JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 24, No. 6, pp. 1327-1338, Jun. 2023

Check for updates

엣지 임펄스를 이용한 음성 처리 기법의 성능 분석

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

Performance Analysis of Speech Processing Techniques Using Edge Impulse

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, Gyeongsang Nat'l University, Jinju 52828, Korea

[요 약]

엣지 컴퓨팅 환경에서 음성 신호의 분석은 실시간 환경에서 중요하다. 기존의 클라우드 기반 시스템에서 음성 처리는 데이터 전송 및 처리 지연 문제가 발생한다. 엣지 임펄스는 저전력 임베디드 환경에서 기계 학습 및 신호 처리를 위한 오픈 소스 플랫폼으로 음성을 로컬에서 처리하여 데이터 보안과 개인 정보 보호 측면에서 우수하다. 본 논문에서는 엣지 임펄스를 사용하여 음성 데이터를 수집하고 신호 처리 및 기계 학습을 적용 후 음성의 특성 추출과 분류를 수행하여 성능을 분석하였다. 데이터 세트에 차원 축소 기법을 각각 적용하고, 음성 처리는 MFE와 MFCC를 적용하여 각각의 파라미터 값과 특성 등을 확인하고 EON 튜너를 적용하여 최적의 모델을 선정하였다. 모델 테스팅에서 MFE의 정확도는 86.82%, MFCC의 지연시간은 26ms를 보였다. 이결과로 음성 특성 추출 단계에서 모델별 성능을 확인하여 개발 목적에 맞는 최적의 모델을 선택할 수 있다.

[Abstract]

Speech signal analysis is important in a real-time edge computing environment. In conventional cloud-based systems, voice processing faces data transmission and processing delay problems. Edge Impulse is an open-source machine learning and signal processing platform in a low-power embedded environment. Moreover, it offers excellent data security and privacy protection by processing voice locally. In this study, speech data was collected using edge impulse, and performance was analyzed by performing speech feature extraction and classification after applying signal processing and machine learning. Different dimension reduction technique was applied to the data set; MFE and MFCC were applied for speech processing. Each parameter value and characteristic were checked, and the optimal model was selected using an EON tuner. In model testing, the accuracy of MFE was 86.82%, and the delay time of MFCC was 26 ms. Thus, selecting an optimal model for development was possible by checking each model's performance during speech feature extraction.

색인어 : 엣지 임펄스, 음성 처리 기법, MFE, MFCC, 임베디드 시스템

Keyword : Edge Impulse, Speech Processing Techniques, MFE, MFCC, Embedded System

http://dx.doi.org/10.9728/dcs.2023.24.6.1327

Received 26 April 2023; **Revised** 11 May 2023
Accepted 18 May 2023

***Corresponding Author; Yeong Geon Seo**

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

Ⅰ. 서 론

엣지 컴퓨팅[1],[2] 환경에서 음성 신호의 처리는 실시간 환경에서 개인 정보 등을 다루므로 중요하다. 음성 신호의 처리는 방대한 계산이 필요하므로 주로 클라우드 기반의 시스템을 사용하며, 이로 인해 데이터 전송 및 처리 지연 문제가 발생한다. 엣지 임펄스[3]-[5]는 저전력 임베디드 디바이스에서 음성 신호 처리 및 기계 학습 알고리즘을 개발하기 위한 TinyML(Machine Learning)[6],[7] 기반의 오픈 소스 플랫폼이다. 음성 처리를 디바이스에서 처리하므로 데이터 보안과 개인 정보 보호 측면에서 우수하다. 엣지 임펄스의 활용으로 기계 학습을 통한 음성 인식을 간소화하여 빠른 개발이 가능하다.

본 논문에서는 엣지 임펄스를 이용하여 음성 처리 기법의 성능을 비교 분석한다. 먼저 음성인식을 위해서 데이터 세트를 구성하고 입력된 데이터를 2차원 형태로 표현하기 위해서 차원 감소 기법인 t-SNE[8]와 PCA[9]를 각각 적용하여 비교하였다. 다음으로 입력 블록에서 시계열 데이터를 설정하고 처리 블록에서 음성 신호나 음악 신호에서 다양한 특성을 추출하는 기술 MFE[10]와 오디오 신호에서 사람의 청각 시스템에 적합한 특성을 추출하는 MFCC[11]-[14]를 각각 적용하여 DSP 결과와 특성을 비교한다. 학습 블록에서는 분류를 적용하여 각각의 신경망 설정과 구조를 확인하고 모델의 성능을 확인한다. 모델 테스트 전 단계에서 EON 튜너를 적용하여 가장 높은 정확도와 지연 시간이 빠른 모델을 기준으로 선정한 후에 DSP 적용과 NN의 성능을 확인한다. 최종적으로 모델 테스팅 이후 정확도와 해당 모델 기준으로 타깃 배포 시 정확도, 지연시간, 램 사용량, 플래시 사용량을 확인한다. 자원 제약적인 임베디드 환경에서 음성 인식을 위한 ML을 구현 시 음성 특성 분석과 모델별 특성을 파악할 수 있다.

Ⅱ. 관련 연구 및 구현 환경

음성 처리를 위해서 입력된 데이터 세트를 2차원 형태로 표현하는 차원 축소 기법인 t-SNE와 PCA를 살펴보고, 임펄스 디자인의 처리 블록에 해당하는 MFE와 MFCC의 설명한다. 마지막으로 음성 처리에 사용되는 도구인 엣지 임펄스와 기반 기술인 TinyML에 대해서 순서대로 살펴본다.

2-1 t-SNE & PCA

차원 축소 기법(Dimensionality reduction technique)은 데이터의 복잡성을 줄여서 데이터 품질을 향상시킨다. 대표적인 알고리즘 중 하나인 t-SNE(t-distributed Stochastic Neighbor Embedding)는 높은 차원에 데이터를 2차원으로 차원을 축소하는 비선형 기법이다. 낮은 차원의 시각화에 사용하여 데이터 구조의 이해에 도움을 준다. t-SNE를 사용하면 높은 차원의 공간에서 비슷한 데이터 구조는 낮은 차원 공간에서 가깝게 대응하며 멀리 떨어지거나 비슷하지 않은 데이터 구조는 멀리 위치하도록 포함한다. PCA(Principal Component Analysis)는 데이터의 차원을 축소하는 비지도 학습 기법이다. 데이터의 특성을 추출하여 새로운 축으로 변환한다. PCA는 데이터의 상관관계를 이해하고 새로운 변수를 생성하여 데이터를 표현한다.

2-2 MFE

MFE(Multi-Feature Extraction)는 음성 신호나 음악 신호에서 다양한 특성을 추출하는 기술로 주로 음성 인식, 음질 향상, 음성 감정 인식 등 다양한 분야에서 사용된다. 음성 인식을 위해서 음성신호에서 말하는 사람의 목소리를 분리해내고, 각 단어의 발음을 구별할 수 있는 특성을 추출해야 한다. 이를 위해서는 주파수, 에너지, 스펙트럼 등의 특성을 추출할 수 있어야 한다. MFE는 이러한 다양한 특성들을 추출하기 위해서 FFT(Fast Fourier Transform)를 이용해 주파수 스펙트럼을 분석하거나, MFCC(Mel-Frequency Cepstral Coefficients)를 이용해 음성 신호의 주파수 특성을 분석한다. MFE는 신호 처리, 딥러닝, 패턴인식 등의 분야와도 밀접한 관련이 있으며, 최근에는 신경망(Neural network)과 결합하여 더욱 정확하고 다양한 특성 추출이 가능한 기술들도 개발되고 있다.

2-3 MFCC

MFCC(Mel-Frequency Cepstral Coefficients)는 오디오 신호에서 효과적으로 특성을 추출하여 사람의 청각 시스템에 가장 적합한 멜-스케일(Mel-scale)로 변환하고, 이를 통해 신호를 일련의 주파수 대역으로 분할한 후, 각 대역에서 얻은 캡스트럴 계수(Cepstral Coefficients)를 계산하는 기법이다. 주로 음성 인식과 합성, 화자 인식 등의 처리에 사용된다. MFCC는 음성신호를 프레임별로 나누어 FFT(Fast Fourier Transform)를 적용하여 스펙트럼을 구하고 멜-필터 뱅크(Mel-Filter Bank)를 거쳐 멜 스펙트럼을 구한다. 멜 스펙트럼에서 캡스트럴(Cepstral) 분석을 적용하여 MFCC를 구한다. 고속 푸리에 변환(FFT)은 신호를 주파수 성분으로 변환하는 알고리즘으로 이산 푸리에 변환(DFT)를 더욱 빠르게 수행할 수 있도록 최적화한 알고리즘을 말한다. MFCC 추출 프로세스는 먼저 오디오 신호를 작은 프레임(20ms ~ 40ms)으로 분할하여 각 프레임에서 STFT(Short-Time Fourier Transform)을 계산한다. 계산값은 프레임 내 신호의 주파수 성분을 나타내는 복소수 형태의 값이다. STFT의 결과를 멜-필터 뱅크를 적용하여 멜-스케일로 변환한다. 멜-스케일은 사람의 청각 시스템에서 인식하는 주파수 스케일과 비슷하다. 멜-스케일로 변환한 값에 로그를 적용 후

DCT(Discrete Cosine Transform)을 적용하여 켑스트럴 계수(Cepstral Coefficients)를 계산한다. MFCC를 사용하여 음성 신호에서 특성을 추출하면, 각 프레임마다 일련의 켑스트럼 계수 벡터가 생성되며, 이 벡터는 음성 신호의 주파수 특성을 캡처하고, 다양한 음성 분류, 인식 등에 사용될 수 있다. 멜-스케일은 음성 신호에서 주파수를 표현하는 비선형적인 스케일로 사람의 청각 시스템에서 주파수를 인식하는 방식을 모방한 것이다. 사람의 달팽이관은 주파수가 높은 대역에서는 잘 인지하지 못하고 낮은 대역에서는 잘 감지하는 특성을 반영하여 물리적인 주파수와 실제 사람이 인식하는 주파수의 관계를 표현한 것이 멜-스케일이다. 멜-스케일은 주파수를 Hz 단위가 아닌 Mel 단위로 표현하며, Mel 단위의 주파수를 변환하는 공식은 수식(1)과 같다.

$$Mel(f) = 2595\log\left(1 + \frac{f}{700}\right) \quad (1)$$

여기서, Mel은 Mel 단위의 주파수를 나타내며, f는 Hz 단위의 주파수를 나타낸다. 또한 700은 1,000 Hz 주파수가 멜-스케일에서 약 100 Mel에 해당하는 것을 나타내는 상수이다. MFCC와 같은 기술에서는 Mel 스케일로 변환된 주파수 범위에서 필터 뱅크를 만들어 오디오 신호의 중요한 주파수 대역을 더 잘 파악할 수 있으므로 음성 인식과 합성 등의 작업을 수행할 수 있다.

2-4 Edge Impulse

엣지 임펄스는 딥러닝 모델을 구축하고 배포하기 위한 TinyML 기반의 클라우드 플랫폼이다. 그림 1과 같이 데이터 수집 임펄스 디자인, EON 튜너, 모델 테스팅, 배포까지 제공한다. 데이터 수집은 기존 음성 데이터나 타깃 디바이스에서 직접 수집할 수 있으며 비율은 훈련 데이터 80%와 테스트 데이터 20%를 유지한다.

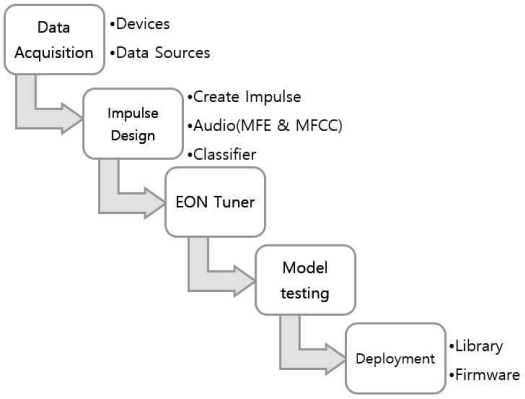

그림 1. 엣지 임펄스 워크플로
Fig. 1. Edge impulse workflow

데이터 수집이 끝나면 시계열 데이터에서 윈도우 크기와, 주파수 등을 설정하고 데이터 처리를 위한 MFE와 MFCC를 할당한다. 학습 블록에서 분류를 선택하여 입력된 음성을 학습시킨다. 다음으로 EON 튜너를 통해서 정확도나 지연 시간 기준으로 최적화된 모델을 선택하여 모델 테스트를 수행하고 최종적으로 타깃 보드에 맞춰 배포를 수행한다. 엣지 임펄스의 기반 기술인 TinML은 자원 제약 특성을 가지는 임베디드 디바이스에서 기계학습 구현을 지원하는 기술이다. TinyML을 지원하는 대표적인 도구는 엣지 임플스를 포함하여, 텐서플로 라이트[15], ELL(Embedded Learning Library)[16] 등이 있다. 텐서플로 라이트는 텐서플로로 훈련한 모델을 안드로이드나 iOS 및 라즈베리파이 등에서 사용할 수 있게 변환하는 기술이다. ELL은 마이크로소프트사의 임베디드 AI 및 머신 러닝용 오픈 소스 라이브러리이다. TinyML은 마이크로컨트롤러(MCU)로 구동되는 내장형 장치에 기계 학습 및 심층 신경망 모델을 배포하는 데 중점을 두며 대기 시간과 통신 대역폭의 제약이 낮은 애플리케이션을 위한 솔루션을 제공한다. 주로 센서 데이터와 같은 간단한 입력을 처리하고 디바이스 내에서 실시간으로 추론을 수행한다. TinyML의 주요 특성으로 작은 크기, 저전력, 빠른 추론, 데이터 전처리, 경량화, 딥러닝 아키텍처 등으로 임베디드 디바이스에 적합하다.

다음 장에서 엣지 임펄스를 이용하여 차원 축속 기법과 음성 처리 기법인 MFE와 MFCC를 비교 및 분석한다.

Ⅲ. 음성 처리 기법의 비교 및 분석

3장에서는 음성 특성 추출을 위한 전체적인 구조를 살펴보고, 음성 인식에 사용되는 데이터 세트와 차원 축소 기법인 t-SNE와 PCA를 비교한다. 다음으로 입력 블록 이후 처리 블록에서 MFE와 MFCC를 비교 분석한다. 마지막으로 학습 블록에서 분류를 수행하여 특성을 확인한다.

3-1 음성 특성 추출을 위한 절차

음성 인식을 위해서 먼저 타깃 보드나 스마트폰 혹은 데스크톱에서 음성 데이터를 입력받아서 데이터 세트를 구성한다. 다음으로 입력된 데이터 세트를 2차원 형태로 표현하기 위해서 차원 감소 기법(t-SNE, PCA)을 각각 적용하여 비교한다. 이후에 입력 블록에서 시계열 데이터를 설정하고 처리 블록에서 MFE와 MFCC를 각각 적용하여 파라미터 값, DSP 결과와 특성을 비교한다. 마지막으로 학습 블록에서 신경망 설정과 구조를 비교한다. 그림 2는 데이터 세트에서 분류까지의 흐름을 보인다. 다음으로 본 논문에서 사용한 데이터 세트를 설명한다.

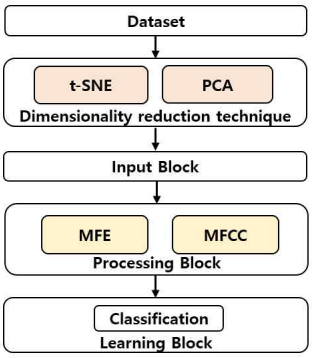

그림 2. 차원 축소 기법과 프로세싱 블록의 흐름
Fig. 2. Flow of dimensionality reduction techniques and processing blocks

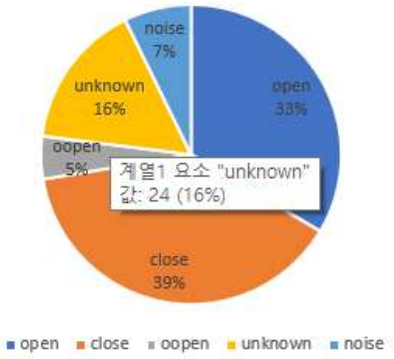

그림 3. 데이터 세트 비율
Fig. 3. Data set ratio

3-2 데이터 수집(데이터 세트)

음성 데이터는 QR code를 통해서 스마트폰으로 입력하거나 컴퓨터의 마이크 혹은 타깃 디바이스의 마이크로 직접 수집이 가능하다. 준비한 데이터 세트는 "Open", "Close", “Nano", "Noise", Unknown", "Uncertain" 등을 사용하였으며 “Open”이라고 말하면 타깃 보드에서 음성을 인식하여 서보모터를 동작하여 실제로 문이 열리는 방식이다. “Close”를 말하면 문을 닫을 수 있다. 훈련 및 테스트 데이터의 비율은 약 82 대 18을 유지했다. 데이터 세트의 비율은 그림 3과 같으며 "Open" 33%, "Close" 39%, "Unknown" 16%, ”oopen" 5%, "Noise" 7% 등의 비율이다. 모델에서 과적합이 발생하면 테스트에서 성능 저하가 일어날 수 있으므로 매개변수를 수정하거나 추가 데이터를 수집하여 과적합을 방지한다. 훈련 데이터는 기계 학습 모델을 학습시키는 데 사용하며 입력된 음성에 대한 패턴이나 규칙을 포함하며 입력에 대한 레이블로 구성된다. 훈련 데이터로 예측을 수행하고 예측과 인식 결과의 오차를 줄이기 위해서 파라미터 값 등을 수정한다. 테스트 데이터는 훈련된 모델의 성능 평가 목적으로 사용되며 입력된 음성 데이터에 대한 예측을 수행한다. 이로서 실제 환경에서 수행이 원활한지 확인할 수 있다.

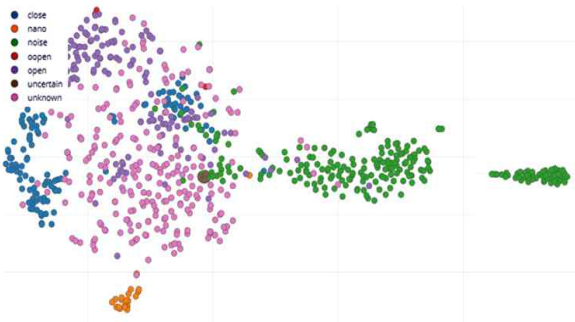

그림 4. 차원 축소 알고리즘(t-SNE)
Fig. 4. Dimensionality reduction technique(t-SNE)

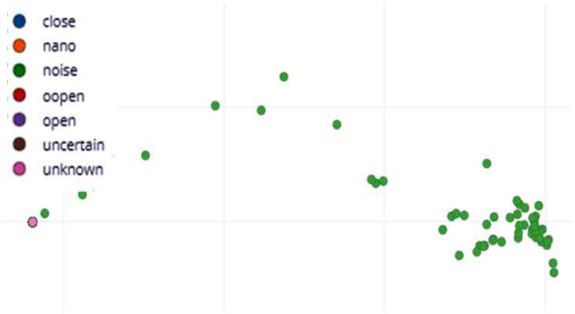

그림 5. 차원 축소 알고리즘(PCA)
Fig. 5. Dimensionality reduction technique(PCA)

원하는 명령(open, close)이 아닌 유사한 명령 혹은 인식 불가한 음성 데이터를 주입하여 실제 명령과 잘못된 명령을 비교하여 정확도를 높이는 데 사용된다. 테스트 데이터는 훈련 데이터와 독립성을 유지하지 않으면 과적합이 일어난다. 데이터 탐색기에서 모든 데이터 세트의 전체 보기를 수행하여 이상값 등을 확인하고 레이블을 지정할 수 있다. 데이터 탐색기는 음성 1초에 해당하는 윈도우에 맞는 키워드에 적합(Using a pretrained keywords model)한 방식을 사용한다. 추가로 레이블 지정 이후 일부 데이터를 수집하고 훈련된 모델이 있는 경우(Using your trained impulse)와 아직 데이터 레이블이 없으므로 전체 모델을 교육할 수 없는 경우(Using the preprocessing blocks in you impulse)로 나뉜다. 다음으로 입력된 데이터의 구조를 2차원 형태로 표현하기 위해서 차원 축소 알고리즘(Dimensionality reduction technique)을 선택한다. 엣지 임펄스는 t-SNE와 PCA를 선택할 수 있다. t-SNE(t-distributed Sochastic Neighbor Embedding)은 비선형적인 차원 축소 방법으로 높은 차원의 데이터의 구조를 시각적으로 이해하기 쉬운 2차원 형태로 축소하는 방법이다. 낮은 차원 공간 시각화에 주로 사용되며 주어진 데이터 세트에서 각 데이터 간의 유사도를 측정한 후 이를 저차원 데이터에 대한 유사도로 변환한다. 분류에 문제가 있다면 음원을 확인하고 음원에 맞는 레이블을 부여한다. 각 데이터 항목을 해당 레이블에 해당하는 색상으로 구분하여 전체 데이터 세트를 보여준다.

그림 6. 임펄스의 구성
Fig. 6. Composition of impulse

모든 항목을 확인하여 잘못된 클러스터에 있는 항목을 찾고, 항목을 클릭하여 음성을 확인할 수도 있다. 이를 통해서 데이터 세트가 ML에 적합한지 여부를 확인하는 데 유용하게 사용된다. 그림 4와 그림 5는 모두 동일한 테이터 세트에서의 결과이다. 예를 들어 녹색은 모두 “noise”로 구분되며 특성에 따라서 그룹을 형성함을 확인할 수 있다. 그림 5는 차원 축소 알고리즘 중 PCA 적용 결과로 PCA는 행렬 분해(Matrix factorization)을 기반으로 분산이 최대인 축을 찾고, 이 축과 직교이면서 분산이 최대인 두 번째 축을 찾아서 투용시키는 방식을 사용한다. 그림 4의 t-SNE보다 그림 5의 PCA가 확연하게 덜 분리되지만 모든 데이터 세트 크기에서 작동하는 이점이 있다.

본 논문에서는 짧은 음성 인식 데이터를 확인하기 위해서 t-SNE를 사용하였다.

3-3 임펄스 디자인

데이터 수집 이후 다음 단계로 임펄스 디자인을 수행한다. 임펄스는 그림 6과 같이 3가지의 블록(입력, 처리, 학습)으로 구성되며, 원시 데이터를 취득하고 신호처리를 사용하여 특성을 추출한 이후에 학습 블록에서 새로운 데이터를 분류한다. 다음 절에서 각 블록을 단계적으로 설명한다.

1) 입력 블록

시계열 데이터(Time series data)의 입력축 필드에는 훈련 데이터 세트에서 참조되는 모든 축이 나열된다. 오디오의 창 크기(Windows size)는 음성 분류별로 처리될 데이터의 크기를 밀리초 단위로 지정하며, 짧은 음성 명령을 고려하여 1,000ms로 지정했다. 윈도우 증가(Window increase)는 슬라이딩 윈도를 증가시키는 데 걸리는 시간으로 샘플이 창 크기보다 크다면 슬라이딩 창을 사용하여 해당 샘플의 검토가 필요하며, 각 단계에서 슬라이딩 윈도우를 증가시키는 데 걸리는 시간은 1,000ms로 설정하였다. 주파수는 데이터의 빈도를 설정하는 부분으로 훈련 샘플을 기반으로 자동으로 계산된다. 값의 설정으로 데이터가 다운 샘플링되거나 업 샘플링되며 설정값이 보드와 맞지 않으면 타깃에서 실행 시 오류가 발생할 수 있으므로 초기 설정값에서 16,000Hz로 설정하였다.

2) 처리 블록

처리 블록에서는 원시 데이터 블록, 스펙트럼 특성 블록, 오디오 MFE 블록, 오디오 MFCC 블록, 스펙트로그램 블록 등을 선택할 수 있다. 먼저 원시 데이터 블록은 전처리 작업 없이 데이터 샘플에서 창을 생성하며, 사전 처리된 신호와 데이터를 신경망 블록에 공급해야 하는 경우에 적합하다. 스펙트럼 특성 블록은 신호의 주파수와 전력 및 특성을 추출하며, 신호의 반복 패턴 분석에 유용하다. 오디오 MFE 블록은 신호에서 시간 및 주파수 특성을 추출하며, 음성보다 오디오에 적합한 멜필터 뱅크(Mel-Filter bank) 기능을 사용하여 오디오 신호에서 스펙트로그램을 추출하여 비음성 인식 데이터에서 원활하게 수행된다. 오디오 MFCC 블록은 오디오 신호에서 계수를 추출하며, 음성인식에서 원활히 동작한다. 스펙트로그램(방출되는 소리 영역을 주요한 특성을 중심으로 표시) 블록은 신호에서 시간 및 주파수 특성을 추출하며, 오디오 데이터나 연속적인 주파수가 발생하는 데이터에서 우수한 성능을 보인다. 동일한 데이터 세트를 기준으로 MFE와 MFCC를 각각 적용하여 결과를 비교 분석하였다.

- MFE

MFE의 특성 추출 단계에서는 데이터의 설정을 수정하고 시간 및 주파수 특성을 추출하며 분류 기법에서 음성을 인식하는 데 사용된다. DSP 알고리즘에 적합한 매개변수를 선택하는 것은 쉽지 않으므로 오토튠(Autotune) 파라미터를 적용한다. 표 1은 자동으로 적용된 값을 보인다. 프레임 길이는 프레임의 길이를 0.05초, 프레임 스트라이드(Frame stride)는 0.01초로 연속 프레임 사이의 단계를 의미한다. 필터 넘버는 스펙트로그램에 적용된 삼각형 필터의 수이며 41이다. FFT(Fast Fourier Transform) 길이는 신호를 주파수 성분으로 변환하는 알고리즘이며 이산 푸리에 변환(DFT, Discrete Fourier Transform)를 고속으로 계산하는 알고리즘을 지칭했지만, 지금은 동일한 의미로 사용된다. 설정값은 512이다. 낮은 주파수와 높은 주파수는 멜-스케일 필터 뱅크의 최저 대역 및 최고 대역 에지를 말한다.

표 1. MFE 파라미터 값
Table 1. MFE parameters values

Mel-filterbank energy features	
Frame length	0.05
Frame stride	0.01
Filter number	41
FFT length	512
Normalization window size	151
Low frequency	80
High frequency	0
Normalization noise floor(dB)	-100

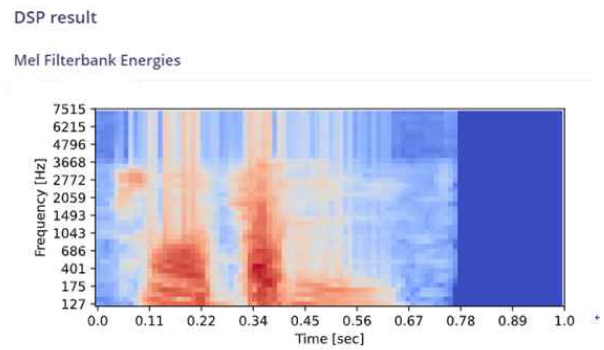

그림 7. DSP 결과(MFE, 처리 블록)
Fig. 7. DSP result(MFE, Processing block)

그림 8. 특성 탐색기(MFE, 처리 블록)
Fig. 8. Feature explorer(MFE, Processing block)

노이즈 플로워(Noise floor)는 -100dB로 설정된 값보다 낮으면 드롭된다. 스펙트로그램 계산 이후 멜-스케일에 필터를 적용하여 주파수 대역을 추출한다. 필터 번호와 저주파 및 고주파 매개변수로 구성되어 주파수 대역과 추출할 주파수 특성 수를 선택한다. 멜-스케일은 듣는 거리가 같다고 판단하는 음조의 지각 척도를 말한다. 낮은 주파수에서는 더욱더 많은 특성(필터 뱅크 많음)을 추출하고 높은 주파수에서는 적게 추출하여 사람이 구별할 수 있는 소리에 적합하다. 마지막으로 노이즈 플로어 값을 파워 스펙트럼에 적용하여 신호의 로컬 평균 정규화를 수행한다. 그림 7은 MFE에서 DSP의 결과를 보인다. 멜-스케일은 특정 피치에서 발견한 음을 인지 기준(Threshold)에 반영한 스케일 변환 함수이며, 멜-필터뱅크는 멜-스케일에서 선형적으로 구간을 N 개로 나누어 구현한 삼각형 필터(Triangular filter)이다. 설정값을 기반으로 특성 추출을 그림 8과 같이 시각적으로 확인할 수 있다. 수직 축은 주파수(주파수 대역의 수는 '계수의 수')를 나타내고, 수평 축은 시간('프레임 스트라이드' 및 '프레임 길이')을 나타낸다. 스펙트로그램의 패턴으로 어떤 종류의 소리를 표현하는지 확인할 수 있다. 특성 탐색기는 입력 기능의 공간 분포를 나타낸다. 이로서 올바르게 분류된 항목과 그렇지 않은 항목을 시각화한다.

- MFCC

MFCC(Mel Frequency Cepstral Coefficients)는 사람의 목소리에 적합하게 설계되어 오디로 신호에서 특성을 추출한다. 음성 데이터에서 불필요한 정보는 버리고 중요한 특성만을 남긴다. MFCC는 오디오 신호에서 추출할 수 있는 특성으로 소리의 고유한 특성을 나타내는 수치를 말한다. 주로 음성 인식과 합성 및 화자 인식 등의 오디오 분석에 사용한다. 표 2와 같이 매개변수의 값은 MFE와 대부분의 항목에서 유사하므로 일부 내용만 설명한다. 먼저 계수의 수(Number of coefficients)는 이산 코사인 변환을 적용한 후 유지할 켑스트럴 계수의 수이다.

표 2. MFCC 파라미터 값
Table 2. MFCC parameter values

Mel Frequency Cepstral Coeffients	
Number of coefficients	13
Frame length	0.025
Frame stride	0.02
Filter number	32
FFT length	512
Normalization window size	151
Low frequency	80
High frequency	0
Pre-emphasis coefficient	0.98

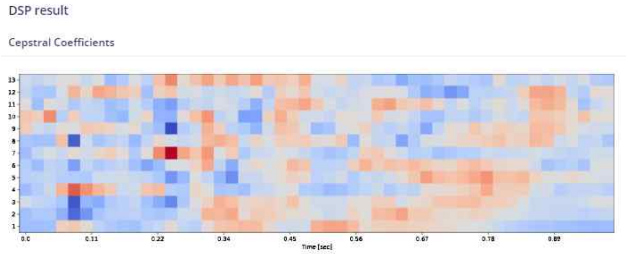

그림 9. DSP 결과(MFCC, 처리 블록)
Fig. 9. DSP result(MFCC, Processing block)

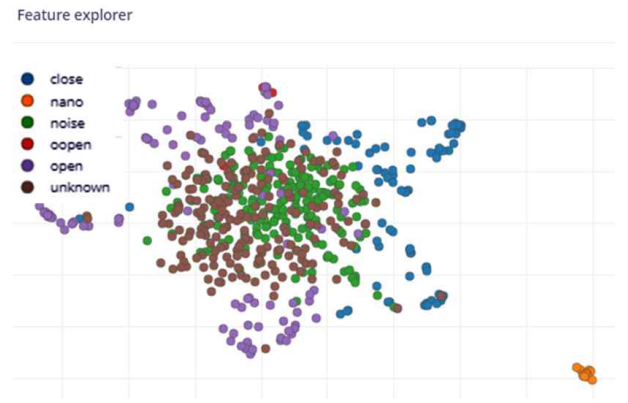

그림 10. 특성 탐색(MFCC, 처리 블록)
Fig. 10. Feature explorer(MFCC, Processing block)

프리 엠퍼시스(pre-emphasis) 계수는 입력 신호에 적용할 사전 강조 계수로 0은 필터링 없음을 의미한다. 동일한 데이터 세트에서 MFCC의 DSP 결과는 그림 9와 같이 MFE와 차이를 보이며, 그림 10의 음성 특성 패턴도 중앙 집중적으로 인식되고 있음을 보인다. 동일 음성 데이터 세트에 대해서 처리 블록에서 MFE와 MFCC 적용에 따른 파라미터 값과, DSP 및 특성에서의 차이를 비교하였다. 다음 절에서 학습 블록의 신경망 설정과 구조 및 정확도를 살펴본다.

3) 학습 블록

학습 블록은 신호 처리를 사용하여 원시 신호에서 의미 있는 기능을 추출한 후 학습 블록을 사용하여 모델을 훈련할 수 있다. 엣지 임펄스에서 지원하는 학습 블록은 분류, 회귀, 전이 학습, 이상 탐지 등으로 구성되어 있다. 분류는 DSP 블록에서 특성을 추출한 후에 사용할 수 있으며 데이터에서 패턴을 학습한 이후에 새로운 음성 데이터에서 적용할 수 있다. 오디오 인식이나 움직임 등의 분류에 적합하다. 회귀는 변수들 간의 관계를 모델링 하는 것으로서 연속된 숫자 값을 예측하는데 적합하며, 전이학습은 이미 학습된 모델의 일부 또는 전체를 가져와 새로운 문제를 해결하는 데 사용된다. 이상 탐지는 주어진 데이터에서 이상 값을 식별하고 분석한다. 본 연구에서는 학습 블록으로 분류를 적용하였다. 신경망 분류기는 일부의 입력 데이터를 기반으로 특정한 클래스에 속할 가능성을 나타내는 확률 점수를 출력한다. 신경망은 여러 개의 레이어로 구성되며 각 레이어는 여러 개의 뉴런으로 구성된다. 첫 번째 계층의 뉴런은 두 번째 계층의 뉴런에 연결되며 레이어에 있는 두 뉴런 간의 연결 가중치는 훈련 프로세스가 시작 시에 무작위로 결정된다. 이후에 신경망에는 예측해야 하는 예제 집합인 훈련 데이터 집합이 제공된다. 출력된 결과를 비교 후 그 결과에 따라 계층의 뉴런 간 연결 가중치를 조정한다. 이 프로세스는 네트워크가 학습 데이터에 대한 정답을 예측하는 방법을 학습할 때까지 여러 번 반복한다. 계층의 특정 배열을 아키텍처라고 하며 다른 아키텍처는 다른 작업에 유용하다. 많은 반복 후에 신경망을 학습하며, 결국 새로운 데이터를 훨씬 더 잘 예측하게 된다. 먼저 MFE에 분류를 적용한 결과를 보인다.

- MFE

학습 블록의 분류에서 신경망 설정을 표 3에서 보인다. 훈련 사이클 수(Number of training cycles)는 교육 주기 수이다. 교육 알고리즘이 역전파를 사용하여 모든 교육 데이터를 한 번 완전히 통과하고 진행되는 동안 모델의 매개변수를 업데이트할 때마다 이를 에포크 또는 교육 주기라고 한다. 학습률은 학습 프로세스의 각 단계에서 모델 내부 매개변수가 업데이트되는 정도를 제어한다.

표 3. 신경망 설정(MFE)
Table 3. Neural network settings(MFE)

Neural Network settings	
Number of training cycles	100
Learning rate	0.005
Validation set size	20%
Auto-balance dataset	no
Profile int8 model	check

그림 11. 신경망 구조(MFE)
Fig. 11. Neural network architecture(MFE)

또한 신경망이 얼마나 빨리 학습하는지 확인할 수 있으며 네트워크가 빠르게 과적합되면 학습률을 줄일 수 있다. 유효성 세트 크기(Validation set size)는 검증을 위해 분리된 훈련 세트의 백분율로 기본값은 20%이다. 자동-균형 데이터 세트(Auto-balance dataset)은 일반적이지 않은 클래스의 데이터 사본을 더 많이 혼합한다. 일부 클래스에 대한 데이터가 거의 없는 경우 과적합에 대해 모델을 보다 강력하게 만드는 데 도움이 될 수 있다. 프로젝트 유형에 따라 시작하는 데 도움이 되도록 다양한 아키텍처의 사전 설정 중에서 선택할 수 있다. 그림 11과 같이 신경망 아키텍처는 추출된 기능을 입력으로 사용하고 기능을 아키텍처의 각 계층에 전달한다. 필터링의 결과를 뉴런의 첫 번째 레이어로 공급하고, 첫 번째 레이어의 결과를 다시 두 번째 레이어에 공급한다. 클래스 중 하나에 속할 확률을 제공하는 것은 이 마지막 레이어이다. 신경망의 학습과 네트워크 과적합 등의 문제가 발생하면 학습률을 낮춘다. 그림 12는 훈련이 완료된 결과를 보이며 모델의 성능을 평가하는 지표이다. 정확도 75.8%는 올바르게 분류된 오디오 창의 백분율이다. 정확도가 100%에 가까워지면 때로는 모델이 훈련 데이터에 과적합되었다는 뜻이다. 일반적으로 85% 이상의 정확도는 매우 우수한 것으로 판단 할 수 있다. 모델의 동일한 비교를 위해서 정확도를 올리기 위한 작업을 최소화하였기에 정확도가 상대적으로 낮게 나왔다. 혼돈 행렬은 모델을 평가하는 데 가장 유용한 도구 중 하나로 분류 모델의 예측 결과를 평가하는 데 사용된다.

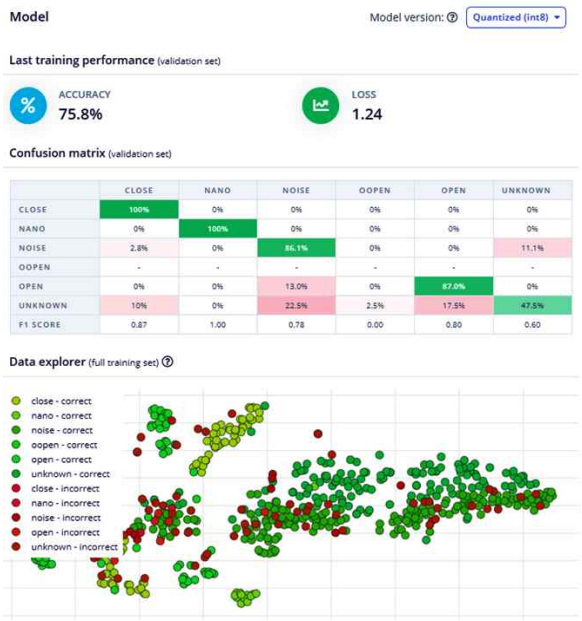

그림 12. 특성 탐색기(MFE)
Fig. 12. Feature explorer(MFE)

주어진 데이터 세트에서 모델이 생성하는 모든 정답 및 오답 응답을 표로 작성한다. 왼쪽의 레이블은 각 샘플의 실제 레이블이며, 상단의 레이블은 모델에서 예측된 레이블을 나타낸다. 그림 12의 MFE 특성 탐색기는 입력 기능의 공간 분포를 나타낸다. 음성 인식이 올바르게 분류된 항목과 올바르지 않은 항목을 시각적으로 확인하고 수정 가능하다. 온 디바이스 성능은 모델이 디바이스에서 실행될 가능성에 대한 값으로 추론 시간은 모델이 아두이노에서 1초의 데이터를 분석하는 데 걸리는 시간을 추정한 것이다. 디바이스에서 모델을 실행하는데 필요한 최대 RAM 사용량은 9.6k이며, 추론 시간은 43ms, 플래시 사용량은 34.1k이다.

- MFCC

그림 13. 신경망 구조(MFCC)
Fig. 13. Neural network architecure(MFCC)

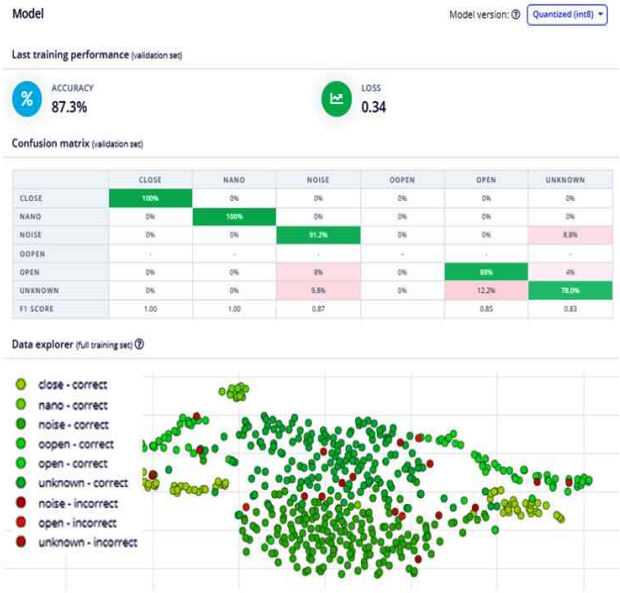

그림 14. 특성 탐색기(MFCC)
Fig. 14. Feature explorer(MFCC)

MFCC를 적용한 그림 13의 신경망 아키텍처의 입력 레이어는 650, 리쉐이프 레이어 13으로, MFE 입력 레이어 3960과 리쉐이프 레이어 40과 차이를 보인다. 그림 14와 같이 학습 블록에서 모델은 정확도 87.3%를 보이며, 데이터 탐색기에서 음성별로 고른 인식 분포를 확인할 수 있다. 온 디바이스 성능은 추론 시간 27ms, 최대 램 사용량은 7.4K, 플래시 메모리 사용량은 65.4K이다. 다음 장에서 각각의 모델을 최적화하고 모델 테스팅 결과와 성능을 비교한다.

Ⅳ. 성능 평가 및 결과

3장에서 MFE와 MFCC로 각각 분류한 모델을 EON(Edge Optimized Neural) 튜너를 적용하여 정확도, 지연시간을 기준으로 성능과 결과를 비교하였다. EON 튜너에서 타깃 보드에 가장 적합한 ML 아키텍처를 찾고 선택할 수 있다. EON 튜너는 입력 데이터, 신호 처리 블록 및 신경망 아키텍처를 분석하고 선택한 장치의 대기 시간 및 메모리 요구 사항에 맞는 가능한 모델 아키텍처에 대한 정보를 제공한다. 먼저 MFE에 EON 튜너를 적용하여 정확도와 지연 시간 기준으로 성능을 비교한다.

4-1 MFE

MFE 모델에서 EON 튜너를 적용하면 튜닝된 다양한 모델을 선택할 수 있다. 가장 높은 정확도를 보이는 모델과 동일한 환경에서 가장 지연 시간이 빠른 모델을 선정하였다. 각각의 특성을 가지는 모델에서 DSP와 NN의 지연시간과 RAM, ROM 사용을 확인한다.

그림 15. EON 튜너 적용 결과(MFE, 정확도)
Fig. 15. Result of applying EON tuner(MFE, Accuracy)

그림 16. EON 튜너 적용 결과(MFE, 지연시간)
Fig. 16. Result of applying EON tuner(MFE, Latency)

그림 15는 EON 튜너 적용 후 정확도 기준으로 가장 높은 모델(93%)을 보이며, 그림 16은 가장 낮은 지연시간(115.08ms)을 가지는 모델이다.

표 4. EON 튜너 적용 후 비교(MFE, 정확도, 지연시간)
Table 4. Comparison after applying EON tuner(MFE, Accuacy, Latency)

	Accuracy 93%		Latency 115.08ms	
	DSP	NN	DSP	NN
Latency	79ms	43ms	79ms	36ms
RAM	8kb	10kb	8kb	7kb
ROM	0kb	34kb	0kb	65kb

그림 17. 모델 테스팅 결과(MFE)
Fig. 17. Model testing results(MFE)

x축은 각각의 데이터 세트를 나타내고 검정 박스의 숫자는 정확도를 나타낸다. 표 4에 정확도와 지연시간 기준에서 DSP와 NN의 값을 정리하였다. 가장 높은 정확성(93%)과 지연시간(115.08ms)이 빠른 모델에서 DSP의 지연시간과 램 사용량을 동일하며 NN에서 일부 차이가 보였다. 표 4의 결과를 기준으로 정확도가 높은 모델 혹은 지연 시간이 빠른 모델을 선택했을 경우에 모델 선택에 따른 차이는 크지 않음을 알 수 있다. 그림 17에서 모델의 학습 이후에 모델 테스트의 결과를 보인다. 레이블이 잘못 구분된 경우에는 레이블을 업데이트하거나 해당 항목을 훈련 세트로 이동하여 재구성을 진행할 수 있다. 일부 항목을 훈련 세트로 이동할 경우에는 다시 훈련을 수행한다. 그림 17에서 정확도는 86.82%로 입력 모델이 충분하지 않음에도 좋은 결과물이 나온 이유는 학습 데이터와 대부분 동일한 환경에서 수행되기 때문이다.

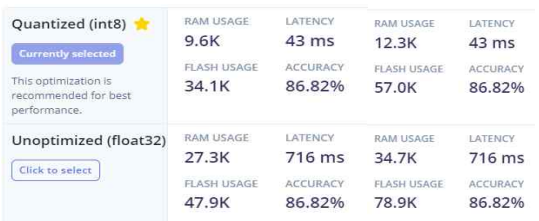

그림 18. 배포(MFE)
Fig. 18. Deployment(MFE)

입력 데이터에서 일부 변화가 발생하면 성능 저하가 발생할 수 있다. 그림 18은 타깃 보드를 아두이노 기준으로 배포했을 때 지연시간과, 정확도, 램 사용량 및 플래시 사용량에 대한 추정값을 보인다. 최적화 전의 float32와 양자화한 이후 값을 확인할 수 있다.

4-2 MFCC

그림 19. EON 튜너 적용 결과(MFCC, 정확도)
Fig. 19. Result of applying EON tuner(MFCC, Accuracy)

그림 20. EON 튜너 적용 결과(MFCC, 지연시간)
Fig. 20. Result of appllying EON tuner(MFCC, Latency)

다음으로 MFCC에서 타깃 보드에 가장 적합한 매개변수와 최상의 모델을 찾기 위해서 EON 튜너의 적용의 결과를 가장 높은 정확도와 가장 낮은 지연 시간 모델을 선택하여 비교한다. 그림 19는 정확도 91% 모델에서 음성별로 정확도를 확인할 수 있다. 그림 20은 지연 시간 기준(115.08ms)이므로 지연 시간 대비 음성의 정확도가 일부 낮아지는 것을 확인할 수 있다. 예를 들어 "close"가 100%에서 88%, "open"이 100%에서 90%, “unkown”이 80%에서 71%이다. 정확도와 지연 시간을 모두 만족할 수 없으므로 값을 참고하여 정확도가 중요하다면 지연 시간이 조금 늦어지더라도 해당 모델을 선택한다. 표 5의 EON 튜너 적용 후 DSP와 NN의 비교 값을 참고하여 모델을 선택한다.

표 5. EON 튜너 적용 후 비교(MFCC, 정확도, 지연시간)
Table 5. Comparison after applying EON tuner(MFCC, Accuacy, Latency)

	Accuracy 91%		Latency 115.08ms	
	DSP	NN	DSP	NN
Latency	153ms	33ms	79ms	36ms
RAM	11kb	9kb	8kb	7kb
ROM	0kb	66kb	0kb	65kb

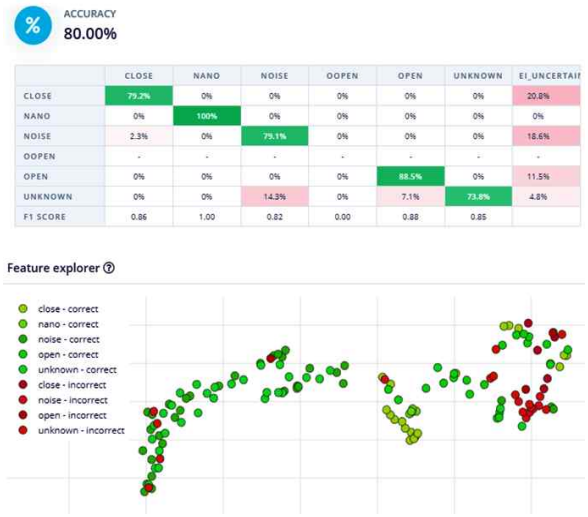

그림 21. 모델 테스팅 결과(MFCC)
Fig. 21. Model testing results(MFCC)

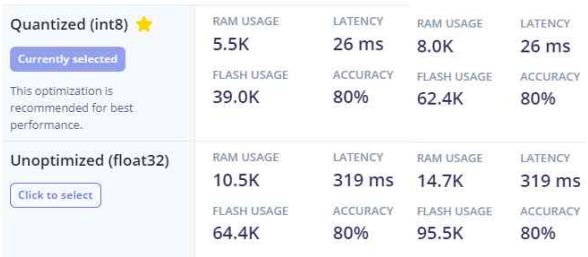

그림 22. 배포(MFCC)
Fig. 22. Deployment(MFCC)

표 6. MFE와 MFCC의 비교
Table 6. Comparison of MFE and MFCC

	MFE		MFCC	
	int8	float32	int8	float32
Accuracy	86.82%	86.82%	80%	80%
Latency	43ms	716ms	26ms	319ms
Ram usage	9.6k	27.3k	5.5k	10.5k
Flash usage	34.1k	47.9k	39.0k	64.4k

정확도(91%)는 지연시간이 153ms이므로 지연 시간 우선 모델 대비 74ms의 속도 차이가 발생한다. 다음으로 MFCC의 모델 테스팅 이후의 결과는 그림 21과 같다. 정확도 80%에서 음성별로 인식률과 특성 분포를 확인할 수 있다. 모델 테스트가 완료되면 타깃 보드에 배포 작업을 수행한다. 아두이노에 배포 시 예상값은 그림 22와 같다. 표의 왼쪽과 오른쪽값은 EON 적용 전후를 나타내며 EON 적용 효과가 큼을 알 수 있다. 양자화 적용 값의 비교에서 동일한 정확도 80%를 유지하면서 float32의 지연시간 319ms가 적용 이후에 26ms로 줄어들었음을 확인할 수 있다.

마지막으로 MFE와 MFCC 모델에서 최종 타깃 보드에 배포 시 양자화 전후의 정확도와 지연시간, 램 사용량, 플래시 사용량을 표 6에서 비교하였다. 동일한 데이터 세트에서 MFE가 정확도 기준에서 86.82%로 MFCC의 80% 대비 약 7% 우수하며, MFCC가 지연 시간 기준에서 26ms로 MFE의 43ms 대비 17ms가 빠름을 확인할 수 있다. 1초 이내의 짧은 음성 명령을 인식하여 타깃 보드의 센서와 장치를 구동하는 연구에서, 음성 데이터의 주파수 성분을 자세히 분석하고 높은 정확도가 필요한 경우는 MFE 모델을 선택하여 특성을 파악하고 잘못 학습된 모델을 수정하여 더욱 높은 정확도를 유지할 수 있다. MFCC는 음성 데이터의 주파수와 시간적 특성을 모두 고려하여 음성 신호를 표현하므로 음성 신호를 간결하게 표현할 수 있다. 실험 결과와 같이 조금이라도 지연시간이 빠른 모델이 필요하다면 MFCC 모델을 선택하여 정확도는 조금 떨어지지만 원하는 지연 시간내에 수행할 수 있다. 음성 모델에 따른 정확도와 지연 시간에서 차이가 발생하므로 개발 목적에 맞게 다양한 비교와 분석으로 최적의 모델을 선택에 도움이 될 수 있다.

V. 결 론

엣지 컴퓨팅 환경에서 음성 인식 처리는 중요해지고 있다. 자원 제약적인 임베디드 디바이스에서 기계 학습 기반의 음성 처리를 위해서 엣지 임펄스를 사용하였다. 음성 데이터 세트에서 비선형 차원 축소 알고리즘인 t-SNE와 PCA를 각각 적용하여 t-SNE가 PCA 대비 높은 차원의 데이터 구조를 시각적으로 이해하기 쉽고 분류에도 용이함을 확인하였다. 다음으로 처리 블록에서 MFE와 MFCC를 적용하여 각각의 파라미터 값과, DSP 결과 및 특성을 확인하고 EON 튜너를 적용하여 최적의 모델을 선택하였다. 모델 테스트의 수행 결과는 MFE의 정확도 86.82%, MFCC의 정확도 80% 수준이므로 짧은 음성 인식에는 MFE 적용 모델이 우수하며, 아두이노 대상의 온디바이스 성능을 양자화 전후로 비교하면 MFE의 int8 지연 시간은 43ms, float32는 716ms이며, MFCC의 int8 지연 시간은 26ms, float32는 319ms로 MFCC가 낮은 지연시간을 보였다. 데이터 세트와 학습 방법 및 설정값에 따라 결과의 차이는 존재하지만, 실시간에 민감한 디바이스에서 음성 인식을 기계학습으로 수행할 때 성능을 확인하고 개발 목적에 맞는 모델을 선택할 수 있다.

향후 연구과제로 다양한 센서를 활용하여 기계 학습에 특화된 에지 시스템을 구현하고, 상대적으로 저렴한 비용으로 접근할 수 있는 MCU에 기계 학습을 적용하는 교육에 적용해 보고자 한다.

참고문헌

[1] M. Satyanarayanan, “The Emergence of Edge Computing,” *2017 Eleventh International Conference on Sensing Technology (ICST),* Vol. 50, No. 1, pp. 30-39, January 2017. https://doi.org/10.1109/MC.2017.9

[2] S. Yang, Z. Gong, K. Ye, Y. Wei, Z. Huang, and Z. Huang, “EdgeRNN: A Compact Speech Recognition Network with Spatio-Temporal Features for Edge Computing,” *IEEE Access*, Vol. 8, pp. 81468-81478, April 2020. https://doi.org/10.1109/ACCESS.2020.2990974

[3] 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

[4] 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

[5] I. N. Mihigo, M. Zennaro, A. Uwitonze, J. Rwigema, and m. Rovai, “On-Device IoT-Based Predictive Maintenance Analytics Model: Comparing TinyLSTM and TinyModel from Edge Impulse,” *Sensors*, Vol. 22, No. 14, 5174, July 2022. https://doi.org/10.3390/s22145174

[6] 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

[7] 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

[8] M. Wattenberg, F. Viégas, and I. Johnson, “How To Use t-SNE Effectively,” *Distill*, 2016. http://doi.org/10.23915/distill.00002

[9] F. Anowar, S. Sadaoui, and B. Selim, “Conceptual and Empirical Comparison of Dimensionality Reduction Algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE),” *Computer Science Review*, Vol. 40, February 2021. https://doi.org/10.1016/j.cosrev.2021.100378

[10] I. Martín-Morató, M. Cobos, and F. J. Ferri, “A Case Study on Feature Sensitivity for Audio Event Classification Using Support Vector Machines,” *2016 IEEE 26th International Workshop on Machine Learning for Signal Processing (MLSP)*, Vietri sul Mare, Italy, September 2016. https://doi.org/10.1109/MLSP.2016.7738834

[11] N. Sato and Y. Obuchi, “Emotion Recognition Using Mel-frequency Cepstral Coefficients,” *Information and Media Technologies*, Vol. 2, No. 3, pp. 835-848, 2007. https://doi.org/10.11185/imt.2.835

[12] V. Tiwari, “MFCC and Its Applications in Speaker Recognition.” *International Journal on Emerging Technologies*, Vol. 1, No. 1, pp. 19-22, 2010.

[13] C. Ittichaichareon, S. Suksri, and T. yingthawornsuk, “Speech Recognition Using MFCC,” *International Conference on Computer Graphics, Simulation and Modeling (ICGSM 2012)*, Pattaya, Thailand, July 2012.

[14] J. Martinez, H. Perez, E. Escamilla, and M. M. Suzuki, “Speaker Recognition Using Mel Frequency Cepstral Coefficients (MFCC) and Vector Quantization (VQ) Techniques,” *CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers*, Cholula, Mexico, pp. 248-251, February 2012. http://doi.org/10.1109/CONIELECOMP.2012.6189918

[15] Tensorflow. Tensorflow Lite for Mobile & Edge [Internet]. Available: https://www.tensorflow.org/lite/.

[16] Microsoft. Embedded Learning Library [Internet]. Available: https://microsoft.github.io/ELL/.

구금서(Geum-Seo Koo)

2003년 : 경상대학교 컴퓨터과학과 (이학사)
2005년 : 경상대학교 컴퓨터과학과 (공학석사)
2007년 : 경상대학교 컴퓨터과학과 (박사수료)

2005년～현 재: 경상국립대학교 강사
※관심분야 : Embedded System, Edge AI, Machine Learning, 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 : 컴퓨팅 사고, 의료 영상 처리, SLAM, 영상 인식, 컴퓨터 네트워크