Deep Convolutional Neural Networks for Indoor Localization with CSI Images

Xuyu Wang, *Student Member, IEEE*, Xiangyu Wang, and Shiwen Mao, *Senior Member, IEEE*

Abstract—With the increasing demand of location-based services, Wi-Fi based localization has attracted great interest because it provides ubiquitous access in indoor environments. In this paper, we propose CiFi, deep convolutional neural networks (DCNN) for indoor localization with commodity 5GHz WiFi. Leveraging a modified device driver, we extract phase data of channel state information (CSI), which is used to estimate the angle of arrival (AoA). We then create estimated AoA images as input to a DCNN, to train the weights in the offline phase. The location of mobile device is predicted based using the trained DCNN and new CSI AoA images. We implement the proposed CiFi system with commodity Wi-Fi devices in the 5GHz band and verify its performance with extensive experiments in two representative indoor environments.

Index Terms—Indoor localization, fingerprinting, deep convolutional neural network, 5 GHz commodity Wi-Fi, Channel state information

1 Introduction

The rapid development of mobile devices and wireless techniques has promoted many location-based services, such as indoor tracking, robot navigation, health sensing, and activity recognition [1], [2], [3], [4], [5]. Many such applications require accurately determining the location of a mobile device indoors. Because of the complex wireless propagation in indoor environments, including shadow fading, multipath propagation, and blockage, indoor localization with wireless signals is a challenging problem that has attracted considerable research efforts. In particular, indoor fingerprinting based on Wi-Fi signals has become a research hot-spot, which first builds a database with measurements of Wi-Fi signals in the offline phase, and then determines the location of a mobile device by matching the newly received Wi-Fi data with that stored in the database.

Many Wi-Fi based fingerprinting systems use received signal strength (RSS) as fingerprint, largely because it is easy to obtain (e.g., from a smartphone or laptop) and has low requirements on hardware. The first such work, termed Radar, is to leverage RSS-based fingerprinting with a deterministic method for location estimation [6]. To improve localization accuracy, Horus, another RSS-based fingerprinting scheme, employs a probabilistic method based on K-nearest-neighbor (KNN) [7]. Other RSS-based systems employ various machine learning techniques for improved performance, such as neural networks, support vector machine, and compressive sensing [8]. RSS based fingerprinting has two main shortcomings that limit their performance [9]. First, RSS values for a given location are usually not stable for continuously received packets, due to the complex propagation environment. Second, RSS values only provide coarse channel information.

Recently, open-source device drivers for several Wi-Fi network interface cards (NIC), such as the Intel Wi-Fi Link 5300 NIC [10] and the Atheros AR9580 chipset [11], provide an interface to extract channel state information (CSI) for each received packet. Unlike RSS, CSI represents fine-grained channel information, including subcarrier-level channel measurements in orthogonal frequency division multiplexing (OFDM) systems. Moreover, CSI can capture the multipath effect, and is relatively more stable for a given location. Several indoor fingerprinting systems based on CSI have been proposed for better performance. For instance, FIFS [12] leverages the weighted average of CSI amplitudes over three antennas, while DeepFi [13], [14] exploits the 90 CSI amplitudes from all the subcarriers at all the three antennas with a deep autoencoder network. To address the firmware problem for phase information in the 2.4 GHz band, Phaser [15] first uses CSI phase for angle of arrival (AoA) estimation with 5 GHz Wi-Fi and shows that phase difference is stable with an Intel 5300 NIC.

In this paper, we propose to exploit phase difference data with 5 GHz Wi-Fi to estimate AoA, which is then used for indoor localization. Estimated AoA values for a given location are relatively more stable due to the stability of phase difference data. Thus AoA estimation is highly robust for complex indoor environments. For example, when Wi-Fi signal is blocked by, e.g., chairs or computers, the CSI amplitudes will be strongly weakened. However, the estimated AoA remains the same. Furthermore, we employ the deep convolutional neural network (DCNN) [16] to train the AoA data from all the training locations as a supervised learning. DCNN is a powerful deep learning technique that has been successfully applied for image recognition [17], human

• *The authors are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201. E-mail: {xzw0029, xzw0042}@tigermail.auburn.edu, smao@ieee.org.*

Manuscript received 1 Apr. 2018; revised 31 Aug. 2018; accepted 14 Sept. 2018. Date of publication 19 Sept. 2018; date of current version 5 Mar. 2020. (Corresponding author: Shiwen Mao.)
Recommended for acceptance by H. Wang.
Digital Object Identifier no. 10.1109/TNSE.2018.2871165

activity recognition [18], [19] and social networks [20]. Specifically, we create AoA images based on a large number of received packets as input to the DCNN. The proposed method is to exploit the time-frequency feature of AoA data for improving localization performance. Moreover, since DCNN is a supervised method, *it only requires to train one group of weights for all the training data with related labels*, which is different with our prior work DeepFi that requires training weights for every training location [13], [14]. Thus, the the storage requirement can be greatly reduced.

In particular, we present CiFi, a deep **C**onvolutional neural networks (DCNN) based scheme for **i**ndoor localization with commodity 5 GHz Wi**Fi**. In CiFi, we first obtain 90 CSI data from the three antennas for every received packet from the modified Intel 5300 firmware, and extract the phase information. Then, we compute two sets of CSI data, each including 30 phase differences, from antennas 1 and 2, and from antennas 2 and 3, respectively. The phase difference data is used to estimate AoA. Next, CiFi uses the estimated AoA values from 960 received packets to construct 16 images with size 60×60. These images are then used as input to a DCNN for weight training. For offline training, we use all the constructed images from all training locations to train the DCNN, which consists of a convolutional layer, a subsampling layer, and a fully-connected layer. For the convolutional layer, we obtain the feature map and extract time-space features for AoA images. The mean pooling function is implemented in the subsampling layer to reduce training time. We use the squared error loss function based on back propagation (BP) for weight training. In the online stage, we propose a probabilistic method to predict the location of the mobile device based on the trained DCNN and the new CSI AoA images received from the device.

The main contributions of this paper are summarized as follows.

- We theoretically and experimentally verify the feasibility of exploiting AoA values of CSI data for indoor localization. In particular, we derive a model for measured phase and analyze phase errors. We prove that phase difference is stable, and can be used to estimate AoA.
- This is also the first work to employ DCNN for indoor localization. We use estimated AoA image from CSI data as input to the DCNN. By executing four convolutional and subsampling layers, CiFi can automatically extract the features of the estimated AoA image, to obtain training weights with the BP algorithm. Furthermore, we implement DCNN training algorithm for CSI images. In the online phase, we present a probabilistic method for location estimation.
- We implement the proposed CiFi system with commodity 5 GHz Wi-Fi, and verify its performance in two representative indoor environments with extensive experiments. The results show that CiFi achieves better location accuracy than three existing schemes. Moreover, the impact of various system parameters on CiFi performance is evaluated.

In the remainder of this paper, we provide the preliminaries in Section 2. We present the CiFi design in Section 3 and performance evaluation in Section 4. Related work is discussed in Section 5. Section 6 concludes this paper.

2 Preliminaries

2.1 Channel State Information Preliminaries

The physical layer (PHY) of Wi-Fi systems, such as IEEE 802.11 a/g/n, is based on OFDM. With OFDM, the wireless channel is partitioned into orthogonal subcarriers, each of which is a narrowband flat fading channel. Data is transmitted over these subcarriers, aiming to mitigate frequency selective fading in indoor environments with multiple paths. Recently, OFDM is not only used for wireless communications, but also for wireless sensing and localization. From the device driver of off-the-shelf NICs, such as the Intel 5300 NIC [10] and the Atheros AR9390 chipset [11], we can obtain CSI that represents fine-grained PHY information and provides indoor channel characteristics including shadow fading and multipath effects.

For a Wi-Fi link, each antenna of the receiver with the Intel 5300 NIC can provide CSI values from 30 out of the 56 subcarriers for a 20 MHz or 40 MHz channel. Let H_i denote the CSI value of subcarrier i, which is a complex value defined as

$$H_i = |H_i| \exp(j \angle H_i), \tag{1}$$

where $|H_i|$ and $\angle H_i$ are the amplitude response and phase response of subcarrier i, respectively. To improve indoor localization accuracy, we employ phase difference information between two adjacent antennas, instead of amplitude information as in [13], [14], as the feature for indoor localization.

2.2 Phase Difference Information

In this section, we show that the phase difference values between two adjacent antennas are stable for consecutively received packets. From the Intel 5300 NIC, we can extract CSI phase data, which is usually random and cannot be directly used for indoor localization. This randomness stems from the unsynchronized time and frequency of the transmitter and receiver NICs as well as environment noise. To remove the randomness, two effective methods have been proposed for CSI phase calibration. The first is to implement a linear transformation of the coarse phase values [21], [22], [23]. The second is to leverage the phase difference between two adjacent antennas. This approach has been adopted for 2.4 GHz Wi-Fi in [24]. However, the approach in [24] is to measure the average of phase difference and then remove it, and the proposed scheme is for LOS identification, which requires further steps to obtain an indoor localization solution. Furthermore, in our recent work BiLoc [25], we present a comparison study and show that the 5GHz band has much better performance than the 2.4 GHz band for CSI based indoor localization.

To validate the stability of measured phase difference, we first model the measured phase of subcarrier i as [11], [26], [27], [28]

$$\angle \widehat{H}_i = \angle H_i + (\lambda_p + \lambda_s) m_i + \lambda_c + \beta + Z, \tag{2}$$

where $\angle H_i$ is the true phase, m_i is the index of subcarrier i, β is the initial phase offset of the phase-locked loop (PLL), Z

is the measurement environment noise, λ_p, λ_s, and λ_c are phase errors due to packet boundary detection (PBD), the sampling frequency offset (SFO), and central frequency offset (CFO), respectively [26], [27], [28], given by

$$\lambda_p = 2\pi \frac{\Delta t}{N},\ \lambda_s = 2\pi \left(\frac{T' - T}{T}\right) \frac{T_s}{T_u},\ \lambda_c = 2\pi \Delta f T_s n, \quad (3)$$

where Δt is the packet boundary detection delay, N is the FFT size, T' and T are the sampling periods at the receiver and the transmitter, respectively, T_u is the length of the data symbol, T_s is the total length of the data symbol and the guard interval, n is the sampling time offset for current packet, Δf is the center frequency difference between the transmitter and receiver. Because the device driver only provides CSI data, we do not know the values of Δt, $\frac{T'-T}{T}$, n, Δf, and β in (2) and (3). Further, λ_p, λ_s, and λ_c vary over time because different packets have different Δt and n. Therefore, the measured phase is not a good indicator for the true phase.

Fortunately, the measured phase difference is more stable, which can be employed for indoor localization. The three antennas (radios) are on the same NIC, thus having the same down-converter frequency and the same system clock. Therefore, the measured phase differences on subcarrier i from different antennas have the same frequency difference, packet detection delay, and sampling period [15]. The measured phase difference on subcarrier i is given by

$$\Delta \angle \widehat{H}_i = \Delta \angle H_i + \Delta\beta + \Delta Z, \quad (4)$$

where $\Delta \angle H_i$ is the true phase difference of subcarrier i, $\Delta\beta$ is the unknown difference in phase offsets, which is a constant [15], and ΔZ is the noise difference. We can see from (4) that $\Delta \angle \widehat{H}_i$ is more stable because the random items Δt, Δf, and n are removed.

For example, in the polar coordinate system, Fig. 1 presents a comparison between phase differences (marked as red dots) and phases from a single antenna (marked as blue crosses) of subcarrier 25 for 960 received packets. We can see that the single antenna phase values nearly follow a uniform distribution between 0 and 360 degree. However, the phase difference values of the same subcarrier basically focus on a sector between 130° and 170°.

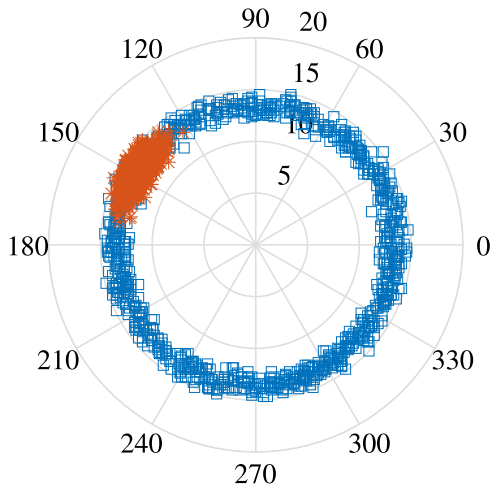


Fig. 1. A comparison between phase differences (marked as red stars) and phases from a single antenna (marked as blue squares) of subcarrier 25 in the polar coordinate plot for 960 back-to-back packets.

After obtaining the measured phase difference, we compute the estimated AoA as

$$\theta_i = \arccos\left(\Delta \angle \widehat{H}_i \lambda / (2\pi d)\right), \quad (5)$$

where d is the distance between two adjacent antennas and λ is the wavelength. In our experiments, we set $d = 0.5\lambda$, while the estimated AoA is in $[0, \pi]$. Because the measured phase difference is relatively more stable, the estimated AoA is also more stable, which can thus be leveraged for precise indoor localization.

2.3 CSI Image Construction

The Intel 5300 NIC provides readings on 30 subcarriers from each of the three antennas. Then, we compute two sets of CSI data, including 30 phase differences between antennas 1 and 2, and 30 phase differences between antennas 2 and 3. Thus, 60 estimated AoA values for each received packet can be obtained using (5). We take 960 packet samples for every training location, and construct 16 images with size 60 × 60 based on the estimated AoA values. Each image consists of 60 packets (row) and the corresponding 60 estimated AoA values for each packet (column). For example, Fig. 2 shows the CSI images for three different

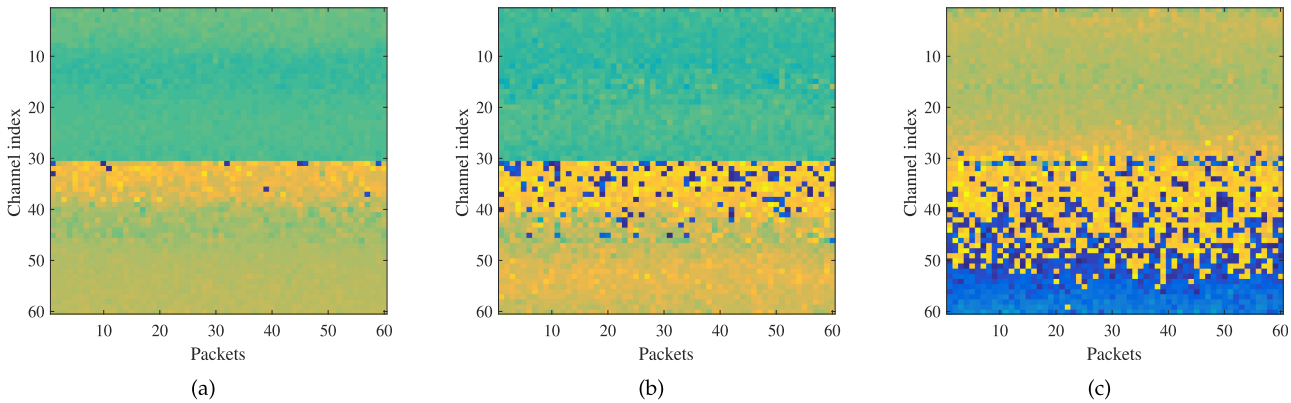


Fig. 2. CSI images for three different locations.

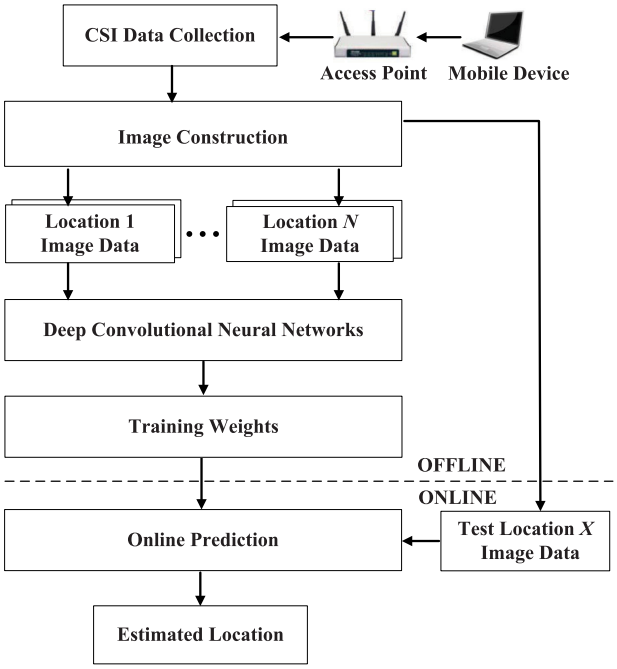


Fig. 3. The CiFi system architecture.

locations. Note that three CSI images have different data distributions, which can be used as fingerprints for indoor localization. In CiFi, the constructed images will be used to train the DCNN.

3 The CiFi System

3.1 CiFi System Architecture

Fig. 3 presents the CiFi system architecture. The CiFi system uses one mobile device and one access point as Wi-Fi transmitter and receiver, respectively, both equipped with the Intel 5300 NIC. The transmitter and receiver are set to the injection and monitor modes, respectively. The 5 GHz band is used for improved channel stability [25]. CiFi employs CSI images for two reasons. First, the estimated AoA values are highly stable for a given location. When Wi-Fi signal is blocked by a wall or chair, the CSI amplitudes will be strongly weakened, which affects the localization accuracy. However, the estimated AoA values are more robust if the transmission distance is not changed. Second, the constructed CSI image can leverage all subcarrier information from all received packets, which contains rich time and frequency features of the channel.

The CiFi procedure includes two stages: offline training and online location predication. In the offline phase, the constructed images from all locations are used to train a DCNN. This method is quite different from traditional fingerprinting based methods, where measurement data is stored for every training location; either the measured raw data or learned features are stored as fingerprints. However, *our CiFi system only trains one group of weights for all the training locations*, which is analogous to classification or regression in machine learning. The proposed method can not only decrease the amount of stored data, but also improve the robustness of the system. In the online phase, we employ an enhanced probabilistic approach for location estimation based on the constructed images of newly received CSI data.

3.2 Offline Training

The DCNN incorporates several convolutional and subsampling layers as well as one or more fully connected layers. It exploits local correlations by sharing the same weights between neurons of adjacent layers, thus reducing the training time. DCNN can also obtain local dependency and scale invariant features from input data. More important, it can extract more abstract representations of the input image data from lower layers to higher layers in the hierarchical architecture, with strengthened feature extraction of CSI AoA data. We introduce three main components of DCNN in the following.

The convolutional layer can extract feature maps within local regions in the previous layer's feature maps with linear convolutional filters followed by nonlinear activation functions. Denote θ_i^l as the ith feature map in layer l, defined as

$$\theta_i^l = \sigma\left(\sum_{m \in \mathcal{S}_{l-1}} w_{im}^l * \theta_m^{l-1} + b_i^l\right), \tag{6}$$

where $\sigma(t) = \frac{1}{1+\exp(-t)}$ is the sigmoid function, b_i^l is the bias of the ith feature map in layer l, $\mathcal{S}_{l-1}$ is the set of feature maps in layer $(l-1)$ that connect to the current feature map, w_{im}^l is the convolutional kernel to generate the ith feature map in layer l, which is the same for different m due to local weights sharing. The convolution operation can obtain the shift-invariance of input data and extract robust features. Then, the activation function $\sigma(t)$ is used to avoid obtaining trivial linear combinations of input data.

The subsampling layer or the pooling layer is to 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 in the inputs. The feature maps in the previous layer are pooled over a local temporal neighborhood by the mean pooling function, as

$$\theta_{ij}^{l+1} = \frac{1}{\left|\mathcal{G}_j^l\right|} \sum_{k \in \mathcal{G}_j^l} \theta_{ik}^l, \tag{7}$$

where $\mathcal{G}_j^l$ is the set of pooling region for the jth value in feature map i in layer l, which is identical for all i (so the index i is omitted for brevity), θ_{ik}^l is the kth value of feature map i in layer l. Other methods such as the sum or max pooling function can be also used in this stage to reduce training time.

The fully-connected layer consists of a basic neural network with one hidden layer to train the output data of the convolutional and subsampling layers. A loss function is employed to measure the difference between true location label and output data of the DCNN. By minimizing the values of the loss function with the BP algorithm, we can update the convolutional weights with the stochastic gradient descent (SGD) method. In the proposed DCNN, we use the squared error loss function, which is defined as

$$\mathcal{E} = \frac{1}{2K} \sum_{i=1}^{K} (y_i - o_i)^2, \tag{8}$$

where K is the number of training locations, y_i is the true label of location i, and o_i is the DCNN output for location i.

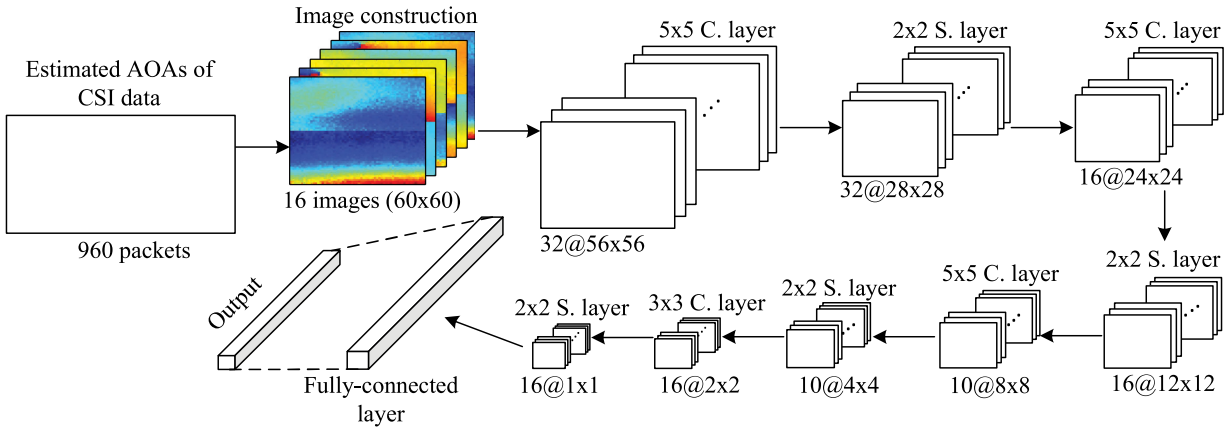


Fig. 4. CSI data training using the DCNN (C. and S. indicate convolutional and subsampling, respectively).

Fig. 4 illustrates CSI data training with the DCNN. To obtain input AoA images, we first estimate AoA value for each of the 960 received packets as in (5). Then, we construct 16 images, each with size 60×60, out of the 960 AoA values. The images are convenient for the DCNN to process in its convolution and subsampling layers. For each input image in the first convolutional and subsampling layer, we employ 32 convolutional filters with size 5×5 to obtain the same number of feature maps with size 56×56. To reduce training data and guarantee the invariance of feature maps, the same number of feature maps with size 28×28 can be obtained by subsampling with size 2×2. Then, by implementing other three convolutional and subsampling layers as in Fig. 4, we can obtain 16 feature maps with size 1×1. Finally, we obtain the forward output results and then combine the labels of training data, which can be used to update the training weights such as the convolutional filers using the loss function with the BP algorithm.

Algorithm 1. CiFi Weight Training Algorithm

Input: CSI images from all training locations, location labels, network architecture, Max_epoch, and learning rate α ;
Output: Trained weights w and b;
Randomly initialize w and b;
while $epoch < Max\_epoch$ **do**
  Randomly select a mini-batch from inputs;
  //Forward propagation;
  // L is the number of layers of the DCNN;
  for $l = 2 : L-2$ **do**
    if *the current layer is a convolution layer* **then**
      $\theta_i^l = \sigma\left(\sum_{m \in \mathcal{S}_{l-1}} w_{im}^l * \theta_m^{l-1} + b_i^l\right)$;
    else
      //The current layer is a subsampling layer layer;
      $\theta_{ij}^{l+1} = \frac{1}{|\mathcal{G}_j^l|} \sum_{k \in \mathcal{G}_j^l} \theta_{ik}^l$;
  // The last layer is a fully-connected layer;
  $v = \text{Dense}(\theta^{L-1})$;
  $o = \sigma(w^L \times v + b^L)$;
  //Loss function;
  $\mathcal{E} = \frac{1}{2K} \sum_{i=1}^{K} (y_i - o_i)^2$;
  Call the DCNN BP algorithm;

Algorithm 2. DCNN BP Algorithm

//Compute δ^{L-1} as delta value of layer $L-1$;
// $\odot$ denotes the element-wise product;
$\delta^{L-1} = \left(w^L\right)^T \times (o - y) \odot (o \odot (1 - o))$;
$\delta_i^L = Reshape\left(\delta^{L-1}\right)$;
//Reshape δ^{L-1} into feature map style; i is the index of feature maps in layer $L-1$;
for $l = L-2 : 2$ **do**
  if *the current layer is a subsampling layer layer* **then**
    for $i = 1 : M_l$ **do**
      // M_l is the number of feature maps in layer l ;
      $\delta_i^l = \sum_{m \in S_l} \delta_m^{l+1} * rot_{180}(w_{i,m}^{l+1})$;
  else
    for $i = 1 : M_l$ **do**
      $\text{Upsampling}(\delta_i^{l+1}) = \delta_i^{l+1} \otimes \varphi$;
      // φ is an all-ones matrix with size of scale $\times$ scale;
      // $\otimes$ denotes the Kronecker product ;
      $\delta_i^l = \text{Upsampling}(\delta_i^{l+1}) \odot \sigma'(\theta_i^l)$;
// Update weights;
for $l = 2 : L-1$ **do**
  if *the current layer is a convolution layer* **then**
    for $j = 1 : M_l$ **do**
      while $i \in \mathcal{S}_j$ **do**
        $w_{i,j}^l = w_{i,j}^l - \alpha \times (Mean_{\{mini-batch\}}(rot180(\theta_i^{l-1}) * \delta_j^l))$;
        $//Mean_{\{mini-batch\}}$ is the average of the results over mini-batch data;
      $b_j^l = b_j^l - \alpha \times (Mean_{\{mini-batch\}}(\delta_j^l))$;
  else
    $w^l = w^l - \alpha \times Mean_{\{mini-batch\}}(\mathcal{E} \odot (o \odot (1 - o) \times v^T))$;
    $b^l = b^l - \alpha \times Mean_{\{mini-batch\}}(\mathcal{E} \odot (o \odot (1 - o)))$;

The pseudocode for offline training of CiFi is presented in Algorithms 1 and 2. The inputs to Algorithm 1 are CSI images from all training locations, location labels, Max_epoch and learning rate. First, we randomly initiate all weights and biases (step 3). Then, for each epoch, we randomly select a mini-batch from CSI images from all training locations, which are fed into the DCNN defined by the network architecture (step 5). In the proposed CiFi system, the first layer and the last layer are the input layer and the output layer, respectively. From the second layer, the input data is processed by the convolutional layer and down sampling layer sequentially (step 8-16). The outputs of the last third layer are compressed as the inputs to the fully-connected layer (step

18-19). Based on the outputs of the fully-connected layer and location labels, the loss function is used to measure the difference between true location label and output data of DCNN (step 21). After forward propagation, the errors between network outputs and labels are used as inputs to the BP algorithm to train the DCNN.

The pseudocode for the DCNN BP Algorithm is presented in Algorithm 2. We calculate the values of delta for every layer and convolutional kernel, which are used to update weights and biases. First, the errors are calculated as the difference between outputs of the neural network and labels, which are for computing delta in layer $L-1$ (step 2). Because the inputs to the fully connected layer are compressed data from the previous layer in forward propagation, its shape should be restored in the DCNN BP algorithm (step 4). To obtain the delta values for the current layer, if the current layer is a sub-sampling layer, the weights of the later layer are rotated 180 degrees and convoluted with the delta values from the later layer (step 10). Specifically, connected to the feature maps in the current layer, only kernels from the later layer are calculated in this step. When the current layer is a convolutional layer, the delta values for the later layer are upsampled by the Kronecker product (step 15). Then, the delta values for the current layer are obtained by the element-wise product between the upsampled delta values and derivatives of the sigmoid function (step 18). We define the values of scale as the quotient of the size of feature maps in the previous layer and the current layer. Depending on the delta values of each layer, the training weights are updated (step 22-37). The learning rate α controls the speed of adjusting the weights of the DCNN. We will examine its impact in Section 4. The mean gradient over the mini-batch is calculated because a random mini-batch from CSI images from all training locations is fed into the DCNN in each epoch.

3.3 Online Algorithm

In the online phase, we adopt a probabilistic method to predict the location of the mobile device, using the trained DCNN and newly received CSI AoA images from the mobile device. Let M denote the number of images from one location, and o_{ij} be the prediction output of the DCNN for location i using image j. We obtain a matrix $\boldsymbol{O}$ as the output of the DCNN for K training locations using the M images, given by

$$\boldsymbol{O} = \begin{bmatrix} o_{11} & o_{12} & o_{13} & \dots & o_{1M} \\ o_{21} & o_{22} & o_{23} & \dots & o_{2M} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ o_{K1} & o_{K2} & o_{K3} & \dots & o_{KM} \end{bmatrix}. \tag{9}$$

With matrix $\boldsymbol{O}$, we propose a greedy method to select R candidate locations and compute a weighted average of these locations as the estimated location of the mobile device. We first select location indexes of the R largest outputs from the DCNN in every column of matrix $\boldsymbol{O}$, thus producing a new matrix $\boldsymbol{S}$ with size $R \times M$ as

$$\boldsymbol{S} = \begin{bmatrix} s_{11} & s_{12} & \dots & s_{1j} & \dots & s_{1M} \\ s_{21} & s_{22} & \dots & s_{2j} & \dots & s_{2M} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ s_{R1} & s_{R2} & \dots & s_{Rj} & \dots & s_{RM} \end{bmatrix}, \tag{10}$$

where s_{ij} is the location index of the ith largest output of image j. Every element of matrix $\boldsymbol{S}$ belongs to the set of location indexes $\{1, 2, \dots, K\}$. The R largest location indexes are obtained by computing the frequencies of all location indexes in matrix $\boldsymbol{S}$. Moreover, the weight of the location i index can be computed by averaging all the selected outputs for the location i index, denoted as p_i.

The position of the mobile device is estimated as a weighted average of the R selected locations, as

$$\hat{L} = \sum_{i=1}^{R} l_i \times \frac{p_i}{\sum_{i=1}^{R} p_i}, \tag{11}$$

where l_i is the ith training location. We set $R = 2$ in our experiments for better localization performance.

4 Experimental Study

4.1 Experiment Configuration

We utilize a Dell desktop computer and a Dell laptop as access point and mobile device, respectively. Both devices are equipped with an Intel 5300 NIC. The operating system is Ubuntu desktop 14.04 LTS OS. We set the PHY parameters as QPSK modulation and 1/2 coding rate for the OFDM system. We set the access point in the monitor model and the distance between its two adjacent antennas is $d = 2.68$ cm, i.e., a half wavelength for 5 GHz Wi-Fi on channel 116. The mobile device is set in the injection model with one antenna. Using the packet injection technique with LORCON version 1, we can extract 5 GHz CSI data from the receiver NIC.

We compare CiFi with three representative approaches, including DeepFi [13], [14], FIFS [12], and Horus [7]. To guarantee a fair comparison, we use the same CSI dataset in the 5 GHz band for estimating the position of the mobile device. We experiment with the four schemes in the following two indoor environments.

Computer Laboratory: This is a 6×9 m^2 computer laboratory in the Broun Hall in the Auburn University campus. The indoor space is a cluttered environment with many desktop computers, chairs, and tables, which block most of the LOS paths. The floor plan is shown in Fig. 5. We use 15 training locations (marked as red squares) and 15 test locations (marked as green dots). The access point is put at the center of the room. We set the distance between two adjacent training locations to 1.8 m, and obtain CSI data from 1000 packets for each training position and test position.

Corridor: This is a long corridor in Broun Hall with dimension 2.4×24 m^2. As shown in Fig. 6, we place the access point at one end on the floor. The LOS path is the dominant component in this scenario. We use 10 training locations (red squares) and 10 test locations (green dots) along a straight line. The distance between two adjacent training locations is also 1.8 m. We extract 5 GHz CSI data from 1000 packets for each training and test location.

4.2 Accuracy of Location Estimation

Fig. 7 presents the training errors over iterations of the DCNN, for the laboratory and corridor experiments. We set the threshold of training error to 0.06 to guarantee successful training and to avoid overfitting. Moreover, the iterations indicate the times of training input AoA images with

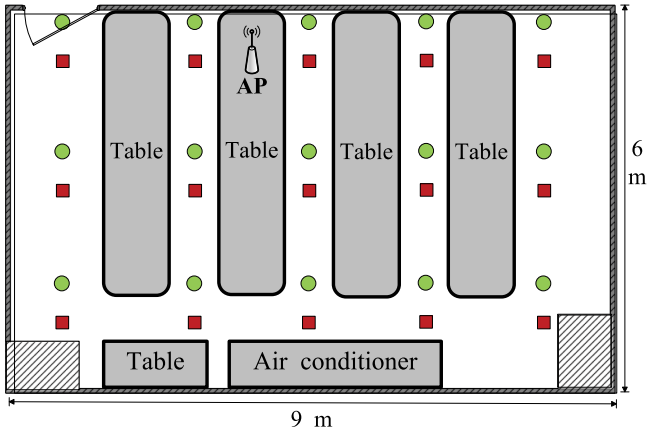


Fig. 5. Layout of the computer laboratory: training locations are marked as red squares and testing locations are marked as green dots.

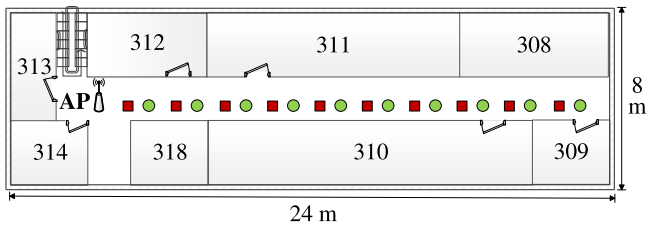


Fig. 6. Layout of the corridor: training locations are marked as red squares and testing locations are marked as green dots.

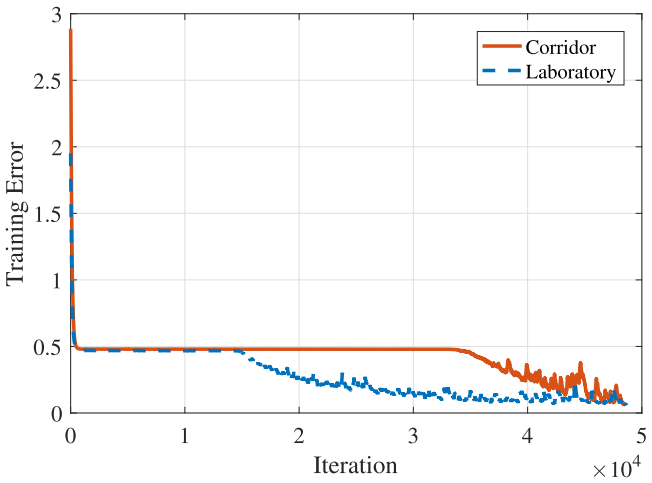


Fig. 7. Convergence of training errors for the laboratory and corridor experiments.

the DCNN. For the laboratory experiments, the training error curve starts to converge after 1.48×10^4 iterations, and finally reaches the preset threshold of 0.06 after 4.85×10^4 iterations. For the corridor case, the training error curve begins to converge after 3.33×10^4 iterations, which is slower, and eventually reaches the preset threshold after 4.86×10^4 iterations.

Tables 1 and 2 present the mean and standard deviation (STD) of localization errors, as well as the execution times for the four schemes in the two indoor environments, respectively. In the laboratory experiments, the proposed CiFi scheme achieves a mean error of 1.7882 m and an STD error of 1.2489 m. For the corridor environment, CiFi achieves a mean error of 2.3863 m and an STD error of 1.4575 m. CiFi outperforms the other three schemes in both cases. This is because

TABLE 1
Localization Error and Execution Time (Laboratory)

Algorithm	*Mean error (m)*	*Std. dev. (m)*	*Mean execution time (s)*
CiFi	1.7882	1.2489	0.5496
DeepFi	2.0411	1.3804	0.3340
FIFS	2.7151	1.0805	0.2918
Horus	3.0537	1.0623	0.2849

TABLE 2
Localization Error and Execution Time (Corridor)

Algorithm	*Mean error (m)*	*Std. dev. (m)*	*Mean execution time (s)*
CiFi	2.3863	1.4575	0.6484
DeepFi	2.8953	2.5665	0.3707
FIFS	4.4296	3.4256	0.2535
Horus	4.8000	3.5242	0.2505

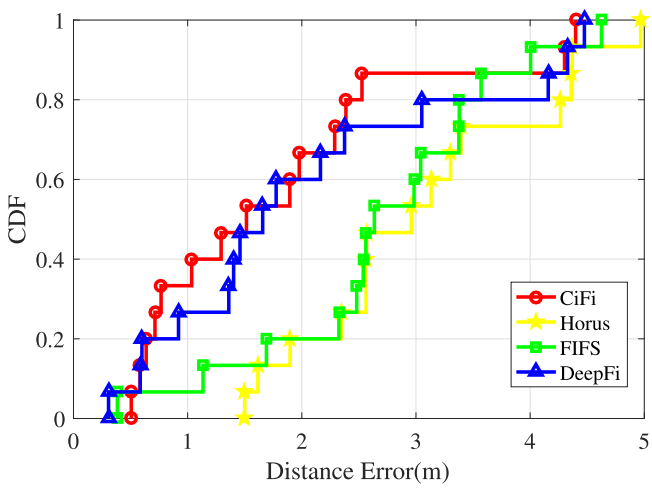


Fig. 8. CDF of localization errors for the laboratory experiment.

CiFi utilizes AoA estimation, which is more stable and robust in complex indoor environments, compared to RSS or CSI amplitude used in the benchmark methods. We use the standard offline training in DCNN, where its computation time greatly depends on the deep learning software used. Because some operations such as convolution can be implemented using tensors in Tensorflow, the computation speed can be greatly accelerated. Since more test packets are used to construct AoA images in the online phase, the mean execution time of CiFi is the highest among all the schemes. The mean execution time of CiFi for the computer laboratory and corridor cases are 0.5496 s and 0.6484 s, respectively, which, however, are still sufficient for realtime indoor localization.

Fig. 8 presents the cumulative distribution function (CDF) of distance errors of the four schemes in the computer laboratory case. For this environment with complex multipaths, CiFi can utilize the unique multiple path features for location estimation. CiFi has 40 percent of the test locations having an error less than or equal to 1 m, while that for the other schemes is 30 percent. We also find that about 87 percent of the test locations for CiFi have an error under 3 m, while the percentage of test locations having a smaller error than 3 m are 73, 60, and 52 percent for DeepFi, FIFS,

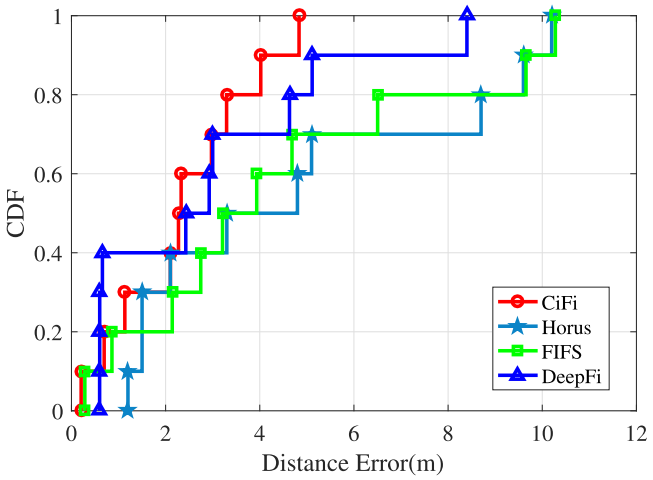


Fig. 9. CDF of localization errors for the corridor experiment.

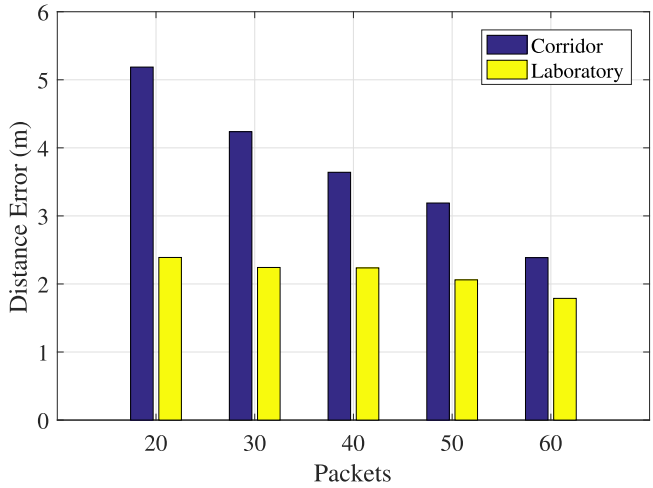


Fig. 10. Mean localization errors for different numbers of training packets in the laboratory and corridor experiments.

and Horus, respectively. Thus, CiFi achieves the best performance in this experiment. This is because, when the magnitude of wireless signal is always influenced by obstacles such as computers in the laboratory environment, the estimated AoA values of CiFi are more robust to the indoor multipath environment, which results in smaller localization errors.

Fig. 9 presents the CDF of localization errors in the corridor environment. We can see that the maximum error for CiFi is 4.8 m, while that for the other schemes is more than 8 m. This validates that the CiFi system is more robust than the other three schemes. Moreover, about 60 percent of the test locations for CiFi and DeepFi have an error under 3 m, while it is 40 percent for FIFS and Horus. CiFi achieves a close localization performance to that of DeepFi in this scenario, while both outperform the other two schemes. However, unlike DeepFi, the proposed CiFi system does not require to build up a database for every training location, thus greatly reducing the storage requirement.

4.3 Impact of Various System Parameters

4.3.1 Impact of Number of Training Packets

To evaluate the effect of training packets, we construct training CSI images with different numbers of packets. From each received packet, we can extract 30 phase values from 30 subcarriers on each antenna. In the proposed CiFi system, the size of a CSI image is 60×60, which consists of 60 AoA values estimated by two antenna pairs (in columns) from 60 received packets (in rows). When the number of packets is smaller than 60, estimated AoA values can be duplicated until the image is completely filled.

Fig. 10 shows the mean localization errors for different numbers of packets in the laboratory and corridor experiments. We evaluate the performance of CiFi with five training datasets that each contains training images constructed by a different number of received packets. We find the distance error decreases as the number of packets is increased in both scenarios. The minimum distance error, i.e., 2.386 m in the corridor cases and 1.788 m in the lab case, are achieved when 60 packets are used to generate a training image. Apparently more AoA information contributes to improved precision. We choose 60 packets in all other experiments to balance between accuracy and complexity.

4.3.2 Impact of Number of Training Images

To study the impact of the number of training images, we create five datasets, which contain different numbers of images for each training location, in the two indoor environments. For the sake of fairness, 60 AoA values estimated from two antenna pairs are used to generate CSI images in all the datasets. We set the packet transmit rate to 1000 Hz, which guarantees 16 images be generated within 1 second.

Fig. 11 illustrates the mean localization errors for different number of training images in the laboratory and corridor experiments. As the number of images per training location is decreased, the mean localization error increases. When 16 images are generated for each training location, the mean distance errors reach 1.788 m and 2.386 m in the lab and corridor experiments, respectively. The largest distance errors are 2.781 m and 3.282 m in the lab and corridor cases, respectively, which are acceptable for many location-based services. In other words, our CiFi system can not only achieve a better performance with a larger input dataset but also achieve an acceptable precision with a limited number of training images.

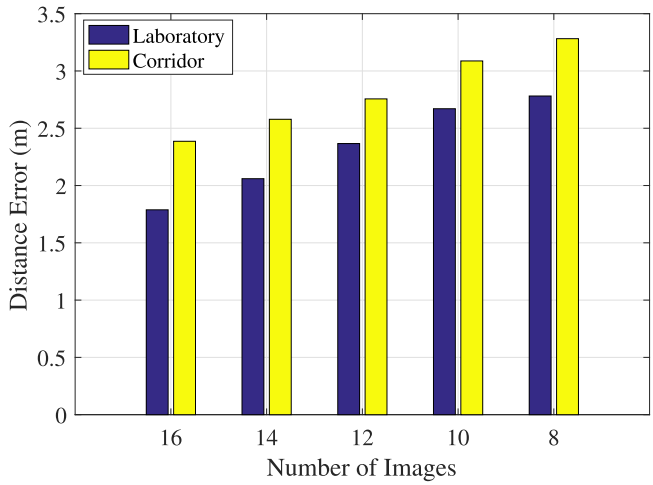


Fig. 11. Mean localization errors for different numbers of training images in the laboratory and corridor experiments.

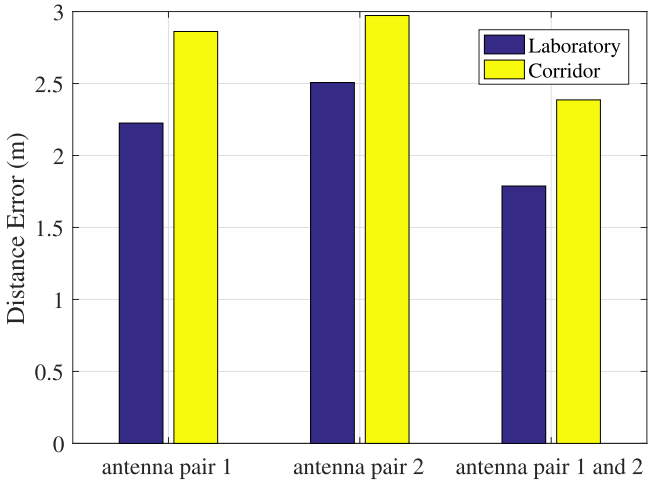


Fig. 12. Mean localization errors for different antenna pairs in the laboratory and corridor experiments.

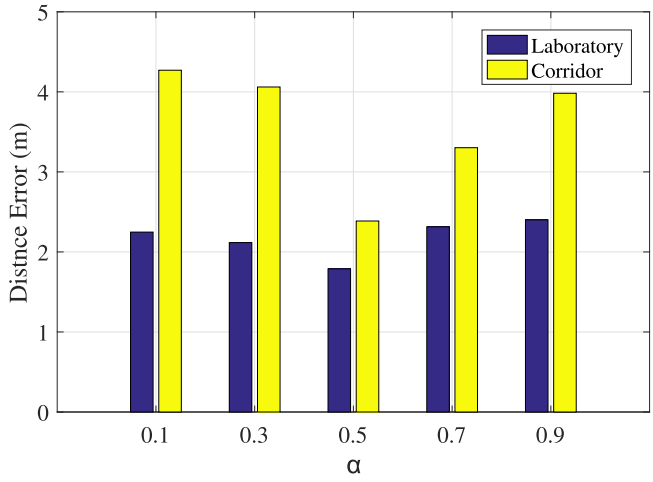


Fig. 13. Mean localization errors for increased learning rate α in the laboratory and corridor experiments.

4.3.3 Impact of Antenna Pairs

Since Intel Wi-Fi Wireless Link 5300 has three antennas and the CSI data could be collected from all three antennas for each received packet, we construct three datasets to study the impact of different antenna pairs. For each packet, 30 AoA values could be estimated from 30 subcarriers of each antenna pair. Similarly, we also construct a training image with the size of 60×60. If only one antenna pair is used, the image of estimated AoA values is duplicated and concatenated together, i.e., a 30×60 image is generated first with 60 packets collected from an antenna pair, and then the image is duplicated and concatenated with itself to generate a 60×60 image.

Fig. 12 presents the mean localization errors for different antenna pairs in the laboratory and corridor experiments. It is obvious that the best localization precisions are achieved in both cases when CSI AoA values from all antennas are leveraged to construct training images. Furthermore, we notice that our CiFi system performs well even when input images are produced by one antenna pair. The highest distance errors in the lab and corridor are 2.507 m and 2.972 m, respectively, where antenna 2 and 3 are leveraged to generate training images.

4.3.4 Impact of Learning Rate α

We next design a specific experiment by setting different learning rates to evaluate their effect on localization precision. In the experiment, the number of epochs is set to 1200 to guarantee fairness.

Fig. 13 illustrates the mean errors for increased α in the laboratory and corridor experiments. As the learning rate is increased from 0.1 to 0.5, the minimum distance errors for both scenarios are obtained when the learning rate is 0.5. After that, the mean distance error goes up with the increase of α. Basically, a low learning rate may not allow the DCNN to converge within 1200 epochs. However, for a higher leaning rate such as 0.7 or 0.9, the DCNN could not reach the best convergence point because the BP algorithm hops back and forth over the valley repeatedly. For CiFi, the training time does not jeopardize user experience in the offline stage. Thus, in order to reach the lowest distance error, the learning rate is set to 0.5 for both scenarios.

4.3.5 Impact of Candidate Locations R

In the proposed CiFi system, we propose a greedy method to select R candidate locations and then use their weighted average as the estimated location. In our experiment, we find that most of the correct location predictions are always included in the top five outputs of the DCNN. Thus, to improve localization precision, only the top five outputs are leveraged for location estimation in CiFi.

Fig. 14 shows the mean localization errors for increased R in the laboratory and corridor experiments. When the value of R is 2, the errors are the lowest in both scenarios. Thus, we set R to 2 in CiFi. Furthermore, with increased R, the mean distance error becomes only slightly larger, which means that CiFi is robust with respect to R.

4.3.6 Impact of Number of Training Locations K

To evaluate the effect of the gap between neighboring training locations, five training and testing datasets are created from the corridor scenario, each with a different numbers of training locations. The training dataset with a gap of 1.2 m includes 15 training locations; the training dataset with a 1.5 m gap contains 14 training locations; the

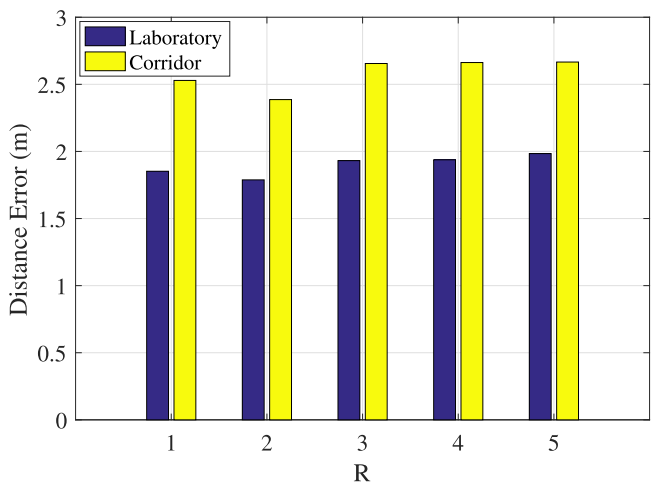


Fig. 14. Mean localization errors for increased R in the laboratory and corridor experiments.

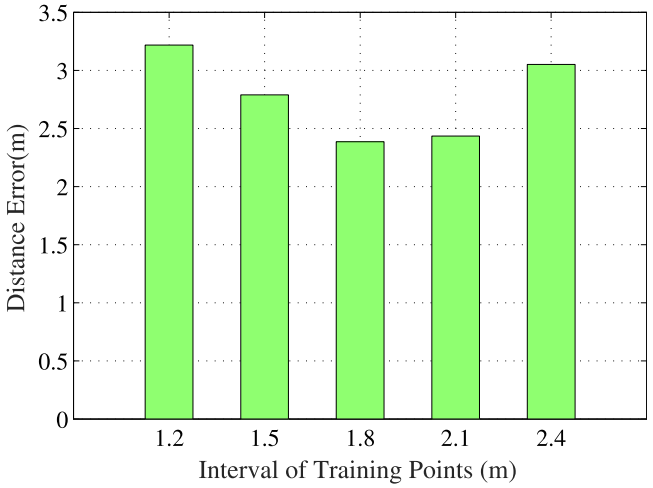


Fig. 15. Mean localization errors for different gaps between adjacent training locations in the corridor scenario.

dataset of 10 training locations are obtained with a gap of 2.1 m, and that of 9 training locations are obtained with a gap of 2.4 m. All the testing locations are set in the middle of two neighboring training locations. For fairness, all training datasets are processed with the same parameter setting in CiFi.

Fig. 15 presents the mean localization errors for different gaps between adjacent training locations in the corridor experiment. The mean distance error reaches the lowest value 2.3863 m when the interval is 1.8m. The mean error keeps going up as the gap is further increased. It can be seen that the training location gap has a significant impact on the localization performance. When the gap becomes small, CSI data collected from the training locations and testing locations become more similar. As a result, it is difficult for DCNN to make a precise prediction. In fact, the distance error does not increase significantly because of the small gaps. On the other hand, when the gap becomes larger, the CSI data collected from training locations vary widely. Thus, the testing data could be matched with the correct training location more easily. However, a large gap contributes to more built-in error. Thus, there is a trade-off between built-in error and prediction precision.

5 Related Work

Indoor localization is an essential part of many applications and a well-studied problem [29], [30]. We discuss two types of indoor localization technologies related to the proposed CiFi system in this section, i.e., Fingerprinting-based and AOA-based techniques.

Fingerprinting based indoor localization uses different kinds of fingerprints such as WiFi [31], magnetism [32], light [31], FM radio [33], acoustic signals [34], and RFID [35]. WiFi is widely used because of its ubiquitous availability and low hardware requirement. The first work based on WiFi is RADAR [6], which establishes fingerprints with RSS using one or more access points and uses a deterministic method for location estimation. Based on RSS, a probabilistic method is proposed in the Horus system to improve indoor localization accuracy [7]. On the other hand, CSI based fingerprinting can mitigate the instability of RSS values, thus enhancing the localization performance. FIFS is the first fingerprinting localization system using CSI amplitudes [12]. Other schemes are also proposed based on multilevel discrete wavelet transform [36], fusion between CSI amplitude and magnetic [37], and CSI-based time-reversal fingerprinting [38]. To improve accuracy, three different indoor localization systems, i.e., DeepFi [14], PhaseFi [23], and BiLoc [25], are proposed by employing a deep autoencoder network with CSI amplitude, CSI calibrated phase, and bimodal CSI values, respectively. These three deep learning based methods require training CSI weights for every training location. Moreover, they do not consider CSI image features. The proposed CiFi method with DCNN is a supervised learning method, which can not only reduce the amount of stored data, but also increase the robustness of the localization system.

AoA-based indoor localization is to estimate the incoming angles with multiple antennas and then apply triangulation to localize. Even though this method achieves a high accuracy, the number of antennas that is available at mobile devices limits the performance of such systems. For example, CUPID [39] uses the MUSIC algorithm [40] to achieve a mean error of about 20° with three antennas. To improve the resolution of antenna array, more antennas are required [41]. Recently, researchers propose many novel ideas to overcome the limit of the number of antennas. WiDeo [42] and ROArray [43] leverage sparse recovery to retrieve ToA and AoA, instead of only estimating AoA. Also, SpotFi achieves a high accuracy using MUSIC with signal smoothing and multiple subcarrer values [44]. IndoTrack proposes a Doppler-AoA method to combine Doppler velocity and AoA spectrum information to jointly estimate the absolute trajectory [45]. Even though AoA based approaches achieve high precision, they are usually computationally expensive and may not be practical for real-time applications.

6 Conclusions

In this paper, we proposed CiFi, a DCNN based fingerprinting system for indoor localization with 5 GHz Wi-Fi. We theoretically and experimentally verified the feasibility of using AoA values for indoor localization. We then presented the CiFi system, which first formed AoA images to train the DCNN, and then used newly received AoA images to estimate the location of the mobile device. Through extensive experiments, we demonstrated the superior performance of the proposed CiFi system.

Acknowledgments

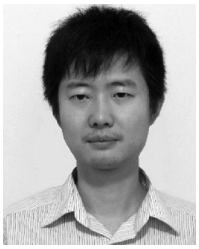
This work is supported in part by the NSF under Grant CNS-1702957 and through the Wireless Engineering Research and Education Center (WEREC) at Auburn University. This work was presented in part at IEEE ICC 2017, Paris, France, May 2017.

References

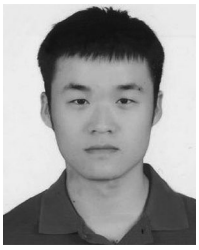
[1] X. Wang, X. Wang, and S. Mao, "CiFi: Deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi," in *Proc. IEEE Int. Conf. Commun.*, May 2017, pp. 1–6.

[2] D. Zhang, S. Zhao, L. T. Yang, M. Chen, Y. Wang, and H. Liu, "NextMe: Localization using cellular traces in internet of things," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 302–312, Apr. 2015.

[3] C. Qiu and M. W. Mutka, "Airloc: Mobile robots assisted indoor localization," in *Proc. IEEE 12th Int. Conf. Mobile Ad Hoc Sensor Syst.*, 2015, pp. 407–415.
[4] 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 20th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2014, pp. 617–628.
[5] A. Ghosh and S. Sarkar, "Pricing for profit in Internet of Things," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 2, pp. 130–144, Apr.-Jun. 2018.
[6] P. Bahl and V. N. Padmanabhan, "RADAR: An in-building RF-based user location and tracking system," in *Proc. IEEE 19th Annu. Joint Conf. IEEE Comput. Commun. Societies*, Mar. 2000, pp. 775–784.
[7] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in *Proc. ACM 3rd Int. Conf. Mobile Syst. Appl. Services*, Jun. 2005, pp. 205–218.
[8] 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.
[9] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput. Surveys*, vol. 46, no. 2, pp. 25:1–25:32, Nov. 2013.
[10] D. Halperin, W. J. Hu, A. Sheth, and D. Wetherall, "Predictable 802.11 packet delivery from wireless channel measurements," in *Proc. ACM SIGCOMM Conf.*, Sep. 2010, pp. 159–170.
[11] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity WiFi," in *Proc. ACM 21st Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2015, pp. 53–64.
[12] J. Xiao, K. Wu, Y. Yi, and L. Ni, "FIFS: Fine-grained indoor fingerprinting system," in *Proc. IEEE 21st Int. Conf. Comput. Commun. Netw.*, Aug. 2012, pp. 1–7.
[13] X. Wang, L. Gao, S. Mao, and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Mar. 2015, pp. 1666–1671.
[14] 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.
[15] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson, "Phaser: Enabling phased array signal processing on commodity WiFi access points," in *Proc. ACM 20th Annu. Int. Conf. Mobile Comput. Netw.*, Sep. 2014, pp. 153–164.
[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.
[17] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1106–1114.
[18] F. J. Ordonez and D. Roggen, "Deep convolutional and LSTM recurrent neural networks for multimodal wearable activity recognition," *MDPI Sensors*, vol. 16, Jan. 2016, Art. no. 115.
[19] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc. 6th Int. Conf. Mobile Comput. Appl. Services*, Nov. 2014, pp. 197–205.
[20] B. Mei, Y. Xiao, R. Li, H. Li, X. Cheng, and Y. Sun, "Image and attribute based convolutional neural network inference attacks in social networks," *IEEE Trans. Netw. Sci. Eng.*, to be published, doi: 10.1109/TNSE.2018.2797930.
[21] K. Qian, C. Wu, Z. Yang, Y. Liu, and Z. Zhou, "PADS: Passive detection of moving targets with dynamic speed using PHY layer information," in *Proc. IEEE 20th IEEE Int. Conf. Parallel Distrib. Syst.*, Dec. 2014, pp. 1–8, doi: 10.1109/TNSE.2018.2797930.
[22] X. Wang, L. Gao, and S. Mao, "PhaseFi: Phase fingerprinting for indoor localization with a deep learning approach," in *Proc. IEEE Global Commun. Conf.*, Dec. 2015, pp. 1–6.
[23] X. Wang, L. Gao, and S. Mao, "CSI phase fingerprinting for indoor localization with a deep learning approach," *IEEE Internet Things J.*, vol. 3, no. 6, pp. 1113–1123, Dec. 2016.
[24] C. Wu, Z. Yang, Z. Zhou, K. Qian, Y. Liu, and M. Liu, "PhaseU: Real-time LOS identification with WiFi," in *Proc. IEEE Conf. Comput. Commun.*, Apr. 2015, pp. 2038–2046.
[25] X. Wang, L. Gao, and S. Mao, "BiLoc: Bi-modal deep learning for indoor localization with commodity 5GHz Wi-Fi," *IEEE Access J.*, vol. 5, no. 1, pp. 4209–4220, Mar. 2017.
[26] M. Speth, S. Fechtel, G. Fock, and H. Meyr, "Optimum receiver design for wireless broad-band systems using OFDM–Part I," *IEEE Trans. Commun.*, vol. 47, no. 11, pp. 1668–1677, Nov. 1999.
[27] X. Wang, C. Yang, and S. Mao, "PhaseBeat: Exploiting CSI phase data for vital sign monitoring with commodity WiFi devices," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, Jun. 2017, pp. 1230–1239.
[28] X. Wang, C. Yang, and S. Mao, "Tensorbeat: Tensor decomposition for monitoring multiperson breathing beats with commodity WiFi," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 1, pp. 8:1–8:27, Sep. 2017.
[29] S. He and S.-H. G. Chan, "Wi-fi fingerprint-based indoor positioning: Recent advances and comparisons," *IEEE Commun. Surveys Tutorials*, vol. 18, no. 1, pp. 466–490, Jan.-Mar. 2016.
[30] B. Spinelli, E. Celis, and P. Thiran, "A general framework for sensor placement in source localization," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 2, pp. 86–102, Apr.-Jun. 2018.
[31] C. Zhang and X. Zhang, "Litell: Robust indoor localization using unmodified light fixtures," in *Proc. 22nd Annu. Int. Conf. Mobile Comput. Netw.*, 2016, pp. 230–242.
[32] X. Wang, Z. Yu, and S. Mao, "DeepML: Deep LSTM for indoor localization with smartphone magnetic and light sensors," in *Proc. IEEE Int. Conf. Commun.*, May 2018, pp. 1–6.
[33] S. Yoon, K. Lee, and I. Rhee, "FM-based indoor localization via automatic fingerprint db construction and matching," in *Proc. 11th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2013, pp. 207–220.
[34] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: Mobile phone localization via ambience fingerprinting," in *Proc. 15th Annu. Int. Conf. Mobile Comput. Netw.*, 2009, pp. 261–272.
[35] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil, "LANDMARC: Indoor location sensing using active RFID," *Wireless Netw.*, vol. 10, no. 6, pp. 701–710, 2004.
[36] S.-H. Fang, W.-H. Chang, Y. Tsao, H.-C. Shih, and C. Wang, "Channel state reconstruction using multilevel discrete wavelet transform for improved fingerprinting-based indoor localization," *IEEE Sensors J.*, vol. 16, no. 21, pp. 7784–7791, Nov. 2016.
[37] 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 Mobile Comput.*, vol. 41, pp. 150–165, 2017.
[38] C. Chen, Y. Chen, Y. Han, H.-Q. Lai, and K. R. Liu, "Achieving centimeter-accuracy indoor localization on WiFi platforms: A frequency hopping approach," *IEEE Internet Things J.*, vol. 4, no. 1, pp. 111–121, Feb. 2017.
[39] S. Sen, J. Lee, K.-H. Kim, and P. Congdon, "Avoiding multipath to revive inbuilding WiFi localization," in *Proc. 11th Annu. Int. Conf. Mobile Syst. Appl. Services*, 2013, pp. 249–262.
[40] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propagation*, vol. 34, no. 3, pp. 276–280, Mar. 1986.
[41] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *Proc. ACM 10th USENIX Conf. Netw. Syst. Des. Implementation*, Apr. 2013, pp. 71–84.
[42] K. R. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "Wideo: Fine-grained device-free motion tracing using RF backscatter," in *Proc. 12th USENIX Conf. Netw. Syst. Des. Implementation*, 2015, pp. 189–204.
[43] W. Gong and J. Liu, "Robust indoor wireless localization using sparse recovery," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst.*, 2017, pp. 847–856.
[44] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using WiFi," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, 2015, pp. 269–282.
[45] D. Zhang, H. Wang, and D. Wu, "Toward centimeter-scale human activity sensing with Wi-Fi signals," *IEEE Comput.*, vol. 50, no. 1, pp. 48–57, Jan. 2017.



Xuyu Wang (S'13) received the BS degree in electronic information engineering and the MS degree in signal and information processing from Xidian University, Xi'an, China, in 2009 and 2012, respectively. Since 2013, he has been working toward the PhD degree in electrical and computer engineering at Auburn University, Auburn, AL. His research interests include indoor localization, deep learning, software defined radio, and big data. He received a Woltolsz fellowship at Auburn University, and is a co-recipient of the Second Prize of Natural Scientific Award of Ministry of Education, China in 2013, the Best Demo Award of IEEE SECON 2017, and the Best Student Paper Award of IEEE PIMRC 2017. He is a student member of the IEEE.



Xiangyu Wang received the BS degree in electrical engineering from the Taiyuan Institute of Technology, Taiyuan, China in 2014, and the MS degree in electrical and computer engineering (ECE) from Auburn University, Auburn, AL, in 2017. He has been working toward the PhD degree in ECE at Auburn University since Spring 2018. His research interests include machine learning, indoor localization, and IoT. He is a co-recipient of the Best Student Paper Award of IEEE PIMRC 2017.

Shiwen Mao (S'99-M'04-SM'09) received PhD degree in electrical and computer engineering from Polytechnic University, Brooklyn, NY. He is the Samuel Ginn distinguished professor, and director of the Wireless Engineering Research and Education Center (WEREC) at Auburn University, Auburn, AL. His research interests include 5G wireless, Internet of Things, and smart grid. He is a Distinguished Speaker for 2018-2021 and was a Distinguished Lecturer for 2014-2018 of the IEEE Vehicular Technology Society. He is on the editorial board of the *IEEE Transactions on Mobile Computing*, the *IEEE Transactions on Multimedia*, the *IEEE Internet of Things Journal*, the *IEEE Multimedia*, ACM GetMobile, among others. He received the Auburn University Creative Research & Scholarship Award in 2018, the 2017 IEEE ComSoc ITC Outstanding Service Award, the 2015 IEEE ComSoC TC-CSR Distinguished Service Award, the 2013 IEEE ComSoc MMTC Outstanding Leadership Award, and the NSF CAREER Award in 2010. His student won the Best Student Paper Award at IEEE PIMRC 2017. He is a co-recipient of the 2018 MMTC Best Conference Paper Award, the Best Paper Awards from IEEE GLOBECOM 2016, IEEE GLOBECOM 2015, IEEE WCNC 2015, and IEEE ICC 2013, the Best Demo Award from IEEE SECON 2017, and the 2004 IEEE Communications Society Leonard G. Abraham Prize in the Field of Communications Systems. He is a senior member of the IEEE.