DeepRange: Acoustic Ranging via Deep Learning

WENGUANG MAO*, University of Texas at Austin, USA
WEI SUN*, University of Texas at Austin, USA
MEI WANG, University of Texas at Austin, USA
LILI QIU, University of Texas at Austin, USA

Acoustic ranging is a technique for estimating the distance between two objects using acoustic signals, which plays a critical role in many applications, such as motion tracking, gesture/activity recognition, and indoor localization. Although many ranging algorithms have been developed, their performance still degrades significantly under strong noise, interference and hardware limitations. To improve the robustness of the ranging system, in this paper we develop a Deep learning based Ranging system, called DeepRange. We first develop an effective mechanism to generate synthetic training data that captures noise, speaker/mic distortion, and interference in the signals and remove the need of collecting a large volume of training data. We then design a deep range neural network (DRNet) to estimate distance. Our design is inspired by signal processing that ultra-long convolution kernel sizes help to combat the noise and interference. We further apply an ensemble method to enhance the performance. Moreover, we analyze and visualize the network neurons and filters, and identify a few important findings that can be useful for improving the design of signal processing algorithms. Finally, we implement and evaluate DeepRangeusing 11 smartphones with different brands and models, 4 environments (*i.e.*, a lab, a conference room, a corridor, and a cubic area), and 10 users. Our results show that DRNet significantly outperforms existing ranging algorithms.

CCS Concepts: • **Human-centered computing → Interaction techniques**; **Ubiquitous and mobile computing**; • **Computing methodologies → Neural networks**;

Additional Key Words and Phrases: Acoustic Sensing; Ranging; Motion Tracking; Convolutional Neural Network.

ACM Reference Format:
Wenguang Mao, Wei Sun, Mei Wang, and Lili Qiu. 2020. DeepRange: Acoustic Ranging via Deep Learning. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 4, Article 143 (December 2020), 23 pages. https://doi.org/10.1145/3432195

1 INTRODUCTION

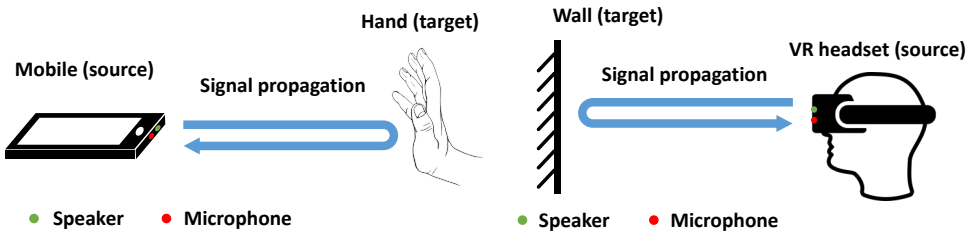
1.1 Motivation

Ranging is the technique to estimate the distance from a signal source to a target object. It is a fundamental building block for localization, motion tracking, gesture and activity recognition, which have a wide variety of applications. For example, it enables gesture based interface to remotely control smart appliances, virtual reality (VR), augmented reality (AR), and gaming. It offers an effective way of sensing environments for wireless

*Both authors contributed equally to this research.

Authors' addresses: Wenguang Mao, wmao@cs.utexas.edu, University of Texas at Austin, 2317 Speedway, Stop D9500, Austin, Texas, USA, 78712; Wei Sun, weisun@cs.utexas.edu, University of Texas at Austin, 2317 Speedway, Stop D9500, Austin, Texas, USA, 78712; Mei Wang, meiwang@cs.utexas.edu, University of Texas at Austin, 2317 Speedway, Stop D9500, Austin, Texas, USA, 78712; Lili Qiu, lili@cs.utexas.edu, University of Texas at Austin, 2317 Speedway, Stop D9500, Austin, Texas, USA, 78712.

2474-9567/2020/12-ART143 $15.00
https://doi.org/10.1145/3432195



(a) Motion tracking or gesture recognition (b) Indoor mapping or obstacle detection

Fig. 1. Applications for passive acoustic ranging.

optimization (*e.g.*, beamforming, AP selection), habitat monitoring, disaster recovery, user tracking, health monitoring, and context aware applications.

Ranging can be classified into (i) active ranging, where the target can either send or receive signals and (ii) passive ranging, where the target can be any object and simply reflects the signal. In this paper, we consider passive ranging using the reflected signals. As shown in Figure 1, in this case the source sends and receives signals to measure the round trip propagation delay from the target (*e.g.*, a user's hand) and compute the distance.

Passive ranging is useful for many applications, such as gesture recognition, indoor mapping, virtual reality (VR), and augmented reality (AR). For example, we can use ranging techniques to create a map for indoor environments by measuring the distance to walls and furniture. We can also estimate the distance to nearby obstacles for safety applications.

A number of ranging approaches have been developed, based on different signals, such as audios [22, 26, 27, 30, 36, 41, 44, 51, 52, 56], radio frequency (RF) [2, 3, 12, 15, 19, 21, 32, 34, 40, 43, 48, 50], and infrared lights [10]. Compared to other types, acoustic ranging has following advantages. First, the slow propagation speed of acoustic signals is beneficial to achieve high accuracy in distance estimation. Second, most devices are already equipped with speakers and microphones, and we do not need to deploy extra hardware (*e.g.*, RFID readers and tags, millimeter wave antennas, and depth cameras) for ranging. Third, the sampling rate of acoustic signals is low so that the processing can be done in software. This makes acoustic ranging easily available on commodity devices (*e.g.*, smartphones and VR/AR headsets). Due to these advantages, we focus on acoustic ranging in this paper.

1.2 Limitation

Existing acoustic ranging approaches exploit various signal processing techniques. Frequency-modulated continuous wave (FMCW) is one of the most widely used approaches (*e.g.*, [22, 26]). Other works use correlation with known signal patterns [27, 30], or measures the phase changes [41, 44, 52]. However, the performance of signal processing approaches degrades significantly under low signal-to-noise ratio (SNR) and multipath propagation. As shown in Figure 3(b), when the SNR is low, there are many false peaks in the FMCW profile and it is challenging to locate the correct peak corresponding to the target. As result, the existing passive ranging approaches can only work when the target is within tens of centimeters [27, 44, 52]. Figure 3(c) further shows the impact of multipath where the reflected signals from the target and nearby objects interfere and result in a merged peak, which makes it hard to locate the target.

1.3 Our Approach

Existing ranging algorithms are designed by domain experts. In this paper, we explore the following interesting questions: Can a deep neural network automatically learn the features in the received acoustic signals to estimate the distance? Can it outperform traditional signal processing algorithms designed by domain experts?

This direction is interesting for several significant reasons. Scientifically, it is interesting to understand the feasibility of the machine learning approach in automatically learning features. This learning task seems challenging since unlike images or videos, where humans can easily determine the correct answers (*e.g.*, image label), human is not good at estimating the distance from acoustic signals. Practically, if successful, the resulting approach can improve the accuracy of ranging and benefit a wide range of applications. Moreover, it can also shed light on the limitations and potential of signal processing versus machine learning approaches. Such insights will help us design new algorithms that achieve the best of both worlds.

Note that our work is inspired by the tremendous success of deep neural network (DNN) and its advantage in nonlinear problems. Motivated by its success in vision and speech recognition communities, we have seen applications of neural networks to ranging and tracking. For example, RF-Echo [6] applies a neural network with a single hidden layer to estimate the propagation delay of RF signals based on the correlation profile. RF-Pose [57, 58] develops a convolutional neural network (CNN) to estimate a user's pose based on the heatmap generated by applying FMCW to RF signals. Different from these works, which use the features designed by domain experts as the neural network input, we aim to automatically learn the features from raw acoustic signals. Moreover, unlike the existing works, which require training data from real testbeds and can be time-consuming and labor intensive, we aim to automatically generate the training data.

In order to apply DNN to the received signals, we need to address two major challenges. First, DNN requires a large volume of training data to work well, and it is important to have an efficient way of generating lots of training data. Second, we need to design a DNN that works well for distance estimation.

To address the first challenge, we develop a simulator that models how acoustic signals propagate through the environment. Through extensive trials, we find that not only noise and multipath (*i.e.*, reflections from objects other than the target) affect the learner performance, but also self interference (i.e., the signal directly going from the transmitter to the receiver) and speaker/microphone distortion have a significant impact. Therefore, our simulator captures all these factors. To derive a general model, we add randomness when generating signals to prevent the network from overfitting specific values or patterns. For example, not only the noise in our synthetic signals is a random Gaussian variable, but its standard deviation is also randomly chosen. Similarly, we randomly synthesize a piece-wise polynomial function to capture the frequency responses of speakers and microphones. In this way, our simulator achieves simplicity, generality, and realism. We only use the data generated from our simulator to train DNNs for ranging and show that they work well for real signals using 11 different smartphones, 10 users, different targets, and 4 different locations.

To address the second challenge, we start with a generic multi-layer fully connected neural network. Interestingly, we find the weights connected to each neuron in the first layer have a high correlation with the transmission signals at different shifts. This insight motivates us to develop a CNN. We first try the traditional CNNs, such as AlexNet [16] and VGG [35], but find they do not work well because their filters have too short kernel size (*e.g.*, 3×3 or 11×11) and fail to detect local patterns under noise and interference. To robustly capture signal patterns, we develop a CNN with filter sizes comparable to the length of transmission signals.

Moreover, we find the network weights converge to different values during different runs. Intuitively, these weights correspond to different ways of feature extraction for ranging. Therefore, instead of training a single network, we train a set of networks and combine them using an ensemble model. To maximize the effectiveness of ensemble learning, we add randomness during training by using random initialization, applying dropout, and using different sets of training data.

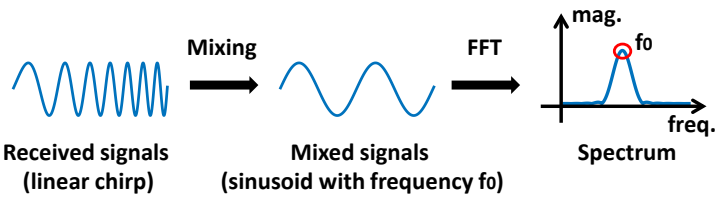


Fig. 2. FMCW processing stages.

We evaluate our approach by training DRNet using synthetic data generated from our simulator and testing on the acoustic signals collected from 11 phones with different brands, 10 users, different targets, and 4 environments including a lab, a conference room, a corridor, and a cubic area. The evaluation results show that our network generalizes well to different scenarios. Compared to three baseline approaches that use FMCW, correlation, and phases, our learning based approach achieves up to 5 times improvement on the ranging accuracy when the SNR is low and the interference is severe. Our contributions are summarized as follows:

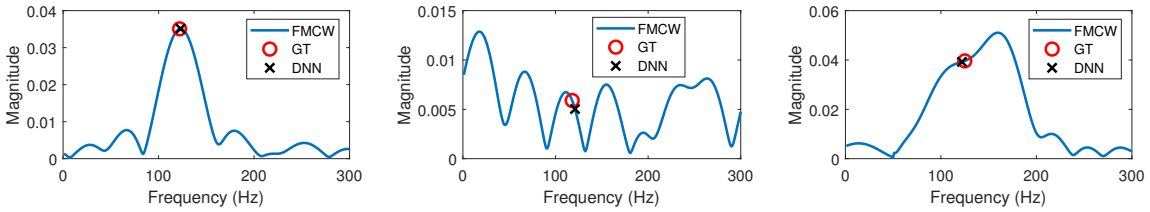
- We develop the first DNN based ranging that takes the raw received acoustic signals without feature extraction as the input. It significantly outperforms the existing signal processing algorithms designed by domain experts.
- We show our simulator not only captures noise, speaker/mic distortion, and interference but also generates diverse enough training data so that the model learned from the simulation data generalizes well to real data collected from a variety of scenarios.
- We analyze the DNN structure and identify several important findings that can potentially help improve existing signal processing methods.

2 LIMITATIONS OF EXISTING APPROACHES

There are a number of acoustic ranging algorithms, such as frequency-modulated continuous-wave (FMCW) [22, 51], correlation with known sequence [27, 30, 56], and monitoring phase changes [41, 44, 52]. According to our experiments, FMCW and correlation based methods out-perform the phase-based method under low SNR and/or strong multipath. In this section, we briefly describe how FMCW works and use it to illustrate why existing approaches do not work well under strong noise and multipath.

To estimate the distance propagated by the signals, we let the speaker periodically send chirps whose frequency linearly increases over time as shown in Figure 2. Upon receiving the chirp reflected by the target, we perform a mixing operation (*i.e.*, multiply the received chirp with the transmission signal) and apply a low-pass filter. It can be shown that the mixed signal is a sinusoid with the frequency proportional to the signal propagation delay [3, 22]. To determine the delay, we estimate the frequency of the mixed signal by applying Fast Fourier Transform (FFT) on the mixed signal and finding the peak frequency in the spectrum. Figure 3(a) shows an example for the spectrum derived by FMCW, where the ground truth frequency is labeled by a red circle.

With other ranging techniques, the performance of FMCW degrades significantly when the SNR is low. In this case, the spectrum derived as above becomes very noisy as shown in Figure 3(b), and it is difficult to locate the target because the target location does not correspond to the highest peak in the spectrum. The performance of FMCW also degrades under severe multipath. For example, when we place another object next to the target, the microphone will receive reflections from both the target and the nearby object. The peaks corresponding to both the target and object may interfere and merge, which makes it difficult to locate the target as shown in Figure 3(c). However, while it is challenging to design a hand-crafted heuristic to select a peak corresponding to the target, it may be possible for deep learning to automatically learn the pattern by mining a large amount of data. For example, the shape of a real peak may be quite different from the noise. Similarly, by analyzing the shape of a



(a) Ideal (b) Low SNR (c) Interference

Fig. 3. The spectrum derived by FMCW under ideal, low SNR, and severe interference scenarios.

merged peak, it seems possible to locate the first peak if the target can be assumed to be the closest object after interference cancellation.

We also evaluate the impact of speaker/microphone frequency response and find it does not have a significant impact on FMCW performance. However, learning based approaches require training traces with realistic speaker/microphone frequency response; otherwise, there is significant performance degradation as shown in Section **??**.

In summary, the existing ranging algorithms are developed based on a solid theoretical foundation. However, they face challenges arising from low SNR, multipath interference, and speaker/microphone frequency response. Deep neural network (DNN) has the potential to address these challenging scenarios by automating feature extraction. In fact, our DNN can accurately estimate the distance to the targets in the examples shown in Figure 3 by automatically learning from labeled data.

3 APPROACH

In this section, we develop a DNN to estimate the distance based on received signals. To minimize the overhead of collecting training data, we develop a simulator to synthetically generate training data that captures noise (*e.g.*, ambient sounds or random acoustic noise), interference (*e.g.*, the signals propagated from the direct path and reflections from non-target objects), and speaker/mic distortion (*e.g.*, uneven frequency response of the speakers and microphones). We further develop a convolutional neural network (CNN) with long kernel sizes to achieve high accuracy and outperform both classic CNN and fully connected neural networks. We also propose an ensemble method to further enhance the performance.

3.1 Signal Generation

Basic model: We use $x(t)$ to denote the transmission signal over time t, which is a chirp as discussed in Section 2, where $0 \leq t \leq T$, and T is the transmission period. We use $y(t)$ to represent the received signal and it is what we need to generate. Ideally the received signal is the transmission signal after certain attenuation and delay. Therefore, one can generate received signals as follows: $y(t) = a_0 x(t - t_0)$, where a_0 is the attenuation coefficient and t_0 is the delay. We train a neural network based on signals generated in this way, and test it using real data (refer Section 4 for details about our neural network and testing data). It achieves a 1.9 cm median distance estimation error, which is much higher than 0.75 cm achieved by FMCW.

To achieve better performance, we should generate training data that is more similar to real data and captures various factors in real environments.

Noise: First, we add a noise term $n(t)$ to our received signal as $y(t) = a_0 x(t - t_0) + n(t)$. We generate $n(t)$ following Gaussian distributions with a zero mean and standard deviation σ. σ has a significant impact on learning performance. If σ is too small, the learner cannot gain knowledge about how to deal with low SNR. If σ

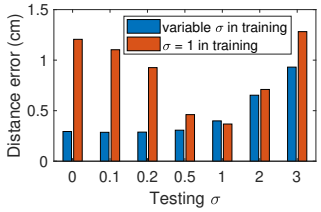


Fig. 4. The median errors with fixed and variable noise strength.

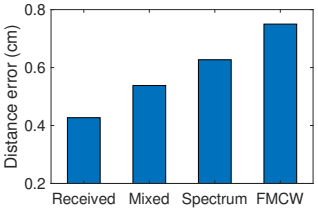


Fig. 5. The median errors using signals at various stages.

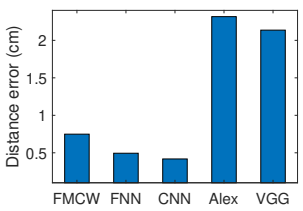


Fig. 6. The median errors with different network structures.

is too large, the signals are too noisy to learn features from them. Moreover, we cannot use a single value for σ, since the learner may overfit the particular noise level and does not generalize well to other situations. This is shown in Figure 4: when we use the synthetic data with a fixed σ to train a neural network, it does not work well even for the test cases with less noise. Note that the testing data in this experiment is generated using our simulator so that we can control the noise level. For all other experiments in this section, we use the testing data collected from real environments as described in Section 4.

Therefore, our approach uses σ from a range $[0, 4]$. As a reference, the magnitude of signals reflected by the target at 0.3 m is set to 1. The upper bound of the range is tuned based on the experiments to give the best performance. It covers low SNR scenarios but prevents signals from getting too noisy to provide information. For each transmission period, we randomly choose a value from that range and generate noise following a Gaussian distribution. After taking into account the noise, the distance estimation error of our neural network reduces to 1.1 cm, but is still much higher than FMCW.

Multipath: Multipath is a common phenomenon in wireless signal propagation [39], where the signal generated from the transmitter takes multiple paths to reach the receiver. The signal along each path is a delayed and attenuated version of the transmission signal. The final received signal is the superposition of signals along all the paths. To capture this effect, our generation model becomes

$$y(t) = \sum_{i=0}^{L} a_i x(t - t_i) + n(t), \tag{1}$$

where t_i is the propagation delay sorted in an ascending order. The parameter we need to decide is how many reflection paths (*i.e.*, L) should be added into signals. While there could be many reflection paths in practice, most of them are static (*e.g.*, reflection paths from furniture and walls) and can be removed using interference cancellation [5, 25]. The main idea is to record the reflection from these objects in advance when the target is absent and then remove them when collecting the signals to estimate the distance to the target. After interference cancellation, only a few reflection paths remain. In our approach, we generate data to emulate signals after interference cancellation, and use them as the neural network input. This not only reduces the number of reflection paths, but also removes the environment dependency because most reflections from the environment are removed. Based on our experiments, we only need to generate 0–4 reflections (excluding the one coming from the target) with random propagation delay in the training signals and our neural network generalizes well to the cases with more reflections. Further increasing the number of reflection paths leads to little additional improvement.

For the attenuation coefficient a_i, we first set its value inversely proportional to the propagation delay of the i-th path so that the signal energy on that path follows the inverse-square law [39]. Then we multiply the above value with a random number in the range $[1/2, 2]$ to take into account the other factors that affect the reflection strength, such as object materials and sizes. The multiplier larger than 1 indicates a stronger reflector with a

larger reflection area or less absorption. Note that choosing another range (*e.g.*, $[1/3, 3]$) may achieve similar performance. The key point here is to add randomness to prevent the neural network from relying on only the signal strength. After incorporating multipath in the training data, our neural network out-performs FMCW and achieves a 0.58 cm median distance estimation error.

Self interference: In addition to the target reflection, the transmission signals also propagate directly from the speaker to the microphone, which we call self interference. Since the relative position between the speaker and microphone on a mobile is fixed, the signals through the direct path can be removed by self interference cancellation. However, the transmission signals sent at different time are slightly different in practice due to variation in device temperature, power supply, and self interference channel. Therefore self interference cancellation is not perfect. Since self interference is orders of magnitude higher than the target reflection due to a small separation between the speaker and microphone, the residual self interference can still be relatively large compared with the target reflection and should be taken it into account in our signal generation.

To model the residual self interference, we use $[1 + \epsilon(t)]x(t)$ to represent the transmission signal, where $\epsilon(t)$ captures the variation. Without loss of generality, we generate $\epsilon(t)$ using random splines whose magnitude is within ϵ_{max}. According to our observations, the transmission signal variations are usually on the order of 10^{-3}. Therefore, we set the upper bound ϵ_{max} as 0.01.

By applying interference cancellation, the residual self interference becomes $a_s\epsilon(t - t_s)x(t - t_s)$, where a_s and t_s are the attenuation coefficient and propagation delay of the self interference. Since the signal on the direct path always has the shortest propagation delay, it corresponds to the first term in Equation 1. Based on the above discussion, our signal generation model becomes

$$y(t) = a_0\epsilon(t - t_0)x(t - t_0) + \sum_{i=1}^{L} a_i x(t - t_i) + n(t).$$

By incorporating the self interference, the median ranging error of our neural network reduces to 0.51 cm.

Transceiver frequency response: Another important factor needed to be taken into account is the frequency response[31] of the speaker and microphone. Ideally, they should have the same gain across the entire frequency. However, real speakers and microphones have different gains across different frequencies, especially for those above 18 KHz (used by our transmission signals), as they are hardly audible and not optimized. To emulate this effect, we let our synthetic signals pass through a digital filter with uneven frequency gains. For each transmission period, we use different filters to cover various possibilities. The frequency responses of these filters are generated using random splines. Thus, our signal generation model becomes

$$y(t) = a_0\epsilon(t - t_0)\tilde{x}(t - t_0) + \sum_{i=1}^{L} a_i\tilde{x}(t - t_i) + n(t), \tag{2}$$

where $\tilde{x}$ stands for the signals distorted by uneven frequency response. After incorporating this effect, the median distance estimation error of our neural network reduces to 0.42 cm.

As we will show, our synthetic data generated in this way are both realistic and diverse by capturing the important real-world effects, such as multipath, noise, and speaker/microphone distortions. So we transform the target distance x to the received signal y as $y = f(x)$. However, it is challenging to infer x based on y since $f()$ is non-linear, unknown in advance, and varies over time and across environments and speakers/microphones. Neural network is an effective way to model a non-linear and complex relationship between the input (*i.e.*, the received signals) and output (*i.e.*, the target distance). In the following sections, we develop a DNN to estimate the distance based on the raw acoustic signals.

3.2 Deep Neural Network for Ranging

A neural network includes three elements: input, output, and network structure. The output of our neural network is the distance between the transceiver and target. Since there could be a few objects whose reflections are not removed by interference cancellation [25], we assume that our target is the one closest to the transceiver to avoid ambiguity. In our signal model in Equation 2, the target reflection corresponds to $a_1\tilde{x}(t - t_1)$. Note that any passive ranging technique needs certain assumptions to distinguish the target reflection from others. We assume the target is the first reflection after static background cancellation since it holds more often than alternative assumptions (*e.g.*, the target is the largest reflection). For example, when a user puts his hand toward a mobile for tracking, the hand is closer to the phone than the arm and body, but the body reflection may be stronger due to the larger reflection area.

There is an interesting trade-off regarding which input to use for training. Using the signals at later stages as the input means relying more on feature extraction and less on machine learning. Since feature extraction can reduce the input dimension and make the relationship between the input and output more clear, training becomes easier. On the other side, since feature extraction may also remove some useful information, using the signals at earlier stages might potentially achieve better performance. Figure 5 shows the performance of the DNNs trained with signals at different stages. Refer to Section 4 for the details about the neural network and testing data. We see that using received signals after pre-processing (*i.e.*, band-pass filtering and interference cancellation) provides the best performance. Since the pre-processing only removes unwanted artifacts from signals (*e.g.*, out-of-band noise and background reflections), it is beneficial to apply pre-processing to get a cleaner version of received signals. Therefore, we choose them as our network input.

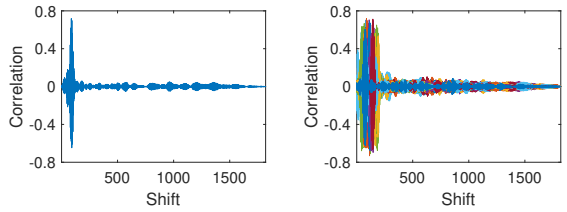
For the network structure, we start with a multi-layer fully-connected neural network (FNN), which is the most general structure. Using 6 hidden layers, 50 neurons in each layer, a 1920-element vector as the input (representing the received signal duration each period), and 200 K synthetic training samples, FNN can achieve much lower distance estimation error than FMCW, as shown in Figure 6. The median errors of FMCW and FNN are 0.75 cm and 0.49 cm, respectively.

To further improve the network structure, we examine the weights in our FNN. Interestingly, the weights for the first layer have high correlation with the delayed versions of the transmission signal. To illustrate that, we take one neuron from the first layer and use $\mathbf{w}$ to denote the weights connected to it. Let $\mathbf{x}$ represent the discretized transmission signal. We calculate the cross-correlation between $\mathbf{c}$, $\mathbf{w}$ and $\mathbf{x}$ as

$$c[i] = \frac{< \mathbf{w}, \mathbf{x_i} >}{|\mathbf{w}| \cdot |\mathbf{x_i}|},$$

where $< \cdot, \cdot >$ represents the inner product, $|\cdot|$ stands for the L2 norm, and $\mathbf{x_i}$ denotes $\mathbf{x}$ delayed by i elements. If $c[i]$ is close to one, $\mathbf{w}$ has high similarity to $\mathbf{x_i}$. Figure 7 plots $\mathbf{c}$ for all neurons in the first layer. We see that the neuron weights have a high correlation with the transmission signal shifted by different amounts. This reminds us CNN, where the same set of weights are shifted by different amounts and applied to different portions of the input. Based on the above observation, we next develop a CNN for ranging.

We first investigate if traditional CNNs can be used for ranging. We train AlexNet and VGG networks customized to fit our application. The number of layers and the structure of these networks remain the same. The filters are changed to one-dimensional kernel (*e.g.*, a 3×3 filter becomes a 3×1 filter) because our inputs are one-dimensional. The number of filters in convolutional layers and the neurons in FC layers are tuned to have the same amount of parameters as our final model in Section 4. Further increasing the model size requires more training data and longer training time with only marginal improvement. As shown in Figure 6, both networks perform significantly worse than FMCW. In fact, traditional CNNs rely on convolutional filters with small kernel sizes (*e.g.*, from 3×3 to 11×11) to capture local patterns. Based on the local patterns, CNNs gradually constructs a global view about the input at upper network layers. This does not work well in our case since a short convolution is very sensitive



(a) Single neuron (b) All neurons

Fig. 7. The correlation between shifted transmission signals and the weights in the first layer of our FNN.

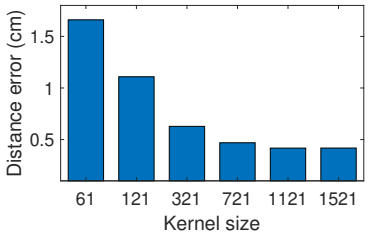


Fig. 8. The performance of CNNs with various kernel sizes.

to noise and interference, which are common in acoustic signals. If the low-level pattern detection is erroneous, it is challenging for upper layers to mitigate these errors.

To tolerate noise and interference, we develop a CNN using convolutional filters with long kernel sizes. Intuitively, a convolutional filter is used to detect a specific pattern in the signals. When the pattern is longer, it is less likely for noise or interference to resemble the specific pattern. Therefore long kernel sizes up to the transmission signals are more robust for detecting patterns. However, patterns longer than the transmission signal does not help improve the performance. Therefore, we use convolutional filters in the first layer to have similar length to that of the transmission signal. Figure 8 shows the performance of neural networks with one convolution layer and 5 fully connected layers but using different filter sizes. We see that the distance estimation errors first reduce, and then taper off as the filter size gets close to the length of our transmission signal (*i.e.*, 1440). Note that we use a grid search to determine the hyper-parameters in DRNet. To illustrate how a particular parameter affects the performance, we show figures by varying one parameter while keeping the other parameters the same.

The final design for DRNet is described in Section 4. Although it has a slightly smaller number of weights than our FNN, it achieves 20% lower distance estimation error as shown in Figure 6. As discussed, both the first layers in our FNN and CNN are used to detect certain patterns with different shifts, but the convolutional layer is more effective in capturing the pattern and achieves better accuracy.

3.3 Ensemble

A major advantage of using synthetic data is that it is easy to generate an arbitrary amount of training data. One way to leverage this benefit is to train larger networks with more data to improve the performance. Figure 9 shows the performance of CNNs with different sizes, measured by numbers of weights used by them. We change the network size by scaling the number of neurons in each layer of our default CNN and adjust the training data proportionally. We see the performance improvement tapers off when the number of weights in the network is larger than 100 K, which is the size for our FNN and CNN. As shown in Figure 18, further increasing the network depth does not help.

Another way to exploit a large amount of data is to train multiple CNNs and apply the ensemble method. [29] shows that a bagging ensemble nearly always outperforms a single classifier. The key observation is that our network converges to different local minimums in different runs, and these local minimums lead to comparable distance estimation errors when applied to real signals. Intuitively, different converging points indicate that the networks capture different features from the signals for distance estimation. These features respond differently to the noise and interference in the signals. By combining these networks (*e.g.*, using the median of outputs from all networks), we can average out the impact of noise and interference and potentially improve the ranging accuracy.

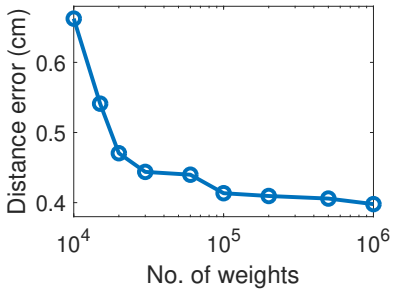


Fig. 9. The performance of CNNs with various network sizes.

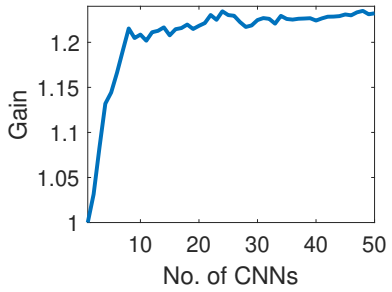


Fig. 10. The gain with various no. of CNNs for ensemble.

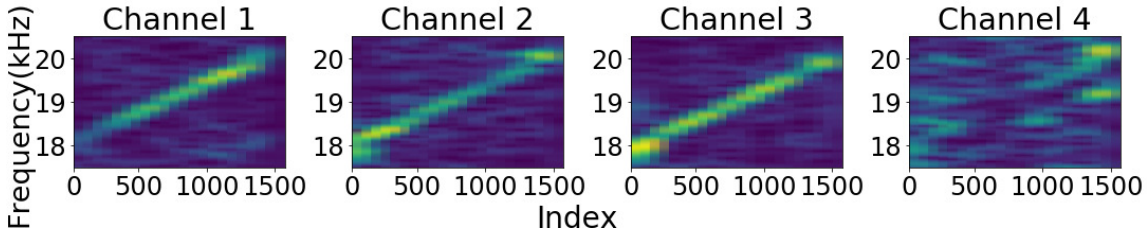


Fig. 11. The spectrograms of CNN filters.

A similar idea is explained in [9]. Figure 10 shows the performance gain of ensemble learning. For our application, it is interesting to see that using an ensemble of multiple CNNs is more effective than using a larger CNN.

To maximize the effectiveness of ensemble learning, we try to increase the diversity across the networks by using 1) random initialization, 2) a different set of synthetic data to train each network, and 3) random dropout with the probability equal to 0.95 at the input layers. The last strategy not only helps increase the randomness during training but also improves the generalization of networks.

3.4 Observations from CNN

In this section, we use visualizations to better understand DRNet. Instead of figuring out exactly how the neural network works, which remains an open challenge, we would like to gain insights about why CNN performs well and what we can learn from CNN to improve ranging algorithms.

Observation 1: Three convolution filters are chirps with different energy distributions across frequencies. Figure 11 shows the spectrograms of filter coefficients for four channels in the convolutional layers of DRNet. In this figure, the x-axis represents the index of filter coefficients in each channel, and the y-axis stands for the frequency. The color indicates the strength of a specific frequency at a certain portion of the coefficient sequence. As we can see, the spectrograms of the first three filters show the pattern of chirps from 18 KHz to 20 KHz. However, different from a standard linear chirp used by FMCW, these filter coefficients have different energy distributions across frequencies. For example, the filter in the first channel has more energy at the end of the filter sequence since we see a spot there, while the second filter has more energy at the beginning. The third filter has relatively uniform energy across the whole band. This structure may be helpful to handle uneven frequency response caused by speaker/mic distortion. Based on this observation, interesting future work is to explore chirps with non-uniform energy distribution across frequencies for FMCW.

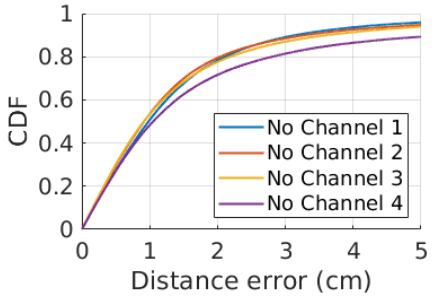


Fig. 12. Performance of removing one channel information.

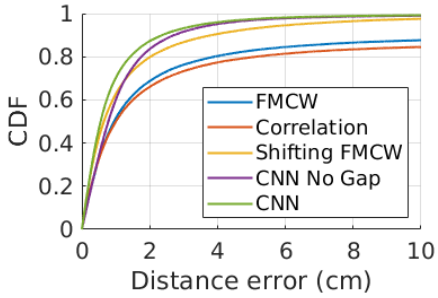


Fig. 13. Performance of different methods.

Observation 2: Combining multiple FMCW with transmission chirps shifted by different amounts is helpful to improve the performance. The first three filters in DRNet have similar patterns to the chirps, and received signals are multiplied by these filters multiple times with different shifts. In contrast, traditional FMCW only multiplies received signals with the transmitted chirp once with no additional shift.

Inspired by CNN, we explore whether it is beneficial to mix the received signals with the transmission signals with different shifts. More specifically, we shift the transmitted chirps by different numbers of samples so that each one corresponds to a new propagation delay. For each shifted transmitted chirp, we multiply it with the received signals and apply FMCW techniques to estimate the propagation delay. It has an offset from the true propagation delay due to the additional shift introduced to the transmitted chirp. We compensate for the offset and then average the estimation after compensation. As shown in Figure 13, this approach (denoted by "shifting FMCW") significantly outperforms traditional FMCW, though its ranging errors are still larger than DRNet. The improvement can be because using different shifts smooths out the errors arising from noise, speaker/mic distortion and multipath.

We also observe from Figure 13 that DRNet outperforms correlation based approach that selects the offset with the maximal correlation coefficient. This is likely because correlation only takes the peak but DRNet uses different filters to remove outliers and improve estimation.

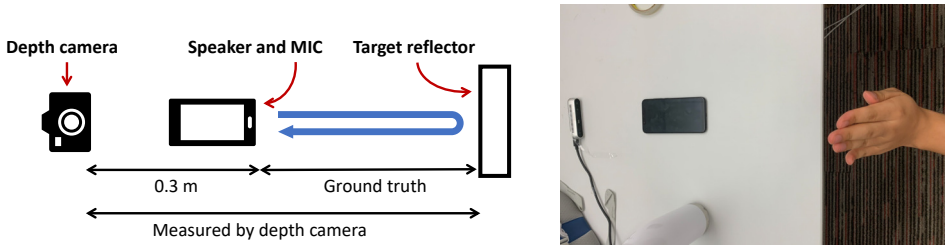
Observation 3: One convolution filter is not a chirp but has a strong impact on the performance. The fourth filter has a very different pattern from the others but has the most significant impact on the performance. Figure 12 shows the performance of removing one convolutional channel by setting the corresponding channel outputs as zeros. It shows that removing the fourth channel degrades performance more than removing any other channels.

To better understand the role of the fourth filter, we find that it has more energy near the head and tail of the filters. We expect it to leverage knowledge of noise, speaker/mic distortion and interference hidden in the non-chirp component of received signals. More specifically, the transmission signal is composed of two parts, a chirp and trailing zeros. Traditional FMCW only processes the chirp part of the receiving signal to compute the beat frequency. The part with zeros can provide information about noise and interference. The fourth filter leverages this part to improve the ranging performance. To verify that, we replace the received signals in this part with zeros, the error increases by 41%, as indicated by Figure 13 (denoted by "CNN No Gap").

4 IMPLEMENTATION

4.1 Acoustic Signals

To test our deep learning based acoustic ranging, we use the built-in speaker a smartphone to send chirp signals. The chirp frequency sweeps from 18 KHz to 20 KHz. According to [38], the absolute threshold of hearing (ATH)



(a) Illustration (b) Experiment at the office cubic

Fig. 14. Top down view of our experiment setup.

increases rapidly beyond 10 KHz. For example, human can hear the sound of 1 KHz at 0 dB sound pressure level (SPL), but over 75 dB SPL for sound beyond 17 KHz (10,000x). Our sound level at 18 KHz is 35 dB at 0.5 m from the speaker, well below ATH.

The chirp duration is 30 ms and the transmission period is 40 ms. We use the microphone on the same smartphone to receive the signals reflected from the target. The sampling rate of acoustic signals is 48 KHz so that the number of samples in a transmission period is 1920. We use the samples in one period as the input to our network for estimating the distance between the smartphone and target.

4.2 Training

For training, we generate synthetic data based on the signal parameters mentioned above and the model parameters discussed in Section 3. We generate ground truth distance and signals for 200 K transmission periods to train each CNN.

To tune the hyper parameters for DRNet, we generate synthetic testing signals for 100 K transmission periods. By applying the grid search, the final design is described as follows. DRNet has one convolutional layer with 4 filters. The kernel length for each filter is 1521. The convolution layer is followed by a max pooling layer with the stride and window size equal to 4. Then there are 5 fully connected layers with 200, 100, 50, 50, 50 neurons, respectively. We train our network in PyTorch [1] using Adam [14] optimization algorithm. The loss function is the mean square error. The batch size is 200. The initial learning rate is 0.001, which decays by a factor of 0.2 every 50 epochs. In total, we train 25 CNNs for ensemble learning.

4.3 Ground Truth

To evaluate the CNNs trained by synthetic data, we use them to estimate the distance between the mobile and target as shown in Figure 14 based on the signals recorded by the mobile. In the user study, the target is a user's hand. In other evaluations, the target is a 10 cm×10 cm white cardboard. The distance between the mobile and target varies from 0.2 m to 1.2 m in all experiments except the long range one (i.e., Figure 20), where the testing range is extended to 5 m.

To get the ground truth distance, we use the Intel RealSense D415 camera system [10]. Its accuracy is 1mm when the distance is between 0.3 m and 1.5 m based on our calibration. It comes with an RGB camera and a depth camera. We use the RGB image to find our target and read the depth from corresponding pixels in the depth image. Since the depth camera does not work when the distance is less than 0.3 m, we put it 0.3 m behind the phone as shown in Figure 14 so that the minimum distance between the camera and the target is larger than 0.3 m. The accuracy of our depth camera significantly degrades when the distance is larger than 1.5 m. For long range experiments, we manually calibrate the distance between the phone and target.

4.4 Testing

We collect testing traces in the following steps: 1) place the camera and smartphone as described above; 2) send the chirp signals with the smartphone speaker; 3) let the microphone record the signals for one second when the target is not present to capture the background reflection, which is used for interference cancellation [25]; 4) place the target and move it in front of the phone for one minute; 5) use the depth camera to get the ground truth distance; 6) use the microphone to receive the signals; 7) perform pre-processing on received signals, including band-pass filtering and interference cancellation [25]. Note that Step 5 is only used to quantify the accuracy but not required by our approach.

To demonstrate that our approach generalizes well, we use 11 smartphones with different brands and models to collect data, including Samsung S9 plus, Samsung S7 Active, Samsung S7 international version, Samsung S7 US version, iPhone X, iPhone 6, iPhone 5S, Huawei Mate 9, Huawei Honor 8, Xiaomi 8, and Google Pixel. The speakers and microphones on these phones have very different acoustic characteristics (*e.g.*, frequency responses) as shown in Figure 15. In this figure, the y-axis represents the normalized speaker and microphone amplification on signals at a specific frequency. We use the approach developed in [8] to measure the frequency response for a mobile. The testing traces are collected from 4 real environments, including a lab, a corridor, a meeting room, and a cube area. There are furniture and walls in all these locations. Besides static objects, there is also dynamic inference in our testing environments. For example, there are other people walking by our experiment setup. Moreover, the user's body and arm also exhibit some movements and introduce non-static interference.

We collected 119 testing traces. Each trace has 1375 transmission periods. In each period, we take the received signals as the input and the corresponding ground truth distance as the output label. The traces are divided into four groups, where all groups except the last one track a board. (i) 33 traces in ideal scenarios where the SNR is around 10 dB and there is no object near the target; (ii) 38 traces where SNR falls into -15–5 dB due to low speaker volume or large separation between the mobile and target, (iii) 28 traces with SNR around 10 dB and an object (a 10 cm×10 cm cardboard) behind the target (0–15 cm) to introduce severe interference. (iv) 20 traces for tracking different users' hands.

5 EVALUATION

In this section, we evaluate the performance of our ranging approach. We use the median and cumulative distribution functions (CDF) of distance estimation errors as our performance metrics. Except for the results shown in Figure 17, all evaluation uses the testing data collected from real environments as described in Section 4.

5.1 Micro Benchmark

We first evaluate various aspects of our approach, including signal generation, network generalization, network depth, and inference time. We report the ranging performance using a single CNN, and compare it with FMCW.

Signal generation: We evaluate the impact of key components in our signal generation. For this purpose, we generate training signals without noise (denoted by -N), without multipath (-M), without self interference (-S), or without frequency response (-F), respectively, and evaluate the performance of the CNN trained using signals without certain component. Figure 16 shows the testing performance on different testing data. We see that without adding noise, the group with low SNR has the largest median ranging errors, *i.e.*, 5.3 cm. Without adding multipath, the CNN does not work well for the group with strong multipath interference. Neglecting self interference and frequency response also degrades the performance. DRNet is trained using signals with all these components to achieve good performance across a wide range of scenarios.

Network generalization: We evaluate the performance of our network for the cases not covered by the training data. As mentioned in Section 3, σ (*i.e.*, standard deviation of noise) varies from 0 to 4 in our training data, and

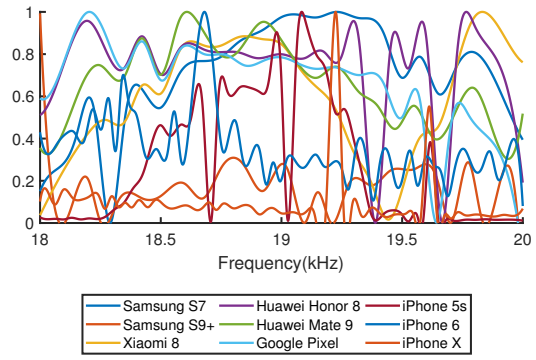


Fig. 15. Combined frequency response of speakers and mics for different phones at 18–20 KHz.

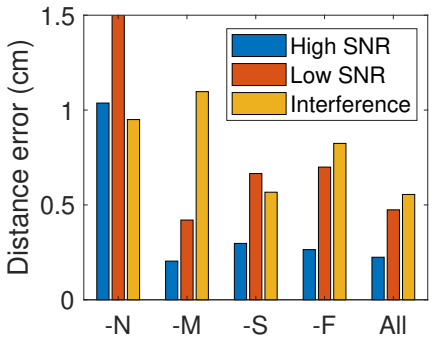
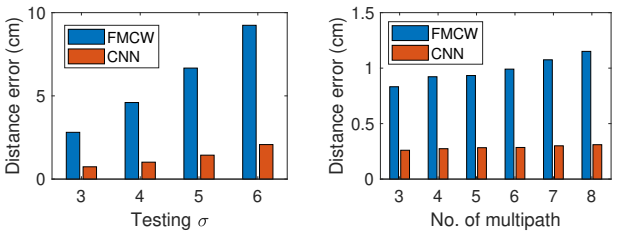


Fig. 16. The impact of signal generation.



(a) Noise (b) Multipath

Fig. 17. Generalization of our CNN.

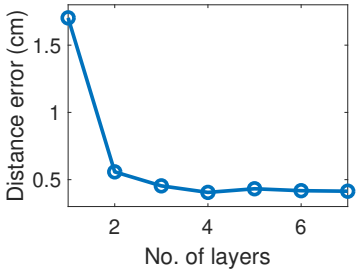


Fig. 18. The impact of numbers of layers.

the number of multipath (excluding the target reflection) is randomly chosen from 0 to 4. To test if DRNet works when the noise and multipath go beyond these ranges, we generate two sets of synthetic data with 1) σ varying from 3 to 6 and zero multipath and 2) the number of multipath varying from 3 to 8 and σ equal to 1. We use synthetic testing data in this experiment because we need to control the noise and multipath level. The testing performance on the two data sets is shown in Figure 17(a) and 17(b), respectively. We observe that the estimation errors of DRNet increase only marginally when the noise and multipath are outside the ranges covered by the training data, DRNet degrades slower than FMCW since it is more robust to noise and multipath. These results demonstrate our model generalizes well to the uncovered scenarios.

Network depth: We evaluate the impact of the number of fully connected layers in our neural network. The deepest CNN we evaluate has 7 fully connected layers with 200, 100, 50, 50, 50, 50, 50 neurons, respectively. Each time we remove the last layer before the output and create a shallower CNN, until there is only one fully connected layer. As in Figure 18. the estimation error first reduces significantly and then tapers off. The result shows 5 fully connected layers are sufficient for our application.

Inference time: We run DRNet on a desktop with Ubuntu 16.04, Nvidia GTX 980 [28], and 6 GB VRAM. We also run it on Android 9, Snapdragon 845, and 6 GB RAM [33]. For the phone, we implement DRNet with Android NDK and uses Eigen [11] as matrix calculation library. The inference time for a single CNN is 0.36 ms on the desktop and 1.98 ms on the phone. The total time for 25-CNN ensemble learning is 7.1 ms on the desktop and

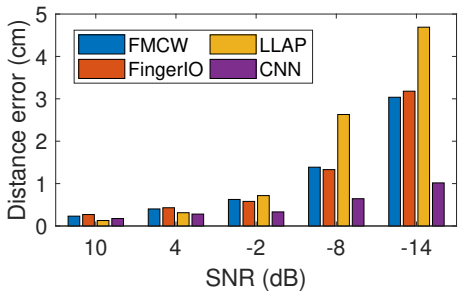


Fig. 19. The median ranging errors at various SNR.

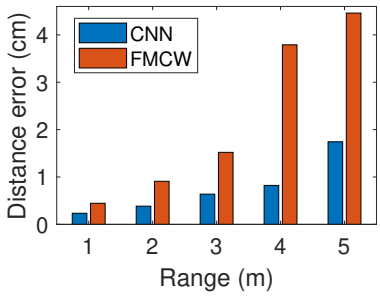


Fig. 20. The median ranging errors at various ranges.

36.5 ms on the phone. Since the transmission period is 40 ms, our approach can support real time ranging even on the phone. In comparison, the running time for FMCW is 0.55 ms on the desktop and 4.7 ms on the phone.

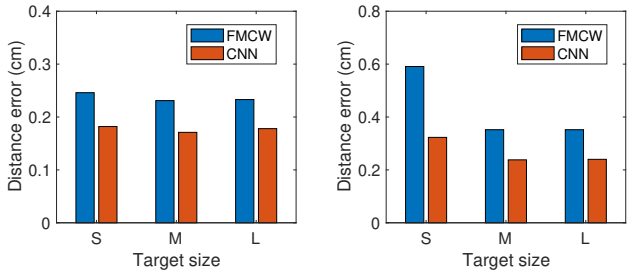
5.2 Overall Performance

In this part, we evaluate the overall performance of our approach using an ensemble of 25 CNNs. We compare our method with FMCW, FingerIO [27], and LLAP [44]. FingIO uses correlation and LLAP uses phase to estimate the distance. We use 18-20KHz OFDM signals in FingerIO, and send five sinusoids at 18, 18.5, 19, 19.5, and 20 KHz in LLAP. All approaches use the same experimental setup.

Impact of SNR: We evaluate the performance of our learning based ranging under different SNRs. For this purpose, we measure the SNR of our testing data and show the ranging performance for the data for given SNR values (allowing ±3 dB variance). As shown in Figure 19, when the SNR is around 10 dB, the phase-based approach (*i.e.*, LLAP) achieves the best performance – its median ranging error is 1.3 mm. It is followed by our approach with 1.7 mm median error, while FMCW and FingerIO have errors larger than 2 mm. As the SNR reduces, the performance advantage of our learning based approach becomes more significant. At SNR of -14 dB, our approach reduces the median ranging error by a factor of 3 over FMCW and FingerIO. LLAP has the worst ranging accuracy in this case, indicating that phase-based approaches are most sensitive to noise.

Impact of range: We evaluate our approach under long distances. In this case, the propagation delay of reflected signals can be large (*e.g.*, 29 ms for a target at 5 m away). If we use the signals aligning with our transmission period for distance estimation, the target reflection is only present in the last few milliseconds of the signals. This has a negative impact on estimation accuracy. Instead, we choose a 40 ms window roughly aligning with the target reflection, and use the signals in this window for distance estimation. The delay estimated this way starts from the beginning of a selected window. We get the propagation delay by adding the offset between the starts of the transmission period and the selected window. Note that the rough knowledge about propagation delay of the target reflection is obtained by correlating with transmitted signals and detecting the second peak since the first peak is the direct transmission from speaker to microphone). We use the same correlation approach in all schemes for fair comparison. The results are shown in Figure 20. As we can see, the distance estimation error of our approach is still within 1 cm at 4 m, while FMCW has the error close to 4 cm in this case. This experiment indicates that our approach has a larger working range.

Target size: We evaluate the ranging errors for targets with different sizes: a 2 cm×2 cm cardboard (denoted by S), a 10 cm×10 cm one (denoted by M), and a 40 cm×40 cm one (denoted by L). We perform experiments under both high SNR and low SNR, where signals reflected by the 2 cm cardboard ranges between 0 and 10 dB SNR. A large cardboard tends to reflect more signals and yields a higher SNR. However, as the cardboard gets even larger, its gain becomes marginal since the regions far away from the perpendicular reflection point reflect little energy



(a) High SNR (b) Low SNR

Fig. 21. The ranging errors for target objects with different sizes

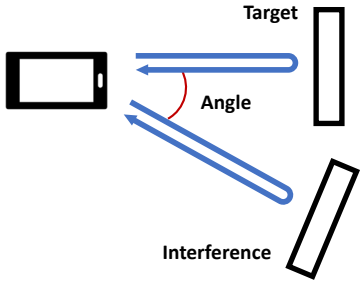


Fig. 22. The interference at different angle.

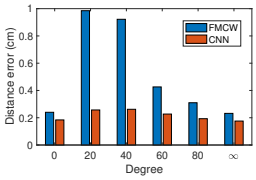


Fig. 23. The interfering object at different angles.

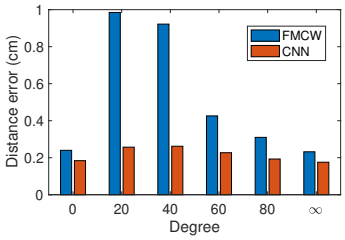


Fig. 24. The median ranging errors with interference at various distance.

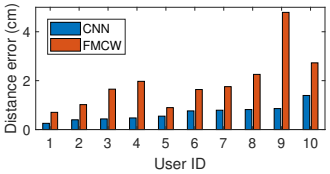
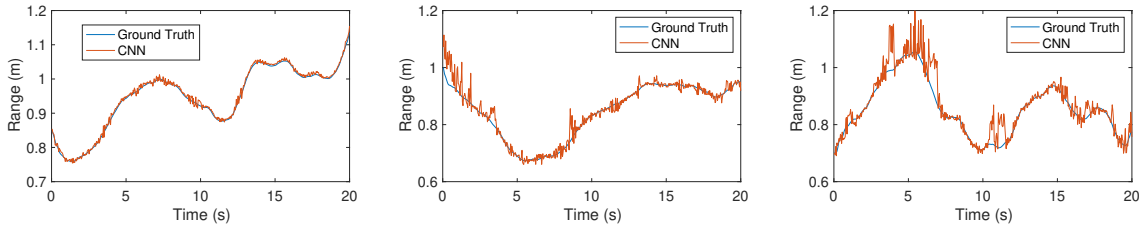


Fig. 25. The median ranging performance for different users.

back to the microphone. We find the signals reflected by 10 cm cardboard are 6–7 dB stronger than those for 2 cm one. However, the SNRs for 10 cm cardboard and 40 cm one are similar. When the SNR is sufficiently high, the accuracy is comparable across all reflectors as shown in Figure 21(a). When the SNR is low, larger targets have higher SNR than smaller targets and experience smaller estimation errors as shown in Figure 21(b).

Impact of interference angle: We evaluate the impact when an interfering object (a cardboard with the same size as the target) is placed at different angles, as shown in Figure 22. The interfering object is 10 cm farther away from the mobile than the target so that the target is always the first reflector. The results are illustrated in Figure 23. For comparison, we show the ranging performance without interfering objects (denoted by ∞). As we can see, when the angle is zero, the interfering object has no impact on ranging because its reflection has been occluded by the target. When the interfering object is placed at other angles, the impact of interference varies significantly because the speaker radiates different portions of energy across different directions, which can be characterized by the speaker's radiation pattern. For our speaker, the interference is maximized at 20 degrees while minimized at 80 degrees. Therefore, we use 20 degrees as the default angle for interference experiments in this paper.

Interference distance: We evaluate the ranging performance under interference at different distances. For this purpose, we place the interfering object at a 20-degree angle from the target and 2.5-12 cm farther away from the mobile. Figure 24 shows the distance estimation errors with various separation between the target and interference reflection, where ∞ indicates no interference. Our approach achieves the best performance under multipath. It reduces the ranging errors by a factor of 4.4 when the interference is 7.5 cm away, which is the most challenging cases. LLAP has the largest estimation errors under interference. We see that the errors first increase and then decrease, and reach the maximum when the interference reflection is 5–7 cm away from the target. This



(a) top 20% (b) median (c) bottom 20%

Fig. 26. The sample user traces with different performance.

is because if two reflections have a similar distance, the interfering object does not affect the distance estimation. If the interference reflection is well separated from the target, they can be easily differentiated.

5.3 User Study

In this experiment, we use our ranging approach to track the distance between the user's hand and a smartphone, as shown in Figure 1(a). This is a key building block in the motion tracking. We conduct the experiment with 10 users, including 4 women and 6 men. Their ages are between 20s and 50s. For each user, we let him or her stand 1.2 meters in front of the phone, raise the hand to roughly the same height as the phone, and move the hand back and forth in an arbitrary pattern. The details of trace collection are described in Section 4.

Tracking samples: To provide intuition about how our approach performs for hand tracking, we show the raw traces with different performance. For this purpose, we sort all the traces based on the median ranging errors and plot the traces ranked at 20%, 50%, and 80%, as shown in Figure 26. The traces for our approach are directly generated from CNN outputs without any additional filtering. The median ranging errors of selected traces are 0.3 cm, 0.7 cm, and 1.2 cm, respectively.

Performance for different users: Figure 25 and 27 show the median ranging errors and CDF across different users, respectively. We rank the users according to their median errors. For comparison, we also show the performance using FMCW. We see that our approach out-performs FMCW for all users, and the median errors are reduced by a factor of 1.6x - 5.6x. Moreover, we observe that there is a large performance variation across different users. User 1 achieves 0.255 cm median error and 0.5 cm 80-percent error, while User 10 has 1.39 cm median error and 3.4 cm 80-percent error. The performance variation mainly comes from different hand inclination and body posture. When the user's hand is not perpendicular to the signal propagation path, (*e.g.*, the inclination angle α in Figure 28 is less than 90 degrees), most of the reflection propagates downward, instead of returning to the phone, as shown in the figure. Different users use different inclination angles. The smaller inclination angle reduces SNR. In addition, the body posture affects the interference. When the separation between the body and hand is small and/or the area of the body facing the phone is larger, there is stronger interference. Since our approach is more robust to low SNR and strong interference, its performance benefit is higher in these traces.

Performance under practical situations: Furthermore, we evaluate our approach under the following practical situations:

- There is another person near the target user. We consider two cases. In the first case, the second person stands around the position A as shown in Figure 29 and performs semi-static activities like playing games on a mobile phone. In the second case, the second person walks roughly along the trajectory B shown in Figure 29.

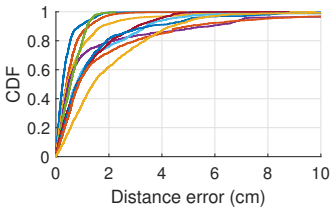


Fig. 27. The CDF of ranging performance for different users.

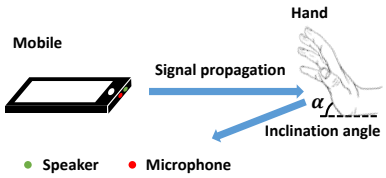


Fig. 28. The side view of the user's hand.

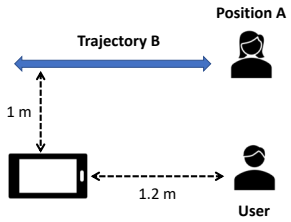
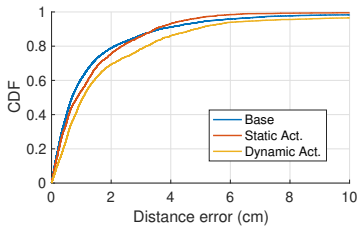
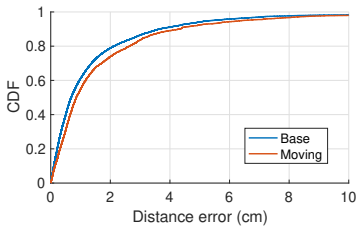


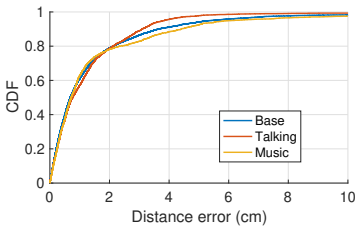
Fig. 29. The setup with human interference.



(a) Presence of second person



(b) Moving user



(c) Ambient sounds

Fig. 30. The ranging performance under various practical situations.

- The user is allowed to walk back and forth towards the mobile. In this case, the tracked motion is the net effect of the user walking and the hand movement with respect to the user body.
- There are ambient sounds during the experiments. We consider two common types of sounds: music and voice. We set the music to the same volume as that of inaudible tracking signals, while the loudness of voice is the same as in normal conversion (*i.e.*, around 60 dB). The sound source is at 1.5 m away from the mobile.

Figure 30(a) compares the tracking accuracy with and without the second person, where the latter is denoted as "Baseline". We observe that the second person with semi-static activities has little impact on the tracking performance, while the one with dynamic activities (*e.g.*, walking around) affects the ranging performance and the median error increases from 0.7 cm to 1.0 cm. This is expected since dynamic interference is harder to remove. Moreover, the second person may be temporarily closer than the user's hand. Under such challenging situations, our approach can still provide reasonable tracking performance with 1 cm median errors.

Figure 30(b) compares the performance with a semi-static and moving second user. As we can observe, our approach is fairly robust under the moving user. The median ranging error is 0.8 cm.

Figure 30(c) compares the performance with and without ambient sounds, where the latter is denoted as Baseline. The ambient sounds, such as music and voice, has little impact on the accuracy since our tracking signals are at 18–20 KHz band while common ambient sounds like music and voice have little energy in such high frequency band.

6 DISCUSSION

In this section, we discuss the overheads and limitations of DeepRange and speculative ideas for DRNet .

6.1 Overhead

Deployment Overhead: DeepRange can be deployed on off-the-shelf devices, such as mobile phones and VR/AR headset. It uses its internal speakers and microphones to transmit and receive acoustic signals in real-time.

Learning Overhead: The major overhead in machine learning usually is collecting training data. Since we can train DeepRange using synthetic data after capturing noise, interference, and frequency response, we remove the major bottleneck in ML. Training can be done offline. Mobile devices use the pre-trained model to estimate the range in an online manner.

Inference Overhead: DeepRange takes 36.5 ms inference time on mobile phones, which is shorter than the chirp duration. This is sufficient to achieve real-time ranging.

Collecting Background Reflections: DeepRange requires to measure background reflections for interference cancellation. Our evaluation uses 1 second received signals, which include 25 reflected chirps, to estimate background reflections. We also try measuring for a shorter measurement period – 0.4 s and observe similar performance.

6.2 Limitations

DeepRange works well if the mobile phone stops and stays static for a short period to measure the range to the target. Many use cases can fit the stop and measure model. For example, the mobile phone pauses to estimate the range. DeepRange does not support continuous movement scenarios, and we leave that for future work.

DeepRange assumes the target is the closest object. This assumption has been widely used and holds for many scenarios such as VR/AR, gesture recognition, etc.

6.3 DRNet Interpretation

[53–55] attempt to visualize intermediate feature layers to interpret knowledge representations. They find that CNN layers play the role of spatial filters to extract texture-level and part-level semantics. DRNet shows a similar mechanism that CNN channels extract finer granularity features. Considering the scenario without interference and noise, a small bandwidth FMCW chirp is able to estimate the range. Using wider bandwidth is the heuristic aggregation of each small band for robustness and high resolution in complex scenarios. DRNet has an implicit aggregation as well. It performs spatial filtering on received signals by non-uniform FMCW-like channels. It is a weighted correlation whose weights are learned by synthetic data with interference, noise and frequency response. Four channels provide even more diverse spatial filters so that it has a larger capability to estimate the range for different scenarios. We believe that other signal processing algorithms based on pattern recognition have the potential to improve performance with deep learning.

7 RELATED WORK

We classify the related work into (i) acoustic motion tracking, (ii) RF motion tracking, and (iii) neural network based motion tracking.

7.1 Acoustic Motion Tracking

A number of systems have been developed for motion tracking using acoustic signals. BeepBeep [30] measures the distance between two mobiles by correlating the pseudo-random sequence. Based on BeepBeep, SwordFight [56] further improves the efficiency and supports the interaction for mobile motion games. ApneaApp [26] uses FMCW to measure the chest and abdomen movements for apnea detection. AAMouse [51] relies on Doppler shifts of the signals to capture the distance changes over time. CAT [22] develops a distributed FMCW to estimate the distance between separated speakers and microphones. LLAP [44] leverages the phase changes of raw acoustic

signals to determine the distance changes, while Strata [52] and VSkin [36] uses the phases of channel taps. FingerIO [27] estimates the distance based on correlation and uses the properties of OFDM symbols to refine the estimation. DroneTrack [24] develops an approach to estimate the distance between a drone and a user based on MUSIC algorithm. MilliSonic [41] combines FMCW and phase measurements to estimate the distance. It achieves impressive estimation accuracy at high SNR, but does not work well under low SNR because it relies on the assumption that the estimation error is always less than the wavelength (*e.g.*, about 2 cm for 17 KHz acoustic signals) to avoid the phase ambiguity. However, this assumption cannot hold under low SNR. These algorithms are developed by human experts. This paper complements the existing work by exploring the possibility of applying DNN to raw acoustic signals for distance estimation. The insights gained from DNN can potentially help improve the existing signal processing methods.

7.2 RF Based Motion Tracking

Many motion tracking systems use radio frequency signals, such as WiFi, RFID, and millimeter-wave transceivers. ArrayTrack [49] estimates the angle-of-arrivals to different access points based on an array of antennas. RF-IDraw [43] uses the phase difference between a pair of RFID tags to estimate the incoming angles of the signals. Tagoram [50] estimates the locations of RFID tags by generating holograms with an array of RFID readers. TurboTrack [19] exploits the physical properties of RFID tags to emulate large bandwidth for accurate motion tracking. MTrack [46] measures the phases of 60 GHz waves for motion tracking with highly directional and steerable antennas. WiTrack [2, 3] leverages FMCW sweeping from 5.5 GHz to 7.2 GHz to estimate the positions of multiple people in the room. The above approaches require access to raw signals. When raw signals are not available, channel state information (CSI) is used to infer the position of a target. CUPID [34] and Splicer [48] derive the power delay profiles for the paths from the transmitter to the receiver by applying IFFT on CSI measurements. Widar [32] constructs the model between CSI and target motion and uses it for tracking. SpotFi [15] applies 2D MUSIC to jointly estimate the distance and angle-of-arrival of a target based on CSI measurements. Chronos [40] combines the CSI measurements at different bands to improve the tracking accuracy. WiDeo [12] determines the propagation delays of all paths by finding the best match between the CSI calculated according to these paths and measured values. WiDraw [37] uses CSI to estimate the angle-of-arrival of a target and relies on multiple WiFi APs for localization. The CSI-based approaches work on commodity devices (*e.g.*, WiFi APs and smartphones) and do not require any special hardware (*e.g.*, RFID tags and steerable antennas). However, these approaches only achieve decimeter level tracking accuracy, which is insufficient for fine-grained tracking applications.

7.3 Neural Network Based Tracking

RF-Echo [6] applies a neural network with a single hidden layer to estimate the propagation delay of RF signals based on the correlation profile. RF-Pose [57] develops a convolutional neural network to estimate a user's poses based on heat maps generated by RF signals. RF-Pose3D [58] further extends above approaches to 3D pose estimation. Other works [4, 18] apply the recurrent neural network to determine a user's indoor location using the received signal strength of RF signals. RF-Finger [42] identifies the multi-touch gestures by applying CNN. WordRecorder [7] extracts the spectrogram feature from acoustic signals and combines with CNN for handwriting classification. A multi-LSTM neural network is designed in [17] to fingerprint mobile device sensor, instead of using handcrafted features. RTrack [23] develops an RNN to automatically learn the mapping between the 2D profile and target position to exploit the temporal locality. All these works need features extracted from the received signals before applying neural networks, while our network directly takes the received signals as the input. Using the raw signals is beneficial to achieve better performance as demonstrated by our experiments. In addition, these approaches require collecting lots of data to train the networks, which can be time-consuming and labor-intensive.

Recently, there have been many CNN-based object detectors improving the accuracy of detecting sonar images. [47] first implements CNN over synthetic aperture sonar image and argument data by mirroring mugshots. [13] makes use of the efficient YOLO model on forward-looking sonar images for real-time detection. [59] extracts target features by AlexNet and classify objects by applying SVM to side-scan sonar images. [45] further proposes an adaptive weights CNN to fuse the generated weights of the deep belief network and normalize the adaptive weights by local response normalization. They directly apply detectors for optical images to sonar images and ignore the inherent differentiation. [20] designs a Noise Adversarial Network as the sideway network to introduce perturbation with specific noise to sonar images during training to generalize the object detector in sonar images. These works apply deep learning to images derived from processing acoustic signals and hence are more similar to image processing work. In comparison, our works directly feed raw acoustic signals to DNN. This is more challenging but can achieve higher gain since post-processing using existing signal processing methods may already reduce accuracy, which can be hard to recover at the later stage. Our results demonstrate our approach outperforms the approaches that apply DNN to outputs from signal processing stages.

8 CONCLUSION

In this paper, we develop a deep learning based ranging. To eliminate the need of collecting large volumes of training data, we generate synthetic signals by incorporating important factors in real environments, such as noise, multipath, self interference, and transceiver frequency response. We develop a DNN that uses filters with long kernel sizes to detect signal patterns and applies the ensemble method to enhance the estimation accuracy. We evaluate DRNet on real data collected using 11 phones across 4 locations, and show it achieves significant performance gain over FMCW.

ACKNOWLEDGMENTS

This work is supported in part by NSF Grant CNS-1718585 and CNS-2032125. We are grateful to anonymous reviewers for their insightful comments and suggestions.

REFERENCES

[1] Adam Paszke, Sam Gross, Soumith Chintala and Gregory Chanan 2019. PyTorch. https://pytorch.org/.
[2] Fadel Adib, Zachary Kabelac, and Dina Katabi. 2015. Multi-person localization via RF body reflections. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*. 279–292.
[3] Fadel Adib, Zach Kabelac, Dina Katabi, and Rob Miller. 2014. WiTrack: Motion Tracking via Radio Reflections off the Body. In *Proc. of NSDI*.
[4] Luis A Castro and Jesus Favela. 2005. Continuous tracking of user location in WLANs using recurrent neural networks. In *null*. IEEE, 174–181.
[5] Jung Il Choi, Mayank Jain, Kannan Srinivasan, Phil Levis, and Sachin Katti. 2010. Achieving single channel, full duplex wireless communication. In *Proceedings of the sixteenth annual international conference on Mobile computing and networking*. ACM, 1–12.
[6] Li-Xuan Chuo, Zhihong Luo, Dennis Sylvester, David Blaauw, and Hun-Seok Kim. 2017. RF-Echo: A Non-Line-of-Sight Indoor Localization System Using a Low-Power Active RF Reflector ASIC Tag. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*. ACM, 222–234.
[7] Haishi Du, Ping Li, Hao Zhou, Wei Gong, Gan Luo, and Panlong Yang. 2018. Wordrecorder: Accurate acoustic-based handwriting recognition using deep learning. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1448–1456.
[8] Gated frequency response measurement [n.d.]. Making gated-impulse frequency measurements using ARTA. https://pearl-hifi.com/06_Lit_Archive/15_Mfrs_Publications/ARTA%201.7/FR%20_Measurement_Using_ARTA.pdf.
[9] Lars Kai Hansen and Peter Salamon. 1990. Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence* 12, 10 (1990), 993–1001.
[10] Intel 2018. Intel RealSense D415 Camera. https://ark.intel.com/content/www/us/en/ark/products/128256/intel-realsense-depth-camera-d415.html.
[11] Benoit Jacob and Gael Guennebaud. 2019. Eigen. http://eigen.tuxfamily.org/index.php?title=Main_Page.

[12] Kiran Joshi, Dinesh Bharadia, Manikanta Kotaru, and Sachin Katti. 2015. WiDeo: Fine-grained Device-free Motion Tracing using RF Backscatter. In *Proc. of NSDI*.

[13] Juhwan Kim and Son-Cheol Yu. 2016. Convolutional neural network-based real-time ROV detection using forward-looking sonar image. In *2016 IEEE/OES Autonomous Underwater Vehicles (AUV)*. IEEE, 396–400.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. Spotfi: Decimeter level localization using WiFi. In *ACM SIGCOMM Computer Communication Review*, Vol. 45(4). ACM, 269–282.

[16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *neural information processing systems* 141, 5 (2012), 1097–1105.

[17] Xiang-Yang Li, Huiqi Liu, Lan Zhang, Zhenan Wu, Yaochen Xie, Ge Chen, Chunxiao Wan, and Zhongwei Liang. 2019. Finding the stars in the fireworks: Deep understanding of motion sensor fingerprint. *IEEE/ACM Transactions on Networking* 27, 5 (2019), 1945–1958.

[18] Yuan Lukito and Antonius Rachmat Chrismanto. 2017. Recurrent neural networks model for WiFi-based indoor positioning system. In *Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS), 2017 International Conference on*. IEEE, 121–125.

[19] Zhihong Luo, Qiping Zhang, Yunfei Ma, Manish Singh, and Fadel Adib. 2019. 3D backscatter localization for fine-grained robotics. In *16th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 19)*. 765–782.

[20] Qixiang Ma, Longyu Jiang, Wenxue Yu, Rui Jin, Zhixiang Wu, and Fangjin Xu. 2020. Training with Noise Adversarial Network: A Generalization Method for Object Detection on Sonar Image. In *The IEEE Winter Conference on Applications of Computer Vision*. 729–738.

[21] Yunfei Ma, Nicholas Selby, and Fadel Adib. 2017. Minding the Billions: Ultra-wideband Localization for Deployed RFIDs. In *Proc. of ACM MobiCom*.

[22] Wenguang Mao, Jian He, and Lili Qiu. 2016. CAT: High-Precision Acoustic Motion Tracking. In *Proc. of ACM MobiCom*.

[23] Wenguang Mao, Mei Wang, Wei Sun, Lili Qiu, Swadhin Pradhan, and Yi-Chao Chen. 2019. RNN-Based Room Scale Hand Motion Tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*. 1–16.

[24] Wenguang Mao, Zaiwei Zhang, Lili Qiu, Jian He, Yuchen Cui, and Sangki Yun. 2017. Indoor Follow Me Drone. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 345–358.

[25] Rajalakshmi Nandakumar, Krishna Kant Chintalapudi, and Venkata N. Padmanabhan. 2013. Dhwani : Secure Peer-to-Peer Acoustic NFC. In *Proc. of ACM SIGCOMM*.

[26] Rajalakshmi Nandakumar, Shyam Gollakota, and Nathaniel Watson. 2015. Contactless Sleep Apnea Detection on Smartphones. In *Proc. of ACM MobiSys*.

[27] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. 2016. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proc. of ACM CHI*. 1515–1525.

[28] Nvidia 2015. Nvidia GTX 980. https://www.geforce.com/hardware/desktop-gpus/geforce-gtx-980/specifications.

[29] David Opitz and Richard Maclin. 1999. Popular ensemble methods: An empirical study. *Journal of artificial intelligence research* 11 (1999), 169–198.

[30] Chunyi Peng, Guobin Shen, Yongguang Zhang, Yanlin Li, and Kun Tan. 2007. BeepBeep: A High Accuracy Acoustic Ranging System using COTS Mobile Devices. In *Proc. of ACM SenSys*.

[31] Swadhin Pradhan, Wei Sun, Ghufran Baig, and Lili Qiu. 2019. Combating replay attacks against voice assistants. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 3, 3 (2019), 1–26.

[32] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. 2018. Widar2. 0: Passive human tracking with a single wi-fi link. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 350–361.

[33] Qualcomm Inc. 2018. Snapdragon 845. https://www.qualcomm.com/products/snapdragon-845-mobile-platform.

[34] Souvik Sen, Jeongkeun Lee, Kyu-Han Kim, and Paul Congdon. 2013. Avoiding multipath to revive inbuilding WiFi localization. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 249–262.

[35] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. *international conference on learning representations* (2015).

[36] Ke Sun, Ting Zhao, Wei Wang, and Lei Xie. 2018. Vskin: Sensing touch gestures on surfaces of mobile devices using acoustic signals. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 591–605.

[37] L. Sun, S. Sen, D. Koutsonikolas, and K. Kim. 2015. WiDraw: Enabling Hands-free Drawing in the Air on Commodity WiFi Devices. In *Proc. of ACM MobiCom*.

[38] Yoiti Suzuki and Hisashi Takeshima. 2004. Equal-loudness-level contours for pure tones. *The Journal of the Acoustical Society of America* 116, 2 (2004), 918–933.

[39] David Tse and Pramod Viswanath. 2005. *Fundamentals of wireless communication*. Cambridge university press.

[40] Deepak Vasisht, Swarun Kumar, and Dina Katabi. 2016. Decimeter-level localization with a single WiFi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. 165–178.

[41] Anran Wang and Shyamnath Gollakota. 2019. MilliSonic: Pushing the Limits of Acoustic Motion Tracking. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, 18.

[42] Chuyu Wang, Jian Liu, Yingying Chen, Hongbo Liu, Lei Xie, Wei Wang, Bingbing He, and Sanglu Lu. 2018. Multi-touch in the air: Device-free finger tracking and gesture recognition via cots rfid. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 1691–1699.

[43] Jue Wang, Deepak Vasisht, and Dina Katabi. 2014. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals. In *Proc. of ACM SIGCOMM*.

[44] Wei Wang, Alex X Liu, and Ke Sun. 2016. Device-free gesture tracking using acoustic signals. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*. ACM, 82–94.

[45] Xingmei Wang, Jia Jiao, Jingwei Yin, Wensheng Zhao, Xiao Han, and Boxuan Sun. 2019. Underwater sonar image classification using adaptive weights convolutional neural network. *Applied Acoustics* 146 (2019), 145–154.

[46] Teng Wei and Xinyu Zhang. 2015. mTrack: High Precision Passive Tracking Using Millimeter Wave Radios. In *Proc. of ACM MobiCom*.

[47] David P Williams. 2016. Underwater target classification in synthetic aperture sonar imagery using deep convolutional neural networks. In *2016 23rd international conference on pattern recognition (ICPR)*. IEEE, 2497–2502.

[48] Yaxiong Xie, Zhenjiang Li, and Mo Li. 2015. Precise power delay profiling with commodity WiFi. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 53–64.

[49] Jie Xiong and Kyle Jamieson. 2013. Arraytrack: A fine-grained indoor location system. In *Presented as part of the 10th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 13)*. 71–84.

[50] Lei Yang, Yekui Chen, Xiang-Yang Li, Chaowei Xiao, Mo Li, and Yunhao Liu. 2014. Tagoram: Real-Time Tracking of Mobile RFID Tags to High Precision Using COTS Devices. In *Proc. of ACM MobiCom*.

[51] Sangki Yun, Yi chao Chen, and Lili Qiu. 2015. Turning a Mobile Device into a Mouse in the Air. In *Proc. of ACM MobiSys*.

[52] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. 2017. Strata: Fine-Grained Acoustic-based Device-Free Tracking. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM, 15–28.

[53] Matthew D Zeiler and Rob Fergus. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*. Springer, 818–833.

[54] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. 2018. Interpretable convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 8827–8836.

[55] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. 2019. Interpreting cnns via decision trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6261–6270.

[56] Zengbin Zhang, David Chu, Xiaomeng Chen, and Thomas Moscibroda. 2012. SwordFight: Enabling a New Class of Phone-to-Phone Action Games on Commodity Phones. In *Proc. of ACM MobiSys*.

[57] Mingmin Zhao, Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, and Dina Katabi. 2018. Through-wall human pose estimation using radio signals. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7356–7365.

[58] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. 2018. RF-based 3D skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. ACM, 267–281.

[59] Pingping Zhu, Jason Isaacs, Bo Fu, and Silvia Ferrari. 2017. Deep learning feature extraction for target recognition and classification in underwater sonar images. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE, 2724–2731.