Detecting Mid-Air Gestures for Digit Writing With Radio Sensors and a CNN

Seong Kyu Leem, *Graduate Student Member, IEEE*, Faheem Khan, *Member, IEEE*, and Sung Ho Cho, *Member, IEEE*

***Abstract*— In this paper, we classify digits written in mid-air using hand gestures. Impulse radio ultrawideband (IR-UWB) radar sensors are used for data acquisition, with three radar sensors placed in a triangular geometry. Conventional radar-based gesture recognition methods use whole raw data matrices or a group of features for gesture classification using convolutional neural networks (CNNs) or other machine learning algorithms. However, if the training and testing data differ in distance, orientation, hand shape, hand size, or even gesture speed or the radar setup environment, these methods become less accurate. To develop a more robust gesture recognition method, we propose not using raw data for the CNN classifier, but instead employing the hand's mid-air trajectory for classification. The hand trajectory has a stereotypical shape for a given digit, regardless of the hand's orientation or speed, making its classification easy and robust. Our proposed method consists of three stages: signal preprocessing, hand motion localization, and tracking and transforming the trajectory data into an image to classify it using a CNN. Our proposed method outperforms conventional approaches because it is robust to changes in orientation, distance, and hand shape and size. Moreover, this method does not require building a huge training database of digits drawn by different users in different orientations; rather, we can use training databases already available in the image processing field. Overall, the proposed mid-air handwritten digit recognition system provides a user-friendly and accurate mid-air handwriting modality that does not place restrictions on users.**

***Index Terms*— Convolutional neural network (CNN), gesture recognition, human–computer interaction, image, impulse radio ultrawideband (IR-UWB) radar, localization, mid-air handwriting, sensor.**

I. Introduction

COMPARED to traditional touch-based interfaces, gesture-based interfaces can provide a more intuitive and convenient user experience. Camera-based gesture recognition has been widely studied and commercialized [1]–[3] but it is difficult to use in dark places and creates the possibility of leaking personally identifying information about its users. In many studies of gesture recognition, Zhu and Sheng [4] and Zhang and Harrison [5] attach sensors to the users' body; the drawback of such methods is that the user may feel uncomfortable wearing the related hardware.

Recently, research on gesture recognition using radar has been pursued as an alternative for overcoming the problems inherent in other methods. Radar operates in a noncontact manner, provides a comfortable user experience, presents no privacy issues, and is unaffected by light [6].

In previous studies, spectrograms obtained from reflected signals using Doppler radar were analyzed using a convolutional neural network (CNN) to recognize gestures [7]. In addition, spectrograms obtained using low-power frequency-modulated continuous wave (FMCW) radar have also been analyzed using a CNN [8]. Although these studies showed good accuracy at angles and distances at which the training was performed, accuracy deteriorated drastically at distances and angles at which training was not performed. In a study of gesture recognition using impulse radio signals, a training set including gestures at different angles was employed to solve the problem of reduced accuracy due to angle changes [9]. However, if gesture verification is performed at an angle not used for training or the shape of the hand is different, the accuracy of such methods may decrease, limiting these methods' real-world applications. Thus, existing studies that use raw data—such as spectrograms—in their training sets show lower accuracy at angles and distances that have not been trained. Moreover, accuracy is also reduced for users who are not part of the training set because the shapes of their hands are different. Despite the improvements achieved by the researchers in [9], the method reported is not completely reliable for real-world scenarios.

In this paper, we propose a method to recognize digits written by a hand moving through the air. Hand trajectories are obtained using multiple radar sensors, and the CNN is trained using handwritten numeric digit images already accumulated by image processing researchers. This paper can be differentiated from existing methods in several ways: First, a similar pattern can be obtained by using trajectory information, instead of raw data such as a spectrogram, even if direction, distance, and hand shape are different. Second, since the gesture is recognized by the CNN constructed using existing image data, new users need not undergo a separate training process. These two points make the proposed method more robust to distance and angle changes and have equal recognition accuracy even for new users. Conventional CNN methods that utilize raw

Manuscript received November 29, 2018; revised February 4, 2019; accepted March 13, 2019. Date of publication April 9, 2019; date of current version March 10, 2020. This work was supported by the Bio and Medical Technology Development Program (Next Generation Biotechnology) through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT and Future Planning under Grant 2017M3A9E2064626. The Associate Editor coordinating the review process was V. R. Singh. *(Seong Kyu Leem and Faheem Khan are co-first authors.) (Corresponding author: Sung Ho Cho.)*

The authors are with Department of Electronics and Computer Engineering, Hanyang University, Seoul 04763, South Korea (e-mail: dragon@hanyang.ac.kr).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIM.2019.2909249

data have lower accuracy when the distance, direction, or hand shape that must be recognized changes because raw radar data varies with changes in these parameters: even if the CNN is trained on all these parameters, a slight change of angle or environment will result in different patterns of raw data because of clutter and the multiple paths associated with a new setup. It is not possible to train all these parameters for each user with unique setups at different locations that have different clutter environments. Instead, we use three impulse radio ultrawideband (IR-UWB) radar sensors in an indoor environment to obtain trajectory patterns using the distance information from each sensor. However, there is clutter in indoor environments which can distort radar data. Average filtering is used to remove clutter. Also, since the radar cross section (RCS) of the hand is not uniform during the gesture, outliers exist in the distance information obtained from each sensor and in the trajectories obtained from this distance information. A median filter and a Kalman filter (KF) were applied to remove these outliers and smooth the trajectory data. An image transforming algorithm was applied to the trajectory matrix to obtain image data suitable for CNN input.

This paper makes the following major contributions. First, to the best of our knowledge, this study is the first to use radar to recognize digits from mid-air handwriting gestures. Although previous studies [7]–[9] have addressed general radar-based gesture recognition, none have focused on mid-air digit writing using radar sensors. Second, because the trajectory is used to recognize these gestures, the proposed method is more robust against distance and angle changes than those in previous studies [7]–[9]. Third, since the CNN is trained independently using an image data set, the user does not need to perform a separate training session using the radar; thus, recognition accuracy is user-independent. In this paper, we confirm the high recognition accuracy of using multiple IR-UWB radar sensors and the CNN, finding that the conventional method of using raw data and the CNN with IR-UWB radar results in good accuracy only if the training and testing take place at the same distance and with the same hand orientation. Then, we prove that our proposed trajectory-based method is more robust to changes in orientation, distance, speed, and user than existing methods.

II. Methodology

The radar sensors are set up to form a virtual plane, as shown in Fig. 1. The digits are written in mid-air in the plane of the three radar sensors. Signals from the radar sensors are reflected back and through the receiver antenna. Our proposed mid-air handwriting recognition problem can be divided into three stages: signal preprocessing, accurate positioning and localization, and image transformation and classification, as shown in Fig. 2.

The first stage involves preprocessing the raw data, including removing clutter from the received signal and finding the meaningful window that contains gesture data. An averaging filter is used to remove the background signal from the reflected signal. To differentiate the gesture interval from random motion or stationary periods, we use the magnitude

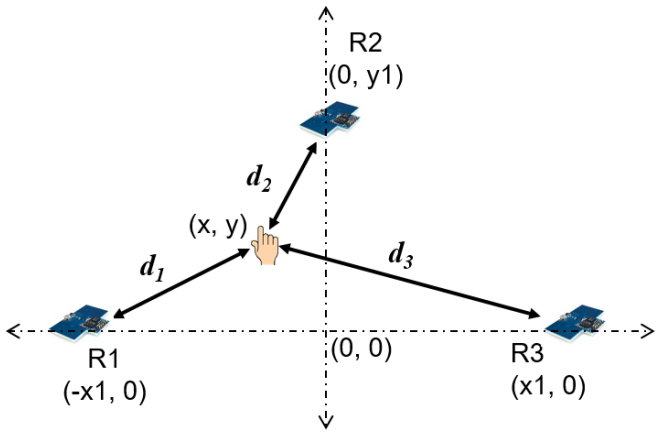

Fig. 1. Three radar sensors configured in a plane.

Fig. 2. Three stages of our proposed gesture recognition method.

and slow time duration of the reflected signals. A magnitude histogram is transformed into a log-normal function and the resulting spread σ is compared to a threshold value to detect any hand motion. If hand motion is detected for certain duration, then it is considered to be a gesture; otherwise, it is random motion.

The second stage is to localize and track hand motions during the mid-air gesture. The hand-to-radar distance is estimated using time of arrival (TOA) estimation. The deviation of each sample TOA from the mean value is used to detect outliers. Hand positioning is carried out through trilateration. Noise due to delays and TOA measurement is reduced using least-squares (LS) estimation. After getting the positioning data, we use a median filter for outlier rejection and KF estimation for hand tracking.

The third and final stage is the digit classification. We use a digit database that is used in image processing to train the CNN. First, the tracking data obtained in the second stage are transformed into an image. The image obtained is then resized and converted to grayscale so that it looks similar to the database images. The final images obtained are then classified using the CNN.

In Fig. 1, R_i shows the ith radar sensor and d_i represents the distance of the hand from radar i, where $i = 1, 2, 3$. The coordinates of the radar sensors are known and fixed; the coordinates of the hand (x, y) are determined using trilateration. The requisite signal processing and image classification techniques are detailed in Sections II-A–II-C.

A. Stage I: Preprocessing

1) Clutter Removal: The raw data received contains information about the moving hand's trajectory, as well as stationary clutter in the background. The raw signal is passed through a clutter removal filter to remove this unwanted signal. The background subtraction filter is explained in [10] in detail. Reflected raw signal $r_m(n)$ contains details of each object. The

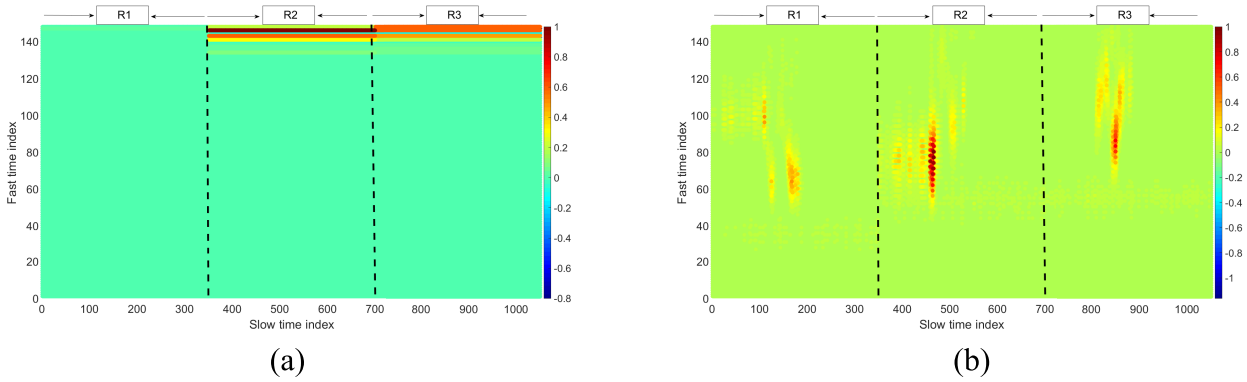

Fig. 3. Received radio signal matrix $W_{m\times n}$ (a) before and (b) after clutter removal.

unwanted echoes, called clutter, are removed using a simple loopback filter represented by the following equations:

$$c_m(n) = \alpha c_{m-1}(n) + (1-\alpha) r_m(n) \tag{1}$$

$$y_m(n) = r_m(n) - c_m(n) \tag{2}$$

where m is the slow-time index, n is the fast-time index, α is the estimated ratio of signal to clutter, $c_m(n)$ is the clutter signal, and $y_m(n)$ is the background-subtracted signal from which the clutter signal is removed. Here, α is the weighting constant that controls the sensitivity of the clutter removal process. We set this value to 0.8 for our experiments in order to pass the signal due to hand motion and subtract the signal reflected from the static background.

2) Meaningful Gesture Interval Determination: To find the meaningful slow time duration for evaluating a gesture, it is important to find the gesture's starting and finishing time. We use the spread of the log-normal distribution of the data obtained from the three radar sensors as a parameter for determining the hand motion in the plane of the radar sensors. The data received from the radar sensors are stored in a matrix, as shown in the following equation:

$$W_{m\times n} = \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & \ddots & \vdots \\ w_{m1} & \dots & w_{mn} \end{bmatrix}. \tag{3}$$

In (3), the matrix represents the combination of radar waveforms, as shown in Fig. 3. Element W_{mn} of the matrix represents background-subtracted signal $y_m(n)$, where m is the slow-time index and n is the fast-time index. The mth row of the matrix represents the background-subtracted signal received at the mth slow time. Fig. 3(a) clearly shows that the patterns in the raw signal are overshadowed by the heavy clutter signal; Fig. 3(b) shows the clutter-subtracted signal and has patterns from the three radar sensors due to the hand movements. For clarification, in Fig. 3, we plotted dotted black vertical lines to separate the data of the three radars.

In our case, we combine the data from the three radar sensors, resulting in a matrix of size $m\times 3n$. We then transform this data matrix into a magnitude histogram. Then, we apply log-normal fitting to the magnitude histogram, which returns the σ value [11]. The magnitude of σ can be used to determine if there is a meaningful hand motion gesture. As shown in Fig. 4, a large value of σ means that the received signal has a higher magnitude over a certain period, which means a meaningful hand gesture has occurred inside the plane of the three radar sensors. Algorithm 1 explains this method in detail. In Algorithm 1, we have used two thresholds: *Threshold1* and *movement_index_threshold*. *Threshold1* represents the level of the signal below which a signal value is considered to be noise. We have carefully chosen this level experimentally with hands at different distances and selected the minimum threshold value that can be created by the hand in the plane created by the radar triangle. Hand motions were repeated ten times each for a slow time window of 4 s; the average lowest level was set to *Threshold1*. The other threshold, *movement_index_threshold*, represents the overall magnitude of the data collected from the three radar sensors over the whole gesture interval. This value is the spread of σ and was also chosen by experimentation. Detailed explanations and accuracy information for different values of *movement_index_threshold* are given in Table II in Section III-B2. After the motion is detected using Algorithm 1, it is monitored for a duration equal to the minimum duration required for a gesture motion. We have chosen that minimum duration experimentally based on handwriting speed. If the motion is continuously detected for the minimum duration required for a gesture, then it is regarded as a meaningful gesture motion and the following steps in Sections II-B and II-C for digit classification are applied.

B. Stage II: Accurate Localization and Tracking

1) Ranging and Outlier Rejection: The distance between the hand and radar is measured using TOA estimation. After removing clutter from the received signal, the absolute of the signal is taken; then, the fast-time index n of the maximum value of the signal in each slow-time index m is measured, representing the location of the hand

$$R(m) = \arg\max_n \{W(m,n)\} \times r_{\text{step}}/2. \tag{6}$$

where $R(m)$ is the distance from the radar to the hand at the slow-time index m and r_{step} is the distance corresponding to one fast-time index of radar.

The reflected signal varies in magnitude with its location due to the RCS of the hand, antenna characteristics, and the

Algorithm 1 Meaningful Gesture Interval Detection Using Log-Normal Distribution

Procedure
Input:
Data matrix $W_{m\times n}$
Time step k
Threshold for meaningful motion detection: *movement_index_threshold*
Output:
Hand_motion_detected (true or false)
Initialize:
Assign $k = 1$;
$W_{m\times n} = W_{(k:m)\times(1:n)}$
Convert matrix $W_{m\times n}$ into a single vector W_l, where the size of l *is* $m \times n$.
Remove extremely small values to make the value of σ less dependent on small values in steps 13–17.
for $l = 1$: $m \times n$
if $(W_l < threshold1)$
Assign $W_l = [\,]$;
end if
end for
Find the magnitude histogram of vector W_l.
Fit the magnitude histogram to a log-normal (represented by X in (4)), as shown by the red line in Fig. 4.

$$X = e^{(\mu+\sigma W)} \tag{4}$$

where σ represents the *movement_index*.
Detect the meaningful hand motion duration by comparing σ with a predefined threshold value.
if (*movement_index_threshold*),
Hand_motion_detected = True;
end if
Update the motion sensing time window: $k = k + 10$; (5)
$W_{m\times n} = W_{(k:k+m-1)\times(1:n)}$
Go to step 11
end procedure

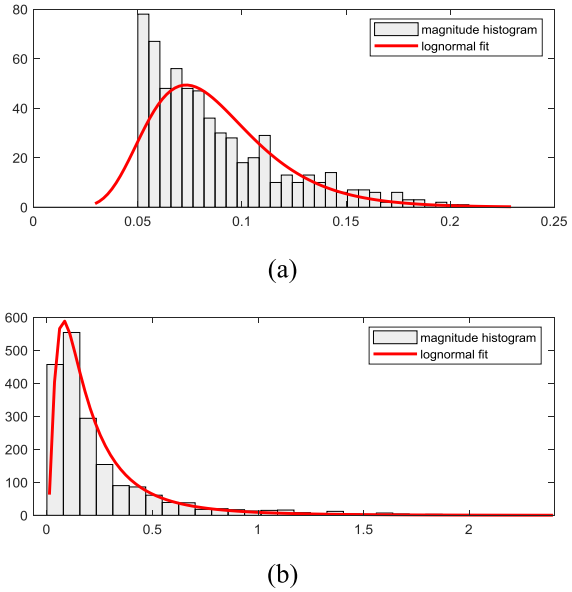

Fig. 4. Log-normal histogram fitting for the data set. (a) When there is no motion activity in the plane of the radar sensors and σ is 0.34. (b) When there is motion activity for a meaningful gesture and σ is 0.7.

influence of the multipath. In some cases, reflected waves due to surrounding objects or multipath, rather than the position of the hand, are detected with the largest magnitude. To eliminate this error, if the magnitude of the reflected signal is smaller than a certain threshold (set as 0.001 in this paper), the corresponding TOA is treated as "not a number" (NAN), and a 1-D median filter is applied to the TOA data to remove the outlier [12], [13]

$$d(m) = \text{med}\{R[m-K], \ldots, R[m-1], R[m], R[m+1], \ldots, R[m+K]\} \tag{7}$$

where $d(m)$ represents the value after applying the median filter. The window length of the median filter is $(2K + 1)$ slow time samples. The median filter is used to eliminate random impulse TOA errors. The window length of the median filter is determined by the repetitions of the TOA error and how long it occurs consecutively. This pattern of TOA error varies depending on the multipath characteristics caused by the clutter, the beam pattern of the antenna, and the RCS variation of the moving hand. The shape and speed of the hand also affect the TOA error pattern. In this paper, we have selected the most optimal window length $(2K + 1)$ through experiments on all cases from various palm and finger shapes of user hands, various orientations, and gesture speed changes. If the window length is too short, outliers generate errors. If we select too long a duration, the pattern is smoothed and errors occur. Therefore, considering all experimental cases, the window length that minimizes the error was experimentally

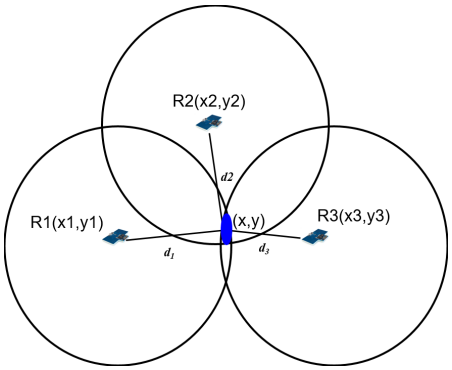

Fig. 5. Target localization by the trilateration.

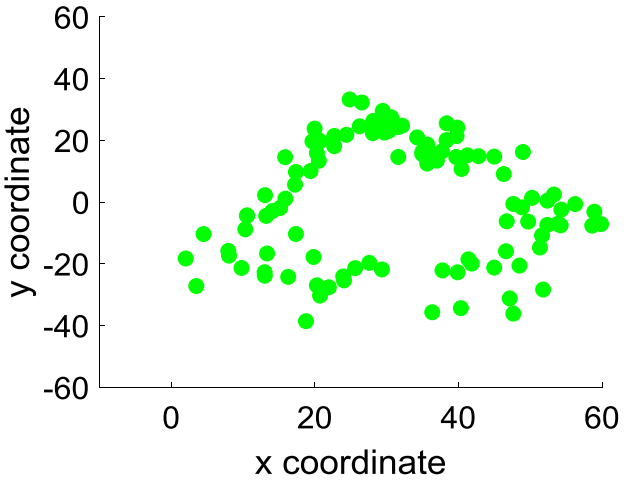

Fig. 6. Positioning result of the trilateration using LS for the digit "0".

determined by trial and error. The same window length was applied to all experiments. Even if some outliers remain, they are corrected by the KF at a later stage.

2) Trilateration Technique for Hand Localization: Since we have the distance of the hand from the three radar sensors and we also know the location of the radar sensors, we use trilateration for hand localization. The details of trilateration are given in [14].

As shown in Fig. 5, the target location is in the blue area that represents the cross section of the three distances between the target and the radar sensors. The relationship between the distances and the coordinates of the target and the radar sensors is given by the following equation. LS was applied to estimate the position of the hand [15]

$$d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad i = 1, \ldots, 3 \tag{8}$$

where d_i is the distance from the ith radar to the hand and (x_i, y_i) is the position of the radar. The position of the hand obtained through LS is $Z(x, y)$.

Fig. 6 shows positioning data for the digit 0 that contain many outlier values resulting from multipath signal reflection or reflection from moving clutter in the surrounding environment.

3) Outlier Rejection and KF for Tracking: After getting hand position information $Z(x, y)$ in the 2-D plane using trilateration, we must find a smoother estimate of the trajectory because the output of trilateration contains noise due to measurement delays and multipath signals. Furthermore, we have to eliminate outliers in the localization data. To this end, we have implemented a median filter and a KF [16], [17]. For outlier rejection, the median of recent observations and the current observation with a deviation above some threshold is discarded. The window size for the median filter is determined using the same technique as employed in Section I-B1. The equations for the KF are modeled as follows. The hand motion in the x and y coordinates can be represented by the following equations:

$$x(k) = x(k-1) + v_x * \Delta t + w \tag{9}$$

$$y(k) = y(k-1) + v_y * \Delta t + w. \tag{10}$$

In the above-mentioned equations, x and y are the hand motion coordinates, v_x and v_y are the corresponding velocities, w is the system noise, and Δt represents the time update step, which is calculated from the sampling frequency of the radar. The state space representation of the 2-D hand motion, when there is no input to the system, is given as follows:

$$X_k = AX_{k-1} + \varepsilon_{k-1} \tag{11}$$

$$Z_k = HX_k + \theta_k. \tag{12}$$

In the above-mentioned equations, the matrices can be defined as

$$X = [x_e \ y_e \ v_x \ v_y], \quad A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

and

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

In the above-mentioned equations, k represents the discrete time unit, X is the state vector, A is the state transition matrix, and H is the output matrix. $(x_0, \varepsilon_k, \theta_k)$ are the Gaussian, uncorrelated white noise sequences with mean $(\bar{x}_0, 0, 0)$ and covariance (P_0, Q_k, R_k), respectively. The initial velocity is set to zero, whereas the current velocity is calculated from the previous estimated samples. The estimated output position values through KF are (x_e, y_e), whereas the input measured values are $Z(x, y)$. Fig. 7 shows the sequence of the outlier rejection median filter and the classic KF for smoothing the data received from the localization step in Section I-B2.

The classic KF [16], [17] is applied which returns the estimated position and velocity as $X = [x_e \ y_e \ v_x \ v_y]$, the estimated hand position values, i.e., (x_e, y_e) at each time step k, are stored to obtain the hand motion trajectory. The trajectory after the KF is plotted in Fig. 8.

In Fig. 8, the tracking data is smoother than that shown in Fig. 6 because of the KF filtering, median-filter-based outlier detection techniques, and the proposed movement index method.

C. Stage III: Image Classification

1) Transformation of Tracking Data Into Images: Since we are using a large data set of images available in MATLAB,

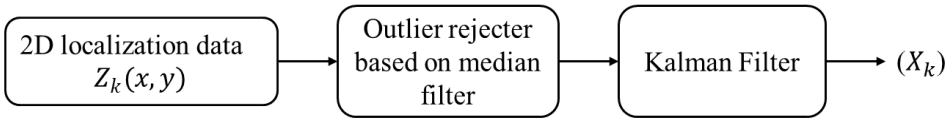

Fig. 7. Outlier detector and the KF block diagram for smoothing the trajectory.

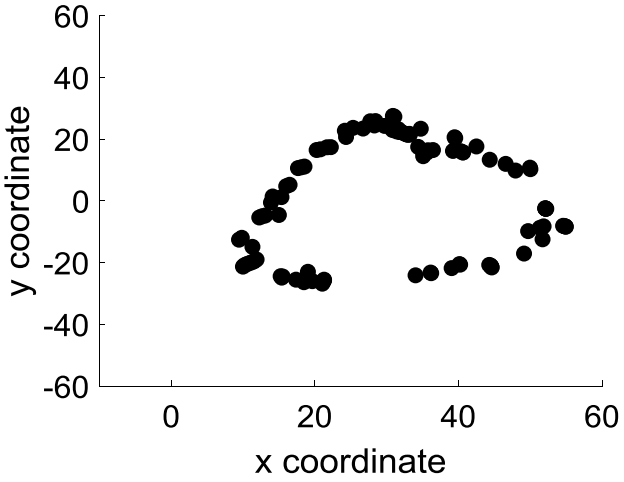

Fig. 8. Tracking result for the digit "0" after applying the KF with outlier rejection.

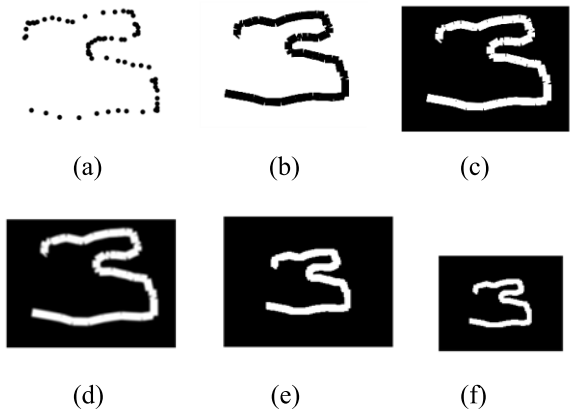

Fig. 9. Digit image after image transformation. (a) Raw tracking data. (b) After connecting with line. (c) After changing color. (d) After averaging filtering. (e) After applying (14). (f) After resizing.

we need to do some image processing before applying the CNN to those images. First, the points immediately adjacent to each other in time are connected by a line, as shown in Fig. 9(b). We change the background color to black and digit color to white, as shown in Fig. 9(c). To remove the image distortion caused by the discontinuity of the process of connecting the points, a 2-D averaging filter, as given in the following equation, is applied [18]:

$$g(x, y) = \frac{\sum_{s=-a}^{a} \sum_{t=-b}^{b} f(x+s, y+t)}{(2a+1) \times (2b+1)} \tag{13}$$

where $f(x, y)$ is the pixel value of the image before applying the averaging filter and $g(x, y)$ is the value after applying the filter. Where $(2a + 1)$ is the horizontal window length of the filter and $(2b + 1)$ is the vertical window length of the filter. Fig. 9(d) shows the result of applying the averaging filter.

After applying the averaging filtering, the conditional equation (14) is applied to sharpen the blurred image. If the intensity of the pixel is larger than a certain value, it is changed to 255 (which is the color value for white). The threshold used in this paper is 50. The results after applying the following equations are shown in Fig. 9(e).

$$\text{If } (g(x, y) > \text{threshold}), \quad \text{then } g(x, y) = 255. \tag{14}$$

Finally, the image is centered by calculating the center of mass of pixels and placing the image at the center point of the image matrix, and we reduce the image size to fit the CNNs input size 28×28, as shown in Fig. 9(f).

2) CNN for Image Classification: In this section, we show how images obtained through signal processing can be recognized as numbers using a classifier. Research on recognition of handwritten digits has been extensive in the computer vision field. Among the classifiers used, the CNN has had the best recognition accuracy. Therefore, the CNN was used as a classifier in this paper. The CNN consists of three main components: a convolutional filter, an activation function, and pooling. The convolutional filter extracts features by a convolution operation while sliding the whole image. Each filter acts as a feature detector and as many feature maps as the number of filters are generated. The size and number of filters are experimentally determined. The second component is the activation function. Common activation functions include sigmoid activation function, the tanh function, and restricted linear units (ReLU). ReLU, $f(x) = \max(0, x)$, has a faster convergence rate than the tanh and sigmoid activation functions because it does not activate all neurons at the same time. It also shows better results than other activation functions experimentally [19]. Because of these advantages, it is widely used in neural networks, including CNNs. Therefore, ReLU is used as the activation function in these experiments. The third CNN component is a pooling process that reduces the size of data by downsampling. Pooling methods include max, average, and min pooling. In CNN, max pooling is often used. The max pooling process reduces the size of the data, reducing the amount of computation required and making the network more robust to noise. Thus, it is adopted here.

The structure and a description of the CNN used are shown in Fig. 10. The CNN consists of four convolutional layers and three max pooling layers. The classification layer consists of a fully connected layer with ten outputs and a softmax function. We have optimized the layers of the CNN for our radar image classification problem using MATLAB functions [20]. The number of CNN convolutional layers is an important parameter that determines the performance and complexity of the entire CNN. In this paper, we first experimented with two layers

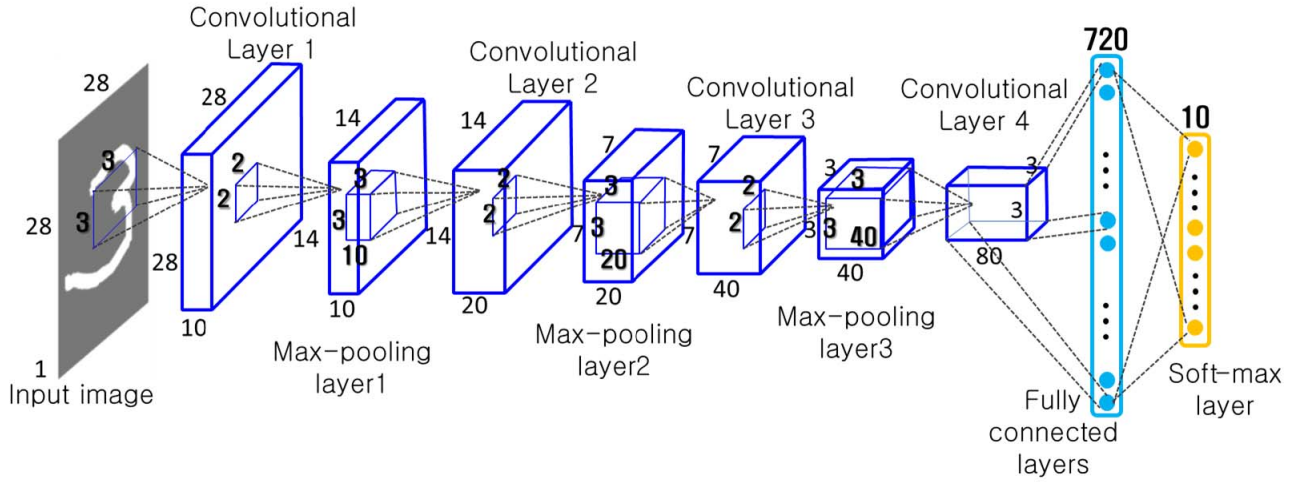

Fig. 10. Structure of the CNN for the proposed method.

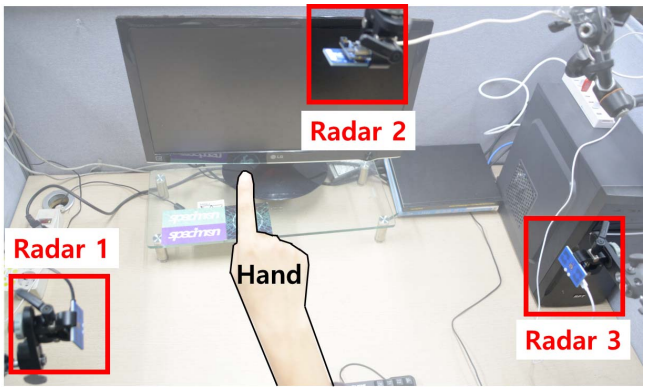

Fig. 11. Experimental setup.

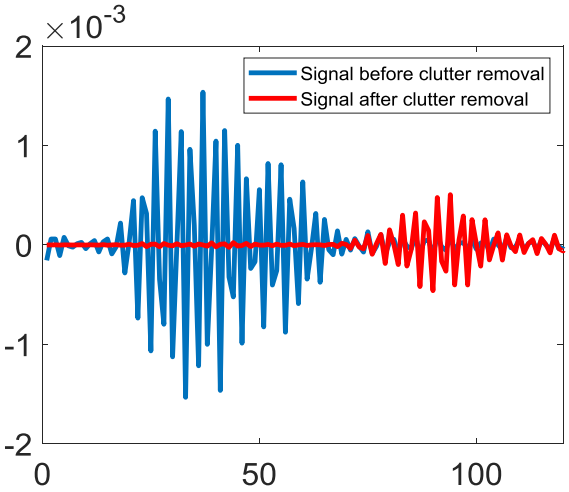

Fig. 12. Reflected waveform before and after clutter removal.

and increased the number of layers until satisfactory accuracy was obtained. We used four convolutional layers, in this paper, because recognition accuracy was 99.7% or greater when using four layers and was similar even when using five layers. We also optimized parameters such as convolutional filter size, pooling size, and pooling stride by repeated trial and error.

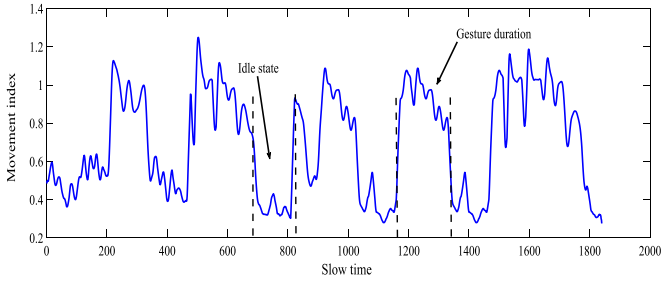

Fig. 13. Motion index plot to differentiate between meaningful gesture durations and idle time.

The CNN was trained for a large, already available data set of digits written in different orientations and styles. However, we did not use the entire Modified National Institute of Standards and Technology (MNIST) database, which is mainly used for handwriting recognition. The images produced in this paper have almost the same line thickness and are written with a fixed intensity by (14). MNIST data, on the other hand, includes all cases—from bold to blurred characters—in consideration of actual handwriting. It also contains nonnumeric points and noise lines that can be eliminated by the outliner rejection algorithm. For this reason, we constructed a data set based on the characteristics of the images created using radar signals without using the entire MNIST data. Using 1000 images for each number, we trained the CNN with a total of 10 000 images. In the vision research field, there is a highly optimized CNN that uses MNIST data; however, in this paper, we use a simpler CNN structure that provides satisfactory accuracy using a small data set containing images similar to those made using radar sensors.

III. Results and Discussion

In this section, we describe the experimental environment, including the radio sensor placement and the radio sensor used, and describe the experimental results of preprocessing techniques such as clutter removal and gesture interval recognition in an indoor environment. We also show the results of trajectory extraction and image transformation algorithms.

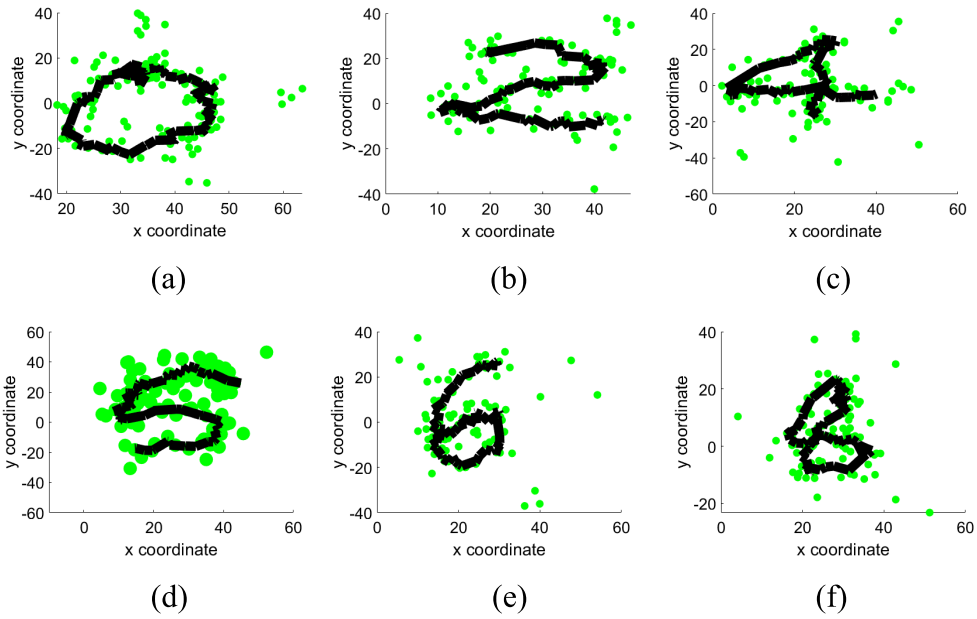

Fig. 14. Green dots show the result of LS estimation before KF; the black line shows the result after KF for (a) 0, (b) 2, (c) 4, (d) 5, (e) 6, and (f) 8 after image transformation.

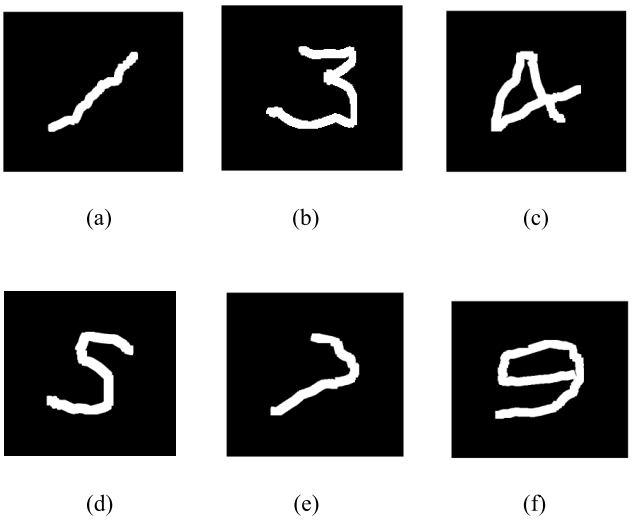

Fig. 15. Images showing (a) right tilted 1, (b) normal 3, (c) left tilted 4, (d) right tilted 5, (e) right tilted 7, and (f) normal 9 after image transformation.

Finally, the accuracy of each gesture after applying the CNN is shown in the confusion matrix form. The results of the conventional CNN technique, which uses raw data for training are shown, and accuracy results of the proposed method are also described, ultimately demonstrating that the proposed method is more robust to changes in orientation and hand movement speed.

A. Experimental Setup

1) Hardware Setup: We used three IR-UWB radar sensors for data acquisition. The data were taken in real time using the setup shown in Fig. 11. The distance between two radar sensors on the horizontal axis was 71.3 cm; the third radar sensor was installed at the midpoint between the other two at a height of 38.4 cm. The number of sensors was selected considering the accuracy of hand position estimation. When the hand moves, the received signal of the specific radar becomes instantaneously smaller or distorted due to instantaneous decrease in the RCS and the multipath generated by the clutter of the surrounding environment. This distortion of the received signal causes an error in position estimation, which is a factor that degrades recognition accuracy. To eliminate these errors, we applied median and KFs, as described in Section II-B. However, when two sensors were used, satisfactory results were not obtained. We used three sensors to increase the diversity effect and we were able to obtain satisfactory results under the given experimental conditions. We used commercially available radar X4 sensors (Novelda, Oslo, Norway). The specifications of the radar sensors are detailed in Table I.

The center frequency of the radar is 7.29 GHz and its bandwidth is 1.4 GHz. Range resolution is generally inversely proportional to bandwidth and is 6.4 mm for the sensor used in this experiment. This range resolution is much smaller than the gesture motion range of tens of centimeters; therefore it has little effect on this experiment. To recognize fine finger movements, it would be necessary to use a frequency of several tens of gigahertz. However, for the relatively large motion in the present experiment, the frequency used is appropriate.

2) Experimental Design: Several experiments were conducted to show that the proposed method is more robust than existing methods. In the experimental design, we considered mid-air writing at three different hand orientations—straight, left-tilted, and right-tilted—to prove that the proposed algorithm works even when the orientation of the hand changes. We also used multiple volunteers to prove that the proposed method is robust to differences in human hand shape and size.

The user was located 70 cm from the *xy* plane in the *z*-axis direction and extended the hand to the sensing area, writing the numbers using hand movements. First, to show the robustness

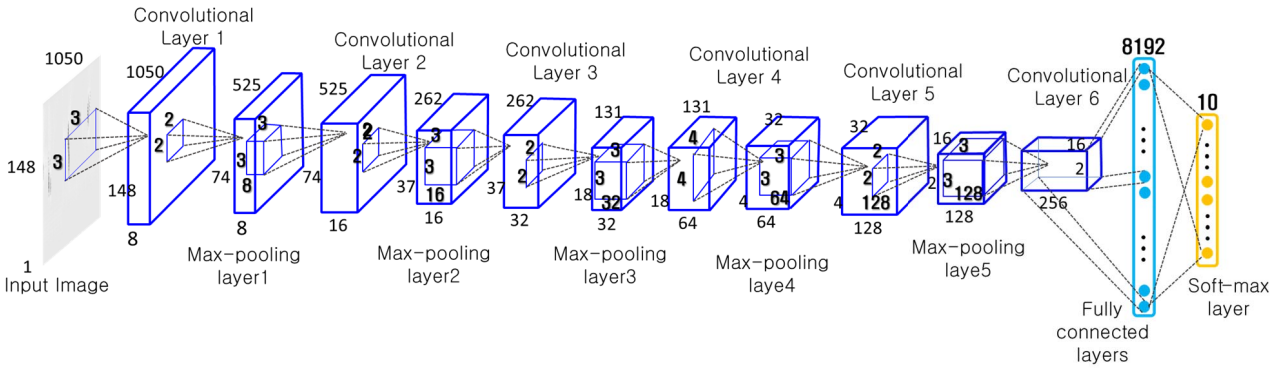

Fig. 16. Structure of the CNN for the conventional method.

TABLE I
RADAR SENSOR SPECIFICATION

Specification	Value
Peak pulse output power	4.1 dBm
Center frequency	7.29 GHz
Pulse repetition frequency	40.5 MHz
Bandwidth (-10dB)	1.4 GHz
Slow time sampling frequency	20 samples/s
Antenna beam width	65°

of orientation change, experiments were carried out when the numbers were written with a clockwise or counterclockwise tilt. To accomplish this, one user wrote a number 250 times each with counterclockwise and clockwise tilts at an angle of 10°. To confirm the method's accuracy when the distance from the radar sensor to the hand was changed, the experiment was carried out when digits were shifted in the positive or negative direction of the x-axis. In this experiment, one user moved the center of the number about 10 cm in the positive x-axis direction from the center of the xy plane, wrote the number 250 times, and moved the center of the number about 10 cm in the negative x-axis direction to write the number 250 more times. Raw radar data change with gesture speed. Therefore, to confirm robustness to changes in gesture speed, gesture accuracy was checked by changing speed. In this experiment, one user wrote a number 500 times at normal speed (7.52 cm/s) and 500 times at a speed of 13.5 cm/s which is almost two times faster. Finally, we confirmed the change of recognition accuracy for different users. To do this, five users wrote numbers in the air in the same way. The five users wrote a number 500 times with a vertical orientation at normal speed at the center of the xy plane. Using the data obtained from these experiments, we checked the accuracy of the existing and proposed methods. By comparing the accuracy obtained by each method, we confirmed how robust the proposed method is compared with existing methods.

B. Preprocessing

1) Clutter Removal: In Fig. 12, we have shown an example of a radar waveform before and after the clutter removal as discussed (Section II-A.1). Because of the objects in the indoor environment, the magnitude of the signal reflected by clutter at fast-time index 18–65 before the clutter removal is very large. As a result, the movement of the hand at fast-time index 78–105 is not detected. After clutter removal, the magnitude of the signal in the clutter area is greatly reduced and the signal reflected by the hand is clearly visible.

2) Meaningful Gesture Interval: Meaningful interval separation was discussed in Algorithm 1 in Section II. The movement index for the whole slow time is plotted in Fig. 13.

Fig. 13 shows the graph of the *movement_index* values for the entire duration, including gesture and nongesture time. The threshold value (0.6) was selected by trial and error. Fig. 13 clearly shows that the movement index was above the threshold during gestures and was below the threshold in nongesture and idle periods. Therefore, this method can be used to estimate the gesture interval. When the detection accuracy was experimentally measured for different values of movement index values, then it gave different results. The detection accuracy for finding the meaningful gesture duration is shown in Table II, which demonstrates that the movement index threshold should be carefully selected. Decreasing the value of this threshold allows noise to be included as gestures.

TABLE II
MEANINGFUL GESTURE DETECTION ACCURACY

movement index value	Meaningful gesture detection accuracy
0.3	38.7%
0.35	49.4%
0.47	73.5%
0.50	86.3%
0.55	98.4%
0.57	100.0%
0.60	100.0%
0.63	100.0%
0.87	79.7%
1.05	20.1%

TABLE III
RMSE FOR TRACKING DATA BEFORE AND AFTER APPLYING KF

RMSE (sample units) before outlier rejection and KF (x axis)	RMSE (sample units) before outlier rejection and KF (y axis)	RMSE (sample units) after outlier rejection and KF (x axis)	RMSE (sample units) after outlier rejection and KF (y axis)
9.72	11.44	3.58	4.27

In contrast, if the movement index threshold is very high, then some parts of gestures will be discarded and considered to be noise. The optimal values, however, give the correct meaningful gesture duration.

C. Tracking Results

The tracking results for digits (0, 2, 4, 5, 6, and 8) with the median and KF applied are shown in Fig. 14.

The average root-mean-square error (RMSE) values of the error before and after the KF are shown in Table III.

D. Image Transformation

The 2-D matrix of the tracked hand position is converted into an image suitable for CNN input. The results of applying the image transformation algorithm described in Section II-C1 are shown as follows. As shown in Fig. 15, an image similar to that written on paper with a pen is generated after applying the image transformation algorithm. Even if the number was written with a rightward or leftward tilt, an accurate undistorted image was produced.

E. CNN Accuracy Using Raw Data for Radar Gesture Recognition

1) Conventional CNNs Using Raw Data for Radar Gesture Recognition: The conventional CNN technique, which uses raw data for training, can briefly be described as follows.

1) The signals received from the radar sensor are passed through clutter removal for background subtraction.
2) The background-subtracted signals are then combined into a matrix.
3) The matrix that contains the signals from the three radar sensors is then converted into an image.
4) Using these images, the CNN is trained and evaluates gestures in real time.

The structure of the CNN is shown in Fig. 16. Six convolutional layers and five max pooling layers were used. The classification layer consisted of a fully connected layer with ten outputs and a softmax function. ReLU was used for the activation function and major parameters such as convolutional filter size, pooling size, and pooling stride were experimentally optimized. Fig. 17 shows some of the images created for some digits by combining the data from the three radar sensors.

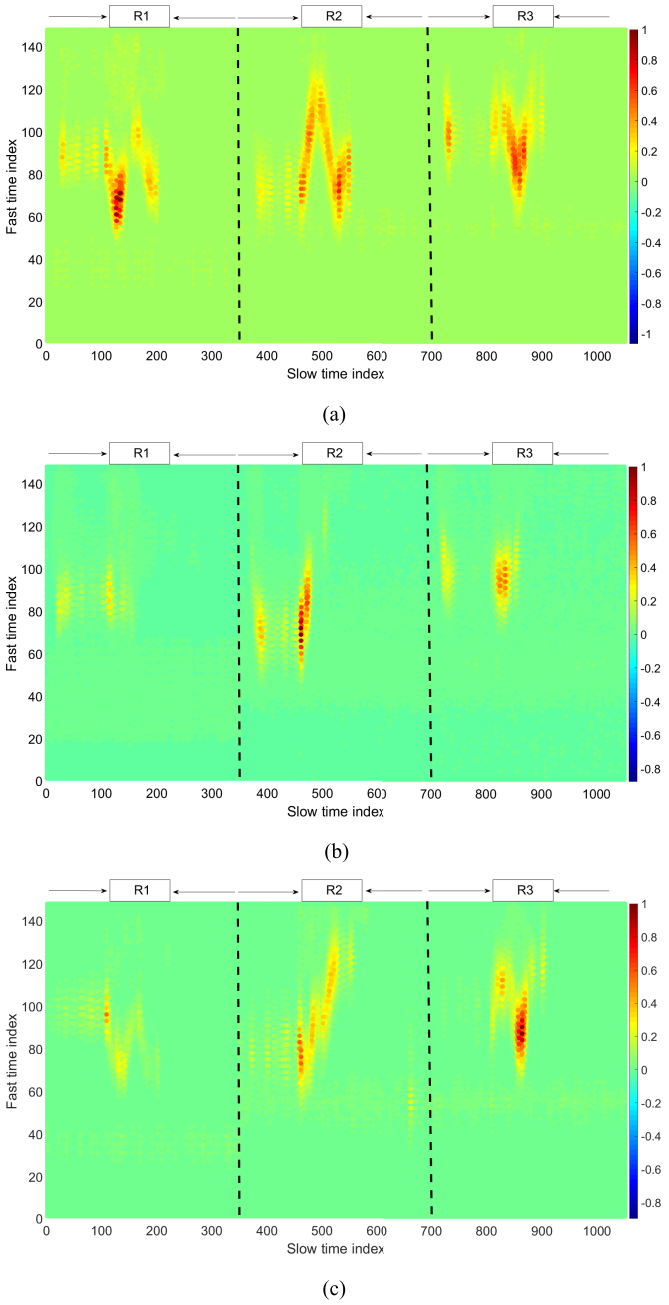

Fig. 17. Colored images of radar data referring to digits (a) 0, (b) 1, and (c) 5 after image transformation.

Fig. 17(a)–(c) shows the data images for digits 0, 1, and 5, respectively. Each image consists of three regions (left, center, and right), which contain the data obtained from radar sensors R1, R2, and R3, respectively. Although there are clear differences among the three patterns, there are further distinctions that are not clearly visible. Note that the starting samples of the digit 0 have higher values compared with the digit 1. Similarly, the right side of the digit 5 is denser than that of 1.

2) Results of Conventional CNN Using Raw Data for Digit Classification: Since our aim was to overcome the problem of decreased accuracy when the orientation or speed of a gesture changed, we present the result of conventional CNN-based gesture recognition for four different cases. In the first case,

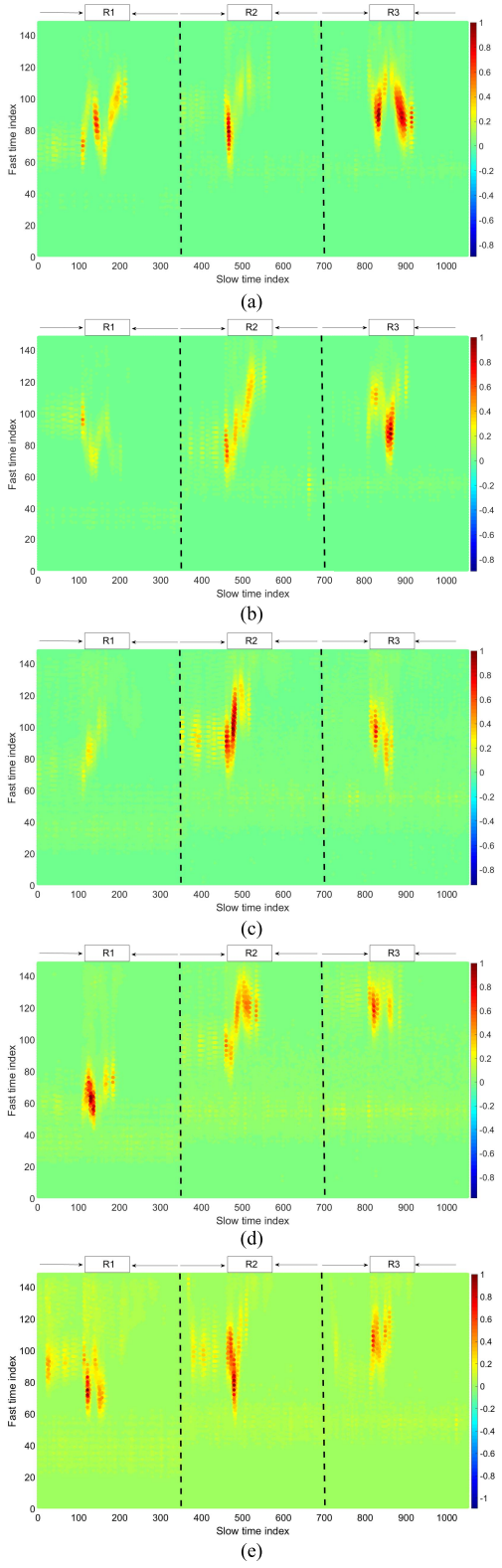

Fig. 18. Colored images of radar data referring to the digit 2 with (a) slow speed and straight orientation, (b) different users with varied shape/size of the hand, (c) slow speed and left-tilted orientation, (d) slow speed, straight orientation, and shifted on the right side (distance shift), and (e) fast speed and straight orientation.

the trained and tested gestures have the same orientation and the same volunteer was used for training and testing. In the second case, different volunteers were used for testing to show dependence on users' hand shape and size. In the third

TABLE IV

ACCURACY OF DIGIT RECOGNITION WHEN TRAINING AND TESTING PERFORMED ON THE SAME INDIVIDUAL

		Detected gesture									
Original gesture		0	1	2	3	4	5	6	7	8	9
	0	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	1	0%	96%	4%	0%	0%	0%	0%	0%	0%	0%
	2	0%	0%	100%	0%	0%	0%	0%	0%	0%	0%
	3	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%
	4	0%	0%	0%	0%	100%	0%	0%	0%	0%	0%
	5	0%	0%	0%	0%	0%	98%	2%	0%	0%	0%
	6	0%	0%	0%	0%	0%	0%	100%	0%	0%	0%
	7	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%
	8	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
	9	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%

TABLE V

ACCURACY OF DIGIT RECOGNITION WHEN TRAINING AND TESTING DATA WERE PERFORMED ON DIFFERENT PERSONS

		Detected gesture									
Original gesture		0	1	2	3	4	5	6	7	8	9
	0	28%	0%	0%	2%	0%	0%	10%	0%	0%	60%
	1	0%	0%	14%	78%	0%	0%	0%	0%	8%	0%
	2	0%	2%	36%	0%	8%	0%	0%	6%	30%	18%
	3	0%	0%	4%	42%	0%	0%	0%	18%	36%	0%
	4	0%	0%	0%	0%	58%	0%	10%	0%	0%	32%
	5	0%	0%	0%	14%	0%	32%	0%	26%	0%	28%
	6	0%	0%	0%	18%	0%	0%	48%	0%	0%	34%
	7	0%	30%	0%	0%	6%	0%	0%	52%	0%	12%
	8	0%	0%	16%	0%	0%	10%	4%	0%	64%	6%
	9	8%	16%	0%	0%	0%	0%	0%	0%	4%	72%

case, the testing gestures are tilted left or right or shifted to the left or right as compared with the trained gestures. In the fourth case, there is a change in gesture speed between the training and testing data. For all the cases, we have used five volunteers to collect the gesture classification data, for a total of 400 repetitions of each digit. During CNN training and evaluation, 75% of the data were used for training and 25% were used for evaluation.

a) Results when trained and tested gestures have the same orientation and speed:

(i) When the training and testing are performed for the same individual: We trained the digits from 0 to 9 for different volunteers using the conventional CNN technique. However, the testing was performed on the same individuals involved in training. The CNN that we employed in the above case has epochs of 60, 1 iteration per epoch, and a

TABLE VI

ACCURACY OF DIGIT RECOGNITION WHEN TRAINING AND TESTING DATA HAVE DIFFERENT ORIENTATIONS AND DISTANCES

		Detected gesture									
Original gesture		0	1	2	3	4	5	6	7	8	9
	0	73%	0%	5%	0%	0%	17%	0%	0%	3%	2%
	1	0%	12%	0%	0%	32%	0%	51%	5%	0%	0%
	2	1%	7%	39%	11%	0%	27%	0%	0%	7%	8%
	3	0%	12%	16%	14%	0%	2%	17%	0%	34%	5%
	4	0%	0%	13%	0%	57%	6%	0%	16%	8%	0%
	5	0%	2%	0%	2%	38%	41%	14%	0%	3%	0%
	6	5%	0%	15%	0%	12%	0%	32%	0%	22%	14%
	7	0%	8%	0%	2%	7%	0%	3%	74%	5%	1%
	8	10%	0%	0%	5%	0%	0%	16%	0%	65%	4%
	9	9%	0%	7%	0%	8%	0%	17%	0%	21%	38%

TABLE VII

ACCURACY OF DIGIT RECOGNITION WHEN TRAINING AND TESTING DATA HAVE DIFFERENT SPEEDS

		Detected gesture									
Original gesture		0	1	2	3	4	5	6	7	8	9
	0	26%	0%	0%	24%	14%	32%	0%	0%	4%	0%
	1	0%	54%	26%	0%	0%	6%	2%	8%	4%	0%
	2	0%	6%	28%	0%	14%	22%	0%	30%	0%	0%
	3	0%	54%	0%	32%	0%	4%	0%	8%	2%	0%
	4	0%	6%	4%	0%	62%	18%	6%	4%	0%	0%
	5	0%	0%	8%	0%	0%	70%	12%	10%	0%	0%
	6	8%	0%	0%	0%	14%	32%	40%	0%	6%	0%
	7	0%	0%	4%	18%	0%	0%	0%	78%	0%	0%
	8	0%	0%	4%	0%	0%	68%	0%	24%	4%	0%
	9	12%	0%	0%	8%	0%	36%	0%	14%	0%	30%

learning rate of 0.01. The average elapsed training time was 5 min and 57 s.

Table IV shows almost perfect accuracy when the mid-air hand gestures are performed with the same speed and in the same orientation.

(ii) When the testing is performed for different individuals: When we tested classification accuracy on different volunteers from those who produced the training data, the results changed considerably and classification accuracy was degraded due to the dependence of the raw data on the shape and size of the human hand; thus, the conventional method performed poorly when the volunteers were changed for testing the trained system. Five different volunteers were used for experiments and the size of data set was 500.

TABLE VIII
ACCURACY OF DIGIT RECOGNITION BY MULTIPLE USERS (VERTICAL CASE)

		Detected gesture									
Original gesture		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
	0	100%	0%	0%	0%	0%	0%	0%	0%	0%	0%
	1	0%	100%	0%	0%	0%	0%	0%	0 %	0%	0%
	2	0%	0%	100 %	0%	0%	0%	0%	0%	0%	0%
	3	0%	0%	0%	100%	0%	0%	0%	0%	0%	0%
	4	0%	0%	1%	0%	98%	0%	1%	0%	0%	0%
	5	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%
	6	0%	0%	0%	0%	0%	0%	97 %	0%	3 %	0%
	7	0%	0%	0%	0%	0%	0%	0%	100%	0%	0%
	8	0%	0%	0%	0%	0%	0%	0%	0%	100%	0%
	9	0%	0%	0%	0%	0%	0%	0%	0%	0%	100%

TABLE IX
ACCURACY OF DIGIT RECOGNITION WITH DIFFERENT ORIENTATIONS AND DISTANCES

		Detected gesture									
Original gesture		0	1	2	3	4	5	6	7	8	9
	0	98%	0.0%	0.0%	0.0%	0.0%	0.0%	2%	0.0%	0.0%	0.0%
	1	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	2	0.0%	0.0%	99%	0.0%	0.0%	0.0%	0.0%	0.0%	1%	0.0%
	3	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	4	0.0%	0.0%	0.0%	0.0%	97%	0.0%	1%	0.0%	2%	0.0%
	5	0.0%	0.0%	0.0%	0.0%	0.0%	98%	1%	0.0%	1%	0.0%
	6	2%	0.0%	1%	0.0%	0.0%	0.0%	97%	0.0%	0.0%	0.0%
	7	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%
	8	0.0%	0.0%	0.0%	0.0%	1%	0.0%	2%	0.0%	97%	0.0%
	9	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	3%	97%

Table V shows the accuracy decrease that occurs if training and testing are performed with different subjects, which means that the conventional algorithm is not robust against the shape and size of the users' hands.

b) Results when trained and tested gestures have different orientations and distances: In this case, the digits were trained in one orientation (straight) and the test was performed at different orientations. The total gesture set for testing comprised of 1000 gestures. Of these gestures, 25% gestures were left-tilted, 25% were right-tilted, 25% were shifted to the left from the center of the plane, and 25% were right shifted. The accuracy results of this experiment are presented in Table VI.

The overall accuracy decreased to 44.5% because when the image is tilted, the raw data is completely different from the original version of the trained data.

c) Results when trained and tested gestures have different speeds: To find the effect of gesture speed on the accuracy of the conventional CNN algorithm, we trained gestures at normal speed (an average speed of 7.52 cm/s) and tested them on higher hand motion speeds (an average speed of 13.5 cm/s). The total gesture set for testing contained 500 gestures. Table VII shows the confusion matrix of the result of each gesture.

For illustration, we show image data for the digit 2 when it was written in a straight orientation at slow speed, by a

TABLE X
ACCURACY OF DIGIT RECOGNITION WITH DIFFERENT SPEEDS

		Detected gesture									
		0	1	2	3	4	5	6	7	8	9
Original gesture	0	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	1	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	2	0.0%	0.0%	96.0%	0.0%	4%	0.0%	0.0%	0.0%	0.0%	0.0%
	3	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	4	0.0%	0.0%	0.0%	0.0%	90.0%	0.0%	8%	0.0%	2%	0.0%
	5	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%
	6	0.0%	0.0%	4%	0.0%	0.0%	0.0%	96.0%	0.0%	0.0%	0.0%
	7	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%
	8	6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	94%	0.0%
	9	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%

different user, left-tilted at a slow speed, shifted on the left side, and with a straight orientation but a high speed in Fig. 18. Comparing Fig. 18(a) with Fig. 18(b)–(e), the original digit 2 pattern shown in Fig. 18(a) changes; the classification algorithm classified it as 9, 1, 5, and 7 which are false negatives. The pattern of the straight but quickly written 2 is very different from the one written at regular speed. Similarly, the signal pattern in all regions of Fig. 18(b) and (c) is distorted as compared with Fig. 18(a). The main reason why a conventional CNN using raw input data fails to produce accurate results when testing and training were performed on different users, or at different orientations or speeds, is that the raw radar data do not maintain spatial information when these parameters are changed. Hence, the results have too many false detections. On the contrary, localization-based handwriting only depends on the hand's trajectory, which has a stereotyped shape across orientations, speeds, and hand shapes.

F. Accuracy of Our Proposed Mid-Air Gesture Recognition

In our proposed algorithm, using hand tracking data for image creation to match text writing data, we have improved classification accuracy in cases in which the orientation and speed of the hand motions change. We collected a total of 2500 images for all digits. Of the total data set, 20% of the data were collected when the handwriting was performed in a straight orientation and tested on the same user, 20% gesture were tested on different volunteers, 10% of the gestures were tilted left, 10% of the gestures were tilted right, 10% were shifted to the left of the center, and 10% were shifted to the right. The remaining 20% were performed twice as fast as the other gestures.

1) Accuracy of Gestures Performed by Multiple Users (Vertical Case): To show the heterogeneity of the proposed algorithm, we obtained gesture data from five volunteers. The gestures were made in a vertical direction. The total gesture set comprised of 1000 gestures. The confusion matrix that shows the accuracy of these experiments is given in Table VIII.

2) Accuracy of Gestures With Different Orientations and at Different Distance Shifts: In this section of the experiment, we performed gestures in different orientations and at different locations inside the plane of the radar sensors. The total gesture set comprised 1000 gestures. Of these, 25% were made in a left-tilted orientation and 25% were performed in a right-tilted orientation. These gestures were centered in the sensing plane of the radar sensors. However, to prove that the proposed algorithm also works if the center is shifted at different distances, we translated 25% gestures to the right and 25% to the left. The results showed that the orientation change did not significantly change the accuracy of our proposed algorithm, as shown by the confusion matrix in Table IX.

3) Accuracy of Gestures With Variable Speed: To show the effect of gesture speed on the proposed algorithm's performance, we made gestures with different speeds. Some gestures were slow, with an average speed of 7.46 cm/s, whereas others were fast, at an average speed of 12.92 cm/s. The total gesture set was composed of 500 gestures. The accuracy results of recognizing gestures with different hand motion speeds are given in Table X.

Table X shows that even changes in hand motion speed do not change the accuracy results, proving that gesture recognition based on trajectory data is robust against orientation, user, and gesture speed changes.

4) Processing Time for Proposed and Conventional Algorithms: Finally, we have calculated the average processing time for both the conventional and proposed algorithm. We processed and classified gestures in real time. Performance was evaluated on a PC with an Intel Core i5-4460 processor with a 3.2-GHz cycle frequency and 8 GB of RAM. We used MATLAB to evaluate both algorithms. The average processing time for the conventional algorithm was found to be 0.0364 s, whereas that of our proposed method was 0.0522 s. The time taken by the preprocessing step was 0.0007 s, the time for the localization and tracking step was 0.0068 s, and image transformation and classification took 0.0447 s. This processing time was fast enough for making real-time inferences.

IV. Conclusion

In this paper, we presented a technique for mid-air digit writing using radio sensors. Our proposed method uses hand tracking information to generate an image of the intended digit. After getting the tracking data, we used a CNN for digit classification, a method that proved to be highly accurate. We also checked the accuracy of digit recognition using raw data as input to the CNN. The accuracy obtained using the raw data and the CNN was high when the training and testing data had the same orientation and distance; however, after changing the orientation during the evaluation, the resulting accuracy was much lower. In contrast, using our proposed method, which uses the CNN to transform the tracking data into an image for classification, even when we wrote the digits in mid-air in different orientations, the resulting accuracy was still very high. Another main advantage of our algorithm was that we did not use any special training data, simply a huge already available database for image-based digit recognition, which yielded high gesture recognition accuracy for different users. Since that database had thousands of examples for digits in different styles, orientations, and shapes, orientation changes had almost no effect on the recognition accuracy of our proposed technique. In this paper, we studied single digit writing in mid-air. However, because radar sensors with a narrow beam pattern were used, the sensing area does not cover a 3-D space, so this paper has focused on 2-D gesture recognition. If gesture recognition was performed using a sensor with an omnidirectional antenna, the sensing area could be expanded to 3-D. When using such sensor hardware, it would be possible to recognize a number in 3-D space by simply adding the z-axis to the proposed algorithm. In the future work, we would like to perform mid-air alphabet character recognition and combine character streams into words.

References

[1] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, Jan. 2015.

[2] N. H. Dardas and N. D. Georganas, "Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 11, pp. 3592–3607, Nov. 2011.

[3] G. Plouffe and A.-M. Cretu, "Static and dynamic hand gesture recognition in depth data using dynamic time warping," *IEEE Trans. Instrum. Meas.*, vol. 65, no. 2, pp. 305–316, Feb. 2016.

[4] C. Zhu and W. Sheng, "Wearable sensor-based hand gesture and daily activity recognition for robot-assisted living," *IEEE Trans. Syst., Man, Cybern.*, vol. 41, no. 3, pp. 569–573, May 2011.

[5] Y. Zhang and C. Harrison, "Tomo: Wearable, low-cost electrical impedance tomography for hand gesture recognition," in *Proc. 28th Annu. ACM Symp. User Interface Softw. Technol.*, Nov. 2015, pp. 167–173.

[6] F. Khan, S. K. Leem, and S. H. Cho, "Hand-based gesture recognition for vehicular applications using IR-UWB radar," *Sensors*, vol. 17, no. 4, p. 833, Apr. 2017.

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

[8] B. Dekker, S. Jacobs, A. S. Kossen, M. C. Kruithof, A. G. Huizing, and M. Geurts, "Gesture recognition with a low power FMCW radar and a deep convolutional neural network," in *Proc. Eur. Radar Conf.*, Oct. 2017, pp. 163–166.

[9] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, and T. W. Kim, "A hand gesture recognition sensor using reflected impulses," *IEEE Sensors J.*, vol. 17, no. 10, pp. 2975–2976, May 2017.

[10] S. K. Leem, F. Khan, and S. H. Cho, "Vital sign monitoring and mobile phone usage detection using IR-UWB radar for intended use in car crash prevention," *Sensors*, vol. 17, no. 6, p. 1240, May 2017.

[11] M. Ruzi, E. Robertson, and C. Mätzler, "MATLAB functions for the extraction of refractive indices from aerosol extinction spectra," Dept. Inst. Appl. Phys., Univ. Bern, Bern, Switzerland, Res. Rep. 2018-01-MW, 2018.

[12] S. K. Mitra, *Digital Signal Processing: A Computer—Based Approach*, vol. 2. New York, NY, USA: McGraw-Hill, 2006.

[13] W. K. Pratt, *Digital Image Processing: PIKS Scientific Inside*, vol. 4. Hoboken, NJ, USA: Wiley, 2007.

[14] W. Dargie and C. Poellabauer, *Fundamentals of Wireless Sensor Networks: Theory and Practice*. New York, NY, USA: Wiley, 2010.

[15] W. S. Murphy and W. Hereman, "Determination of a position in three dimensions using trilateration and approximate distances," *Decis. Sci.*, vol. 7, p. 19, Oct. 1995.

[16] N. Khan, L. U. Khan, M. I. Khattak, M. Shafi, and N. Ullah, "Recovery of information through linear prediction technique in attitude estimation of spacecraft systems," *Measurement*, vol. 66, pp. 253–262, Apr. 2015.

[17] N. Khan *et al.*, "Implementation of linear prediction techniques in state estimation," in *Proc. 10th Int. Bhurban Conf. Appl. Sci. Technol.*, Jan. 2013, pp. 77–83.

[18] R. C. Gonzalez, *Digital Image Processing Basics*, 3rd ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2008.

[19] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, Fort Lauderdale, FL, USA, Apr. 2011, pp. 315–323.

[20] M. H. Beale, M. T. Hagan, and H. B. Demuth, *Neural Network Toolbox User's Guide*. Natick, MA, USA: The Math Works Inc., 2017.

Seong Kyu Leem (GS'18) was born in Sejong-si, South Korea, in 1980. He received the B.S. degree in electrical engineering from Korea University, Seoul, South Korea, in 2003, and the M.S. degree in electrical engineering and computer science from Seoul National University, Seoul, in 2005. He is currently pursuing the Ph.D. degree in electronics and computer engineering with Hanyang University, Seoul.

His current research interests include RF impairment compensation for radio communication system and radar signal processing.

Faheem Khan (M'19) was born in Bannu, Pakistan, in 1988. He received the Ph.D. degree in electronics and computer engineering from Hanyang University, Seoul, South Korea, in 2018.

He is currently a Post-Doctoral Researcher with Hanyang University. He has authored several conference papers and articles in reputed journals. His current research interests include radar signal processing, computer vision, and localization, tracking, and gestures recognition.

Sung Ho Cho (S'86–M'88) received the Ph.D. degree in electrical and computer engineering from the University of Utah, Salt Lake City, UT, USA, in 1989.

From 1989 to 1992, he was a Senior Member of Technical Staff with the Electronics and Telecommunications Research Institute, Daejon, South Korea. In 1992, he joined the Department of Electronic Engineering, Hanyang University, Seoul, South Korea, where he is currently a Professor. His current research interests include applied signal processing, machine learning for signal processing, context aware computing, radar sensors, and smart space and wireless network.

Dr. Cho was a recipient of the High-Level Foreign Experts Fellowship at the Beijing University of Posts and Telecommunications, Beijing, China, led by the Ministry of Education of China.