

WiFi-based Contactless Gesture Recognition Using Lightweight CNN

Keegan Kresge*, Sophia Martino[†], Tianming Zhao[‡], and Yan Wang[‡]

*Rochester Institute of Technology, Rochester NY, USA 14623

[†]Lehigh University, Bethlehem, PA, USA 18015

[‡]Temple University, Philadelphia, PA, USA 19122

Email: kjk6690@rit.edu, som324@lehigh.edu, {tum94362,y.wang}@temple.edu

Abstract—Gesture recognition has the potential to become a part of contactless interactions with devices to improve accessibility and ease with applications. As the presence of portable devices remains standard, WiFi will continue to constantly connect these devices. Leveraging this availability, instead of relying on installing special sensors, ubiquitous WiFi sensing devices can decipher motion, thus mitigating additional costs. We develop a low-cost hand gesture recognition system utilizing Channel State Information (CSI) from a few subcarriers in prevalent WiFi signals. This information is sent through a lightweight signal segmentation algorithm and Convolutional Neural Network (CNN) that learns the gestures and successfully distinguishes them. Computationally demanding feature extraction is avoided as it increases processing time and does not scale well with additional gestures. Our model obtains an 96% accuracy rate across three different gestures on average.

I. INTRODUCTION

The ever-growing advancement of mobile technology has enabled new human-computing interfaces. We envision a future with more contactless interactions with devices. This technology can increase convenience and accessibility to home and mobile applications. For example, a gesture command can function like a voice command, thus extending technology assistance to the deaf community.

Wearable and portable devices have enabled the advancement of gesture recognition [1]–[4]. They have used built-in sensors to identify body movements and gestures. However, these sensor-based approaches are potentially intrusive and have explicit and implicit installation costs. Vision-based techniques [5]–[9] can accomplish gesture recognition, but with limitations. This technology requires the gesture to be in the line of sight (LOS) of a dedicated camera.

Recently, researchers have utilized the prevalent WiFi signals to perform non-intrusive activity recognition [10]–[15]. Fine-grained channel state information (CSI) of WiFi signals have been exploited to perform fine-grained sensing in various applications, including daily activity tracking, human counting, and gesture recognition. However, these works usually adopt traditional machine learning methods with a feature extraction procedure, which do not generalize well and are time-consuming. These methods are not practical for deploying on hardware limited mobile devices.

In this work, we implement a practical non-intrusive gesture recognition system using low-cost WiFi signals. We utilize CSI

from commodity WiFi devices (e.g., laptops and smartphones) to enable fine-grained gesture recognition. The insight is that hand gestures change the propagation paths of WiFi signals, which are captured in CSI. To extract gesture-related information from CSI and use it for gesture recognition in practice, we face two major challenges: First, we need to accurately determine CSI data for gesture recognition. Since CSI could be affected by any body movement, it is not efficient or robust to track and examine all CSI changes for gesture recognition. Second, it is known that CSI is susceptible to environmental noises and interference, which may impact the CSI signatures.

To address these challenges, we first process the CSI with denoising techniques. Next, we propose a preamble gesture that generates unique CSI patterns to help accurately determine the starting and ending of a given gesture. We implement a low-cost segmentation method using the short time energy (STE) of CSI moving variance and Dynamic Time Warping (DTW) [16] to accurately determine the data segment containing the target gesture. The power of a neural network helps us to distinguish patterns in the CSI works to reliably identify a gesture. We develop a lightweight CNN to ensure high accuracy and robustness of our gesture recognition system in practical environments with noises and interference. As opposed to creating profiles and matching patterns to known gestures, deep learning has greater potential to include a broader range of gestures in future work. We list our contributions in this paper as follows:

- We design a unique preamble gestures to help achieve accurate and robust gesture recognition performance.
- Our system deploys a low-cost data segmentation method using Dynamic Time Warping (DTW) and Short Time Energy (STE) on the moving variance of WiFi signals.
- Our robust hand gesture classifier differentiates hand gestures in practical environments without feature extraction, leveraging lightweight deep learning technology.
- We conduct extensive experiments with three hand gestures that can be used for simple commands. Results show that our system can achieve over 96% accuracy.
- Our null gesture detection system distinguishes pretrained gestures from others with 84% accuracy.

The rest of paper is organized as follows. Section II begins

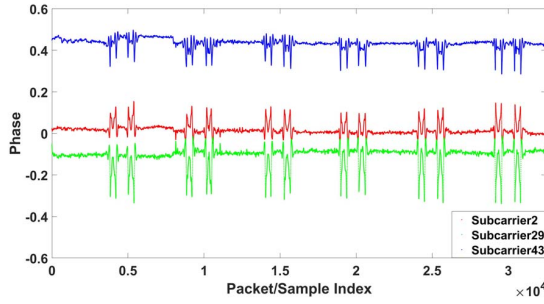


Fig. 1: CSI phase collected when performing a designated preamble gesture. A significant and repeatable pattern is seen in subcarriers 2, 29 and 43.

with an extensive review of related work in gesture recognition and considers the uniqueness and advantage of our system that can bring into this research field. Section III describes feasibility studies which are used as the basis for our system. Section IV introduces an overview of the design and process flow of our system. Section V introduces our data segmentation approach and deep learning model design. Section VI presents our experimental methodology and results of evaluating this system. Section VII discusses various aspects in this system and our future work plans. Section VIII concludes this work with a discussion.

II. RELATED WORK

Current techniques for gesture recognition can be roughly categorized into three categories: body sensor-based, vision-based, and RF-based recognition.

Body Sensor-Based. The customized wrist-worn sensing platforms are designed to capture hand gestures. For instance, Risq [1] leverages the inertial sensors on a wristband for monitoring smoking habits. Geng *et al.* [2] adopt the surface electromyography (sEMG) sensor worn on the user forearm to recognize finger gestures using deep learning models. Zhao *et al.* [4] explore the feasibility of using both photoplethysmography and motion sensors in wearables to improve the sign language gesture recognition accuracy when there are limited body movements. However, these solutions need extra hardware support, which are not commonly compatible with existing wearable devices.

Vision-Based. Many vision-based approaches have been developed to recognize hand/body gestures with the help of cameras. For example, Leap motion and kinect sensors have been used in previous work [5] to track hand motions for Virtual Reality applications. Lei *et al.* [7] leverage visual sensors for recognition of kitchen activities for kitchen assistance. Additionally, Yu *et al.* [9] use LED and photo detectors to detect simple hand motions such as pushing and circles. All these systems achieve robust results while accomplishing touchless Human Computer interactions. However, those vision-based system still require the user to use an additional device and operate within the LOS of the sensor.

RF-Based. Radio Frequency based approaches have become increasingly important due to the prevalent wireless

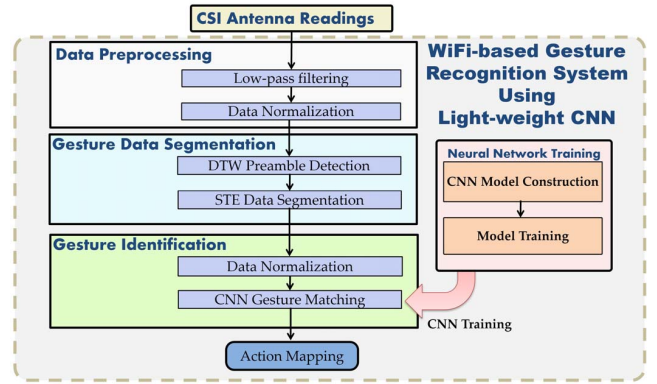


Fig. 2: Overview of the proposed WiFi-based gesture recognition system.

environments. Macks *et al.* [17] introduce three different RF sensors, including a 77 GHz FMCW radar, for recognizing sign language from both native and imitation ASL speakers. WiGest [10] leverages the Received Signal Strength of Wifi to recognize the hand gestures. WiFinger [13] examines the unique patterns exhibited in CSI of WiFi to identifies fine-grained movements of finger gestures. However, these Radio Frequency based approaches either require dedicated devices or can be easily affected by environmental changes such as people walking by.

This paper presents a low-cost, contactless human-computer interaction system using CSI. Leveraging existing commodity WiFi, our system requires no dedicated intrusive devices. Additionally, our system enables preamble gestures to accomplish accurate data segmentation in practice. Without relying on feature extraction, we develop a lightweight CNN that achieves robust gesture recognition using CSI of only several subcarriers. Compared to existing work, our system is more feasible to deploy on resource-limited mobile devices.

III. FEASIBILITY STUDY

In this section, we discuss the background of fine-grained CSI from WiFi signals, our design intuitions, and feasibility study.

A. Contactless Sensing Using CSI

Commodity WiFi is largely accessible and is continually connecting devices. We look to take advantage of the prevalence of WiFi and use it for contactless sensing and recognition. Traditional 802.11n WiFi uses Orthogonal Frequency Division Multiplexing (OFDM) transmission, which splits the information of a single packet into 56 subcarriers that measure fluctuations in the surrounding environment. The amplitude and phase of the CSI on each subcarrier provide rich information per packet about the environment where the signal traveled.

As the various WiFi signal frequencies, or subcarriers, propagate through the air, obstacles alter the signal phase and amplitude through reflection and refraction. Some approaches use the sum of the subcarriers, known as the Received Signal

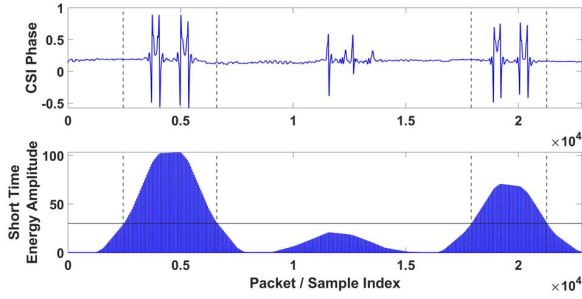


Fig. 3: Illustration of segmenting the CSI data of a Circle Y gesture between two preamble gestures using STE.

Strength (RSSI), as a coarse-grained method for determining gestures. In comparison, CSI has 56 subcarrier signals to interpret rather than a single signal strength to support more fine-grained sensing applications. Previous leveraging of WiFi [18] indicates that specific activities can make specific signatures on the CSI.

B. Feasibility Study

We conduct experiments asking a participant to make a wave motion between a pair of WiFi transmitter and receiver using Atheros Card QCA9590 at 1000 packets/sec. We extract CSI from 56 subcarriers using Atheros-CSI-Tool [19]. We find that among the subcarriers, some are more sensitive to gestures, having more significant variances in the amplitude and phase than other subcarriers. Figure 1 shows that hand waving motions (i.e., our designated preamble gesture) can generate recognizable patterns in the phase of subcarriers 2, 29, and 43, which encourage us to develop a system to recognize gestures using few subcarriers.

IV. CHALLENGES & SYSTEM DESIGN

A. Challenges

Developing a gesture recognition system with WiFi signals poses environmental obstacles. As we collect data on the third floor of a University building, we encounter noise and interference from the surrounding environment. This interference can arise from other 802.11n devices and environmental changes, such as a person walking around. A noisy signal may result in a coarse gesture that the neural network cannot discern. Moreover, the gestures are influenced by individual diversity, hand size, tempo, posture and gesture inconsistency. To mitigate these challenges, we test with a preamble that is unique and robust to noise for segmentation and apply denoising techniques to highlight gesture patterns.

B. System Overview

The basic idea of our system utilizes the patterns in the phase shifts of the CSI after a gesture is made between a transmitter and receiver. As outlined in Figure 2, we take the raw data and in our *Data Pre-Processing* step and prepare it for analysis. Because human activity does not occur at higher frequencies, the values of the phase are low-pass filtered

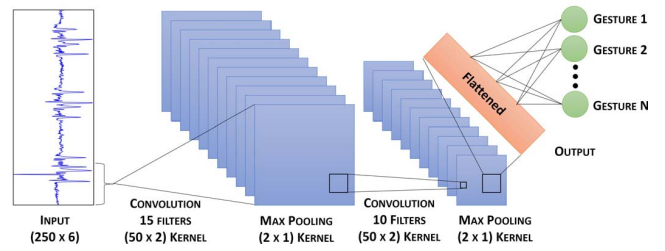


Fig. 4: Our CNN takes signal input through two convolutional layers, two max pooling layers, and one final dense layer and outputs the gesture it recognizes.

to remove any high frequency noise. Next, we implement a *Gesture Data Segmentation* step. We use the STE on the moving variance of certain subcarriers' phase signatures to threshold when our preamble occurs. If the STE on the moving variance is above a predetermined threshold, we apply DTW to ensure a preamble gesture. Once ensured, the gesture in between our preambles is fed to our *Gesture Identification* step. Within this step, we train and test a CNN to determine which gesture is performed. The model outputs probabilities for each gesture, and if none of them are above a threshold of certainty, then the gesture is defined as a null gesture. Otherwise, the application performs some function based on the predicted gesture.

V. GESTURE CLASSIFICATION

A. Gesture Design

The system is built with the intention of being implemented on mobile devices. Processing a neural network is costly, so to make the system lightweight enough for usage on a mobile device, we utilize a preamble gesture to minimize use of the neural network. The preamble is a unique starting and ending gesture that is used to notify the system to interpret the next gesture as a command. We choose the gesture of a wave as the preamble, because of its distinct peak in amplitude and phase, as seen in Figure 1. In designing the two other gestures, we work to develop easy-to-implement movements that can be assigned meaning in a mobile application: Circle X and Circle Y. In our proof of concept trials, we observed the unique signatures that these gestures leave on the CSI.

B. Data Pre-processing

High frequency noise is prevalent in the raw WiFi signal. To combat this we apply a second order Butterworth low pass filter to remove the aberrant noise above $20Hz$. Additionally, we normalize the data in preparation for segmentation.

C. Data Segmentation Using Moving Variance and Short-time Energy

We design a segmentation scheme to effectively segment all gesture-related CSI signals using STE on the moving variance of the phase. Our scheme uses a sliding window with the typical length of the gestures to traverse and calculate the moving variance of the signal with a certain step size. After calculating the moving variance, we adopt the same length

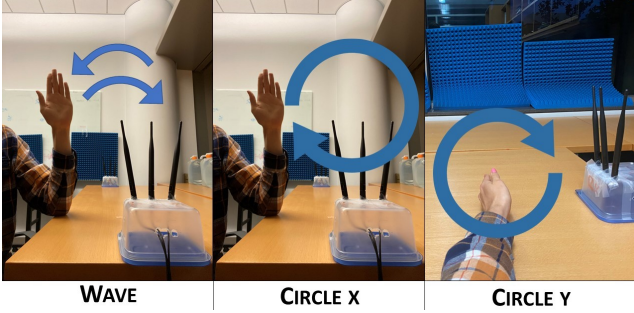


Fig. 5: Three gestures used in experiments: Wave, Circle X, and Circle Y.

of window and step size to traverse the calculated moving variance to get its short-time energy, which further amplifies the fluctuation for better segmentation. At last, we adopt a threshold-based method to find the starting and ending points of a gesture. Particularly, we find the intersections between the threshold and the short-time energy waveform. These intersection timestamps are later used for segmentation. An example of a gesture segmented using our method is outlined in Figure 3.

In this work, we use a preamble gesture to generate unique CSI pattern with significantly large fluctuations for accurate segmentation. The intuition is that the target gesture being recognized occasionally may not have sufficiently large fluctuations, however, the preamble always does. During our experiments, the user is required to perform the preambles before and after each target gesture. Since our segmentation scheme can locate both target gestures and preambles, we adopt the DTW to differentiate a preamble gesture from other CSI fluctuations. With adding certain guarding samples (e.g., 0.3 secs) between two adjacent preambles, we can always get the segments containing the target gestures in between from the previously found STE timestamps, which solves the issue of segmenting target gestures with insignificant CSI fluctuations.

Based on our experiments from three different gestures, we determine the length of the sliding window as 1000 and the step size 50 to ensure the accuracy and efficiency of our segmentation scheme. Note that the threshold is user-specific and needs to be dynamically determined by the maximum STE when there is no gesture detected. Figure 3 illustrates the STE corresponding to the Circle Y gesture between two preambles. We can clearly see that the intersections between the STE waveform and the threshold covers the entire gestures in CSI raw data, showing our algorithm segments the gestures in CSI signal accurately.

D. Light-Weight Gesture Recognition

Recent years have witnessed the success of Deep Neural Networks (DNNs) on Time Series Classification (TSC). Previous work [20] in identifying gestures from sensor readings inspire a DNN technique for WiFi gesture recognition. This work especially adopts a light-weight CNN with gesture-related CSI segments as the input for the CSI-based gesture

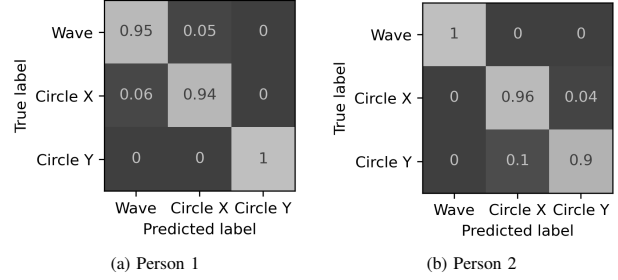


Fig. 6: Confusion Matrix for recognition of hand gestures across 3 separate gestures.

recognition. Our model avoids the feature extraction process, which does not generalize well for various gestures and is time consuming, especially when the model is deployed on mobile or IoT devices with limited resources.

Particularly, we design a CNN consisting of five layers including two convolutional layers, two max pooling layers, and one final dense layer. The convolutional layers have 15 and 10 filters respectively with kernel sizes of 50×2 and we use a ReLU activation function. Figure 4 shows that after each convolution, the data is downsized using a max pooling function over a window of 2×1 units. We add a dense layer with 3 neurons after the last convolution layer. In the end, we use a softmax layer to output the prediction probability results. It worth noting that we test different kernel size (e.g., 20×1 , 35×2 , 50×6) and different filter numbers (e.g., 10, 15, 20, 30). And we find the 50×2 kernel size and 15 & 10 filter numbers provide the best result in terms of the performance.

To train our CNN model, for each user we feed the CNN model with a user's segmented CSI data of N samples $\{(CSI_i, y_i)\}$, where CSI_i and y_i represent the segmented gesture-related CSI input data (i.e. the phase of subcarriers 2, 12, 22, 32, 42, and 52) and the corresponding label with respect to one specific gesture (i.e., $y_i = k$ represents that y_i is from the gesture k , $k = 1, \dots, K$). The specified subcarriers were chosen with the intention of getting a wide array of varying patterns among the 56 subcarriers, while also downsampling for computational efficiency. Particularly, we randomly select the specified number of the training data from the input data set of the current user and use the rest of the user's data as the testing data. In our implementation, we adopt 64 as the batch size and 15 as epoch.

It is possible that a gesture is made that is not one of the three predefined gestures. We call these undefined motions null gestures. In order to distinguish a null gesture from the others, we define a threshold value. If the CNN's maximum probability value across the three gestures are not above the threshold, it is deemed a null gesture, and is thus not passed onto the application for further interpretation.

VI. EXPERIMENT AND EVALUATION

We run experiments with two people to test out the variances between persons of the gesture shape and speed.

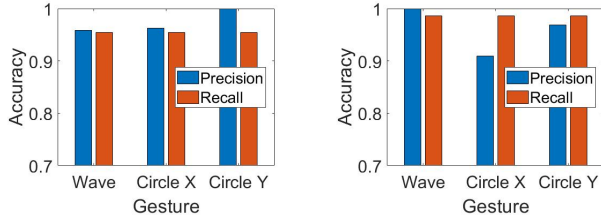


Fig. 7: Overall performance of different people.

A. Experimental Methodology

In a university office, we set up a receiver and a transmitter four feet apart. Using Atheros-CSI-Tool [19] with the Atheros QCA9590 chip, we extract the Channel State Information. The receiver has three antennas, but only one is used for data collection. Person 1 and person 2 perform the gestures about 6 inches from the receiver on the LOS. Each gesture is completed at least 40 times by each person to generate a training and a testing data set.

B. Data Collection.

We recruit 2 participants including 1 male and 1 female to perform hand gestures for evaluation. We focus on the typical hand gestures involving three movements of hand as shown in Figure 5. The participants are asked to respectively perform the three hand-level gestures for 50 repetitions each. In total, we collect 40 CSI segments for the experimental evaluation for each gesture. For null gesture identification testing, we recruited one participant to perform four common household movements for ten repetitions each: walking, sitting up/down, eating, and drinking. Unless mentioned otherwise, our results are derived using 66% of our data set for training and rest for testing. The data is processed by our system implemented by MATLAB and Python, which is run on an Inspiron 7573.

C. Evaluation Metrics

Precision. Given N_g segments of a gesture type g , precision of recognizing the gesture type g is defined as $Precision_g = N_g^T / (N_g^T + M_g^F)$, where N_g^T is the number of gesture segments correctly recognized as the gesture g , and M_g^F is the number of gesture segments corresponding to other gestures which are mistakenly recognized as the gesture type g .

Recall. Recall of the gesture type g is defined as the percentage of the segments that are correctly recognized as the gesture type g among all segments of the gesture type g , which is defined as $Recall_g = N_g^T / N_g$.

D. Gesture Recognition Performance

Figure 6 depicts the confusion matrix for the recognition of the three hand gestures across both participants. The diagonal entries show the average accuracy of recognizing each gesture, respectively. Specifically, the average precision is 95.8% among all three gestures. We observe that all participants have high accuracy on recognizing these hand gestures. Specifically,

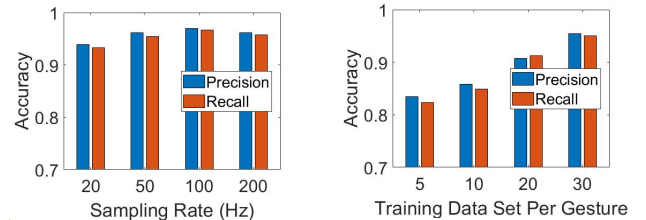


Fig. 8: Impact of sampling rate and training size on accuracy performance.

the average precision and recall of the participants are 96.6% and 97.0%. Overall, the results confirm that it is promising to use our lightweight CNN system with the CSI of commodity WiFi. With just phase information from 11% of the available subcarriers, we can accurately train and test our CNN to perform hand gesture recognition, while also being robust across different users.

Furthermore, to test our null gesture detection system, we gathered CSI data on household activities such as walking, sitting down, eating, and drinking. Our Null Gesture Detection Rate is the ratio of a gesture detected as the null gesture to all the gestures of a type (e.g., a target gesture) being detected. We were able to correctly distinguish null gestures 84% of the time with a threshold of 80%. Since this method requires no additional training for our model, it is promising for real life implementation.

E. Various Impacts on the System

Impact of Different Users. Figure 7 presents the average precision and recall of recognizing three hand gesture across different participants. We observe that all participants have high accuracy on recognizing these hand gestures. Specifically, the average precision and recall of the participants are 96.6% and 97.0%. The results show the robustness and scalability of our proposed system across different users, and demonstrate the robustness that will be integral for implementation in mobile devices.

Impact of Sampling Rate. The sampling rate of sensing hardware is one of the critical impact factors on affecting the power consumption. In this study, we collect the CSI readings with sampling rates of 20Hz, 50Hz, 100Hz, and 200Hz to evaluate the system. Figure 8 (a) shows the average precision and recall of the gesture recognition under different sampling rates. We find that the precision/recall maintain as high as 93% even under the lowest sampling rate (i.e., 20Hz). As the results imply, our system can operate normally on lightweight hardware because lower sampling rates have negligible impacts on performance.

Impact of Training Data Size. We change the percentage of data used for training to study the performance of our system under different training data size as shown in Figure 8 (b). In particular, we choose 5, 10, 20, and 30 gesture segments with respect to each gesture for training, and use the rest of our data for testing. We observe that our system can achieve an

average precision and recall of 90.76% and 91.3% respectfully for recognizing three hand gestures using only 20 segments of each gesture for training. These results indicate our system can achieve good recognition performance with a limited set of training data, which ensures great convenience for practical usage.

VII. DISCUSSION AND FUTURE WORK

Recognizable Gestures. The CSI has the potential to recognize the arm-level, hand-level, and even finger-level gestures. We look to expand to more gestures (e.g., volumes up or down, searching, and even spelling in ASL) in the experiments. In the future, we hope to implement a model like ours into an application that can recognize these gestures and provide convenience analogous to voice assistance.

Scalability To More Users. Our current system requires training a dedicated model for each user due to a little difference among users while performing the same gesture. However, there are still a lot of similarities among different users that have not been leveraged. So, we look to implement a transfer learning technique to expand the system to more users with a low training cost in the future.

Implementation Programming Language. The data pre-processing and gesture data segmentation modules in our system are currently implemented using MATLAB. The lightweight CNN model in the gesture identification module is implemented using the Keras library in Python. In our future work, we will migrate all MATLAB processing code to Python for improving the actual deploy-ability. Moreover, we will manage to leverage edge-based hardware (e.g., an NVIDIA eGPU) to accelerate this application.

Targeted Hardware Platform. Our current system is collecting the CSI data from the Wifi card in the personal computer. In our future work, we will leverage the CSI data from the Wifi card in the commodity smartphone to achieve better convenience and wider usability.

VIII. CONCLUSION AND FUTURE WORK

In this paper we discuss gesture recognition using the CSI of WiFi signals and a lightweight CNN. Utilizing the CSI, we get a fine-grained gesture pattern that our deep learning model can recognize with a significantly low sampling rate and without costly feature extraction. We select unique gestures that can be interpreted by a computer to ultimately execute a command in response. We achieve an average of 96% accuracy, which is promising, especially considering a lightweight CNN is a computationally efficient enough model to work on mobile devices. In the future, we look to recognize more gestures and implement a transfer learning technique to lower the training effort on new user.

ACKNOWLEDGMENT

This work was partially supported by the National Science Foundation Grants CNS1757533, CNS1566455, and CCF2000480.

REFERENCES

- [1] A. Parate, M.-C. Chiu, C. Chadowitz, D. Ganesan, and E. Kalogerakis, "Risq: Recognizing smoking gestures with inertial sensors on a wrist-band," in *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, 2014, pp. 149–161.
- [2] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface emg images," *Scientific reports*, vol. 6, no. 1, pp. 1–8, 2016.
- [3] T. Zhao, J. Liu, Y. Wang, H. Liu, and Y. Chen, "Ppg-based finger-level gesture recognition leveraging wearables," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1457–1465.
- [4] —, "Towards low-cost sign language gesture recognition leveraging wearables," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1685–1701, 2019.
- [5] G. Marin, F. Dominio, and P. Zanuttigh, "Hand gesture recognition with leap motion and kinect devices," in *2014 IEEE International conference on image processing (ICIP)*. IEEE, 2014, pp. 1565–1569.
- [6] —, "Hand gesture recognition with jointly calibrated leap motion and depth sensor," *Multimedia Tools and Applications*, vol. 75, no. 22, pp. 14991–15015, 2016.
- [7] J. Lei, X. Ren, and D. Fox, "Fine-grained kitchen activity recognition using rgb-d," in *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, 2012, pp. 208–211.
- [8] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," *IEEE Signal Processing Letters*, vol. 23, no. 9, pp. 1188–1192, 2016.
- [9] L. Yu, H. Abuella, M. Z. Islam, J. F. O'Hara, C. Crick, and S. Ekin, "Gesture recognition using reflected visible and infrared lightwave signals," *IEEE Transactions on Human-Machine Systems*, vol. 51, no. 1, pp. 44–55, 2021.
- [10] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *2015 IEEE INFOCOM*. IEEE, 2015, pp. 1472–1480.
- [11] R. H. Venkatnarayan, G. Page, and M. Shahzad, "Multi-user gesture recognition using wifi," in *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, 2018, pp. 401–413.
- [12] M. Shahzad and S. Zhang, "Augmenting user identification with wifi based gesture recognition," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–27, 2018.
- [13] S. Tan and J. Yang, "Wifinger: Leveraging commodity wifi for fine-grained finger gesture recognition," in *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*, 2016, pp. 201–210.
- [14] W. He, K. Wu, Y. Zou, and Z. Ming, "Wig: Wifi-based gesture recognition system," in *2015 24th International Conference on Computer Communication and Networks (ICCCN)*. IEEE, 2015, pp. 1–7.
- [15] M. A. A. Al-qaness and F. Li, "Wiger: Wifi-based gesture recognition system," *ISPRS International Journal of Geo-Information*, vol. 5, no. 6, p. 92, 2016.
- [16] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD workshop*, vol. 10, no. 16. Seattle, WA, USA, 1994, pp. 359–370.
- [17] T. Macks, *American Sign Language Recognition using Adversarial Learning in a Multi-Frequency RF Sensor Network*. The University of Alabama, 2020.
- [18] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: Device-free location-oriented activity identification using fine-grained wifi signatures," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 617–628. [Online]. Available: <https://doi.org/10.1145/2639108.2639143>
- [19] Y. Xie, Z. Li, and M. Li, "Precise power delay profiling with commodity wifi," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '15. New York, NY, USA: ACM, 2015, p. 53–64. [Online]. Available: <http://doi.acm.org/10.1145/2789168.2790124>
- [20] M. Zeng, L. T. Nguyen, B. Yu, O. J. Mengshoel, J. Zhu, P. Wu, and J. Zhang, "Convolutional neural networks for human activity recognition using mobile sensors," in *6th international conference on mobile computing, applications and services*. IEEE, 2014, pp. 197–205.