at 50 epochs.

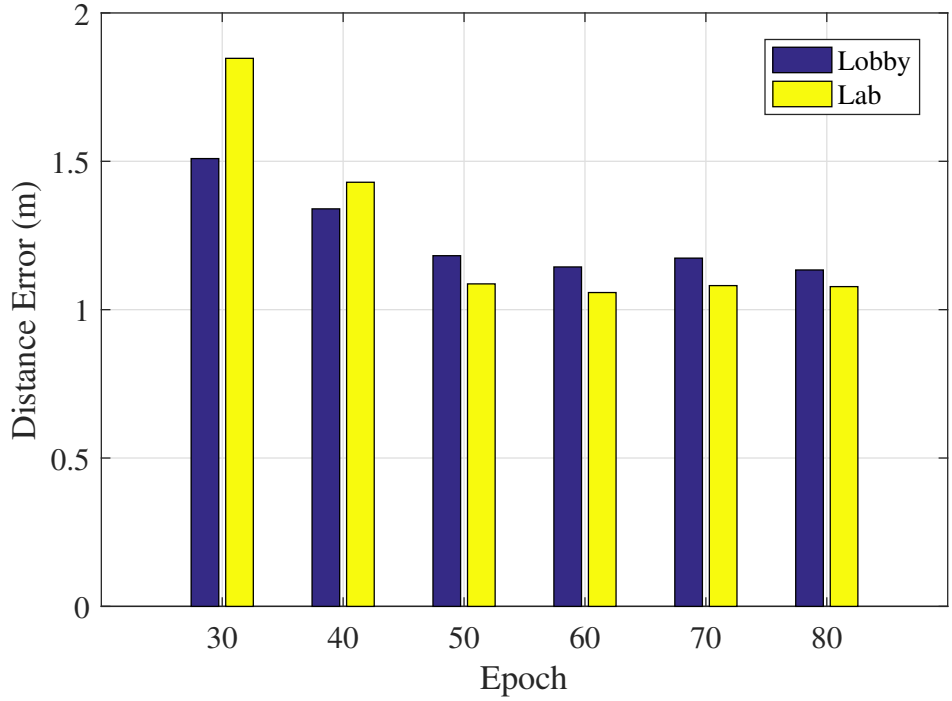

Figure 5.15: The average distance error for different epoch.

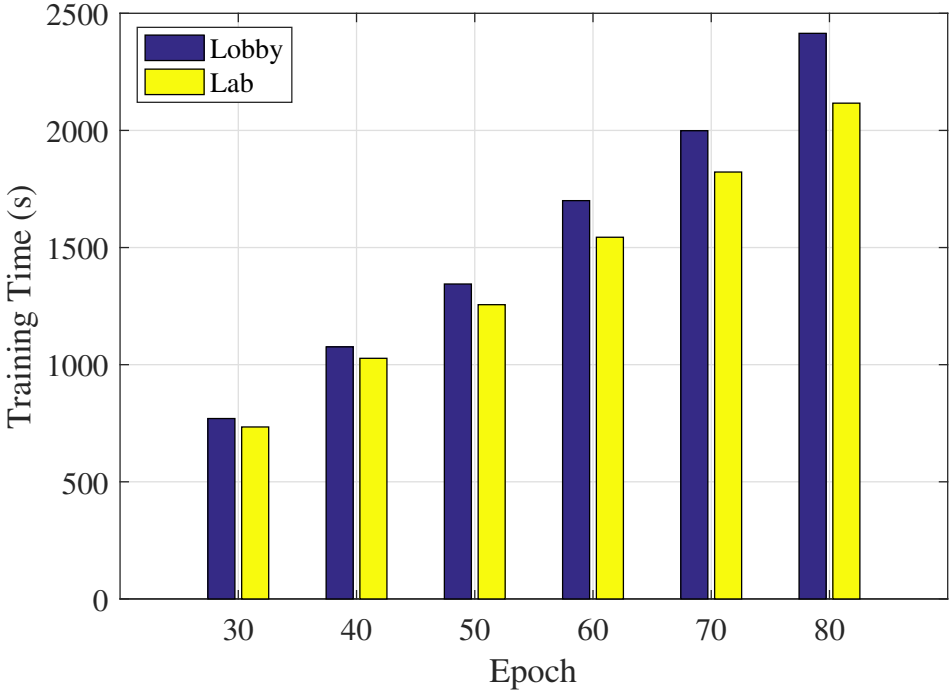

Figure 5.16: The average training time for different epoch.

Figure 5.16 depicts the training time against the value of epoch. As is shown, the training time increases as the value of the epoch increases in both scenarios. It is consistent with our intuitive result that the more epoch loops the more time is consumed. To reach the lowest

distance error, we spend about 1344 seconds and 1256 seconds to train the network in the lobby and the lab respectively.

Chapter 6

Conclusions and Future Work

6.1 Summary

In this paper, we presented ResLoc, a deep residual sharing learning based system for indoor localization with two channels CSI tensor data. We introduced CSI in WiFi network with OFDM system and discussed how to build CSI tensor data for indoor localization. Then, we designed the ResLoc system, which leverages two channels CSI tensor data to train the deep network by using the proposed deep residual sharing learning. For online test, we used newly received CSI tensor data to compute the location of the mobile device based on the probabilistic method. Finally, the experimental results showed the superior performance of the proposed ResLoc system.

6.2 Future work

Even though ResLoc is a high accuracy indoor localization system using off-the shelf WiFi devices, there are still many interesting problems that deserve further study. In the future, we plan to consider the following three aspects for performance enhancement.

To evaluate the performance of ResLoc, we implemented the system in the corridor and lab where no passenger occurred. However, it is impossible to avoid the effect from other passengers in real life. The moving objects and people, generating multipaths and making environment dynamic, lead to significant change to wireless channels. The changes nullify fingerprints we collected. Thus, an effective background elimination algorithm is essential in our future study.

We trained two residual net for two scenarios respectively, which is time-consuming. In the future work, we plan to propose a general localization method based on our ResLoc system. Considering that most of position information is related, we plan to use transfer learning to shrink training time, even zero training time for a new environment.

With fingerprinting based location, we have to collect a huge fingerprint dataset to train the neural network. However, this process is too laborious when the system would be implemented at large public facilities, like airports or shopping malls. To overcome this time-consuming process, we would like to propose an data generating method to generate fingerprints. With the help of such techniques, we could train our ResLoc system with a relatively small dataset.

References

[1] D. Hu, S. Mao, and J. H. Reed, "On video multicast in cognitive radio networks," in *INFOCOM 2009, IEEE*. IEEE, 2009, pp. 2222–2230.

[2] M. X. Gong, B. Hart, and S. Mao, "Advanced wireless lan technologies: Ieee 802.11 ac and beyond," *GetMobile: mobile computing and communications*, vol. 18, no. 4, pp. 48–52, 2015.

[3] Y. Xu and S. Mao, "User association in massive mimo hetnets," *IEEE Systems Journal*, vol. 11, no. 1, pp. 7–19, 2017.

[4] M. Feng, S. Mao, and T. Jiang, "Base station ON-OFF switching in 5G wireless systems: Approaches and challenges," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 46–54, Aug. 2017.

[5] ——, "Joint duplex mode selection, channel allocation, and power control for full-duplex cognitive femtocell networks," *Elsevier Digital Communications and Networks Journal*, vol. 1, no. 1, pp. 30–44, Feb. 2015.

[6] M. Feng, T. Jiang, D. Chen, and S. Mao, "Cooperative small cell networks: High capacity for hotspots with interference mitigation," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 108–116, Dec. 2014.

[7] M. Feng, S. Mao, and T. Jiang, "Boost: Base station on-off switching strategy for energy efficient massive mimo hetnets," in *Proc. IEEE INFOCOM 2016*, San Francisco, CA, Apr. 2016, pp. 1395–1403.

[8] X. Wang, "Deployment of high altitude platforms in heterogeneous wireless sensor network via mrf-map and potential games," in *Wireless Communications and Networking Conference (WCNC), 2013 IEEE*. IEEE, 2013, pp. 1446–1451.

[9] K. Xiao, S. Mao, and J. Tugnait, "MAQ: A multiple model predictive congestion control scheme for cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 16, no. 4, pp. 2614–2626, Apr. 2017.

[10] ——, "Congestion control for infrastructure-based CRNs: A multiple model predictive control approach," in *Proc. IEEE GLOBECOM 2016*, Washington DC, Dec. 2016, pp. 1–6.

[11] ——, "QoE-driven resource allocation for DASH over OFDMA networks," in *Proc. IEEE GLOBECOM 2016*, Washington, DC, Dec. 2016, pp. 1–6.

[12] Z. He, S. Mao, and S. Kompella, "Quality of experience driven multi-user video streaming in cellular cognitive radio networks with single channel access," *IEEE Transactions on Multimedia*, vol. 18, no. 7, pp. 1401–1413, July 2016.

[13] Z. He, S. Mao, and T. Rappaport, "On link scheduling under blockage and interference in 60 ghz ad hoc networks," *IEEE Access Journal*, vol. 3, pp. 1437–1449, Sept. 2015.

[14] Y. Wang and S. Mao, "On distributed power control in full duplex wireless networks," *Elsevier Digital Communications and Networks Journal*, vol. 3, no. 1, pp. 1–10, Feb. 2017.

[15] N. Tang, S. Mao, Y. Wang, and R. Nelms, "LASSO-based single index model for solar power generation forecasting," in *Proc. IEEE GLOBECOM 2017*, Singapore, Dec. 2017.

[16] Y. Wang, S. Mao, and T. Rappaport, "On directional neighbor discovery in mmwave networks," in *Proc. IEEE ICDCS 2017*, Atlanta, GA, June 2017, pp. 1704–1713.

[17] Z. Jiang and S. Mao, "Energy delay trade-off in multi-channel full-duplex wireless LANs," *IEEE Internet of Things Journal*, vol. 4, no. 3, pp. 658–669, June 2017.

[18] X. Li, T. Jiang, S. Cui, J. An, and Q. Zhang, "Cooperative communications based on rateless network coding in distributed mimo systems [coordinated and distributed mimo]," *IEEE wireless communications*, vol. 17, no. 3, 2010.

[19] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Computing Surveys*, vol. 46, no. 2, pp. 25:1–25:32, Nov. 2013.

[20] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proc. ACM NSDI'13*, Lombard. IL, Apr. 2013, pp. 71–84.

[21] 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 *Proc. ACM Mobicom'14*, Maui, HI, Sept. 2014, pp. 617–628.

[22] P. Bahl and V. N. Padmanabhan, "Radar: An in-building RF-based user location and tracking system," in *Proc. IEEE INFOCOM'00*, Tel Aviv, Israel, Mar. 2000, pp. 775–784.

[23] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. ACM MobiSys'05*, Seattle, WA, June 2005, pp. 205–218.

[24] H. Liu, H. Darabi, P. Banerjee, and L. Jing, "Survey of wireless indoor positioning techniques and systems," *IEEE Trans. Syst., Man, Cybern. C*, vol. 37, no. 6, pp. 1067–1080, Nov. 2007.

[25] D. Halperin, W. J. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *Proc. ACM SIGCOMM'10*, New Delhi, India, Sept. 2010, pp. 159–170.

[26] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity WiFi," in *Proc. ACM Mobicom'15*, Paris, France, Sept. 2015, pp. 53–64.

[27] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the Mona Lisa: Spot localization using PHY layer information," in *Proc. ACM MobiSys'12*, Low Wood Bay, UK, Jun. 2012, pp. 183–196.

[28] J. Xiao, K. Wu., Y. Yi, and L. Ni, "FIFS: Fine-grained indoor fingerprinting system," in *Proc. IEEE ICCCN'12*, Munich, Germany, Aug. 2012, pp. 1–7.

[29] X. Wang, L. Gao, S. Mao, and S. Pandey, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[30] X. Wang, L. Gao, and S. Mao, "CSI phase fingerprinting for indoor localization with a deep learning approach," *IEEE Internet of Things J.*, vol. 3, no. 6, pp. 1113–1123, Dec. 2016.

[31] ——, "Biloc: Bi-modal deep learning for indoor localization with commodity 5ghz wifi," *IEEE Access*, vol. 5, pp. 4209–4220, 2017.

[32] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inform. Proc. Syst. 2012*, Lake Tahoe, NV, Dec. 2012, pp. 1106–1114.

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[34] ——, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[35] X. Wang, H. Zhou, S. Mao, S. Pandey, P. Agrawal, and D. Bevly, "Mobility improves LMI-based cooperative indoor localization," in *Proc. IEEE WCNC 2015*, New Orlean, LA, Mar. 2015, pp. 2215–2220.

[36] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4. ACM, 2015, pp. 269–282.

[37] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot *et al.*, "Mastering the game of

go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[38] X. Wang, C. Yang, and S. Mao, "ResBeat: Resilient breathing beats monitoring with online bimodal CSI data," in *Proc. IEEE GLOBECOM 2017*, Singapore, Dec. 2017.

[39] X. Wang, R. Huang, and S. Mao, "Sonarbeat: Sonar phase for breathing beat monitoring with smartphones," in *Proc. ICCCN 2017*, Vancouver, Canada, July/Aug. 2017, pp. 1–8.

[40] ——, "Demo abstract: Sonarbeat: Sonar phase for breathing beat monitoring with smartphones," in *Proc. IEEE SECON 2017*, San Diego, CA, June 2017, pp. 1–2.

[41] X. Wang, C. Yang, and S. Mao, "Phasebeat: Exploiting csi phase data for vital sign monitoring with commodity wifi devices," in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on*. IEEE, 2017, pp. 1230–1239.

[42] X. Li, Y. Zhang, I. Marsic, A. Sarcevic, and R. S. Burd, "Deep learning for RFID-based activity recognition," in *Proceedings of the 14th ACM Conference on Embedded Network Sensor Systems CD-ROM*. ACM, 2016, pp. 164–175.

[43] J. Wang, D. Vasisht, and D. Katabi, "Rf-idraw: virtual touch screen in the air using rf signals," in *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4. ACM, 2014, pp. 235–246.

[44] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard *et al.*, "Learning algorithms for classification: A comparison on handwritten digit recognition," *Neural networks: the statistical mechanics perspective*, vol. 261, p. 276, 1995.

[45] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[46] X. Wang, X. Wang, and S. Mao, "Cifi: Deep convolutional neural networks for indoor localization with 5 ghz wi-fi," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[47] J. Xiao, K. Wu, Y. Yi, and L. M. Ni, "Fifs: Fine-grained indoor fingerprinting system," in *Computer Communications and Networks (ICCCN), 2012 21st International Conference on*. IEEE, 2012, pp. 1–7.

[48] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *Proceedings of the 3rd international conference on Mobile systems, applications, and services*. ACM, 2005, pp. 205–218.

[49] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.

[50] X. Wang, L. Gao, S. Mao, and S. Pandey, "Deepfi: Deep learning for indoor fingerprinting using channel state information," in *Wireless Communications and Networking Conference (WCNC), 2015 IEEE*. IEEE, 2015, pp. 1666–1671.

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

[52] F. J. Ordóñez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *Sensors*, vol. 16, no. 1, p. 115, 2016.

[53] X. Wang, C. Yang, and S. Mao, "Tensorbeat: Tensor decomposition for monitoring multiperson breathing beats with commodity WiFi," *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 1, pp. 8:1–8:27, Sept. 2017.

[54] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, "Phaser: Enabling phased array signal processing on commodity WiFi access points," in *Proc. ACM Mobicom'14*, Maui, HI, Sept. 2014, pp. 153–164.

[55] X. Wang, X. Wang, and S. Mao, "Resloc: Deep residual sharing learning for indoor localization with csi tensors," in *Proc. IEEE PIMRC 2017*, Montreal, Canada, Oct. 2017.

[56] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[57] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[58] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI). Savannah, Georgia, USA*, 2016.