DiSiPro: Digital Signal Processing Tool for Radar-Based Human-Computer Interaction

Nuwan T. Attygalle[0000−0001−8498−1923]
Matjaž Kljun[0000−0002−6988−3046]
Klen Čopič Pucihar[0000−0002−7784−1356]

Abstract In recent years, the development of radar-based human-computer interaction systems using miniature radar-on-chip sensors has attracted significant interest from both academia and industry. This interest is fuelled by the availability of affordable radar chips and advancements in signal processing and machine learning that improve radar signal interpretation accuracy. However, several challenges remain, particularly when we want to compare different radar-based gesture interaction systems. On dimension to compare different systems can be based on radar signal representations since raw voltage data can be used to extract these. Different signal representations include among others range-Doppler, range-Angle, and point clouds. Existing research often limits comparative testing to the same radar signal representation, focusing mainly on gesture recognition algorithms or minimal variations within digital signal processing pipelines. In order to fill this gap we designed and created an open source tool that enables fast and reliable dataset preparation for comparative testing of radar signal representations. The open-source tool enables visualisation of different radar signal representations, and includes a command-line interface for batch processing in order to streamline dataset preparations.

Nuwan T. Attygalle
Faculty of Mathematics, Natural Sciences and Information Technologies, University of Primorska, Koper, Slovenia, e-mail: `nuwan.attygalle@famnit.upr.si`

Matjaž Kljun
Faculty of Mathematics, Natural Sciences and Information Technologies, University of Primorska, Koper, Slovenia, and
Department of Information Science, Stellenbosch University, Stellenbosch, South Africa, e-mail: `matjaz.kljun@upr.si`

Klen Čopič Pucihar at University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies, Koper, Slovenia, and
Faculty of Information Studies, Novo Mesto, Slovenia, and
Department of Information Science, Stellenbosch University, Stellenbosch, South Africa, e-mail: `klen.copic@famnit.upr.si`

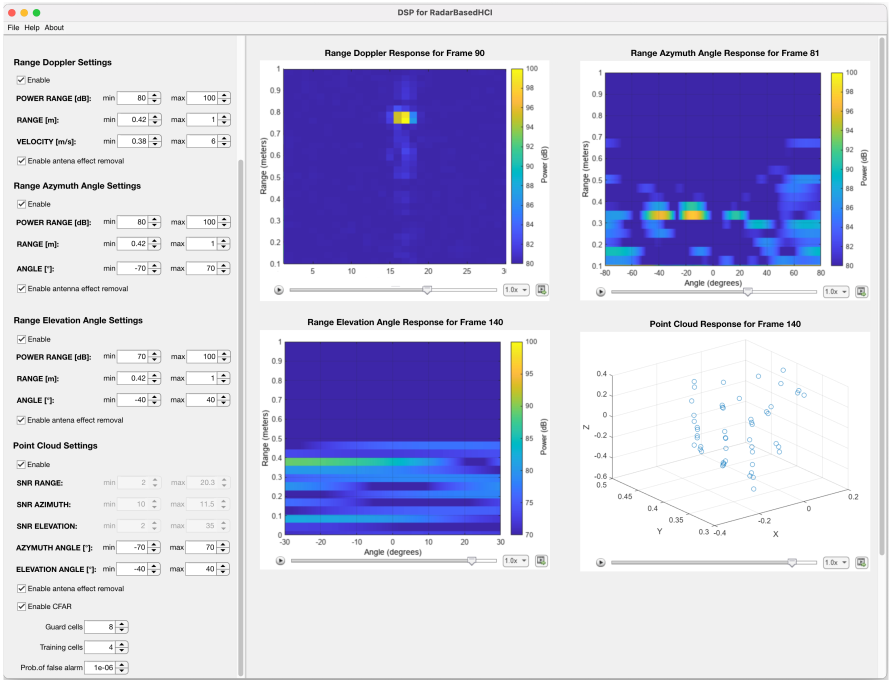

Fig. 1 Digital Signal Processing Tool for Radar-Based Human-Computer Interaction

1 Introduction

In recent years, the development of radar based human-computer interaction systems based on miniature radar-on-chip sensors has gained significant interest from both academia and industry. This surge is driven by two main factors: (i) the availability of affordable radar chips (e.g. Google Soli[1], Walabot[2], Texas Instruments: IWR6843ISK, IWR6843ODS, IWR1443[3]), and (ii) the major advancements in signal processing and machine learning that enhance the accuracy of radar signal interpretation for interactive purposes. Despite these advancements, several challenges remain to be addressed to facilitate practical applications of this technology. On such challenge relates to cooperative dimension of radar-based gesture interaction systems.

To date, only a limited number of research papers have compared radar-based gesture recognisers. Typically, such comparisons are (i) restricted to the same radar signal representation limiting comparative testing parameters to gesture recognition algorithms (e.g. comparison of deep neural network architectures [22, 3] and other

[1] `https://en.wikipedia.org/wiki/Google_ATAP#Project_Soli`

[2] `https://walabot.com/`

[3] `https://www.ti.com/lit/ug/swru546e/swru546e.pdf`

standard gesture recognisers [5]), or (ii) focus on minimal variations within digital signal processing pipelines [3]. Consequently, there is a noticeable gap in the literature regarding comprehensive comparative testing that would encompass a diverse set of parameter:

- Sensor type (e.g. custom vs. commercially available);
- Usage context (e.g. stationary vs. mobile);
- End users and their body parts involved in the interaction (e.g. fingers, wrist, hands, forearm, arm);
- Interaction types (e.g. tangible, grasping, mid-air, air-writing, etc.).
- Environment characteristics (e.g. indoor or outdoor as well as setup such as beneath a surface, through the wall);
- Radar system configuration, including:
  - impulse generation methods (e.g. ultra-wide band impulse, pulse Doppler, frequency modulated continuous waveform);
  - transmission and capture methods using various configuration of receiving and transmitting antennas (e.g. Single Input Multiple Output (SIMO), Multiple Input Multiple Output(MIMO));
- Radar signal representations (e.g. time series of range, permittivity, micro-Doppler, range-Doppler, range-Angle of Arrival (AoA), point cloud data).

In this paper, we focus on *radar signal representations*, with the primary objective of making comparative testing of radar signal representation more accessible. To this end, we have designed and implemented an open source tool that enables visualisation of different types of radar signal representations, such as: range-Doppler, range-Angles and point clouds (see Fig. 1). Additionally, the tool includes a command-line interface for batch processing, which is essential for streamlining digital signal processing pipelines that are required for dataset preparation. The batch processor also enables generation of In-phase and quadrature (IQ) radar cube representations. The open source tool is built with a combination of Matalab and Python and is currently limited to data captured by Texas Instruments radar sensors.

2 Background

Initially, radar systems were designed to gather essential information such as target location, range, velocity, and radar cross section (RCS), by analysing the electromagnetic waves reflected from the targets [16]. However, more recently, radar systems have been adopted for interactive purposes that includes gesture recognition.

Radar-based gesture recognition can be divided into four stages (Fig. 2): (i) *impulse generation*, (ii) *signal transmissions and capture*, (iii) *digital signal processing* and (iv) *feature extraction, and gesture classification*. For *impulse generation* various methods are available, including ultra-wide band impulse, pulse Doppler or frequency modulated continuous waveform. For *signal transmissions and capture* different configurations of receiving and transmitting antennas and operation modes are possible (e.g. Single Input Multiple Output, Multiple Input Multiple Output). Radar systems can also differ in *digital signal processing* techniques used, resulting in a diverse set of radar signal representations that capture intricate information essential for later feature extraction and gesture recognition. These encompass raw signal [15, 17], range [23, 19], permittivity [19], micro-Doppler [2], range-Doppler [9, 12, 24, 25, 28, 7, 1], range-Angle of Arrival (AoA) [9, 29], and point cloud data [14, 13, 8].

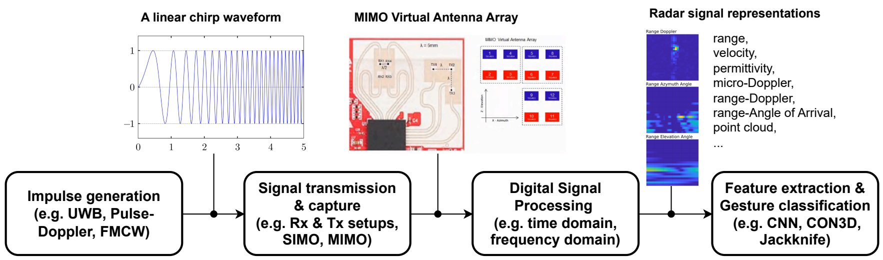

Fig. 2 Stages of radar-based gesture recognition.

Using raw signal representations, some researchers opted against employing pre-processing methods to fine-tune the signal. Therefore, they directly fed the sampled raw signal to the classifier [15, 17]. However, more commonly, various processing techniques have been employed prior to obtaining the other aforementioned signal representations. Sluyters et al. introduced an innovative pre-processing pipeline [19] that effectively removes radar source, antenna effects, and static clutters in the frequency domain, subsequently deducing range and permittivity [4] from the time domain signal. This approach compresses the high-dimensional radar signal into a two-dimensional space defined by ranges and apparent permittivity. Amin et al. [2] divided gesture time series into envelopes, generating Spectrograms through the extraction of maximum and minimum frequencies. Following this, maximum and minimum micro Dopplers were filtered from these Spectrograms, forming feature vectors for each gesture.

Another processing technique was employed by Zheng et al. [29] that created range-Doppler images from the radar signal and conducted static clutter and vibration removal on these images. They further generated a range-angle map based on the range-Doppler images and used both range-Doppler and range-angle maps as classifier inputs. Lee et al. [12] employed range-Doppler images as feature representations, executing static clutter removal and Constant false alarm rate (CFAR) processing. This approach retained real hand without any background information in the range-Doppler images. Additionally, Pantomime [14] harnessed radar data to generate point clouds, encapsulating X, Y, and Z coordinates, radial velocity, and Signal to Noise ratio (SNR). The point cloud generation process employed a distinctive angle of arrival algorithm known as Capon Beamforming [20].

As the last and fourth stage, radar-based gesture detection needs to undertake *feature extraction, and gesture classification*. Researchers have used a variety of Machine Learning (ML) and Deep Learning (DL) algorithms for fine feature extraction and mapping these features into various classes. Hazra et al. and Zheng et al. [9, 29] used range-Doppler and range-angle images for feature representation, feeding them into a DL network that incorporates Conv2D for feature extraction and Long Short Term Memory (LSTM) for implementing the gesture classifier. Similarly, Wang et al. and Hayashi et al. [25, 7] adopted an approach where range-Doppler images were used as feature representations, processed through a Conv2D-based network combined with LSTM for classification. Other researchers have opted for Conv3D layers merged with LSTM layers for classifier implementation [28]. Furthermore, Hazra et al. [8] used point cloud data for feature representation, leveraging LSTM for classifier development. Palipana et al. [14] introduced a hybrid architecture, integrating PointNet++ followed by LSTM modules for frame-wise spatio-temporal feature extraction.

In terms of feature direction, channelling extracted features into fully connected layers for classifier implementation is a prevalent practice among researchers [17, 6, 12, 1, 26], with convolutional layers frequently used for feature extraction. Moreover, for post-extraction of fine-grained features, the deployment of classifiers using ML algorithms has gained traction. Liu et al. [13] compared various ML methods, such as Naïve Bayesian, Decision Trees, Support Vector Machine (SVM), and Random Forest, against renowned DL architectures like VGG-Net [18], ZF-Net [27], SE-Net [11], and ResNet [10]. Similarly, Sun et al. [21] used a k-nearest neighbors (K-NN) classifier with $k = 10$ for classification, subsequent to feature extraction via a CNN network.

As can be see from the literature covered above, the extensive work exists in the domain of radar-based human-computer interaction covering a variety of approaches. However, despite this extensive work, very few papers perform comparative testing that goes beyond exploring different classification algorithms. The lack of comparative testing hinders the ability to gain a deeper understanding of the design space at play, which makes designing radar-based gesture recognition systems challenging.

3 Digital Signal Processing

Our system is designed to convert raw data (a time series of voltage levels) into various signal representations, such as: range-Doppler, range-Angles, point clouds, and IQ radar cube representations. In order to achieve this, we need the following information regarding radar system configuration:

- General: Range and velocity resolution, maximal range, maximal velocity.
- Chirp configuration: Number of chirps, number of samples per chirp, chirp time, chirp sweep bandwidth, carrier frequency.
- Antennas configuration: Number of Tx (transmitting) and Rx (receiving) antennas active, the pulse modulation, antenna multiplexing method (e.g. SIMO,

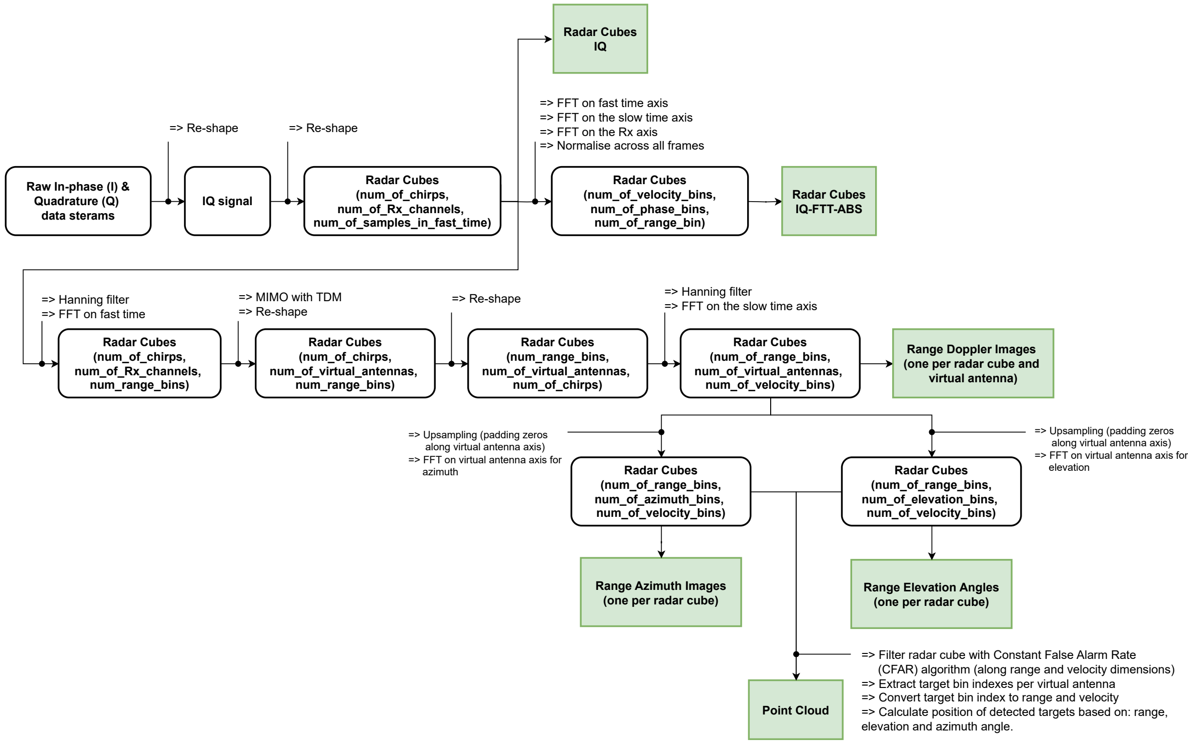

Fig. 3 Digital signal processing pipeline. The plipeline shows how 4 differnt signal rep

MIMO configuration), orthogonality (Time Domain Multiplexing (TDM) or Binary Phase Modulation (BPM)), distances between Rx and Tx antennas.

To achieve this, we were required to apply several steps within the Digital Signal Processing (DSP) pipeline, presented in Fig. 3. At the most rudimentary level, the radar system outputs In-phase (I) and Quadrature (Q) data streams (a pair for each receiving antenna). These streams form the input into our DSP pipeline, which starts by re-shaped data stream pairs into IQ signal. In the next stage the IQ signal is reshaped into the radar cubes (also known as radar frames) with the following cube format: *number of chirps* × *number of Rx channels* × *number of samples in the fast time*. We refer to this representation as IQ Radar Cube. Next, the Hanning filter and Fast Furier Transform (FFT) are applied on the *fast time axis* converting it to *number of range bins*. This step is followed by MIMO with Time Domain Multiplexing.

Knowing that MIMO with Time Domain Multiplexing was used allowed us to isolate virtual antenna chirps and form a radar data cube for each virtual antenna, changing our radar cube dimensions to: *number of chirps*, *number of virtual antennas*, *number of range_bins*. Following a Re-shape, Hannning filter and FFT on the slow time axis we extract velocity bin information transforming *number of chirps* into *number of velocity bins*. This representation is refered to as range-Doppler. The radar cube can be easily transformed to range-Doppler images (one for each virtual antenna).

Next, the FFT is applied on the *virtual antenna axis* converting it to *number of azimuth or elevation bins*.

In order to extract point clouds, we first need to perform target detection where we are interested in moving targets and targets with high Signal to Noise ratio (SNR). To calculate the threshold level for distinguishing moving targets from each other, we apply a Constant False Alarm Rate (CFAR) algorithm along the velocity dimension of the radar data cube. We also run CFAR along the range dimensions calculating threshold for isolating targets with good SNR.

After calculating the threshold values in velocity and range axis, we can isolate the targets by filtering out all the peaks that are below the calculated threshold values. Now we know the bin index in range and velocity dimension for each identified target, thus the actual range, velocity, azimuth and elevation values are known. This allows us to calculate the posting of detected targets.

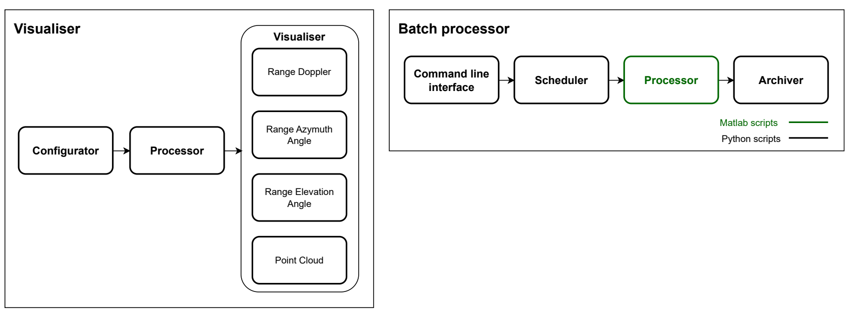

Fig. 4 Architecture of the Digital Signal Processing Tool for Radar-Based Human-Computer Interaction. The application is designed with two parts, the Visualiser and Batch Processor.

4 System Implementation

The Digital Signal Processor for Radar-Based HCI (Fig. 1) is implemented with Matlab 2024 and Python 3.8. The tool consists of the Visualiser and Batch Processor (see Fig. 4). The visualiser is intended as exploratory tool for fine-tuning the pre-processing parameters when optimising signal representations, such as for example Range-Doppler or Range-Angle images. Once the parameters are chosen, we can export them into JSON file for later use with Batch Processor. Batch processor processing.json

```
{
    "metajson": ""
    "source_dir": "./RadarDataFRomE/data/"
    "destination_dir": "./RadarDataFRomE/dataProcessed/"
    “rd_power_min”: 80,
    “rd_power_max”: 100,
    “rd_range_min”: 0.42,
    “rd_range_max”: 80,
    “rd_velocity_min”: 0.42,
    “rd_vlocity_max”: 80,
    “rd_enable_antenna_effect_removal”: true,
    ...
}
```

sensorSettings.json

```
{
 ...
  "systemConfig": {
    "summary": "This is a comments field not passed to device",
    "sceneParameters": {
      "ambientTemperature_degC": 20,
      "maxDetectableRange_m": 8.856,
      "rangeResolution_cm": 4.3,
      "maxVelocity_kmph": 22.2408,
      "velocityResolution_kmph": 1.3896,
      "measurementRate": 4884,
      "typicalDetectedObjectRCS": 1
    }
  },
  "mmWaveDevices": [
  ...
}
```

```
$ batchProcessor.py --jsonProc ./processing.json --jsonSettings ./sensorSettings.json
```

Fig. 5 Sample JSON files and Batch Processing command.

requires two JSON files (Fig. 5), one for configuring sensor settings, and the other one for configuring variable processing elements we make available in Visualiser.

We incorporated several toolboxes such as Phased array system toolbox, Signal processing toolbox, Matlab compiler and Python sub-process. The user interface is built using Matlab GUI layouts toolbox. Batch processor is implemented as a Python script and is parallelised for optimisation purpose: it tries to determine how many parallel processes it should initialise based on available CPU resources.

The current version of the tool was tested only using Texas Instruments radar chips (Fig. 6). As the solution is available as open source and if details about radar system configuration are known, it should be possible to use the tool for data captured with other radar systems.

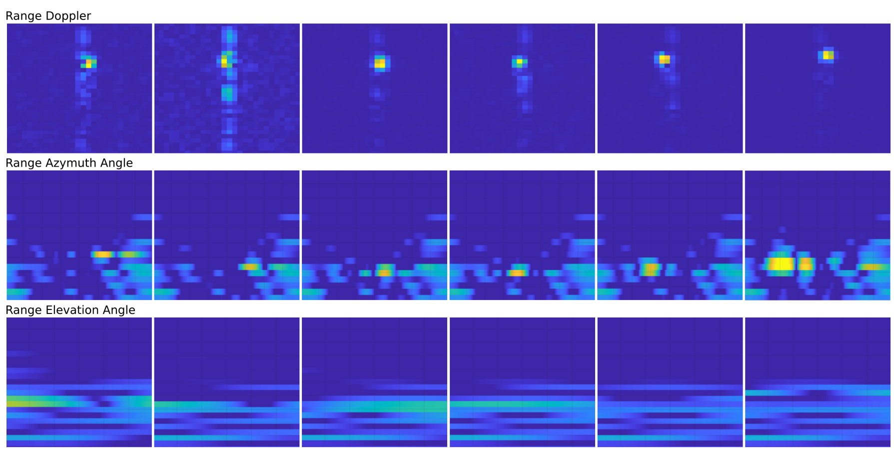

Fig. 6 Example output of processed sequences for gesture swipe right.

5 Conclusion and Future Work

We developed a Digital Signal Processor for Radar-Based HCI, which can generate several different representations (range-Doppler, range-Azimuth-Angle, range-Elevation-Angle, point cloud, IQ) of radar signal. The preliminary evaluation showed the tool performs well, however, it is limited in the number of different representations and sensors it supports. Thus, in the future, we will focus on expanding the functionality of the tool to other radar sensors and signal representations.

Acknowledgements We acknowledge the support from Slovenian Research Agency, grant number P1-0383, P5-0433 and J5-1796 and support from research program CogniCom (0013103) at the University of Primorska.

References

1. M. Altmann, P. Ott, N. C. Stache, and C. Waldschmidt. Multi-modal cross learning for an fmcw radar assisted by thermal and rgb cameras to monitor gestures and cooking processes. *IEEE Access*, 9:22295–22303, 2021.
2. M. G. Amin, Z. Zeng, and T. Shan. Hand gesture recognition based on radar micro-doppler signature envelopes. In *2019 IEEE Radar Conference*, pages 1–6, 2019.
3. N. T. Attygalle, L. A. Leiva, M. Kljun, C. Sandor, A. Plopski, H. Kato, and K. Čopič Pucihar. No interface, no problem: gesture recognition on physical objects using radar sensing. *Sensors*, 21(17):5771, 2021.
4. L.-F. Chen, C. K. Ong, C. Neo, V. V. Varadan, and V. K. Varadan. *Microwave electronics: measurement and materials characterization*. John Wiley & Sons, 2004.
5. K. Čopič Pucihar, C. Sandor, M. Kljun, W. Huerst, A. Plopski, T. Taketomi, H. Kato, and L. A. Leiva. The missing interface: micro-gestures on augmented objects. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.
6. B. Dekker, S. Jacobs, A. Kossen, M. Kruithof, A. Huizing, and M. Geurts. Gesture recognition with a low power fmcw radar and a deep convolutional neural network. In *2017 European Radar Conference*, pages 163–166, 2017.
7. E. Hayashi, J. Lien, N. Gillian, L. Giusti, D. Weber, J. Yamanaka, L. Bedal, and I. Poupyrev. Radarnet: Efficient gesture recognition technique utilizing a miniature radar sensor. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. Association for Computing Machinery, 2021.
8. S. Hazra, H. Feng, G. N. Kiprit, M. Stephan, L. Servadei, R. Wille, R. Weigel, and A. Santra. Cross-modal learning of graph representations using radar point cloud for long-range gesture recognition. In *2022 IEEE 12th Sensor Array and Multichannel Signal Processing Workshop*, pages 350–354, 2022.
9. S. Hazra and A. Santra. Radar gesture recognition system in presence of interference using self-attention neural network. In *2019 18th IEEE International Conference On Machine Learning And Applications*, pages 1409–1414, 2019.
10. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
11. J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018.
12. H. R. Lee, J. Park, and Y.-J. Suh. Improving classification accuracy of hand gesture recognition based on 60 ghz fmcw radar with deep learning domain adaptation. *Electronics*, 9(12), 2020.

13. H. Liu, Y. Wang, A. Zhou, H. He, W. Wang, K. Wang, P. Pan, Y. Lu, L. Liu, and H. Ma. Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 4(4), 2020.
14. S. Palipana, D. Salami, L. A. Leiva, and S. Sigg. Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 5(1), 2021.
15. J. Park, J. Jang, G. Lee, H. Koh, C. Kim, and T. W. Kim. A time domain artificial intelligence radar system using 33-ghz direct sampling for hand gesture recognition. *IEEE Journal of Solid-State Circuits*, 55(4):879–888, 2020.
16. S. M. Patole, M. Torlak, D. Wang, and M. Ali. Automotive radars: A review of signal processing techniques. *IEEE Signal Processing Magazine*, 34:22–35, 2017.
17. T. Sakamoto, X. Gao, E. Yavari, A. Rahman, O. Boric-Lubecke, and V. M. Lubecke. Hand gesture recognition using a radar echo i–q plot and a convolutional neural network. *IEEE Sensors Letters*, 2(3):1–4, 2018.
18. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations*, pages 1–14. Computational and Biological Learning Society, 2015.
19. A. Sluÿters, S. Lambot, and J. Vanderdonckt. Hand gesture recognition for an off-the-shelf radar by electromagnetic modeling and inversion. In *Proceedings of the 27th International Conference on Intelligent User Interfaces*, page 506–522. Association for Computing Machinery, 2022.
20. P. Stoica, Z. Wang, and J. Li. Robust capon beamforming. *IEEE Signal Processing Letters*, 10(6):172–175, 2003.
21. Y. Sun, T. Fei, F. Schliep, and N. Pohl. Gesture classification with handcrafted micro-doppler features using a fmcw radar. In *2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility*, pages 1–4, 2018.
22. K. Čopič Pucihar, N. T. Attygalle, M. Kljun, C. Sandor, and L. A. Leiva. Solids on soli: Millimetre-wave radar sensing through materials. *Association for Computing Machinery*, 6(EICS), 2022.
23. L. Wang, Z. Cui, Z. Cao, S. Xu, and R. Min. Fine-grained gesture recognition based on high resolution range profiles of terahertz radar. In *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, pages 1470–1473, 2019.
24. P. Wang, J. Lin, F. Wang, J. Xiu, Y. Lin, N. Yan, and H. Xu. A gesture air-writing tracking method that uses 24 ghz simo radar soc. *IEEE Access*, 8:152728–152741, 2020.
25. S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges. 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*, pages 851—860. Association for Computing Machinery, 2016.
26. X. Wang, R. Min, Z. Cui, and Z. Cao. Micro gesture recognition with terahertz radar based on diagonal profile of range-doppler map. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 770–773, 2020.
27. M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *European Conference on Computer Vision 2014*, pages 818–833. Springer International Publishing, 2014.
28. G. Zhang, S. Lan, K. Zhang, and L. Ye. Temporal-range-doppler features interpretation and recognition of hand gestures using mmw fmcw radar sensors. In *2020 14th European Conference on Antennas and Propagation*, pages 1–4, 2020.
29. L. Zheng, J. Bai, X. Zhu, L. Huang, C. Shan, Q. Wu, and L. Zhang. Dynamic hand gesture recognition in in-vehicle environment based on fmcw radar and transformer. *Sensors*, 21(19):6368, 2021.