CiFi: Deep Convolutional Neural Networks for Indoor Localization with 5GHz Wi-Fi

Xuyu Wang, Xiangyu Wang, and Shiwen Mao
Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849-5201, USA
Email: {xzw0029, xzw0042}@auburn.edu, smao@ieee.org

***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. First, by leveraging a modified device driver, we extract phase data of channel state information (CSI), which is used to estimate angle of arriving (AOA). We then create estimated AOA images as input to the DCNN, to train the weights in the offline phase. The location of mobile device is predicted based on 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; deep learning; deep convolutional neural networks; angle of arriving (AOA); channel state information; phase difference.**

I. Introduction

The rapid development of mobile devices and wireless techniques has promoted location-based services for the Internet of Things, such as indoor tracking, health sensing, and activity recognition [1]. These applications require accurately determining the location of a mobile device indoors. Because of the complex wireless propagation in indoor environments, due to shadow fading, multipath propagation, and blockage, indoor localization with wireless signals is a challenging problem that has attracted considerable research efforts. Recently, indoor fingerprinting based on Wi-Fi signals has become a research hot-spot, which first builds a database with a large amount of Wi-Fi measurements in the offline phase, and then determines the location of a mobile device by matching the newly received Wi-Fi signal with that 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, smartwart, or laptop) and has low requirement on hardware. The first such work, termed Radar, is to leverage RSS-based fingerprinting with a deterministic method for location estimation [2]. To improve localization accuracy, Horus, another RSS-based fingerprinting scheme, employs a probabilistic method with K-nearest-neighbor (KNN) [3]. Other RSS-based systems employ various machine learning methods for improved performance, such as neural networks, support vector machine, and compressive sensing [4]. RSS based fingerprinting has two main shortcomings [1]. First, RSS values for a given location are not stable among continuously received packets, due to the complex propagation environment. Second, RSS values only provide coarse channel information.

Recently, modified device drivers for several Wi-Fi network interface cards (NIC), such as Intel Wi-Fi Link 5300 NIC [5] and the Atheros AR9580 chipset [6], provide an interface to extract channel state information (CSI) for received packets. 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 instance, FIFS [7] leverages the weighted average of CSI amplitudes over three antennas, while DeepFi [8], [9] and PhaseFi [10], [11] exploit 90 CSI amplitude and calibrated phases data, respectively, 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 [12] first uses CSI phase for angle of arrival (AOA) estimation with 5GHz Wi-Fi and shows that phase difference is more stable with Intel 5300 NIC.

In this paper, phase difference data with 5GHz Wi-Fi is used to estimate AOA, which can be useful 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 robust for complexity 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 when the transmitter location is not changed. Furthermore, we employ the deep convolutional neural network (DCNN) [13] 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 [14] and human activity recognition based on sensors [15] and RFID [16], respectively. Specifically, we create AOA images based on a 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 [9]. Thus, the proposed method can greatly reduce the storage requirement.

978-1-4673-8999-0/17/$31.00 ©2017 IEEE

In particular, we present CiFi, a deep **C**onvolutional neural networks (DCNN) based scheme for **i**ndoor localization with commodity 5GHz Wi**Fi**. In CiFi, we first obtain 90 CSI data from the three antennas for every received packet and extract all 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 next used to estimate AOA. 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 the DCNN. 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 the time-space feature 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) to train the convolutional weights. 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 below.

- We theoretically 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 with Wi-Fi. 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. In the online phase, we present an enhanced probabilistic method for location estimation.
- We implement the proposed CiFi system with commodity 5GHz Wi-Fi, and verify its performance in two representative indoor environments with extensive experiments. The results show that CiFi achieves better location accuracy compared with three existing schemes.

In the remainder of the paper, we provide the preliminaries and phase difference analysis in Section II. We present the CiFi design in Section III and performance evaluation in Section IV. Section V concludes this paper.

II. Preliminaries and Phase Difference Analysis

A. Channel State Information Preliminaries

The physical layer (PHY) of Wi-Fi systems (such as IEEE 802.11 a/g/n) and LTE is based on OFDM [17]. 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 by using the same modulation and coding scheme (MCS), 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, we can obtain CSI that presents fine-grained PHY information, which can provide indoor channel characteristics including shadow fading and multipath effect.

For a Wi-Fi link with OFDM, 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 [9], as the feature for indoor localization.

B. Phase Difference Information

From the Intel 5300 NIC, we can extract CSI phase data, which is highly 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, we consider the phase difference values between two adjacent antennas, which are highly stable for consecutively received packets.

To validate the stability of measured phase difference in 5GHz, we first model the measured phase of subcarrier i, as [6], [18], [19]

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

where $\angle H_i$ is the true phase of CSI data, 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 [18], [19], given by

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

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 Intel 5300 NIC 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 on subcarrier i 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 difference on subcarrier i have the same frequency difference, packet detection delay, and sampling period [12]. The measured phase difference on subcarrier i is given by

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

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 [12], and ΔZ is the noise difference. We can see from (4) that $\Delta\angle\widehat{H}_i$ is more stable over different received packets because the random items Δt, Δf, and n are removed.

After obtaining the measured phase difference, we compute the estimated AOA, which will be used as input to the DCNN. The estimated AOA on subcarrier i is computed by

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

where d is the distance between two adjacent antennas and λ is the wavelength. In our experiments, we set $d = 0.5\lambda$ and make sure that the estimated AOA is in $[0, \pi]$ due to the function $\arccos()$. Because the measured phase difference is relatively more stable, the estimated AOA is also stable, which can thus be leveraged for precise indoor localization.

III. The CiFi System

A. CiFi System Architecture

Figure 1 shows 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. Using the packet injection technique, the transmitter and receiver are set to the injection and monitor modes, respectively. The 5GHz band is used for better channel stability. The Intel 5300 NIC provides readings from 90 subcarriers at the three antennas. Then, we compute two sets of CSI data, including 30 phase differences from antennas 1 and 2, and 30 phase differences from 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 from the 960 packets. Each image consists of 60 packets (x-axis) and the corresponding 60 estimated AOA values for each packet (y-axis). The constructed image will then be used to train the DCNN.

CiFi exploits the constructed images for two reasons. First, the estimated AOA values are highly stable and robust for each given location. When the Wi-Fi signal is blocked by a wall or chair, the CSI amplitudes will be strongly weakened, which influence the localization accuracy. However, the estimated AOA values are more robust if the transmission direction is not changed. Second, the constructed image can leverage all subcarrier information from all received packets, which contains rich time and frequency features of the CSI data.

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 the DCNN.

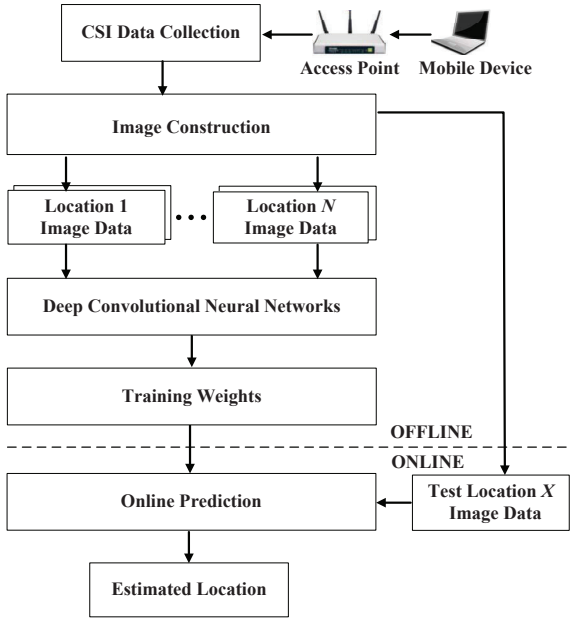

Fig. 1. The CiFi system architecture.

This method is quite different from traditional fingerprinting based methods, where a database is established for every training location, and either the measured raw data or leant features are stored as fingerprints. However, *our CiFi system only trains one group of weights for all the training locations*, which is analogous to a classification or regression problem in machine learning. This 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.

B. Offline Training

The DCNN incorporates several convolutional and subsampling layers as well as one or more fully connected layers. It can exploit local correlations by sharing the same weights between neurons of adjacent layers, thus reducing the training time. DCNN can also obtain the local dependency and scale invariant feature from input data. More important, it can extract more abstract representation of the input image data from the lower layers to the higher layers in the hierarchical architecture of DCNN, which can strengthen the feature extraction of CSI AOA data for indoor localization. We introduce the 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 of the DCNN, which is defined as

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

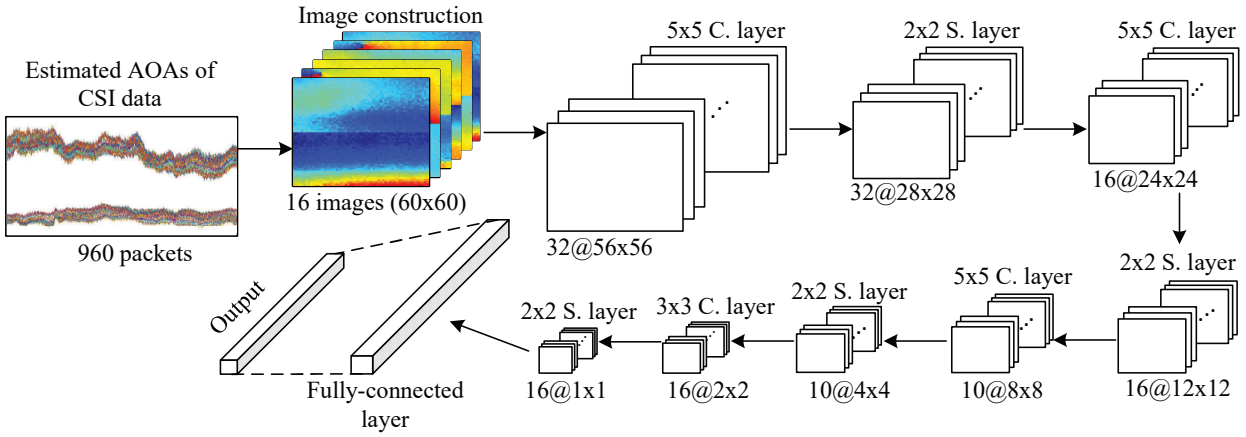

Fig. 2. CSI data training based on deep convolutional neural networks (C. and S. are short for convolutional and subsampling, respectively).

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, S_{l-1} is the set of feature maps in layer $(l-1)$ connected 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 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 on the inputs. The feature maps in the previous layer are pooled over a local temporal neighborhood by the mean pooling function, as

$$\theta_{ij}^{l+1} = \frac{1}{|G_j^l|} \sum_{k \in G_j^l} \theta_{ik}^l, \tag{7}$$

where G_j^l is the set of pooling region for the jth value in the ith feature map in layer l (identical for all i), $|G_j^l|$ is the number of elements in set G_j^l, θ_{ik}^l is the kth value of the ith feature map in layer l. Other methods such as the sum or max pooling function can be also used in this stage for reducing the training time.

For the fully-connected layer, we utilize a basic neural network with one hidden layer to train the output data after all the convolutional and subsampling layers. Moreover, the loss function is employed to measure the difference between the true location label and the output data of DCNN. By minimizing the values of the loss function with the BP algorithm, we can update the convolutional weights with the stochastic gradient descent method. In the proposed DCNN, we use the squared error loss function for training these parameters, which is defined as

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

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

Fig. 2 illustrates CSI data training for the DCNN. To obtain the input AOA images, we first estimate AOA values from 960 received packets as in (5). Then, we construct 16 images with size 60×60 out of the 960 AOA values. The images are convenient for 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, which can extract different characteristics. 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 three more convolutional and subsampling layers as in Fig. 2, we obtain 16 feature maps with size 1×1, which can be fully-connected in the next layer. Finally, we can obtain the forward output results and then combine the label of training data, which can be used to update the training weights such as the convolutional filters based on the loss function with the BP algorithm.

C. Online Algorithm

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

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

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 for the mobile device. We

first select location indexes of the R largest outputs from the DCNN for every column of matrix $\boldsymbol{O}$, thus producing a new matrix $\boldsymbol{S}$ with size $R \times M$ as

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

where s_{ij} is the location index of the ith largest output for the jth image. Every element of matrix 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 ith location index can be computed by averaging all the selected outputs for the ith location index, which is denoted as p_i.

Finally, the position of the mobile device $\hat{L}$ can be estimated as a weighted average of the R selected locations, as

$$\hat{L} = \sum_{i=1}^{R} l_i \times \frac{p_i}{\sum_{j=1}^{R} p_j}, \quad (11)$$

where l_i is the ith training location. In our experiments, we set $R = 2$ for better localization performance.

IV. Experimental Study

A. Experiment Configuration

We implement CiFi with 5GHz commodity Wi-Fi devices and carry out extensive experiments to valid its performance. In particular, we utilize a 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 5.58 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 5GHz CSI data at the receiver.

We compare CiFi with three representative approaches, including DeepFi [9], FIFS [7], and Horus [3]. To guarantee a fair comparison, we implement these three methods with the same CSI dataset in the 5GHz 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 Broun Hall in the Auburn University campus. The indoor space is a cluttered environment with many desktop computers, chairs, and metal tables, which block most of the LOS paths. The floor plan is shown in Fig. 3. 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.

Corridor: This is a long corridor in Broun Hall with dimension 2.4×24 m^2. As in Fig. 4, we place the access point at one end on the floor to measure 5GHz CSI data. There are main LOS paths 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.

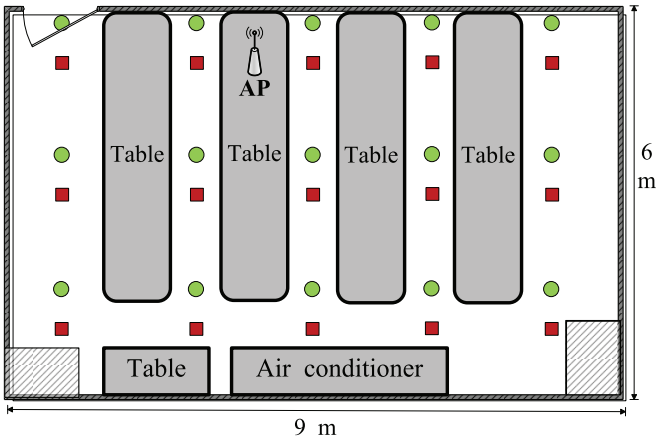

Fig. 3. Layout of the computer laboratory: training locations are marked as red squares and testing locations are marked as green dots.

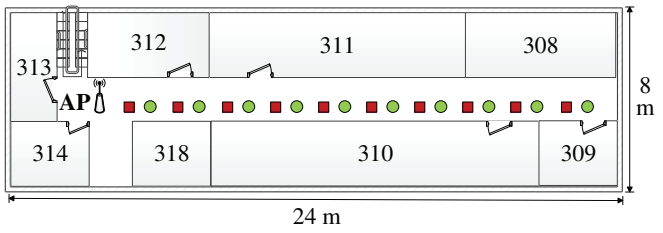

Fig. 4. Layout of the corridor: training locations are marked as red squares and testing locations are marked as green dots.

B. Accuracy of Location Estimation

Figure 5 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 for input AOA images. Moreover, the iterations indicate the times of training input AOA images with the DCNN. For the laboratory experiments, the training error curve starts to converge after 1.48×10^4 iterations, which finally reaches the preset threshold with about 0.06 training error after 4.85×10^4 iterations. For the corridor experiments, 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.

Figure 6 presents the CDF of distance errors of four schemes in the computer laboratory case. For this environment with the complex multipaths, CiFi can utilize the unique multiple path feature for location estimation, which is different for different locations. CiFi has 40% of the test locations having an error less than or equal to 1 m, while that for the other schemes is 30%. We also find that about 87% 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% for

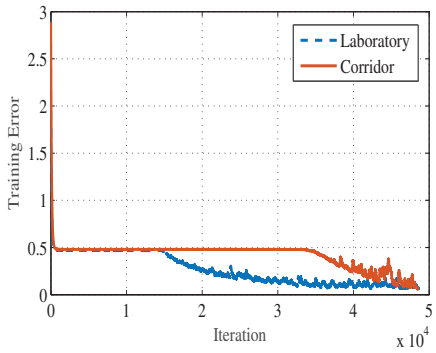

Fig. 5. Training errors over iterations for the laboratory and corridor experiments.

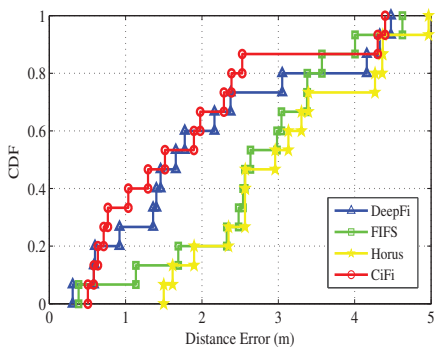

Fig. 6. CDF of localization errors for the laboratory experiment.

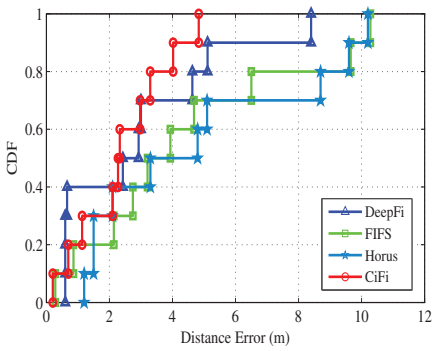

Fig. 7. CDF of localization errors for the corridor experiment.

DeepFi, FIFS, 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.

Figure 7 presents the CDF of localization errors of all the scheme 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% of the test locations for CiFi and DeepFi have an error under 3 m, while it is 40% for FIFS and Horus. This result shows that the CiFi system achieves a close localization performance to that of DeepFi, while both outperform the other two schemes, in the challenging corridor environment. However, different from DeepFi, the proposed CiFi system does not require to build up a database for every training location, thus greatly reducing the storage requirement.

V. Conclusions

In this paper, we proposed CiFi, a DCNN based system for indoor localization with 5GHz Wi-Fi. We first analytically and experimentally verified the feasibility of using AOA values for indoor localization. We then presented the CiFi system design, which first forms AOA images to train the DCNN, and then uses newly received AOA images to estimate the location of a mobile device. Through extensive experiments, we demonstrated the superior performance of the proposed CiFi system over existing schemes.

Acknowledgment

This work is supported in part by the U.S. NSF under Grant CNS-1702957, and the Wireless Engineering Research and Education Center (WEREC) at Auburn University. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the foundation.

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] X. Wang, L. Gao, S. Mao, and S. Pandey, "DeepFi: Deep learning for indoor fingerprinting using channel state information," in *Proc. WCNC'15*, New Orleans, LA, Mar. 2015, pp. 1666–1671.

[9] ——, "CSI-based fingerprinting for indoor localization: A deep learning approach," *IEEE Trans Veh. Technol.*, vol. 66, no. 1, pp. 763–776, Jan. 2017.

[10] X. Wang, L. Gao, and S. Mao, "PhaseFi: Phase fingerprinting for indoor localization with a deep learning approach," in *Proc. GLOBECOM'15*, San Diego, CA, Dec. 2015, pp. 1–6.

[11] ——, "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.

[12] 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.

[13] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, pp. 436–444, May 2015.

[14] 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.

[15] F. J. Ordonez and D. Roggen, "Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition," *MDPI Sensors*, p. 115, Jan. 2016.

[16] 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.

[17] Y. Xu, G. Yue, and S. Mao, "User grouping for massive MIMO in FDD systems: New design methods and analysis," *IEEE Access Journal*, vol. 2, no. 1, pp. 947–959, Sept. 2014.

[18] 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.

[19] X. Wang, C. Yang, and S. Mao, "Tensorbeat: Tensor decomposition for monitoring multi-person breathing beats with commodity WiFi," *ACM Transactions on Intelligent Systems and Technology*, 2017, in press.