EyeFi: Fast Human Identification Through Vision and WiFi-based Trajectory Matching

Shiwei Fang
UNC Chapel Hill
shiwei@cs.unc.edu

Tamzeed Islam
UNC Chapel Hill
tamzeed@cs.unc.edu

Sirajum Munir
Bosch Research and Technology Center
sirajum.munir@us.bosch.com

Shahriar Nirjon
UNC Chapel Hill
nirjon@cs.unc.edu

***Abstract*—Human sensing, motion trajectory estimation, and identification are central to a wide range of applications in many domains such as retail stores, surveillance, public safety, public address, smart homes and cities, and access control. Existing solutions either require facial recognition or installation and maintenance of multiple units, or they lack long-term re-identification capability. In this paper, we propose a novel system – called EyeFi– that combines WiFi and camera on a standalone device to overcome these limitations. EyeFi integrates a WiFi chipset to an overhead camera and fuses motion trajectories obtained from both vision and RF modalities to identify individuals. In order to do that, EyeFi uses a student-teacher model to train a neural network to estimate the Angle of Arrival (AoA) of WiFi packets from the CSI values. Based on extensive evaluation using real-world data, we observe that EyeFi improves WiFi CSI based AoA estimation accuracy by more than 30% and offers 3,800 times computational speed over the state-of-the-art solution. In a real-world environment, EyeFi's accuracy of person identification averages 75% when the number of people varies from 2 to 10.**

I. Introduction

Human sensing, motion trajectory estimation, and identification have a wide range of applications, including in retail stores, surveillance, public safety, public address, and in access control. For example, in retail stores, it is useful to capture customer behavior to determine an optimal layout for product placement, detecting re-appearing shoppers after weeks to track shopper retention rate, separating employees from shoppers to generate an accurate heatmap of motion pattern of (only) shoppers. For surveillance, it is useful to identify and track a limited set of people from a crowd, e.g., tracking undercover police agents from a group of people to ensure their safety. Once a class of people is identified, that can be leveraged for public address based on additional contexts. For example, when an active shooter in a building has been identified through a security camera, targeted and customized messages can be sent to different groups of people in different parts of the building through accurate identification to help to find safe escape routes – instead of sending a generic SMS to everyone, possibly including the shooter.

A wide variety of sensing technologies exist for human sensing, motion trajectory estimation, and identification that uses cameras, WiFi, Bluetooth, and ultrasonic sensors. However, there are shortcomings of each of these sensing modalities. For example, ultrasonic sensor-based identification [10] does not scale to thousands of reappearing shoppers in retail stores. Camera-based solutions suffer from illumination, occlusion, background cluttering, and change of perspective and fail to support long term identification, e.g., detecting a shopper after two weeks when they show up in a different colored dress in a retail store when body appearances based identification is applied [9]. Facial recognition can be potentially used for person identification at scale. However, facial recognition is banned in many places, e.g., San Francisco [2], and it is difficult to employ in some settings such as in retail stores where typically panoramic cameras mounted on ceilings can hardly see faces. Sniffing WiFi MAC addresses provide coarse-grained location information, e.g., a shopper is within 30 meter radius of a WiFi access point without providing location insights to infer customer-product interaction. To achieve precise localization using WiFi, multiple WiFi beacons or receiving units need to be set up, maintained, and coordinated, which can be very expensive [12].

In this paper, we propose to fuse two powerful sensing modalities – WiFi and camera – in order to overcome the aforementioned limitations of the state-of-the-art solutions. We call our proposed solution EyeFi. EyeFi does not require facial recognition, provides long-term re-identification, does not require deployment and maintenance of multiple WiFi units, and has the potential to provide such intelligent capabilities on a standalone device. To this end, EyeFi integrates a WiFi chipset (with multiple antennas) to a camera. As a result, a single EyeFi unit can detect, track, and re-identify people as far as the camera can see. Our current implementation of EyeFi uses a panoramic camera mounted on a ceiling. However, other types of cameras such as a bullet camera will also work.

EyeFi uses the on-board camera to detect, track, and estimate the motion trajectories of the people in its field of view. Simultaneously, using the on-board WiFi chipset, EyeFi overhears WiFi packets from nearby smartphones and extracts the Channel State Information (CSI) data from the WiFi packets. The CSI information is used to estimate the Angle of Arrival (AoA) of the smartphone from the EyeFi unit. Compared to existing WiFi-based AoA estimation techniques [12], EyeFi uses a smartphone in motion as a transmitter (not a stationary desktop computer), uses a low sampling rate (around 23 packets per second), and uses a novel teacher-student based visually guided neural network to speed up the AoA estimation by over 3,800 times. For each person (i.e., smartphone) generating the WiFi traffic, a sequence of AoAs is estimated to capture the motion trajectory of the individual. Then EyeFi performs cross-modal trajectory matching to determine the identity of the individuals. It is based on the assumption that most people use smartphones and the same smartphone is usually used for an extended period of time. Also, as smart watches are

becoming popular and getting equipped with WiFi chipsets (e.g., Samsung Gear S3 and Apple Watch 4), EyeFi can leverage wireless devices beyond smartphones. Note that the MAC addresses can be hashed to safeguard the privacy of the individuals, but it is still useful for long-term re-identification and behavior analysis.

This work has the following contributions:

• First, we design and implement a novel multi-modal sensing system called EyeFi, which is the first system that fuses WiFi CSI with camera for human sensing, motion trajectory estimation, and long-term identification and has the potential to offer such analytics on a standalone device. EyeFi overcomes several limitations of the state-of-the-art solutions as it does not require the use of facial recognition and the cost of deployment and installation of multiple WiFi units.

• Second, since no such system and datasets are available, we collect over 74 GB of data containing videos and WiFi CSI values of over one million WiFi packets with over 15 volunteers from two different environments to develop and test our solution. We annotate a major portion of the dataset[1].

• Third, we develop a novel student-teacher based neural network to estimate AoA from CSI values. Instead of just using camera-based motion trajectory as the ground truth, we force the network to regress the AoA of state-of-the-art SpotFi algorithm [12], and thereby, forcing the network to learn multipaths and estimate AoA more accurately. We also propose novel techniques to smooth the WiFi-based trajectory for cross-modal matching.

• Finally, based on extensive evaluation using real-world data, we find that EyeFi improves WiFi CSI-based AoA estimation accuracy by more than 30% and offers 3,800 times computational speed up over the state-of-the-art solution, enabling EyeFi a real-time system. We observe that the average accuracy of EyeFi for person identification is 75% when the number of people varies from 2 to 10.

II. Usage scenarios

We describe two real-world usage scenarios of EyeFi.

• *Customer Behavior Analysis.* Once a customer arrives at a store, his smartphone generates WiFi traffic to discover local access points. After he connects to the local WiFi access point, his checking of notifications, viewing of websites for better prices or deals of similar items, listening of SpotiFi music, or messaging of friends generates more WiFi traffic. All of these WiFi traffic is overheard by the WiFi chipset of EyeFi system. EyeFi extracts the MAC address and CSI values from the WiFi packets, timestamps each value, and records them. Using our proposed algorithm, EyeFi performs AoA estimation and matches the AoA sequence with one of the trajectories observed from the camera. Due to the use of the MAC address, the customer can be identified over a long period and even at a different store. EyeFi hashes the MAC address to anonymize the customers, but can still generate high-level analytics of aggregated customer behavior.

[1]More information on EyeFi data and deployment can be found at the project page https://github.com/munir01/EyeFi

• *Emergency Situation.* During emergency situations, EyeFi can send location and person-specific targeted messages to guide people to safety. For example, in a retail environment, EyeFi can send different messages to employees who know the store area, law enforcement officers that are armed, and customers who need help. In case of an emergency, such as the presence of an active shooter, the law enforcement officers can be notified of the shooter's exact location (determined using cameras) so that they can take proper actions, employees can be instructed to assist customers and to commence emergency protocol, and customers can be given specific instructions on their phones based on their location (e.g., the nearest and safe escape route or a safe hiding place).

III. Background

A. WiFi AoA Estimation

In a WiFi network, a transmitter and a receiver communicate by sending packets back and forth through a certain band. The communication band contains multiple channels in which each channel is a certain range of frequencies. During the transmission, the *Channel State Information* (CSI), which describes the properties of the channel, is being recorded by the receiver. The properties of such channels are a description of the combined effect of the scattering, fading, and power decay between the transmitter and receiver. The WiFi chipset natively estimates the *Channel State Information* (CSI) to improve communication efficiency. With recent studies, researchers have found that CSI can be used for WiFi sensing, including but not limited to angle of arrival estimation, human detection, and breath detection.

For the angle of arrival (AoA) estimation, one classical method is the MUSIC algorithm [19]. If two antennas are separated d distance apart, the additional phase shift introduced due to the distance is $-2\pi \times d \times \sin(\theta) \times f/c$, where θ is the AoA, f is the signal frequency, and c is the speed of light. The MUSIC algorithm works by estimating the steering matrix A in $X = AF$, where X is the measurement matrix of the received signal, and F is the matrix of complex attenuation. In a recent WiFi-based localization algorithm [12], the AoA of the direct path (which is relevant to the localization problem) is isolated by taking the eigenvector of the matrix, XX^H, for which, the eigenvalue is zero. The eigenvector goes through further processing to obtain the direct path.

IV. System Design

A. Overview

EyeFi is a framework that fuses information from visual domain captured from camera and CSI measurements of WiFi packets to jointly track human motion trajectories and identify them for many applications. Using on board computer vision algorithm, the camera detects people, estimates their location and motion trajectories, but unable to identify and re-identify people across time and/or multiple cameras without a shared field of view across cameras. However, WiFi provides a way of identification through user-specific information, i.e., the MAC address of the user's smartphone, but the derived motion trajectory is coarse grained and inaccurate. EyeFi exploits

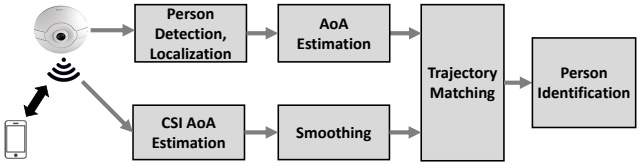


Fig. 1: An overview of EyeFi system architecture.

the properties of these two sensing modalities to fuse the trajectories obtained from both for fast and accurate person identification across time and space. The system primarily consists of a camera and a WiFi sniffer. EyeFi does not require installation of any apps or beacons on the users' smartphone and does not add additional overhead to the phone.

A high-level architecture of EyeFi is shown in Figure 1. A surveillance camera with a WiFi chipset is installed at the desired location and it overhears WiFi packets of intended subjects like shoppers. Smartphones generate WiFi packets after connecting to the local WiFi access point. When a smartphone is not connected to an access point, it still generates WiFi packets to discover nearby access points. These packets are captured by EyeFi along with CSI information, which is used to estimate AoA of the WiFi source. However, since the estimated AoAs are very noisy, they are further processed to smooth the motion trajectory. Meanwhile, the camera reports the locations of detected human subjects (which may contain more/less people than the number of smartphones that the WiFi unit has detected) and their motion trajectories. Both the trajectories from the camera and the WiFi are sent to the trajectory matching module that identifies people using WiFi MAC addresses by performing cross modal trajectory matching.

B. Motivational Experiments

EyeFi is motivated by the poor performance of existing person identification solutions. For example, a possible alternative to EyeFi is to use a camera-based solution that uses facial recognition to track people across time and locations. However, cameras installed in public places like a retail store can barely see the faces of the customers. In order to understand the performance of a camera-based system, we apply a facial recognition algorithm on a video feed that we collected during our empirical study.

Figure 2 shows a frame from our video feed which contains eight human subjects highlighted using red rectangular boxes. We use a Python-based facial recognition software [1] to detect faces in this frame. The result is catastrophic. The software detects 0 faces after running the algorithm on the entire video. This is because as we can see in the example video frame, a human subject can be facing away from the camera and his dress and floor color can be very similar – which poses an additional challenges to object recognition and matching in a purely vision based domain. Also, existing pre-trained vision-based models do not work well with panoramic images.

To complement the vision-based system, one can add WiFi-based localization to the system by running the SpotFi [12] algorithm on the collected CSI data. However, based on our experiments, the Matlab implementation [3] of the SpotFi

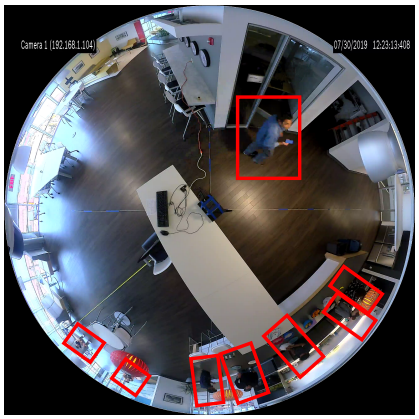


Fig. 2: Facial recognition software can not recognize any of the 8 human subjects present in the view. All figures best viewed in color.

algorithm (provided by the authors) requires around 1.5 - 2 seconds to generate AoA estimation for a single WiFi packet. For 8 hours of continuous WiFi stream at the data rate of 20 packets per second, the total number of WiFi packets is $576,000$. With 1.5 seconds computation time for each packet, the AoA estimation requires 240 hours. Even though the computation time can be reduced by using a more efficient implementation, the computation time will still be too long to be viable for processing a large number of WiFi data points for the intended application. Based on these initial experiments, we develop EyeFi to overcome these limitations and to achieve a faster, accurate, and practical solution that works in real-life scenarios.

V. Algorithm

EyeFi is a modular system that combines information from visual domain with RF domain. For camera based person detection, tracking, and trajectory estimation, EyeFi uses the proprietary software that comes with Bosch Flexidome IP Panoramic 7000 camera. Our evaluation shows that the existing firmware of the camera can estimate AoA of individuals with an average of 1.03 degree error. EyeFi is agnostic of underlying computer vision technique of person detection and tracking as long as the accuracy is similar. So, we focus on WiFi based AoA estimation, trajectory smoothing, and cross modal trajectory matching. However, we use the camera based location information to improve accuracy and execution speed of WiFi based AoA estimation. In this section, we describe each of these components in detail.

A. Camera Assisted WiFi Based AoA Estimation

WiFi communication produces Channel State Information (CSI) which can be used to estimate the Angle of Arrival (AoA) of the incoming WiFi signals. Methods like SpotFi [12] extend computationally expensive MUSIC algorithm which uses linear algebra to decompose and estimate AoA. However, from our experiments, the SpotFi algorithm is evidently slow and does not work well in large AoA scenarios (e.g., 60°-90°). Examples are given in Section VII-B.

To reduce the computation time and to improve overall AoA estimation performance in order to be viable for EyeFi,

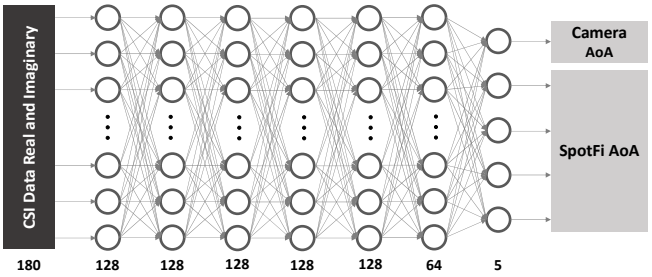


Fig. 3: Neural network model used in the AoA estimation. Training data includes the ground truth AoA from camera and SpotFi generated AoA data. There are 6 hidden layers and the number of neurons for each layer is listed underneath.

we seek a data-driven machine learning-based approach to estimate AoA from CSI data. To that end, we try different neural network architectures and after observing similar performance of multiple complex networks, we choose a fully connected neural network that takes CSI data as input and regresses the AoA values (shown in Figure 3) as it performs equally well. For the CSI data, there are 90 complex numbers for each packet, which correspond to 30 subcarriers from 3 antennas. We format these 90 complex numbers into a vector of 180 numbers which represent the real and imaginary parts. The neural network has 6 hidden layers in addition to the input and output layer. For the activation function, we use Leaky ReLU [15] for improved performance [29] and to accommodate the negative value of the imaginary part:

$$y = \begin{cases} x, & \text{if } x \geq 0 \\ \text{slope} \times x, & \text{otherwise} \end{cases} \tag{1}$$

We also use dropout [23] to reduce overfitting. We use L1 loss function and Adam optimizer for training.

For training the neural network, we take inspiration from knowledge distillation, more specifically, the teacher-student model where the student network learns from the soft labels of a teacher model [6]. We treat the SpotFi [12] algorithm as the teacher model for the training purpose. Even though the goal of our neural network is to regress AoA using CSI and we provide hard label of AoA from camera as ground truth, forcing the network to regress the AoAs of the SpotFi, which are the soft labels in our case, in addition helps to ensure the network is forced to learn multipaths as accurately as the teacher model (SpotFi) and hence has a better chance of generalizing to a different environment. As shown in Figure 3, the output of the network is a vector of size five as the network regresses to one AoA from camera and four multipaths from SpotFi. We test our hypothesis regarding the need for a teacher-student model and find that such a model helps to improve AoA estimation accuracy and helps with generalization in a different environment as described in Section VII-B.

B. WiFi based Trajectory Smoothing

The estimated AoA from CSI data is noisy – which we can see from Figure 4, where both the estimations from SpotFi and our neural network generated ones show similar characteristics. Such noisy characteristic can be caused by sensor measurement noises, body-shadows, and multi-path effects. Even in a controlled environment, the noisy situation improves but is not eliminated. To address this issue, we smooth the data to better align with the ground truth.

Typically smoothing is based on the idea that noise has a certain distribution and added to the underlying real data. For such data, employing moving average usually achieves a good result. However, the data from the AoA estimation does not show such a distribution. Applying moving average in our time series data causes the result to bias towards the noisy direction. An example of applying moving average algorithm on the data from Figure 4(b) is shown in Figure 5(a). We can see that the smoothed results are biased toward the noisy direction.

For non-parametric based methods such as Locally Weighted Scatterplot Smoothing (LOWESS) can produce better results in some cases as it does not assume the data fits some specific distribution. However, in our case, the AoA data can become extremely noisy such as around packet number 200 - 250. These abrupt fluctuations can severely distort the smoothed result as shown in Figure 5(c) after applying LOWESS. Even though it improves over the moving average approach, it is still too noisy to match with a camera based trajectory.

To achieve the best possible smoothing, we develop a two-stage smoothing pipeline. The first stage addresses the noise causing abrupt fluctuations. We define a smoothing window of length N with the targeted smoothing data point $x_t \in X$ at the middle point, where X is a set of all the N data points within the window. The variance of all the data points in the smoothing window is calculated:

$$\sigma_X^2 = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_X)^2 \tag{2}$$

we then calculate the variance without the targeted data point:

$$\sigma_{\hat{X}}^2 = \frac{1}{N}\sum_{i=1, i\neq t}^{N}(x_i - \mu_{\hat{X}})^2 \tag{3}$$

If the difference between the two variances $\delta = \sigma_X^2 - \sigma_{\hat{X}}^2$ is larger than a threshold λ (we set to 1), we treat the target data point as an abrupt noise and replace it with:

$$x_t = \begin{cases} \mu_{X_{biased}}, & \text{if } \sigma_X^2 > \lambda_{var} \\ \mu_X, & \text{otherwise} \end{cases} \tag{4}$$

The σ_X^2 and λ_{var} are variance of the smoothing window and a threshold (set to 300) to determine if this smoothing window is a volatile region. Our empirical study shows that as the neural network learns the multipaths of SpotFi, the estimated AoAs have larger variances when the phone is within larger AoA ranges. Example of such areas can be seen in Figure 4(b) packet 150-225 and 325-375. As they are in the large AoA range, the noise can be large and toward the opposite direction. As a result, we use $\mu_{X_{biased}}$ to replace the targeted data point when $\sigma_X^2 > \lambda_{var}$ is met. X_{biased} is determined as follows:

$$X_{biased} := \begin{cases} \{x \in X : x \geq X_{median}\}, & X_{median} \geq 0 \\ \{x \in X : x < X_{median}\}, & X_{median} < 0 \end{cases} \tag{5}$$

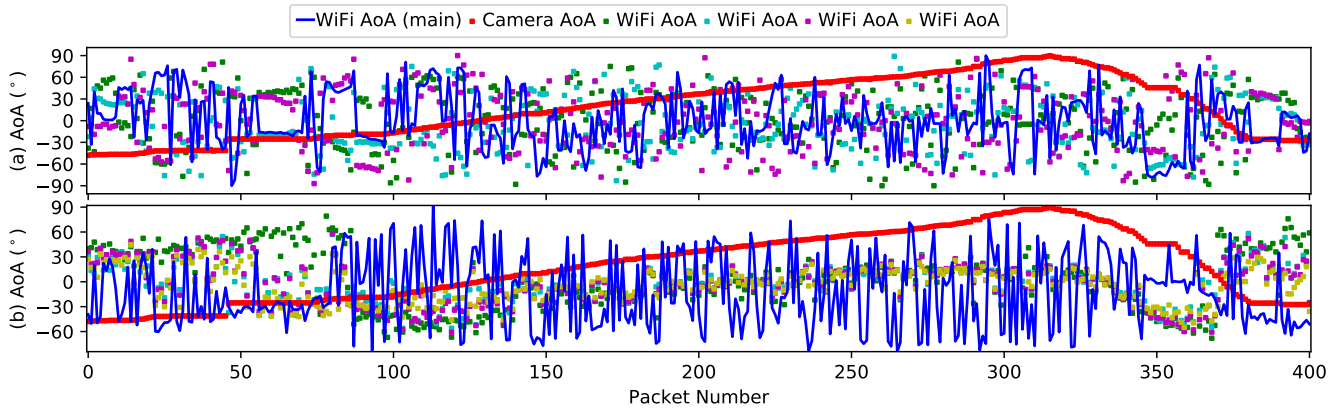


Fig. 4: Both outputs from (a) SpotFi and (b) neural network exhibit noisy characteristics. Red represents the ground truth from camera and blue represents the first multipath from SpotFi (in (a)) and neural network (in (b)) generated AoA, respectively.

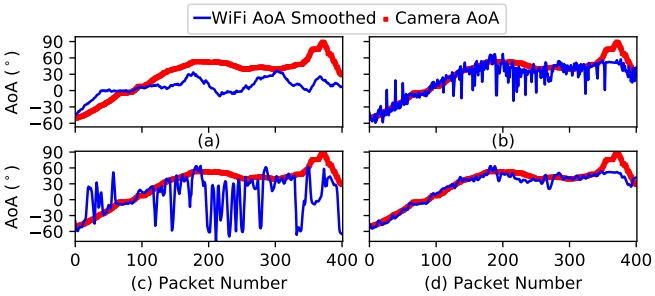


Fig. 5: Comparison between different smoothing techniques. (a) is smoothed with moving average, (b) is smoothed with our variance based initial smoothing, (c) is AoA data smoothed with LOWESS, and (d) is after applying LOWESS on smoothed data from (b).

The X_{median} is the median value of X (considering all N AoA values) in that smoothing window. Expression $\{x \in X : x \geq X_{median}\}$ means elements in X that are larger than their median value. $\mu_{X_{biased}}$ is the average of X_{biased} from Equation 5. For the second case of Equation 4, μ_X is the sample mean of all the AoA data points in the smoothing window. Figure 5(b) shows the result of the variance-based smoothing performance over the first AoA (i.e., the first element of the output vector) generated by the neural network. As we can see from the figure, the noises causing abrupt fluctuations are largely addressed. For the second stage of smoothing, we apply LOWESS to the smoothed data and the results are shown in Figure 5(d). The mean and median values of the absolute differences between the smoothed AoA data and camera generated ground truth AoA of all the figures in Figure 5 are shown in Table I. It shows that applying LOWESS after variance based smoothing significantly reduces the mean and median error of AoA estimation. We perform a detailed evaluation in Section VII-C.

	Moving Average	Variance based	LOWESS	Variance based + LOWESS
Mean (°)	19.45	12.39	22.77	10.14
Median (°)	15.47	8.33	10.93	6.40

TABLE I: Mean and Median error of AoA estimation after applying different smoothing algorithms on the example data shown in Figure 5

C. Identification Through Trajectory Matching

To identify individuals, EyeFi performs a cross-modal trajectory matching. The simplest approach is to apply Euclidean distance between two trajectories to find the shortest distance,

$$d_{j,k} = \sqrt{\sum_{i=1}^{N}(T_{i,j} - T_{i,k})^2} \tag{6}$$

where $T_{i,j}$ and $T_{i,k}$ are the AoA trajectories for subject j (computed using camera) and k (computed using WiFi CSI), respectively from a window of size N. By ranking the $d_{j,k}$ we can find the matched one with the shortest distance.

If each subject walks differently, Euclidean distance could be enough to identify subjects as long as the AoA estimation from each sensing modality is accurate. However, to identify subjects that have a similar path, we also take into consideration of the rate of the change in AoA trajectories. Specifically, we use polynomial functions to represent the trajectories and match with the desired subject. For a matching window of size N, we have can represent the segment with:

$$y = a \cdot x^3 + b \cdot x^2 + c \cdot x + d \tag{7}$$

where $x \in X[1, N]$. The polynomial fitting problem is also a smoothing operation, where small noises are filtered out. The estimated y data points are used for calculating the Euclidean distance between the trajectories from WiFi and camera after fitting polynomials. Also, instead of using the standard Euclidean distance function, we apply weights to each AoA data point. Using the polynomial function, we can find the rate of change at every data point R_i. Then, we calculate the absolute rate of change of differences $\hat{R}_i = |R_{i,j} - R_{i,k}|$ for each pair of trajectories. Then the weighted distance is calculated as follows.

$$\hat{d_{j,k}} = \sqrt{\sum_{i=1}^{N}(T_{i,j} - T_{i,k})^2 \otimes \hat{R}_i} \tag{8}$$

Here, the operator $\otimes$ represents element-wise multiplication. We only apply weighted Euclidean distance to smaller windows (less than 200 packets) as using it for larger matching windows will smooth out sharp trajectory changes that will deteriorate the performance of the matching. For larger

matching windows, the standard Euclidean distance is used as it provides good performance and less computation overhead.

VI. Data Collection

A. Hardware and Software Setup

In this work, we use Bosch Flexidome IP Panoramic 7000 camera to collect vision data and Intel 5300 WiFi Network Interface Card (NIC) installed in an Intel NUC to collect WiFi data. The camera is mounted on a ceiling at a height of 2.85 meters whereas the WiFi card is located at the same location as the camera but at a height of 1.12 meters forming a unified coordinate system. We collect data in two different locations as shown in Figure 6. Figure 6(a) shows the lab area where the majority of the data are collected, and Figure 6(b) shows the Kitchen area where the collected data are used for testing only (not used for training). The lab area is rectangular having dimensions of 11.8m x 8.74m and the kitchen area is irregularly shaped with maximum distances of 19.74m and 14.24m between two walls. The kitchen also has numerous obstacles and different materials that pose different RF reflection characteristics. The change in the environment creates a vastly different RF characteristic that is used to test the robustness and generalizability of the system.

To collect WiFi data, we set up a Google Pixel 2 XL smartphone as an access point and connect the Intel 5300 NIC to it for WiFi communication (both are shown in Figure 7). The phone transmits 20-25 packets per second to the NUC. Such a low packet transmission rate simulates realistic scenarios, e.g., apps in the phone are receiving notifications. We use Linux CSI Tool [8] to record the CSI information from the Intel 5300 WiFi NIC on Intel NUC.

B. Collected Dataset

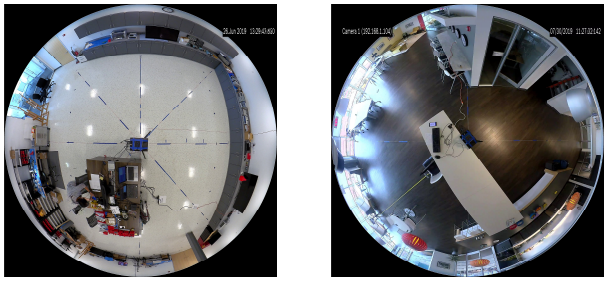
We collect data over multiple days and vary the number of people present in the scene. In the end, we have transmitted over 1.2 million WiFi packets and collected corresponding CSI values of over 13 hours. We also have over 15 different individuals holding the phone to capture various ways people hold phones, their walking patterns, and different heights. In addition to a single person walking in the scene, we also have multiple people walking simultaneously.

VII. Evaluation

A. Evaluation Setup

We evaluate how each component of our system performs. We divide our dataset into different sets for this purpose. Part of the data collected from the lab area is used for our training and algorithm development. Data collected from the kitchen area are only used for testing the robustness and generalizability of the system. To identify the subject with the phone and test the accuracy of identification, we use the data from our camera system as ground truth. The camera detects individuals and provides the (x, y) coordinates of each of the detected subject, which is used to estimate AoA.

We first evaluate the accuracy of the camera in terms of its ability to estimate (x,y) coordinates and AoA by standing in



(a) Lab area. (b) Kitchen area.

Fig. 6: Data collection environment seen from panoramic cameras. (a) lab area which is large and rectangular shaped, (b) kitchen area which is irregular and has many obstacles. An Intel NUC is located at the middle of each figure forming a unified co-ordinate system of the camera and NUC.

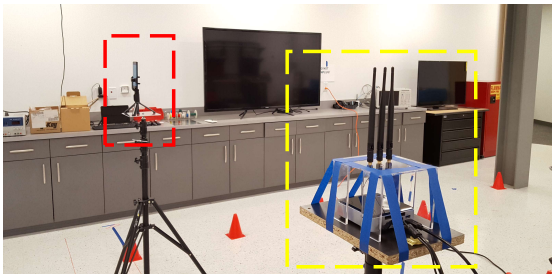


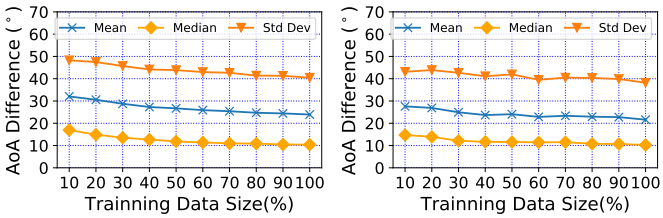
Fig. 7: Data collection equipment. On the right (yellow box) is the Intel NUC with Intel 5300 WiFi card installed with three external antennas. On the left (red box) is the Google Pixel 2 XL phone for communication. Note that all the data are collected while a subject is holding the phone in his/her hand.

16 different locations throughout the lab and comparing the differences between the camera and our actual measurements. We find that the average error of the camera is (0.32, 0.29) meters for estimating (x,y) coordinates and 1.03 degree for AoA. It shows that we can use the location data from the camera system as ground truth for the training and testing.

B. Neural Network Based AoA Prediction

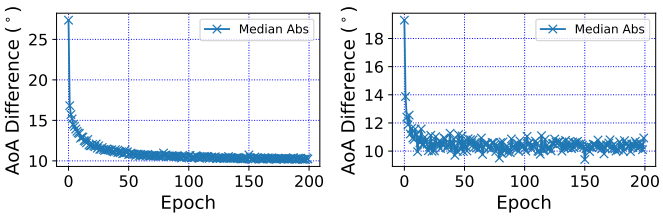
1) Training Data: As discussed in Section V-A, our training data consists of SpotFi generated AoAs and camera generated ground truth AoA. To prepare the dataset, we need to address the phase offsets between the 3 RF chains in the Intel 5300 NIC. [31] states that the phase offsets between these chains are deterministic and the offset between two RF chains only poses two possible values. We determine the two values based on our own measurements using methods stated in [31].

During the data collection, it is impractical to measure the phase offsets each time the system reboots. To address this issue, we apply all possible (four) combinations of the phase offset when calculating AoAs using the SpotFi algorithm. Once all the SpotFi AoA data are generated, we find the correct phase offset by choosing the one with the smallest mean absolution difference between the AoA from SpotFi and AoA from the camera. Then, we use the SpotFi data with correct phase offsets calibrated to train our neural network models.



(a) Result in Lab area. **(b) Result in Kitchen area.**

Fig. 8: Neural network performance for different size of training dataset.



(a) Result in Lab area. **(b) Result in Kitchen area.**

Fig. 9: Neural network performance during training.

	Lab		Kitchen	
	Mean (°)	Median (°)	Mean (°)	Median (°)
SpotFi	59.17	58.08	50.14	44.03
SpotFi Only NN	62.97	63.29	45.55	46.01
Camera only NN	31.87	14.57	36.61	20.63
SpotFi + Camera NN	**30.56**	**13.98**	**35.08**	**18.72**

TABLE II: Mean and Median on AoA estimation performance of different neural network models and SpotFi on data collected from lab and kitchen area.

2) Neural Network Models: For the AoA estimation, we train our neural network models with different outputs to evaluate the performance of the teacher-student model and whether the neural network can learn the SpotFi algorithm. We train three different neural network models: *SpotFi Only NN* that uses SpotFi generated AoA results for training, *SpotFi + Camera NN* that is our teacher-student model, and *Camera Only NN* that uses only the camera generated ground truth AoA for training. All the neural network models are trained with the same training dataset with roughly 400,000 WiFi-camera AoA pairs.

To evaluate how the number of training samples affect the neural network performance, we train our *SpotFi + Camera NN* teacher-student model with different size of the training dataset, and test the performance of the neural network on the validation dataset (roughly 58,000 WiFi-camera AoA pairs from the lab) and a subset of the data collected in the Kitchen area (roughly 22,000 WiFi-camera pairs). The results are shown in Figure 8. The X-axis is the percentage of the dataset (roughly 400,000 WiFi-camera AoA pairs) used to train the neural network and Y-axis is the absolute difference between the neural network prediction and ground truth AoA. We report mean, median and standard deviation of the difference here. In Figure 8(a), we see that the performance of the neural network on the validation dataset improves as the size of the training dataset increases. The same trend are also being observed with the Kitchen area dataset. The improvements become minimal after 80% of the dataset is being used for training. It shows that our training dataset is large enough to train our neural network model.

We also test if our epoch is large enough to complete the training of the network for improving the AoA estimation by calculating the mean and median AoA difference after each epoch with both the datasets from lab and the kitchen. The results are shown in Figure 9. As we can see, the performance improvements level out when the number of epochs is larger than 75. The similarity between Figure 9(a) and Figure 9(b) shows that our neural network model generalizes to a difference environment as it learns.

We test all three models on an unseen test dataset. Table II shows the mean and median of the absolute AoA difference between the predicted one and ground truth for the neural network models and SpotFi. From the table, we see similar performance between the *SpotFi* algorithm and the *SpotFi Only NN* that demonstrates that the neural network can learn the SpotFi algorithm. *Camera Only NN* and *SpotFi + Camera NN* both show improved performance over *SpotFi* and *SpotFi Only NN*, and our teacher-student model *SpotFi + Camera NN* shows the best performance. The neural network learns the underlying relationship between the CSI and AoA, and the teacher-student model further improves the generalizability of the neural network on different test cases and environments.

3) Robustness and Efficiency of AoA Estimation Neural Network: We test the robustness of our teacher-student neural network model on both unseen CSI data collected in the lab and kitchen area using over 30,000 and 20,000 WiFi packets, respectively. The results are shown in Table II. It shows that the performance of the neural network is comparable across different environments and shows better results than SpotFi. The difference between the SpotFi results in two environments can be because of different environment RF characteristics and percentage of the phone in different AoA range (SpotFi performs worse in large AoA range).

In addition to the performance improvement on the AoA estimation, the neural network also improves the execution speed. SpotFi takes around 1.5 seconds to estimate AoA per WiFi packet, thus making it difficult to be useful in a real-time solution. Based on using 22,854 WiFi packets collected from the kitchen, we see that our neural network is around 3809 times faster than SpotFi, which can be further improved using GPU computation and batch data, enabling EyeFi a real-time solution.

C. Smoothing

We use the kitchen data (different environment from the training) from the previous subsection to evaluate the performance of our smoothing algorithm. We measure the mean and median absolute errors between the smoothed AoA data and camera derived ground truth AoA data. The results are shown in Table III. In this table, *NN* is the AoA estimation from our neural network model, *Variance Based* is the results after our first stage smoothing, *Variance + LOWESS* is our full smoothing stack, and we also report the result by only applying the LOWESS algorithm in *LOWESS*. From this table, we can

see that smoothing improves on the original neural network generated AoA estimation. Our smoothing stack produces the best results with both lowest mean and median absolute errors.

	NN	Variance Based	Variance + LOWESS	LOWESS
Mean (°)	24.61	12.98	**10.80**	12.17
Median (°)	11.76	9.59	**7.91**	8.09

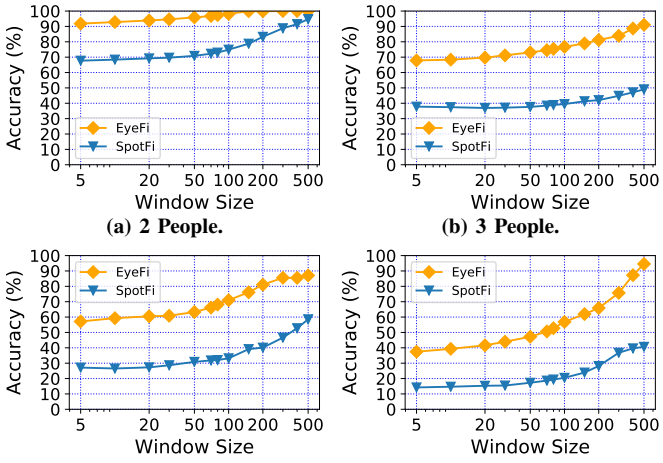
TABLE III: Smoothing performance with different smoothing techniques and combinations.

The results from LOWESS based smoothing is better than our variance based smoothing in Table III. This is different from the results presented in Table I which is the statistics for Figure 5. This is because in Figure 5, we choose a segment of the data where the neural network introduced more noise and the ground truth is in the large AoA range.

D. Identification

To evaluate the performance of identification, we collect datasets with multiple people walking in the scene simultaneously. The results are reported in Figure 10. We consider *SpotFi* as our baseline. It uses AoA data generated by the SpotFi algorithm and identifies the subject through Euclidean distance. *EyeFi* uses weighted Euclidean distance for identification on the two staged smoothed data generated by the teacher-student neural network. As stated in Section V-C, we only apply our weighted distance for the sequence size of fewer than 200 packets. The accuracy is calculated by sliding a window of size N along the time-series data, identify the subject with the AoA data within the window range, then computing the percentage of the number of accurately identified time-segments throughout the test dataset. The window size starts at 5, which means using only 5 packets. For example, in Figure 10(a), we can successfully identify the subject 90% of the time in a normal 2 people scenario. Since we collect data at a rate of 20-25 packets per second, 5 packets represent a duration of 0.25s or less. As more people are present in the scene, the difficulty of identification increases. As shown in Figure 10(b), the accuracy is worse than the 2 people case. The performance further drops with 5 people in the scene as shown in Figure 10(c).

Figure 10(d) shows the result of identification with 10 people walking simultaneously. The results show that EyeFi can still identify even though the accuracy drops in smaller windows. The identification accuracy of 10 people is a bit higher than that of 5 people when the window size is 500. This can be due to the human subject who is holding the phone walks in a different path than the rest of the group. This allows the identification can be 100% accurate when the window size is large. However, in real situations, 500 packets span about 25 seconds during our data collection. With an average human walking speed of 1.1 meters per second, a person can walk about 27.5 meters during that time frame. Given the number of packets large enough and the path is different enough to be distinguished from other trajectories, EyeFi could achieve near 100% accuracy in identification.



(a) 2 People. **(b) 3 People.** **(c) 5 People.** **(d) 10 People**

Fig. 10: Identification accuracy for different number of people.

VIII. Discussion

A. Limitations of Our Proposed Solution

There are two major limitations of EyeFi. First, it requires having a smartphone with WiFi communication. Second, it is unable to identify if multiple people walking together in a way that results in similar AoA trajectories. For the first limitation, the usage of the phone is very common but not all users will connect to public WiFi such as the ones provided by the store in our case. However, EyeFi can still receive WiFi packets generated by phones while discovering local access points. This property allows the system to estimate AoA and track the subject. However, if the WiFi interface is disabled or the smartphone is not generating any WiFi traffic, then EyeFi will not be able to identify people. But the camera can still provide analytics on behavior of people to some extent. For the second limitation, in retail stores, if the trajectories are the same for a few people, identifying either could provide the same analytic regarding customer behavior.

B. Motion Across Multiple Cameras

In areas where multiple cameras are deployed with or without shared field of view, EyeFi can leverage existing camera-based re-identification algorithm [33] to identify the same user across multiple cameras to provide full trajectory in the covered area. By leveraging WiFi, EyeFi can enable identification at different spatio-temporal segments, thus reducing the search space for the vision based identification and enable more accurate long-term re-identification.

C. Generalization to Multiple Phones

The presence of multiple phones in the scene will have minimal effect on the performance of the system. This is due to the nature of WiFi communication that enables multiple devices to talk to each other without interference. This also means that the CSI information collected at the WiFi unit for each device has the same quality. EyeFi only relies on CSI information and does not require a high transmission rate which further reduces the impact of multiple devices. During

our data collection, all other WiFi devices and communications are functioning normally and no effects are observed.

D. Effects of Using Phones

Differs from most previous works that use Intel 5300 NIC for both communication devices, we use a smartphone at one end. The internal antenna design is vastly different from the external antenna used with Intel 5300 NIC devices and the holding of the phone by a human subject also affects the WiFi signal quality. During our experiments, we also observe poor performance and stability issues in AoA estimation with phones in comparison to Intel 5300 NIC with external antennas, and the AoA results are worse than that is reported in previous works such as [12]. Note that Linux CSI Tool [8] offers the best performance and stability using injection mode, which is currently unavailable with the phone.

E. Effects of Environment

In our evaluation, we test EyeFi in two different environments that shows stable performance among them. This demonstrates the robustness and generalizability of our system. However, different environments can affect the performance of WiFi AoA estimation if the environment is very crowded with obstacles between the phone and WiFi access point to create a non-line-of-sight situation. In such a situation, the WiFi signal is distorted which degrades the AoA estimation performance. However, if the trajectories between different subjects in the scene are different from each other, the system should be able to identify. In our experiment, there are times where the phone is blocked by human bodies which distort the WiFi signal as well. With our smoothing pipeline, the identification can still perform well.

F. Privacy Issues

Privacy is a major concern nowadays and we design EyeFi with that in mind. Current camera-based systems mostly use facial recognition for user re-identification across time. However, facial recognition is unreliable in many settings (discussed earlier) and can be racial biased. There are also laws [2] to ban facial recognition to prevent such bias and protect privacy. In contrast to vision-based facial recognition systems to track a user across time, EyeFi only collects the MAC address of the phone and hashes that to obtain a consistent identification marker. Given most human subjects do not change phones very frequently, the hashed MAC address can be used as a reliable marker across time. As EyeFi does not keep the link between a hashed MAC address and its particular user, even if the hashed MAC addresses are compromised, it will be very difficult to use that to identify a particular user.

IX. Related Work

A. WiFi CSI AoA Estimation

WiFi CSI based AoA estimation has been explored in many works. [12] exploits CSI values of WiFi subcarriers to extend the number of virtual antenna for obtaining AoA of the direct path along with several strong multipaths. They cluster AoAs of multipaths from multiple packets and choose AoA of direct path based on cluster quality. To localize a subject their work relies on triangulation from multiple APs. [24] propose an algorithm for sub-nanosecond time-of-flight calculation, from which they localize a subject with one AP. [25] use frequency domain super-resolution algorithms to overcome the bandwidth limitation of WiFi for precise localization. [27] extracts subcarriers less prone to multipaths and uses them to estimate AoA. [22] proposes to use multipath reflections to triangulate the subject instead of using multiple APs. [21], [28] uses fingerprinting to map WiFi physical layer properties for localization. [20] studies human movement's effect on WiFi CSI based AoA estimation. [7] proposes phased signal processing for localization using triangulation that requires coordination of nodes for precise localization. [32], [34] propose CSI based fingerprinting techniques for AoA estimation. Differs from previous works, our system utilizes a data-driven model that results in fast computation that enables real-time application scenarios.

B. Multi-Modal Localization

Recent works have proposed multi-modal localization systems. [5] uses mobile phone sensors to associate a person visual information. The idea is to match a person's movement signature between these two modalities and use this signature as MAC address for enabling the public server to send messages to the particular person. [4] proposes to use WiFi RSSI strength as an indication to depth information of the user. They use that depth information in association with RGB image from the camera to localize a person. [17] uses extended Kalman filter with accelerometer and WiFi RSSI as input for tracking people in a dynamic environment. This work relies on a pre-computed WiFi signal strength map for an environment. [30] uses a particle filter to integrate vision and radio localization system for sub-meter localization accuracy. In [18], [11], the fusion of vision and RF is done for different applications, e.g., fall detection, recalibration, and industrial workspaces. However, these solutions require the deployment of multiple cameras and/or RF units and none uses CSI information.

C. Identification

One of the main motivations behind EyeFi is the ability to identify human subjects across time. ID-Match [13] uses RFID tags and a 3D depth camera to identify and assign IDs to each person. RFID and BLE are used in [14] to identify individuals. These works require additional sensors or devices on the human subject which is inconvenient and can be potentially costly. The use of additional tags also presents difficulties when trying to tack the same subject across time as they can be easily misplaced or forget to carry. For vision based systems such as [26] which uses human motion pattern and color of clothing. However, such systems do not work with panoramic cameras that we are using. FORK [16] uses a depth sensor mounted above a doorway for person detection and identification based on the body shape of individuals.

However, such an approach does not scale across thousands of people.

X. Conclusion

In this work, we propose EyeFi, a multimodal system that fuses WiFi and camera data to identify individuals by capturing motion trajectories from each modality. We design a teacher-student based neural network model to estimate AoA accurately, which speeds up AoA estimation by over 3800 times with 30% higher accuracy, enabling EyeFi to be a real-time system. We test the performance in two different environments and find the neural network based AoA estimation is robust to a change of the environment. When evaluating the accuracy of person identification, we see that EyeFi can achieve an average of 75% accuracy across all number of packets in a 2 to 10 people scenario. For future works, we will improve performance for each component and build an end-to-end system that can identify people in real-time.

Acknowledgement

This paper was supported, in part, by NSF grants CNS-1816213 and CNS-1704469, NIH grant 1R01LM013329-01, and a gift from Bosch.

References

[1] Face Recognition. https://github.com/ageitgey/face_recognition.

[2] San Francisco Banned Facial Recognition. https://www.nytimes.com/2019/07/01/us/facial-recognition-san-francisco.html.

[3] SpotFi Matlab implementation. https://bitbucket.org/mkotaru/spotfimusicaoaestimation/src/master/.

[4] A. Alahi, A. Haque, and L. Fei-Fei. Rgb-w: When vision meets wireless. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3289–3297, 2015.

[5] S. Cao and H. Wang. Enabling public cameras to talk to the public. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(2):63, 2018.

[6] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 742–751, 2017.

[7] J. Gjengset, J. Xiong, G. McPhillips, and K. Jamieson. Phaser: Enabling phased array signal processing on commodity wifi access points. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 153–164. ACM, 2014.

[8] D. Halperin, W. Hu, A. Sheth, and D. Wetherall. Tool release: Gathering 802.11n traces with channel state information. *ACM SIGCOMM CCR*, 41(1):53, Jan. 2011.

[9] O. Hamdoun, F. Moutarde, B. Stanciulescu, and B. Steux. Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. In *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–6. IEEE, 2008.

[10] N. Khalil, D. Benhaddou, O. Gnawali, and J. Subhlok. Sonicdoor: scaling person identification with ultrasonic sensors by novel modeling of shape, behavior and walking patterns. In *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*, page 3. ACM, 2017.

[11] S. Kianoush, S. Savazzi, F. Vicentini, V. Rampa, and M. Giussani. Device-free rf human body fall detection and localization in industrial workplaces. *IEEE Internet of Things Journal*, 4(2):351–362, 2016.

[12] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM computer communication review*, volume 45, pages 269–282. ACM, 2015.

[13] H. Li, P. Zhang, S. Al Moubayed, S. N. Patel, and A. P. Sample. Id-match: A hybrid computer vision and rfid system for recognizing individuals in groups. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 4933–4944. ACM, 2016.

[14] D. F. Llorca, R. Quintero, I. Parra, and M. Sotelo. Recognizing individuals in groups in outdoor environments combining stereo vision, rfid and ble. *Cluster Computing*, 20(1):769–779, 2017.

[15] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[16] S. Munir, R. S. Arora, C. Hesling, J. Li, J. Francis, C. Shelton, C. Martin, A. Rowe, and M. Berges. Real-time fine grained occupancy estimation using depth sensors on arm embedded platforms. In *2017 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 295–306. IEEE, 2017.

[17] S. Papaioannou, H. Wen, Z. Xiao, A. Markham, and N. Trigoni. Accurate positioning via cross-modality training. In *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pages 239–251. ACM, 2015.

[18] S. Savazzi, V. Rampa, F. Vicentini, and M. Giussani. Device-free human sensing and localization in collaborative human–robot workspaces: A case study. *IEEE Sensors Journal*, 16(5):1253–1264, 2015.

[19] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.

[20] M. Schüssel. Angle of arrival estimation using wifi and smartphones. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, page 7, 2016.

[21] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. You are facing the mona lisa: Spot localization using phy layer information. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 183–196. ACM, 2012.

[22] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse. Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver. 2018.

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[24] D. Vasisht, S. Kumar, and D. Katabi. Decimeter-level localization with a single wifi access point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 165–178, 2016.

[25] C. Wang, Q. Yin, and H. Chen. Robust chinese remainder theorem ranging method based on dual-frequency measurements. *IEEE Transactions on Vehicular Technology*, 60(8):4094–4099, 2011.

[26] H. Wang, X. Bao, R. R. Choudhury, and S. Nelakuditi. Insight: recognizing humans without face recognition. In *Proceedings of the 14th Workshop on Mobile Computing Systems and Applications*, page 7. ACM, 2013.

[27] J. Wang, H. Jiang, J. Xiong, K. Jamieson, X. Chen, D. Fang, and B. Xie. Lifs: low human-effort, device-free localization with fine-grained subcarrier information. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 243–256. ACM, 2016.

[28] X. Wang, L. Gao, and S. Mao. Csi phase fingerprinting for indoor localization with a deep learning approach. *IEEE Internet of Things Journal*, 3(6):1113–1123, 2016.

[29] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. *arXiv preprint arXiv:1505.00853*, 2015.

[30] J. Xu, H. Chen, K. Qian, E. Dong, M. Sun, C. Wu, L. Zhang, and Z. Yang. ivr: Integrated vision and radio localization with zero human effort. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 3(3):114:1–114:22, Sept. 2019.

[31] D. Zhang, Y. Hu, Y. Chen, and B. Zeng. Calibrating phase offsets for commodity wifi. *IEEE Systems Journal*, PP:1–4, 03 2019.

[32] L. Zhang, E. Ding, Y. Hu, and Y. Liu. A novel csi-based fingerprinting for localization with a single ap. *EURASIP Journal on Wireless Communications and Networking*, 2019(1):51, 2019.

[33] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang. Camera style adaptation for person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5157–5166, 2018.

[34] R. Zhou, J. Chen, X. Lu, and J. Wu. Csi fingerprinting with svm regression to achieve device-free passive localization. In *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–9. IEEE, 2017.