IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received February 1, 2019, accepted February 24, 2019, date of publication March 7, 2019, date of current version March 29, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2903586

Short-Range Radar Based Real-Time Hand Gesture Recognition Using LSTM Encoder

JAE-WOO CHOI, SI-JUNG RYU, AND JONG-HWAN KIM, (Fellow, IEEE)
School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea

Corresponding author: Jong-Hwan Kim (johkim@rit.kaist.ac.kr)

This work was supported by the National Research Foundation of Korea (NRF) Grant through the Korean Government (MSIT) under Grant NRF-2017R1A2A1A17069837.

ABSTRACT Due to the development of short-range radar with high-resolution, the radar sensor has a high potential to be used in real human-computer interaction (HCI) applications. The radar sensor has advantages over optical cameras in that it is unaffected by illumination and it is able to detect the objects in an occluded environment. This paper proposes a hand gesture recognition system for a real-time application of HCI using 60 GHz frequency-modulated continuous wave (FMCW) radar, Soli, developed by Google. The overall system includes signal processing part that generates range-Doppler map (RDM) sequences without clutter and machine learning part including a long short-term memory (LSTM) encoder to learn the temporal characteristics of the RDM sequences. A set of data is collected from 10 participants for the experiment. The proposed hand gesture recognition system successfully distinguishes 10 gestures with a high classification accuracy of 99.10%. It also recognizes the gestures of a new participant with an accuracy of 98.48%.

INDEX TERMS FMCW radar, gesture recognitio, LSTM encoder, machine learning, real-time interaction.

I. INTRODUCTION

In the past, radar had been widely used for long-range detection and surveillance of objects. However, since the last decade, there have been some studies on object detection using a short-range radar with high resolution such as ultra-wideband (UWB) radar [1], [2]. Unlike optical cameras, a radar sensor is not affected by illumination and has the ability to detect the objects even in an occluded condition. Therefore, radar can be used in a wide variety of applications, both in the outdoor and indoor environments. Furthermore, it can operate at lower power compared to optical cameras and does not need to be exposed to the outside of the device it is attached to because of the radar signal's transmissivity. Due to the property of human-computer interaction (HCI), being able to see through the blocking material, HCI devices can be designed more neatly. Furthermore, due to the recent advances in machine learning, there have been studies on obtaining meaningful knowledge or context from the raw radar signal.

Radars, however, are still regarded as suitable only for detecting the moving objects at a long range [3], [4], and there is little research on recognizing non-rigid objects such as human hands at a short range. Furthermore, very little research on applications such as gesture recognition has been conducted. Even though the radar is a useful sensor to apply machine learning techniques to, research in radar based on machine learning has not been done much. Recently, a short-range radar with high-resolution and low-power, which is called *Soli*, was developed for tracking and recognizing fine hand gestures [5], [6]. In the *Soli* project, the various features that can be obtained from radar signals were defined and feature-based gesture recognition was performed using the random forest classifier. Besides, convolutional neural network (CNN) was employed to classify the driver's hand gesture based on an optical camera, depth camera, and radar sensor [7]. The CNN was used to fuse data from the three sensors and resulted in improved accuracy under the varying lighting conditions. The CNN was also used for gesture recognition using micro-Doppler signatures and classification accuracy was 85.6% for 10 gestures [8]. Furthermore, there are some application studies using a short-range radar. Research on feature-based gesture recognition using 24GHz frequency-modulated continuous wave (FMCW) radar was conducted with classification accuracy of 88.57% for 7 gestures, and feature analysis was also performed [9]. RadarCat was developed for material and object recognition [10]. However, the described studies are less robust to the range/speed of the motions and shape of hands that vary from person to person. Also they are designed to classify a small

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

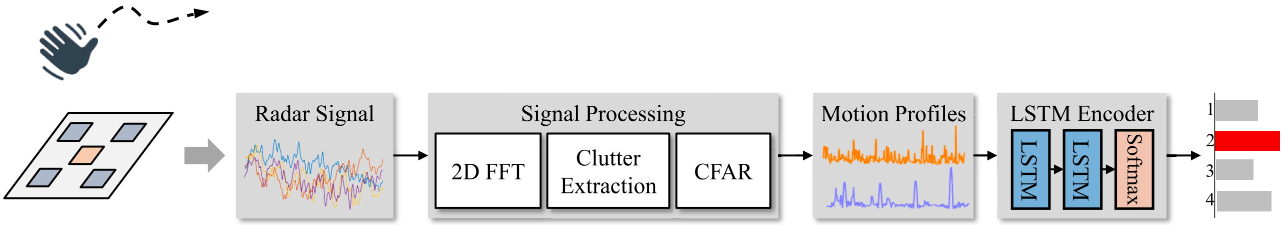


FIGURE 1. The overview of the proposed gesture recognition system.

number of gestures. Since they have an accuracy of about 90%, they are insufficient for practical applications.

This paper proposes a real-time gesture recognition system based on a short-range radar, which is shown in Fig. 1. The proposed system mainly consists of signal processing part and gesture recognition part. In the signal processing, we first employ the 2D fast Fourier transform (FFT) to generate range-Doppler map (RDM) where the distance and radial velocity of the reflected objects are expressed as two dimensions. Then, the clutters caused by reflection from the objects excluding the hand is extracted using background subtraction method. The hand gesture is detected by constant false alarm rate (CFAR), and the RDM sequences in the detected region is used as inputs to the hand gesture recognition part.

One important thing to consider when recognizing human gesture is real-time operation because it requires heavy computation during radar signal processing. For the purpose of real-time recognition, the sampling frequency of the Soli radar is set slightly lower, and several signal processing parameters such as the size of FFT are also set to a proper value so that they are capable of real-time processing. Such a lower hardware specification can decrease range and Doppler resolutions, but it is sufficient to recognize gestures by applying machine learning technique such as long short-term memory (LSTM).

After signal processing, we extract motion profiles from the RDM sequences for low computational complexity. The motion profiles represent the reflected energy distribution over distance and velocity. The proposed LSTM encoder receives the motion profile sequences and performs gesture recognition. The LSTM encoder is able to successfully extract the global temporal features of the data and recognize the hand gestures with high accuracy. Furthermore, practical experiments are conducted on various conditions, and a comparative analysis is performed.

The rest of this paper is organized as follows: the radar system for gesture recognition, signal processing, and post signal processing are briefly described in Section II. Section III proposes the gesture recognition algorithm based on LSTM encoder. The details of experimental settings are described in Section IV. The experimental results are discussed in Section V, and concluding remarks follow in Section VI.

II. RADAR SYSTEM FOR GESTURE RECOGNITION

A. SYSTEM OVERVIEW

An FMCW radar module used in gesture recognition is Soli module, developed by Google, which is only available to the developers [5], [11]. This radar module operates at 60 GHz frequency and has a cm-scale range resolution; in addition, it receives the signal through 4 patch antennas. The radar signals are transformed to a range-Doppler map (RDM) through signal processing procedure, and the resulting RDM sequences are fed into the machine learning algorithm as an input. Due to the combination of signal processing and machine learning algorithm for gesture recognition, the radar parameters governing the range and Doppler resolutions should be determined in consideration of several aspects. The resolutions of range and Doppler, Δr and Δv, are respectively represented as follows:

$$\Delta r = \frac{c}{2B} = 2.50\ cm, \tag{1}$$

$$\Delta v = \frac{c}{2f_c} \cdot \frac{1}{lT} = 122.07\ cm/s. \tag{2}$$

where c is the speed of light, and f_c is the center frequency of the radar which is set to 60 GHz. B and T are the bandwidth and the sweep period of the radar and are set to 6 GHz and 128 μs, respectively. l is the number of the chirps, set to 16.

Since 2D FFT with zero-padding is employed to obtain finer range and Doppler accuracy, more frequency bins are generated after the 2D FFT. Furthermore, we have a research goal in the gesture recognition using machine learning techniques, not in the accurate range and velocity measurements. Therefore, the radar system is not required to have too high range and Doppler resolution. Considering all these reasons, the range and Doppler bins, Δr_f and Δv_f, are respectively determined as follows:

$$\Delta r_f = \frac{c}{2B} \cdot \frac{f_s}{N/T} = 0.313\ cm, \tag{3}$$

$$\Delta v_f = \frac{c}{2f_c} \cdot \frac{1}{LT} = 7.629\ cm/s. \tag{4}$$

where f_s is the sampling frequency, set to 500 kHz. $N \times L$ is the size of the frequency bins for 2D FFT, which is set to $2^9 \times 2^8$ (512 × 256).

B. SIGNAL PROCESSING

Fig. 2 shows a waveform of the FMCW radar in the frequency domain. In the FMCW radar, the transmitted signal is frequency modulated by a periodic sawtooth wave function. There exist time delay τ and Doppler shift f_d between the received signal and the transmitted signal. The distance between the object and the radar causes the time delay, and

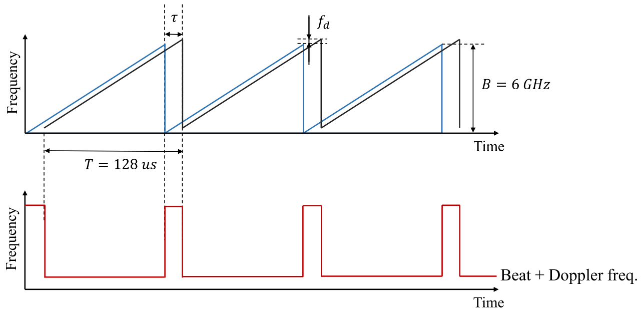


FIGURE 2. FMCW wavefrom in frequency domain. The beat and Doppler frequency are coupled, and the beat frequency is caused by time delay τ.

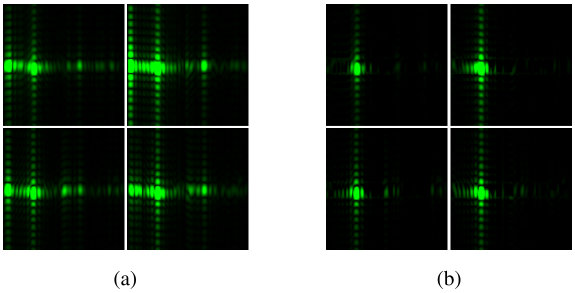


FIGURE 3. The clutter extraction is performed on all four receivers. The figures show the RDMs before and after the clutter extraction at the same frame. (a) RDMs before the clutter extraction. (b) RDMs after the clutter extraction.

the movement of the object receding or approaching the radar causes the Doppler shift [6]. In sawtooth wave modulation, the Doppler shift and beat frequency are coupled, as shown in Fig. 2. Accordingly, the Doppler shift and beat frequency are decoupled by 2D FFT as follows:

$$S(p,q,t) = \sum_{l=1}^{L}\left(\sum_{n=1}^{N} s(n,l,t)e^{-j2\pi pn/N}\right)e^{-j2\pi ql/L}$$
$$RD(r,v,t) = \left|S(\frac{r}{\Delta r_f}, \frac{v}{\Delta v_f}, t)\right| \tag{5}$$

where $S(p,q,t)$ is an output matrix at frame t in the frequency domain after 2D FFT, and $RD(r,v,t)$ represents range-Doppler map (RDM) converted from $S(p,q,t)$. The raw signal is first transformed into the form of a matrix $s(n,l,T)$ where each row of the matrix contains the beat signal of a single chirp. Next, the signal is transformed to the frequency domain, in which each axis respectively represents the range and Doppler, by applying a 2D discrete Fourier transform.

C. CLUTTER EXTRACTION

Before detecting gestures, clutters caused by reflection from other objects except the hand are extracted from raw RDMs. Assuming that all objects except the hand are almost static, the background subtraction method can be applied to extract the clutters. By generating an adaptive background model based on the Gaussian mixture model (GMM), the clusters that might change over time are effectively extracted [12], [13]. After then, the clutters are removed by calculating the difference between the current frame and the background model that contains the clutter of the radar signal. Fig. 3 shows the result of the clutter extraction.

D. GESTURE DETECTION

In radar signal processing, a constant false alarm rate (CFAR) algorithm is primarily used to detect objects. In this paper, the CFAR algorithm using signal difference between moving average and raw signal is proposed. We employ an exponential moving average (EMA), also known as an exponentially weighted moving average (EWMA) to calculate the moving average, as follow:

$$M_t = (1-\alpha)M_{t-1} + \alpha x_t, \tag{6}$$

where $\alpha \in [0, 1]$ represents a constant smoothing factor, and x_t represents the sum of all pixel values on the RDM of all four channels, which is defined as

$$x_t = \sum_i \|RD^i(r,v,t-1)\|, \tag{7}$$

where RD^i is the RDM matrix for i-th channel. The gesture detection occurs if the current raw signal exceeds the threshold, which is defined as

$$|x_t - M_t| > \theta \cdot (M_t + M_{\text{offset}}), \tag{8}$$

where θ is a detection threshold, and M_{offset} is an offset parameter. Fig. 4 shows the result of the gesture detection, especially in a sliding window over time. The cyan and red lines represent M_t and x_t, respectively. If the detection condition defined in (8) is satisfied, the detected intervals are displayed as the gold boxes. The RDM data contained in the detected interval is used as an input to the gesture recognition algorithm, and the remaining data is discarded.

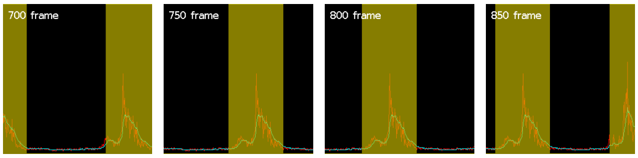


FIGURE 4. Gesture detection is performed based on M_t (cyan line) and x_t (red line) every frame. The gold boxes represent the detected interval in each of the 700th, 750th, 800th, and 850th frames from the left.

III. HAND GESTURE RECOGNITION

A. MOTION PROFILES

RDM sequences can be directly used as inputs to 3D-CNN or convolutional LSTM to extract spatial-temporal features of the hand gestures. These methods, however, require a large amount of computations, which is not suitable for real time applications. Therefore, we extract range and Doppler features, called range profile and Doppler profile, from RDM instead of using them directly. The range profile, RP_t^i, and

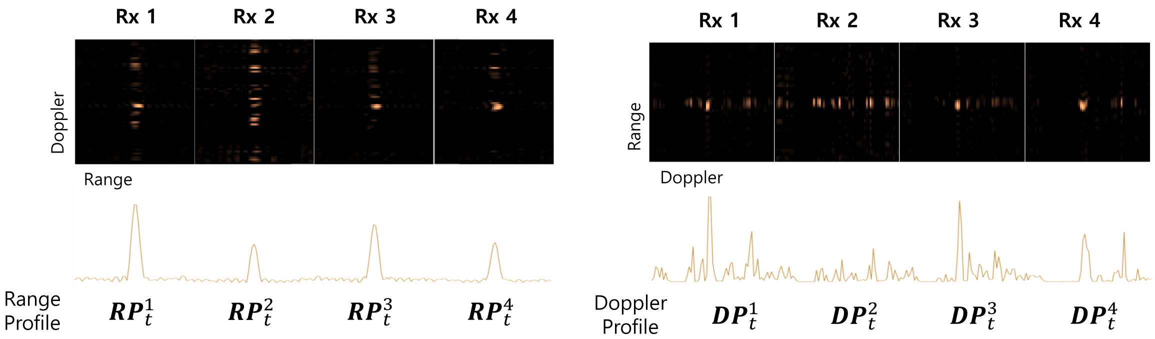


FIGURE 5. Range profiles and Doppler profiles of the four receivers of a radar sensor.

Doppler profile, DP_t^i, for i-th receiver at frame t are respectively defined as follows:

$$\mathbf{RP}_t^i = \sum_v RD^i(r, v, t), \tag{9}$$

$$\mathbf{DP}_t^i = \sum_r RD^i(r, v, t), \tag{10}$$

where $i = 1, \cdots, 4$, and $RD^i(r, v, t)$ is an RDM for i-th receiver at frame t. The range profile and Doppler profile represent the reflected energy distribution over distance and radial velocity, respectively. Fig. 5 shows range profiles and Doppler profiles of the four receivers of a radar sensor.

At each frame t, a motion profile, $\mathbf{MP}_t$, is created by concatenating the four range profiles and four Doppler profiles, as follows:

$$\mathbf{MP}_t = (\mathbf{RP}_t^1, \cdots, \mathbf{RP}_t^4, \mathbf{DP}_t^1, \cdots, \mathbf{DP}_t^4). \tag{11}$$

Finally, the obtained motion profile sequence, $\mathbf{MP}_1, \cdots, \mathbf{MP}_{T_k}$, is used as the input of the machine learning algorithm, where T_k is the length of the k-th gesture data instance.

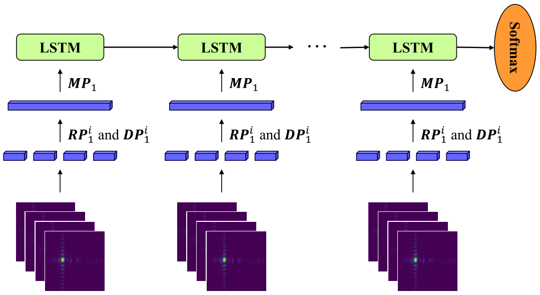


FIGURE 6. The proposed LSTM encoder architecture.

B. GESTURE RECOGNITION

We propose a neural network architecture as shown in Fig. 6 to recognize hand gestures in real time using motion profile sequences. The network consists of an LSTM encoder to extract the global temporal features of the hand gestures and a softmax layer to compute the conditional probabilities of the hand gestures. The details of the proposed network are described below.

First, we employ an LSTM encoder structure [14] to efficiently extract temporal features of the motion profile sequences. The lengths of the motion profile sequences are different depending on the classes of the gestures and the people performing the gestures. Furthermore, the LSTM encoder can map a motion profile sequence of various length to a fixed-dimensional vector representation, called an encoded vector $\mathbf{v}$. Therefore, the LSTM encoder structure is efficient to extract the global temporal features from the gestures having various lengths. In addition, the conventional RNN is a neural network that can effectively model sequential data [15], [16]. The LSTM is a special structure of the RNN, which can model the long term dependencies by alleviating the vanishing gradient problem of the RNN [17].

The LSTM encoder receives the motion profile sequence, $\mathbf{MP}_1, \cdots, \mathbf{MP}_{T_k}$, as input. The input gate, $\mathbf{i}_t$, the forget gate, $\mathbf{f}_t$, output gate, $\mathbf{o}_t$, memory cell, $\mathbf{c}_t$, and hidden state, $\mathbf{h}_t$, at each time step t can be obtained, respectively as follows:

$$\begin{cases} \mathbf{i}_t = \sigma(\mathbf{W}^{mi}\mathbf{MP}_t + \mathbf{W}^{hi}\mathbf{h}_{t-1} + \mathbf{W}^{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) \\ \mathbf{f}_t = \sigma(\mathbf{W}^{mf}\mathbf{MP}_t + \mathbf{W}^{hf}\mathbf{h}_{t-1} + \mathbf{W}^{cf}\mathbf{c}_{t-1} + \mathbf{b}_f) \\ \mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot tanh(\mathbf{W}^{mc}\mathbf{MP}_t + \mathbf{W}^{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \\ \mathbf{o}_t = \sigma(\mathbf{W}^{mo}\mathbf{MP}_t + \mathbf{W}^{ho}\mathbf{h}_{t–1} + \mathbf{W}^{co}\mathbf{c}_t + \mathbf{b}_o) \\ \mathbf{h}_t = \mathbf{o}_t tahn(\mathbf{c}_t) \end{cases} \tag{12}$$

where σ is the sigmoid function, $\odot$ is the element-wise product, and $t = 1, \cdots, T_k$. Once the motion profile sequence is all read, the hidden state becomes the encoded vector, $\mathbf{v} = \mathbf{h}_{T_k}$, and the encoded vector summarizes the whole sequence.

Finally, the encoded vector, $\mathbf{v}$, is connected to the softmax layer which converts the encoded vector to the class-conditional probability, $\mathbf{s}$, as follow:

$$\mathbf{s} = \mathbf{S}(\mathbf{W}_s\mathbf{v} + \mathbf{b}_s), \tag{13}$$

where each element, $[\mathbf{S}(\mathbf{x})]_i = e^{\mathbf{x}_i} / \sum_j e^{\mathbf{x}_j}$, represents the predicted probability of the i-th class.

In training, cross entropy is employed as a loss function. A ground truth label, $\mathbf{y}$, is represented as one-hot vector whose length is the number of classes, C. For a given training

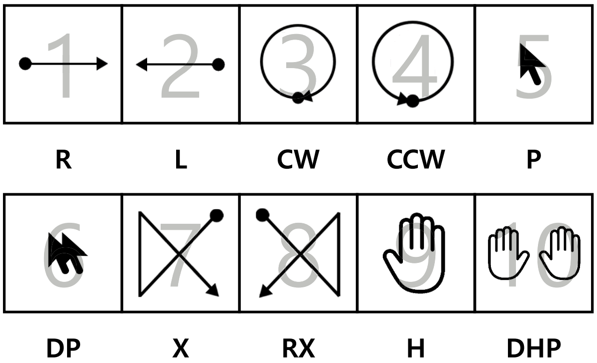


FIGURE 7. Ten hand gestures: (1) sliding left to right (2) sliding right to left (3) rotating clockwise (4) rotating counter-clockwise (5) push (6) double push (7) drawing X (8) drawing reverse-X (9) hold (10) double hand push.

example, the cross entropy loss between the ground truth label, $\mathbf{y}$, and the predicted probability, $\mathbf{s}$, is defined as

$$l = -\sum_{i=1}^{C} \mathbf{y}_i log(\mathbf{s}_i). \tag{14}$$

RMSProp is used as an optimizer [18]. The network parameter θ is updated at every back-propagation step t by

$$G_t = \gamma G_{t-1} + (1-\gamma)(\nabla_\theta L(\theta_t))^2, \tag{15}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \nabla_\theta L(\theta_t), \tag{16}$$

where γ is the momentum, η is the learning rate, and L is the averaged loss function over a mini-batch with respect to the parameter θ.

IV. EXPERIMENTS

A. EXPERIMENTAL SETUP

In this section, we describe the details of the 2D FFT, the network structure, and the training parameters used in the experiments. First, the output size of 2D FFT was set to $2^9 \times 2^8$ (512×256), and the RDM was generated every 25.0 *ms*. Second, we found the best structure of the LSTM encoder through repeated experiments. The LSTM encoder has one hidden layer which consists of 128 nodes. Finally, hyper-parameters used in the training were also determined through repeated experiments. The batch size, number of epochs, learning rate, and momentum were set to 10, 1000, 10^{-5}, and 0.9, respectively.

B. DATA SET

To validate the proposed system, we acquired a data set of the hand gestures consisting of 10 classes: (1) sliding left to right (2) sliding right to left (3) rotating clockwise (4) rotating counter-clockwise (5) push (6) double push (7) drawing X (8) drawing reverse-X (9) hold (10) double hand push. In the rest of the paper, each gesture is denoted in the order of R, L, CW, CCW, P, DP, X, RX, H, DHP. Fig. 7 shows the selected 10 hand gestures. The ten chosen gestures are designed to control or interact with devices. For examples, R is for changing channel, CCW is for rewinding video, and P is for pausing the devices. 10 participants performed the hand.

10 participants performed the hand gestures to collect data. The participants are composed of 8 men and 2 women between ages of 23 and 35 (mean: 30.1). All participants are right-handed and naturally performed the hand gestures using right hand. Each participant performed all 10 gestures 20 times for each gesture throughout 2 recorded sessions. Thus, a total of 4,000 hand gesture data were obtained.

In the first session, the supervisor showed each gesture twice as an example before collecting data. The participants watched the example gesture and repeated the same gesture 20 times. They sat on a chair and performed the gestures at a height ranging from 10.0 *cm* to 20.0 *cm* above the radar sensor attached to a desk. The speed of the gesture was freely performed by the participants at a similar pace to the example gesture. The second session was conducted on a different day from the first session to provide data diversity. In the second session, all participants immediately performed gestures without example gestures. Participants performed gestures in a slightly different way compared to those of the first session. Since the participants were instructed to take the gestures in their own way, the data was collected under the various conditions: the distance between the hand and the sensor, the speed and style of the gesture.

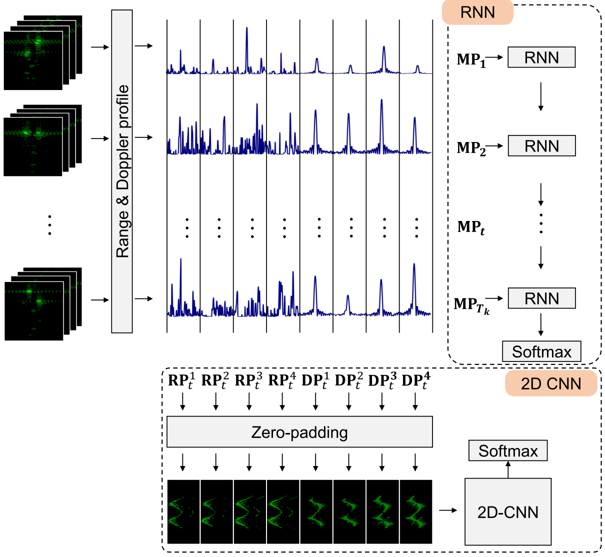


FIGURE 8. The data processing scheme and the network structures of the RNN encoder and the 2D CNN.

C. COMPARED ALGORITHM

We compared the proposed LSTM encoder with two machine learning algorithms, RNN encoder and 2D CNN. Fig. 8 depicted the data processing scheme and the network structures of two compared algorithms. First, the RNN encoder was used to measure the accuracy of recognizing hand gestures. Like the LSTM encoder, the RNN encoder has one hidden layer, and one layer has 128 nodes. In training, all hyper-parameters (batch size, number of epochs, learning rate, and momentum) were the same as those of the LSTM encoder.

Second, we used the 2D CNN similar to the previous study [8]. We processed the RDM sequences to use them as the inputs of 2D CNN as follows. Total eight matrices were constructed by the following: each range profile sequence from each receiver, $\mathbf{RP}_t^i$, is stacked on top of each other to compose a matrix. Consequently, four matrices are obtained from the range profile sequences. We also obtain four matrices from the Doppler profile sequence, $\mathbf{DP}_t^i$ by using the same method. Ultimately we combine the matrices of the range and the Doppler, making the total of eight matrices. However, since all inputs must have the same size in the 2D CNN, they were resized to 256-by-160 with zero padding. The eight matrices were used as the inputs of the 2D CNN. The 2D CNN consists of three layers, and each layer consists of a convolutional layer and a max pooling layer. The size of the convolutional filters is 5-by-5, and their numbers are set to 10, 4, and 2, respectively. The pooling size is 2-by-2. The last pooling layer is connected to the fully-connected layer with 128 nodes, and then to the softmax layer.

D. OFFLINE TEST

We measured the performance of the proposed system by using the 5-fold cross-validation with the data set collected from the 10 participants. The data set was divided into 5 sets, four of which were used for training and the rest for testing. Since the trained models showed slightly different performance depending on the initial weight, we obtained the average accuracy of 10 trained models for each fold.

E. NEW PARTICIPANT

We collected data from the 11th participant to test how well the learned model recognizes the hand gesture of a new participant whose data was not used for training and testing the networks in the offline test. The 11th participant is a 25 year old man. Data were collected in the same process as described in Section IV-B. All the collected data were used only for the test to measure the performance of the trained models.

F. ONLINE TEST

In order to process the radar signals sampled at 500 kHz gathered from four receivers in a real time, CUDA stream processing was applied to our system. We configured the system to process four 2D FFT operations in parallel by allocating one stream for each receiver. The overall architecture of the online gesture recognition system is depicted in Fig. 9. In the experiment, 2.6 GHz quadcore Intel i7-6700HQ laptop was used. One of 50 trained LSTM encoder models was randomly selected and used for online test.

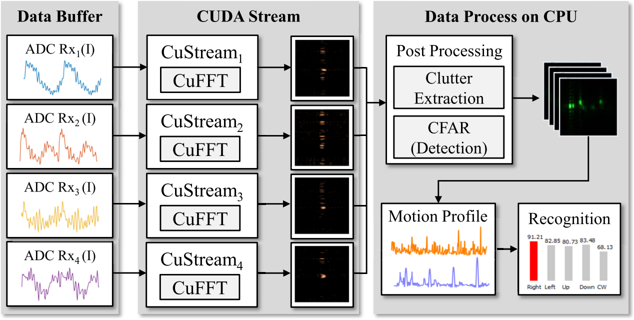


FIGURE 9. **The architecture of the real-time gesture recogntion system based on Soli .**

TABLE 1. **The accuracy per each fold and the average accuracy of three machine learning algorithms for the 10 participants.**

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg. (Std.)
2D CNN	93.51	95.93	96.54	95.49	95.48	95.39 (±1.01)
RNN encoder	90.41	93.90	94.48	93.35	92.39	92.90 (±1.42)
LSTM encoder	98.96	99.51	99.24	99.09	98.73	99.10 (±0.26)

V. RESULTS AND DISCUSSION

A. OFFLINE TEST

The LSTM encoder showed a high average recognition accuracy of 99.10% (std = 0.26) on 5-fold cross-validation. The 2D CNN and RNN encoder also showed high recognition accuracies of 95.39% (std = 1.01) and 92.90% (std = 1.42), respectively. Table 1 summarizes the 5-fold validation and the average accuracies.

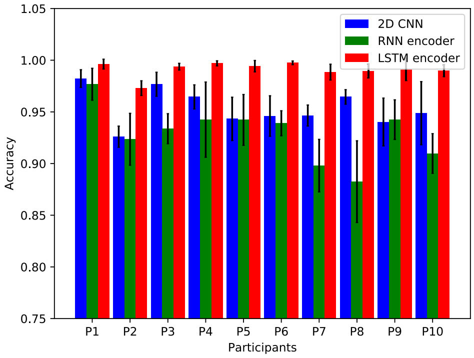
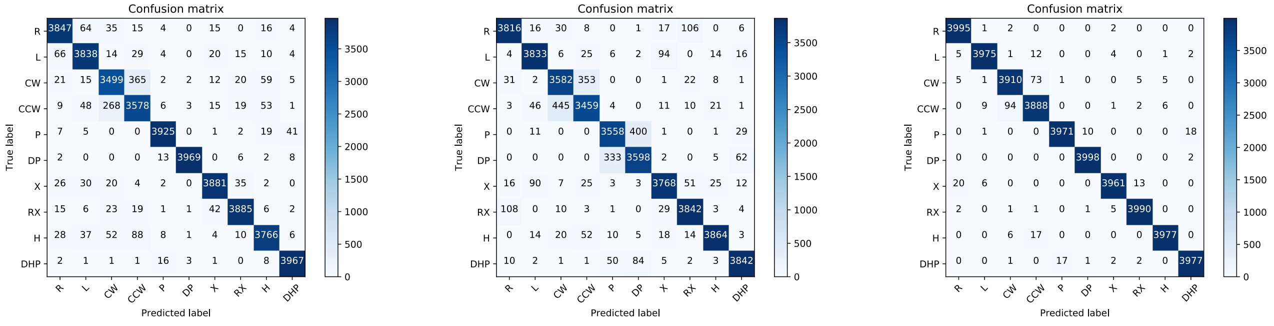


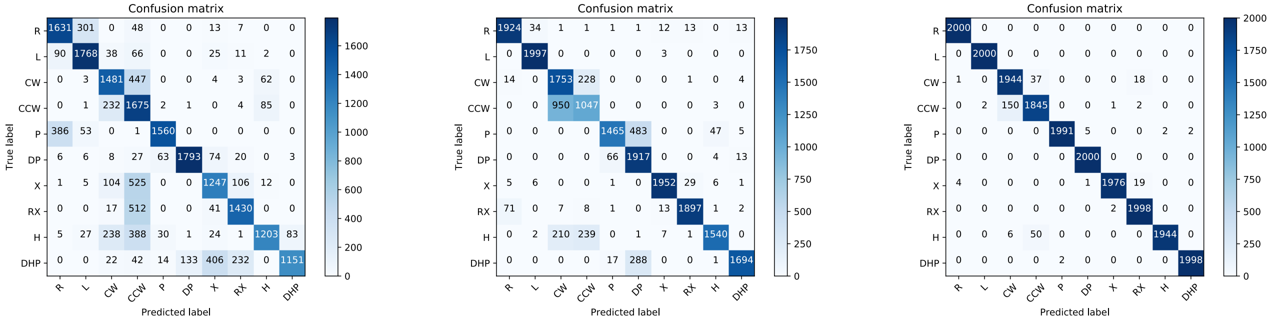
FIGURE 10. **The accuracy of the three machine learning algorithms for each participant.**

Fig. 10 illustrates the accuracy measured by using three machine learning algorithms for each participant. Since the speed and method of performing the gestures differs and the distance between the radar sensor and the hand is different for each participant, the accuracy of gesture recognition varies for each participant. The standard deviation of the accuracy for each participant was 1.67, 2.54, and 0.68 for 2D CNN, RNN encoder, and LSTM encoder, respectively. We found that the LSTM encoder performed not only with the best accuracy, but also with more robustness to the diverse gestures performed by the participants. Dealing with the diversity of human gestures in HCI is one of the important issues, and the proposed system successfully handles the problem with the LSTM encoder.



(a) (b) (c)

FIGURE 11. **The confusion matrices of three machine learning algorithms for the 10 participants. (a) 2D CNN. (b) RNN encoder. (c) LSTM encoder.**



(a) (b) (c)

FIGURE 12. **The confusion matrices of three machine learning algorithms for the 11th participant. (a) 2D CNN. (b) RNN encoder. (c) LSTM encoder.**

Confusion matrices for 10 participants are shown in the Fig. 11. The LSTM encoder classifies most of the gestures adequately other than the fact that it sometimes confuses the CW with CCW and vice versa. The 2D CNN also classifies gestures quite well except CW and CCW, however, the performance is lower than that of the LSTM encoder. The RNN encoder confuses not only CW and CCW, but also R and RX, L and X, P and DP. Since RNN cannot handle long term dependencies, the RNN encoder confuses the gestures that finish in a similar pattern.

TABLE 2. **The accuracy per each fold and the average accuracy of three machine learning algorithms for the new participant.**

	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Avg. (Std.)
2D CNN	70.95	75.30	78.08	74.58	74.58	74.70 (±1.01)
RNN encoder	84.93	87.15	85.75	86.03	85.80	85.93 (±1.42)
LSTM encoder	98.28	98.15	98.80	98.83	98.35	98.48 (±0.26)

B. NEW PARTICIPANT

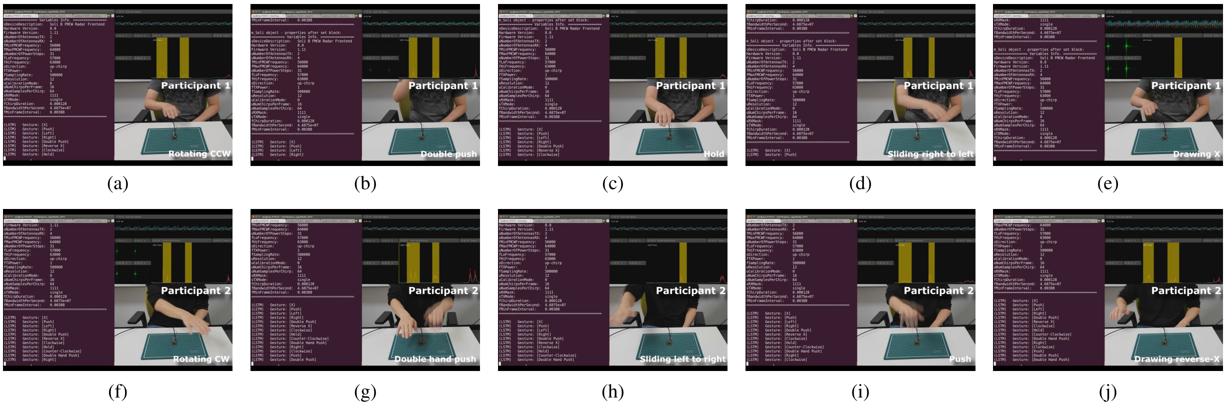
The results of 5-fold cross-validation for the new participant are summarized in Table 2. The LSTM encoder was able to maintain a high accuracy of 98.48% (std=0.28). However, the accuracies of the 2D CNN and RNN encoder dropped to 74.70% (std=2.27) and 85.93% (std=0.72), respectively. In particular, the accuracy of the 2D CNN decreased significantly. The 2D CNN performs zero padding to match the sequence length of all data. Since the sequence lengths of the gesture data for the new participant and for the ten participants are different, the accuracy is greatly reduced. On the other hand, the LSTM encoder is able to effectively extract features even with the variable-length input sequence, and thus it maintains the accuracy for the new participant.

Confusion matrices for the 11th participant are shown in Fig. 12. Like the offline test, the LSTM encoder confused CW and CCW the most but recognized the gestures correctly overall. 2D CNN fails to classify X, RX, H, and DHP in addition to CW and CCW. In the case of RNN encoder, the performance for H and DHP is decreased as compared to the offline test.

C. ONLINE TEST

It took about 7.7 *ms* to generate RDM in 4 receivers by performing 2D FFT in the CUDA stream and took less than 1.0 *ms* for post signal processing. It also took about less than 1.0 *ms* for forward-processing of the LSTM network. Considering that the radar signals, to be processed in CUDA stream, are received every 25.0 *ms*, it can be regarded as operating in real time. Although there is about 118.0 *ms* delay for TCP/IP communication between the recognition client and main server, the real-time gesture recognition is well performed through buffering and parallel computing.

Fig. 13 shows the real-time demonstration for two participants. They are both men and their ages are 25 and 31, respectively. They individually performed 10 gestures



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

FIGURE 13. Video captures of the on-line tests. Each row shows the randomly selected 5 gestures of the each participant. (a) Rotating counter-clockwise. (b) Double push. (c) Hold. (d) Sliding right to left. (e) Drawing X. (f) Rotating clockwise. (g) Double hand push. (h) Sliding left to right. (i) Push. (j) Drawing reverse-X.

randomly one by one. The system successfully recognized the 10 gestures for both participants in real time. [1]

VI. CONCLUSION AND FURTHER WORK

This paper proposed a real-time gesture recognition system using a short-range radar, Soli, developed by Google. We developed the gesture recognition system from the bottom-up including signal processing, machine learning, and communication. In the signal processing, 2D FFT was performed to generate the RDM sequences in real time, and clutters were removed using adaptive background model based on GMM. The gesture was detected by the CFAR algorithm, and then recognized by the LSTM encoder. The LSTM encoder extracted the global temporal features of the motion profile sequences. The motion profile sequences, the inputs to the LSTM encoder, were designed to reduce computational burden. As a result, the proposed system successfully performed with high accuracies under the various conditions.

As the further work, we are trying to improve the proposed algorithm to reduce false positive rate for the gestures that are not included in the training process. In addition, a radar sensor will be attached to a robot, and the proposed gesture recognition technology will be applied to HCI scenarios.

ACKNOWLEDGMENT

(Jae-Woo Choi and Si-Jung Ryu contributed equally to this work.)

REFERENCES

[1] A. Lazaro, D. Girbau, and R. Villarino, "Analysis of vital signs monitoring using an IR-UWB radar," *Prog. Electromagn. Res.*, vol. 100, pp. 265–284, 2010.

[2] B. Schleicher, I. Nasr, A. Trasser, and H. Schumacher, "IR-UWB radar demonstrator for ultra-fine movement detection and vital-sign monitoring," *IEEE Trans. Microw. Theory Techn.*, vol. 61, no. 5, pp. 2076–2085, May 2013.

[3] I. Bilik, J. Tabrikian, and A. Cohen, "GMM-based target classification for ground surveillance Doppler radar," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 1, pp. 267–278, Jan. 2006.

[4] A. Eryildirim and I. Onaran, "Pulse Doppler radar target recognition using a two-stage SVM procedure," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 2, pp. 1450–1457, Apr. 2011.

[5] J. Lien *et al.*, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Trans. Graph.*, vol. 35, no. 4, 2016, Art. no. 142.

[6] 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.

[7] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Multi-sensor system for driver's hand-gesture recognition," in *Proc. 11th IEEE Int. Conf. Workshops Autom. Face Gesture Recognit. (FG)*, vol. 1, May 2015, pp. 1–8.

[8] Y. Kim and B. Toomajian, "Hand gesture recognition using micro-Doppler signatures with convolutional neural network," *IEEE Access*, vol. 4, pp. 7125–7130, 2016.

[9] S.-J. Ryu, J.-S. Suh, S.-H. Baek, S. Hong, and J.-H. Kim, "Feature-based hand gesture recognition using an FMCW radar and its temporal feature analysis," *IEEE Sensors J.*, vol. 18, no. 18, pp. 7593–7602, Sep. 2018.

[10] H.-S. Yeo, G. Flamich, P. Schrempf, D. Harris-Birtill, and A. Quigley, "Radarcat: Radar categorization for input & interaction," in *Proc. 29th Annu. Symp. Interface Softw. Technol.*, 2016, pp. 833–841.

[11] (2015). *Project Soli*. [Online]. Available: http://atap.google.com/soli/

[12] Z. Zivkovic and F. van der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern Recognit. Lett.*, vol. 27, no. 7, pp. 773–780, 2006.

[13] Z. Zivkovic, "Improved adaptive Gaussian mixture model for background subtraction," in *Proc. 17th Int. Conf. Pattern Recognit.*, vol. 2, Aug. 2004, pp. 28–31.

[14] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[16] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural Netw. Mach. Learn.*, vol. 4, no. 2, pp. 26–31, 2012.

JAE-WOO CHOI received the B.S. degree in mathematical sciences and the M.S. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2016 and 2018, respectively, where he is currently pursuing the Ph.D. degree in electrical engineering. His current research interests include machine learning, human–robot interaction, and artificial intelligence for real applications.

[1] The video is available at http://bitly.kr/JWzwJ

SI-JUNG RYU received the B.S. degree in physics and the M.S. and Ph.D. degree in electrical engineering from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 2009, 2011, and 2018, respectively, where he has been a Research Assistant Professor, since 2015. His current research interests include machine learning, evolutionary computation, and artificial intelligence for real applications.

JONG-HWAN KIM (F'09) received the Ph.D. degree in electronics engineering from Seoul National University, Seoul, South Korea, in 1987. Since 1988, he has been with the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, where he is leading the Robot Intelligence Technology Laboratory as a Professor. He is currently the Dean of the College of Engineering, KAIST, and the Director of the KoYoung-KAIST AI Joint Research Center and the Machine Intelligence and Robotics Multi-Sponsored Research Platform. He has authored five books and five edited books, two journal special issues, and around 400 refereed papers in technical journals and conference proceedings. His research interests include intelligence technology, machine intelligence learning, ubiquitous and genetic robots, and humanoid robots. He served as an Associate Editor for the IEEE Transactions on Evolutionary Computation and the *IEEE Computational Intelligence Magazine*.