mmASL: Environment-Independent ASL Gesture Recognition Using 60 GHz Millimeter-wave Signals

PANNEER SELVAM SANTHALINGAM, Computer Science Department, George Mason University
AL AMIN HOSAIN, Computer Science Department, George Mason University
DING ZHANG, Computer Science Department, George Mason University
PARTH PATHAK, Computer Science Department, George Mason University
HUZEFA RANGWALA, Computer Science Department, George Mason University
RAJA KUSHALNAGAR, Department of Science, Technology and Mathematics, Gallaudet University

Home assistant devices such as Amazon Echo and Google Home have become tremendously popular in the last couple of years. However, due to their voice-controlled functionality, these devices are not accessible to Deaf and Hard-of-Hearing (DHH) people. Given that over half a million people in the United States communicate using American Sign Language (ASL), there is a need of a home assistant system that can recognize ASL. The objective of this work is to design a home assistant system for DHH users (referred to as mmASL) that can perform ASL recognition using 60 GHz millimeter-wave wireless signals. mmASL has two important components. First, it can perform reliable wake-word detection using spatial spectrograms. Second, using a scalable and extensible multi-task deep learning model, mmASL can learn the phonological properties of ASL signs and use them to accurately recognize the ASL signs. We implement mmASL on 60 GHz software radio platform with phased array, and evaluate it using a large-scale data collection from 15 signers, 50 ASL signs and over 12K sign instances. We show that mmASL is tolerant to the presence of other interfering users and their activities, change of environment and different user positions. We compare mmASL with a well-studied Kinect and RGB camera based ASL recognition systems, and find that it can achieve a comparable performance (87% average accuracy of sign recognition), validating the feasibility of using 60 GHz mmWave system for ASL sign recognition.

CCS Concepts: • **Human-centered computing → Accessibility systems and tools**; **Personal digital assistants**.

Additional Key Words and Phrases: sign language recognition, 60 GHz milli-meter wave wireless, gesture recognition, personal digital assistants, accessible computing

ACM Reference Format:
Panneer Selvam Santhalingam, Al Amin Hosain, Ding Zhang, Parth Pathak, Huzefa Rangwala, and Raja Kushalnagar. 2020. mmASL: Environment-Independent ASL Gesture Recognition Using 60 GHz Millimeter-wave Signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 4, 1, Article 26 (March 2020), 30 pages. https://doi.org/10.1145/3381010

Authors' addresses: Panneer Selvam Santhalingam, psanthal@gmu.edu, Computer Science Department, George Mason University, Fairfax, Virginia, 22030; Al Amin Hosain, ahosain@gmu.edu, Computer Science Department, George Mason University, Fairfax, Virginia, 22030; Ding Zhang, dzhang13@gmu.edu, Computer Science Department, George Mason University, Fairfax, Virginia, 22030; Parth Pathak, phpathak@gmu.edu, Computer Science Department, George Mason University, Fairfax, Virginia, 22030; Huzefa Rangwala, rangwala@gmu.edu, Computer Science Department, George Mason University, Fairfax, Virginia, 22030; Raja Kushalnagar, raja.kushalnagar@gallaudet.edu, Department of Science, Technology and Mathematics, Gallaudet University, Washington, DC, 20002.

2474-9567/2020/3-ART26 $15.00
https://doi.org/10.1145/3381010

1 INTRODUCTION

With increasing interest in Internet-of-Things (IoT) devices, voice-controlled home assistants (such as Amazon Echo and Google Home smart-speakers) are becoming more and more popular. In the United States alone, more than 50 million home assistant devices were sold in 2018 [9]. Their easy-to-use functionality and ability to assist in one's personal and professional lives have enabled many novel applications. However, these devices are not readily available to Deaf and Hard-of-Hearing (DHH) people due to their voice-controlled interface. About 30 million people in the United States have bilateral hearing loss [49], and about 1 million are functionally deaf [22]. Speech production quality is correlated with hearing loss [53], which can lead to both speaking and listening difficulties. Around half a million people in the United States communicate visually through American Sign Language (ASL), and use it as their primary means of communication [56]. Hence, a home assistant system that can recognize ASL is highly desirable for DHH people. Apart from the convenience and entertainment applications, such devices can create ambient awareness for DHH users improving their safety and well-being [3, 5, 8].

ASL recognition has been studied mostly in the form of gesture recognition using a variety of sensing modalities. Use of RGB [21, 23, 50, 55, 60, 79] and infrared (Kinect [30, 68]) have been investigated extensively, however, they perform poorly in dark and incur privacy concerns due to constant video monitoring. Wearable sensors such as EMG [73], IMU [35, 41] and Leap Motion [32] capture motion that is highly localized to the body part where they are worn, and multiple on-body sensors [102] might be required for higher accuracy, reducing the overall usability compared to device-free solutions. Recently, 2.4/5 GHz WiFi CSI (Channel State Information) has been leveraged for gesture recognition [46, 74, 83, 89]. Although it enables device-free and low-cost sensing, its lower tolerance to change of environment (multi-path), user position, and presence of other moving users pose significant challenges in terms of accuracy and reproducibility. For example, [51] proposed to use WiFi CSI for sign recognition. However, it requires the user to be constrained to a specific location relative to link endpoints. Dealing with environment related changes has been recently looked at in [38] using adversarial learning, but the approach is shown to work for a few activity classes. Similarly, with CSI-based sensing, presence of other people can significantly affect the sensing accuracy, but this challenge has been addressed largely through controlled experiments.

In this paper, we present mmASL, a home assistant system for DHH users that can perform ASL recognition using 60 GHz millimeter-wave (mmWave) signals. 60 GHz mmWave has emerged as a viable candidate for designing the next generation of multi-gigabit WLANs [6, 37]. In order to compensate for high path loss experienced at 60 GHz frequencies, directional antenna such as phased antenna array is employed to concentrate the signal in a desired direction. In this paper, we demonstrate that large available bandwidth, higher frequency and directionality make 60 GHz signals a suitable candidate for gesture sensing and ASL recognition. mmASL has many salient features. First, it leverages the directionality and channel sparsity of 60 GHz indoor channels to make ASL recognition tolerant to the presence of other people (moving or stationary) in a room. Second, mmASL is able to create environment independent motion signatures that can allow training and testing across different indoor environments. Third, it does not restrict the user to be at any specific location within the room while performing the gestures. Lastly, mmASL is designed to extract and learn feature representations such that it can scale for a large number of ASL signs as well as user diversity in signing.

We note that 60 GHz sensing has been explored for near-field hand gesture recognition in [47] and fine-grained tracking in [92]. mmASL demonstrates its potential for sensing in typical indoor environments at larger distances, which bring in new challenges associated with multiple users, change of environment, etc. Through extensive experimentation and measurements, we show that device-free sensing using 60 GHz provides reasonable tolerance to changes in the environment, presence of other moving users, and user position. We leverage these properties to design mmASL, which can perform wake-word detection as well as ASL recognition. Using wake-word (ASL

counterpart to "Alexa" or "Ok Google") detection through spatial spectrogram (space-frequency-time maps), mmASL can detect user's intention to initiate a conversation with the home assistant device. Once the wake-word is detected, mmASL can recognize ASL signs using Doppler spread spectrograms. Given the large vocabulary of ASL lexicon, we focus our attention on 50 ASL signs based on words commonly used (e.g., weather, schedule, etc.) in interaction with home assistant devices. We note that using these subset of words, a user can create many sentences/commands to interact with home assistants. The proposed design of mmASL imposes multiple challenges. We describe the challenges along with our solutions:

(1) Reliable wake-word detection at any location: While initiating an interaction with mmASL, a DHH user can perform the wake-word at any location in a given indoor environment. Even with the directionality of 60 GHz signals, mmASL should be able to detect the wake-word reliably with a short response time. Also, the wake-word detection should be able to tolerate presence and activities of other people (for example, another person walking in the same room).

mmASL uses beam scanning where a predetermined set of beam sectors are scanned continuously and observed reflections are processed to create spatial spectrograms (concatenated spectrogram of all sectors). We utilize a Convolutional Neural Network (CNN) based machine learning model that can learn from spectrogram images and distinguish between wake-word and other activities. The model can even recognize partial wake-word gestures, allowing us to reduce the response time through faster scanning.

(2) Scalable and explainable ASL sign recognition: As a home assistant to DHH users, mmASL should be able to recognize a large number of ASL signs. Also, the representation of 60 GHz signal reflected from the user performing ASL signs and the machine learning algorithms using these representations should support and extend to such large classification.

mmASL exploits observed Doppler spread to create spectrograms for ASL signs which provide sufficient distinguishability between different signs. Instead of blindly applying machine learning on spectrograms, we work with an ASL domain expert and design an explainable multi-task deep learning model for sign recognition. The multi-task model includes separate tasks based on specific phonological properties of signs (e.g., repetitive motion, parallel/perpendicular movement) to learn feature representations that generalize well across a large number of signs and different users. The proposed model is extensible because more tasks for other phonological properties can be added in the future as the number of signs considered in the ASL recognition grows.

(3) mmASL in realistic, uncontrolled settings: ASL sign recognition is challenging because significant variations can exist between signs performed by different or even the same users. This is further complicated by consideration of scenarios such as presence of other interfering person and change of environment. Addressing these challenges requires large-scale data collection, analysis, model design and evaluation in realistic, uncontrolled settings.

We implement mmASL on NI+SiBeam 60 GHz multi-FPGA software radio platform with phased antenna array and analog beamforming. We develop mmASL using a large-scale data collection with 15 users[1] for 50 ASL signs (over 12K gesture instances [16]) in 6 different rooms. Our study includes a diverse set of scenarios such as random user location (varying distances and angles), presence of other interfering user walking or performing other activities in the same or different beams, and blocking of 60 GHz line-of-sight path. We also collect data simultaneously on Kinect and RGB camera for all gesture instances and compare it with our 60 GHz sign recognition system. In contrast with mmASL, Kinect provides data for 25 body joint movements (includes thumb, wrist, hand, elbow and shoulder joints, collectively called skeletal data) with respect to time. Given this fine-grained movement information, Kinect has been used in many existing works on gesture and activity recognition [31, 76, 105] and ASL recognition [45, 61, 99], providing a competitive baseline for comparing mmASL. ASL sign recognition has been predominantly studied using RGB camera based systems [77, 78, 80]. Unlike Kinect

[1]Approved by Institutional Review Board (IRB)

and mmASL, RGB cameras can provide palm and finger joint information, the importance of which has been explored in recent works [29, 40, 98].

With the provided solutions, mmASL advances the state-of-the-art of RF based gesture recognition with the following novel contributions.

- We demonstrate the feasibility of RF based solutions for long range and fine-grained gesture recognition, which can be environment independent, tolerate the presence of other users, does not restrict user's position, and scale for a large number of gestures.
- We propose a gesture-based wake-word detection solution, which to the best of our knowledge has not been studied in existing RF based gesture recognition literature. Our beam scanning based solution enables mmASL to not only detect a wake-word but also locate the intended user for subsequent ASL recognition.
- We develop a novel deep learning approach that combines ASL domain specific knowledge with contemporary deep learning algorithms for effective ASL sign recognition.
- To the best of our knowledge this is the first work to demonstrate that RF based gesture recognition can offer comparable performance with state-of-the-art Kinect and RGB camera based systems.

After an extensive evaluation, we find that mmASL achieves an average sign recognition accuracy of 87% for same users when trained and tested in different rooms. It achieves an accuracy of 73.75% for cross subjects (training and testing on different users) and the accuracy increases to 83% when additional data (from new users who where neither in training set nor in cross subject set) is added to the training set. Kinect and RGB camera-based recognition models achieve 96% and 97.3% for the same subjects, and 76.3% and 77.8% for cross subjects, respectively. Compared to a basic deep learning model, the multi-task learning architecture yields a gain of 10% in accuracy in case of untrained users and as much as 24% in case of random user locations. mmASL can detect wake-word with an average accuracy of 94% for untrained users in untrained environments. Lastly, mmASL is found to be tolerant to the presence of other people in both wake-word detection and sign recognition.

The remaining paper is organized as follows. Section 2 discusses the related work. Section 3 provides an overview of our proposed system and feasibility experiments. Section 4 explains wake-word detection and spatial spectrogram based model. Section 5 discusses multi-task deep learning sign recognition model. Section 6 provides details of data collection and numerical evaluation. Section 7 provides insights, limitations and future directions of improvements and we conclude in Section 8.

2 RELATED WORK

mmWave Sensing: Given their higher frequency and larger bandwidth, mmWave wireless signals have been used for sensing in recent years. Authors in [58], [47] and [87] have built custom FMCW (Frequency Modulated Continuous Wave) radars for short range (less than a meter) hand gesture recognition and fine-grained finger gesture recognition. Designed to provide hands-free interaction to smart devices, they operate by illuminating the hand and utilize Range-Doppler Maps for tracking the movement of ≤ 10 gestures. On the other hand, mmASL has to illuminate the entire upper body of the user because of the nature of the ASL signs (which involve both the hands and displacement of the hands can reach upto a couple of feet) and be available over long-ranges (a requirement for digital assistants). Operating in long-range ($4-6$ meters) brings new challenges such as presence of other people, change of environment, etc. Recently, a 60 GHz radar is used in [62] to recognize 8 gestures using FMCW range information. However, due to limited number of antenna elements, the system cannot perform beam scanning or steering which is necessary in case of mmASL DHH home assistant system. mmWave sensing is also used for fine-grained object tracking [93], and imaging using synthetic aperture radars [106]. mmVital [96] uses mmWave signals for locating a human being and monitoring her vital signs. In comparison, our approach uses beam-scanning and spatial spectrograms that are shown to be more robust than time-series metrics such as energy or variance used in mmVital. Authors in [38] recently proposed an environment independent activity

Table 1. mmASL compared with existing works on ASL recognition

System (Modality)	Device Free / Wearable (DF/W)	Other people impact	Environment impact	User position impact	Lighting impact	Wake word Detection	Non-Intrusive
RGB Camera [21, 55, 80, 97]	DF	Low	Low	Moderate	High	No	No
Depth Camera [30, 36, 68] (Kinect)	DF	Low	Low	Moderate	Moderate	No	No
DeepASL [32] (Leap motion)	W	None	None	None	None	No	Yes
SignSpeaker [35] (IMU)	W	None	None	None	None	No	Yes
SignFI [51] (WiFi CSI)	DF	High	High	High	None	No	Yes
mmASL (mmWave)	DF	Low to Moderate	Low	Moderate	None	Yes	Yes

recognition using mmWave. In comparison, in addition to dealing with larger set of signs, we also show that mmASL is tolerant against presence of another human in the room.

2.4/5 GHz CSI/RSSI based systems: CSI and RSSI based systems have been well studied in recent years for activity recognition [19, 33, 86, 88–91, 100, 107] and gesture recognition [18, 46, 51, 67, 75, 83]. Authors in Wigest[18] utilize primitives built on WiFi signal strength (RSSI) to sense 9 hand gestures. WiSee[67] extracts Doppler shifts from WiFi signals in recognizing 9 gestures where variability in Doppler energy among gestures is used for classifying them. In comparison, 60 GHz signals provide higher Doppler shifts, allowing the recognition problem to scale to ASL domain which is difficult to achieve using variability in Doppler energy at lower frequencies. In [46, 83], authors use CSI information to recognize 10 gestures and signs for numbers. WiSign[75] utilizes multiple antennas with CSI measurements to classify between 5 ASL signs, and SignFi [51] classifies among 276 ASL signs. These CSI-based systems require the user to be sitting/standing at a fixed location usually in close proximity (0.3 meters in case of SignFi) of the access point while performing the gestures. In comparison, mmASL includes a large number of gestures (only studied in WiSign), poses no restrictions on user position, and includes practical scenarios such as presence of other human beings and change of environment (refer Table 1). WiAG [85] performs position and orientation agnostic gesture recognition using CSI where virtual samples for all possible gestures in the perceived configuration are generated and matched with the received samples. mmASL can detect the ASL signs without requiring samples in specific configurations for any user position in range and minor changes to orientations. Additionally, mmASL can tolerate the presence of other interferer while the intended user performs the gestures. CSI measurements have been used by researchers in fall detection [91], human activity recognition [89], determining people count in a crowd [95] etc. [88] authors use spectrograms to capture frequency change in CSI measurements at the receiver in recognizing human gait. In our work, we augment the spectrograms with spatial (beam sector) information to further improve the accuracy of wake-word detection and gesture recognition. In [54] authors showcase the ability of directional antennas in recognizing ASL signs in a constrained setup (car/office chair). mmASL shows feasibility of such recognition for a large number of gestures in many practical scenarios (interferer, varying distances and angles, etc.).

Vision, infrared and other modalities: ASL recognition using Kinect and Leap-motion has been well-studied because of the direct availability of skeletal joint data. Authors [30] and [68] utilize Kinect for recognizing static ASL alphabets by determining the hand shape from depth data. In [36, 64, 81], authors develop different methodologies to improve word level ASL sign recognition accuracy. All of them utilize the mentioned skeletal joints as features in recognition. While Kinect gives specific joint information, it performs poorly in bad lighting conditions (refer Table 1). [28], [69], [43] utilize leapmotion in recognizing ASL alphabets, while [32] utilizes Leap-motion for both word-level and sentence-level recognition. Leap-motion provides skeletal joint data for fingers, palms and fore-arms which is used in detecting different ASL signs. While Leap-motion based systems have proven effective [32], they require the user hand to be in close proximity of the sensor, making them less suitable for home assistant systems. Even when used as a wearable on chest, many ASL signs which have their major location to be head or above shoulder cannot always be accurately recognized. RGB camera based systems [21, 23, 50, 55, 60, 77–80]. They pose multiple challenges including low accuracy in low lighting cases, restricted poses and privacy concerns due to constant monitoring (refer Table 1). Researchers have also utilized wearable sensors in ASL sign recognition. This includes on-body IMU and EMG Sensors used in [41, 66, 73, 102]. Such methods require user to either wear multiple on-body sensors to capture motion of each body part (e.g., both hands) or a smart-glove, reducing the overall comfort in using such systems.

3 SYSTEM OVERVIEW

3.1 Platform Overview & Implementation

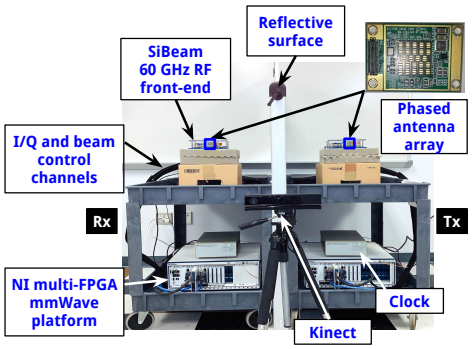


Fig. 1. mmASL's implementation on NI+Sibeam 60 GHz software radio platform

Today's typical home assistant (such as Amazon Echo) is a single device which a user can interact with using voice commands. Inspired by this design, mmASL is built using co-located Tx and Rx similar to a mono-static radar. A highly reflective surface is used to separate Tx and Rx to reduce their direct communication. Fig. 1 shows mmASL setup.

We implement mmASL using 60 GHz reprogrammable software radio platform. This transceiver system is developed using National Instruments (NI) multi-FPGA platform [1], and a V-band RF front-end and phased antenna array from SiBeam [7]. The NI+SiBeam system includes multiple FPGA modules (NI PXIe 7902, 7976 and 3610/3630) for implementing ADC/DAC, modulation/demodulation and encoding/decoding. The SiBeam antenna array (as shown in Fig. 1) has 24 antenna elements (12 for Tx and 12 for Rx) and is capable of performing analog beamforming. The SiBeam codebook includes 25 beam sectors which cover a region of -60° to +60° from antenna broadside. The 3-dB beamwidth of each beam ranges from 25° to 30° for Tx and from 30° to 35° for Rx, and each of the 25 beam sectors are approximately 5° apart. The NI+SiBeam platform is connected to a host which sends/receives data to/from FPGA and implements additional signal processing tasks. In mmASL, Tx and Rx hosts are connected via Ethernet for control and coordination (e.g., setting a specific Tx and Rx beam sector while scanning).

The transmit and receive side processing pipelines are shown in Fig. 2. On transmit side, the host generates a 1 MHz signal that is copied to an intermediate FPGA using host-to-target Direct Memory Access (DMA). The data is then used by a DAC module (sampling rate 3.07 GHz) which generates analog baseband signal and sends it to SiBeam RF front-end board where it is upconverted to 60.48 GHz before being transmitted on the current sector.

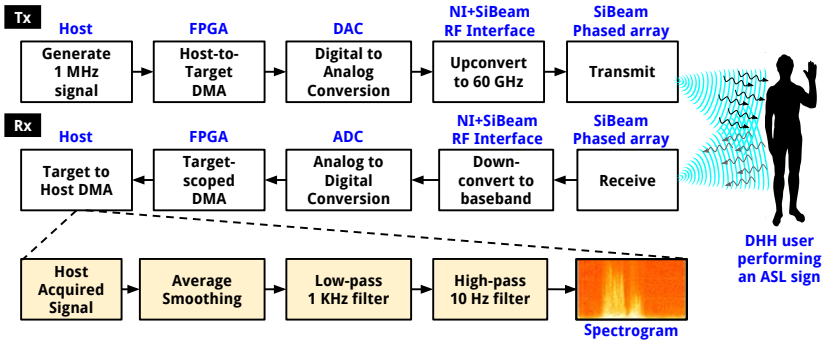


Fig. 2. mmASL signal processing pipeline

On the receive side, the received signal is downconverted and downsampled, first while copying from ADC to the FPGA (3.07 GHz to 192 MHz), and then copying from FPGA to the host (at 16 MHz). We implement the entire processing pipeline using NI's Labview/ Xilinx FPGA modules.

3.2 Doppler Spread and Spectrograms

For mmASL to reliably recognize ASL signs, it should extract components of the received signal which are representative of the performed sign. mmASL utilizes a sinusoid of 1 MHz as baseband signal. Here the reflection profile of a gesture can be understood as a wireless channel between Tx and Rx. Let us assume a wireless communication channel with a single Tx and Rx, the Rx is moving with a velocity v and located close to a reflecting wall. The signals reflected from the wall and ones that reach directly to the moving Rx follow a pattern of constructive and destructive interference [84]. This causes the received signal strength to increase and decrease over time. This can be equivalently represented using Doppler shifts of the direct and reflected signals given by

$$E_r(f,t) = \frac{\alpha cos 2\pi f[(1-\frac{v}{c})t - \frac{r_0}{c}]}{r_0 + vt} - \frac{\alpha cos 2\pi f[(1+\frac{v}{c})t - \frac{(r_0-2d)}{c}]}{2d - r_0 - vt} \quad (1)$$

Here, Tx transmits a sinusoid $\cos 2\pi f t$, α represents Rx beam pattern and E_r is the electric field at the receiver. The Rx is at r_0 at time t_0 and reaches $r = r_0 + vt$ at time t, d is the distance between the Tx and wall, and c is the speed of light. The first term (direct signal) is a sinusoid of frequency $D_1 := -\frac{fv}{c}$ and the second term (reflected signal) is another sinusoid of frequency $D_2 := +\frac{fv}{c}$. The resulting Doppler spread is $D_s := D_2 - D_1$. In case of mmASL, the user acts as a multipoint reflector and scatterer, with different body parts producing different Doppler shifts (each body part moves with a different velocity depending on the sign performed). These individual Doppler shifts add up constructively/destructively resulting in a Doppler spread [84] which can be extracted from the received composite signal. mmASL exploits the change in Doppler spread observed over time for distinguishing different ASL signs.

In order to extract the Doppler spread, we process the received 16 MHz signal on host as shown in Fig. 2. We empirically observe that the maximum Doppler shift does not exceed 1 KHz in our ASL experiments. In order to avoid poor roll-off and stopband attenuation with 1 KHz low-pass filter on 16 MHz signal, we first downsample the signal to 8 KHz using smoothing (window size of 2K samples). We then apply the 1 KHz low-pass filter and 10 Hz high-pass filter. The 10 Hz high-pass filter removes the impact of low frequency human activities such as breathing and posture changes. The resulting filtered signal is then used to plot spectrograms using Short Time Fourier Transform (STFT) with a window size of 0.8K samples (100 ms), while sliding the window at every 1 ms (8 samples). Lastly, we use the log-transformation of amplitude values to normalize (addressed as log normalization) and emphasize on low intensity components (inspired by speech recognition literature [52, 57, 59]). Fig. 3 shows

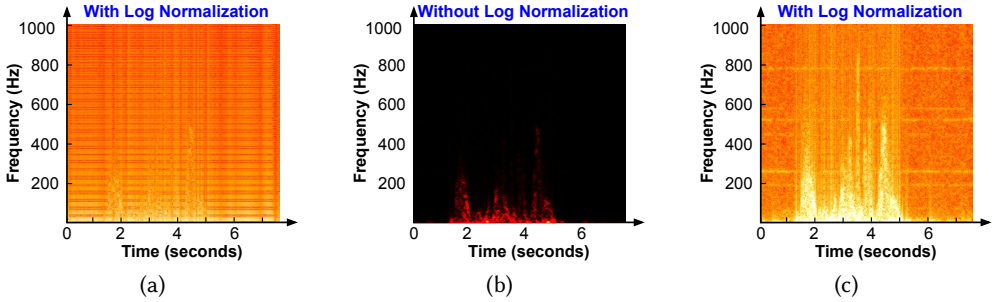


Fig. 3. The impact of signal processing on mmASL (a) Spectrogram generated without filtering the data, (b) spectrogram generated with filtering but without log normalization, (c) spectrogram generated with both filtering and log normalization.

the significance of different signal processing components in mmASL. It is evident from Fig. 3a that removal of low frequency, high intensity components is required for a better visual representation of the underlying ASL sign. Also, log normalization (Fig. 3c) further enhances the spectrogram by bringing forth the low intensity components not visible in Fig. 3b.

3.3 Feasibility Study

In order to verify that (1) the generated spectrograms indeed capture Doppler spread and (2) different ASL sign have a different Doppler spread over time, we perform two types of feasibility experiments. First, as shown in Fig. 4a, we use a plain steel metal plate of size 12" × 18" and attach it with a long wooden stick. The metal plate is then moved at three different speeds (fast: 0.91 m/s, medium: 0.73 m/s and slow: 0.3 m/s). The experiment is repeated at a distance of 10 ft and at two angles (0° and 30°), with Tx and Rx sectors both pointing to the angle. Fig. 4 shows the spectrograms for 30°. It is clear that the observed Doppler spread correlates with the expected Doppler shift, as for the faster movements, the spread has more higher frequency components compared to the slower movements. Results for 0° location also show the same trend.

In the second type of feasibility experiment, we ask a user to perform ASL signs in a continuous as well as segmented manner. In the segmented case, the user pauses between different movements of the gesture. Fig. 5a shows the four segments of an ASL sign for `AIR CONDITIONER` with visual depiction of segments. Fig. 5b shows the spectrogram of the same sign when performed continuously. We find that the Doppler spread over time is consistent in both cases. Figs. 5b and 5c show the difference in Doppler spread between two ASL signs (`AIR CONDITIONER` and `DIRECTION`). We observe that it is feasible to use the spectrograms to recognize ASL signs.

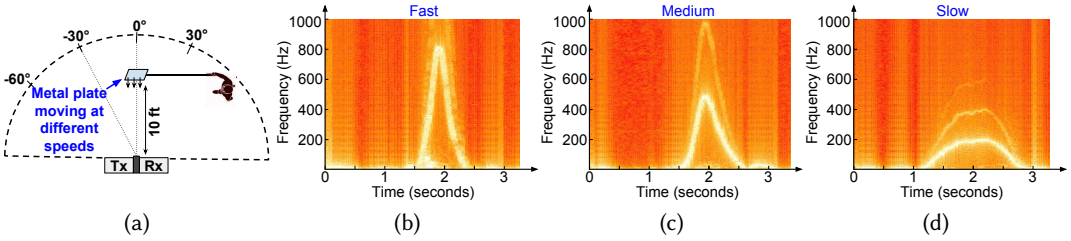
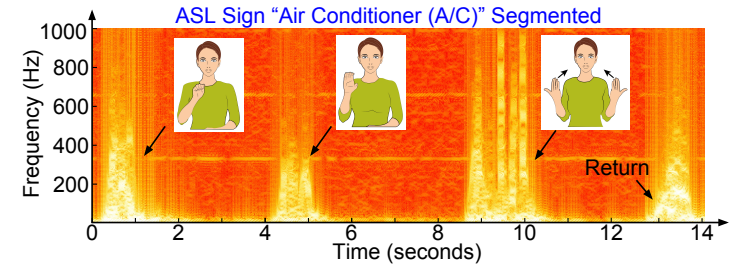
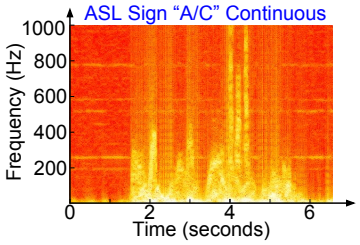


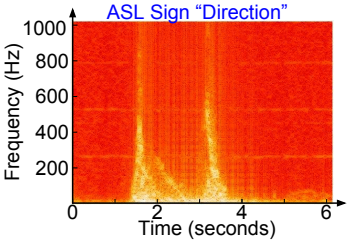
Fig. 4. Observed Doppler spread for a metal plate moving at different speeds



(a) ASL sign A/C performed with pauses



(b) A/C performed continuously



(c) ASL sign for DIRECTION

Fig. 5. Doppler spread spectrograms are (a-b) consistent in segmented or continuous gestures, and (b-c) distinguishable for different signs

3.4 Overview of mmASL

Fig. 6 shows the high-level system design for mmASL. There are two important components: (1) Wake-word recognition and sector determination and (2) ASL sign recognition. mmASL continuously scans through a predetermined set of beam sectors and creates spatial spectrograms. The spatial spectrograms are used to detect if there is a wake-word performed by a user or not using a CNN-based model applied on the spectrogram images. If a wake-word is detected, a CNN-based classifier is used to determine the sector in which the user is currently located. Once a sector has been determined (Tx and Rx sector pointing to the user), her ASL gestures are captured in the form of ASL sign spectrograms. mmASL uses a multitask deep learning model (with auxiliary tasks that are used to learn phonological properties) on the ASL sign spectrograms to recognize the signs.

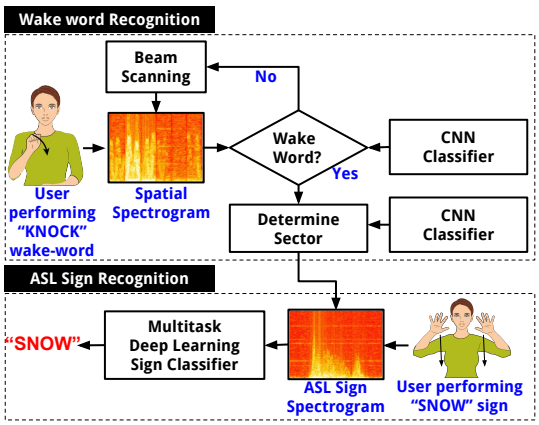


Fig. 6. mmASL’s overview

4 WAKE WORD RECOGNITION

Wake-word (a sign in case of ASL) detection should meet the following requirements: (1) It should not restrict the user to be at a specific location relative to Tx and Rx while performing the wake-word, (2) Presence of other people (moving or stationary) should be tolerated (should not significantly affect wake-word detection

performance), (3) mmASL should work in any environment and changes in position of ambient reflectors should affect the recognition performance minimally, (4) The wake-word should be detected with a short response time. One possible option in meeting the requirements is to utilize omni or quasi-omni beam pattern to recognize the wake-word occurring in any direction. However, this can significantly increase reflections from unwanted objects such as other people walking around. To avoid this, mmASL uses a beam scanning approach where a set of predetermined beams are scanned repeatedly for wake-word detection. As shown in Fig. 6, beam scanning helps in both detecting the wake-word as well as determining the beam sector to be used for subsequent ASL sign recognition phase. We use a cascaded model to accomplish this.

Using beam scanning for wake-word detection brings two important challenges. First, it presents a time-performance trade-off between the number sectors to scan and the amount of time to dwell in each sector. Second, presence and motion of people in the same or other sectors will also be detected with beam scanning. Hence, it becomes necessary to distinguish the wake-word motion from other types of motion (e.g., other person walking in background) to determine if and where (sector) the wake-word was performed.

We address these challenges by first selecting a subset of sectors for beam-scanning and then creating spatial spectrograms based on observed signal reflections. Here, we use KNOCK sign (performed as knocking on a door twice) as the wake-word. Note that choice of wake-word can be subjective and we use KNOCK as its symbolic meaning is similar to a typical wake-word. In evaluation, we compare its performance with another wake-word.

4.1 Selecting Scanning Sectors

Out of the 25 beam sectors (each 5° apart) offered by NI+SiBeam codebook, we select the minimum number of sectors that can provide the same coverage as the 25 sectors and have the minimum overlap between them. We do this by finding beams with least overlapping gain pattern and at least 20° apart from its neighboring chosen beam. Based on the observation, we choose 6 patterns (beam indices 2, 5, 9, 12, 16 and 20 corresponding to angles -60°,-30°,-15°,0°,+30°,and +60° from antenna broadside, respectively as shown in Fig. 7) and use them in the beam scanning process described above. Note that we restrict our search in azimuth plane only because the home assistant device and user are typically at the same elevation. While performing the beam scanning, both Tx and Rx point to the same sector at each step. From Fig. 7, it is clear that although the main lobes of the chosen sectors do not have significant overlap, there is overlap in the sidelobes. This results in wake-word being observed in adjacent sectors, albeit at a lower intensity.

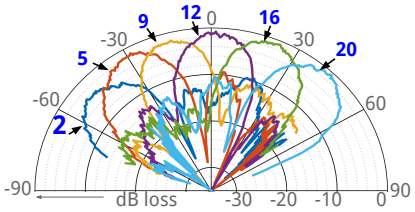
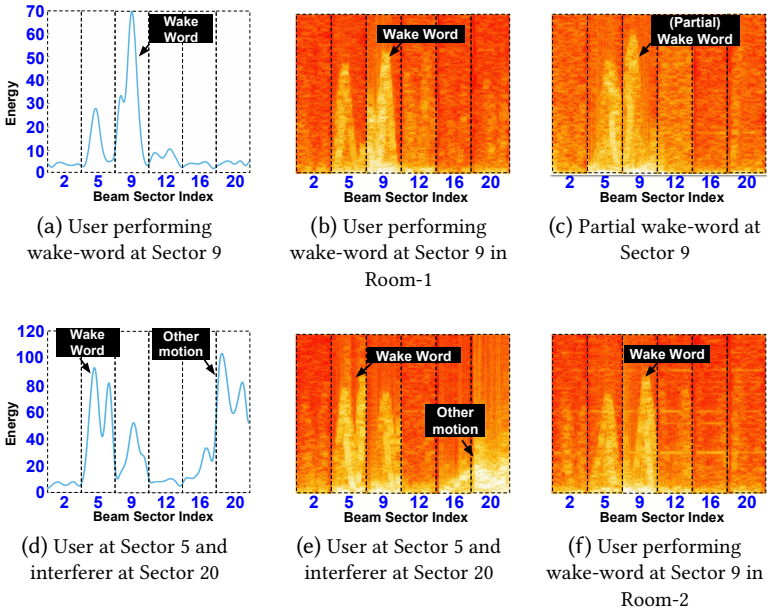


Fig. 7. 6 sectors selected from SiBeam codebook of 25 sectors (Sector index: 2, 5, 9, 12, 16 and 20)

4.2 Spatial Spectrograms

With beam scanning, we need to use the reflected received signal in each sector to detect the wake-word and determine the sector for user interaction. One possible approach is to calculate motion energy which has been used in CSI-based activity recognition [20, 27, 100]. Motion energy has been used to distinguish walking from other activities in [100]. Motion energy can be calculated through frequency domain analysis as Energy = $\sum_{i=1}^{\text{window\_length}/2} M^2$ where M is the magnitude, i.e. normalized FFT values over the given time. However, we find that such drastic reduction of dimensions results in misclassifications. Fig. 8a shows the energy of six beam sectors while the user is performing the wake-word (KNOCK) in Sector 9, and Fig. 8d shows the energy when user is performing wake-word at Sector 5 and another interfering user is walking at Sector 20. We can observe that in presence of other user's motion, energy alone cannot be used for wake-word detection.



(a) User performing wake-word at Sector 9

(b) User performing wake-word at Sector 9 in Room-1

(c) Partial wake-word at Sector 9

(d) User at Sector 5 and interferer at Sector 20

(e) User at Sector 5 and interferer at Sector 20

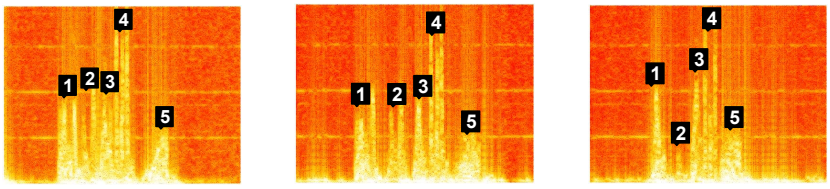
(f) User performing wake-word at Sector 9 in Room-2

Fig. 8. (a, b, d, e) Comparing motion energy with spatial spectrograms, (c) spatial spectrogram with partial wake-word (b, f) Spatial spectrogram in different rooms

We instead use an image representation referred as spatial spectrograms. The spatial spectrograms are generated by collecting samples for each sector in beam scanning, performing STFT for each sector's data separately, and concatenating them. Fig. 8b shows how a wake-word is observed in the spatial spectrograms. Compared to the energy metric, it can be observed in Fig. 8e that motions of other people can be clearly distinguished in spatial spectrograms. We also observe in Fig. 8c that partial wake-word has visual resemblance to the complete gesture. Also, the spatial spectrograms do not change significantly in different rooms or environments (Figs. 8b and 8f). This is because direct reflections from ambient reflectors add only low frequency components and second order reflections (e.g., Tx->wall->user->Rx) usually have very weak signal strength.

However, while spatial spectrograms convey visually distinguishable features, we need machine learning models that can learn feature representations from them. The learned feature representations should be helpful in both detecting the wake-word and determining the beam sector. In order to learn on spatial spectrogram images, we employ a CNN-based machine learning model. The model uses two stacks of convolutional layer (feature representation), a maxpool layer (down sampling) and rectilinear units (non-linearity). This is followed by a fully connected layer, a dropout layer, and another fully connected layer with softmax for prediction. Each convolution layers has kernels of size 5x5 with the filter count of 32 and 64 for the first and second layers.

To establish the improvement that spatial spectrograms offer (without even considering CNN), we build two machine learning models with features based on Principle Component Analysis (PCA) of spatial spectrogram. For determining the features (number of components) to use in PCA, we pick the top features (75 components) that explain 95% of the observed variance in the data. We apply Support Vector Machine (SVM) and Random Forest (RF) on the PCA features. We then compare them with SVM and RF models built using energy metric. The spatial spectrograms achieve an average accuracy of 83% compared to energy based metric which achieves an accuracy of 63%. The difference quantifies the importance of spatial spectrograms in building the feature space for the machine learning model. Average accuracy of wake-word detection using spatial spectrogram with CNN is observed to be 94%. We leave the complete evaluation to Section 6.



(a) User-A performing A/C (Instance 1) (b) User-A performing A/C (Instance 2) (c) User-B performing A/C (Instance 1)

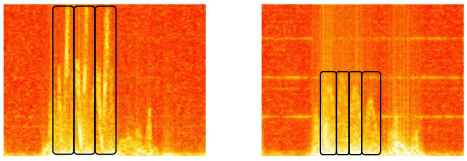
Fig. 9. Commonality and variations observed in instances of the same sign from same and different users. The numbered components are distinct movements of sign: (1) perform "A", (2) Perform "C", (3) lift hand up, (4) move back and forth fast, (5) return.

5 RECOGNIZING ASL SIGNS

Following wake-word recognition, the user performs ASL signs that mmASL should recognize. As shown in signal processing pipeline (Fig. 2), we use variation in Doppler spread captured in the form of spectrogram images for ASL sign recognition. The sign recognition problem is complex due to two reasons. First, compared to wake-word detection which is a binary classification problem, accurate ASL sign recognition should require classifying among a large number of signs. This means that any proposed recognition model should generalize well for a large number of signs. Second, while different instances of the same ASL sign should have some underlying common pattern, they can vary significantly between different users and even different instances of a sign from the same user. This is evident from Fig. 9. This problem is further aggravated by the fact that the amount of training data available is likely to be limited.

CNNs have shown to achieve good performance in image/object recognition tasks. However, use of CNNs in mmASL should answer the following questions:

(1) CNNs learn rich hierarchical feature representations at different layers (For example, CNNs focus attention on eyes, nose, etc. when tasked with face detection [44]). However, the question remains that can CNNs learn such feature representations from spectrogram images of ASL signs?
(2) Can CNNs learn feature representations that are truly discriminatory given limited training data? With limited training data, it is possible that they learn feature representations that are specific to training data, but not true discriminants of the underlying classes (For example, machine learning models learned presence of snow in images to distinguish between huskies and wolves, given a limited data set[71]).



(a) ASL sign SHOPPING (b) ASL sign PLACE

Fig. 10. Two ASL signs involving repetitive motion (three times for SHOPPING and four times for PLACE), also observed in spectrograms

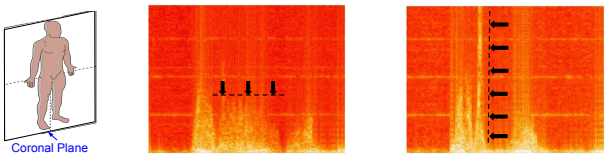
We answer these questions in two parts. First, we identify two phonological properties of ASL signs that incorporate ASL domain specific knowledge in our model. Second, we build a multitask learning model that augments CNN by learning feature representations that focus on these phonological properties. The presented multitask learning model is general and modular which can be scaled with more auxiliary tasks, as the ASL recognition problem grows.

5.1 Exploiting ASL Phonological Properties in Spectrograms

Similar to spoken languages, ASL has phonological properties [24] such as major and minor location of the sign (head, arm, etc.), sign type (symmetric, one-handed, etc.), and movement type (back-and-forth, straight, curved,

etc.). ASL-LEX [26] clustered ASL signs based on the phonological properties to understand their similarity. For example, for the ASL sign of RAIN, the major location is neutral, movement type is back-and-forth/repetitive, and sign type is symmetric (between two hands). While translating all the phonological properties to observable feature representations in spectrograms would be difficult, we leverage two such properties in mmASL: (1) Repetitive and (2) Motion direction.

(1) Repetitive: When a user performs a sign that involves a repetitive motion, it translates to a similar repeating components/patterns in spectrograms. Fig. 10 shows the spectrograms for two repetitive ASL signs: SHOPPING [14] and PLACE [13]. Both signs involve repeating a motion three times which is also observed in spectrograms. In ASL-LEX, 359 signs out of 1000 are repetitive in nature. 17 signs are repetitive out of 50 signs considered in this paper.



(a) Coronal Plane (b) ASL sign SNOW (c) ASL sign EMAIL

Fig. 11. Two ASL signs with parallel and perpendicular movements with respect to the coronal plane

(2) Motion Direction: The second phonological property we leverage is based on the direction of hand movement (parallel or perpendicular) with respect to coronal plane of the human body (Fig. 11a). When the hands move perpendicular to the coronal plane, we observe higher intensity components at higher frequencies in the spectrogram as the movement is likely to result in higher relative velocity compared to Rx. On contrary, signs involving motion parallel to the coronal plane results in relatively low intensity and low frequency components. This is evident in Fig. 11 which shows spectrogram for SNOW (parallel) [15] and EMAIL (perpendicular) [12].

5.2 Multitask Learning Model

Training the multitask learning networks:

Multitask learning (MTL) is a learning technique which is designed for simultaneous learning of multiple and related prediction tasks. By exploiting commonalities and differences across the related tasks, it achieves better generalization for all tasks [103]. MTL has specific characteristics that can address the challenges highlighted earlier: **(1) Attention focusing:** With high dimensional limited data set, multitask learning requires the learning model to focus attention on only features that are relevant, **(2) Representation bias:** Multitask learning biases the learning model to learn features which are representative of all the tasks that are being learned [72].

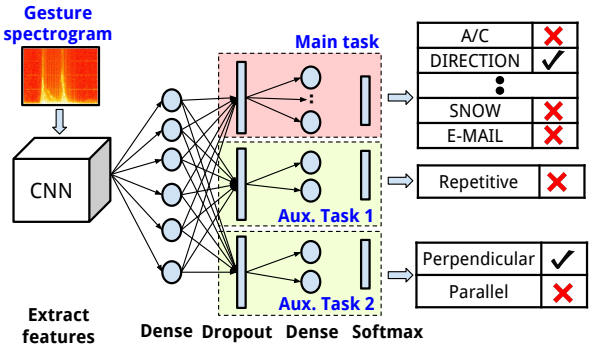


Fig. 12. Proposed multi-task deep learning architecture with main task (ASL sign recognition) and two auxiliary tasks (repetitive and motion direction)

Fig. 12 shows our multitask deep learning model with hard parameter sharing (all tasks share the initial hidden layers followed by task specific output layers [72]). It has two auxiliary tasks (repetitive motion and motion direction) which help in improving the performance of main task of sign recognition through attention focusing and representation bias. Compared to conventional multitask learning where all tasks need to perform well, our

approach (inspired from [104]) cares only about the main task performing well. The auxiliary tasks help the main task in focusing attention on relevant features as well as learning effective representations. As mmASL takes spectrogram images as input, we choose the network design based on well established image recognition model [42]. Alex-Net, utilizes multiple convolutional layers with increasing filter count (enables hierarchical feature representation learning) and maxpool layer (for downsampling) in between, followed by dense and softmax layers (for final prediction). We adopt a similar design for the proposed model with convolutional layers for learning hierarchical feature representations, which is followed by dense layer shared by all the tasks and task specific dropout (to avoid over-fitting) and softmax layers for individual task classification. The model is intuitive, explainable (in terms of what it learns) and scalable (extended for more signs with the addition of more parallel tasks).

We use cross validation to choose the appropriate hyperparameters (parameters which are chosen –in designing the network– and not learned) for the model. The different hyperparameters we validate are the number of convolutional layers, number of dense layers, and dropout rate. For cross validation, we use data collected from 7 users for all the 50 gestures, with 75% of data for training and the remaining for testing. We compare the variation in accuracy when the network is trained for 400 epochs, starting from the 25^{th} epoch we evaluate every 50^{th} epoch. An epoch is the time taken for the network to perform one iteration of training (feed forward, compute the loss and back propagate the losses) on the entire training set. The networks are trained with a learning rate of 0.0001 and batch size of 10 (we observe that smaller batch size results in faster convergence). We use Adam optimizer [39] for optimizing the networks. Unlike traditional deep learning where we minimize a single loss function while optimizing the network, in multitask learning we have as many loss functions as the number of tasks. Let L_m, L_{a1}, and L_{a2} be the loss functions for the main task, auxiliary task-1 and auxiliary task-2. For each task T, we use cross entropy over the predicted and target distributions as the loss function (L_T) given by

$$L_T = \frac{-1}{N} \sum_{i=1}^{N} \sum_{j=1}^{S} y_{ij} \log(\hat{y}_{ij}) \tag{2}$$

where N is the number of samples, S is the number of ASL signs, y and $\hat{y}$ are ground truth and predicted probability, respectively. We define the combined loss function for the three tasks as

$$L = L_m + \lambda_1 L_{a1} + \lambda_2 L_{a2} \tag{3}$$

where λ_1 and λ_2 are importance coefficient of auxiliary task-1 and auxiliary task-2, respectively. We set λ_1 to 0.75, and λ_2 to 0.5 (we observe that setting values greater than 0.5 and less than 0.75 for both λ_1 and λ_2 yield similar results). One of the challenges with adding more tasks is that the tasks themselves can compete against each other, given that we are minimizing the combined loss function just tuning importance coefficient is not helpful. To address this challenge, we adopt a task specific early stopping approach based on the performance of each task [104].

Validating the hyperparmeters: We consider four dropout rates (20%, 40%, 60% and 80%), three different dense layer configurations (1, 2 and 3 dense layers), and three different convolutional layer configurations (3, 6 and 9 convolutional layers) in our hyperparameter validation. All the dense layer configurations have 1024 hidden units in each layer. Irrespective of the number of convolutional layers chosen, we only use 3 maxpool layers (used for down sampling) in the entire network, each with a size of 4×2 and a stride (amount of movement between consecutive applications) of 2. The number of filters in each layer is chosen to be twice that of the previous layer. We start with a filter count of 32 for 3 layer model, 16 for 6 layer model, and 8 for 9 layer model. Drawing inspiration from [65], we use rectangular convolutional kernels of sizes 20×4, 10×4, and 5×2 (in the given order), with padding such that the resulting size post convolution is the same as input and a stride of 1 (for the convolution operation). For models with more than three layers, we use the same convolutional kernel size for consecutive layers before the maxpool layer. The maxpool layer is always inserted between layers of different

convolutional kernel sizes. We use Rectified Linear Units (ReLU) for activation in both convolutional and dense layers.

Fig. 13 shows the impact of different hyperparmeters on the proposed multitask deep learning architecture. From Fig. 13a, it is evident that low dropout rates (20% and 40%) significantly impact the performance (as the networks over-fit), while the dropout rates greater than 50% offer comparable performance. The variation in number of dense layers does not offer much variation in performance (refer Fig. 13b). The same is observed in case of the convolutaional layers where additional layers (6 and 9) do not provide any significant increase in performance compared to 3 layers (refer Fig. 13c). Based on the observed results, we configure the proposed architecture with the following hyperparameters: the number of convolutional layers is set to 3, the number of dense layers is set to 1 and dropout rate is set to 0.8. Fig. 13d shows the learning curve for the proposed architecture with the chosen hyperparameters, with and without multitask learning. For the network without multitask learning, we train without the auxiliary tasks and optimize the network only using the main task's loss (L_m). While the learning curve exhibits expected increase in performance with increase in data, the model with multitask learning performs consistently better (4% more accuracy) than the model without multitask learning. We further quantify the significance of multitask learning in the evaluation section.

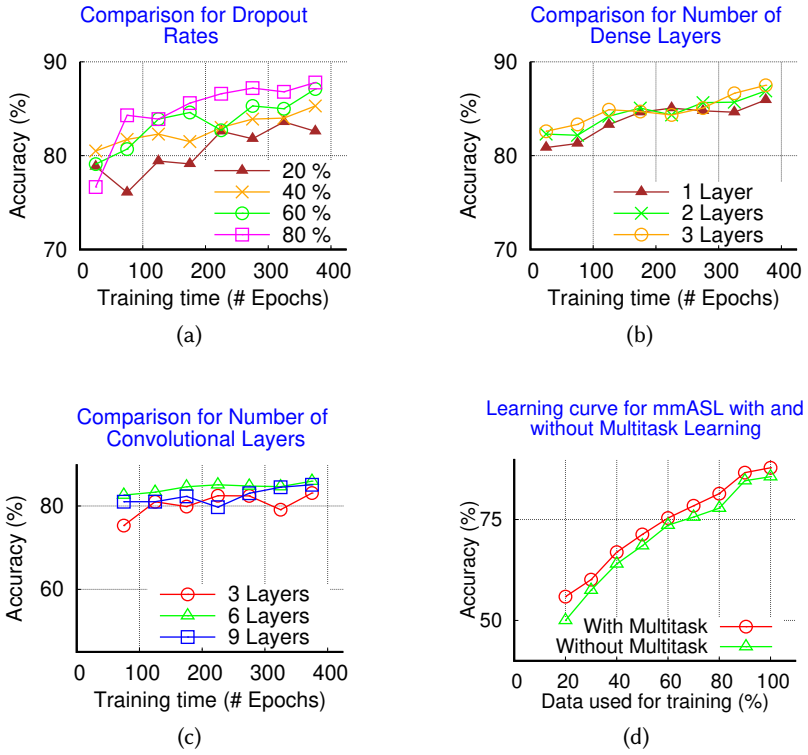
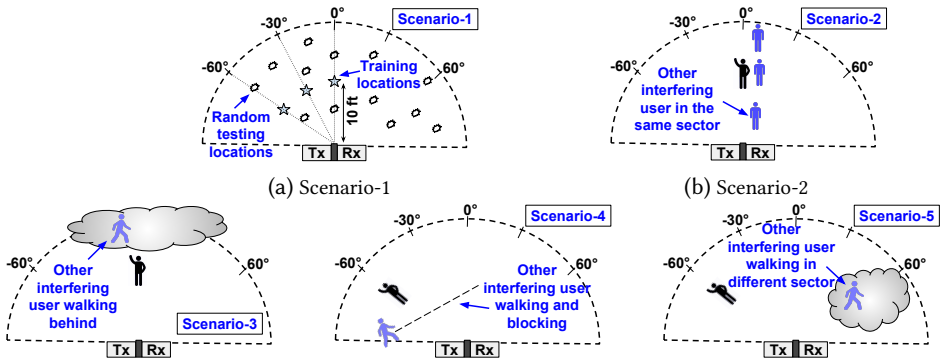


Fig. 13. (a-c) Evaluating mmASL on different model hyperparameters, (d) Leaning curve for mmASL with and without multitask learning.



(a) Scenario-1 (b) Scenario-2 (c) Scenario-3 (d) Scenario-4 (e) Scenario-5

Fig. 14. (a) Scenario 1: Single person scenario with randomly chosen training and testing locations (distance anywhere from 5ft to 15ft, angle anywhere between -60° and +60°) in different rooms (b-d) Scenarios 2-5: Multi-person scenarios with one user performing signs and the other user interfering in different ways

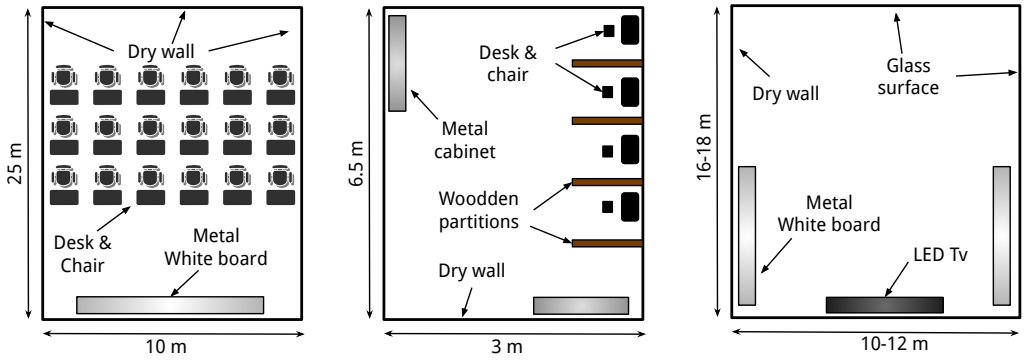
6 EVALUATION

6.1 Data Collection and Implementation

mmASL is evaluated using a large amount of data covering a range of practical scenarios involving multiple users and different environments. The datasets can be divided into two parts: Wake-word dataset and ASL sign dataset.

Scenarios and Environments: Both datasets are collected for different scenarios shown in Fig. 14. They include single-user (Scenario 1) and multi-user (intended user and an interferer as in Scenarios 2-5) scenarios. Figure 15 shows the different environments (university classroom, lab and conference rooms) where the data collection was performed. Note that in every scenario (even in the case of single-user), there are always at least two additional persons in the room behind the 60 GHz system collecting the data and performing uncontrolled movements.

Wake-word dataset: The wake-word dataset includes a total of 3700 samples, with each sample being 3 seconds long. We collect data for three users (User A, B, and C) in three different rooms: conference room (Figure 15c), lab (Figure 15b), and class room (Figure 15a). The data is collected for all the five scenarios shown in Fig 14. For Scenarios 2-5, User A performs the wake word (KNOCK or UP-DOWN), and User B performs random



(a) Environment-1 (Class room) (b) Environment-2 (Lab) (c) Environment-3 (Conference room)

Fig. 15. Different environments where mmASL data collection was performed.

Table 2. ASL sign recognition data set: table summarizes data for 15 subjects detailing the training and test data for different scenarios (shown in Fig. 14) and users. Superscript in ASL signs identifies phonological properties where "h" is horizontal (parallel) and "v" is vertical (perpendicular) to coronal plane, and r means repetitive.

ASL Signs (50)	AC^{-rv}, Alarm^{-rh}, Bedroom^{-h}, Calendar^{-rv}, Camera^{-h}, Cancel^{-h}, Direction^{-h}, Dim^{-h}, Door^{-rv}, DoorBell^{-rh}, Email^{-v}, Event^{-rh}, Food^{-rv}, Game^{-rh}, Good morning^{-v}, Heat^{-v}, House^{-h}, How^{-v}, Kitchen^{-h}, Light^{-rh}, List^{-rh}, Lock^{-v}, Message^{-h}, Movie^{-rv}, Night^{-rh}, Order^{-v}, Phone^{-rh}, Picture^{-v}, Place^{-rv}, Play^{-h}, Rain^{-rv}, Raise^{-h}, Restaurant^{-h}, Room^{-h}, Schedule^{-h}, Shopping^{-rv}, Snooze^{-h}, Snow^{h}, Stop^{-v}, Sunny^{-h}, Temperature^{-rh}, Time^{-rh}, Today^{-h},Traffic^{-rv}, Turn down^{-h}, Turn Off^{-h}, Turn on^{-h}, Wake-up^{-h}, Weather^{-h}, Weekend^{-h}		
Data Samples	Train/Test with Same User	User A-User G (7)	Scenario 1: user at 0° Training : 7 Users x 50 Signs x 17 Instances = 5950 Samples Test : 7 Users x 50 Signs x 7 ± 1 Instances = 2332 Samples
	Cross Subject	User H-User K (4)	Scenario 1: user at 0° 4 Users x 17 ± 2 Signs x 22 ± 1 Instances =1476 Samples
	User Diversity Test	User L-User O (4)	Scenario 1: user at 0° 4 Users x 50 Signs x 10 ± 1 Instances = 2198 Samples
	Other Test Scenarios (not in training)	User A	Scenario-1: 7 Signs x 20 Instances = 140 Samples,
		Users A and B	Scenario-3: 4 Signs x 20 Instances = 80 Samples
			Scenario-5: 3 Signs x 20 Instances = 60 Samples
Rooms	6 different rooms	Lab (6.5m x 3m): 40% of data, Dept. Conf. room (16m x 10m): 30% Data, 4 Univ. Conf. rooms (18m x 12m): 30% Data	
Total Samples for ASL Signs (Each 6 Sec long)			12,236

activities (e.g., drinking water, using phone, putting on coat, etc.) while walking or standing. For each sample in any scenario, the user performs the wake-word while mmASL switches between the predetermined set of beams and collects the data. We do not constrain the user to perform the wake-word at any particular speed, resulting in partial wake-words depending on gesture speed and scanning time as discussed in Section 4.2. For scenarios involving an interfering user in addition to the intended user, the interfering user performs the random activities throughout the sample collection in an arbitrary order. We also collect samples with different beam scanning times (1, 2, and 3 seconds) and different wake-words (`KNOCK` –like knocking a door twice and `UP-DOWN` –like calling someone with both hands) to study the impact of scanning time and wake-word choice on mmASL's performance.

ASL sign dataset: Table 2 summarizes our ASL sign dataset. We select ASL signs for 50 words commonly used in interaction with home assistants like Amazon echo [4]. The signs and their phonological properties are shown in Table 2. We performed data collection in two phases. In Phase-I, we recruited 11 participants (8 male and 3 female) and collected data simultaneously using mmASL, Kinect and RGB camera. The data collected using the Kinect system includes 25 body joint coordinates in 3D over time. It also includes RGB video recordings of

the person performing the gesture with 60 frames per second. As Kinect system has an inbuilt RGB camera and uses the RGB video along with the depth data in determining the different body joints, we were able to collect data for both the modalities using the same system. We do this to compare mmASL's performance with well established sensing modalities like Kinect and camera. More details of the accuracy comparison are provided in Section 6.3. In Phase-II, we collected data for the same set of 50 ASL signs from 4 additional participants using mmASL. This additional dataset is used to evaluate the performance of mmASL in terms of different training and testing splits, diversity in user's signing and cross-subject accuracy. In both phases, each participant stands in front of the system (at distances varying from 10-12 feet) and performs the gesture. In Phase-I, we were only able to collect 17±2 gestures for some users due to availability constraints. The experiments in Phase-I were conducted in six different rooms as shown in Table 2, while the data for Phase-II was collected in a single room (refer Figure 15c). We also make sure that gesture instances for each user are collected in different rooms to validate mmASL's robustness to change in environment. Additionally, we collect data in multi-person scenarios (Scenarios 2 and 3 shown in Figs. 14b and 14c) to evaluate the impact of presence of an interfering user.

Model implementation: The signal processing modules are implemented on NI+SiBeam FPGA, host (LabView/Windows) and a desktop computer running Linux. We implement the deep learning models on computing cluster nodes [11] with NVIDIA Tesla K480 GPU with 1.87 TFLOPS (Tera floating point operations per second), 480 GB/s of GPU memory bandwidth, and 48 GB of memory [2]. We use Tensorflow[17] framework for developing the deep learning models. The approximate training time for the deep learning models on the entire training set for ASL sign recognition on a node with single GPU (11 GB memory) is approximately 2 hour. We test the model on a desktop running Linux which approximately takes 49 seconds for 1978 samples ($\approx 24ms$ per test sample).

6.2 Wake-word Detection

We now evaluate the impact of scanning time, choice of wake-word, and the machine learning model on wake-word detection accuracy. For choice of wake-word and scanning time evaluation, we use CNN as the model for comparison. In choosing the scanning time, we compare 1, 2, and 3 seconds keeping in mind that more than 3 seconds of response time is likely to impact user experience. We compare three machine learning models (1) SVM, (2) Random Forest (RF) with PCA for feature reduction for both (we choose the top 75 features which explain 95% variance) and (3) CNN. The model utilized for wake-word recognition should make the predictions within the chosen scanning time ($\leq$ 3 seconds). To achieve the predictions efficiently, we choose a vanilla CNN [10] with only two layers, relatively less number of filters (32 and 64 for first and second layer) and smaller convolutional kernels (3×3). The vanilla CNN model is based on AlexNet [42] (convolutional layers, followed by dense and softmax layers for prediction) and was optimized using Adam optimizer [39] with a learning rate of 0.0001 and batch size of 10. We do not observe any significant improvement in performance when changing the different hyperparameters of the model. We note that although wake-word detection uses vanilla CNN, the ASL sign recognition model described in Section 5.2 has 3 layers, with filter count of 32, 64 and 128 (for each layer) and convolutional kernel sizes varying from 5×2 to 20×8. For SVM and random forest, we perform a grid search with 10-fold cross validation to determine the optimal parameters. We use User A's data collected in R1 (1590 samples) as training data (for all scenarios) and test on other users and rooms. This is to verify that the wake-word detection is robust for untrained users and environments. We also include instances without user and single user performing non-wake word activities to evaluate false positive rate (FP rate).

Wake-word selection: We compare two wake-words KNOCK and UP-DOWN (involves moving hands up and down twice). Both wake-words have repetitive motion and have movements which symbolize calling someone. Fig. 16a compares the wake-word detection accuracy, we find that UP-DOWN has a higher accuracy with lower FP rate (other activities classified as wake-word). This can be attributed to its higher motion intensity compared to

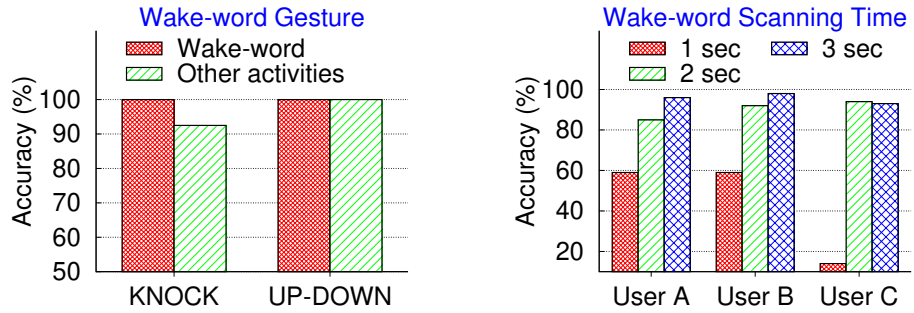


Fig. 16. Impact of choice of wake-word and scanning time on wake-word detection accuracy

KNOCK. Given that UP-DOWN performs better as a wake-word, we only present the results for KNOCK (worse case) due to space limitation.

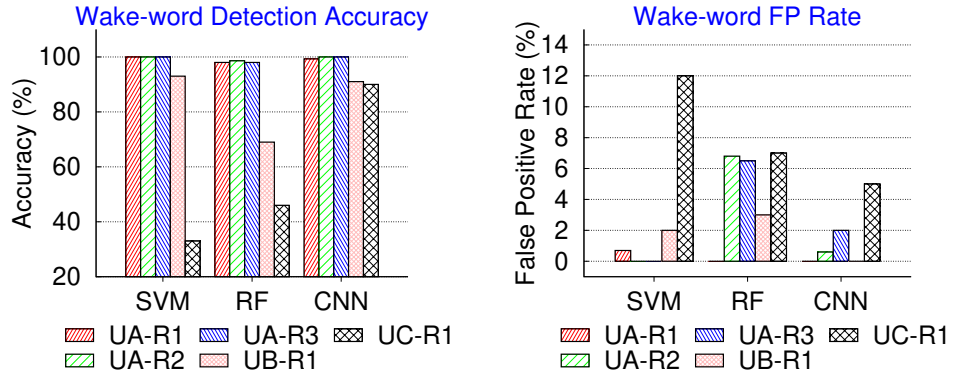


Fig. 17. Wake-word detection accuracy and false postive (FP) rate: Model trained on User A in Room 1 (UA-R1), and tested on other rooms and users (UB-R1, UC-R1, UA-R2 and UA-R3). All wake-word samples collected at random locations as in shown in Scenario-1 (Fig. 14a).

Impact of scanning time: Fig. 16b shows the detection accuracy for different scan times for different users. It is evident that scanning time of 1 second ($\approx 167ms$ per sector) is certainly not sufficient as it reduces the wake-word detection accuracy for all users. The detection accuracy is the lowest for User C because the user performs the wake-word gestures relatively slow compared to others. Increasing the scanning time to 2 and 3 seconds improves the wake-word detection in most cases. We use 3 seconds as the scanning time for the remaining analysis.

Machine Learning model comparison: Figs. 17a and 17b compare the three machine learning models in terms of accuracy and FP rate. We observe that all three models perform comparably well with User A even in rooms not included in training (R2: Lab and R3: Classroom). Random forests suffer from high FP rate compared to SVM and CNN which is likely due to over-fitting with decision trees. In case of untrained users, both SVM and RF suffer in terms of accuracy and FP rate compared to CNN. The better performance of CNNs is due to their ability to learn better feature representations which generalize well for new users. Given the better performance of CNNs, we use them in further evaluations. We note that the ability of other models to perform reasonably well in

most of the scenarios even with a small feature space, further quantifies the importance of spatial spectrograms in solving this problem.

Testing with untrained users and rooms: Fig. 18a shows the wake-word detection and sector determination accuracy with the CNN model applied on spatial spectrograms. We find that mmASL can detect wake-word with an average accuracy of 94% for different (and untrained) users and environments. It can also determine the sector with an average accuracy of 95%. The higher false positive rate (Fig. 17b) and lower wake-word detection accuracy in case of User C is due to him performing the wake-word slowly, resulting in a higher number of partial gestures.

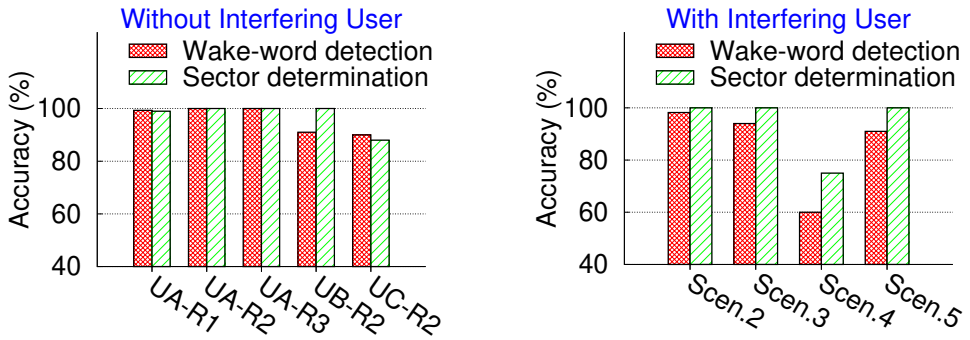


Fig. 18. Wake-word detection and sector determination for (a) single person (different users in different rooms), (b) multi-person Scenarios 2-5 (interfering user present).

Impact of an interfering user: Here we evaluate Scenarios 2-5 (Fig. 14) where another user is present (in the same room) whose activities "interfere" with wake-word detection. Fig. 18b shows the results for different scenarios. In Scenario 2 where the interferer is standing in the same beam sector as the intended user, performing either random activity or being still, the average accuracy of wake-word detection and sector determination are 98.2% and 100%, respectively. This highlights the little impact the interferer has in this scenario. In Scenario 3 where interfering user walks in the background, the wake-word detection accuracy reduces to 94% given that higher motion intensity of walking and taking turns sometimes reduces how clearly wake-word signatures are observed. In Scenario 5, where the interfering user is walking in different sector(s), the wake-word detection accuracy is 91%. The interferer is closer to the user as well as mmASL in Scenario 5 compared to Scenario 3, leading to a slight deterioration in performance. We note that sector determination has higher accuracy than wake-word detection due to the fact that sector determination is a relatively simpler task (only looking for the most probable sector given a spatial spectrogram sample) compared to wake-word detection (distinguish wake-word from walking, standing and other random activities with a lower FP rate). Lastly, in Scenario 4 where the interfering user is walking in between the intended user and mmASL at normal walking speed (0.95 m/s), there are instances where the wake-word is not properly observed due to blockage of 60 GHz signal by the interferer. Here, the wake-word detection and sector determination accuracy are 60% and 75% respectively.

In evaluating mmASL with different wake-words and scanning times, and in different environments and scenarios, we find that user variability in terms of gesture speed can be an important factor affecting the performance of wake-word detection. It can be addressed by incorporating more diverse user data in the training. On the other hand, the directional nature of mmWave systems make it possible to not only detect wake-word occurrence but also locate user at the same time. We also find that mmWave sensing systems can tolerate the presence of other people and impact of the environment is minimal compared to other low-frequency RF sensing

systems. mmASL is prone to blockage (Scenario 4) only when the interfering user is on the LoS path between the transceiver and the intended user. Vision-based systems are also prone to such occlusions, often requiring multiple cameras [63, 70] for sensing. In the case of mmWave, it is possible to exploit reflections from indoor objects (e.g., walls) to circumvent the blockages while sensing. Use of more than one transceiver along with sensing over reflected paths could make mmWave sensing robust against blockages. We observe that mmASL being a digital assistant, there is a trade-off between availability and false positive rate (FPR). This means that the wake-word detection model can be tuned either for higher availability or lower FPR. Accommodating user's preferences based on contemporary voice-based digital assistants can help in addressing the trade-off.

6.3 ASL Sign Recognition

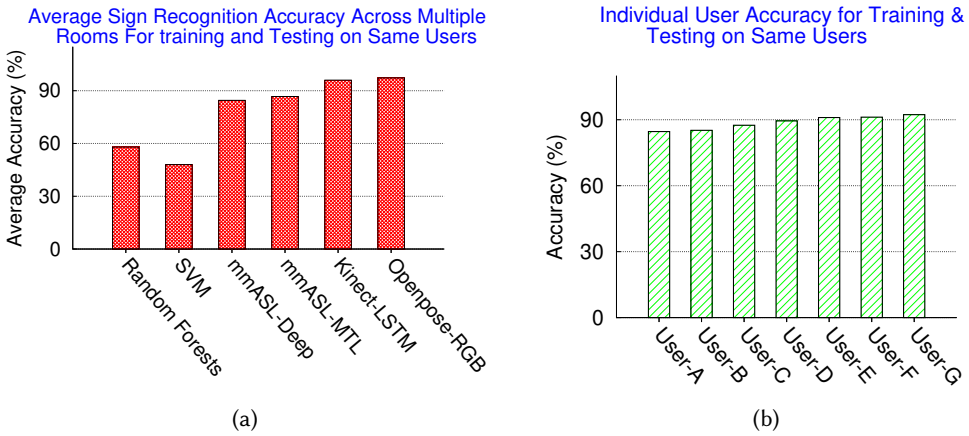


Fig. 19. (a) Average sign recognition accuracy (b) Individual user accuracy

As shown in Table 2, we perform training and testing with the same subset of users, different subset of users (cross-subject), include additional test scenarios with an interfering user, and study the impact of adding more users (and data) on mmASL. We train two sign recognition models using data from 7 users (Users A-User G). These models are referred as mmASL-Deep (deep without multi-task learning) and mmASL-MTL (deep with 2 auxiliary tasks - repetitive and motion direction). In addition to the deep learning models, we also evaluate using random forests (RF) and support vector machine (SVM) models. For both the models, we use principal component analysis (PCA) for dimensionality reduction of the spectrogram images and choose the top components explaining 95% variance as features. The parameters for both the models are determined using grid search with 10 fold cross validation. We compare the performance of these models (deep learning and non-deep learning) with Kinect and RGB camera. For the Kinect data, we use a well-studied learning model proposed in [31, 76, 105] for comparison. The model uses LSTM (Long Short-Term Memory) based hierarchical recurrent neural network with 4 joints data (left wrist, right wrist, elbow left, elbow right) and is shown to achieve superior performance in skeleton based gesture recognition. For camera data, we first perform frame-level pose estimation on the video using OpenPose [25], which results in 48 joint location per frame (6 for the wrist, elbow, and shoulder for both the hands, and remaining 42 are palm and finger level joints). Because of its ability to perform fine-grained pose estimation from images, OpenPose has been previously utilized for ASL sign recognition from continuous videos [29, 40, 98]. We adopt the LSTM models proposed in [40, 98]. The model takes 48 joint coordinates (obtained from OpenPose) as input at each time step and predicts the corresponding ASL sign (with respect to the input) in the final time step.

Recognition accuracy with same users in different rooms: In case of training and testing with the same user (75%-25% training-testing split), to test for the environment independence we chose the test samples with the following constraint. Out of the test data chosen for each user, at least 80% of the test samples were chosen from a room different than the room where the training instances were collected. We are able to do this as we collected instances of each gesture in different rooms as explained in Sec 6.1.

Figs. 19a and 19b show that mmASL-MTL performs marginally better (86.7%) than mmASL-Deep (84.5%). This is expected given that the true advantage of multi-task learning comes with data from untrained users and scenarios. We find that mmASL is robust to change in environment and it is possible to apply model trained in one room to test in other rooms and achieve a reasonably high accuracy. This can be attributed to the directionality of 60 GHz transceivers which reduce the impact of multi-path while capturing motion signatures for signs. Compared to the two models, Kinect LSTM model and OpenPose RGB model achieve an accuracy of 96% and 97.27% respectively. This shows that mmASL can achieve a reasonable performance in sign recognition compared to well-studied Kinect and camera based systems. We also observe that signs which have similar motion (`NIGHT` and `TIME` both involve moving the right hand on left hand twice) are often misclassified. However, we believe that if a relevant phonological property can be identified for distinguishing minor differences between such signs, it is possible to improve the classification performance. In contrast to the deep learning models, random forests (RF) and SVM models achieve relatively lower accuracy, 58% and 48% respectively. This is expected given that deep learning models which can learn feature representations from the data perform better compared to traditional machine learning models requiring feature engineering.

Testing with untrained users (cross-subject): We now take the models developed with 7 users and test

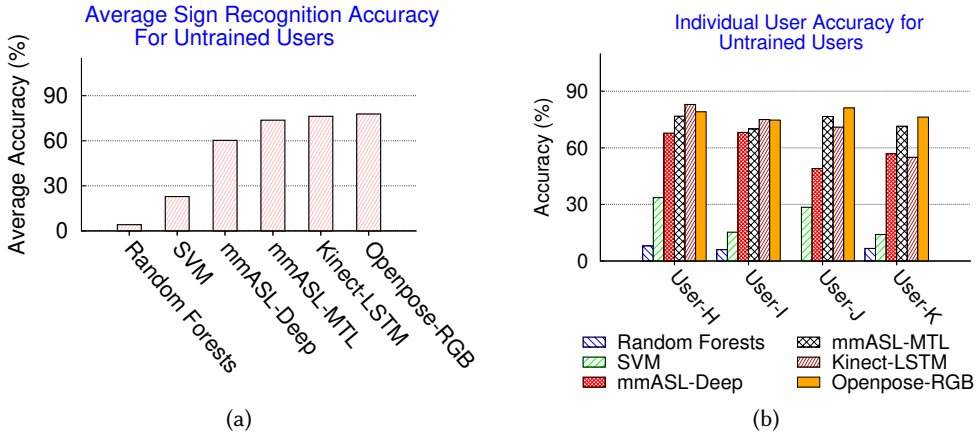


Fig. 20. (a) Average user accuracy and (b) Individual user accuracy for untrained users.

them on 4 additional users added in Phase-I. Fig. 20a shows the average accuracy and Fig. 20b shows the individual user accuracy for different models. As we can observe, Kinect and RGB camera based models offer comparable performance with 76.3% and 77.8% accuracy respectively. Compared to the Kinect-LSTM, RGB camera based model (OpenPose-RGB) provides consistently better performance across all the users (refer Figure 20b), because of the availability of palm and finger joint details. Of the two mmASL models, mmASL-MTL performs significantly better, establishing the significance of the added auxiliary tasks. Specifically, the gap in average accuracy between mmASL and Kinect model reduces from 16.9% to 2.6% as we move from mmASL-Deep to mmASL-MTL. This further validates the need and effectiveness of multitask learning in mmASL. Also, mmASL-MTL consistently

outperforms mmASL-Deep for every untrained user. Both SVM and random forests suffer a decline in performance (in contrast to training and testing on the same user). Compared to SVM, random forests perform poorly which could be the effect of overfitting inherent to the model.

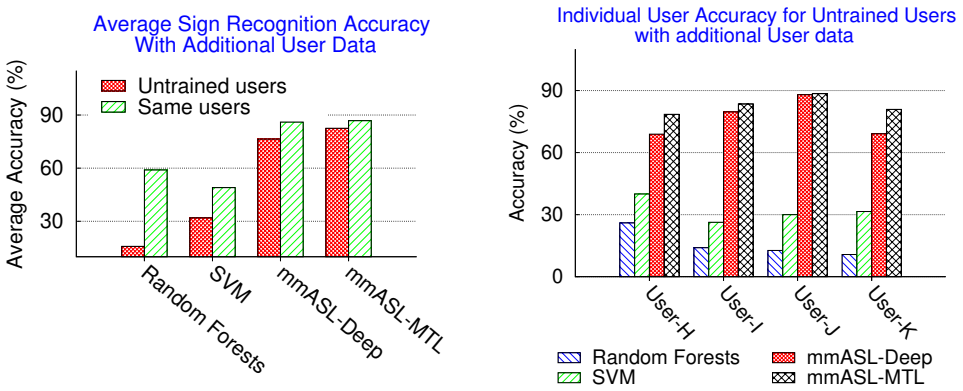


Fig. 21. Impact of including additional user data on mmASL's performance. a) Average user accuracy for same and untrained users b) Individual user accuracy for untrained users

Impact of additional users: We observe that similar to RGB camera and Kinect, the performance of mmASL reduces in the cross-subject scenario. This could mean that the trained model has high variance i.e., variation in the training data has an impact on the performance. A common way to address this issue is to add more (and diverse) data in the training set. To test this hypothesis, we use the data collected for 4 additional users (User L-User O in Phase-II, refer Table 2). Here, we add 75% of their data to the training set of 7 users (User A-User G), remaining is added to the test set. We train the four models (mmASL-MTL, mmASL-Deep, random forests, and SVM) with the new training data and test the models on (i) data from the same 11 users and (ii) data from 4 untrained users (cross-subject). Figs. 21a and 21b show the performance of the trained models when tested on the same users (User A-User G and User L-User O) and untrained users (User H-User K), respectively.

When tested on the same 11 users, all the four models provide similar results as trained on 7 users, confirming that mmASL can scale for more users without any significant decline in performance. The evidence for the high variance of the model is observed when tested on untrained users where all the four models have gained significant benefit from the additional data. Specifically, the average accuracy for random forests and SVM has increased by 11% and 9% respectively. Of the deep learning models, mmASL-Deep has gained an increase of 16% and mmASL-MTL has gained an increase of 9% in the average accuracy, when compared to models trained with 7 users' data. Although the accuracy gap between mmASL-MTL and mmASL-Deep reduces with additional data, it is worth noting that the multitask learning is still needed in order for mmASL to scale for more number of signs.

Impact of user position and interfering user: As shown in Table 2, we consider additional test scenarios (Scenarios 1 (random locations), 2 and 3). Fig. 22a shows the performance of mmASL in all the scenarios. For Scenario 1, mmASL achieves an average accuracy of 78% for randomly chosen user locations in a room. With change in user location (distance and angle) relative to the Tx and Rx, the observed doppler shift and reflected power also changes. mmASL-MTL is resilient to such changes, while mmASL-Deep suffers drop in recognition accuracy. In contrast to the deep learning models, random forests and SVM models perform poorly with accuracy of 27% and 7% respectively. When an interferer is walking in the background (Scenario 3), the sign recognition accuracy is 78%. With distance separation between the user and the interferer and the blockage property of 60

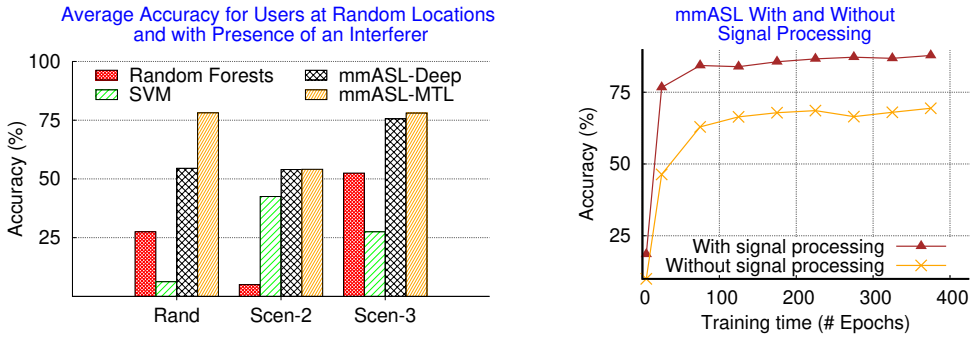


Fig. 22. (a) Average ASL sign recognition accuracy at random user locations (Rand) and with presence of interferer in the same beam (Scen-2) and an interfering user walking behind (Scen-3) as shown in Fig. 14 (b) Impact of singal processing on mmASL's performance

GHz signals, the interferer does not have significant impact on recognition accuracy. On the other hand, when the interfering user is in the same beam as the intended user (Scenario 2), there is a significant drop in accuracy. This is expected given that the reflections capture both sign and interfering motions. We observe similar drop in performance between Scenario 3 and Scenario 2 for random forests. While SVM performs better (still less than deep learning models) in the Scenario 2, there is a drop in accuracy in Scenario 3. The decline in performance in both these models can be attributed to the testing scenarios being completely different from the training data scenario. We note that because of directionality and use of receive beamforming, presence of interferer in other beams (Scenario 5) does not have a significant impact on sign recognition. This is different from wake-word detection with an interferer being present in other beams as wake-word detection requires continuous scanning of all sectors.

Impact of Signal Processing: To understand the significance of signal processing which we established in Section 3.2, we train two instances of mmASL-MTL model - one is trained using data processed with the complete signal processing pipeline, while the other is trained using data without signal processing (specifically no log normalization). Both the models were optimized using Adam optimizer [39] with a learning rate of 0.0001 and batch size of 10 for 400 epochs (epoch – time required to perform training on the complete training set once). Fig. 22b shows the performance of these identical models when trained on data with and without signal processing. We compare the accuracy of the models over multiple epochs as they are trained for 400 epochs. It is clear from the figure that signal processing is crucial for mmASL's performance and the model utilizing data with signal processing consistently performs better (15-20% increase in accuracy) than model utilizing data without signal processing.

After extensive evaluation of mmASL for ASL sign recognition under multiple scenarios, we make the following observations. First, mmASL offers comparable performance with state of the art systems like Kinect and camera. This further validates that it is indeed feasible to use 60 GHz mmWave signals for gesture and activity recognition even at larger distances (compared to short-range solutions such as [48]). Second, we find that similar to wake-word detection, ASL recognition is also tolerant to the presence of other people and change of environment. In contrast to wake-word detection, the presence of another user in the same beam can deteriorate ASL recognition accuracy. We note that the difference in performance can be attributed to the difference in complexity between the two problems (wake-word recognition involves a binary classification while ASL sign recognition is a multiclass –50 classes– classification problem). Also, signal processing is a significant contributor to the performance of mmASL, and simple normalization techniques (log normalization) can be helpful in defending against user

diversity (variation in gesture speed and intensity among different users). mmASL shows that learning on spectrogram representation can be effective and can provide high accuracy. That combined with multi-task learning can be key to scaling mmASL for a large number of ASL signs. It is worth noting that capturing more spatial information such as precise distance and angle of body parts through the mmWave radar can further enhance the recognition accuracy.

7 DISCUSSION

We now discuss various aspects of mmASL that can be improved through further investigation:

60 GHz blockage: As observed in the evaluation, when the intended user is blocked by an interfering user, wake-word recognition accuracy decreases. Such occlusions are also a problem in vision-based systems. Given that 60 GHz blockage has been well-studied recently in networking [34, 82, 94, 101], use of reflections (second order reflections are likely to be weak) or multiple transmitters can be exploited to improve the performance.

Detecting handshape: One reason of classification error in mmASL is signs that have similar hand movements but different handshape (e.g., NIGHT and TIME) cannot be accurately distinguished. While mmASL cannot extract handshape, it would be interesting to augment mmASL with synthetic aperture radar based imaging to estimate handshapes and improve the classification performance.

Orientation: We note that ASL being a visual language, speakers are accustomed to facing the listener (home assistant device) during conversation. In our experiments, the users were facing the system with $\pm 10^{\circ}$ variation in orientation. For such variations, mmASL recognition performance was observed to be robust. However, consideration of other orientations and their impact on Doppler spread should be studied in more details.

Codebook and Interference: mmASL adopts the codebook designed for communication, which utilizes relatively wider beams (with a 3-dB beamwidth of 25° to 30° for Tx and 30° to 35° for Rx) to provide the needed coverage. Instead of using existing codebooks, designing codebooks specifically for gesture sensing (e.g., increasing recognition range with narrower beams) might improve mmASL's performance. In addition, the presence of other mmWave based systems (e.g., 802.11ad WLANs) can interfere with mmASL. Although with directionality, such interference is likely to be less and existing interference mitigation techniques (e.g., use of non-interfering channels) can be employed to address the problem.

Commodity devices: mmASL is designed using access to raw I/Q samples from the 60 GHz software radio system. Current 60 GHz 802.11ad commodity devices do not allow access to such information (in user-space, similar to 2.4/5 GHz WiFi CSI) from firmware/driver. Upon availability of such information, underlying techniques of mmASL can be adapted for sign recognition using 60 GHz commodity devices.

Beyond manual signs: In this work, we showed mmASL can detect wake-words and recognize 50 signs. While this is significant, mmASL should scale from recognizing individual signs to contextual non-manual grammar markers to provide the true context to sentences. The non-manual grammar markers include head and torso movements which can be detected through Doppler spreads but requires further extensions to our proposed models.

8 CONCLUSIONS

In this work, we proposed mmASL, a 60GHz mmWave based home assistant for DHH users. We utilized beam scanning to generate spatial spectrograms, which can be used in detecting wake-words and choosing the beam sector. We established that variation in Doppler spread can be captured in spectrograms to recognize ASL signs. We proposed a multi-task deep learning architecture, which can learn ASL domain specific features from the spectrograms. We compared the performance of mmASL with Kinect and RGB camera and find that mmASL can achieve accurate sign recognition for a variety of practical scenarios including presence of other interfering user, change of environment and different user positions.

ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers for their valuable comments and feedback. This research is supported by NSF grant CNS-1730083 and Google Faculty Research grant.

REFERENCES

[1] 2014. National Instruments mmWave Transceiver System. Retrieved January 1, 2020 from http://www.ni.com/sdr/mmwave/
[2] 2016. Nvidia Tesla 480 Specifications. Retrieved January 1, 2020 from https://en.wikipedia.org/wiki/Nvidia_Tesla
[3] 2017. How Amazon's Alexa is helping with child disability. Retrieved January 1, 2020 from https://themighty.com/2017/11/amazon-alexa-helping-child-disability/
[4] 2018. 50 Best Alexa Commands. Retrieved January 1, 2020 from https://beebom.com/best-alexa-commands-for-amazon-echo/
[5] 2018. Home alerting devices for people who are deaf or hard of hearing. Retrieved January 1, 2020 from https://tap.gallaudet.edu/smarthome/
[6] 2018. IEEE 802.11ay: Enhanced Throughput for Operation in License-Exempt Bands above 45 GHz. Retrieved January 1, 2020 from http://www.ieee802.org/11/Reports/tgay_update.htm
[7] 2018. SiBeam Beam-steering Transceivers. Retrieved January 1, 2020 from http://www.sibeam.com/Products.aspx
[8] 2018. The Smart Devices Transforming the Lives of People with Disabilities. Retrieved January 1, 2020 from https://www.mytherapyapp.com/blog/smart-homes-for-living-with-disabilities
[9] 2018. Smart Speaker Users Growing 48% Annually, To Hit 90M In USA This Year. Retrieved January 1, 2020 from https://www.forbes.com/sites/johnkoetsier2018/05/29/smart-speaker-users-growing-48-annually-will-outnumber-wearable-tech-users-this-year/
[10] 2018. Tensorflow CNN example. Retrieved January 1, 2020 from https://www.tensorflow.org/tutorials/images/cnn
[11] 2019. ARGO: A research computing cluster. Retrieved January 1, 2020 from http://orc.gmu.edu/
[12] 2019. ASL Sign for email. Retrieved January 1, 2020 from https://www.handspeak.com/word/search/index.php?id=659
[13] 2019. ASL Sign for place. Retrieved January 1, 2020 from https://www.signingsavvy.com/sign/PLACE
[14] 2019. ASL Sign for shopping. Retrieved January 1, 2020 from https://www.handspeak.com/word/search/index.php?id=1948
[15] 2019. ASL Sign for snow. Retrieved January 1, 2020 from https://www.handspeak.com/word/search/index.php?id=2003
[16] 2019. mmASL dataset. Retrieved January 1, 2020 from https://cs.gmu.edu/~phpathak/datasets/mmASL.html
[17] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: A System for Large-scale Machine Learning. In *Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation* (Savannah, GA, USA) *(OSDI'16)*. USENIX Association, Berkeley, CA, USA, 265–283. http://dl.acm.org/citation.cfm?id=3026877.3026899
[18] H. Abdelnasser, M. Youssef, and K. A. Harras. 2015. WiGest: A ubiquitous WiFi-based gesture recognition system. In *2015 IEEE Conference on Computer Communications (INFOCOM)*. 1472–1480. https://doi.org/10.1109/INFOCOM.2015.7218525
[19] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke Recognition Using WiFi Signals. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking* (Paris, France) *(MobiCom '15)*. ACM, New York, NY, USA, 90–102. https://doi.org/10.1145/2789168.2790109
[20] Marco Altini, Julien Penders, and Oliver Amft. 2012. Energy Expenditure Estimation Using Wearable Sensors: A New Methodology for Activity-specific Models. In *Proceedings of the Conference on Wireless Health* (San Diego, California) *(WH '12)*. ACM, New York, NY, USA, Article 1, 8 pages. https://doi.org/10.1145/2448096.2448097
[21] Oya Aran, Thomas Burger, Alice Caplier, and Lale Akarun. 2009. A belief-based sequential fusion approach for fusing manual signs and non-manual signals. *Pattern Recognition* 42, 5 (2009), 812 – 822. https://doi.org/10.1016/j.patcog.2008.09.010
[22] B B Blanchfield, J J Feldman, J L Dunbar, and E N Gardner. 2001. The severely to profoundly hearing-impaired population in the United States: prevalence estimates and demographics. *Journal of the American Academy of Audiology* 12, 4 (2001), 183–9. http://www.ncbi.nlm.nih.gov/pubmed/11332518
[23] H. Brashear, T. Starner, P. Lukowicz, and H. Junker. 2003. Using multiple sensors for mobile sign language recognition. In *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.* 45–52. https://doi.org/10.1109/ISWC.2003.1241392
[24] Diane Brentari. 1998. *A prosodic model of sign language phonology.* Mit Press.
[25] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. *CoRR* abs/1812.08008 (2018). arXiv:1812.08008 http://arxiv.org/abs/1812.08008
[26] Naomi K. Caselli, Zed Sevcikova Sehyr, Ariel M. Cohen-Goldberg, and Karen Emmorey. 2016. ASL-LEX: A lexical database of American Sign Language. *Behavior Research Methods* (2016), 1–18. https://doi.org/10.3758/s13428-016-0742-0
[27] Yuanying Chen, Wei Dong, Yi Gao, Xue Liu, and Tao Gu. 2017. Rapid: A Multimodal and Device-free Approach Using Noise Estimation for Robust Person Identification. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 41 (Sept. 2017), 27 pages.

https://doi.org/10.1145/3130906

[28] C. Chuan, E. Regina, and C. Guardino. 2014. American Sign Language Recognition Using Leap Motion Sensor. In *2014 13th International Conference on Machine Learning and Applications*. 541–544. https://doi.org/10.1109/ICMLA.2014.110

[29] Cleison Correia de Amorim, David Macêdo, and Cleber Zanchettin. 2019. Spatial-Temporal Graph Convolutional Networks for Sign Language Recognition. *CoRR* abs/1901.11164 (2019). arXiv:1901.11164 http://arxiv.org/abs/1901.11164

[30] Cao Dong, M. C. Leu, and Z. Yin. 2015. American Sign Language alphabet recognition using Microsoft Kinect. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 44–52. https://doi.org/10.1109/CVPRW.2015.7301347

[31] Yong Du, W. Wang, and L. Wang. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1110–1118. https://doi.org/10.1109/CVPR.2015.7298714

[32] Biyi Fang, Jillian Co, and Mi Zhang. 2017. DeepASL: Enabling Ubiquitous and Non-Intrusive Word and Sentence-Level Sign Language Translation. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems* (Delft, Netherlands) *(SenSys '17)*. ACM, New York, NY, USA, Article 5, 13 pages. https://doi.org/10.1145/3131672.3131693

[33] Xiaonan Guo, Bo Liu, Cong Shi, Hongbo Liu, Yingying Chen, and Mooi Choo Chuah. 2017. WiFi-Enabled Smart Human Dynamics Monitoring. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems* (Delft, Netherlands) *(SenSys '17)*. ACM, New York, NY, USA, Article 16, 13 pages. https://doi.org/10.1145/3131672.3131692

[34] Muhammad Kumail Haider and Edward W. Knightly. 2016. Mobility Resilience and Overhead Constrained Adaptation in Directional 60 GHz WLANs: Protocol Design and System Implementation. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Paderborn, Germany) *(MobiHoc '16)*. ACM, New York, NY, USA, 61–70. https://doi.org/10.1145/2942358.2942380

[35] Jiahui Hou, Xiang-Yang Li, Peide Zhu, Zefan Wang, Yu Wang, Jianwei Qian, and Panlong Yang. 2019. SignSpeaker: A Real-time, High-Precision SmartWatch-based Sign Language Translator. In *To appear in Mobicom 2019* (Los Cabos, Mexico).

[36] Jie Huang, Wengang Zhou, Houqiang Li, and Weiping Li. 2015. Sign Language Recognition using 3D convolutional neural networks. In *2015 IEEE International Conference on Multimedia and Expo (ICME)*. 1–6. https://doi.org/10.1109/ICME.2015.7177428

[37] IEEE P802.11adTM/D4.0. 2012. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 3: Enhancements for Very High Throughput in the 60 GHz Band, IEEE Computer Society. *IEEE Computer Society* (July 2012).

[38] Wenjun Jiang, Chenglin Miao, Fenglong Ma, Shuochao Yao, Yaqing Wang, Ye Yuan, Hongfei Xue, Chen Song, Xin Ma, Dimitrios Koutsonikolas, Wenyao Xu, and Lu Su. 2018. Towards Environment Independent Device Free Human Activity Recognition. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking* (New Delhi, India) *(MobiCom '18)*. ACM, New York, NY, USA, 289–304. https://doi.org/10.1145/3241539.3241548

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

[40] Sang-Ki Ko, Jae Gi Son, and Hyedong Jung. 2018. Sign Language Recognition with Recurrent Neural Network Using Human Keypoint Detection. In *Proceedings of the 2018 Conference on Research in Adaptive and Convergent Systems* (Honolulu, Hawaii) *(RACS '18)*. ACM, New York, NY, USA, 326–328. https://doi.org/10.1145/3264746.3264805

[41] V. E. Kosmidou and L. J. Hadjileontiadis*. 2009. Sign Language Recognition Using Intrinsic-Mode Sample Entropy on sEMG and Accelerometer Data. *IEEE Transactions on Biomedical Engineering* 56, 12 (Dec 2009), 2879–2890. https://doi.org/10.1109/TBME.2009.2013200

[42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 1097–1105. http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[43] A. Kuznetsova, L. Leal-TaixÃĺ, and B. Rosenhahn. 2013. Real-Time Sign Language Recognition Using a Consumer Depth Camera. In *2013 IEEE International Conference on Computer Vision Workshops*. 83–90. https://doi.org/10.1109/ICCVW.2013.18

[44] S. Lawrence, C. L. Giles, Ah Chung Tsoi, and A. D. Back. 1997. Face recognition: a convolutional neural-network approach. *IEEE Transactions on Neural Networks* 8, 1 (Jan 1997), 98–113. https://doi.org/10.1109/72.554195

[45] Greg C. Lee, Fu-Hao Yeh, and Yi-Han Hsiao. 2016. Kinect-based Taiwanese sign-language recognition system. *Multimedia Tools and Applications* 75, 1 (01 Jan 2016), 261–279. https://doi.org/10.1007/s11042-014-2290-x

[46] Hong Li, Wei Yang, Jianxin Wang, Yang Xu, and Liusheng Huang. 2016. WiFinger: Talk to Your Smart Devices with Finger-grained Gesture. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. ACM, New York, NY, USA, 250–261. https://doi.org/10.1145/2971648.2971738

[47] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar. *ACM Trans. Graph.* 35, 4, Article 142 (July 2016), 19 pages. https://doi.org/10.1145/2897824.2925953

[48] Jaime Lien, Nicholas Gillian, M. Emre Karagozler, Patrick Amihood, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. 2016. Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar. *ACM Trans. Graph.* 35, 4, Article 142 (July 2016), 19 pages. https://doi.org/10.1145/2897824.2925953

[49] Frank R Lin, John K Niparko, and Luigi Ferruci. 2011. Hearing loss prevalence in the United States. *Archives of internal medicine* 171, 20 (2011), 1851–1853.

[50] Jingjing Liu, Bo Liu, Shaoting Zhang, Fei Yang, Peng Yang, Dimitris N. Metaxas, and Carol Neidle. 2014. Non-manual grammatical marker recognition based on multi-scale, spatio-temporal analysis of head pose and facial expressions. *Image and Vision Computing* 32, 10 (2014), 671 – 681. https://doi.org/10.1016/j.imavis.2014.02.009 Best of Automatic Face and Gesture Recognition 2013.

[51] Yongsen Ma, Gang Zhou, Shuangquan Wang, Hongyang Zhao, and Woosub Jung. 2018. SignFi: Sign Language Recognition Using WiFi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 2, 1, Article 23 (March 2018), 21 pages. https://doi.org/10.1145/3191755

[52] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki. 2012. Speaker recognition using Mel frequency Cepstral Coefficients (MFCC) and Vector quantization (VQ) techniques. In *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*. 248–251. https://doi.org/10.1109/CONIELECOMP.2012.6189918

[53] Sven L. Mattys, Matthew H. Davis, Ann R. Bradlow, and Sophie K. Scott. 2012. Speech recognition in adverse conditions: A review. *Language and Cognitive Processes* 27, 7-8 (2012), 953–978. https://doi.org/10.1080/01690965.2012.705006

[54] Pedro Melgarejo, Xinyu Zhang, Parameswaran Ramanathan, and David Chu. 2014. Leveraging Directional Antenna Capabilities for Fine-grained Gesture Recognition. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Seattle, Washington) *(UbiComp '14)*. ACM, New York, NY, USA, 541–551. https://doi.org/10.1145/2632048.2632095

[55] Nicholas Michael, Dimitris Metaxas, and Carol Neidle. 2009. Spatial and Temporal Pyramids for Grammatical Expression Recognition of American Sign Language. In *Proceedings of the 11th International ACM SIGACCESS Conference on Computers and Accessibility* (Pittsburgh, Pennsylvania, USA) *(Assets '09)*. ACM, New York, NY, USA, 75–82. https://doi.org/10.1145/1639642.1639657

[56] Ross E Mitchell, Travas A Young, Bellamie Bachleda, and Michael A Karchmer. 2006. How many people use ASL in the United States? Why estimates need updating. *Sign Language Studies* 6, 3 (2006), 306–335.

[57] Bhadragiri Jagan Mohan and Ramesh Babu N. 2014. Speech recognition using MFCC and DTW. In *2014 International Conference on Advances in Electrical Engineering (ICAEE)*. 1–4. https://doi.org/10.1109/ICAEE.2014.6838564

[58] P. Molchanov, S. Gupta, K. Kim, and K. Pulli. 2015. Short-range FMCW monopulse radar for hand-gesture sensing. In *2015 IEEE Radar Conference (RadarCon)*. 1491–1496. https://doi.org/10.1109/RADAR.2015.7131232

[59] Lindasalwa Muda, Mumtaj Begam, and I. Elamvazuthi. 2010. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. *CoRR* abs/1003.4083 (2010). arXiv:1003.4083 http://arxiv.org/abs/1003.4083

[60] Tan Dat Nguyen and Surendra Ranganath. 2011. *Recognizing Continuous Grammatical Marker Facial Gestures in Sign Language Video*. Springer Berlin Heidelberg, Berlin, Heidelberg, 665–676. https://doi.org/10.1007/978-3-642-19282-1_53

[61] M. Oszust and M. Wysocki. 2013. Polish sign language words recognition with Kinect. In *2013 6th International Conference on Human System Interactions (HSI)*. 219–226. https://doi.org/10.1109/HSI.2013.6577826

[62] Avishek Patra, Philipp Geuer, Andrea Munari, and Petri Mähönen. 2018. mm-Wave Radar Based Gesture Recognition: Development and Evaluation of a Low-Power, Low-Complexity System. In *Proceedings of the 2Nd ACM Workshop on Millimeter Wave Networks and Sensing Systems* (New Delhi, India) *(mmNets '18)*. ACM, New York, NY, USA, 51–56. https://doi.org/10.1145/3264492.3264501

[63] C. Piciarelli, C. Micheloni, and G. L. Foresti. 2010. Occlusion-aware Multiple Camera Reconfiguration. In *Proceedings of the Fourth ACM/IEEE International Conference on Distributed Smart Cameras* (Atlanta, Georgia) *(ICDSC '10)*. ACM, New York, NY, USA, 88–94. https://doi.org/10.1145/1865987.1866002

[64] Lionel Pigou, Sander Dieleman, Pieter-Jan Kindermans, and Benjamin Schrauwen. 2015. Sign Language Recognition Using Convolutional Neural Networks. In *Computer Vision - ECCV 2014 Workshops*, Lourdes Agapito, Michael M. Bronstein, and Carsten Rother (Eds.). Springer International Publishing, Cham, 572–578.

[65] J. Pons, T. Lidy, and X. Serra. 2016. Experimenting with musically motivated convolutional neural networks. In *2016 14th International Workshop on Content-Based Multimedia Indexing (CBMI)*. 1–6. https://doi.org/10.1109/CBMI.2016.7500246

[66] N. Praveen, N. Karanth, and M. S. Megha. 2014. Sign language interpreter using a smart glove. In *2014 International Conference on Advances in Electronics Computers and Communications*. 1–5. https://doi.org/10.1109/ICAECC.2014.7002401

[67] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home Gesture Recognition Using Wireless Signals. In *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking* (Miami, Florida, USA) *(MobiCom '13)*. ACM, New York, NY, USA, 27–38. https://doi.org/10.1145/2500423.2500436

[68] N. Pugeault and R. Bowden. 2011. Spelling it out: Real-time ASL fingerspelling recognition. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 1114–1119. https://doi.org/10.1109/ICCVW.2011.6130290

[69] Luis Quesada, Gustavo López, and Luis A. Guerrero. 2015. Sign Language Recognition Using Leap Motion. In *Ubiquitous Computing and Ambient Intelligence. Sensing, Processing, and Using Environmental Information*, Juan M. García-Chamizo, Giancarlo Fortino, and Sergio F. Ochoa (Eds.). Springer International Publishing, Cham, 277–288.

[70] R. Raman, P. K. Sa, and B. Majhi. 2012. Occlusion prediction algorithms for multi-camera network. In *2012 Sixth International Conference on Distributed Smart Cameras (ICDSC)*. 1–6.

[71] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. *CoRR* abs/1602.04938 (2016). arXiv:1602.04938 http://arxiv.org/abs/1602.04938

[72] Sebastian Ruder. 2017. An Overview of Multi-Task Learning in Deep Neural Networks. *CoRR* abs/1706.05098 (2017). arXiv:1706.05098 http://arxiv.org/abs/1706.05098

[73] C. Savur and F. Sahin. 2015. Real-Time American Sign Language Recognition System Using Surface EMG Signal. In *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. 497–502. https://doi.org/10.1109/ICMLA.2015.212

[74] J. Shang and J. Wu. 2017. A Robust Sign Language Recognition System with Sparsely Labeled Instances Using Wi-Fi Signals. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 99–107. https://doi.org/10.1109/MASS.2017.41

[75] J. Shang and J. Wu. 2017. A Robust Sign Language Recognition System with Sparsely Labeled Instances Using Wi-Fi Signals. In *2017 IEEE 14th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*. 99–107. https://doi.org/10.1109/MASS.2017.41

[76] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. 2016. An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data. *arXiv e-prints* (Nov. 2016). arXiv:cs.CV/1611.06067

[77] Thad Starner and Alex Pentland. 1997. Real-time american sign language recognition from video using hidden markov models. In *Motion-Based Recognition*. Springer, 227–243.

[78] T. Starner, J. Weaver, and A. Pentland. 1997. A wearable computer based American sign language recognizer. In *Digest of Papers. First International Symposium on Wearable Computers*. 130–137. https://doi.org/10.1109/ISWC.1997.629929

[79] T. Starner, J. Weaver, and A. Pentland. 1998. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 12 (Dec 1998), 1371–1375. https://doi.org/10.1109/34.735811

[80] T. Starner, J. Weaver, and A. Pentland. 1998. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 12 (Dec 1998), 1371–1375. https://doi.org/10.1109/34.735811

[81] Chao Sun, Tianzhu Zhang, and Changsheng Xu. 2015. Latent Support Vector Machine Modeling for Sign Language Recognition with Kinect. *ACM Trans. Intell. Syst. Technol.* 6, 2, Article 20 (March 2015), 20 pages. https://doi.org/10.1145/2629481

[82] Sanjib Sur, Xinyu Zhang, Parmesh Ramanathan, and Ranveer Chandra. 2016. BeamSpy: Enabling Robust 60 GHz Links Under Blockage. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*. USENIX Association, Santa Clara, CA, 193–206. https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/sur

[83] Sheng Tan and Jie Yang. 2016. 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* (Paderborn, Germany) *(MobiHoc '16)*. ACM, New York, NY, USA, 201–210. https://doi.org/10.1145/2942358.2942393

[84] David Tse and Pramod Viswanath. 2005. *Fundamentals of Wireless Communication*. Cambridge University Press, New York, NY, USA.

[85] Aditya Virmani and Muhammad Shahzad. 2017. Position and Orientation Agnostic Gesture Recognition Using WiFi. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services* (Niagara Falls, New York, USA) *(MobiSys '17)*. ACM, New York, NY, USA, 252–264. https://doi.org/10.1145/3081333.3081340

[86] Guanhua Wang, Yongpan Zou, Zimu Zhou, Kaishun Wu, and Lionel M. Ni. 2014. We Can Hear You with Wi-Fi!. In *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking* (Maui, Hawaii, USA) *(MobiCom '14)*. ACM, New York, NY, USA, 593–604. https://doi.org/10.1145/2639108.2639112

[87] Saiwen Wang, Jie Song, Jaime Lien, Ivan Poupyrev, and Otmar Hilliges. 2016. Interacting with Soli: Exploring Fine-Grained Dynamic Gesture Recognition in the Radio-Frequency Spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology* (Tokyo, Japan) *(UIST '16)*. ACM, New York, NY, USA, 851–860. https://doi.org/10.1145/2984511.2984565

[88] Wei Wang, Alex X. Liu, and Muhammad Shahzad. 2016. Gait Recognition Using Wifi Signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. ACM, New York, NY, USA, 363–373. https://doi.org/10.1145/2971648.2971670

[89] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and Modeling of WiFi Signal Based Human Activity Recognition. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking* (Paris, France) *(MobiCom '15)*. ACM, New York, NY, USA, 65–76. https://doi.org/10.1145/2789168.2790093

[90] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. 2014. 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* (Maui, Hawaii, USA) *(MobiCom '14)*. ACM, New York, NY, USA, 617–628. https://doi.org/10.1145/2639108.2639143

[91] Y. Wang, K. Wu, and L. M. Ni. 2017. WiFall: Device-Free Fall Detection by Wireless Networks. *IEEE Transactions on Mobile Computing* 16, 2 (Feb 2017), 581–594. https://doi.org/10.1109/TMC.2016.2557792

[92] Teng Wei and Xinyu Zhang. 2015. mtrack: High-precision passive tracking using millimeter wave radios. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM, 117–129.

[93] Teng Wei and Xinyu Zhang. 2015. mTrack: High-Precision Passive Tracking Using Millimeter Wave Radios. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking* (Paris, France) *(MobiCom '15)*. ACM, New York, NY, USA, 117–129. https://doi.org/10.1145/2789168.2790113

[94] Teng Wei and Xinyu Zhang. 2017. Pose Information Assisted 60 GHz Networks: Towards Seamless Coverage and Mobility Support. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking* (Snowbird, Utah, USA) *(MobiCom '17)*. ACM, New York, NY, USA, 42–55. https://doi.org/10.1145/3117811.3117832

[95] W. Xi, J. Zhao, X. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang. 2014. Electronic frog eye: Counting crowd using WiFi. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. 361–369. https://doi.org/10.1109/INFOCOM.2014.6847958

[96] Zhicheng Yang, Parth H. Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra. 2016. Monitoring Vital Signs Using Millimeter Wave. In *Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Paderborn, Germany) *(MobiHoc '16)*. ACM, New York, NY, USA, 211–220. https://doi.org/10.1145/2942358.2942381

[97] Y. Ye, Y. Tian, M. Huenerfauth, and J. Liu. 2018. Recognizing American Sign Language Gestures from Within Continuous Videos. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. 2145–214509. https://doi.org/10.1109/CVPRW.2018.00280

[98] T. Yuan, S. Sah, T. Ananthanarayana, C. Zhang, A. Bhat, S. Gandhi, and R. Ptucha. 2019. Large Scale Sign Language Interpretation. In *2019 14th IEEE International Conference on Automatic Face Gesture Recognition (FG 2019)*. 1–5. https://doi.org/10.1109/FG.2019.8756506

[99] Zahoor Zafrulla, Helene Brashear, Thad Starner, Harley Hamilton, and Peter Presti. 2011. American Sign Language Recognition with the Kinect. In *Proceedings of the 13th International Conference on Multimodal Interfaces* (Alicante, Spain) *(ICMI '11)*. ACM, New York, NY, USA, 279–286. https://doi.org/10.1145/2070481.2070532

[100] Yunze Zeng, Parth H. Pathak, and Prasant Mohapatra. 2016. WiWho: Wifi-based Person Identification in Smart Spaces. In *Proceedings of the 15th International Conference on Information Processing in Sensor Networks* (Vienna, Austria) *(IPSN '16)*. IEEE Press, Piscataway, NJ, USA, Article 4, 12 pages. http://dl.acm.org/citation.cfm?id=2959355.2959359

[101] Ding Zhang, Mihir Garude, and Parth Pathak. 2018. mmChoir: Exploiting Joint Transmissions for Reliable 60GHz mmWave WLANs. In *Proceedings of the 19th ACM International Symposium on Mobile Ad Hoc Networking and Computing* (Los Angeles, USA) *(MobiHoc '18)*. ACM, New York, NY, USA, 10.

[102] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang. 2011. A Framework for Hand Gesture Recognition Based on Accelerometer and EMG Sensors. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* 41, 6 (Nov 2011), 1064–1076. https://doi.org/10.1109/TSMCA.2011.2116004

[103] Yu Zhang and Qiang Yang. 2017. A Survey on Multi-Task Learning. *CoRR* abs/1707.08114 (2017). arXiv:1707.08114 http://arxiv.org/abs/1707.08114

[104] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. 2014. Facial Landmark Detection by Deep Multi-task Learning. In *Computer Vision – ECCV 2014*, David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars (Eds.). Springer International Publishing, Cham, 94–108.

[105] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. 2016. Co-occurrence Feature Learning for Skeleton based Action Recognition using Regularized Deep LSTM Networks. *CoRR* abs/1603.07772 (2016). arXiv:1603.07772 http://arxiv.org/abs/1603.07772

[106] Yanzi Zhu, Yibo Zhu, Ben Y. Zhao, and Haitao Zheng. 2015. Reusing 60GHz Radios for Mobile Radar Imaging. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking* (Paris, France) *(MobiCom '15)*. ACM, New York, NY, USA, 103–116. https://doi.org/10.1145/2789168.2790112

[107] H. Zou, Y. Zhou, J. Yang, H. Jiang, L. Xie, and C. J. Spanos. 2018. DeepSense: Device-Free Human Activity Recognition via Autoencoder Long-Term Recurrent Convolutional Network. In *2018 IEEE International Conference on Communications (ICC)*. 1–6. https://doi.org/10.1109/ICC.2018.8422895