

드론 검출을 위한 레이더 신호처리 시스템 설계

Design of Radar Signal Processing System for Drone Detection

김홍석·반규리·서지훈·정운호*

한국항공대학교 항공전자정보공학부

Hong-suk Kim · Gyu-ri Ban · Ji-hun Seo · Yunho Jung*

School of Electronics and Information Engineering, Korea Aerospace University, Gyeonggi-do 10540, Korea

[요 약]

본 논문에서는 FMCW (frequency modulated continuous wave) 레이더 센서를 활용하여 드론 검출이 가능한 시스템 설계 및 구현 결과를 제시한다. 드론 검출 시스템의 구현을 위해, 레이더 센서로부터 입력된 신호를 FFT, CFAR, clustering, tracking으로 이어지는 총 4단계의 신호처리 과정을 통해 객체를 탐지하고, 해당 객체를 드론과 다른 사물로 분류하기 위한 딥러닝 추론 과정을 수행한다. 딥러닝의 높은 연산량과 많은 메모리 요구를 감소시키기 위해 CNN (convolution neural network) 연산을 이진화하여 수행하는 BNN (binary neural network) 구조를 적용하였다. 성능 평가 및 검증 결과 89.33%의 객체 구분 정확도를 확인할 수 있었고, 총 수행 시간은 4 ms로 실시간 동작이 가능함을 확인하였다.

[Abstract]

In this paper, we present the design and implementation results of a system that classifies drones from other objects using an FMCW (frequency-modulated continuous wave) radar sensor. The proposed system detects various objects through a four-stage signal processing procedure, consisting of FFT, CFAR, clustering, and tracking, using signals received from the radar sensor. Subsequently, a deep learning process is conducted to classify the detected objects as either drones or other objects. To mitigate the high computational demands and extensive memory requirements of deep learning, a BNN (binary neural network) structure was applied, binarizing the CNN (convolutional neural network) operations. The performance evaluation and verification results demonstrated a drone classification accuracy of 89.33%, with a total execution time of 4 ms, confirming the feasibility of real-time operation.

Key word : AI Accelerator, BNN, Drone detection, Embedded system, FMCW radar sensor.

<http://dx.doi.org/10.12673/jant.2024.28.5.601>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 24 September 2024; Revised 26 October 2024

Accepted (Publication) 29 October 2024 (31 October 2024)

*Corresponding Author; Yunho Jung

Tel: +82-2-300-0133

E-mail: yjung@kau.ac.kr

1. 서론

최근 드론의 활용 분야는 구조나 관측, 상업적 목적 등 민간 영역으로까지 급속히 확대되고 있다. 그러나 드론 개발의 초기 목적은 정찰 및 감시에 있어, 적국이 정찰 목적으로 드론을 사용하는 상황을 염두에 두고 이에 대한 예방책을 마련할 필요가 있다. 이를 위해서는 다양한 객체와 드론을 구분하여 탐지하는 다중 객체 탐지 시스템이 필요하다. 객체 탐지에 사용되는 다양한 센서 중 대표적으로 카메라 센서가 있는데, 카메라 센서는 상대적으로 가격이 저렴하고 이미지를 통해 대상의 정보를 인식하여 세부 정보를 확인할 수 있다는 장점이 있으나, 저조도 환경에서 성능 저하가 발생한다는 단점이 있다 [1]. 또한 라이다 (LiDAR; light detection and ranging) 센서의 경우 [2], 정밀도가 높고 작은 물체를 우수하게 탐지하는 장점이 있으나, 악천후 조건에서 성능이 저하되는 등 두 센서 모두 환경에 따라 객체 탐지에 실패하거나 객체를 잘못 분류할 위험이 있다. 반면 레이다 (RADAR; radio detection and ranging) 센서의 경우 전파를 사용하여 외부 환경의 영향을 비교적 적게 받으면서도 높은 정확도로 객체를 탐지하고 분류할 수 있다는 장점이 있다. 이러한 특성으로 최근 몇 년간 레이다 센서를 활용한 객체 인식 및 분류 기술에 관한 연구가 활발히 진행되고 있다 [3],[4].

레이다 센서는 전파 신호 변조 여부에 따라 CW (continuous wave) 레이다와 FMCW (frequency-modulated continuous wave) 레이다로 구분할 수 있다 [5]. CW 레이다 센서는 신호 파형의 변형 없이 연속적으로 전파를 방사하여 물체의 Doppler 성분을 통해 물체의 속도 정보를 추출한다. 따라서 신호처리의 복잡도가 낮으나, 시간 정보를 저장하지 않아 물체의 거리를 측정할 수 없다는 단점이 존재한다. 반면 FMCW 레이다 센서는 주파수가 변조된 신호를 사용하여 연속적으로 전파를 방사함으로써 물체의 속도 및 거리 정보를 동시에 추정할 수 있다. 이러한 특성은 다중 객체를 탐지하는 데 적합하다.

레이다 센서로부터 획득한 데이터는 FFT (fast Fourier transform), CFAR (constant false alarm rate), DBSCAN (density-based spatial clustering of application with noise)으로 구성된 신호 전처리 과정을 거친다. 이때 Doppler 성분을 통해 객체의 특징을 구분하므로, 각 객체의 Doppler 정보를 분리하여 검출할 필요성이 있다. 이를 저장하기 위해 tracking 알고리즘을 사용한다 [6].

신호처리 결과를 바탕으로 각 객체를 분류하기 위해 CNN (convolution neural network)과 같은 딥러닝 알고리즘을 사용할 수 있다 [7],[8]. 그러나 CNN은 연산량이 많고 메모리 사용량이 크다는 단점이 있다. 이를 개선하기 위해 본 논문에서는 연산에 사용되는 각종 parameter를 이진화한 BNN (binarized neural network)를 사용하였다 [9]. 해당 방식은 연산의 복잡도와 메모리 사용량을 감소시켜 하드웨어 구현에 유리하다. 따라서, 본 논문에서는 CNN을 사용한 대부분의 기존 연구를 발전시켜 BNN을 하드웨어 가속기 형태로 구현한 드론 탐지 연구를 진행

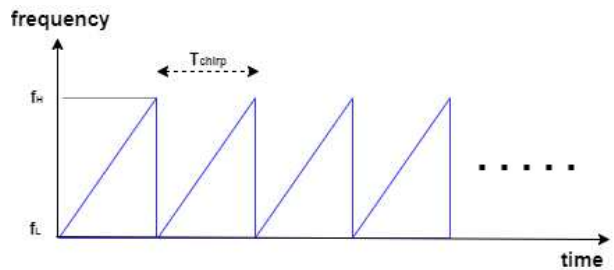


그림 1. FMCW 레이다 센서의 파형
Fig. 1. Waveforms from FMCW radar sensors.

하였다. 탐지된 각 객체의 정보를 추적하기 위해 tracking 알고리즘을 활용하였고, 객체의 움직임 특성을 BNN 모델에 학습시켜 분류를 수행하였다. Raspberry-Pi에서 레이다 센서 신호를 처리하는 소프트웨어를 실행하여 BNN 연산에 필요한 micro Doppler profile을 생성한다. 또한, 실시간 구현을 위해 BNN 연산을 FPGA (field programmable gate array) 기반의 하드웨어 가속기로 구현한다.

본 논문의 구성은 다음과 같다. II장에서는 제안된 시스템에 사용된 기술과 알고리즘에 관해 설명하고, III장에서는 시스템의 구조와 성능 평가 결과를 제시한다. IV장에서는 시스템 설계 결과에 관한 내용을 다루며, V장에서는 실험 결과를, VI장에서는 본 논문의 결론을 요약하며 마무리한다.

II. 배경

2-1 FMCW 레이다 센서

FMCW 레이다 센서는 그림 1과 같이 시간에 따라 송신 주파수를 연속적으로 변조하여 방사하는 방식으로 동작한다. 레이다 신호가 방사되면 주파수는 f_L 부터 f_H 까지 변조되며, 이 구간을 chirp이라고 한다. 한 chirp이 소요하는 시간을 T_{chirp} 이라 하며, chirp이 모여 하나의 frame을 구성한다. FMCW 레이다는 이러한 frame을 연속적으로 방사하여 객체까지의 거리와 속도를 측정한다.

2-2 FFT 알고리즘

FMCW 레이다 센서로부터 측정된 거리와 속도 정보는 FFT 알고리즘을 통해 검출할 수 있다. FFT 알고리즘은 푸리에 변환에 근거하여 함수의 근사값을 계산하는 알고리즘이다. FFT 알고리즘을 이용해서 거리와 속도를 구하는 방법에는 1D FFT 방식을 두 번 실행하는 방법과 2D FFT를 한 번 실행하는 방법이 있다. 이 중 2D FFT를 한 번 실행하는 방법은 많은 수행 시간이 필요하다는 단점이 있기 때문에, 본 논문에서는 1D FFT를 두 번 실행하는 방법을 채택하였다.

1D FFT를 두 번 실행하여 거리와 속도 정보를 얻는 방법은

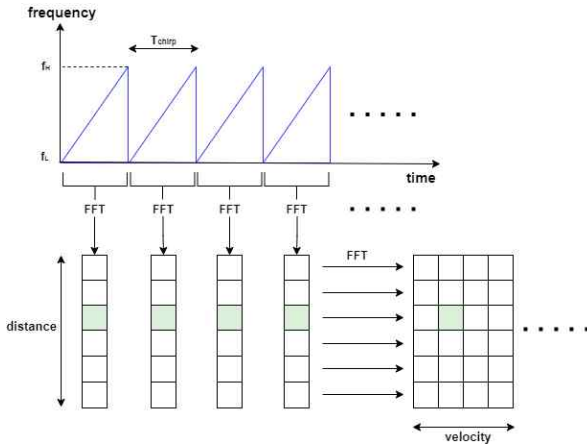


그림 2. FFT 알고리즘 구조
Fig. 2. FFT algorithm structure.

다음과 같다. 먼저 다수의 sample로 구성된 chirp 하나를 1D FFT 함으로써 거리 정보를 얻는다. 그 후 얻은 거리 정보를 frame 단위로 다시 한 번 1D FFT 하여 속도 정보를 도출한다. 이러한 방식으로 두 번의 1D FFT를 거치면 frame 당 R-D map (range doppler map)을 만들 수 있는데, 이 R-D map 상에 나타나는 객체를 탐지 및 추적하여 객체의 class를 구분하게 된다. 그림 2에는 이와 같은 과정이 나타나 있다.

2-3 CFAR 알고리즘

일반적인 객체 탐지 방식에서는 일정한 임계값을 설정하여 이를 초과하는 신호를 객체로 판단한다. 그러나 임계값이 고정된 경우 주위 잡음 신호나 객체 신호의 세기에 따라 오검출이 발생할 가능성이 있다. 따라서 레이더 센서 시스템에서는 임계값을 유동적으로 설정하여 일정한 오경보 확률을 유지하는 CFAR 알고리즘을 사용한다. CFAR는 target cell의 객체 여부를 판단하기 위해 주변 cell을 g_{cell} (guard cell) 및 r_{cell} (reference cell)로 구분하여 설정한다. r_{cell} 을 통해 임계값을 계산하고, 이를 target cell 값과 비교하여 해당 cell의 객체 여부를 판단한다.

레이더 센서 시스템에서 주로 사용되는 CFAR 알고리즘에는 CA-CFAR (cell average CFAR)와 OS-CFAR (ordered statistic CFAR) 등이 있다. CA-CFAR는 임계값 설정 시 r_{cell} 의 평균값을 이용하므로 구조가 간단하고 연산량이 적어 고속 연산이 가능하다. 그러나 다중 표적 환경이나 clutter 환경에서는 r_{cell} 의 평균값이 높아져 객체 탐지 확률이 낮아진다 [10]. 반면 OS-CFAR는 그림 3과 같이 r_{cell} 을 크기 순서대로 정렬한 후 k번째 cell 값을 이용하여 임계값을 설정한다. 널리 사용되는 OS-CFAR에서는 정렬된 전체 r_{cell} 의 3/4번째 셀을 k번째 cell로 선택한다 [11]. 해당 알고리즘은 정렬을 통해 기준을 설정하므로 연산량이 증가하는 단점이 있으나, 다중 표적이 존재하거나 불균일한 환경에서도 각 객체를 안정적으로 검출할 수 있다.

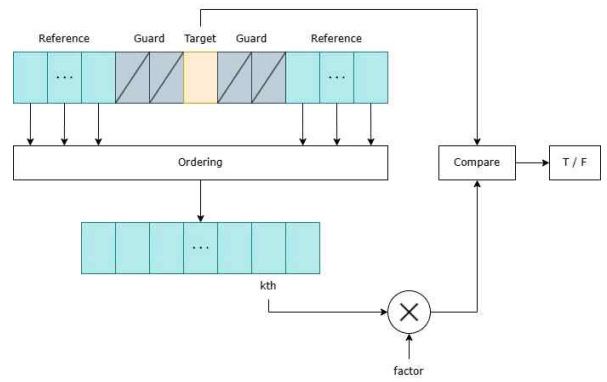


그림 3. OS-CFAR 알고리즘 구조
Fig. 3. OS-CFAR algorithm structure.

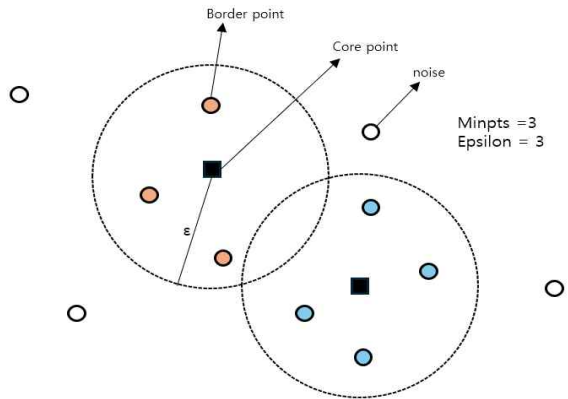


그림 4. DBSCAN 알고리즘의 구조
Fig. 4. DBSCAN algorithm structure.

다는 장점이 있다. 이에 따라 본 논문에서는 OS-CFAR 알고리즘을 사용하여 신호 처리 프로세서를 구현하였다.

2-4 DBSCAN 알고리즘

DBSCAN은 clustering 알고리즘 중 하나로 데이터를 target과 noise로 분류하는 알고리즘이다. 분류 과정은 CFAR에서 검출한 표적에 대한 데이터로 core point와 core point를 중심으로 하는 border point를 형성하고 core point를 중심으로 사전에 설정한 epsilon 범위 안에 border point의 개수가 minpts 이상일 경우, 유효한 core로 인정한다. 이때 minpts는 minimum points의 약어로, cluster로 인정받는 데 필요한 border point의 최소 개수를 의미한다. 유효한 core와 해당 core부터 epsilon 범위 내의 border point들을 묶어 하나의 cluster로 구분한다. 그리고 이렇게 묶이지 못한 point들을 noise로 인식한다 [12]. 이러한 분류 알고리즘은 다중 표적을 분류하고, 노이즈를 제거하는 데 적합하다.

2-5 Tracking 알고리즘

Tracking은 DBSCAN을 통해 얻은 cluster를 target으로 분류

Track Table							
NO.	1	2	3	4	5	6	7
Usage	Frame Number	Current Range	Current Speed	Predicted Range	Predicted Speed	State	Detection Count

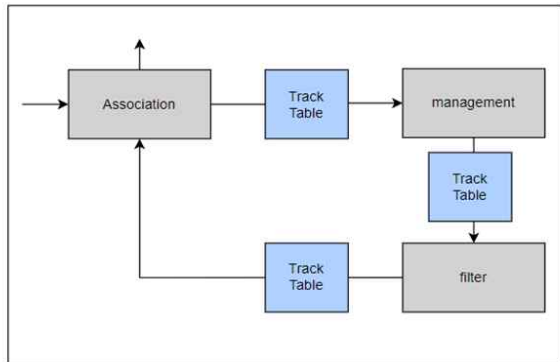


그림 5. Tracking 알고리즘의 구조
Fig. 5. Tracking algorithm structure.

하는 과정으로 그림 5와 같이 동작한다. Tracking은 association, management, filter 총 3단계로 구성되고 각각의 단계에서 track table을 갱신하고 모든 단계에서 해당 table을 공유한다.

Association 단계는 이전 프레임에서 예측한 거리, 속도 정보와 가장 인접한 값을 가지는 cluster를 현재 프레임의 거리, 속도 정보로 update한다. 이때 인접한 값을 찾는 기준에 임계 값을 두어 일정한 거리 내에 인접한 값이 존재하지 않는 경우 target이 해당 프레임에서 소실된 것으로 간주한다.

Management 단계는 association으로 갱신된 track table의 각 entry가 노이즈인지, 아니면 실제 객체인지를 분석하는 작업을 진행하는 단계이다. Track table에 새로운 데이터가 저장된 경우 해당 데이터가 연속해서 3번 이상 연속된 프레임에서 검출된 경우 해당 데이터를 target으로 인식하고 그전에는 noise로 인식한다. 그리고 노이즈로 인식되던 entry는 도중에 한 번만 소실되어도 바로 track table에서 삭제하는 반면 한번 target으로 인식한 entry는 연속된 프레임에서 3번 소실되지 않는 경우 entry를 지우지 않고 해당 data를 target으로 인식한다.

마지막으로 filter 단계의 경우 현재 거리와 속도 정보를 바탕으로 표적의 다음 프레임에서의 거리와 속도를 alpha-beta filter를 통해 예측하는 단계이다. Alpha-beta filter는 센서 데이터의 노이즈를 제거하고 객체의 위치와 속도를 추정하는 간단한 알고리즘으로 위치와 속도 예측에 오차를 반영하여 예측 정확도를 높이는 필터이다.

각 단계들을 통해 tracking은 프레임별로 정보를 관리하고 noise를 제거해 주며 multi target을 검출하는 역할을 수행한다.

2-6 BNN

CNN은 입력 데이터와 학습된 kernel을 convolution 연산하여 이미지의 특징을 추출한다. 이후 추출된 특징을 바탕으로

FCL (fully connected layer)을 통해 node 간의 연산으로 확률적 분류를 진행한다. BNN은 CNN의 모든 파라미터를 이진화하여 학습 및 추론을 진행하는 딥러닝 네트워크로, floating 연산을 bit-wise XNOR 게이트 연산과 popcount로 대체하여 연산량을 크게 감소시킬 수 있다 [13]. 이러한 특징은 실시간 사용이 필요한 임베디드 시스템에 적합하다.

III. 제안된 시스템 개요

3-1 제안된 시스템 구조

제안된 시스템 구조는 그림 6과 같다. 먼저 FMCW 레이더 센서(Demorad 레이더 센서)를 동작시키기 위해 레이더 센서와 키보드, 마우스를 Raspberry-Pi의 USB port를 통해 연결한다. 이후 키보드를 통해 명령어를 입력하고, UART (universal asynchronous receiver / transmitter) 통신을 사용해 Raspberry-Pi에서 FMCW 레이더로 command 신호를 송신한다. 신호를 수신한 FMCW 레이더는 raw 데이터 파일을 수집한 뒤, DC removal을 통해 잡은 성분을 제거하고 Raspberry-Pi로 송신한다.

FMCW 레이더로부터 수신된 raw 데이터 파일은 신호처리 과정을 거쳐 micro Doppler profile로 변환된다. 신호처리는 FFT, CFAR, DBSCAN, tracking 순으로 이루어진다. FFT에서는 거리와 속도를 처리하기 위해 복잡도가 높은 2D FFT 방식 대신 1D FFT 방식을 두 번 사용한다. 거리 정보는 chirp 단위로 FFT를 실행하여 검출하고, 속도 정보는 검출된 거리 정보를 기준으로 frame 단위의 FFT를 통해 검출한다. 이후 OS-CFAR를 통해 객체 여부를 판단한다. FFT와 CFAR 과정이 끝나면 DBSCAN과 tracking 과정이 진행된다. Cluster를 생성하고 신호의 거리와 세기에 따라 측정된 객체가 불필요한 객체인지 구분하고 추적하여, 필요한 객체의 동작을 확인한다. 이러한 확인

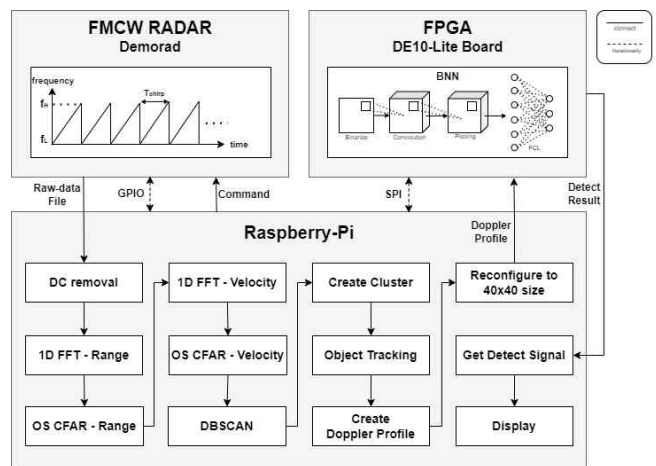


그림 6. 제안된 시스템 개요
Fig. 6. Overview of the proposed system.

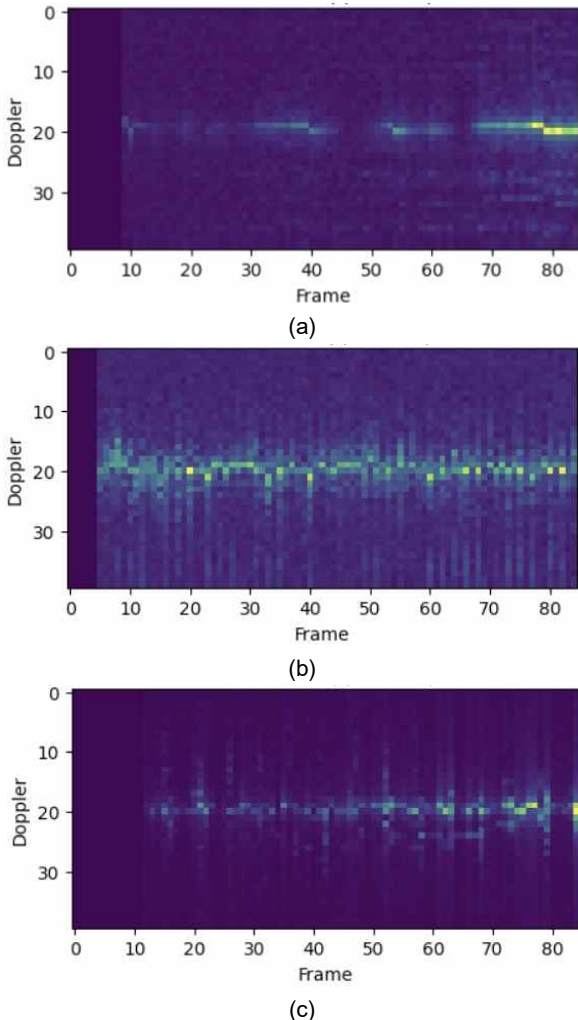


그림 7. (a) 드론, (b) 새, (c) 사람에 대한 micro Doppler profile
 Fig. 7. micro Doppler profile for (a) drone, (b) bird, (c) human.

과정은 micro Doppler profile을 생성하여 수행된다. 각 객체의 micro Doppler profile은 그림 7과 같다.

생성된 micro Doppler profile은 40x40의 크기로 재구성 후 이진화한다. 이후 SPI 통신으로 FPGA (DE10-lite board)로 전송하고, BNN 알고리즘을 통해 객체의 class를 특정한다. 특정된 결과는 SPI 통신을 통해 다시 Raspberry-Pi로 전송한다. 전송이 완료되면, Raspberry-Pi와 연결된 모니터를 통해 결과를 확인한다.

IV. 시스템 구조 설계 및 구현 결과

제안된 시스템을 검증하기 위해 Demorad 레이더를 사용하여 그림 8과 같은 환경을 구축하였다 [14]. Raspberry-Pi 4 Model B에서 소프트웨어를 동작하여 레이더 센서에서 수신한

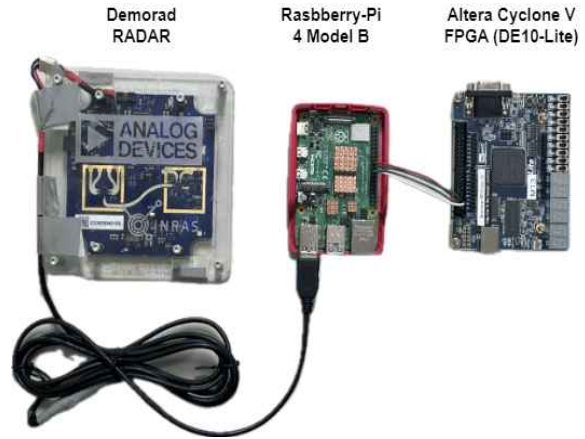


그림 8. 제안된 드론 검출 프로세서의 검증 환경
 Fig. 8. Verification environment for the proposed drone detection processor.

표 1. OS-CFAR 동작에 사용된 파라미터
 Table 1. Parameters used in OS-CFAR operations

Parameters	Value
r_{cell}	10
g_{cell}	2
P_{fa}	10^{-5}
$factor$	4

데이터를 처리하고 [15], 신호 전처리의 결과로 생성된 micro Doppler profile을 Altera Cyclone V FPGA (DE10-Lite) 상의 BNN 하드웨어 가속기로 전송하여 연산을 진행하고 class를 분류한다 [16]. 이후 Raspberry-Pi와 연결된 모니터에서 결과를 확인한다.

4-1 소프트웨어 설계 및 구현

소프트웨어는 레이더 센서 통신부, 신호 처리부, 결과 출력부로 구성된다. 레이더 센서 통신부는 크게 레이더 센서의 초기 parameter를 전달하는 부분과 센서에서 얻은 데이터를 받는 부분으로 이루어져 있다. 본 논문에서는 24GHz의 동작 주파수, 128 chirp/frame, 256 sample/chirp 구성으로 초기 parameter를 설정하였다. 신호 처리부는 수신한 데이터를 바탕으로 FFT, CFAR, DBSCAN, tracking 알고리즘을 수행한다. 이때 OS-CFAR 동작에 사용된 parameter는 표 1과 같다. 출력부는 micro Doppler profile과 class 분류 결과를 모니터에 출력한다.

4-2 BNN 성능 평가

사용한 BNN은 드론, 새, 사람의 총 3개 class를 구분한다. class 당 각 500개씩 총 1,500개의 데이터를 사용하되, 이 중

1,200개는 학습에, 나머지 300개는 평가에 사용하였다. 학습 과정에서는 cross entropy loss function과 ADAM (adaptive moment estimation) optimizer를 사용하였으며, learning rate는 0.02, epoch는 300, batch size는 20으로 설정하였다.

BNN은 FPGA 상에서 하드웨어 가속기 형태로 구현되므로 network의 정확도와 메모리 사용량을 고려하여 설계하였다. 표 2는 convolution layer와 FCL 구성에 따라 BNN의 정확도와 메모리 사용량을 측정된 결과이다. 이때 convolution layer 3개, FCL 1개의 조합이 가장 높은 정확도를 갖는 것을 확인하였다. 표 3은 depth에 따라 BNN의 정확도와 메모리 사용량을 측정된 결과이다. 따라서 본 연구에서는 89.33%의 정확도와 약 47.13KB의 메모리를 사용하는 BNN을 구현하였다.

최종 network 구성은 그림 9와 같다. 입력 데이터는 40×40 크기의 micro Doppler profile이며, convolution layer의 kernel은 3×3 크기를 사용하였다. 각 layer는 hard hyperbolic tangent activation function을 사용하였으며, batch normalization과 max pooling을 적용하였다.

표 2. BNN layer 구성에 따른 정확도 및 메모리 사용량
Table 2. Accuracy and memory usage based on BNN layer configuration.

# Conv	# FCL	Accuracy (%)	Parameter (KB)
2	1	74.33	43.25
	2	56.33	2134.77
3	1	89.33	47.13
	2	67.33	193.98
4	1	76.33	194.04
	2	67.67	210.36

표 3. BNN depth에 따른 정확도 및 메모리 사용량
Table 3. Accuracy and memory usage based on BNN depth.

Conv depth	Accuracy (%)	Parameter (KB)
128	37.33	28.23
256	89.33	47.13
512	84.67	167.89

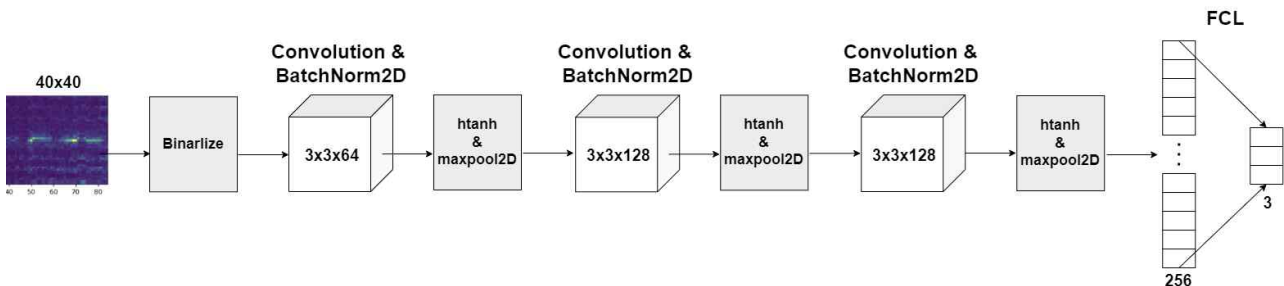


그림 9. 제안된 시스템의 BNN 구조
Fig. 9. BNN structure of the proposed system.

4-3 하드웨어 설계 및 구현

그림 10과 같이 하드웨어의 구성은 통신을 위한 SPI block과 class 분류를 위한 BNN block으로 나누어진다.

두 block은 dual port RAM으로 데이터를 공유한다. BNN block은 연산을 수행하는 BNN core, memory controller, 연산에 필요한 weight와 임계값을 저장하기 위한 ROM으로 구성하였다. BNN core에서는 BNN 연산을 위한 XNOR, popcount 연산기를 통해 CNN의 convolution 작업을 대체하였고, Batch normalization과 hard hyperbolic tangent는 ROM을 통해 저장한 임계값과의 비교기로 구현하였다. 또한, 모든 작업 사이에 플립 플롭을 배치하여 파이프라이닝을 구현함으로써 작업 처리 속도를 높였다. BNN core의 동작 과정은 그림 11과 같이 진행되었다.

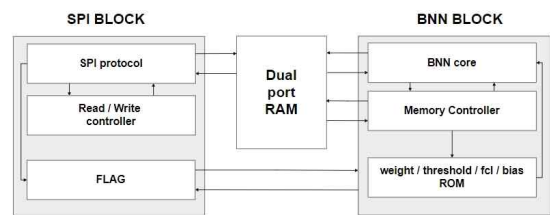


그림 10. 하드웨어 가속기의 SPI 통신 구조
Fig. 10. SPI communication structure of the hardware accelerator.

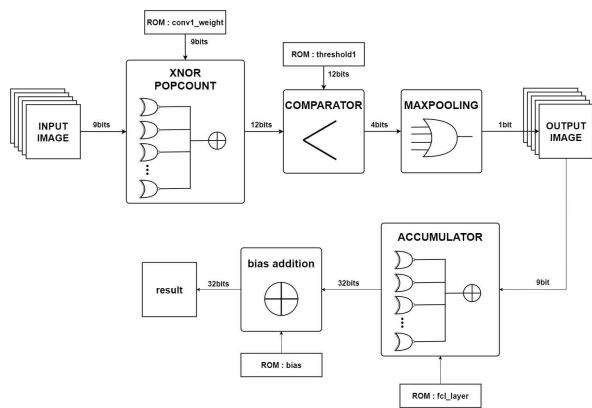


그림 11. BNN 연산을 수행하는 하드웨어 구조
Fig. 11. Hardware structure of performing BNN operations.

표 4. 하드웨어의 리소스 사용량

Table 4. Resource usage of the hardware.

Parameters	Value
Logic utilization	5,707
Total memory bits	426,500
Total registers	1,262

BNN 가속기는 Verilog HDL로 작성되었고 FPGA(DE10-lite board) 상에서 구현 및 검증이 진행되었다. 50 MHz의 동작 주파수에서 실행 시간은 소프트웨어 기반 대비 약 100배 가속한 4 ms로 측정되었고 하드웨어 설계에 사용된 resource의 사용량은 표 4와 같다.

V. 실험 결과

실험 환경은 그림 12와 같이 Demorad 센서를 지면으로부터 1 m 높이에 배치한 뒤, frame 당 0.1초로 총 10초 간 측정을 진행하였다. 드론과 사람은 동일하게 15 m 지점에서 5 m 지점까지 약 1 m/s의 속도로 센서와 가까워지는 방향으로 이동하였고, 새는 실제 새가 아닌 새 모양의 드론을 사용하였다. 이때 조작의 어려움으로 인해 불가피하게 5 m 지점에 고정시켜 측정하였다.

획득한 데이터에 전처리 과정을 거쳐 micro Doppler profile을 생성하였으며, 이를 BNN을 통해 드론, 사람, 새로 분류하였다. 분류 결과는 그림 13과 같이 터미널 창에 출력하였다. 이를 통해 움직이는 객체의 탐지 및 추적이 가능하고, 다중 객체의 분리 및 객체의 종류별 분류가 가능함을 확인할 수 있었다.



그림 12. 실험 환경 구성

Fig. 12. Setup of experiment environment.

```
team01@raspberrypi:~$ cd Desktop/demorad-master
team01@raspberrypi:~/Desktop/demorad-master$ source ~/miniforge3/bin/activate
(base) team01@raspberrypi:~/Desktop/demorad-master$ conda activate pytorch-env
(pytorch-env) team01@raspberrypi:~/Desktop/demorad-master$ python RPI_FPGA_SPI.py
Prediction: drone
done
(pytorch-env) team01@raspberrypi:~/Desktop/demorad-master$ ss
```

그림 13. 출력 터미널 창

Fig. 13. Output terminal window.

VI. 결 론

본 논문에서는 BNN을 활용한 드론 검출용 레이더 신호처리 시스템을 제안하였다. 드론, 새, 사람을 객체로 정의하여 micro Doppler profile을 추출하고, FFT, CFAR, DBSCAN 및 tracking 과정을 포함한 신호 전처리 알고리즘을 통해 각 객체를 인식하였다. 실험을 통해 획득한 총 1,500개의 데이터로 BNN 학습을 진행하였으며, 이를 하드웨어 가속기로 설계하여 연산 속도를 향상시켰다. 연산 시간을 측정한 결과, 하드웨어 가속기로 설계하기 전보다 연산 시간이 약 100배 감소하였다. 또한 confusion matrix를 사용하여 각 객체 분류 성능을 평가한 결과, 드론 84.67%, 새 98.43%, 사람 86.67%의 정확도로 분류하였다.

Acknowledgments

본 연구는 2024년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행되었으며 (No. 2017-0-00528), CAD tool은 IDEC에 의해 지원되었음.

References

- [1] C. Briese and L. Guenther, "Deep learning with semi-synthetic training images for detection of non-cooperative UAVs," in *2019 International Conference on Unmanned Aircraft Systems*, Atlanta: GA, pp. 981-988, Jun. 2019. DOI: 10.1109/ICUAS.2019.8797731.
- [2] M. Jung and S. Lee, "Object perception algorithm based on LiDAR for autonomous vehicle," in *2017 IEIE Fall Conference*, Incheon: Korea, pp. 639-642, Nov. 2017. Retrieved from <https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE07276399>.
- [3] C. J. Li and H. Ling, "An investigation on the radar signatures of small consumer drones," in *IEEE Antennas and Wireless Propagation Letters*, Vol. 16, pp. 649-652, Jul. 2017. DOI: 10.1109/LAWP.2016.2594766.
- [4] X. Wang, H. Liang and P. Wang, "Detection and tracking of UAVs using interferometric radar," in *2019 International Radar Conference*, Toulon: France, pp. 1-6, Sep. 2019. DOI: 10.1109/RADAR41533.2019.171312.
- [5] H. Rohling and M. Kronauge, "New radar waveform based on a chirp sequence," in *2014 International Radar Conference*, Lille: France, pp. 1-4, Oct. 2014. DOI: 10.1109/RADAR.2014.7060246.
- [6] E. Hyun and J.-H. Lee, "Multi-target tracking scheme using a track management table for automotive radar systems," in *2016 17th International Radar Symposium*, Krakow:

- Poland, pp.1-5, May. 2016. DOI: 10.1109/IRS.2016.7497283.
- [7] L. Wang, J. Tang and Q. Liao, "A study on radar target detection based on deep neural networks," in *IEEE Sensors Letters*, Vol. 3, No. 3, pp. 1-4, Mar. 2019. DOI: 10.1109/LSENS.2019.2896072.
- [8] J. H. Lee, Y. Yim, H. Choi, and Y. H. Jung, "Deep learning-based drone detection through sensor fusion technique of image sensor and FMCW radar," in *2020 IEIE Summer Conference*, Jeju: Korea, pp. 690-691, Aug. 2020. Retrieved from <https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE10447956>.
- [9] H. V. Quang, L. Kim, C. S. Hong, "A deep learning accelerator for a typical binary neural network layer," in *Korea Computer Congress 2021*, Jeju: Korea, pp.597-599, June 2021. Retrieved from <https://www.dbpia.co.kr/Journal/articleDetail?nodeId=NODE10583012>.
- [10] B. R. Mahafza, *Radar Systems Analysis and Design Using Matlab*, 2nd ed. New York, NY: CRC Press, 2005.
- [11] Y. -J. Kong, S. -K. Woo, S. Park, S. -Y. Shin, Y. H. Jang, E. Yang, "OS CFAR computation time reduction technique to apply radar system in real time," *The Journal of Korean Institute of Electromagnetic Engineering and Science*, Vol. 29, No. 10, pp. 791-798, Oct. 2018. DOI: <https://doi.org/10.5515/KJKIEES.2018.29.10.791>.
- [12] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, Portland: Oregon, pp. 226-231, Aug. 1996. Retrieved from <https://dl.acm.org/doi/10.5555/3001460.3001507>.
- [13] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: training deep neural networks with weights and activations constrained to +1 or -1," *arXiv*, Feb. 2016. DOI: <https://doi.org/10.48550/arXiv.1602.02830>.
- [14] Raspberry Pi. raspberry-pi-4-model-b [Internet]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>.
- [15] Analog Devices. EVAL-DEMORAD [Internet]. Available: <https://www.analog.com/en/resources/evaluation-hardware-and-software/evaluation-boards-kits/eval-demorad.html#e-b-overview>.
- [16] Terasic Technologies. DE10-Lite Board [Internet]. Available: <https://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=234&No=1021>.



김 홍 석 (Hong-Suk Kim)

2019년 3월 ~ 현재: 한국항공대학교 항공전자정보공학부
※관심분야: SoC 설계, 임베디드 시스템, 고속 연산 회로 설계



반 규 리 (Gyu-Ri Ban)

2021년 3월 ~ 현재: 한국항공대학교 항공전자정보공학부
※관심분야: 임베디드 시스템, SoC 설계, 디지털 신호 처리



서 지 훈 (Ji-Hun Seo)

2019년 3월 ~ 현재: 한국항공대학교 항공전자정보공학부
※관심분야: SoC 설계, 임베디드 시스템, 디지털 HW 설계, 고속 연산 회로 설계



정 운 호 (Yunho Jung)

1998년 2월: 연세대학교 전자공학과 (공학사),

2005년 2월: 연세대학교 전기전자공학과 (공학박사),

2007년 ~ 2008년: 연세대학교 전기전자공학과 연구교수

2008년 ~ 현재: 한국항공대학교 항공전자정보공학부 교수

※ 관심분야 : 무선통신시스템, 항공통신시스템, 레이더시스템, 인공지능시스템 SoC 설계

2000년 2월: 연세대학교 전기전자공학과 (공학석사)

2005년 ~ 2007년: 삼성전자 정보통신연구소 책임연구원