IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 July 2022, accepted 12 August 2022, date of publication 17 August 2022, date of current version 29 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3199411

RESEARCH ARTICLE

End-to-End Dynamic Gesture Recognition Using MmWave Radar

ANUM ALI, (Senior Member, IEEE), PRIYABRATA PARIDA, VUTHA VA, (Member, IEEE), SAIFENG NI, KHUONG NHAT NGUYEN, (Member, IEEE), BOON LOONG NG, (Member, IEEE), AND JIANZHONG CHARLIE ZHANG

Standards and Mobility Innovation Laboratory, Samsung Research America, Plano, TX 75023, USA

Corresponding author: Anum Ali (anum.ali@samsung.com)

ABSTRACT Millimeter-wave (mmWave) radar sensors are a promising modality for gesture recognition as they can overcome several limitations of optic sensors typically used for gesture recognition. These limitations include cost, battery consumption, and privacy concerns. This work focuses on finger level (called micro) gesture recognition using mmWave radar. We propose a set of 6 micro-gestures that are not only intuitive and easy to perform for the user but are distinguishable based on Doppler and angle variation in time. For gesture recognition, we propose an end-to-end solution including an activity detection module (ADM) that automatically segments the data and the gesture classifier (GC) that takes the segmented data and predicts the gesture. Both the ADM and GC are based on machine learning (ML) tools. We evaluate the proposed solution using data collected from 11 users and our proposed solution achieves an end-to-end accuracy of 95%.

INDEX TERMS Human-computer interface, activity detection, gesture recognition, radar, machine learning.

I. INTRODUCTION

Voice and gestural interactions are becoming increasingly popular in the context of ambient computing. These input methods allow the user to interact with digital devices, e.g., smart TVs, smartphones, tablets, smart home devices, AR/VR glasses, etc., while performing other tasks, e.g., cooking and dining. Gestural interactions can be more effective than voice, particularly for simple interactions such as snoozing an alarm or controlling a media player. For such simple interactions, gestural interactions have two main advantages over voice-based interactions, namely, complication and social acceptability. First, the voice-based commands can often be long, and the user has to initiate with a hot word. Second, in quiet places and during conversations, voice-based interaction can be socially awkward.

Gestural interaction with a digital device can be based on different sensor types, e.g., ultrasonic [1], IMU [2], optic [3], and radar [4]. Optical sensors give the most favorable gesture recognition performance. The limitations of optic sensor based solutions, however, are sensitivity to ambient lighting conditions, privacy concerns, and battery consumption - hence the inability to run for long periods of time. LIDAR based solutions can overcome some of these challenges such as lighting conditions and privacy, but the cost is still prohibitive (currently, only available in high-end devices). These limitations are overcome by the radar based solutions. Specifically, millimeter-wave (mmWave) radar sensors are a particularly suitable choice. In addition to overcoming all the limitations of optic sensor based solutions, the mmWave radars are small in size making them suitable for mobile devices. Further, due to the ability of electromagnetic waves to pass through dielectric materials, the radar does not need to be visible on a mobile device.

Considering radar characteristics, we select dynamic gestures rather than static hand/finger poses in our solution. Radars have limited resolution in both the angle and range. Due to form-factor constraints, radar modules on mobile devices have only a few antennas, and as a result have limited angle resolution. While large bandwidths are available at mmWave bands, the range resolution is still several centimeters, which is too coarse for differentiating fingers'

The associate editor coordinating the review of this manuscript and approving it for publication was Orazio Gambino.

positions. Fortunately, radars have superb Doppler (speed) measurement capability. The Doppler resolution is inversely proportional to the duration of the radar coherent processing duration, which is a design parameter. The high Doppler resolution enables the radar to capture and potentially distinguish between subtle movements. As such, radars are suitable for distinguishing dynamic micro-gestures.

A. CONTRIBUTIONS

In this work, we propose an end-to-end solution for dynamic micro-gesture recognition. We consider 3 pairs of dynamic micro-gestures, i.e., total 6 gestures. After processing the raw radar data, first, the activity detection module (ADM) determines the portion of the data containing a gesture. This portion of data is then fed to the gesture classifier (GC) which predicts the performed gesture. The main contributions of this work are

1) We propose a set of 6, i.e., 3 pairs of intuitive dynamic micro-gesture. The selected gestures are easy to perform and remember, hence suitable for a good user experience. Further, the selected gestures have clearly distinguishable features, and as such are conducive to good classification.
2) We develop an ADM based on simple features of the input signal to determine a gesture end. The ADM is based on a tree based machine learning (ML) model chosen to have good gesture end detection performance and low computational complexity.
3) We develop a convolutional neural network (CNN) based GC. The proposed CNN model can predict the gestures with high accuracy and generalizes well to unseen users.
4) We evaluate our end-to-end solution on the data collected from 11 users and show a leave-one-out cross-validation (LOOCV) accuracy of 95%.

B. PRIOR WORK

There is a considerable amount of prior work on hand gesture recognition using radar, see e.g., [5], for a recent review. We start by discussing a series of articles by a group of researchers at Google [6], [7], [8]. While the first work [6] discussed the idea of using high range and Doppler resolution radars for dynamic gesture recognition, no evaluation results were provided. The subsequent work [7] considered 11 gestures - only 5 of these were micro-gestures and the rest of the gestures required full hand movements. The difficulty of classifying micro-gestures was clear from the results, as the classification accuracy on unseen users was reported to be as low as 59% for a micro-gesture. Further, only pre-segmented data was considered in [7], in which each segment contains one gesture. In a more recent work [8], the gesture vocabulary is simplified to only contain hand swipes (4 directional swipes and 1 swipe in any direction), in part to get good classification accuracy. In comparison with [7], we do not consider pre-segmented data, consider 6 micro-gestures, and can show 95% LOOCV accuracy. In comparison with [8], our gesture set is more complicated and allows richer interaction with the devices.

Most of the other prior work on gesture recognition using radar, e.g., [4], [9], [10], [11], [12], [13], and [14], considers macro-gestures, i.e., gestures based on hand level movements. Hand level movements have stronger signatures and hence are easier to classify. In comparison, we consider micro-gestures in this work. The prior work that does consider micro-gestures or a mix of macro-gestures and micro-gestures e.g., [15], [16], [17], [18], and [19], has the data pre-segmented, and hence the solution is incomplete. In this work, we provide a data-segmentation solution for online segmentation. Finally, the prior work that considers some micro-gestures and data-segmentation, e.g., [20], [21], [22], and [23], does not extend their evaluations to unseen users. In this work, we evaluate our data-segmentation and micro-gesture recognition solution for unseen users.

The rest of this paper is organized as follows: In Sec. II, we discuss the preliminaries of radar signal processing and the method to extract the required information from the processed signal. In Sec. III, we discuss the selected gesture vocabulary. In Sec. IV, we outline the details of the proposed approach. In Sec. V, we provide evaluation results to show the promise of the proposed strategy. Finally, we conclude the paper in Sec. VI and outline directions for future work.

Notation: We use the following notation throughout the paper. Bold lowercase $\mathbf{x}$ is used for column vectors, bold uppercase $\mathbf{X}$ is used for matrices, and non-bold letters x, X are used for scalers. Superscript T and $*$ represent the transpose and conjugate transpose respectively. The fast Fourier transform (FFT) output of a vector $\mathbf{x}$ is denoted as $\mathcal{X}$. The $N \times N$ identity matrix is represented by $\mathbf{I}_N$, and the $N \times 1$ zero vector is $\mathbf{0}_{N\times 1}$. The sets of complex and real numbers are denoted by $\mathbb{C}$ and $\mathbb{R}$, respectively.

II. RADAR SIGNAL PROCESSING

A. PRELIMINARIES OF RADAR SIGNAL PROCESSING

In this work, we use a mmWave monostatic frequency-modulated continuous wave (FMCW) radar with sawtooth linear frequency modulation. Let the operational bandwidth of the radar be $B = f_{\max} - f_{\min}$, where $f_{\min}$ and $f_{\max}$ are minimum and maximum sweep frequencies of the radar, respectively. The radar is equipped with a single transmit and N_r receive antennas. The receive antennas form a uniform linear array (ULA) with spacing $d_0 = \lambda_{\max}/2$, where $\lambda_{\max} = \frac{c}{f_{\min}}$ and c is the velocity of the light. As shown in Fig. 1, the transmitter transmits a frequency modulated sinusoid chirp of duration T_c over the bandwidth B. Hence, the range resolution of the radar is $r_{\min} = \frac{c}{2B}$ [24]. In the time domain, the transmitted chirp $s(t)$ is given as [25]

$$s(t) = A_T \cos(2\pi(f_{\min} t + \frac{1}{2} S t^2)), \tag{1}$$

where A_T is the transmit signal amplitude and $S = \frac{B}{T_c}$ controls the frequency ramp of $s(t)$. The reflected signal from an object is received at the N_r receive antennas. Let the object,

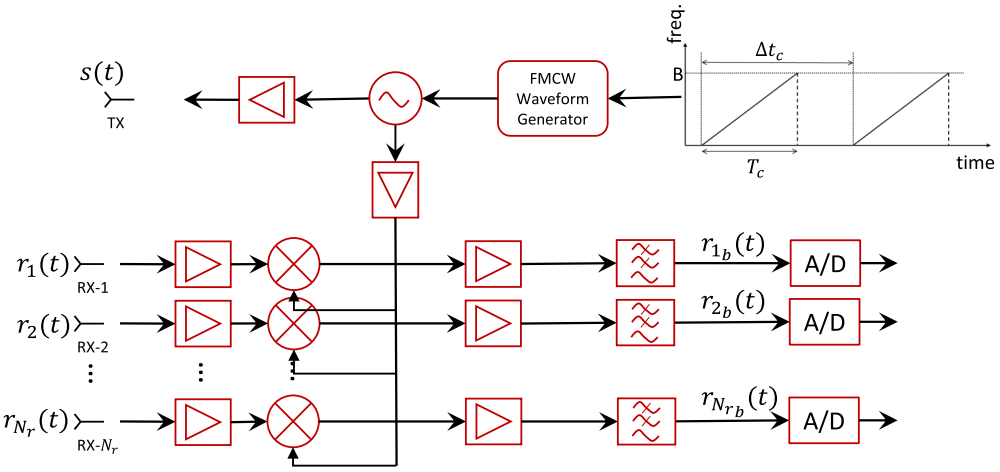


FIGURE 1. FMCW transceiver system diagram.

such as a finger or hand, be at a distance R_0 from the radar. Assuming one dominant reflected path, the received signal at the reference antenna is given as

$$r(t) = A_R \cos(2\pi(f_{\min}(t-\tau) + \frac{1}{2}S(t-\tau)^2), \tag{2}$$

where A_R is the amplitude of the reflected signal which is a function of A_T, distance between the radar and the reflecting object, and the physical properties of the object. Further, $\tau = \frac{2R_0}{c}$ is the round trip time delay to the reference antenna. The beat signal for the reference antenna is obtained by low pass filtering the output of the mixer. For the reference antenna, the beat signal is given as

$$\begin{aligned} r_b(t) &= \frac{A_T A_R}{2} \cos\left(2\pi\left(f_{\min}\tau + S\tau t - \frac{1}{2}S\tau^2\right)\right) \\ &\approx \frac{A_T A_R}{2} \cos(2\pi S\tau t + 2\pi f_{\min}\tau), \end{aligned} \tag{3}$$

where the last approximation follows from the fact that the propagation delay is orders of magnitude less than the chirp duration, i.e., $\tau \ll T_c$. The beat signal in (3) has two important parameters, namely the beat frequency $f_b = S\tau = S2R_0/c$ and the beat phase $\phi_b = 2\pi f_{\min}\tau$. The beat frequency is used to estimate the object range R_0. Further, for a moving target, the velocity can be estimated using beat phases corresponding to at least two consecutive chirps. For example, if two chirps are transmitted with a time separation of $\Delta t_c > T_c$, then the difference in beat phases is given as

$$\Delta\phi_b = 4\pi\frac{\Delta R}{\lambda_{\max}} = 4\pi\frac{v_0\Delta t_c}{\lambda_{\max}}, \tag{4}$$

where v_0 is the velocity of the object.

We obtain the beat frequency by taking the Fourier transform of the beat signal (3) that directly gives us the range R_0. To do so, the beat signal $r_b(t)$ is passed through an analog to digital converter (ADC) with sampling frequency $F_s = \frac{1}{T_s}$, where T_s is the sampling period. As a consequence, each chirp is sampled N_s times where $T_c = N_s T_s$. The ADC output corresponding to the n-th chirp is $\mathbf{x}_n \in \mathbb{R}^{N_s \times 1}$ and defined as $\mathbf{x}_n = [\{x[k,n]\}_{k=0}^{N_s-1}]$, where $x[k,n] = r_b(n\Delta t_c + kT_s)$. Let the N_s-point fast Fourier transform (FFT) output of $\mathbf{x}_n$ be denoted as $\mathcal{X}_n$. Assuming a single object, as we have considered so far, the frequency bin that corresponds to the beat frequency can be obtained as $k^* = \arg\max \|\mathcal{X}_n\|^2$. Since the radar resolution is $\frac{c}{2B}$, the n-th bin of the FFT output corresponds to a target located within $[\frac{kc}{2B} - \frac{kc}{4B}, \frac{kc}{2B} + \frac{kc}{4B}]$ for $1 \le k \le N_s - 1$. As the range information of the object is embedded in $\mathcal{X}_n$, it is also known as the range FFT.

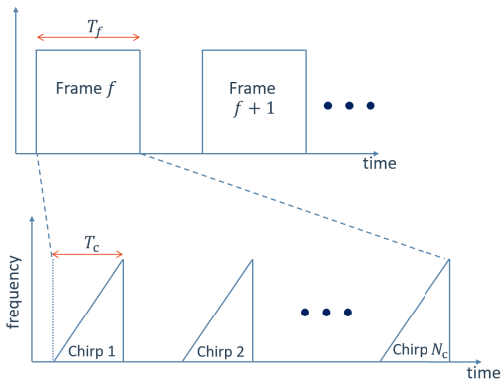


FIGURE 2. Frame-based radar transmission timing structure.

To facilitate velocity estimation, we adopt a radar transmission timing structure as shown in Fig. 2. The radar transmissions are divided into frames, where each frame consists of N_c equally spaced chirps. The range FFT of each chirp gives us the phase information on each range bin. For a given range bin, the Doppler spectrum, which has the velocity information, is obtained by applying N_c-point FFT across the range FFTs of chirps corresponding to that range bin. We construct the range-Doppler map (RDM) by repeating the above step for each range bin. Mathematically, we define $\mathbf{R} \in \mathbb{C}^{N_c \times N_s}$ as $\mathbf{R} = [\mathcal{X}_0, \mathcal{X}_1, \ldots, \mathcal{X}_{N_c-1}]^\mathsf{T}$. The RDM $\mathbf{M}$ is obtained by taking N_c-point FFT across all the columns of $\mathbf{R}$. The minimum velocity that can be estimated corresponds to the Doppler

resolution, which is inversely proportional to the number of chirps N_c and is given as

$$v_{\min} = \frac{\lambda_{\max}}{2N_c\Delta t_c}. \tag{5}$$

Further, the maximum velocity that can be estimated is given by

$$v_{\max} = \frac{N_c}{2} v_{\min} = \frac{\lambda_{\max}}{4\Delta t_c}. \tag{6}$$

1) CLUTTER REMOVAL

Since we have considered a monostatic radar, the RDM obtained using the above-mentioned approach has significant power contributions from direct leakage from the transmitting antenna to the receiving antennas. Further, the contributions from larger and slowly moving body parts such as the fist and forearm can be higher compared to the fingers. Since the transmit and receive antennas are static, the direct leakage appears in the zero-Doppler bin in the RDM. On the other hand, the larger body parts such as the fist and forearm move relatively slowly compared to the fingers. Hence, their signal contributions mainly concentrate at lower velocities. Since the contributions from both these artifacts dominate the desired signal in the RDM, it is desirable to remove them using appropriate signal processing techniques. The static contribution from the direct leakage is simply removed by nulling the zero-Doppler bin. To remove the contributions from slowly moving body parts, we pass the sampled beat signal of all the chirps in a frame through a first-order infinite impulse response (IIR) filter. For the reference frame f, the clutter removed samples corresponding to all the chirps can be obtained as

$$\hat{x}_f[k,n] = x_f[k,n] - \bar{y}_f[k,n-1] \tag{7}$$

$$\bar{y}_f[k,n] = \alpha x_f[k,n] + (1-\alpha)\bar{y}_f[k,n-1], \quad \text{for} \quad 0 \le k \le N_s - 1, 0 \le n \le N_c - 1, \tag{8}$$

where $\bar{y}_f[k,n]$ has contributions from all previous samples of different chirps in the frame.

III. GESTURE VOCABULARY

We considered several gestures before selecting the appropriate gesture set. We considered two attributes during the selection. The first attribute was the intuitiveness and simplicity of the gesture. Intuitive gestures are easy for the users to remember, and simpler gestures also imply more uniformity across users. To this end, we also considered gestures in pairs. The second desirable attribute was the distinguishability of the gesture in the considered features. As we use the time-velocity diagram (TVD) and the time-angle diagram (TAD) as the features for gesture classifications, we were interested in a gesture set, in which each gesture is distinguishable from the other gestures either in TVD or TAD (more on TVD and TAD in Sec. IV-A, and more on the distinguishability of gestures in Sec. V). With these attributes in mind, we considered a total of 14 gestures. In addition to the selected 6 gestures, the other considered 8 gestures include, index extension, in which the index finger is extended towards the radar and is subsequently contracted, clockwise circle, counter-clockwise circle, left-half circle, right-half circle, and slide (of thumb on index finger), open only (from touching thumb and index fingers separating), and close only (from the separated thumb and index fingers to touching). The selected gesture set is shown in Fig. 3. The gesture set contains 3 pairs of gestures, i.e., a total of 6 gestures. Specifically, there is a pair of circles, a pair of pinches, and a pair of swipes. The pair of circles contains a radial circle and a tangential circle. The names radial and tangential come from the movement of the finger relative to the radar. As the name implies in the radial circle the movement of the finger is radial to the radar, whereas in the tangential circle the movement is tangential to the radar. The pair of pinches contain a single pinch and a double pinch. Finally, the pair of swipes contains two directional swipes, i.e., a left-to-right swipe and a right-to-left swipe.

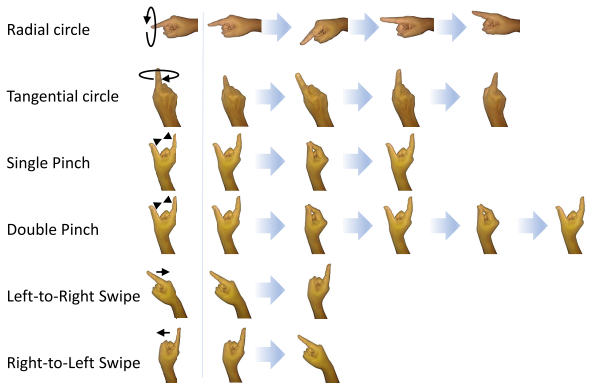


FIGURE 3. The proposed gesture set of 6 gestures.

IV. APPROACH AND METHODOLOGY

The overall system diagram of the proposed solution is given in Fig. 4. The first block is the triggering mechanism that triggers the gesture detection mode. The next block is the ADM which determines the end of a gesture to trigger the GC. The GC predicts the gesture performed by the user.

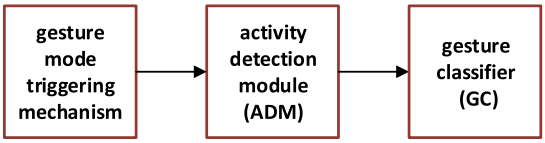


FIGURE 4. Overall system diagram of the proposed solution.

First, we discuss the signal processing performed on the radar signal discussed in Sec. II to obtain the features that are used for gesture detection. Subsequently, we discuss all the three blocks of the overall system diagram in Fig. 4.

A. SIGNAL PROCESSING TO OBTAIN THE FEATURES

As we are interested in dynamic micro-gestures, the variation in the received signal as a function of time should

highlight the unique signatures of different gestures. The variation in the range is not particularly important in the context of micro-gesture recognition because the movement of the fingers is on the order of a few centimeters. As the range resolution itself is several centimeters, e.g., 3 cm for a radar with 5 GHz bandwidth, the variation in range, if any, is quite coarse. The variation in Doppler and angle is thus more important for our application. The variation in Doppler as a function of time is captured through TVD. Similarly, the variation in angle is captured via TAD. In this work, we use TVD and TAD as features for gesture classification. In the next two subsections, we highlight how to obtain the TVD and TAD from the radar data.

1) TIME-VELOCITY DIAGRAM (TVD)

The procedure for obtaining the TVD from the RDM is shown in Fig. 5. Using the clutter removed and zero-Doppler nulled RDM, for a given frame, we obtain the range profile by summing the power across all Doppler bins. The range profile is compared with a detection threshold to extract the range information of the target of interest. In this work, we consider the first detected peak in the range profile as the location of the desired object. Specifically, the first peak above the detection threshold is considered to contain the moving finger. This is based on the observation that in a typical use case, the gesture is the closest moving target to the radar. The detection threshold itself varies with range to accommodate the leakage residual in the first few taps. As such, the detection threshold on the first few taps is chosen higher than the subsequent taps. How many taps and how much offset is applied to the detection threshold is determined based on measurements. We observed that for the radar kits we experimented with, these thresholds only depend on the choice of the radar parameters and stay consistent across kits and time. Thus, these thresholds, once determined, can be used across platforms and time. Once the first peak is known, the Doppler from the RDM for the tap corresponding to the first peak is used to construct the TVD.

2) TIME-ANGLE DIAGRAM (TAD)

For the TVD generation, the received signal from any of the antennas can be considered. However, for the TAD generation, the beat signals from all antennas need to be considered. The process of TAD generation is shown in Fig. 6. Assuming the target is located at an angle θ_0 with respect to the end fire of the ULA, the beat signal for antenna $i \geq 1$ is given as

$$r_{i_b}(t) \approx \frac{A_T A_{R_i}}{2} \cos(2\pi S\tau_i t + 2\pi f_{\min} \tau_i), \tag{9}$$

where $\tau_i = \frac{2R_0+(i-1)d_0\cos(\theta_0)}{c}$. Since $R_0 \gg d_0$, the beat frequency at the i-th antenna is $S\tau_i \approx S\tau_1, \forall i$. On the other hand, the spatial angle information is easily extracted using the phases of the beat signals across the antennas. For a given frame f, the sampled ADC output corresponding to the n-th chirp for the i-th antenna is given as

$$\mathbf{x}_{i,n,f} = [\{x_{i,f}[k,n]\}_{k=0}^{N_s-1}], \tag{10}$$

where $x_{i,f}[k,n] = r_{i_b}(n\Delta t_c + kT_s)$. To extract the angle information using all the chirps in f-th frame, in the first step, we compute the range FFT $\mathbf{R}_{i,f} \in \mathbb{C}^{N_c \times N_s}$ of all the chirps for each antenna i. Note that $\mathbf{R}_{i,f} = [\mathcal{X}_{i,0,f}, \mathcal{X}_{i,1,f}, \ldots, \mathcal{X}_{i,N_c-1,f}]^\mathsf{T}$, where $\mathcal{X}_{i,n,f}$ is the range FFT corresponding to $\mathbf{x}_{i,n,f}$. Let the target location corresponding to the range bin index b_0 and $\mathbf{r}_{i,f,b_0}$ be the corresponding column in $\mathbf{R}_{i,f}$. We define $\mathbf{B}_{f,b_0} \in \mathbb{C}^{N_r \times N_c}$ as $\mathbf{B}_{f,b_0} = [\mathbf{r}_{1,f,b_0}, \mathbf{r}_{2,f,b_0}, \ldots, \mathbf{r}_{N_r,f,b_0}]^\mathsf{T}$. In the second step, we empirically obtain the co-variance matrix of the received signal across the antennas. Mathematically, we write

$$\mathbf{C}_{f,b_0} = \frac{\mathbf{B}_{f,b_0}\mathbf{B}_{f,b_0}^*}{N_c}. \tag{11}$$

In the third step, using $\mathbf{C}_{f,b_0}$, we obtain the spatial spectrum of the target at bin b_0 leveraging the MUSIC algorithm [26]. Other direction estimation algorithms can also be used instead of MUSIC. Formally, we decompose $\mathbf{C}_{f,b_0}$ using the eigenvalue decomposition and separate the signal and noise subspaces as follow

$$\mathbf{C}_{f,b_0} = \mathbf{U}_{s,f,b_0}\Lambda_{s,f,b_0}\mathbf{U}_{s,f,b_0}^* + \sigma_n^2\mathbf{U}_{n,f,b_0}\mathbf{U}_{n,f,b_0}^*, \tag{12}$$

where Λ_{s,f,b_0} is a diagonal matrix containing the eigenvalues of $\mathbf{C}_{f,b_0}$ corresponding to the signal and σ_n^2 is the noise variance. Separating signal and noise subspaces is not a trivial problem, but for our current setup we assume single target and always pick the subspace corresponding to the strongest eigenvalue as the signal subspace. In the next step, the angular spectrum is obtained as

$$P_{f,b_0}(\beta) = \frac{1}{\mathbf{a}(\beta)\mathbf{U}_{n,f,b_0}\mathbf{U}_{n,f,b_0}^*\mathbf{a}(\beta)}, \tag{13}$$

where $\mathbf{a}(\beta) = [1, e^{-j\frac{2\pi d_0}{\lambda_{\min}}\cos(\beta)}, \ldots, e^{-j\frac{2\pi(N_r-1)d_0}{\lambda_{\min}}\cos(\beta)}]$. The peak of the spectrum is attained at $\beta = \theta_0$. To construct the TAD column for the f-th frame, we evaluate $P_{f,b_0}(\beta)$ for $\beta \in \left[0, \frac{\pi}{N_c}, \ldots, \frac{\pi(N_c-1)}{N_c}\right]$. This choice of β is selected to match the dimension of TVD.

B. TRIGGERING THE GESTURE DETECTION MODE

The first block in Fig. 4 is the gesture mode triggering mechanism. There are several possibilities for this triggering mechanism, e.g., based on proximity detection and/or active applications, etc. In a proximity detection based trigger, the gesture mode is activated only when an object near the radar is detected. The proximity detection mode can itself be based on the radar used for gesture detection. The benefit of triggering the gesture mode based on proximity detection comes in reduced power consumption. It is expected that a simpler task of proximity detection can be achieved reliably with radar configurations that have low power consumption. It is only when an object is detected in radar's proximity, that we switch to the gesture detection mode, which could be based on radar configuration that consumes more power. Another possibility for triggering the gesture mode is application based. As an

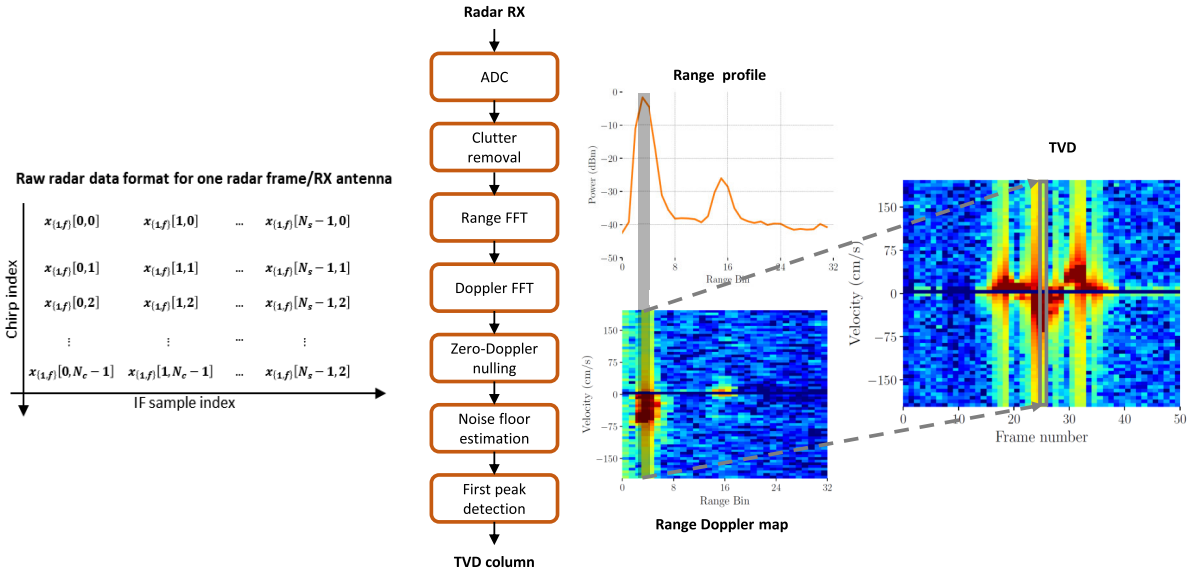


FIGURE 5. The process of TVD column generation from a frame.

example, the dynamic finger gestures may be used with only a few applications, and as such, the gesture mode needs to be triggered only when the user is actively using the application exploiting gestural interaction. As our primary focus in this work is gesture detection, we do not delve deeply into the implementation of the triggering mechanism.

C. ACTIVITY DETECTION MODULE (ADM)

The second block in Fig. 4 is the ADM. When the gesture recognition system is activated, the data is continuously captured from the radar. The GC - discussed later in Sec. IV-D, however, needs to be triggered only when the gesture is performed. The purpose of the ADM is to determine the end of a gesture and subsequently trigger the GC. From a design perspective, the ADM can be based on some rules devised to determine the activity. For example, the rules could be based on the level of Doppler and how it varies with time to determine the gesture ends. The limitation of the rule based method is that if the gesture vocabulary or the radar parameters are to be revised, it is likely that the rules may also need to be revised and/or refined, making the rule based method laborious in practice. An alternative design is data-driven in which an ML model is trained to determine the gesture ends. The data-driven method will require data-collection and training whenever the gesture set or radar parameters change but eliminates the need to re-engineer the rules.

Our solution is based on a binary classifier followed by an accumulator, as shown in Fig. 7. The function of the accumulator is to keep track of the predictions of the binary classifier. As long as the condition to trigger the GC is not satisfied, the operation of the binary classifier and the accumulator continues. Once the condition is satisfied, the GC is triggered. We now discuss the operation of the binary classifier and the accumulator in detail.

1) THE BINARY CLASSIFIER

Features derived from TVD are used for this classifier. Although more information such as range and angle may also be considered, we found that using TVD alone already provides satisfactory performance. In the following, we describe our solution that only uses features derived from TVD.

The binary classifier predicts whether a gesture has ended or not. As the TVD is updated at the frame rate, the binary classifier will operate at the same or lower rate than the frame rate. For ease of exposition, we assume that the binary classifier makes one prediction per frame in subsequent discussions. In every frame, the prediction of the classifier is either "class 0" implying that the gesture has not ended, or "class 1" implying that the gesture has ended. For training, "class 0" and "class 1" samples are generated. Specifically, consider the TVD of a gesture shown in Fig. 8. For this given TVD, the ground truth ending of the gesture is marked by the user as shown via the red vertical line. The ground truth endings are marked based on visual observations. Subsequently, the TVD is shifted so that the ground truth ending frame is now at the last frame (in our example 50), as shown in Fig. 8b. Subsequently, a "class 0" sample is generated by shifting the TVD to the right by a random number of frames. The "class 0" sample obtained by a shift of 5 frames is shown in Fig. 8c. This way the end of the gesture is not within the TVD, and hence the TVD corresponds to the case when the gesture has not ended. Similarly, a "class 1" sample is generated by shifting the TVD to the left as shown in Fig. 8d for a shift of 5 frames. This way the end of the gesture is within the TVD, and hence the TVD corresponds to a case when the gesture has ended. Note that since "class 0" and "class 1" samples are generated by applying random offsets/shifts to the same TVD, multiple "class 0" and "class 1" samples can be generated from a single TVD. Further, the

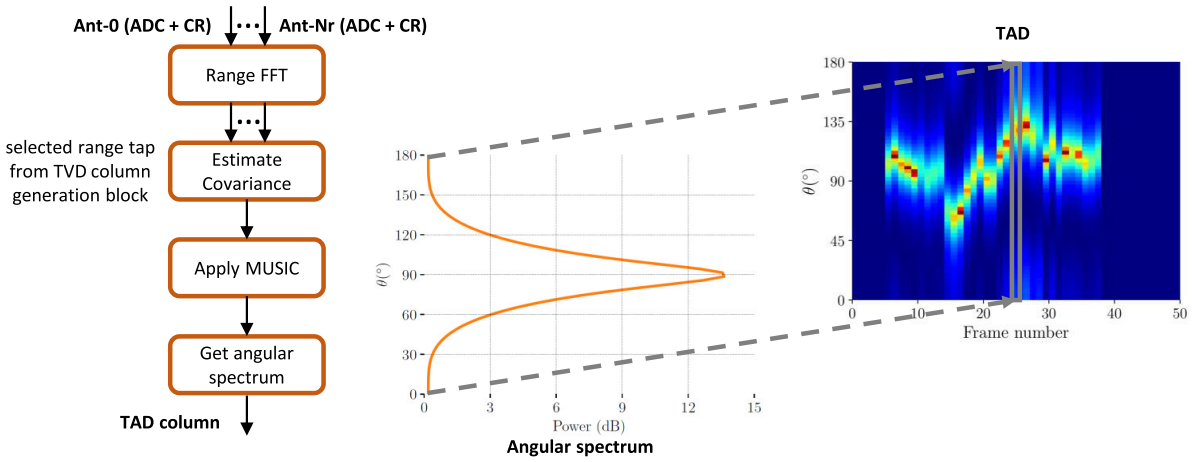


FIGURE 6. The process of TAD column generation from a frame.

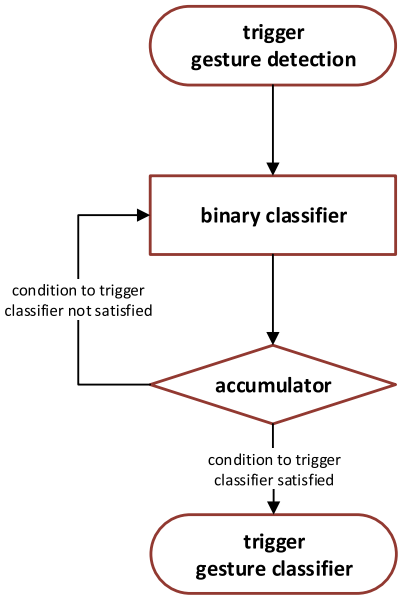


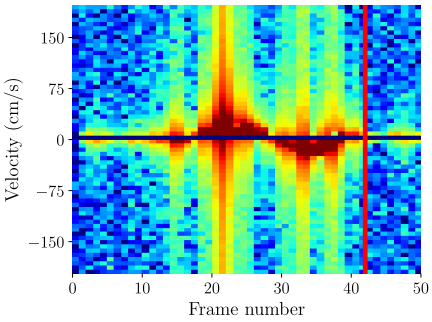
FIGURE 7. Binary classifier and accumulator based ADM.

"class 0" samples thus generated represent the case in which the gesture has started but not ended. The ADM, however, is continuously operational and it needs to make predictions even in cases when no gesture is being performed. To deal with this, we need to have representation in the training set of the case when the gesture has not even started, i.e., no-activity. For this, we collect data at the ending position of each gesture in the gesture set. An example of the static finger or no-activity "class 0" sample is shown in Fig. 8e.

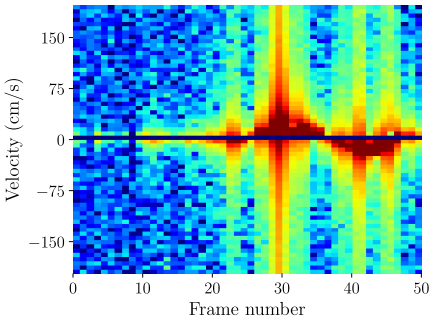
The "class 0" and "class 1" TVD samples shown in Fig.8 can be considered $N_c \times F$ gray-scale images. In the examples in Fig. 8, $F = 50$. A binary classifier can thus be trained based on these images directly, e.g., based on a convolutional neural network (CNN). It is, however, vital to keep the complexity of the ADM model low. This is because the ADM classifier can be required to predict the frame rate. To reduce the computational complexity of the classifier, we seek simpler features so that a simpler model can be trained. Specifically, we seek to collapse the Doppler dimension of the TVD. To this end, several strategies can be tried, e.g., taking the mean in the Doppler dimension in the linear or the log scale. For brevity, however, we limit our discussion to a feature that gave the most promising results. This feature $\mathbf{d} \in \mathbb{R}^F$ is calculated from the linear version of the TVD $\mathbf{T}_l = 10^{\mathbf{T}/10}$. Here $\mathbf{T}$ is the TVD in dB, and the power and division operations are element-wise. The feature $\mathbf{d}$ called the power weighted Doppler normalized by maximum (PWDNM) is given below

$$\mathbf{d} = \frac{\bar{\mathbf{d}}}{\max_j |\bar{\mathbf{d}}[j]|}, \quad \bar{\mathbf{d}}[j] = \sum_{k=-\frac{N_c}{2}}^{\frac{N_c}{2}-1} k\mathbf{T}_l[k,j]. \tag{14}$$

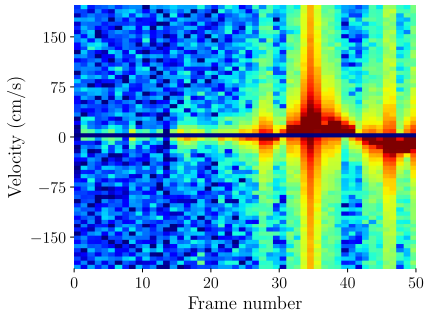
The rationale for the name PWDNM is clear from the definition (14). The Doppler k is weighted by the power $\mathbf{T}_l[k,j]$ and the result $\bar{\mathbf{d}}$ is normalized by the maximum absolute value to get the feature entries $\mathbf{d}[j] \in [-1,1]$. This feature is designed to have some desirable properties. The first is to put higher weight on high Doppler bins by scaling the power in k-th bin by k. The lower Doppler bins might contain some leftover clutter - after clutter removal - in addition to the signal. As such, the power in the high frequency bins is a stronger indicator of gestural activity, and can better distinguish the gesture part from the non-gesture part for activity detection. The normalization by the maximum absolute value helps to generalize the feature across users (some users having a stronger/weaker signature than the others). The PWDNM feature of the "class 0" and "class 1" samples in Fig. 8 are shown in Fig. 9. From Fig. 9a, we see that the feature assumes a non-negligible value for $j = 50$, implying a high activity. From Fig. 9b, we see a small value (close to 0) for $j = 50$. As the preceding values, i.e., for $j = 20$ to $j = 43$ are high, followed by a small value, it is a good indication that the gesture has ended. Finally, in Fig. 9c, we see that almost



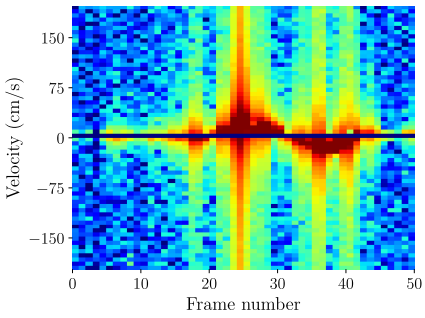
(a) The TVD of a gesture with ground truth marked.



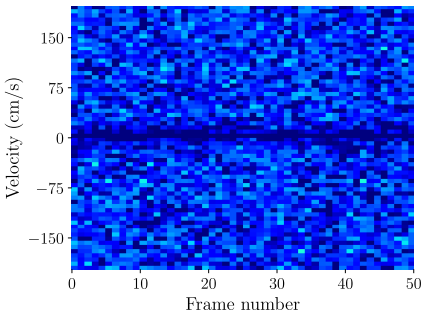
(b) The TVD of a gesture with ground truth shifted to frame 50.



(c) The TVD of a "class 0" sample obtained from Fig. 8b with 5 frame shift to the right.



(d) The TVD of a "class 1" sample obtained from Fig. 8b with 5 frame shift to the left.



(e) The TVD of a "class 0" sample representing no-activity.

FIGURE 8. **The "class 0" and "class 1" samples generated from a gesture, and "class 0" sample representing no-activity.**

for all $j \in 1, 2, \cdots, 50$, the feature assumes a non-negligible value. Note that, though there is no-activity, the normalization means that the largest value will always be 1. As such, large values for all j are an indication that the gesture has not ended.

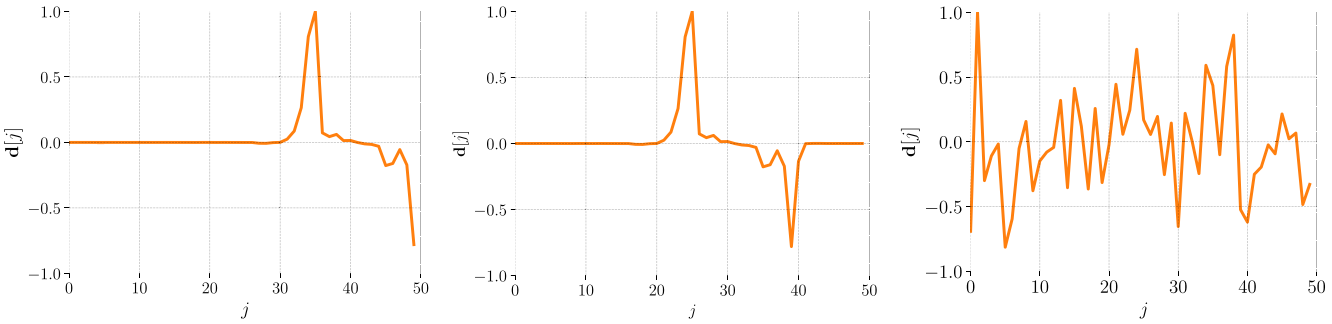
2) THE ACCUMULATOR

The purpose of the accumulator is to robustify the prediction of the binary classifier, and it declares a gesture end detected only when it has enough confidence. The binary classifier predicts the end of a gesture. These predictions are then collected through the accumulator. The GC is triggered only when a predetermined accumulation condition is met. The rationale for accumulating predictions is twofold. First, the classifier is imperfect, and occasionally it predicts that the gesture has ended, whereas, in reality, it has not. Secondly, some delay is required to make sure that the gesture has ended in reality. To this end, a good example is the case of single pinch and double pinch (see Fig. 13c and Fig. 13d). The double pinch inherently contains two single pinch gestures. If the user intends to perform a double pinch, then after the first pinch, if there is no delay, the gesture classifier will be triggered and will predict a single pinch. In contrast, if there is enough delay, then the user will start the second pinch in a double pinch, and hence only after the user completes the whole double pinch gesture, the classifier will be triggered.

Just like other design choices, several accumulation methods can be used. One simple possibility is to wait for consecutive N "class 1" outcomes before triggering the gesture classifier. Here N is a parameter that provides a trade-off between accuracy and delay. In this accumulation method, the counter to N is completely reset whenever the prediction is "class 0". The limitation of this simple method is due to the imperfection in the classifier predictions. If due to imperfection, the classifier predicts "class 0" instead of "class 1", the counter will be reset. Particularly, if the counter has already reached a value close to N, resetting the counter to 0 based on a single "class 0" prediction, implies throwing away all the information contained in the previous few frames. To overcome this limitation, we penalize the counter whenever the prediction is "class 0", but do not completely reset it. The proposed accumulation algorithm is given in Algorithm 1. If there is a "class 1" prediction, the counter c is incremented, if the prediction is "class 0", the counter c is decremented. Whenever the counter c reaches the value N, the gesture classifier is triggered, and the counter is reset to 0 to look for the subsequent gesture. In the proposed strategy, a higher value of N will still give more delay but also more confidence in the prediction of gesture ends.

D. GESTURE CLASSIFIER (GC)

The third block in Fig. 4 is the GC. We now discuss the neural network architecture that was designed to classify the gestures in Fig. 3. We choose to design a deep CNN because CNN has been shown to work remarkably well with tasks that involve unstructured data such as images, audio, and text [27] due to its ability to automatically extract features in multiple



(a) The feature of a "class 0" sample shown in Fig. 8c.

(b) The feature of a "class 1" sample shown in Fig. 8d.

(c) The feature of a "class 0" sample shown in Fig. 8e.

FIGURE 9. The PWDNM feature of the "class 0" and "class 1" samples in Fig. 8.

Algorithm 1 The Accumulation Algorithm

Initialization: $c \leftarrow 0$
Input: p_i
if $p_i == 1$: **then**
  $c \leftarrow c + 1$
else if $p_i == 0$: **then**
  $c \leftarrow \max(c - 1, 0)$
end if
if $c == N$ **then**
  Trigger the gesture classifier
  $c \leftarrow 0$
end if

stages (layers) and learn the patterns and representations of the input from raw data.

Fig. 10 summarizes the architecture of our network. Specifically, it shows the transformation of the input through multiple layers of the network. In discussing the dimensions of the network, we assume that $N_c = 64$ and $F = 50$, which are also the parameters used in evaluations given later in Sec. V. The architecture consists of

Two Convolutional layers
- First with 64 channels and a kernel size of (7, 8).
- Second with 32 channels and a kernel size of (2, 3).

Two MaxPool layers
- First with a kernel size of (4, 4).
- Second with a kernel size of (2, 2).

Two Dense layers
- First with a size of 32.
- Second with a size of 6 (number of gestures).

Two Blurpool layers [28]
- To improve the robustness of the model against input-shifting, one BlurPool layer is inserted after each convolutional layer.
- Both of the BlurPool layers have a kernel size of (3, 3).
- We will provide some experimental results on the effect of BlurPool in Section V-F.

A Softmax layer
- We add a softmax activation function to the last dense layer to enable multi-class classification.

Regularization layers
- To avoid over-fitting, batch normalization [29], and drop-out with a drop-out rate of 0.1 [30] is used.
- 3 batch normalization layers and 2 drop-out layers are used.

V. EXPERIMENTAL RESULTS AND ANALYSIS

A. EXPERIMENTAL PLATFORM

The platform used for the gesture recognition is shown in Fig. 11. The Infineon radar baseboard MCU7, which is a 60 GHz radar system platform is used. The sensor board is BGT60TR24C which has 2 transmit and 4 receive antennas. The layout of the transmit and receive antennas is shown in Fig. 12. Only TX1, RX1, and RX3 are used in our evaluation, which are underlined in Fig. 12. The sensor board is connected to a laptop computer via USB. Subsequently, the data collected from the radar is processed in Python, including clutter removal, RDM calculation, range profile estimation, TVD/TAD generation, and calculating the feature for ADM. The ADM module is based on XGBoost, and the GC is implemented using the open-source Keras library [31] supported by Tensorflow [32].

B. RADAR AND SYSTEM PARAMETERS

We summarize the radar and system parameters in Table 1. The radar operates in the 60 GHz band, with $f_{\max} = 63$ GHz, $f_{\min} = 58$ GHz, bandwidth $B = 5$ GHz, and the range resolution $r_{\min} = 3$ cm. We set the number of samples per chirp N_s, and the number of chirps per frame N_c both to 64. Note that, TVD is selected as a feature in this work because due to the dynamic nature of the gestures, the Doppler - and the variation of Doppler across time - contain useful information about the gestures. The purpose of the radar parameters selection is then to maximize the Doppler resolution while ensuring a high enough frame rate. To this end, we set the frame rate to be 25 frames per second, i.e., $F = 50$ frames in the 2 seconds period considered to contain the gesture. After setting

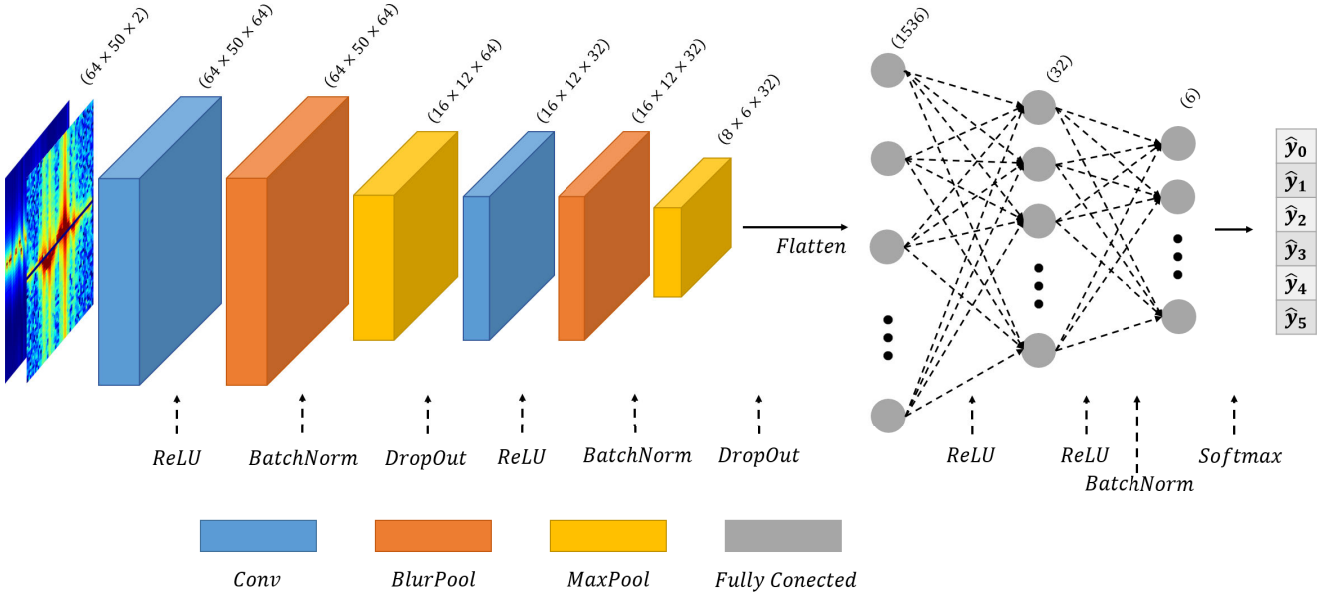


FIGURE 10. Summary of our CNN architecture used for gesture classification in this study.

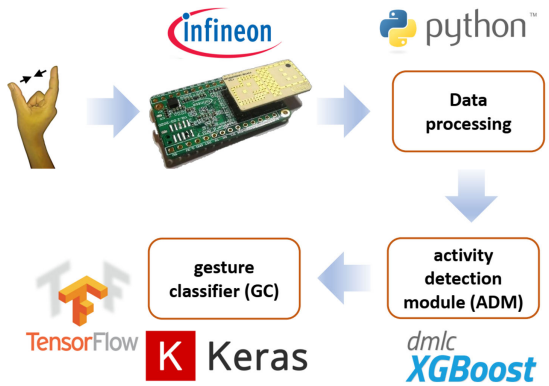


FIGURE 11. The platform used in evaluations.

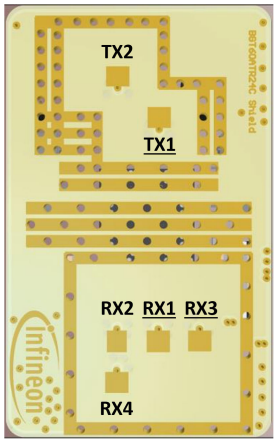


FIGURE 12. The layout of the transmit and receive antennas in BGT60TR24C. Only the underlined antennas are used in evaluations.

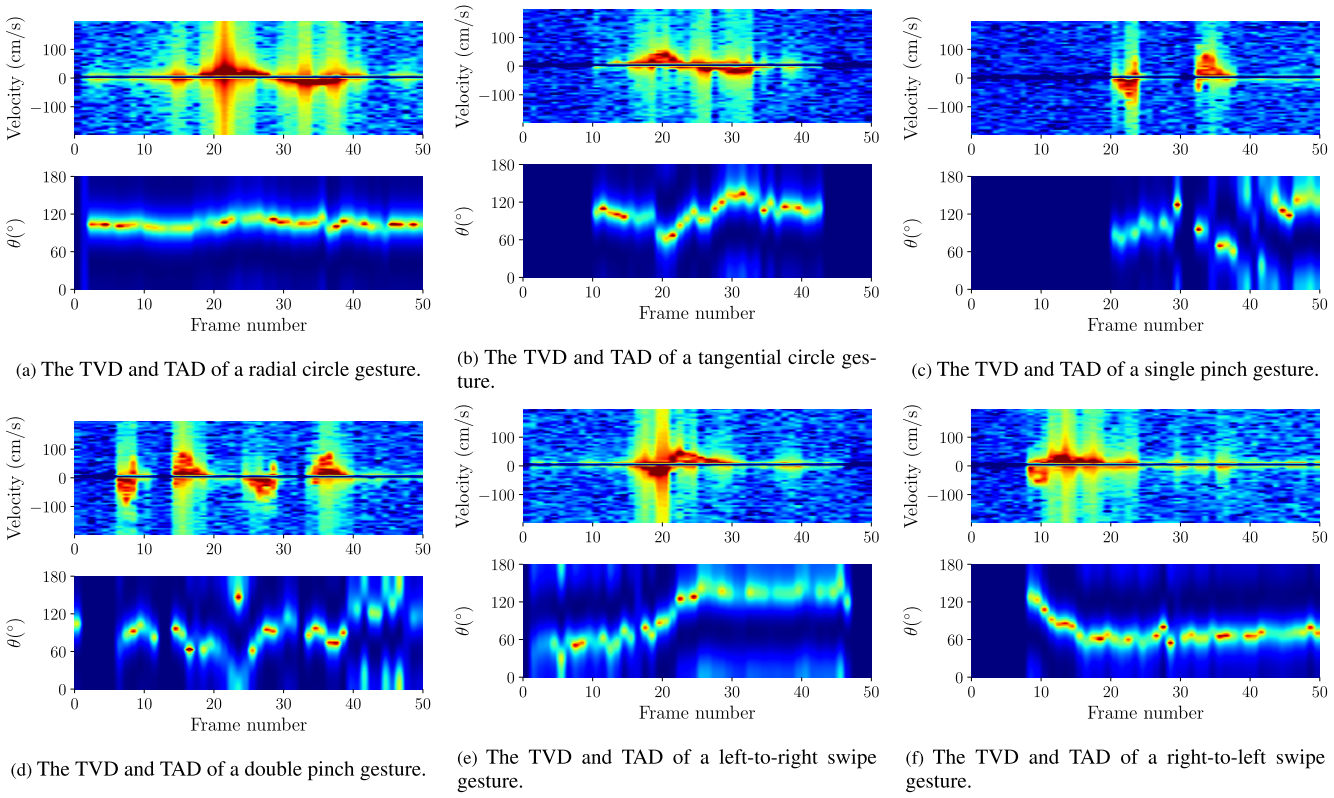
the aforementioned parameters for the Infineon radar, the chirp-to-chirp time T_f/N_c was 640 µs. The aforementioned parameters result in the Doppler resolution of $v_{\text{min}} = 6.15\,\text{cm}\,\text{s}^{-1}$. For the clutter removal, we use a value of $\alpha = 0.2$ in our experiments.

TABLE 1. Summary of the parameters.

Parameter	Value
Minimum sweep frequency - f_{min}	58 GHz
Maximum sweep frequency - f_{max}	63 GHz
Bandwidth - B	5 GHz
Samples per chirp - N_{s}	64
Chirps per frame - N_{c}	64
Antennas - N_{r}	2
Frames in 2 seconds - F	50
IIR filter parameter - α	0.2

C. DISTINGUISHABILITY OF THE GESTURES IN TVD/TAD

In Fig. 13, we show the TVD/TAD of all the gestures in the gesture vocabulary set. In each figure, the TVD is shown on top, whereas the TAD is shown at the bottom. Due to the clear movement towards and away from the radar, the radial circle has a clear and strong signature in the TVD. The TAD, however, does not change much throughout the gesture duration. In contrast, for tangential circle, the Doppler signature is not as strong, but the variation of the angle in TAD is quite clear. Further, for a single pinch, we can see two regions of activity in the TVD, whereas, for a double pinch, we can see four regions of activity in the TVD. Finally, for the left-to-right swipe and right-to-left swipe, the starting and finishing angles are opposite of each other. In summary, all six gestures have a unique TVD/TAD signature that allows its prediction when TVD/TAD are used as features.



(a) The TVD and TAD of a radial circle gesture.

(b) The TVD and TAD of a tangential circle gesture.

(c) The TVD and TAD of a single pinch gesture.

(d) The TVD and TAD of a double pinch gesture.

(e) The TVD and TAD of a left-to-right swipe gesture.

(f) The TVD and TAD of a right-to-left swipe gesture.

FIGURE 13. The TVD and TAD of all the gestures in the gesture vocabulary set based on the selected radar parameters. The examples show that each gesture is distinguishable from all other gestures in either TVD or TAD.

D. DATA-COLLECTION

In our experimental setup, the gestures are performed at 10 cm in the radar boresight. Each gesture sample is performed in a window of 125 frames ($\sim$ 5 s) with the understanding that the duration of the gesture would not exceed 50 frames ($\sim$ 2 s). The data is collected from 11 users, i.e., U1-U11. Each of the 11 users performed 600 gestures, i.e., 100 gestures per gesture-class per user. For users U1-U4, we collected an additional 600 samples from each user for ADM training.

E. ACTIVITY DETECTION MODULE

The evaluation of the ADM is based on all 11 users. We use 100 gestures per gesture-class per user in the evaluation. For training, we use the additional data collected from users U1-U4. For all the data, ground truths are marked as discussed in Sec. IV-C1. For each of the samples, 3 "class 1" samples are generated by applying a random offset to the left. This offset is picked uniformly at random in $\{1, 2, \cdots, 10\}$. This gives us number of gestures-per-class $\times$ number of gesture-classes $\times$ number of users $\times$ number of offsets $= 100 \times 6 \times 4 \times 3 = 7200$ "class 1" samples. For "class 0", 2 samples are generated from each gesture. The offsets to the right is also picked uniformly at random in $\{1, 2, \cdots, 10\}$. With 2 samples per gesture, we have a total of 4800 "class 0" samples. The remaining $7200 - 4800 = 2400$ samples are obtained from the data containing no activity. This data is based on the ending positions of all the 6 gestures. For data collection, the user placed his hand/finger in one of the gestures ending position for 40 s and 1000 frames were collected. For generating "class 0" samples from the collected data, 1 out of the 6 gesture ending position is selected at random, and subsequently, a consecutive window of 50 frames starting at a random position in the $1000 - 50$ frames is used. The PWDNM feature discussed in Sec. IV-C1 is calculated for all the $7200 \times 2 = 14400$ samples, and a binary classifier is trained. In this work, we compare extreme gradient boosting (XGBoost) [33] and long short-term memory (LSTM) [34]. We train the XGBoost classifier with a binary logistic objective and the area under the precision-recall curve (AUCPR) evaluation metric. The number of early stopping rounds is 100, and the remaining parameters are the default parameters of the XGBoost [35]. The training/test split ratio is 0.9/0.1. The LSTM based binary classifier has two layers. The first is an LSTM layer with 32 units, and the second is a dense layer with 1 unit. The hyperbolic tangent activation is used for the LSTM layer, and sigmoid activation is used for the dense unit. The classifier is trained with binary cross-entropy loss and adaptive moment estimation (ADAM) optimization, and the learning rate is 0.01. The model is trained for 200 epochs, and we pick the model with the best validation accuracy. The train/validation/test split is 0.81/0.09/0.1. The parameter N of the accumulator is set to 11.

TABLE 2. The performance of the ADM based on XGBoost binary classifier.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	Mean
Radial Circle	100	100	100	100	100	100	99.0	100	100	99.0	100	99.82
Tangential Circle	100	100	100	100	100	100	87.0	100	100	100	100	98.82
Double Pinch	100	100	100	100	99.0	100	100	98.0	100	100	100	99.73
Single Pinch	100	100	100	100	100	100	99.0	99.0	100	100	100	99.82
Left-to-Right Swipe	100	95.0	100	100	100	100	97.0	100	100	100	100	99.27
Right-to-Left Swipe	100	99.0	100	100	100	100	100	100	100	98.0	100	99.73
Mean	100	99.0	100	100	99.8	100	97.0	99.5	100	99.5	100	99.53

TABLE 3. The performance of the ADM based on LSTM binary classifier.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	Mean
Radial Circle	100	100	100	100	97.0	89.0	100	100	98.0	99.0	100	98.45
Tangential Circle	100	100	100	98.0	94.0	99.0	91.0	68.0	72.0	98.0	84.0	91.27
Double Pinch	100	98.0	100	98.0	99.0	68.0	96.0	88.0	98.0	94.0	100	94.45
Single Pinch	100	98.0	100	96.0	97.0	67.0	93.0	98.0	99.0	94.0	99.0	94.64
Left-to-Right Swipe	100	98.0	97.0	100	93.0	87.0	99.0	100	98.0	98.0	99.0	97.18
Right-to-Left Swipe	91.0	99.0	99.0	31.0	95.0	100	94.0	100	98.0	99.0	96.0	91.09
Mean	98.5	98.8	99.3	84.6	95.8	85.0	95.5	92.3	93.8	97.0	96.3	94.26

The evaluation results are presented in Table 2 and 3. The evaluation metric is accuracy, which is the percentage of the correct gesture end detections. We consider the gesture end detection to be a failure if it is either early detection or no detection. An early detection is an event in which the gesture end predicted by the ADM happens before the marked ground truth gesture end. The no detection is an event in which the ADM does not predict a gesture end within 15 frames (i.e., approximately 600 ms) from the marked ground truth. From Table 2 and 3, we observe that the mean accuracy across the users and gestures for XGBoost is 5.3% better compared to the LSTM. Further, the worst user average accuracy is $> 97\%$ for XGBoost, whereas the worst user average accuracy for LSTM is just 84.6%. In addition, the gesture with the worst accuracy for XGBoost is the tangential circle of U9, with an accuracy of 87%. In comparison, the gesture with the worst accuracy for LSTM is the right-to-left swipe, with an accuracy of only 31%. Finally, the LSTM has $3\times$ higher run time compared to XGBoost. As such, we recommend the use of XGBoost based ADM.

F. GESTURE CLASSIFIER (GC)

To evaluate the GC, we perform LOOCV that can provide better assessment of the generalizability of the classifier. Specifically, in each fold, we hold one user out for testing and use the other 10 users for training. In this evaluation, we only use the samples that were correctly detected by the ADM. Since ADM accuracy is very high, for each fold, we have ~ 6000 samples for training and ~ 600 samples for testing. For training, we select the categorical cross-entropy as the loss function and use the Adam optimizer with a learning rate of 5×10^{-5}. We use the Glorot uniform initialization [36] (we also tried He initialization [37], but did not see any significant performance difference and thus results for He initialization are not reported here) and train for 150 epochs. An example of a typical training loss curve is shown in Fig. 14. We did not observe any irregular behavior for all the training conducted for this LOOCV. We employ ensemble method with 5 models using majority voting to stabilize the prediction output.

In the following, we will first report the leave-one-out performance for the network architecture described in Section IV-D. Then, we provide additional results to show the effect of the BlurPool layer [28], which supports our decision to include BlurPool in our CNN architecture.

Table 4 shows the LOOCV performance for 11 users and the corresponding confusion matrix is shown in Fig. 15. The samples used here are those detected by the ADM as described in Section IV-C with the counting threshold $N = 11$. The average accuracy for the 11 users is 95.5%. We observe some performance variations across the users, with a majority of users (10 out of 11) having $> 90\%$ accuracy. The relatively low accuracy of U2 and U5 could be attributed to the uniqueness in how those two users perform the gestures. This hypothesis is based on our observation that those confusion cases of a single pinch as a radial circle mainly come from U5 (31 out of the total confusion cases of 35 in Fig. 15), and all 41 confusion cases of right-to-left swipe as radial circle belong to U2. If this hypothesis is true, we expect the accuracy will improve if we have more users in the training set. The verification of this hypothesis is left for future work. For the average accuracy for each gesture, we see a similar level of accuracy for all gestures except for a single pinch. Based on our empirical analysis, we conjecture that this can be due to the weaker movement of the thumb in a single pinch gesture which weakens its signature and might make it prone to confusion with other gestures. This is because our radar processing solution is based on Doppler which may amplify the faster moving index finger and mask the contribution from the slower moving thumb in some single pinch gestures. The verification of this hypothesis is also left for future work.

We next report a comparison result to show the effect of the BlurPool layer [28]. As mentioned in Section IV-D, our reason to adopt BlurPool is that it can improve robustness against the shift of the input features. To verify this, we conduct the

TABLE 4. LOOCV performance on 11 users using the CNN architecture with BlurPool.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	Mean
Radial Circle	81.0	100	92.0	100	100	99.0	99.0	100	98.0	100	95.0	96.7
Tangential Circle	100	71.0	100	100	96.0	99.0	91.5	99.0	100	100	100	96.1
Double Pinch	100	99.0	70.7	98.0	90.0	100	97.0	96.0	100	100	98.0	95.4
Single Pinch	100	83.8	88.0	98.0	58.0	98.0	87.0	100	97.0	100	94.0	91.3
Left-to-Right Swipe	100	86.0	96.0	100	100	97.0	100	99.0	99.0	99.0	99.0	97.7
Right-to-Left Swipe	100	58.6	100	100	100	100	98.0	100	100	100	99.0	96.0
Mean	96.8	83.1	91.1	99.3	90.7	98.8	95.4	99.0	99.0	99.8	97.5	95.5

TABLE 5. End-to-end performance results of the proposed solution.

	U1	U2	U3	U4	U5	U6	U7	U8	U9	U10	U11	Mean
Radial Circle	81.0	100	92.0	100	100	99.0	98.0	100	98.0	99.0	95.0	96.5
Tangential Circle	100	71.0	100	100	96.0	99.0	79.6	99.0	100	100	100	94.9
Double Pinch	100	99.0	70.7	98.0	89.1	100	97.0	94.1	100	100	98.0	95.0
Single Pinch	100	83.8	88.0	98.0	58.0	98.0	86.1	99.0	97.0	100	94.0	91.1
Left-to-Right Swipe	100	81.7	96.0	100	100	97.0	97.0	99.0	99.0	99.0	99.0	97.0
Right-to-Left Swipe	100	58.0	100	100	100	100	98.0	100	100	98.0	99.0	95.7
Mean	96.8	82.3	91.1	99.3	90.5	98.8	92.5	98.5	99.0	99.3	97.5	95.0

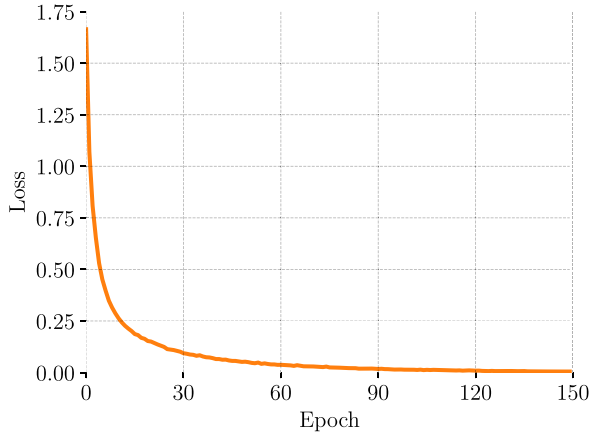


FIGURE 14. The training loss as a function of the number of epochs.

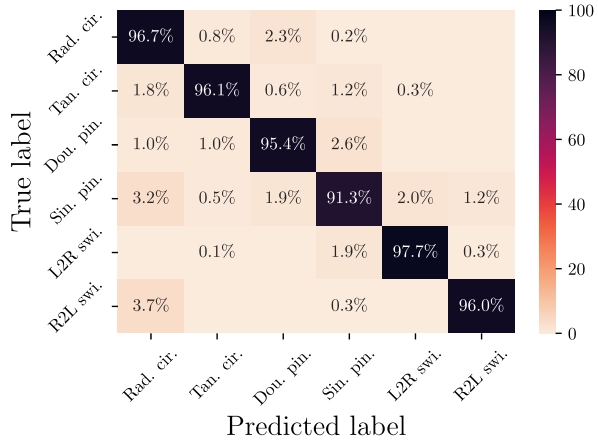


FIGURE 15. The confusion matrix of the LOOCV evaluation result of Table 4.

following experiment. We train two sets of models (still using the LOOCV setting as above): one using the architecture

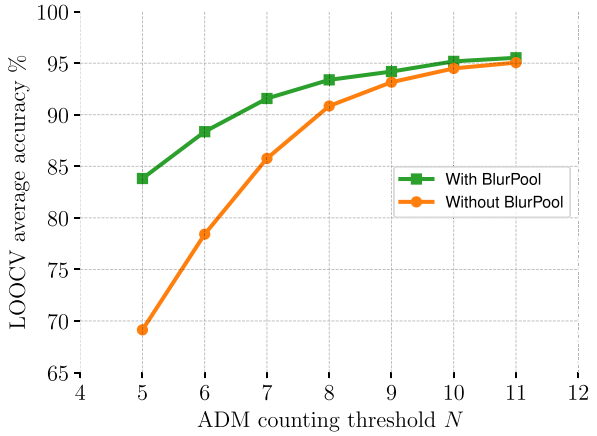


FIGURE 16. Comparison of the LOOCV average accuracy for the architecture with and without BlurPool when trained using data generated by ADM with $N = 11$ and testing with data samples generated with variable N.

with BlurPool as described in Section IV-D and the other one without BlurPool. We use the samples detected by ADM with a counting threshold $N = 11$ for training. For testing, we generate another set of samples using ADM with counting thresholds $N = \{5, 6, 7, 8, 9, 10, 11\}$. By setting different N between training and testing, there will be different linear shifts of the input features in the training and testing samples. The LOOCV average accuracy of the 11 users versus the choice of N in the testing is shown in Fig. 16. The plots show that there is some degradation as we perform the test on data generated with ADM using a different threshold N. As N becomes smaller and deviates more from $N = 11$ (for which the models are trained), the degradation increases. While this trend is true for both architectures, we can see clearly from the plots that the architecture with Blurpool has less degradation, and thus is more robust against variations in N. This is the main reason for our decision to adopt BlurPool in our GC architecture.

G. END-TO-END ACCURACY RESULTS

The end-to-end performance is reported in Table 5. The end-to-end accuracy is defined as the percentage of the test samples whose end is determined accurately by the ADM and the prediction by the GC is correct. Because the ADM has very high accuracy at 99.5%, the end-to-end accuracy is highly correlated with the accuracy results of the GC in Table 4. Our end-to-end accuracy is 95.0%.

VI. CONCLUSION AND FUTURE WORK

We presented an end-to-end solution for micro-gesture recognition using mmWave radar. We proposed 3 pairs of intuitive micro-gestures to be classified. Subsequently, we determined TVD and TAD to be meaningful features for gesture classification and confirmed that the proposed gestures are all distinguishable in TVD and/or TAD. We segmented the continuous stream of radar data using an ADM based on a binary classifier - that predicted whether the gesture has ended or not - and an accumulator - which gathered the predictions of the binary classifier. Subsequently, we classified segmented TVD/TADs containing gesture information using a CNN that includes a Blurpool layer to increase robustness against the shift in input features. The ADM and GC were both evaluated on data collected from 11 users. The end-to-end results show that the proposed solution can achieve 95% average accuracy.

More evaluations are required to establish the robustness of the proposed solution. Specifically, further generalization in terms of the number of users for training as well as the testing is needed. Also, we limited our evaluations to the 10 cm distance and the boresight of the radar. Extensions are required both to the distance and the angle relative to the radar. Though it is expected that the solution will be robust to small variations, e.g., distance variation of $2-5$cm, and angle variation of $[-10, 10]°$, extension for larger variance in distance and angles will likely require obtaining data and re-training the models. Further, the evaluations are made on a static radar, whereas the radar mounted on mobile devices is likely to move within the duration of the gesture, and evaluations catering to the mobile nature of the device are required. Finally, evaluating the computational complexity of the proposed solution, and the development of low complexity alternatives (e.g., similar in spirit to [17], [20], [23]) is another direction for future work.

REFERENCES

[1] K. Kalgaonkar and B. Raj, "One-handed gesture recognition using ultrasonic Doppler sonar," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Apr. 2009, pp. 1889–1892.

[2] R. Xie and J. Cao, "Accelerometer-based hand gesture recognition by neural network and similarity matching," *IEEE Sensors J.*, vol. 16, no. 1, pp. 4537–4545, Jun. 2016.

[3] W. Lu, Z. Tong, and J. Chu, "Dynamic hand gesture recognition with leap motion controller," *IEEE Signal Process. Lett.*, vol. 23, no. 9, pp. 1188–1192, Sep. 2016.

[4] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Short-range FMCW monopulse radar for hand-gesture sensing," in *Proc. IEEE Radar Conf. (RadarCon)*, May 2015, pp. 1491–1496.

[5] S. Ahmed, K. D. Kallu, S. Ahmed, and S. H. Cho, "Hand gestures recognition using radar sensors for human-computer-interaction: A review," *Remote Sens.*, vol. 13, no. 3, p. 527, Feb. 2021.

[6] J. Lien, N. Gillian, M. Karagozler, P. Amihood, C. Schwesig, E. Olsen, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–19, 2016.

[7] 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 *Proc. 29th Annu. Symp. User Interface Softw. Technol.*, Oct. 2016, pp. 851–860.

[8] 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 *Proc. Conf. Hum. Factors Comput. Syst.*, May 2021, pp. 1–14.

[9] S. Hazra, H. Feng, G. Naz 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," 2022, *arXiv:2203.17066*.

[10] J.-W. Choi, S.-J. Ryu, and J.-H. Kim, "Short-range radar based real-time hand gesture recognition using LSTM encoder," *IEEE Access*, vol. 7, pp. 33610–33618, 2019.

[11] Y. Sun, T. Fei, F. Schliep, and N. Pohl, "Gesture classification with handcrafted micro-Doppler features using a FMCW radar," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Apr. 2018, pp. 1–4.

[12] Z. Zhang, Z. Tian, and M. Zhou, "Latern: Dynamic continuous hand gesture recognition using FMCW radar sensor," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3278–3289, Feb. 2018.

[13] Y. Li, D. Zhang, J. Chen, J. Wan, D. Zhang, Y. Hu, Q. Sun, and Y. Chen, "Towards domain-independent and real-time gesture recognition using mmWave signal," 2021, *arXiv:2111.06195*.

[14] H. Liu, K. Cui, K. Hu, Y. Wang, A. Zhou, L. Liu, and H. Ma, "MTransSee: Enabling environment-independent mmWave sensing based gesture recognition via transfer learning," in *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2022, vol. 6, no. 1, pp. 1–28.

[15] Z. Zhang, Z. Tian, and M. Zhou, "SmartFinger: A finger-sensing system for mobile interaction via MIMO FMCW radar," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–5.

[16] M. Scherer, M. Magno, J. Erb, P. Mayer, M. Eggimann, and L. Benini, "TinyRadarNN: Combining spatial and temporal convolutional neural networks for embedded gesture recognition with short range radars," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10336–10346, Jul. 2021.

[17] T. Stadelmayer, A. Santra, R. Weigel, and F. Lurz, "Light-weight gesture sensing using FMCW radar time series data," 2021, *arXiv:2111.11219*.

[18] I. J. Tsang, F. Corradi, M. Sifalakis, W. Van Leekwijck, and S. Latré, "Radar-based hand gesture recognition using spiking neural networks," *Electronics*, vol. 10, no. 12, p. 1405, Jun. 2021.

[19] P. Zhao, C. Xiaoxuan Lu, B. Wang, N. Trigoni, and A. Markham, "CubeLearn: End-to-end learning for human motion recognition from raw mmWave radar signals," 2021, *arXiv:2111.03976*.

[20] M. Arsalan, A. Santra, and V. Issakov, "RadarSNN: A resource efficient gesture sensing system based on mm-wave radar," *IEEE Trans. Microw. Theory Techn.*, vol. 70, no. 4, pp. 2451–2461, Apr. 2022.

[21] A. Sluÿters, S. Lambot, and J. Vanderdonckt, "Hand gesture recognition for an off-the-shelf radar by electromagnetic modeling and inversion," in *Proc. 27th Int. Conf. Intell. User Interfaces*, Mar. 2022, pp. 506–522.

[22] G. Mauro, M. Chmurski, L. Servadei, M. Pegalajar-Cuellar, and D. P. Morales-Santos, "Few-shot user-definable radar-based hand gesture recognition at the edge," *IEEE Access*, vol. 10, pp. 29741–29759, 2022.

[23] M. Arsalan, A. Santra, and V. Issakov, "Spiking neural network-based radar gesture recognition system using raw ADC data," *IEEE Sensors Lett.*, vol. 6, no. 6, pp. 1–4, Jun. 2022.

[24] A. G. Stove, "Linear FMCW radar techniques," *IEE Proc. F Radar Signal Process.*, vol. 139, no. 5, pp. 343–350, Oct. 1992.

[25] V. Winkler, "Range Doppler detection for automotive FMCW radars," in *Proc. Eur. Microw. Conf.*, 2007, pp. 166–169.

[26] R. O. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 276–280, Mar. 1986.

[27] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, "A survey of the recent architectures of deep convolutional neural networks," *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, 2020.

[28] R. Zhang, "Making convolutional networks shift-invariant again," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7324–7334.

[29] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, Jun. 2014.
[31] F. Chollet. (2015). *Keras*. [Online]. Available: https://github.com/fchollet/keras
[32] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, and M. Isard, "TensorFlow: A system for large-scale machine learning," in *Proc. USENIX Symp. Oper. Syst. Design Implement. (OSDI)*, 2016, pp. 265–283.
[33] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.
[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[35] *XGBoost Parameters*. Accessed: Jul. 15, 2022. [Online]. Available: https://xgboost.readthedocs.io/en/stable/parameter.html
[36] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
[37] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

ANUM ALI (Senior Member, IEEE) received the B.S. degree in electrical engineering from COMSATS University Islamabad (CUI), Islamabad, Pakistan, in 2011, the M.S. degree in electrical engineering from the King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia, in 2014, and the Ph.D. degree in electrical engineering from The University of Texas at Austin, Austin, TX, USA, in 2019. From 2011 to 2012 and from 2014 to 2015, he held research positions with CUI and the King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, respectively. He has held visiting positions with the California Institute of Technology, Pasadena, CA, USA, in 2015; and Aalborg University, Aalborg, Denmark, in 2017. In 2016, 2017, and 2018, he held internship positions with Intel Corporation, Santa Clara, CA, USA; Qualcomm, Bridgewater, NJ, USA; and Nokia Bell Labs, Naperville, IL, USA. He is currently a Senior Engineer with Samsung Research America, Plano, TX, USA. His research interests include application of signal processing and machine learning tools to wireless communication and sensing systems.

PRIYABRATA PARIDA received the B.Tech. degree in electronics and communications engineering from the National Institute of Technology, Durgapur, India, in 2010, the M.S. degree in telecommunications from the Indian Institute of Technology, Kharagpur, India, in 2015, and the Ph.D. degree in electrical engineering from Virginia Tech, in 2021.

He is currently a Senior Engineer at Samsung Research America, Plano, TX, USA. In the past, he has held a full-time position at Ideal Cellular (now Vodafone Idea) Ltd., India, and internship positions at MediaTek Inc., USA, and Black Danube Systems Inc., USA. His research interest includes modeling and analysis of wireless communication networks using tools from stochastic geometry. His other research interests include the application of signal processing and machine learning techniques to wireless communication and sensing systems. He has been an Exemplary Reviewer of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, in 2017, and IEEE WIRELESS COMMUNICATIONS LETTERS, in 2019.

VUTHA VA (Member, IEEE) received the B.E. and M.E. degrees in electrical and electronic engineering from the Tokyo Institute of Technology, in 2011 and 2013, respectively, and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin, in 2018. He is currently a Staff Engineer at the Standards and Mobility Innovation Laboratory, Samsung Research America. His research interests include vehicular communications, millimeter wave communications, the fifth generation (5G) cellular networks, millimeter wave radar, and machine learning.

SAIFENG NI received the B.E. and M.S. degrees from the University of Science and Technology of China, in 2009 and 2012, respectively, and the Ph.D. degree from The University of Texas at Dallas, in 2018. Her research interests include several topics in computer graphics, computer vision, radar sensing, machine learning, and VR/AR, with emphasis on mesh optimization, 3D human body and face modeling, reconstruction animation, motion tracking, and radar-based motion sensing.

KHUONG NHAT NGUYEN (Member, IEEE) received the B.S. degree in computer engineering from The University of Texas at Arlington, TX, USA, in 2014, and the Ph.D. degree in computer science from Texas A&M University, College Station, TX, USA, in 2019. He is currently a Staff Research Engineer with the Standards and Mobility Innovation Laboratory, Samsung Research America, Plano, TX, USA. His research interests include artificial neural networks, reinforcement learning, general artificial intelligence involving cognitive science, and applied machine learning in telecommunication.

BOON LOONG NG (Member, IEEE) received the Bachelor of Engineering degree in electrical and electronic engineering and the Ph.D. degree in engineering from The University of Melbourne, Australia, in 2001 and 2007, respectively.

He is currently the Senior Research Director with the Standards and Mobility Innovation (SMI) Laboratory, Samsung Research America, Plano, TX, USA. He had contributed to 3GPP RAN L1/L2 standardizations of LTE, LTE-A, LTE-A Pro, and 5G NR technologies, from 2008 to 2018. He holds over 60 USPTO-granted patents on LTE/LTE-A/LTE-A Pro/5G. He has more than 100 patent applications globally. Since 2018, he has been leading a research and development team that develops system and algorithm design solutions for commercial 5G and Wi-Fi technologies.

JIANZHONG CHARLIE ZHANG received the Ph.D. degree from the University of Wisconsin, Madison. From 2009 to 2013, he has worked as the Vice Chairperson of the 3GPP RAN1 working group and led development of LTE and LTE-Advanced technologies, such as 3D channel modeling, UL-MIMO, CoMP, and carrier aggregation for TD-LTE. He is currently a SVP and the Head of the Standards and Mobility Innovation Laboratory, Samsung Research America, where he leads research, prototyping, and standards for 5G and future multimedia networks.