JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 24, No. 6, pp. 1285-1294, Jun. 2023

Check for updates

아두이노를 이용한 TinyML 기반의 음성인식 구현

구 금 서[1] · 서 영 건[2*]
[1]경상국립대학교 컴퓨터소프트웨어전공 강사
[2*]경상국립대학교 컴퓨터과학전공 교수

Implementing TinyML-based Speech Recognition using Arduino

Geum-Seo Koo[1] · Yeong Geon Seo[2*]

[1]Lecturer, Department of Computer Software, Gyeongsang Nat'l University, Jinju 52725, Korea
[2*]Professor, Department of Computer Science, Geyeongsang Nat'l University, Jinju 52828, Korea

[요 약]

방대한 데이터의 AI 처리는 주로 클라우드 기반의 서버에서 이루어지고 있다. IoT 및 다양한 센서 등에서 실시간으로 수집되는 데이터의 양이 폭발적으로 증가하고 있으므로 엣지 AI 기술로 클라우드나 서버가 아닌 로컬에서 AI 알고리즘을 처리하는 기술이 중요하다. 엣지 AI를 통해서 임베디드 디바이스에서 기계학습을 수행하면 높은 응답성과 비용 절감 등에 이점이 있다. 본 논문에서는 엣지 AI의 주요 구성요소인 TinyML기반의 Edge Impulse에서 음성을 학습시키고 그 결과를 Arduino Nano 33 BLE Sense에 업로드하여 서보모터 제어를 구현한다. 이를 통해서 자원 제약적인 임베디드 디바이스에서 기계학습을 처리하여 클라우드의 부하 없이 마이크로컨트롤러에서 다양한 AI 처리가 가능함을 보인다. TinyML은 클라우드 처리에 의존하지 않고 매우 낮은 전력 범위에서 작동하는 임베디드 디바이스에서 딥러닝 알고리즘을 수행하는 데 중점을 두고 있다. 앞으로 저전력 임베디드 디바이스에서 다양한 기계학습 활용이 가능할 것으로 기대한다.

[Abstract]

AI processing of massive data is mainly conducted on cloud-based servers. With increasing real-time data collected from IoT and various sensors, edge AI technology has been developed using edge computing to process AI algorithms locally rather than in the cloud or server. Machine learning on embedded devices through edge AI has advantages such as high responsiveness and cost reduction. In this paper, we learn voice in TinyML-based Edge Impulse, a major component of Edge AI, and upload the result to Arduino Nano 33 BLE Sense to implement servo motor control. Thus, machine learning is processed in resource-constrained embedded devices, enabling various AI processing in microcontrollers without cloud load. TinyML performs deep learning algorithms on embedded devices operating in the low power range without relying on cloud processing. In the future, various machine-learning techniques will be used in low-power embedded devices.

색인어 : TinyML, 기계 학습, 아두이노 나노 33 BLE 센스, 엣지 AI, 음성 인식

Keyword : TinyML, Machine Learning, Arduino Nano 33 BLE Sense, Edge AI, Speech Recognition

http://dx.doi.org/10.9728/dcs.2023.24.6.1285

Received 14 April 2023; **Revised** 10 May 2023
Accepted 23 May 2023

***Corresponding Author; Yeong Geon Seo**

Tel: +82-55-772-1392
E-mail: young@gnu.ac.kr

Ⅰ. 서 론

시장조사기관 IDC(International Data Corporation) IGIS에 따르면 인류가 2018년까지 축적한 디지털 데이터는 33ZB이며 2025년에는 약 79.4ZB가 축적될 것으로 예측하고 있다. 방대한 데이터의 처리를 위해서 고성능의 중앙 집중식의 클라우드 시스템을 위주로 AI 기술이 운영되고 있지만 최근 IoT 및 센서 등에서 실시간으로 습득되는 데이터의 폭발적인 증가로 클라우드 시스템 및 서버에 의존하기에는 한계가 존재한다. 엣지 AI[1],[2]는 엣지 컴퓨팅을 기반으로 임베디드 디바이스에서 인공지능 ML(Machine Learning)을 처리할 수 있다. 엣지 기반의 추론은 클라우드 서버로 데이터를 전송하지 않아도 되는 장점이 있으며 이를 통해 지연 시간을 줄이고 대역폭 비용을 절약할 수 있다. 또한, 서버 전송에 따른 프라이버시 문제가 발생하지 않는다. 엣지 기반 추론을 위한 TinyML(Tiny Machine Learning)[3]-[8]을 활용하여 마이크로컨트롤러와 같은 자원제약 환경에서도 ML을 처리할 수 있다. TinyML 도구로는 공개 소스 플랫폼인 TensorFlow[9]가 존재한다. 이에 더해 저전력 마이크로컨트롤러로 수십 킬로바이트 메모리를 사용하여 추론을 수행할 수 있다. TensorFlow Lite의 코어 런타임 바이너리 라이브러리는 Arm Cortex M3로 18KB 메모리 안에 들어간다. TensorFlow lite for microcontroller는 마이크로컨트롤러에 적합하게 변형한 것이다. Edge Impulse[10],[11]는 임베디드 마이크로컨트롤러 디바이스로 ML을 구축하는 플랫폼이다. 모델 유형으로 TensorFlow와 Keras[12]를 포함한다. 이러한 도구의 활용으로 신경망의 복잡성을 간소화하여 빠른 개발이 가능하다.

본 논문에서는 엣지 AI를 위해서 마이크로컨트롤러 보드인 아두이노에서 TinyML 기반의 음성인식 제어를 구현한다. TinyML을 통한 ML은 Edge Impulse 클라우드 플랫폼을 사용하여 설계하였다. NN(Neural Network) 신경망을 채택하여 음성("Open", "Close" 등)을 인식시키고, 학습 결과를 Arduino Nano 33 BLE Sense[13],[14]에 업로드 후 음성 제어로 SG90 서보 모터의 각도를 제어한다. 실행 결과의 확인은 아두이노 Serial Monitor에 측정값을 출력하고 Arduino Nano 33 BLE Sense의 LED_BUINTIN을 점등한다. 이로써 IoT 마이크로컨트롤러에서 ML이 원활하게 동작함을 확인하였으며, TinyML 기반의 자원제약 환경에서 다양한 활용이 가능함을 확인하였다.

Ⅱ. 관련 연구 및 구현 환경

IoT, 센서 등에서 발생하는 데이터는 기하급수적으로 늘어나고 있다. 클라우드 및 서버 기반의 처리는 실시간을 요구하는 환경에 한계가 존재한다. 또한 데이터 전송에 따른 트래픽과 중요한 정보가 서버로 전송됨으로써 개인 정보 등의 침해의 가능성도 존재한다. 이러한 이슈로 엣지 컴퓨팅이 부각되고 있다. 엣지 컴퓨팅의 장점은 지연 시간 감소, 비용 절감, 확장성, 신뢰성, 보안 등을 꼽는다. 최근 AI의 급 성장으로 엣지 컴퓨팅에 AI를 접목한 엣지 AI 또한 중요한 기술이다. 엣지 AI를 위해서 자원 제약적인 임베디드 디바이스에 ML의 적용하는 것은 쉽지 않지만 오픈소스 기반의 Tensorflow Lite 등으로 기계 학습을 수행하는 연구가 진행되고 있다. 추가로 마이크로컨트롤러 수준에서 ML을 수행하기 위해서 Tensorflow lite for microcontroller가 존재한다. 또한 TinyML 기반의 ML 구축 도구로 Edge Impulse 등이 있다. 다음 절에서는 TinyML과 Edge Impulse 및 Arduino Nano 33 BLE Sense를 순서대로 살펴본다.

2-1 TinyML

머신러닝은 신경망모델, 지도학습 등으로 대량의 데이터와 연산이 필요한 작업이다. 그러므로 성능이 뛰어난 GPU 기반에서 작업하거나 클라우드 시스템 위주로 운영되고 있다. 하지만 IoT 디바이스 등에서 수집되는 각종 데이터의 폭발적인 증가로 엣지 단계에서 ML을 처리하는 기술은 중요하다. TinyML은 임베디드 하드웨어의 낮은 성능과 저전력 마이크로컨터롤러에서 ML을 구현할 수 있도록 지원하는 기술이다. TinyML의 지원하는 도구는 TensorFlow Lite, Edge Impulse 등이 있다. ML은 방대한 데이터 셋, 다중 알고리즘 및 대규모 컴퓨팅 성능이 요구된다. ML의 학습 방식은 지도학습(Supervised Learning), 비지도 학습(Unsupervised Learning), 준지도학습(Semi-supervised Learning), 강화 학습(Reinforcement Learning)으로 나뉜다. 신경망의 대표적인 유형은 CNN(Convolution Neural Network)과 RNN(Recurrent Neural Network)이 있으며 신경망마다 연결 계층 (Fully connected layers)이 다르다. 지도학습은 신경망으로 대형의 데이터를 학습 시켜서 알고리즘이 결과를 추론하도록 한다. 이미지 및 음성 인식 작업용으로 CNN이 가장 적합하다. 클라우드 기반의 데이터 처리나 ML 알고리즘은 정확한 예측을 제공하지만 임베디드 장치에서 생성되는 대량의 데이터를 처리해야 하기 때문에 계산 부담이 높다. 또한 엣지에서 클라우드로 데이터를 전송함에 있어 병목현상(Bottleneck)이 발생한다. 그림 1은 TinyML의 워크플로를 보인다.

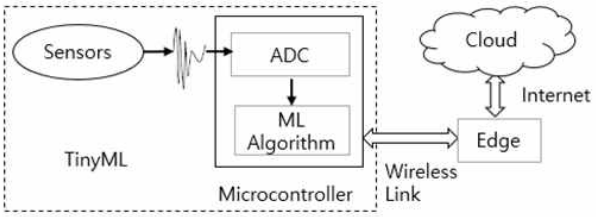

그림 1. TinyML 워크플로
Fig. 1. TinyML workflow

임베디드 장치에서의 ML 알고리즘의 처리는 클라우드 컴퓨팅의 단점을 보완할 수 있다.

본 연구에서는 TinyML 처리를 위한 도구로 Edge Impulse를 활용한다. 다음 절에서 Edge Impulse를 알아본다.

2-2 Edge Impulse

마이크로컨트롤러에서 ML 모델을 실행하면 데이터를 클라우드에 업로드할 필요가 없으므로 인터넷 연결 없이 실행이 가능하다. TinyML을 지원하는 도구인 Edge Impulse는 임베디드 시스템 및 TinyML 개발을 위한 클라우드 기반의 기계 학습 플랫폼이다. ML 모델 생성 시 프로세스를 단순화하여 ML 모델 학습, 테스트 및 배포까지 간단한 UI(User Interface)를 제공한다. 그림 2는 Edge Impulse에서의 머신러닝 워크플로우를 보인다. Edge Impulse로 데이터 수집에서 모델 교육 및 평가를 간소화할 수 있다.

Edge Impulse를 통해서 음성 인식을 위한 라이브러리를 타깃 보드에 맞게 배포하고 동작 확인을 위한 코드를 작성한다. 다음 절에서는 타깃보드로 지정한 Arduino Nano 33 BLE Sense를 살펴본다.

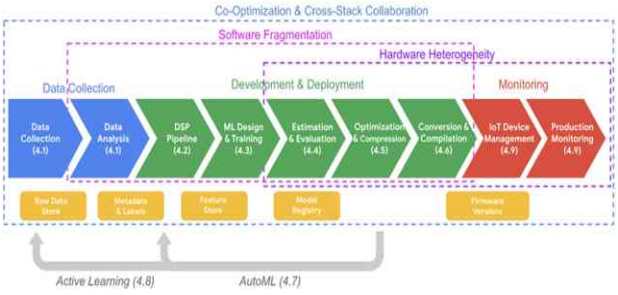

그림 2. 엣지 임펄스 작업절차
Fig. 2. Edge impulse workflow

2-3 Nano 33 BLE Sense

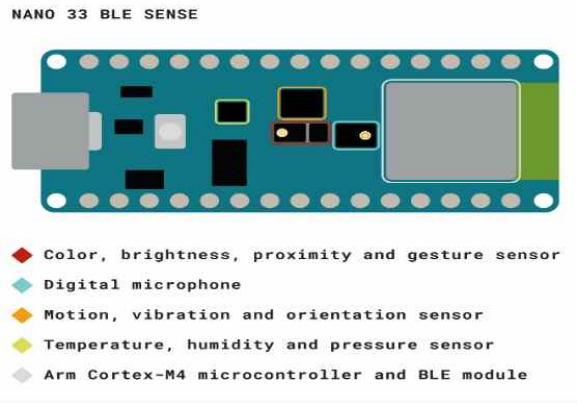

그림 3. 나노 33 BLE 센스
Fig. 3. Nano 33 BLE Sense

아두이노는 피지컬 컴퓨팅에 특화된 공개 소스 기반의 단일 보드 마이크로컨트롤러이다. 영국 라즈베리파이 재단의 Raspberry Pi와 더불어 대표적인 OSHW(Open Source Hardware) 중 하나이다. Atmel사의 AVR 마이크로컨트롤러를 장착하고 다양한 센서를 연결하여 주위 환경 변화를 감지할 수 있다. 개발목적에 따라 다양한 보드와 라이브러리가 제공되며 아두이노 개발환경(Arduino IDE, Integrated Development Environment)에서 상대적으로 쉽게 학습이 가능하다. 본 연구에서 사용하는 Arduino Nano 33 BLE Sense(이하 Nano 33 BLE Sense)는 Nano family의 한 종류이며 목적에 따라 Arduino Nano, Arduino Nano 33 IoT, Arduino Nano 33 BLE, Arduino Nano 33 BLE Sense 등 다양한 보드가 존재하므로 명확한 구분이 필요하다. Nano 33 BLE Sense는 Nordic Semiconductors의 nRF52840, 64MHz에서 실행되는 32비트 ARM® Cortex®-M4 CPU를 갖추고 있으며 1MB(nRF52840)의 CPU Flash Memory로 구성되어 있다. 표준에 해당하는 Arduino UNO R3(ATmega328P, 16MHz, 32KB) 대비 높아진 성능과 용량으로 ML(Machine Learning) 등 다양한 활용이 가능하다. Nano 33 BLE Sense 보드의 이름에서 알 수 있듯이 BLE는 Bluetooth Low Energy를 의미하며 저전력 환경에서 블루투스 통신이 가능하다. Sense는 다양한 센서를 갖추고 있음을 의미한다.

표 1. 아두이노 나노 33 BLE 센스 스펙
Table 1. Arduino Nano 33 BLE Sense spec

Arduino Nano 33 BLE Sense	
Microcontroller	nRF52840[15]
Operating Voltage	3.3V
Input Voltage(limit)	21V
DC Current per I/O Pin	15mA
Clock Speed	64MHz
CPU Flash Memory	1MB(nRF52840)
SRAM	256KB(nRF52840)
EEPROM	none
Digital Input / Output	14
PWM Pins	all digital pins
UART	1
SPI	1
I2C	1
Analog Input Pins	8 (ADC 12bit 200k samples)
Analog Output Pins	Only through PWM(no DAC)
External Interrupts	all digital pins
LED_BUILTIN	13
USB	Native in the nRF52840 Processor
IMU	LSM9DS1
Microphone	MP34DP05
Gesture, light, proximity	APDS9960
Barometric Pressure	LPS22HB
Temperature, humidity	HTS221
Length	45mm
Width	18mm
Weight	5gr

본 연구에서 음성 입력에 사용되는 센서는 그림 3의 파란색 부분(Digital microphone)에 해당하는 MP34DT05-A이며 무지향성 디지털 마이크로폰이다. 인식된 음성을 하드웨어에 직접 구현하므로 민감한 작업시 보드의 스펙과 데이터 시트는 중요하다. IMU(Inertial Measurement Unit) 센서 등의 위치는 그림 3의 색으로 구분 가능하며, 상세 스펙은 표 1과 같다. 센서별 자세한 내용은 각각의 데이터시트에서 확인할 수 있다. Nano 33 BLE Sense는 다양한 센서를 장착하고 있으므로 연구 주제별로 활용성이 높다.

다음 장에서 Edge Impulse를 기반으로 기계학습 절차와 구현 내용을 살펴본다.

Ⅲ. TinyML 기반의 음성인식

3장에서는 TinyML 기반의 음성인식 제어를 위한 설계와 구현을 다룬다. 먼저 전체 시스템의 입출력 흐름을 살펴보고 개발 환경과 연결 구조를 설명한다. 다음으로 기계학습에 사용한 Edge Impulse의 단계를 순차적으로 구현한다. 마지막으로 학습된 결과를 Nano 33 BLE Sense에 업로드하고 SG90을 제어하는 방법을 설명한다.

3-1 전체 시스템 구조

음성인식으로 모터를 제어하는 구조는 그림 4와 같다. 마이크로폰으로 "open", "close" 등의 음성이 입력되면 ML으로 학습된 음성을 인식하고 음성별로 인식률이 90% 이상인 경우에 서보모터(SG90)를 구동하는 구조이다. 그림 5의 개발 환경을 살펴보자. 타킷 보드에서 직접 개발이 불가능하므로 교차 개발 환경에서 구현한다. 호스트 영역은 Intel i9-9900KF 3.6GHz가 장착된 데스트탑이며, ML처리를 위해서 Edge Impulse와 연동하고 아두이노 통합 개발 환경(IDE)과 관련 라이브러리 등을 설치한다. Edge Impulse에서 음성 데이터를 취합 후 학습 시키고 학습된 결과에서 아두이노 보드와 모터 제어에 필요한 코드를 작성한다. 코드 작성이 완료되면 컴파일 후 보드에 업로드하고 음성에 맞춰 서보모터가 동작하는지 확인한다.

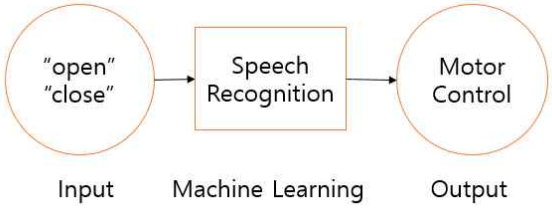

그림 4. 동작 순서
Fig. 4. Operation flow

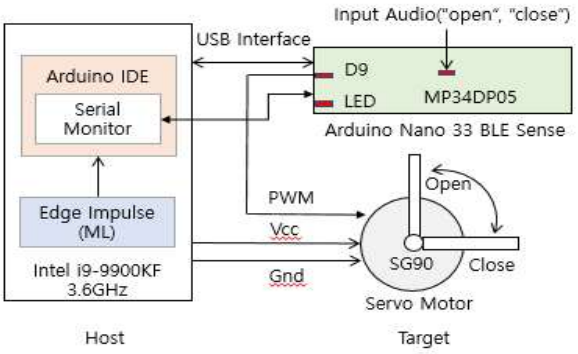

그림 5. TinyML 기반의 음성인식 개발 환경
Fig. 5. TinyML based speech recognition development environment

3-2 Edge Impulse를 활용한 기계 학습 구현

Edge Impulse는 TinyML 기반의 클라우드 플랫폼이다. 신경망 등 다양한 알고리즘을 선택하여 음성을 비롯한 이미지 등의 학습에 사용된다. ML을 위한 절차는 그림 6과 같이 크게 네 가지 부분으로 구분하여 설명한다.

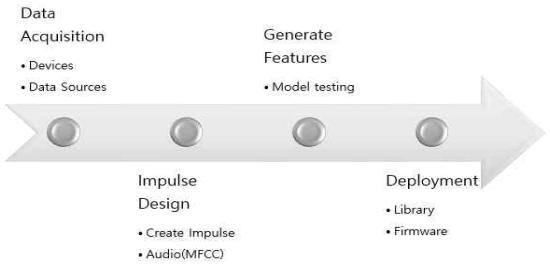

그림 6. ML 절차
Fig. 6. ML processing

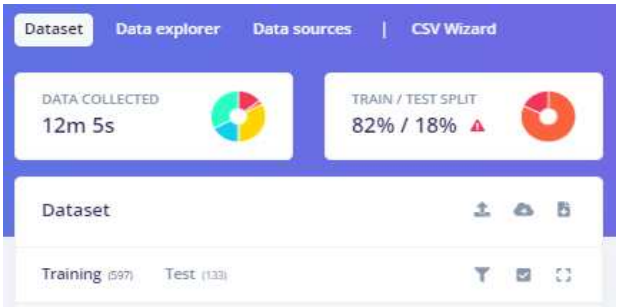

그림 7. 데이터 수집
Fig. 7. Data collection

1) Data Acquisition

먼저 Edge Impulse로 음성인식을 위한 프로젝트를 생성하고 'Devices'에서 데이터 업로드에 사용할 디바이스를 선택한다. 선택 가능한 디바이스는 QR코드를 활용하여 스마트

폰을 연결하거나, 작업 중인 데스크탑 혹은 타깃 디바이스를 직접 연결하여 데이터를 획득할 수 있다. Data Acquisition에서 수집된 데이터("Open", "Close", "Noise", "Unknown" 등)별로 레이블(Label)을 확인할 수 있다. 훈련과 테스트의 데이터 비율은 약 80대 20을 권장한다. 모델에서 과적합 부분은 테스트에서 성능 저하가 될 수 있으므로 매개변수나 추가 데이터를 수집한다. 그림 7은 수집된 데이터의 비율을 보인다. 훈련 데이터(Training) 597건(82%)과 테스트 데이터(Test) 133건(18%)이다. 세부적으로 open 105건, close 85건, nano 16건, noise 194건, unknown 194건 이다.

2) Impulse Design

Data Acquisition 이후 다음 단계로 Impulse design을 수행한다. 임펄스는 원시(raw) 데이터를 취득하고 신호처리를 사용하여 특성을 추출한 다음 학습 블록을 사용하여 새로운 데이터를 분류한다. 입력된 데이터에 대해 수행되는 작업을 정의하여 ML에 적합하도록 만들고 분류 알고리즘을 정의하는 학습 블록을 정의한다. 그림 8에서 Windows size는 분류별로 처리될 데이터의 크기를 조정한다. 설정값은 1,000ms이다. Window increase는 슬라이딩 윈도우를 증가시키는데 걸리는 시간이며 수집 데이터가 Windows size보다 크다면 검토가 필요하다. 설정값은 500ms이다. Frequency(Hz)는 데이터의 빈도를 설정하는 부분으로 다운샘플링 혹은 업 샘플링 한다. 기본 설정값으로 작업하면 타깃 보드에서 실행시 오류가 발생하므로 16,000Hz로 수정하였다. 프로세싱 블록은 Audio(MFCC), Audio(MFE), Spectrogram, Audio(Syntiant), Raw Data 등을 선택할 수 있다.

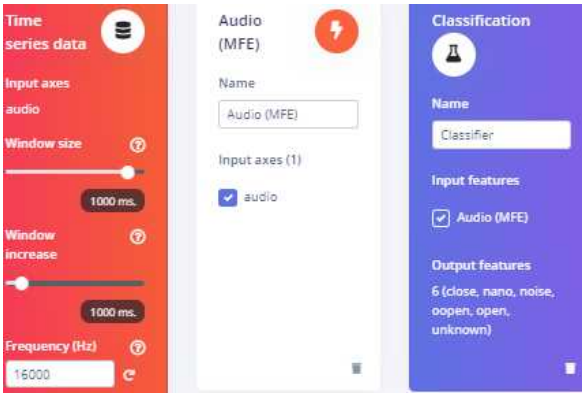

그림 8. 임펄스 디자인
Fig. 8. Impulse design

Audio(MFCC)는 사람의 목소리에 적합한 Mel Frequency Cepstral Coefficients를 사용하여 오디로 신호에서 특징을 추출한다. MFCC는 음성 인식에서 불필요한 정보는 버리고 중요 특징을 남긴 특성을 말한다. Audio(MFE)는 음성보다 오디오에 적합한 Mel-Filterbank 기능을 사용하여 오디오 신호에서 스펙트로그램(Spectrogram: 방출되는 소리 영역을 주요한 특징 중심으로 표시)을 추출한다. 오디오 혹은 센서 데이터에서 스펙트로그램을 추출하여 음성이 아닌 오디오 또는 연속 주파수가 있는 데이터에 사용한다. Audio(Syntiant)는 신티언트 전용 오디오 신호에서 로그 Mel-filterbank 에너지 특성 계산시 사용한다. 원시 데이터는 전처리 작업 없이 데이터를 사용한다. 딥 러닝을 사용하여 기능을 학습하는 경우에 적합하다. 학습 블록은 Classification, Regression, Transfer Learning, Anomaly Detection, Classification(Keras)로 구성되어 있다. Classification은 데이터에서 패턴을 학습하고 이를 새로운 데이터에 적용할 수 있다. 오디오 인식이나 움직임 분류에 적합하다. Regression는 연속된 숫자값 예측에 적합하며, Transfer Learning은 사전 훈련된 키워드 발견 모델을 조정한다. Anomoly Detection은 새로운 데이터에서 이상값을 찾는다. 본 연구에서는 Audio(MFE), Classfication을 적용하였다.

3) Generate Features & Model Test

특징 추출 단계에서는 원시 데이터에서 설정을 수정하고 특성을 추출한다. 분류 알고리즘에서 음성을 인식하는데 사용되는 속성이다. 그림 9는 "Close"에 대한 DSP 결과를 보인다. Mel-scale은 특정 피치(pitch)에서 발견한 음을 인지하는 기준(Threshold)을 반영한 scale 변환 함수이며, Mel-filterbank는 Mel-scale에서 리니어하게 구간을 N개로 나눠서 구현한 Triangular filter이다.

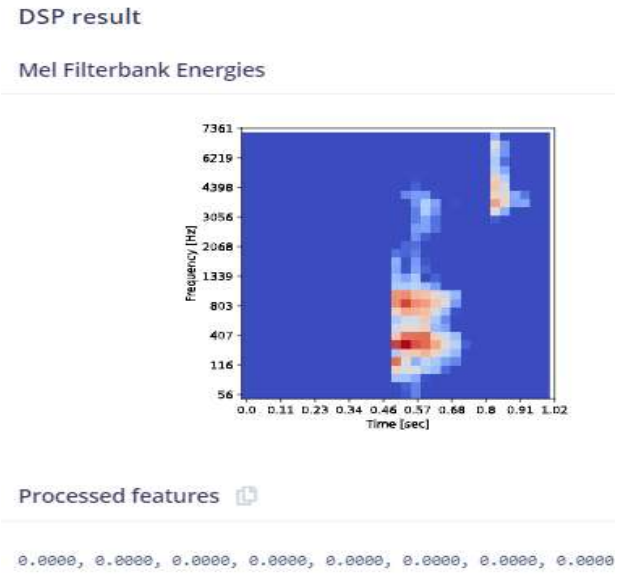

그림 9. "Close"의 DSP 결과
Fig. 9. DSP result of "Close"

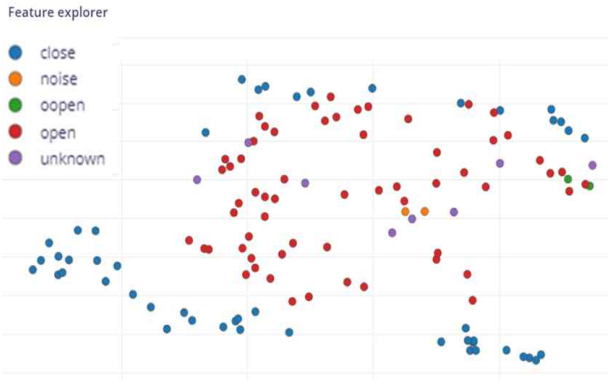

그림 10. 특성 탐색기
Fig. 10. Feature explorer

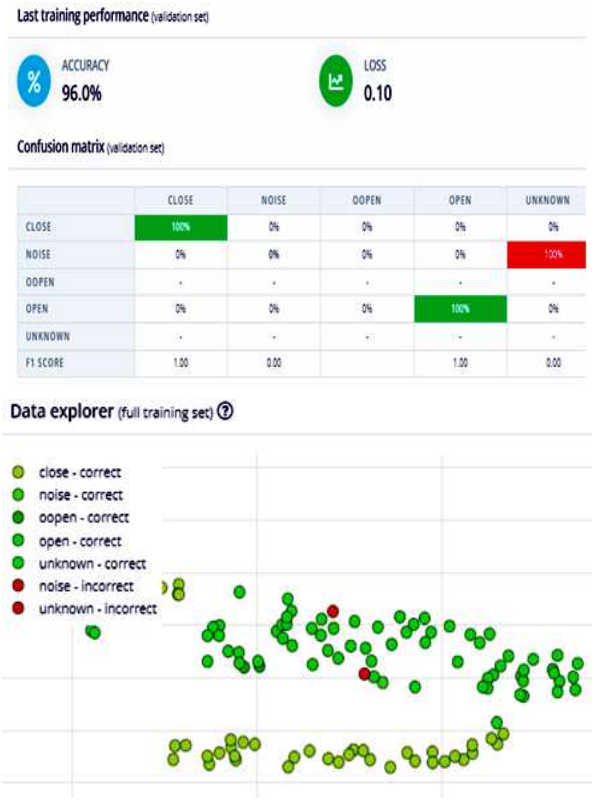

그림 11. 모델 성능
Fig. 11. Training performance

그림 10은 2D 기반의 음성 특성을 보인다. 그림 11은 학습의 결과로 정확성(Accuracy)은 96%, 손실(Loss)은 0.10이다. 녹색 부분("open", "close", "noise" 등)은 올바르게 분류된 항목이며, 빨간색은 일부 잘못 분류된 항목을 뜻한다. 학습 방법과 데이터 분류에 따라 정확도의 조율이 가능하다.

모델 학습 이후에 모델 테스트를 수행한다. 입력 모델이 충분하지 않음에도 일반적으로 결과물이 좋은 이유는 학습 데이터와 동일한 환경에서 수행되기 때문이며 입력 데이터의 변화가 발생하면 성능이 저하될 수 있다.

4) Deployment

학습된 ML 모델을 Nano 33 BLE Sense에 배포하는 방법은 크게 라이브러리와 펌웨어로 나뉜다. 또한 컴퓨터나 모바일 폰에서 ML 모델을 직접 수행할 수 있다. 본 연구에서는 Nano 33 BLE Sense를 타겟 보드로 지정하여 라이브러리를 배포하고 서보 모터를 음성으로 제어한다. 다음 장에서 Nano 33 BLE Sense의 구현 과정을 설명한다.

3-3 Arduino Nano 33 BLE Sense에서 구현

Edge Impulse에서 학습된 모델로 Nano 33 BLE Sense에서 음성 인식을 구현하는 절차는 다음과 같다. 먼저 학습된 결과를 Arduino Nano 33 BLE Sense에 맞게 Zip형태의 라이브러리로 빌드하여 아두이노 IDE에 추가하고, 음성 인식 신호에 반응하도록 ino 코드를 구현한다. 구현의 주 내용은 입력된 음성에 따른 인식값과 인식된 문구("Open", "Close")를 Arduino Serial Monitor에 출력한다. 동시에 "Open"으로 인식되었을 때 Nano 33 BLE Sense의 LED_BUILTIN을 "On" 처리하고 서보 모터를 0도에서 90도로 회전시킨다. 만약 "Close"로 인식되면 Nano 33 BLE Sense의 LED_BUILTIN을 "Off"로 처리하고 서보 모터를 90도에서 0도로 회전시킨다. Nano 33 BLE Sense의 LED_BUILTIN의 위치와 서보 모터 연결 번호 확인을 위해서 Nano 33 BLE Sense의 핀맵(그림 12)을 참고하여 D9번 핀에서 제어 신호를 출력하도록 작성하였다.

음성에 인식하는 서보모터는 SG90이다. SG90은 약 180도의 회전각을 가지며 별도의 모터 컨트롤러 구축없이 동작이 가능하다.

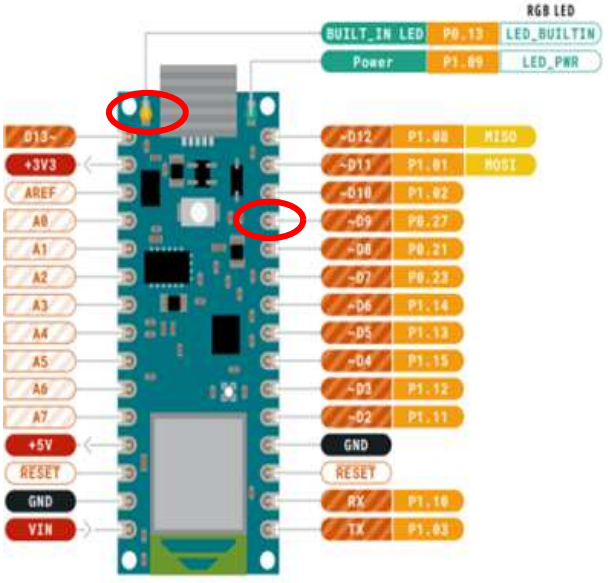

그림 12. 아두이노 Nano 33 BLE 센스(핀 맵)
Fig. 12. Arduino Nano 33 BLE Sense(pin map)

표 2. SG90 스펙
Table 2. SG90 specifications

Specifications	
Speed	0.1(sec)
Torque	2.5(kg-cm)
Weight	14.7(g)
Voltage	4.8 ~ 6(V)

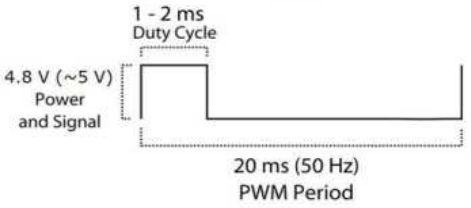

그림 13. SG90 PWM 주기
Fig. 13. SG90 PWM period

```
#include <PDM.h>
#include <vo-test_inferencing.h>
#include <Servo.h>

/** Audio buffers, pointers and selectors */
typedef struct {
    signed short *buffers[2];
    unsigned char buf_select;
    unsigned char buf_ready;
    unsigned int buf_count;
    unsigned int n_samples;
} inference_t;

static inference_t inference;
static bool record_ready = false;
static signed short *sampleBuffer;
static bool debug_nn = false; // Set this to true
static int print_results = -(EI_CLASSIFIER_SLICES_
Servo servo;
int open_voice;
int close_voice;
int noise_voice;
```

```
143  Servo servo;
144  int servoPin = 9;
145  void setup(){
146    servo.attach(servoPin);
147    if(open_voice>90){
148      int read_signal = 90;
149      int angle = map(read_signal, 0, 1023, 5, 170);
150      servo.write(angle);
151      Serial.println("open");
152      digitalWrite(LEDG, HIGH);
153      delay(100);
154  }
155
156   if(close_voice>90){
157     int read_signal = 0;
158      int angle = map(read_signal, 0, 1023, 5, 170);
159      servo.write(angle);
160      Serial.println("close");
161      digitalWrite(LEDG, HIGH);
162      delay(100);
```

그림 14. SG90 제어 코드
Fig. 14. SG90 control code

SG90은 1~2ms의 Duty Cycle을 가지며 5V에서 동작한다. SG90의 간략한 스펙은 표 2와 같으며 PWM 주기는 그림 13과 같다.

Nano 33 BLE Sense와 SG90의 DataSheet를 확인하면 기본 동작 전압은 각각 3.3V와 4.8~6V임을 알 수 있다. Nano 33 BLE Sense의 전원을 SG90을 직결 연결하면 동작하지 않으므로 Nano 33 BLE Sense의 5V 전압으로 변경하기 위해 점프선을 연결하거나 외부 전원을 연결하여 동작을 확인할 수 있다. 본 연구에서는 USB 5V를 SG90의 Red Line(Vcc)과 Brown Line(GND)으로 연결하여 전원을 공급하고 Nano 33 BLE Sense의 D9번 핀에서 전송된 PWM(Pulse Width Modulation) 제어 신호는 Orange Line으로 연결하였다. 그림 14는 작성한 코드의 일부이다. 크로스 컴파일에 사용된 데스크탑의 사양은 다음과 같다.

- Intel Core i9-9900KF 3.60GHz
- 32GB(2666MHz) Memory & 1TB SSD

MbedOS를 사용하는 Nano 33 BLE Sense의 특성으로 초기 컴파일에 대략 10분 이상이 소요되며, 사양에 따라 15~20분 이상 소요된다. 초기 컴파일 이후 재컴파일을 수행하면 수 분 내로 컴파일이 완료된다. 컴파일 이후 Nano 33 BLE Sense에 업로드를 완료한 결과는 그림 15와 같다.

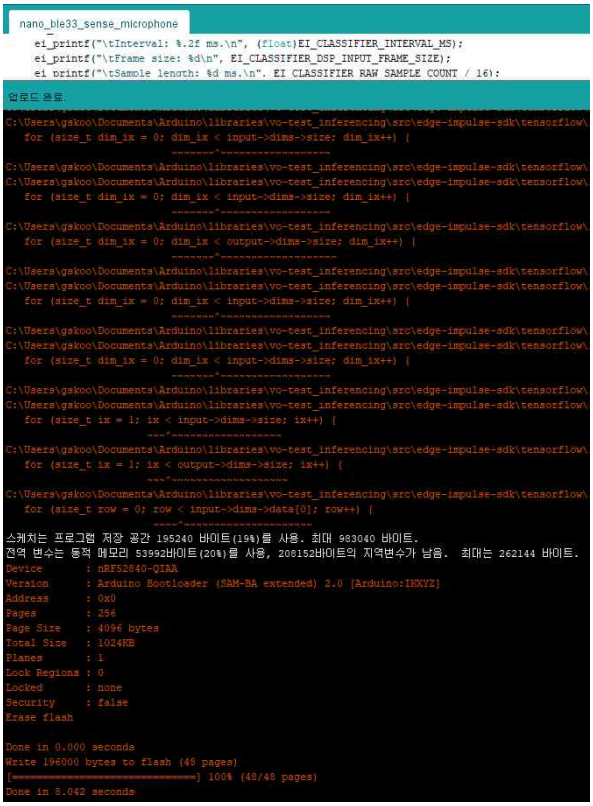

그림 15. 컴파일 & 업로드
Fig. 15. Compile & upload

Ⅳ. 실험 및 결과

4장에서는 Nano 33 BLE Sense에서 음성인식이 정상적으로 수행되는지 실험하고 인식률에 따른 결과를 보인다. 먼저 ML 이후 Classifier에 사용 가능한 최적화 값을 확인한다. 표 3과 같이 RAM 사용량은 9.6K 수준으로 극히 낮다. 지연시간도 53ms 수준이므로 적용 분야별로 해당 값을 참고하여 다양하게 ML을 적용할 수 있다.

다음으로 음성인식에 따른 동작을 확인한다. 먼저 "open" 음성을 입력하면 그림 16과 같이 타깃 보드에서 음성을 인식하여 Arduino Serial Monitor에서 음성별 인식률 값과 인식 문구인 "Open"을 출력(붉은색 동그라미로 표시)한다. 동시에 Nano 33 BLE Sense의 LED_BUILTIN의 LED를 "On"으로 출력하고 SG90의 모터 회전각을 초기 각 0도에서 90도로 제어하여 정상적으로 동작함을 확인(그림 17)하였다. 표 3은 Nano 33 BLE Sense에 빌드시 양자화된 값과 최적화가 되지 않았을 때의 값을 보인다. 양자화는 자원 제약적인 임베디드 환경에서 자원의 효율성을 위해서 불필요한 정보를 제거하여 데이터의 크기를 줄이는 것으로 신경망 모델에서 가중치와 함수의 파라미터 정밀도를 줄인다.

표 3. 양자화 값의 비교
Table 3. Comparison of quantized values

	Quantized(int8)	Unoptimized(float32)
RAM Usage	9.6K	13.9K
Latency	53ms	796ms
Flash Usage	65.3K	151.1K

```
Predictions (DSP: 19 ms., Classification: 40 ms., Anomaly: 0 ms.):
    close: 0.10156
    noise: 0.12109
    oopen: 0.17578
    open: 0.30469
    unknown: 0.30469
open : 99
Predictions (DSP: 19 ms., Classification: 40 ms., Anomaly: 0 ms.):
    close: 0.00000
    noise: 0.00000
    oopen: 0.00000
    open: 0.99609
    unknown: 0.00000
```

그림 16. "Open" 음성 인식
Fig. 16. "Open" speech recognition

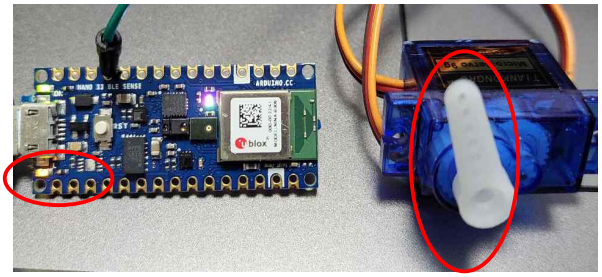

그림 17. "Open" 음성인식(서보모터 90°)
Fig. 17. "Open" speech recognition(Servo motor 90°)

신호처리에서 아날로그-디지털 변환(ADC: Analog to Digital Converter)과정에서 입력 신호를 샘플링하고 샘플링된 값들을 디지털 값으로 양자화하여 저장한다. 양자화는 손실 압축의 형태이므로 모델의 크기를 줄이면 성능이 저하될 수 있지만 ML의 정확도에 영향을 주지 않는 수준에서 처리된다. 만약 일부 성능에 문제가 발생한다면 데이터나 레이어를 추가하여 해결이 가능하다. Edge Impulse에서 제공하는 양자화 기법으로 타깃 보드에 최종 배포를 수행할 때 기존의 float32가 아닌 내부의 int8 표현을 사용하므로 대부분의 보드에서 메모리 사용량을 줄일 수 있으며 계산 속도 또한 빨라진다. 표 3의 양자화 전후 값을 비교해보면 램 사용량 측면에서 최적화 전(float32) 환경에서 13.9K의 크기가 양자화(int8) 이후 9.6K 크기로 4.3K가 줄어든 것을 확인 할 수 있다. 지연시간은 float32에서 796ms의 속도에서 양자화 이후 53ms으로 743ms가 줄어들었으며, 플래시 사용량도 기존 151.1K에서 65.3K로 85.8K가 줄어들었음을 확인할 수 있다.

다음으로 "Close" 음성을 입력하면 Arduino Serial Monitor에서 "Close" 인식률에 대한 값(0.99609)과 인식 문구인 "Close"을 그림 18과 같이 출력한다. 동시에 Nano 33 BLE Sense의 LED_BUILTIN의 LED를 "Off"로 처리하고 SG90 서보 모터의 회전각을 90도 상태에서 0도로 회전한다. 그림 19의 붉은색 동그라미에서 LED와 서보모터 동작을 확인할 수 있다.

```
Predictions (DSP: 19 ms., Classification: 40 ms., Anomaly: 0 ms.):
    close: 0.09766
    noise: 0.14062
    oopen: 0.16797
    open: 0.29688
    unknown: 0.29688
Close : 99
Predictions (DSP: 19 ms., Classification: 40 ms., Anomaly: 0 ms.):
    close: 0.99609
    noise: 0.00000
    oopen: 0.00000
    open: 0.00000
    unknown: 0.00000
```

그림 18. "Close" 음성 인식
Fig. 18. "Close" speech recognition

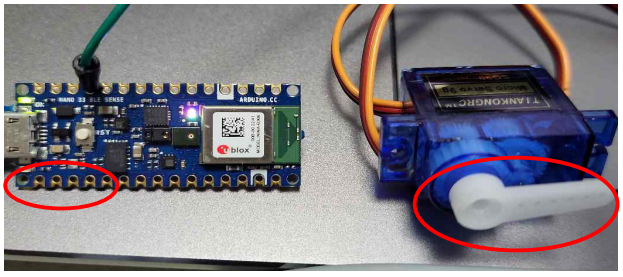

그림 19. "Close" 음성인식(SG90 0°)
Fig. 19. "Close" speech recognition(SG90 0°)

표 4. ML 알고리즘 처리 기술 비교
Table 4. Comparison of ML algorithm processing techniques

Technologies	Structure	Latency	Bandwidth Requirement	Computation Capability
Cloud	Fully Centralized	Very High	Very High	Very High
Edge/Fog	Not Centralized	Low	Low	High
MEC	Not Centralized	Low	Low	High
TinyML	May Operate Independently	Very Low	Very Low	Low

표 5. 타깃 보드의 성능 비교
Table 5. Performance comparison of target board

	Latency	Flash Usage	TinyML	Internet Connection
Nano 33 BLE Sense	53ms	65.3K	Yes	No
Raspberry Pi RP2040	128ms	90.6k	Yes	No
ST IoT Discovery Kit	44ms	90.6K	Yes	No
Texas Instruments TIDL-RT Library	53ms	90.6k	Yes	No
SiLabs Thunderboard Sense 2	247ms	90.6k	Yes	No
General Embedded Device	N/A	N/A	No	Yes

본 연구에서는 Edge AI의 주요 구성 요소인 TinyML기반의 Edge Impulse로 음성을 학습하고, 클라우드 등의 인터넷 연결 없이 학습된 키워드를 기반으로 자원제약 환경인 Arduino Nano 33 BLE Sense에서 독자적으로 음성인식을 수행하였다. 그 결과로 학습된 음성에 반응하여 Serial Monitor와 LED_BUILTIN 및 서보모터가 정상 동작함을 확인하였다. 표 4와 같이 TinyML을 기반으로 ML을 처리했을 때 인터넷 연결없이 독립적으로 동작 가능하며, 지연시간, 대역폭 요구사항 측면에서 Cloud나 Edge 및 MEC(Mobile Edge Computing)와 비교하여 우위에 있음을 확인할 수 있다. 다만 계산 능력은 자원 제약 환경의 MCU(Micro Controller Unit) 특성상 가장 낮지만 특정한 목적으로 수행되므로 크게 문제되지 않는다. 또한, 표 5과 같이 Nano 33 BLE Sense와 타 디바이스 플랫폼을 지연 시간, 플래쉬 사용량, TinyML 적용 여부, 인터넷 연결 등으로 나눠서 비교하였다. RAM 사용량은 ML을 적용한 대부분의 디바이스 플랫폼에서 9.6K 수준으로 비슷하였기에 표에서 제외하였다. 지연시간에서 Nano 33 BLE Sense의 경우 53ms로 짧은 지연시간을 가지고 있다. Raspberry Pi RP2040은 128ms, SiLabs Thunderboard Sense 2는 247ms이다. 동일 작업 조건에서 지연 시간은 ST IoT Discovery Kit나 Nano 33 BLE Sense가 우수하다. 플래쉬 사용량에서는 Nano 33 BLE Sense가 65.3K로 타 플랫폼의 90.6K 대비 적은 공간을 차지했다. 마지막으로 TinyML 기반으로 인터넷 연결 없이 ML을 독자적으로 처리할 수 있으므로 기존 임베디드 디바이스 대비 큰 장점을 가진다. 또한, 다양한 센서를 장착하고 있으므로 다양한 주제로 연구가 가능하다.

V. 결 론

빅데이터와 AI로 대표되는 시대적인 상황에서 기존의 클라우드 및 중앙 집중식 서버에서 데이터 처리는 한계가 존재한다 엣지 컴퓨팅 기술로 IoT 기기 혹은 임베디드 디바이스에서 데이터를 분산처리에 활용하고 있으며, 추가로 AI 기술을 적용한 엣지 AI가 부각되고 있다. TinyML은 매우 낮은 전력과 컴퓨팅 파워를 가지는 마이크로컨트롤러에서 ML을 처리하는 기술을 말한다. 공개 소스 기반의 Tensorflow lite for microcontroller로 ML을 처리할 수 있다. 또한, Edge Impulse 클라우드 플랫폼으로 ML을 위한 설계, 학습, 배포를 통합적으로 개발할 수 있다. 본 논문에서는 Edge Impulse로 음성 인식("open", "close" 등) 학습을 수행한 뒤 음성인식 상황과 성공률을 Arduino Serial Monitor에서 확인하는 코드를 작성하고 음성에 따라 서보모터를 동작하는 코드를 완성하였다. 타킷 보드는 피지컬 컴퓨팅에 많이 사용되는 Arduino Nano 33 BLE Sense를 선정하여 업로드하고 내장 LED_BUILTIN의 출력과 SG90의 동작 전원은 각 부품의 데이터 시트를 통해서 구현하였다. 음성 인식 모델의 훈련 데이터와 테스트 데이터의 비율은 82%와 18%이며 모델의 정확도는 96%이다. TinyML 기반의 학습은 기존의 클라우드와 엣지/포그 컴퓨팅 대비 매우 낮은 지연시간과 대역폭을 가지므로 이점을 가진다. 자원 제약을 가지는 보드이므로 계산 능력에서 일부 부족하지만 특정한 목적만을 수행하는 임베디드 시스템의 특성상 문제되지 않는다. TinyML을 지원하는 타 보드와의 비교에서 지연 시간과 플래쉬 사용량에서 우수함을 확인하였으며 ML 처리를 위해서 클라우드 등에 접속해야하는 기존의 임베디드 시스템 대비 장점을 가진다. 초저전력 마이크로컨트롤러에 ML 처리가 가능하므로 차후에 다양한 응용이 가능하다.

향후 연구과제로 다양한 마이크로컨트롤러 개발 보드와 플랫폼으로 ML 결과를 비교 분석하고, 문제점을 개선해 보고자 한다. TinyML을 기반으로 클라우드 처리에 의존하지 않고 엣지 추론을 수행하는 연구가 활발하게 진행되면 AI 처리에 새로운 변화가 많이 생길 것으로 기대한다.

참고문헌

[1] E. Li, L. Zeng, Z. Zhou, and X. Chen, “Edge AI: On-Demand Accelerating Deep Neural Network Inference via Edge Computing,” *IEEE Transactions on Wireless*

Communications, Vol. 19, No. 1, pp. 447-457, January 2019. https://doi.org/10.1109/TWC.2019.2946140

[2] I. Choi, Edge AI-based Power Saving for Embedded Devices, Ph.D. Dissertation, Korea University, February 2023.

[3] H. Han and J. Siebert, "TinyML: A Systematic Review and Synthesis of Existing Research," *2022 International Conference on Artificial Intelligence in Information and Communication (ICAIIC),* Jeju, Korea, pp. 269-274, February 2022. https://doi.org/10.1109/ICAIIC54071.2022.9722636

[4] P. P. Ray, "A Review on TinyML: State-of-the-art and Prospects," *The Journal of King Saud University Computer and Information Science*, Vol. 34, No. 4, pp. 1595-1623, April 2022. https://doi.org/10.1016/j.jksuci.2021.11.019

[5] L. Dutta and S. Bharali, "TinyML Meets IoT: A Comprehensive Survey," *Internet of Things*, Vol. 16, December 2021. https://doi.org/10.1016/j.iot.2021.100461

[6] J. Kwon and D. Park, "Hardware/Software Co-Design for TinyML Voice-Recognition Application on Resource Frugal Edge Devices," *Applied Sciences*, Vol. 11, No. 22, 11073, November 2021. https://doi.org/10.3390/app112211073

[7] S. Soro, "TinyML for Ubiquitous Edge AI," arXiv Preprint, February 2021. https://doi.org/10.48550/arXiv.2102.01255

[8] V. J. Reddi, B. Plancher, S. Kennedy, L. Moroney, P. Warden, L. Suzuki, ... and D. Tingley, "Widening Access to Applied Machine Learning with TinyML," *Harvard Data Science Review*, Vol. 4, No. 1, January 2022. https://doi.org/10.1162/99608f92.762d171a

[9] Tensorflow. Tensorflow Lite for Microcontroller [Internet]. Available: https://www.tensorflow.org/lite/microcontrollers

[10] S. Hymel, C. Banbury, D. Situnayake, A. Elium, C. Ward, M. Kelcey, ... and V. J. Reddi, "Edge Impulse: An MLOps Platform for Tiny Machine Learning," arXiv: 2212.03332, November 2022. https://doi.org/10.48550/arXiv.2212.03332

[11] S. H. Hong, "Edge Impulse Machine Learning for Embedded System Design," *Journal of Korea Society of Digital Industry and Information Management*, Vol. 17, No. 3, pp. 9-15, September 2021. https://doi.org/10.17662/ksdim.2021.17.3.009

[12] Keras [Internet]. Available: https://keras.io/.

[13] Arduino [Internet]. Available: https://www.arduino.cc/.

[14] Arduino, Arduino Nano 33 BLE Sense, Arduino Product Reference Manual, SKU: ABX00031, Spr, 2023.

[15] NORDIC, nRF52840 Product Specification v1.1, Nordic Semiconductor, 4431_417 v1.1, 02, 2019.

구금서(Geum-Seo Koo)

2001년 : 경상대학교 컴퓨터과학과 (이학사)
2003년 : 경상대학교 컴퓨터과학과 (공학석사)
2005년 : 경상대학교 컴퓨터과학과 (박사수료)
2005년～현 재: 경상국립대학교 강사
※관심분야 : Edge AI, Machine Learning, Embedded System, Physical Computing

서영건(Yeong Geon Seo)

1987년 : 경상대학교 전산과(이학사)
1997년 : 숭실대학교 전산과(공학박사)
1989년～1992년: 삼보컴퓨터
1997년～현 재: 경상국립대학교 컴퓨터과학과 교수
2022년～현 재: 경상국립대학교 정보전산처장
※관심분야 : PBL, 컴퓨팅 사고, 의료 영상 처리, SLAM, 영상 인식, 컴퓨터 네트워크