Near Real-Time ASL Recognition Using Millimeter Wave Radar

Oladipupo O. Adeoluwa, Sean J. Kearney, Emre Kurtoglu,
Charles J. Connors, and Sevgi Z. Gurbuz

Department of Electrical and Computer Engineering, University of Alabama, Tuscaloosa, USA

ABSTRACT

Most research in technologies for the Deaf community have focused on translation using either video or wearable devices. Sensor-augmented gloves have been reported to yield higher gesture recognition rates than camera-based systems; however, they cannot capture information expressed through head and body movement. Gloves are also intrusive and inhibit users in their pursuit of normal daily life, while cameras can raise concerns over privacy and are ineffective in the dark. In contrast, RF sensors are non-contact, non-invasive and do not reveal private information even if hacked. Although RF sensors are unable to measure facial expressions or hand shapes, which would be required for complete translation, they can provide accurate measurement of signing dynamics, including distance, velocity, and higher-order derivatives. This paper investigates the classification accuracy and issues involved in near real-time ASL recognition using RF sensors for the design of smart DeafSpaces. In this way, we hope to enable the Deaf community to benefit from advances in technologies that could generate tangible improvements in their quality of life.

Keywords: sign language, American Sign Language, ASL, gesture recognition, human-computer interaction, radar, RF sensors, micro-Doppler

1. INTRODUCTION

In recent years, there has been a great deal of interest in human-computer interaction applications enabled by RF sensing. Examples include gesture recognition[1,2] for non-contact device control,[3] intelligent driver assistance,[4,5] and air writing.[6] The gestures considered in these applications typically feature movements such as required to operate a virtual button, knob or slider. In the case of air writing, the gestures are constrained to the 26 letters of the English alphabet.

This work aims at extending RF sensing capabilities to a related means of movement-based device control based on sign language recognition. While voice-controlled personal assistants, such as Alexa, Siri or Echo, offer significantly augmented capabilities via language-based human-computer interaction, this technology is not accessible to deaf individuals, whose primary mode of communication is American Sign Language (ASL). Development of RF-sensor based ASL-sensitive user interfaces has the potential to improve the lives of the over 1 million users of ASL in North America via new technology for intelligent DeafSpaces,[7] robotic personal assistants, health monitors, or even emergency response.

However, sign language fundamentally differs from gestures in that it is more complex, nuanced, and communicative in nature.[8,9] This is reflected by the greater information content, as indicated by the fractal complexity, of ASL in contrast to daily activities, such as folding clothes.[10] Signing also presents additional challenges relating to fine-grained temporal dynamics and linguistic parameters, such as prosody (e.g. pauses and suprasegmental components, such as phrase-final lengthening) and grammatical structure. Indeed, studies of sign language have

Further author information: (Send correspondence to S.Z.G.)
O.O.A: E-mail: oadeoluwa@crimson.ua.edu
S.J.K: E-mail: sjkearney@crimson.ua.edu
E.K.: E-mail: ekurtoglu@crimson.ua.edu
C.J.C.: E-mail: cjconnors@crimson.ua.edu
S.Z.G.: E-mail: szgurbuz@eng.ua.edu

shown that it can take ASL learners at least 3 years to produce signs in a manner that is perceived as fluent by native signers.[11]

Much of the research in ASL recognition to-date has focused on translation to facilitate the communication between deaf and hearing individuals. Two principle sensing modalities have been investigated: wearables and video.[12] Although sensor-augmented gloves have been reported to yield high gesture recognition rates,[13] they cannot capture the information expressed through head and body motion. More importantly, gloves are intrusive and inhibit users in their daily lives and have been strongly disliked in the Deaf community.[14] In contrast, video has been used to great benefit by deaf and hard-of-hearing (HoH) individuals for interpersonal communications. But when considering the design of smart DeafSpaces, where sensors would be on in the home all day and night, 7 days a week, video has elicited many concerns over privacy and potential abuse for unwanted surveillance. Moreover, video is not effective in the dark, when a smart environment should still need to be responsive to the user.

Although RF sensors are limited by their inability to perceive facial expressions or hand shapes, they do acquire a unique source of information about the dynamics of signing: namely, a visual representation of the kinematic patterns of motion via the micro-Doppler (mD) signature. Micro-Doppler[15] refers to frequency modulations that appear about the central Doppler shift, which are caused by rotational or vibrational motions that deviate from principle translational motion.

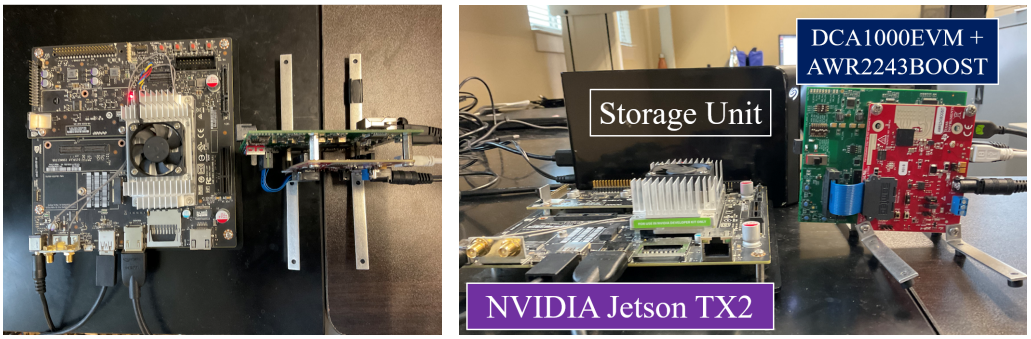
In prior work,[16] where we first proposed RF sensor-based ASL recognition,[17] we found that the signing of fluent ASL users, e.g. participants who are Deaf or a Child-of-Deaf-Adults (CODA), could be discriminated from that of hearing imitation signers through classification of micro-Doppler signatures alone. By imitation signing, we are referring to the use of data from hearing non-signers who are imitating the signs shown in a video. Even though both users aim at articulating the same ASL sign, the non-signer's imitation contains significant kinematic errors so that the articulation of the sign is actually distinct from that of fluent ASL users. Another linguistic property of ASL that is observable in RF data is *coarticulation* - the difference in the shape of the sign in the time-frequency representation, e.g. micro-Doppler signature. Because the micro-Doppler reflects velocity, which differs depending on the prior sign and from which position the user transitions into the articulation of the sign, the micro-Doppler signature of a sign slightly differs based on position within connected discourse. Thus, in the design of ASL-sensitive user interfaces, which use machine learning for ASL recognition, it is important to take into consideration the linguistic properties of ASL.

Another important aspect of user interface design is the appropriate selection of a trigger sign to wake the device and initiate the recognition of subsequent command signs. The trigger sign should be selected not only based on feedback from the Deaf community to ensure its suitability from a cultural perspective, but also such that it is easily articulated and not easily confused with commonly used signs. Because RF sensors perceive only the kinematics of motion, and not facial expressions or hand shape, it is also important that the trigger sign have distinctive dynamics. The trigger sign selected should be distinguishable not only from daily connected discourse, but also from other daily activities, such as walking or reaching for a book.

Towards this aim, this paper focuses on two aspects of near real-time RF sensor-based ASL-sensitive user interfaces: 1) the recognition accuracy of 15 ASL signs and 3 daily activities, and 2) computational complexity of implementation on embedded edge-computing platforms, such as the NVIDIA Jetson TX2. In Section 2, we describe the RF sensor and embedded platforms utilized, as well as the ASL signing data acquired. In Section 3, the data pre-processing steps are described, while in Section 4, the classification accuracy achieved for different deep neural networks (DNNs) and their computational complexity are examined. Section 5 discusses the implementation for near-real time processing, while Section 6 concludes and provides an outline for future research.

2. HARDWARE AND DATA COLLECTION

In this work, an edge computing device is used to capture raw I/Q data of articulated ASL signs, perform data pre-processing, generate micro-Doppler signatures, and implement a deep neural network (DNN) for classification. The system components leveraged in this work are briefly described along with their specifications, communication protocols and setup below:



(a) Top view (b) Front view

Figure 1: Hardware setup of Nvidia Jetson TX2, AWR2243BOOST and DCA1000EVM.

- An Nvidia Jetson TX2 System-on-A-Chip (SoC) embedded board served as our computing core for the entire project. The TX2 boasts a Dual-core NVIDIA Denver-2 64 bit CPU as well as a quad-core Arm Cortex-A57 MPCore processor complex, 1.3 Teraflops of graphical computing power made possible by the 256 Nvidia Cuda cores on board the GPU and an 8GB LPDDR4 memory.[18] The TX2 is mounted on a development board in order to increase its accessibility to external peripheral devices.
- TI's AWR2243BOOST Rev. B Module 77Ghz transceiver was used in this study. The radar is a multiple input multiple output (MIMO) FMCW radar with two transmitters and four receivers, operating at center frequency of 77 GHz with a 4 GHz bandwidth. The radar is coupled and operates in conjunction with the DCA1000 EVM module for capturing low voltage differential signalling(LVDS) data from the radar.

The entire system ran on a net power supply of 19 VDC and was augmented with an external hard drive of 10 TB that served as the local storage unit. The communication protocol between the Jetson and the radar was an serial peripheral interface (SPI) connection in a single master and single slave configuration while UDP communication over Ethernet was responsible for the transmission of data packets containing radar data from the data capture card to the processor and subsequently the storage unit. All connections to the Jetson were facilitated through the TX2 Development board. An illustration of the overall system configuration is shown in Figure 2.

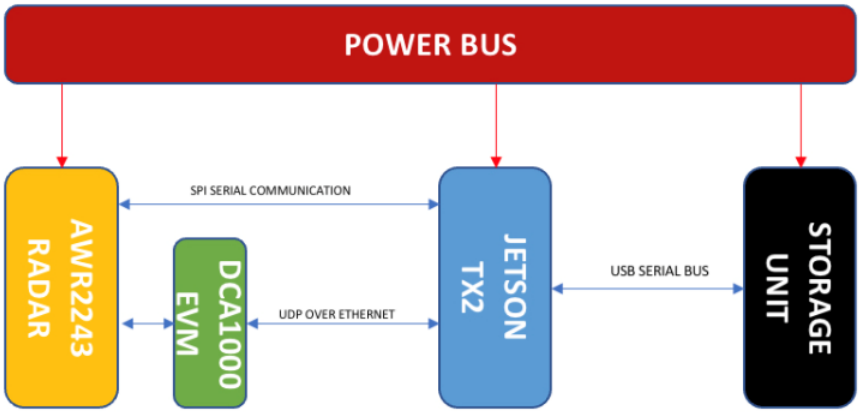


Figure 2: System Configuration.

Table 1: ASL and Daily Activity Classes

ASL/Motion Activity	Samples per Class	ASL/Motion Activity	Samples per Class	ASL/Motion Activity	Samples per Class
Walking	166	Sitting	161	Standing Up	161
Tired	106	Book	106	Sleep	106
Evening	129	Ready	129	Hot	129
Month	119	Cook	119	Again	119
Summon	122	Maybe	122	Night	122
Something	122	Teacher	122	Teach	122

2.1 ASL Dataset

During data collection, participants sat on a chair at a distance of roughly 1 meter facing the RF sensor, which was located at a height of 1 meter from the ground. Participants were prompted with slides indicating which sign should be articulated. A total of 11 participants articulated (on average) 118 samples per class. Note that articulations of the same class were avoided by having the participants repeat the entire sequence of signs, rather than the same sign multiple times. This was done to avoid "muscle memory" and ensure the statistical independence of each articulation. A total of 15 different ASL signs (capitalized) and 3 different daily activities were recorded, as given in Table 1.

In this work, due to the restrictions imposed by the COVID-19 pandemic, hearing imitation signers were utilized; however, the ASL signs considered were chosen as those that offered the best replicability by hearing non-signers. To identify the most replicable sub-set of signs, the Discrete Frechet's Distance and Dynamic Time Warping metrics were extracted from the envelopes of signatures acquired from fluent ASL users and hearning non-signers for 100 different ASL signs. The 15 signs that had the greatest similarity between fluent ASL users and hearing non-signers were utilized in this study.

3. DATA PRE-PROCESSING

The received I/Q data is reshaped into 3D radar data cube (RDC) array with the dimensions of fast-time samples (the number of ADC samples) $\times$ slow-time samples (the number of pulses) $\times$ channels, according to the binary data format provided by the data capture card. The Fast Fourier Transform (FFT) across fast-time samples can be used to find the frequency difference between the transmitted and the received signals. The observed frequency difference, also known as the beat frequency, f_b, can be used to compute the target range, R, as $R = cf_b/2\gamma$, where c is the speed of light and γ is the chirp rate. Similarly, an FFT across slow-time samples can be employed to obtain the velocity, v of the targets in the scene as $v = cf_d/2f_t$, where f_d is the Doppler frequency shift stemming from moving targets and f_t is the transmit (center) frequency.

A target's velocity variation over time can be visualized using various time-frequency transforms, but the most commonly used is the spectrogram, which is the square modulus of the short-time Fourier transform across slow-time. Sample spectrograms for walking and three different ASL signs, Teacher, Maybe, and Summon, are shown in Figure 3.

While generating spectrograms, there are few parameters which affect both the computation time and image resolution in time and frequency (velocity). While shorter window sizes provide higher temporal resolution, they suffer from low frequency resolution and vice versa. This tradeoff effects the ability of the spectrogram to represent the motion of interest, and, hence, the features that are automatically computed in DNNs. While longer spectrograms may give more accurate frequency estimates, they also take longer to compute. In near real-time applications, the processing time to generate a spectrogram can be a significant source of delay in the system. This is reflected in the graphs shown in Figure 4, which show how the processing time on an edge computing device, in this case the NVIDIA Jetson board, varies as a function of the recording duration and number of FFT points (NFFT) used in computing the spectrogram. Notice that the spectrogram for a given data sample can be generated under 1 sec for recordings shorter than 10 sec. The processing time linearly increases with recording duration. In the case of NFFT, once the number of FFT points exceeds roughly 250 samples, the processing

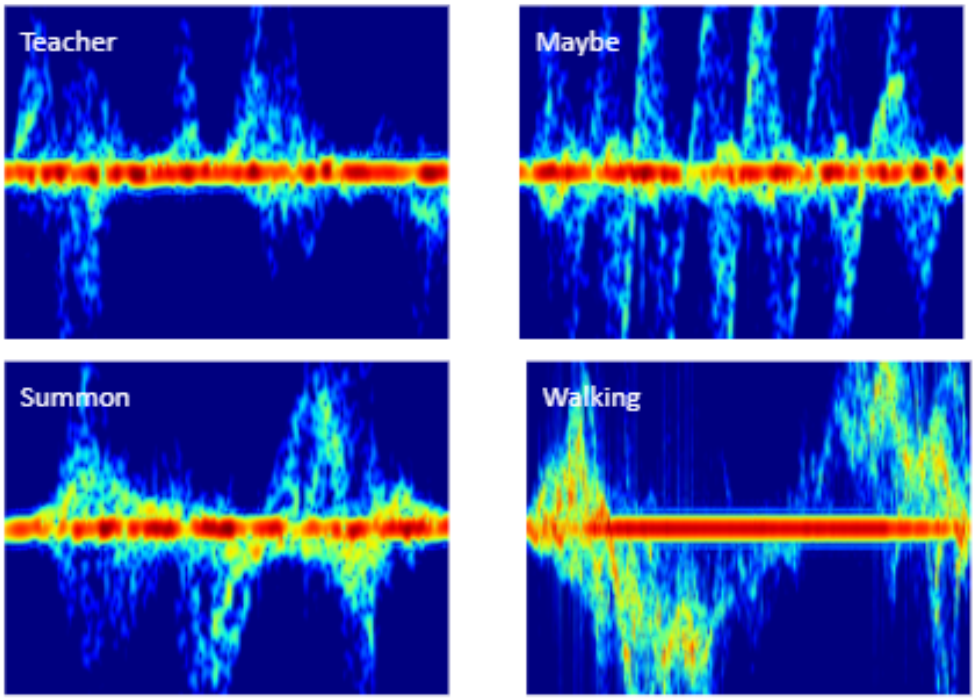
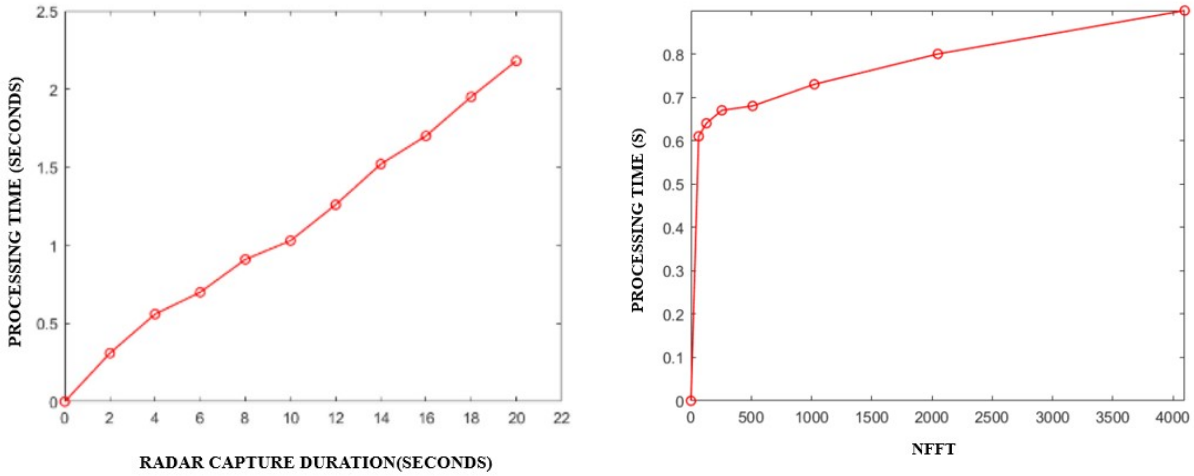


Figure 3: Spectrograms for four sample classes: walking and three ASL signs.



(a) Processing time versus recording duration.

(b) Processing time versus the number of FFT points.

Figure 4: Relationship between computation time and spectrogram generation parameters.

time also linearly increases with NFFT; but a 10-fold increase in NFFT results in just a 0.2 second increase in processing time. In contrast, increasing the recording duration 10-fold will increase the processing time by about 2 seconds.

4. CLASSIFICATION

Deep learning has provided significant performance increases in computer vision, natural language processing, and RF sensing for indoor monitoring.[19] A principle challenge, however, is that typically low training sample support in RF applications. This problem is especially acute in applications requiring human subjects testing,

and ASL recognition, for which there may be few individuals available who are fluent in ASL. Thus, we compare the performance attainable using a convolutional neural network (CNN) trained on measured RF samples only with that attained using transfer learning by VGG16[20] pre-trained on the 1.2 million optical image database, ImageNet.[21] To ascertain the best performing CNN given the RF ASL dataset, we compared the accuracy achieved for varying depths and number of convolutional filters. The results, given in Figure 5, show that the optimal CNN architecture is that with 4 convolutional layers and 64 convolutional filters. Thus, our comparisons are based on this 4-layer CNN, depicted in Figure 6. The four convolutional layers are followed by a flatten layer, two fully connected layers, and, lastly, a softmax layer for prediction.

		Number of Layers				
		2	3	4	5	6
Number of Filters	16	85.42	87.17	93.00	92.41	80.75
	32	81.92	88.62	94.75	93.87	87.17
	64	73.17	89.79	95.04	91.54	93.58
	128	47.81	89.21	91.25	94.75	90.96

Figure 5: Accuracy of the CNN for different layer and filter parameters.

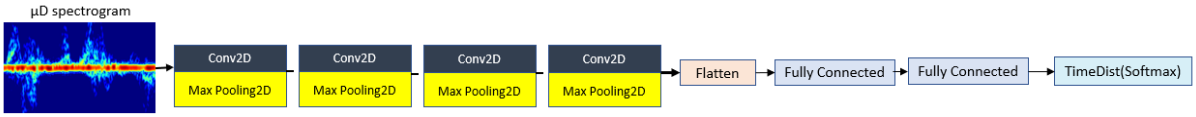


Figure 6: Architecture of the 4-layer CNN utilized.

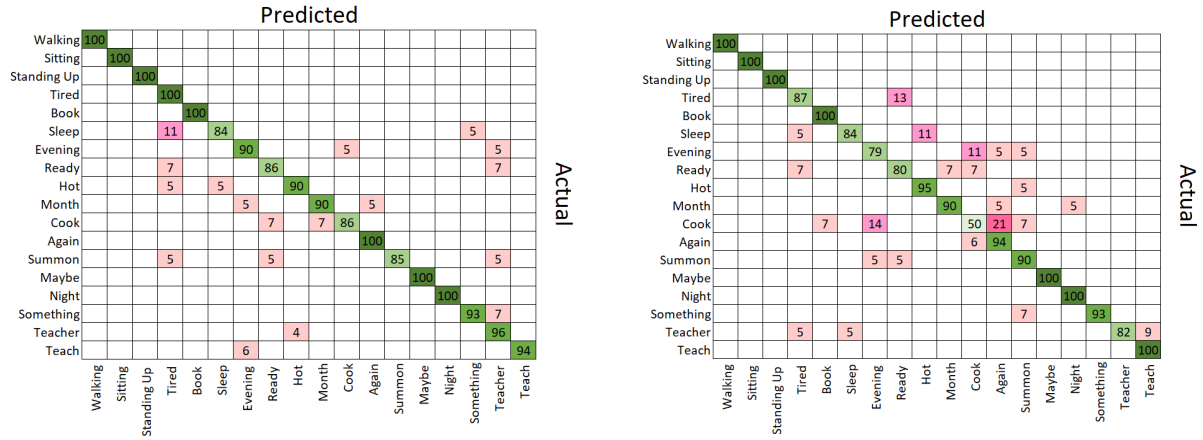
Both the 4-layer CNN and VGG16 were trained using Google Colab with a dataset of 2282 spectrograms composed of 18 classes, 15 of which were ASL commands and 3 of which were daily activities. The 2282 spectrogram samples were then split with 85% of the samples used for training and 15% of the samples used for testing. In the case of VGG16, the network was pre-trained on ImageNet and the final layer fine-tuned using the real RF training samples. As listed in Table 2, the batch size for the VGG Net was set to 24 while the batch size for the CNN network was set to 32. With these parameters, the CNN was found to have an accuracy of 95.04% and the VGG network was found to have an accuracy of 91.55%. Confusion matricies for both networks are shown in Figure 7, from which the following observations were made:

- The **4-Layer CNN** had an accuracy of over 80% for predicting the correct label for all 18 classes. It was found that it always correctly predicted the motion classes (Walking, Sitting, Standing Up) and also always correctly predicted the ASL signs MAYBE and NIGHT. The signs for which it performed worst were SLEEP, READY, and COOK.
- **VGG16** had over a 79% accuracy at predicting the correct label for 17 of the 18 classes for which it was trained and tested. VGG16 always correctly predicted the motion classes (Walking, Sitting, Standing Up) and also always correctly predicted the ASL sign MAYBE. The only sign that it had a problem accurately recognizing was COOK, for which it was only only correctly predicted 50% of the time.

In addition to the classification accuracy, another critical metric for near real-time applications is the prediction time for each model; i.e., how long it takes the embedded platform to actually test newly acquired samples. The prediction time for each network is presented in Figure 8. The results were for motion activities and ASL signs is reported separately due to the difference in spectrogram duration. The spectrograms for daily activities

Table 2: Description of Networks and Resulting Classification Accuracy

Model	No. of Classes	No. of Training Samples	No. of Test Samples	Batch Size	No. of Trainable Parameters	% Accuracy
VGG16	18	1939	343	24	64	91.55
CNN	18	1939	343	24	128	95.04



(a) Confusion Matrix of the Convolution Neural Network

(b) Confusion Matrix of the VGG16 Network

Figure 7: Confusion Matrix of (a) CNN and (b) VGG.

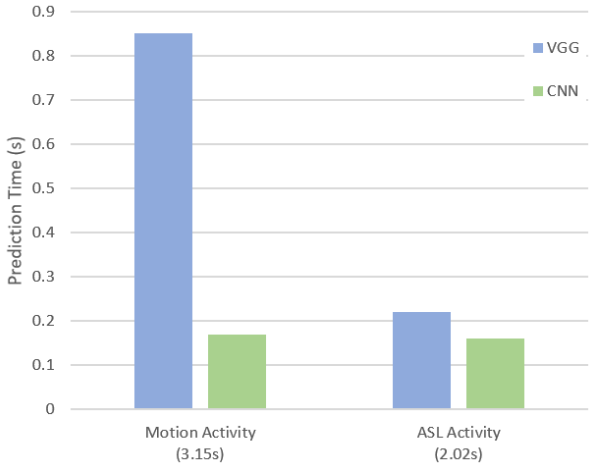


Figure 8: Prediction time for each DNN.

is about 1 second longer in duration than that of ASL signs. For ASL signs, the CNN model took 0.16s on average to make a prediction while the VGG model took 0.22s on average. For motion activities, the CNN model was only slightly slower making the prediction in 0.17s on average, while the VGG model had a much more significant prediction time for motion activities versus ASL activities, taking 0.85s on average. It is interesting to observe that the prediction time for VGG16 on the NVIDIA Jetson board was longer than that for the CNN, especially for daily activities. In contrast, the CNN yielded predictions for daily activities and ASL signs at about the same prediction time.

5. PRACTICAL CONSIDERATIONS IN IMPLEMENTATION OF NEAR-REAL TIME ASL RECOGNITION

5.1 Sensor Selection

Over the past decade, a plethora of RF sensors have become commercially available, which target a range of applications, including respiration and heart rate monitoring, indoor localization and tracking, and automotive perception. None of these systems have been specifically designed for ASL recognition, but some RF systems have been advertised for gesture recognition, including Google Soli[22] and a 60 GHz frequency modulated continuous wave (FMCW) transceiver by Infineon,[23] among others. While a comprehensive survey and comparison of available RF sensors is beyond the scope of this paper, we have observed that some of the nuances in the commercially available sensors can impact recognition performance and suitability for use in embedded systems. In particular, we would like to discuss four different RF sensors with which we have conducted experiments on ASL recognition in a controlled laboratory environment: 1) Texas Instruments (TI) 77 GHz multi-channel FMCW transceiver, which was utilized in this paper; 2) Ancortek 25 GHZ FMCW transceiver; 3) Xethru 7-10 GHz ultra-wide band (UWB) impulse radar; and 4) Acconeer 60 GHz pulse Doppler (PD) radar. Spectrgrams for each of these sensors are shown in Figure 9. We discuss each of these sensors in turn:

- **TI 77 GHz multi-channel FMCW Radar:** It may be observed that the richest details are visible in the TI 77 GHz FMCW radar data. This is not surprising because among all the other sensors, the TI sensor has the highest transmit center frequency (hence greater Doppler shift for the same velocity) and highest bandwidth (4 GHz), which allows for greater range resolution. This sensor also has the advantage of being one of the few multi-channel sensors commercially available (2 or 3 TX/4 RX), and thus gives the opportunity for beamforming and direction-of-arrival estimation. Although normally configured to be operated through its own Graphical User Interface (GUI), it can be operated from the command line driven by an edge computing platform. Details of command line operation are provided later in this section. Both the TI GUI and command line modes of operation all the user to control a wide range of customizable parameters, e.g. chirp profile, waveform, center frequency, bandwidth, number of analog-to-digital converter (ADC) samples, and so forth. The ability to completely control and design the transmit FMCW waveform is an attractive advantage of this platform.

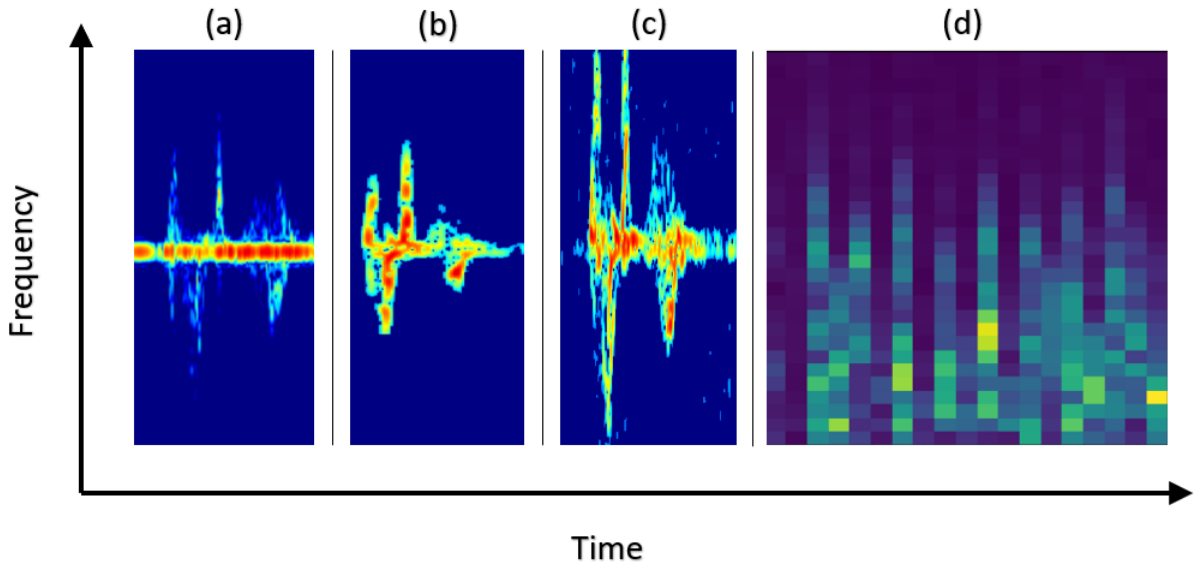


Figure 9: Spectrogram for ASL sign "hello" using (a) TI 77GHz FMCW Radar, (b) Xethru 10 Ghz UWB Radar, and (c) Ancortek 25 GHz FMCW Radar, and an erasing motion of the hand with (d) the Acconeer 60 GHz PD Radar.[24]

- **Acconeer 60 GHz PD Radar:** This sensor is quite small in size, which makes it advantageous for indoor monitoring applications and human-computer interaction, and is designed to be operable with a Raspberry Pi. It also has a high transmit frequency, suggesting that it would offer good sensitivity to small movements. However, its configuration as a pulse Doppler radar reduces the signal-to-noise ratio of the received RF returns. Operation of the 60 GHz PD radar relies on developer provided source code hardware for the hardware abstraction layer, device drivers, and build environment. This renders the user entirely reliance on the Acconeer-provided modules to operate the device from the command line, which in turn limits programmability. Two types of data outputs are provided by the read modules: I/Q and "sparse" data streams. Because the documentation does not include specific information on the transmit waveform, user-written pulse compression functions are precluded, while the "sparse" data stream does not yield spectrograms with good resolution. Furthermore, because of the limited computational capability of the Raspberry PI, practical utilization of the device still benefits from use of a more capable edge computing platform. Thus, operation from the Raspberry PI simply adds hardware to the implementation and overall was found to not be as user-friendly or effective for ASL recognition as the TI 77 GHz transceiver.
- **Ancortek 25 GHz FMCW Radar:** The Ancortek system provides a maximum bandwidth of 1.5 GHz, which is less than that of the TI 77 GHz sensor, but through its GUI provides above average quality range-Doppler maps and micro-Doppler signatures. The received signal does include however significant sensor artifacts, including persistent horizontal lines a specific positive and negative micro-Doppler frequencies, and I/Q imbalance. Use of the provided GUI is not required, Ancortek does provide MATLAB code for command line operation upon request, but most edge computing platforms require python scripts for direct data acquisition requiring the MATLAB code to be converted to Python. Size-wise, the Ancortek is the largest of all sensors.
- **Xethru 7-10 GHz UWB Radar:** The Xethru sensor is small and about the same size as the Acconeer, but the lower transmit frequency offers less Doppler bandwidth than millimeter wave alternatives. In prior work comparing the ASL recognition accuracy attainable from the TI 77 GHz, Ancortek 25 GHz, and Xehtru UWB radars, we found that it offered about 10% lower accuracy than the other two sensors.

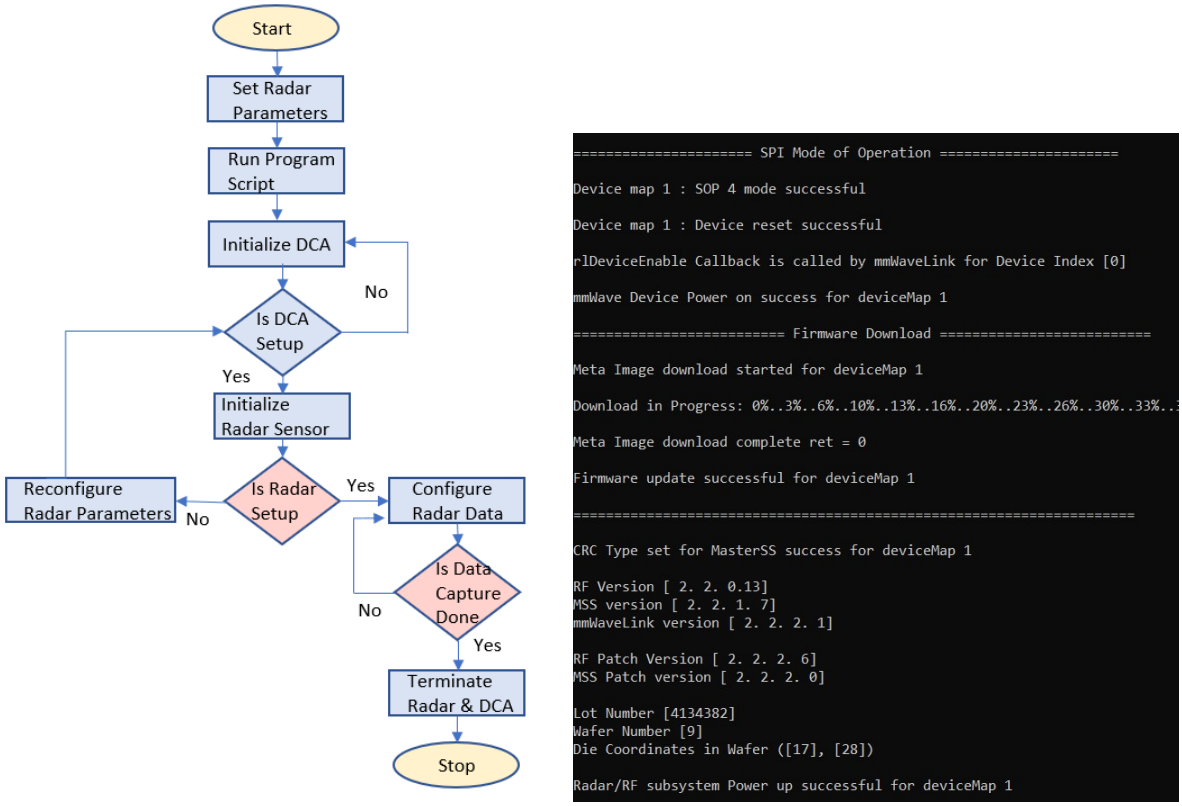
Given the above considerations, we found that the TI 77 GHz transceiver was the platform most suitable for near real-time applications. Notably, a team of researchers has recently proposed use of the TI 77 GHz board for the establishment of large-scale datasets for benchmarking[25] and has provided open-source some scripts for data acquisition and processing this with device.

5.2 Command Line Operation of 77 GHz FMCW Transceiver

The AWR2243 radar is traditionally operated from desktop or personal computers with the manufacturers Graphic User Interface (GUI) program (mmWaveStudio), serving as a control center. This program runs exclusively on a Windows Operating System. TI has however provided a device firmware package that enables developers to replicate the complete functionality of the GUI on non-standard Operating systems such as the Jetson's.[26] Nvidia's Jetpack SDK contains all the drivers needed to configure the Jetson TX2 to operate as an embedded platform. It is also bundled with an Ubuntu 18.06 Linux distribution which serves as its operating system. For the smooth operation of the 77 Ghz RF Sensor and DCA1000 EVM the required hardware drivers needed to foster communication between the Jetson and RF sensor need to first be installed. As with the GUI, before a data capture session can commence, the radar parameters must be initialized, the only major difference being that an editable text file containing the parameters is what is overwritten whenever a value update is required. This approach gives some flexibility as to how the radar parameters can be updated, as long as they remain within manufacturer and hardware limitations of the sensor. After complete installation of all dependencies, a Command Line (CLI) Terminal script triggers two distinct executable files from the CLI Terminal which prime the RF Sensor and DCA Card for Data Capture using the filled in radar parameters.[25] Upon successful frame trigger of the radar, a data stream is opened and raw I/Q data is dumped in a binary file.

In switching over to command-line operation rather than data acquisition through the TI-provided GUI, we have noticed that for some radar settings the micro-Doppler signatures generated exhibit I/Q imbalance. We are

continuing to work on the device-control scripts to try to find a solution to these issues. For the implementation of near real-time applications, however, it is worthy of note that the RF sensor and DCA Capture card is never terminated and data is continuously streamed into binary files for continuous generation of mD spectrogram images prior to feeding the images into the pre-trained models discussed previously.



(a) Command Line Flow Chart

(b) Command Line Output

Figure 10: Command Line Flow and Command Line Output.

5.3 Latency

Following data capture, the data is first stored in binary files before any processing or classification operation commences. This approach introduces an increase in the computation time for the entire system because we attempt to cut our live data binary dumps at specified time intervals. In this work, the interval choices were selected based on how long it took to perform a single ASL sign in conjunction with the Data Generation times displayed in Figure 4. Hence, the total latency of our implementation is characterized by:

$$TotalLatency(s) = RadarCaptureDuration + DataGenerationTime + ActivityClassificationTime \quad (1)$$

The total latency for classifying normal motion activity data is greater than ASL activity because they take a longer duration to acquire during the Data Capture Phase. Note that the models we ran on the Jetson were trained on external device before deployment on the board and as a result were not optimized to utilize the 1.33 TFLOPs computing power of the on-board GPU.

6. CONCLUSION

This paper has shown that ASL signs can be distinguished from daily activities, achieving an overall classification accuracy of 95% for 18 motion classes. The computation time for spectrogram generation and model prediction on the NVIDIA Jetson board has been investigated for two different DNN models and various spectrogram durations. Practical implementation issues involved with using four different RF sensors, each with different transmit waveforms, center frequencies and bandwidths, as well as command-line operation of these RF sensors using an edge computing platform were discussed. The results in this paper form important first steps toward developing near real-time ASL-sensitive user interfaces empowered by non-invasive, ambient RF sensing.

ACKNOWLEDGMENTS

We would like to thank Daniel Gusland of the Norwegian Defence Research Establishment for help with software for the Jetson board. This work was funded in part by the National Science Foundation (NSF) Award 1932547. Human studies research was conducted under UA Institutional Review Board (IRB) Protocol 18-06-1271.

REFERENCES

[1] Zhang, Z., Tian, Z., and Zhou, M., "Latern: Dynamic continuous hand gesture recognition using fmcw radar sensor," *IEEE Sensors Journal* **18**(8), 3278–3289 (2018).

[2] Wang, Z., Yu, Z., Lou, X., Guo, B., and Chen, L., "Gesture-radar: A dual doppler radar based system for robust recognition and quantitative profiling of human gestures," *IEEE Transactions on Human-Machine Systems* **51**(1), 32–43 (2021).

[3] Gu, C., Wang, J., and Lien, J., "Motion sensing using radar: Gesture interaction and beyond," *IEEE Microwave Magazine* **20**(8), 44–57 (2019).

[4] Molchanov, P., Gupta, S., Kim, K., and Pulli, K., "Multi-sensor system for driver's hand-gesture recognition," (May 2015).

[5] Zhang, X., Wu, Q., and Zhao, D., "Dynamic hand gesture recognition using fmcw radar sensor for driving assistance," in [*2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*], 1–6 (2018).

[6] Arsalan, M. and Santra, A., "Character recognition in air-writing based on network of radars for human-machine interface," *IEEE Sensors Journal* **19**(19), 8855–8864 (2019).

[7] Tsymbal, K., "Deaf space and the visual world - buildings that speak: An elementary school for the deaf," *PhD Dissertation, The University of Maryland* (2010).

[8] Malaia, E., Borneman, J. D., and Wilbur, R. B., "Assessment of information content in visual signal: analysis of optical flow fractal complexity," *Visual Cognition* **24**(3), 246–251 (2016).

[9] Borneman, J. D., Malaia, E., and Wilbur, R. B., "Motion characterization using optical flow and fractal complexity," *Journal of Electronic Imaging* **27**(5), 1 – 6 (2018).

[10] Gurbuz, S. Z., Gurbuz, A. C., Malaia, E. A., Griffin, D. J., Crawford, C., Rahman, M. M., Aksu, R., Kurtoglu, E., Mdrafi, R., Anbuselvam, A., Macks, T., and Ozcelik, E., "A linguistic perspective on radar micro-doppler analysis of american sign language," in [*2020 IEEE International Radar Conference (RADAR)*], 232–237 (2020).

[11] Beal, J. S. and Faniel, J. S. K., "Hearing l2 sign language learners: How do they perform on asl phonological fluency?," *Sign Language Studies* **19**, 204 – 224 (2018).

[12] Ye, Y., Tian, Y., Huenerfauth, M., and Liu, J., "Recognizing american sign language gestures from within continuous videos," in [*2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*], 2145–214509 (2018).

[13] Tubaiz, N., Shanableh, T., and Assaleh, K., "Glove-based continuous arabic sign language recognition in user-dependent mode," *IEEE Transactions on Human-Machine Systems* **45**(4), 526–533 (2015).

[14] Erard, M., "Why sign-language gloves don't help deaf people," in [*The Atlantic*], (November 2017).

[15] Chen, V. C., [*The Micro-doppler Effect in Radar*], Artech House, Boston, MA (January 2011).

[16] Gurbuz, S. Z., Gurbuz, A. C., Malaia, E. A., Griffin, D. J., Crawford, C. S., Rahman, M. M., Kurtoglu, E., Aksu, R., Macks, T., and Mdrafi, R., "American sign language recognition using rf sensing," *IEEE Sensors Journal* **21**(3), 3763–3775 (2021).

[17] Gurbuz, S., Gurbuz, A., Crawford, C., and Griffin, D., "Radar-based methods and apparatus for communication and interpretation of sign languages," in [*U.S. Patent Application No. US2020/0334452 (Invention Disclosure filed Feb. 2018; Provisional Patent App. filed Apr. 2019.)*], (October 2020).

[18] Developer, N., "Jetson tx2 module," https://developer.nvidia.com/embedded/jetson–tx2.

[19] Gurbuz, S. Z. and Amin, M. G., "Radar-based human-motion recognition with deep learning: Promising applications for indoor monitoring," *IEEE Signal Processing Magazine* **36**(4), 16–28 (2019).

[20] Simonyan, K. and Zisserman, A., "Very deep convolutional networks for large-scale image recognition," (2015).

[21] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L., "Imagenet: A large-scale hierarchical image database," in [*2009 IEEE conference on computer vision and pattern recognition*], 248–255, Ieee (2009).

[22] Lien, J., Gillian, N., Karagozler, M. E., Amihood, P., Schwesig, C., Olson, E., Raja, H., and Poupyrev, I., "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Trans. Graph.* **35** (July 2016).

[23] Vaishnav, P. and Santra, A., "Continuous human activity classification with unscented kalman filter tracking using fmcw radar," *IEEE Sensors Letters* **PP**, 1–1 (04 2020).

[24] Dagasan, E., "Hand gesture classification using millimeter wave pulsed radar," (2020). Student Paper.

[25] D. Gusland, J. C., Torvik, B., Fioranelli, F., Gurbuz, S., and Ritchie, M., "Open radar initiative: Large scale dataset for benchmarking of micro-doppler recognition algorithms," in [*IEEE Radar Conference, Atlanta, GA*], (May, 2021).

[26] Instruments, T., "Mmwave-dfp," https://www.ti.com/tool/MMWAVE–DFP.