Progress in Human Computer Interaction
2021 Volume 1, Issue 1
ISSN: 2630-4635

Design and Implementation of Human and Object Classification System Using FMCW Radar Sensor

Yunsung Sim[1], Seungjun Song[1], Seonyoung Jang[1], Yunho Jung[2]*

[1]School of Electronics and Information Engineering, Korea Aerospace University, Goyang, Gyeonggi, Republic of Korea

[2]Department of Smart Air Mobility, Korea Aerospace University, Goyang, Gyeonggi, Republic of Korea

***Corresponding author:** Yunho Jung, yjung@kau.ac.kr

Abstract

This paper proposes the design and implementation results for human and object classification systems utilizing frequency-modulated continuous wave (FMCW) radar sensors. Such a system requires the process of radar sensor signal processing for multi-target detection and the process of deep learning for the classification of humans and objects. Since deep learning requires such a great amount of computation and data processing, the lightweight process is of utmost essential. Therefore, a binary neural network (BNN) structure was adopted, operating convolution neural network (CNN) computation in a binary condition. In addition, for the real-time operation, a hardware accelerator was implemented and verified via the field-programmable gate array (FPGA) platform. Based on performance evaluation and verified results, it is confirmed that the accuracy for multi-target classification of 90.5%, reduced memory usage by 96.87% compared to CNN, and the run time of 5 ms are achieved.

Keywords

BNN accelerator

Embedded system

Frequency-modulated continuous wave (FMCW) radar

Field-programmable gate array (FPGA)

Multi-target tracking

1. Introduction

With the recent advancement of Internet of Things (IoT) technology, the demand for security in indoor spaces has been increasing. Technologies for detecting intruders in specific spaces, including not only personal living spaces but also public areas, are being developed using cameras, infrared sensors, and radar sensors [1,2]. These indoor spatial object recognition technologies require small and lightweight embedded systems due to their continuous operation. Cameras and infrared sensors are sensitive to external factors such as light and temperature, which can lead to the inability to detect objects or misclassification. In addition, cameras require human monitoring, and there are privacy concerns related to hacking. In contrast, radar sensors are less affected by external conditions and can recognize objects in both dark and bright environments. Furthermore, they can classify and observe objects based on frequency changes caused by motion, addressing privacy issues. In recent years, radar sensor-based object recognition and classification technologies have been researched [3,4].

Radar sensors can be categorized into continuous wave (CW) radar sensors and frequency-modulated continuous wave (FMCW) radar sensors depending on whether the signal is modulated [5]. CW radar sensors, also known as Doppler radar sensors, continuously emit a specific frequency without signal modulation. They measure the velocity of objects based on the signal reflected from them. However, since they do not store time information, they cannot provide distance information about objects, which is a drawback. FMCW radar sensors operate by continuously emitting signals with modulated frequencies. The time information of detected objects is used to detect distance information. This feature makes them suitable for detecting multiple objects at different distances.

The data obtained from radar sensors are processed for object detection using signal processing techniques such as fast Fourier transform (FFT), constant false alarm rate (CFAR) [6-9], and density-based spatial clustering of application with noise (DBSCAN) [10,11]. Each object is characterized by a Doppler profile that accumulates Doppler components. Therefore, if the Doppler information for each object is not separated and detected, information about the objects may be lost. To address this issue, tracking algorithms are applied to store the Doppler information of each object [12].

Object classification using radar sensors typically involves the application of deep learning techniques such as convolutional neural network (CNN) rather than traditional machine learning methods like support vector machine (SVM) and k-nearest neighbors algorithm (KNN). Nevertheless, CNNs require high computational power and a significant amount of memory. To solve this problem, binary neural networks (BNNs) are used, where information used in CNN operations such as features and weights are binarized, reducing memory usage and computational complexity, making it suitable for hardware implementation [13].

Previous object detection studies using CW radar sensors focused on detecting single objects and may not be suitable for real-life situations with various objects [14,15]. Therefore, we have advanced previous research by conducting multiple object detection research using FMCW radar. In this process, tracking algorithms were used to track the information of each detected object. Additionally, BNNs were applied to classify objects based on Doppler characteristics that differentiate between the movements of humans, objects, and dogs. To process the radar sensor signals, software running on Raspberry Pi was designed to generate Doppler profiles required for BNN operations. To ensure real-time operation, BNN operations are implemented using a field-programmable gate array (FPGA) hardware accelerator.

The structure of this paper is as follows: Section 2 provides the background knowledge to understand the system; Section 3 covers the overall structure and performance evaluation of the system; Section 4 discusses the system's design and implementation; Section 5 presents the experimental results; and Section 6 concludes the paper.

2. Background

2.1. FMCW radar sensor

FMCW radar sensor operates by modulating the transmission frequency and emitting it. The frequency is modulated from f_L to f_H, as shown in **Figure 1**, and this range is called a 'chirp'. The time for one chirp is denoted by T_{chirp}, and multiple chirps form a single frame. The time for one frame is denoted by T_{frame}. FMCW radar sensors continuously transmit and receive frames to measure distance and velocity.

Figure 1. Process of R-D map configuration using FFT

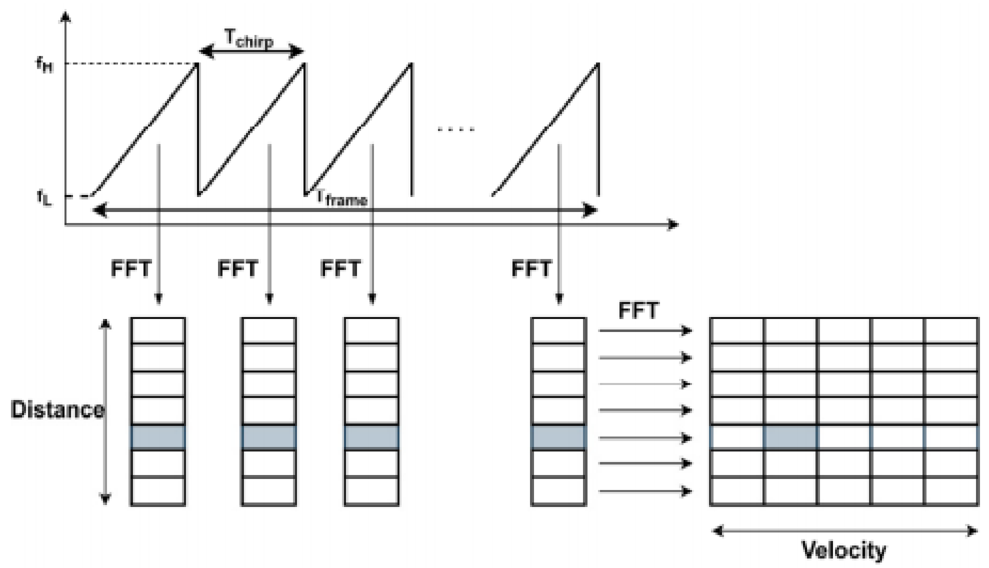
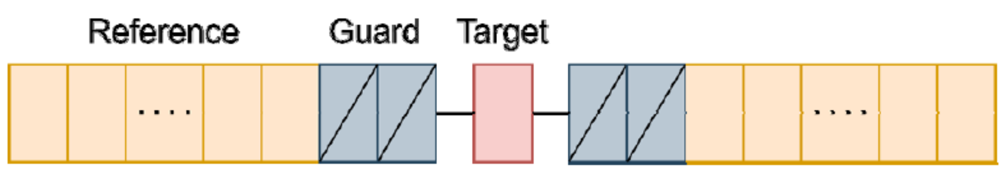


Figure 2. Cell configuration of CFAR algorithm



2.2. FFT algorithm

The distance and velocity information of objects are obtained by performing FFT on the data received from the radar sensor. While there is a 2D FFT method that can obtain both distance and velocity information simultaneously, we applied the 1D FFT method twice to reduce processing time. First, the object's distance information is obtained by performing FFT on samples within a chirp. Subsequently, the FFT results for chirps are accumulated at the frame level, and FFT is performed again to detect the object's velocity information. Two FFT operations result in the formation of one Range-Doppler (R-D) map per frame, with each axis containing distance and velocity information. **Figure 1** shows a pictorial representation of the process.

2.3. CFAR algorithm

The presence of objects in the frequency domain is typically determined by detecting signals larger than a predefined threshold. When the threshold is fixed, there can be false detections when the signal strength is weak or when clutter signals exceed the threshold. Thus, radar sensor systems use the CFAR algorithm to calculate variable thresholds to maintain a constant false alarm rate. As shown in **Figure 2**, CFAR consists of a target cell surrounded by a reference cell (r_{cell}) and a guard cell (g_{cell}). It uses false probability (P_{fa}) to calculate the threshold.

CFAR is commonly categorized into cell average CFAR (CA-CFAR) and ordered statistic CFAR (OSCFAR). CA-CFAR calculates the threshold based on the average of r_{cell} values, resulting in low computational requirements. However, it has the drawback of lower detection probability when cell information between objects overlaps [6,7]. On the other hand, OS-CFAR arranges r_{cell} values and determines the threshold value based on the *k*th cell, allowing it to distinguish between noise and actual objects [6,8]. This method provides excellent performance even when multiple objects are present within r_{cell}. Nonetheless, it increases computational complexity and, consequently, processing time due to the sorting process. To address this, an algorithm that finds the *k*th value through comparison operations has been proposed [16]. In this paper, we implemented OS-CFAR using this algorithm.

2.4. DBSCAN algorithm

The DBSCAN algorithm groups multiple nonlinear data sets into clusters, representing them as a single data set. It identifies data as valid if there are more than minimum points (Minpts) within a specific range and treats the rest as noise. In the implemented system, each data point represents a target detected by the CFAR algorithm. The DBSCAN algorithm simultaneously constructs clusters for individual targets and separates data from different targets into different clusters, distinguishing between multiple objects.

2.5. Tracking algorithm

To classify objects detected by the radar sensor, Doppler profiles are used. In a multi-object environment, tracking algorithms are essential to accumulate Doppler information for each object's movement. Tracking algorithms are divided into association, management, and filtering. Association updates the track based on the closest data from the previous frame's predicted object data and the current observed data. Management determines whether to create new tracks or terminate tracks based on updated data. Filtering predicts the distance and velocity of objects in the next frame based on the current distance and velocity information.

2.6. BNN

CNN, a deep learning technology, extracts features through convolutional operations with learned kernels and input images. Subsequently, a fully connected layer (FCL) classifies features probabilistically. BNN is a type of deep learning network that binarizes all parameters of conventional CNNs and performs both training and inference. As a result, it can significantly reduce computational requirements by replacing floating-point multiplication and accumulation processes with XNOR and popcount operations. Moreover, it reduces memory usage for storing parameters and operation results, enabling fast computation. This structure is suitable for embedded systems that require lightweight and compact design.

3. Overview of the system

3.1. System structure

The system overview is depicted in **Figure 3**. Raspberry Pi transmits command signals to drive the radar sensor through universal asynchronous receiver /transmitter (UART) communication and receives raw data generated by the radar sensor. The received data undergoes DC removal to eliminate noise components. For signal processing of the radar sensor, FFT and CFAR algorithms are applied. To reduce time complexity, 2D operations are separated into 1D operations for distance and velocity [9]. A 64-point FFT is performed for each chirp unit to detect frequency values according to distance, and OS-CFAR is used to detect object distances. Based on the detected frequencies, a 64-point FFT is performed at the frame level to obtain velocity information. The R-D map generated from the two FFT results undergoes subtraction with the previous frame's results to remove static signal components [17]. OS-CFAR is then applied along the velocity axis to detect objects, including distance and velocity information.

Multiple detected objects are separated into humans and objects, and Doppler profiles are required for each. Therefore, a tracking process is applied to create Doppler profiles for humans and objects. The tracking process first generates clusters using the DBSCAN algorithm and selects representative values based on signal strength, followed by the execution of tracking algorithms. The Doppler profiles generated for each object are shown in **Figure 4**. The Doppler profiles are transmitted to the FPGA through serial peripheral interface (SPI) communication. BNN

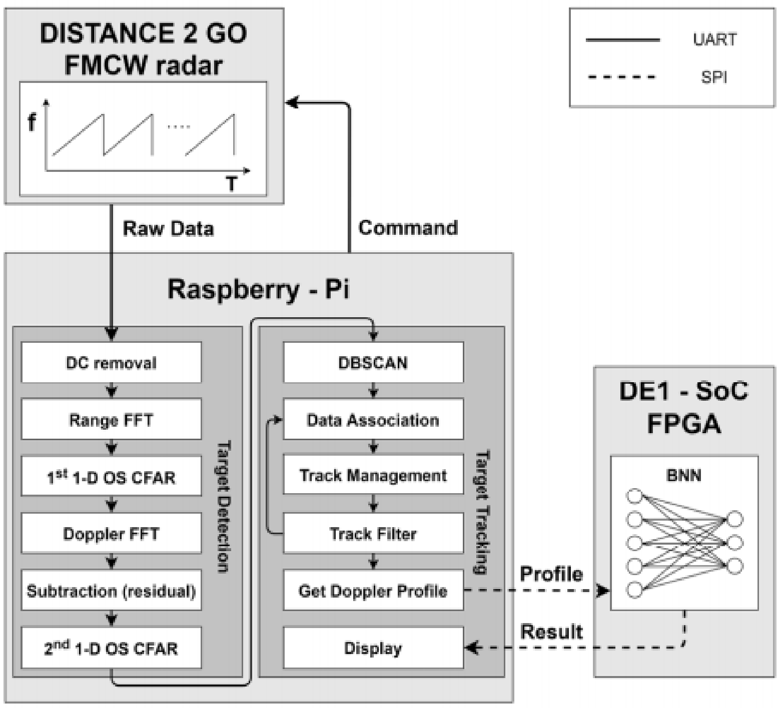


Figure 3. Data processing flowchart for the proposed system

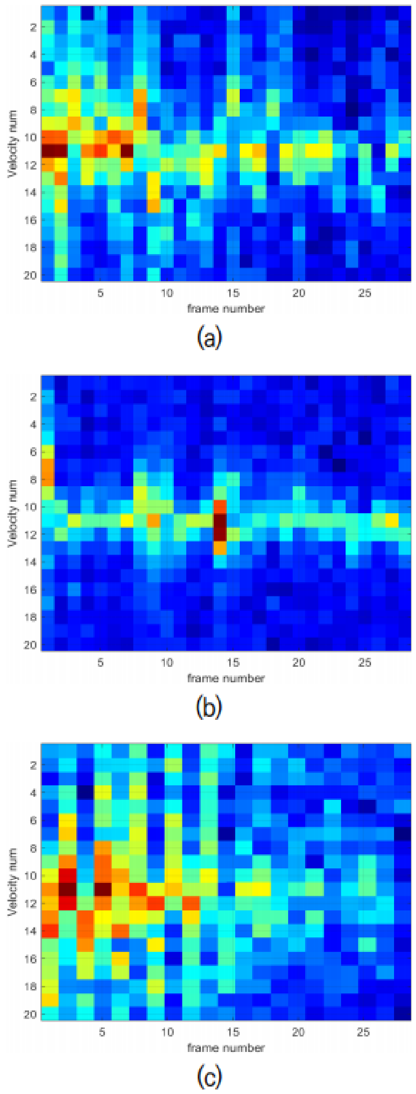


Figure 4. Doppler profile for **(a)** human, **(b)** object, and **(c)** dog

operations classify the objects and transmit the results to Raspberry Pi.

3.2. BNN performance evaluation

BNN is implemented as a hardware accelerator on the FPGA, considering network size, accuracy, and memory usage. The network classifies three classes: humans, objects, and dogs, using a total of 1,454 data, including 471 humans, 495 objects, and 488 dogs. Out of these, 1,154 data points were used for training, and the remaining 300 were used for evaluation. During training, cross-entropy loss function and Adam optimizer were employed, with a learning rate of 0.0005, 300 epochs, and a batch size of 8.

Table 1 presents the results of accuracy and memory usage based on the composition of convolution layers and FCL in BNN. The combination of three convolution layers and two FCL layers achieved the highest accuracy. **Table 2** compares accuracy and memory usage between BNN and CNN while varying the depth at each stage to reduce memory usage. In this study, a BNN with an accuracy of 90.5% and approximately 198 KB of memory usage was implemented, which uses 3.13% less memory compared to an identically configured CNN.

The final network structure is shown in **Figure 5**. The input image is a Doppler profile size of 20×28. The first convolutional layer uses a 5×5 kernel considering

Table 1. Accuracy and memory usage according to the BNN layer configuration

# Conv	# FCL	Accuracy (%)	Memory (KB)
2	1	80.3	23.9
	2	88.8	791.6
3	1	88.0	113.3
	2	92.3	420.8
4	1	70.0	385.5
	2	83.3	2,226.8
5	1	68.3	923.7
	2	79.3	1,537.9

Table 2. Accuracy and memory usage according to the BNN depth configuration

	CNN					
FCL nodes	**1024**			**512**		
Conv depth	**128**	**256**	**512**	**128**	**256**	**512**
Accuracy (%)	98.3	99	99	99	99	99
Memory (KB)	2,777	7,902	25,229	1,985	6,323	22,077
	BNN					
FCL nodes	**1024**			**512**		
Conv depth	**128**	**256**	**128**	**256**	**128**	**256**
Accuracy (%)	85.3	91.1	92.3	79	90.5	92
Memory (KB)	86.8	246.9	788.4	62.0	197.6	689.0

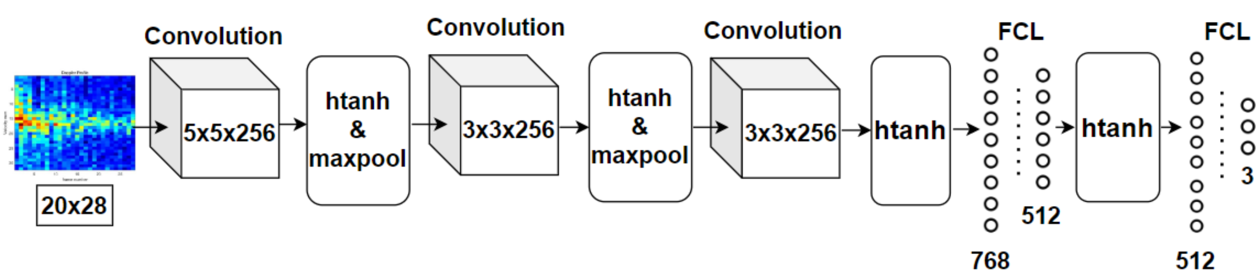


Figure 5. BNN structure of the proposed system

Doppler characteristics due to motion, and subsequent convolution layers use 3×3 kernels. Each layer employs the hard hyperbolic tangent as the activation function, along with max pooling and batch normalization.

4. System design and implementation results

To validate the operation of the proposed system, the Infineon Distance2GO product was used [18]. Additionally, software running on the Raspberry Pi 4 Model B processed signals obtained from the radar sensor [19]. BNN operations for classifying detected Doppler profiles were implemented as a hardware accelerator on the Altera Cyclone V5CSEMA5F31C6N FPGA [20] and the results were displayed on a monitor connected to Raspberry Pi. **Figure 6** shows the connection of the system.

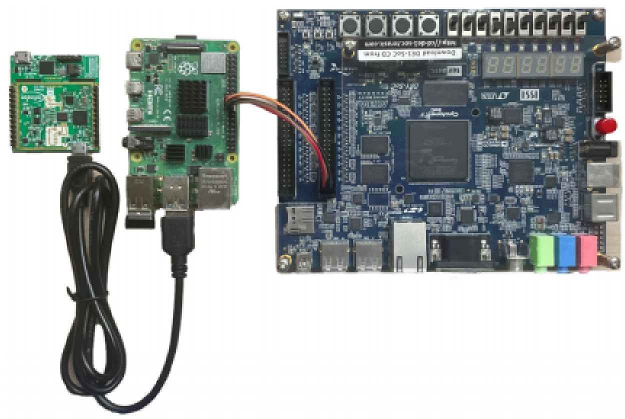


Figure 6. Verification environment of the proposed human and object classification system

Table 3. Parameter used in OS-CFAR operation

Parameters	Value
r_{cell_r} (range)	10
g_{cell_r} (range)	0
r_{cell_v} (velocity)	10
g_{cell_v} (velocity)	2
P_{fa}	10^{-5}
$k_{\_r}$ (range)	8
$k_{\_v}$ (velocity)	10

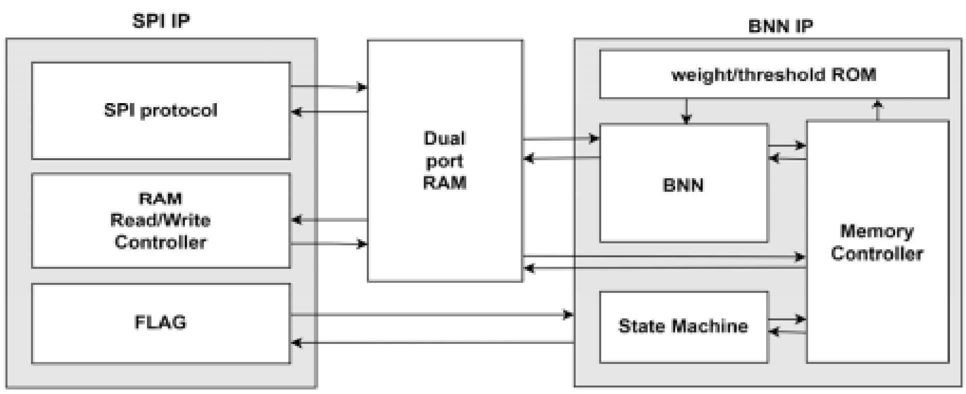


Figure 7. Structure of the proposed hardware accelerator

4.1. Software design and implementation

The software consists of the radar sensor communication module, signal preprocessing module, and output module. The radar sensor communication module is divided into two parts: one for sending initial parameters for using the radar sensor and the other for receiving data obtained from the radar sensor. The initial parameters include the radar sensor's operating frequency of 24 GHz, bandwidth of 200 MHz, 32 samples per chirp, and 48 chirps per frame. The signal preprocessing module performs FFT, CFAR, DBSCAN, and tracking algorithms. CFAR parameters for classifying objects and noise are listed in **Table 3**. The output module displays the R-D map and class classification results on the monitor.

4.2. Hardware design and implementation

The designed hardware configuration consists of an SPI block for communication and a BNN block for classification, as shown in **Figure 7**. These two blocks share data through dual-port RAM. In the BNN block, there are components such as the BNN unit for performing computations, a state machine for controlling the computation steps, a memory controller, and a read-only memory (ROM) for storing required weights and thresholds. The BNN unit performs multiplication and accumulation operations using XNOR and popcount operators. Furthermore, batch normalization and the activation function, hard hyperbolic tangent, are implemented as simple comparators. **Figure 8** illustrates this process.

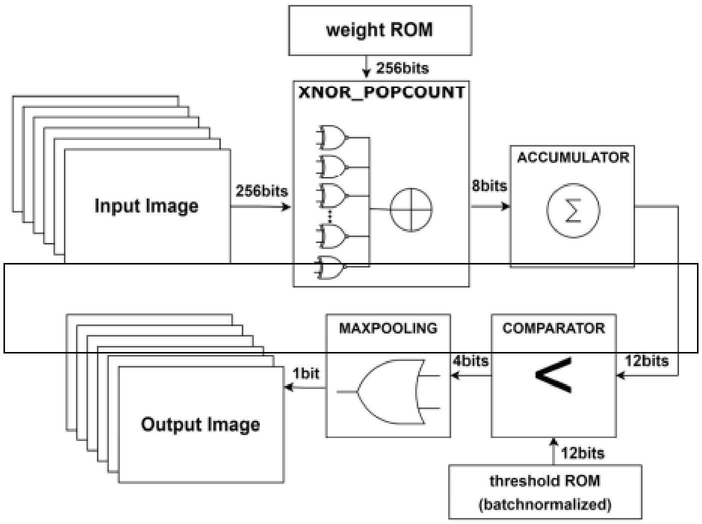


Figure 8. Unit structure of performing BNN operations

Table 4. Implementation results of the proposed system

Parameters	Value
Logic utilization	3,029
Total memory bits	1,688,876
Total registers	1,202

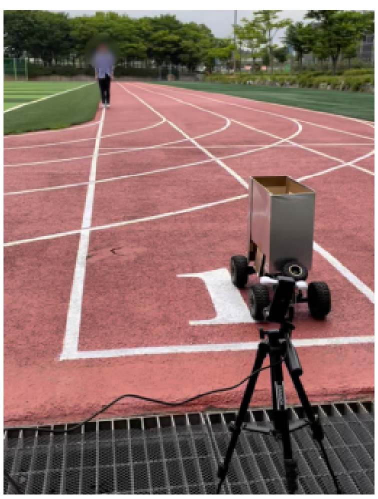


Figure 9. Experiment environment

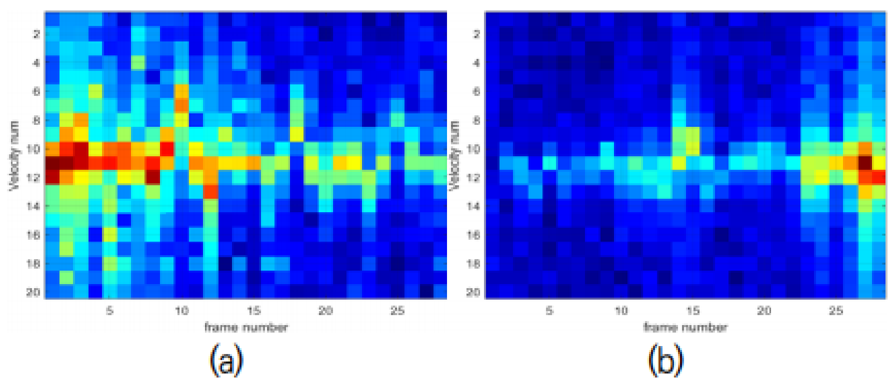


Figure 10. Doppler profile observed through experiment. **(a)** Human; **(b)** Object.

The BNN accelerator was developed using Verilog HDL, implemented, and verified on the FPGA, with an execution time of 5 ms confirmed. As shown in **Table 4**, the hardware system implementation utilized 3,029 logic, 1,689 Kbits of memory, and 1,202 registers.

5. Experimental results

The experimental setup is shown in **Figure 9**, where the radar sensor is placed at a height of 80 cm from the ground, and the time of one frame is 0.1 seconds. A person moved toward the radar sensor at a speed of about 1.8 m/s from a distance of 12 m, and an object moved away from the radar sensor at a speed of about 1.2 m/s from a distance of 3 m. The measurement time was 5 seconds. The measurement time is 5 seconds, and 50 frames of motion are accumulated for a person moving 9 meters and an object moving 6 meters. The preprocessed Doppler profile of the obtained data is shown in **Figure 10**.

Figure 11 displays Doppler profiles of objects obtained after preprocessing, classified into humans and objects by BNN, and represented with colors to indicate class. It is evident that the values of each object are tracked, enabling the recognition of humans and objects in a multi-object environment.

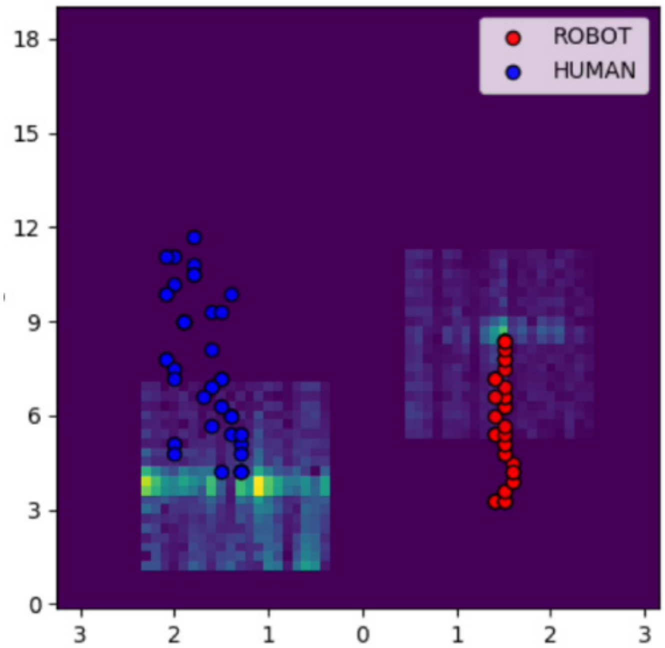


Figure 11. R-D map of the last frame

6. Conclusion

In this paper, a multi-object detection and classification system utilizing BNN was proposed. Profiles for indoor environment objects, including humans, objects, and dogs, were defined, and data preprocessing algorithms, including CFAR and tracking processes for recognizing each object, were implemented. To accelerate computations, the system was designed with BNN operations as a hardware accelerator, and it was confirmed to operate in real-time with a processing time of 5 ms. Performance evaluation was conducted on a total of 1,454 data points through experiments and simulations. The results showed classification accuracies of 88.2% for objects, 91.2% for people, and 92.1% for dogs, with a 96.87% reduction in memory usage compared to CNN.

Disclosure statement

The authors declare no conflict of interest.

Funding

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (No. 2020-0-00201, 2022-0-00960), and CAD tools were supported by IDEC.

References

[1] Jalil A, Matalangi M, 2021, Object Motion Detection in Home Security System Using the Binary-Image Comparison Method Based on Robot Operating System 2 and Raspberry Pi. ILKOM Jurnal Ilmiah, 13(1): 1–8. https://doi.org/10.33096/ilkom.v13i1.686.1-8

[2] Desnanjaya IGMN, Arsana INA, 2021, Home Security Monitoring System with IoT-Based Raspberry Pi. Indonesian Journal of Electrical Engineering and Computer Science, 22(3): 1295–1302. https://doi.org/10.11591/ijeecs.v22.i3

[3] Wang G, Muñoz-Ferreras J-M, Gu C, et al., 2014, Application of Linear-Frequency-Modulated Continuous-Wave (LFMCW) Radars for Tracking of Vital Signs. IEEE Transactions on Microwave Theory and Techniques, 62(6): 1387–1399. https://doi.org/10.1109/ TMTT.2014.2320464

[4] Lee H, Kim B-H, Yook J-G. 2018 IEEE Asia-Pacific Conference on Antennas and Propagation (APCAP), August 5–8, 2018: Path Loss Compensation Method for Multiple Target Vital Sign Detection with 24-GHz FMCW Radar. 2018, Auckland, 100–101. https://doi.org/10.1109/APCAP.2018.8538182

[5] Rohling H, Kronauge M. 2014 International Radar Conference, October 13–17, 2014: New Radar Waveform Based on a Chirp Sequence. 2014, Lille, 1–4. https://doi.org/10.1109/RADAR.2014.7060246

[6] Kronauge M, Rohlin H, 2013, Fast Two-Dimensional CFAR Procedure. IEEE Transactions on Aerospace and Electronic Systems, 49(3): 1817–1823. https://doi.org/10.1109/ TAES.2013.6558022

[7] Rohlin H, 1983, Radar CFAR Thresholding in Clutter and Multiple Target Situations. IEEE Transactions on Aerospace and Electronic Systems, AES-19(4): 608–621. https://doi.org/10.1109/TAES.1983.309350

[8] Rohling H. 12th International Radar Symposium (IRS), September 7–9, 2011: Ordered Statistic CFAR Technique – An Overview. 2011, Leipzig, 631–638.

[9] Zheng Q, Yang L, Xie Y, et al., Q. Zheng et al., 2021, A Target Detection Scheme with Decreased Complexity and Enhanced Performance for Range-Doppler FMCW Radar. IEEE Transactions on Instrumentation and Measurements, 70: 1–13. https://doi.org/10.1109/TIM.2020.3027407

[10] Ester M, Kriegel H-P, Sander J, et al., 1996, A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96), 1996: 226–231.

[11] Chauhan NS. DBSCAN Clustering Algorithm in Machine Learning. https://www.kdnuggets.com/2020/04/dbscan-clustering-algorithm-machine-learning.html

[12] Hyun E, Lee J-H. 17th International Radar Symposium (IRS), Mat 10–12, 2016: Multi-Target Tracking Scheme Using a Track Management Table for Automotive Radar Systems. 2016, Krakow, 1–5. https://doi.org/10.1109/IRS.2016.7497283

[13] Quang HV, Kim LW, Hong CS, 2021, A Deep Learning Accelerator for a Typical Binary Neural Network Layer. KCC, 2021: 597–599.

[14] Jang D, Kim H, Jung Y, 2020, Design and Implementation of Human-Detecting Radar System for Indoor Security Applications. Journal of Institute of Korean Electrical and Electronics Engineers, 24(3): 783–790. https://doi.org/10.7471/ikeee.2020.24.3.783

[15] Nam J, Kang C, Kook J, et al., 2021, Design and Implementation of CW Radar-Based Human Activity Recognition System. Korea Citation Index (KCI), 25(5): 415–425.

[16] Kong Y-J, Woo S-K, Park S, et al., 2018, OS CFAR Computation Time Reduction Technique to Apply Radar System in Real Time. The Journal of Korean Institute of Electromagnetic Engineering and Science, 29(10): 791–798. https://doi.org/10.5515/KJKIEES.2018.29.10.791

[17] Hossain MA, Hossain MI, Hossain MD, et al., 2019, Performance Assessment of Background Subtraction Algorithm. Korea Software Congress, 2019: 1049–1051.

[18] Infineon Technologies. DEMO-DISTANCE2GO. https://www.infineon.com/cms/en/product/ evaluation-boards/demo-distance2go/

[19] Raspberry-Pi. raspberry-pi-4-model-b. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

[20] Terasic Technologies. DE1-SoCBoard. https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=836.