2024년 2월

석사학위 논문

속도와 노이즈 증강 데이터를 통한 음성 감정 인식 성능 향상

조선대학교 산업기술창업대학원

소프트웨어융합공학과

안 진 성

속도와 노이즈 증강 데이터를 통한 음성 감정 인식 성능 향상

Speech Emotion Recognition Performance Improvement with Speed and Noise Augmentation Data

2024년 2월 23일

조선대학교 산업기술창업대학원

소프트웨어융합공학과

안 진 성

속도와 노이즈 증강 데이터를 통한 음성 감정 인식 성능 향상

지도교수 신 주 현

이 논문을 공학 석사학위신청 논문으로 제출함.

2023년 10월

조선대학교 산업기술창업대학원

소프트웨어융합공학과

안 진 성

안진성의 석사학위논문을 인준함

위원장 조선대학교 교수 김 판 구 (인)

위 원 조선대학교 교수 신 주 현 (인)

위 원 조선대학교 교수 최 준 호 (인)

2023년 11월

조선대학교 산업기술창업대학원

목 차

표 목 차

그 림 목 차

ABSTRACT

Speech Emotion Recognition Performance Improvement with Speed and Noise Augmentation Data

Ahn, JinSung
Advisor : Prof. Shin, JuHyun Ph.D.
Department of Software Convergence Engineering
Graduate School of Industrial Technology and Entrepreneurship, Chosun University

With the recent development of AI in the field of voice emotion recognition research is expanding to utilize the characteristics of emotions revealed in human voices through tone of voice or non-verbal elements. However, the voice data sets currently distributed are limited, and there were practical difficulties in collecting the desired data sets directly. Therefore, this study proposed an experiment to improve voice emotion recognition by applying speed and noise enhancement methods and MFCC and Mel-Spectrogram feature vector extraction techniques based on a small amount of data sets. The study used CREMA datasets classified into a total of five emotional classes: 'anger', 'fear', 'happiness', 'sad', and 'neutral'. Data was augmented by adjusting speed and noise expansion, and vectors were extracted using two feature vector extraction techniques, MFCC and Mel-Spectrogram. As a result of the experiment, data augmentation was applied rather than the accuracy of learning only the original data, and the accuracy of learning resulted in a performance improvement of about 3% or more.

Ⅰ. 서론

A. 연구 배경 및 목적

음성은 사람의 감정을 인식할 수 있는 중요한 정보를 제공한다. 최근 COVID-19로 인한 비대면 서비스 확산으로 온라인 소통이 증가함에 따라 음성이나 텍스트, 이미지 등 모달리티를 통한 사람의 감정 인식에 관한 연구가 다양하게 진행되고 있다. 음성 감정 인식(Speech Emotion Recognition, SER) 분야에서는 사람의 음성에 숨겨져 있거나 잘 드러나 있지 않은 감정을 학습하여 기계가 인식할 수 있도록 하는 것에 초점을 두고 있다. 하지만, 기계가 사람의 감정을 정확하게 인식하는 것은 쉬운 일이 아니다[1].

사람의 음성에서 감정을 정확하게 인식하기 위한 구성 요소 중 하나는 어조가 있다. [2]의 연구에서는 어조가 사람의 음성과 관련된 요소이며, 음성의 속도, 높낮이, 음량, 세기, 장단, 음질 등을 포함한다고 설명하였다. 특히, 대화 간 비언어적 요소 중 음성적 언어인 어조와 감정 간에 연관이 있으며 비언어적 신호가 더 상대방의 감정을 파악하는 데 유효하다고 하였다. 예를 들어, 감정이 격앙되고 분노한 화자의 음성은 평상시의 상태와 비교했을 때 음성의 평균 높낮이가 상승한다. 즉, 화자는 더 높거나 낮은 음성을 발화하게 되어 감정의 강도와 관련이 있게 되므로 감정 상태를 파악하는 데 중요한 단서를 제공한다. 그리고 같은 단어로 이루어진 문장을 발화할 때도 감정이 격앙되면 문장 내에서의 음성 높낮이의 변화가 급격해진다. 이는 화자가 감정을 표현하기 위해 문장 내에서 특정 부분을 강조하거나 감정을 더 명확하게 전달하기 위해 음성의 억양을 활용하기 때문이다. 게다가, 인간의 감정은 표정, 몸짓, 자세 등 다양한 비언어적 신호를 통해서도 전달될 수 있다. [3]의 연구에서는 사람의 감정을 중립, 화남, 슬픔, 행복으로 구분하여 BPM 분석을 통해 슬픔과 분노, 일반적 감정과 분노를 명확하게 구분하였으며, 어조에서 도 감정 추론이 가능하다는 점을 파악하여 감정과 어조 간에 상호 밀접한 연관이 있음을 제시하였다.

[4]의 보고서에 따르면 감정 인식 체계에 대해 조사한 음성 감정 데이터를

통해 모델은 억양, 발음, 언어, 감정 등 다양한 음성 특성을 학습할 수 있다고 한다. 이로 인해 여러 종류의 음성 데이터는 모델이 다양한 환경과 상황에서의 음성을 포함하도록 도와 실제 환경의 음성 특징을 처리하고 인식하는 기술 과정에 반영할 수 있다. 또한, 충분한 음성 데이터를 사용하여 훈련된 모델은 일반화 능력이 강화되어 새로운 데이터나 다른 환경에서 사용자, 환경, 언어 등 특성에 따른 차이를 모델이 학습하고 처리할 수 있다. 이는 음성 기반 인터페이스 및 서비스를 다양한 사용자에게 제공하고 모델이 음성 데이터의 다양한 변화와 특징을 이해하므로 에러 발생률을 감소시키는 데 도움을 줄 수 있다. 따라서 충분한 음성 데이터는 음성 처리 및 인식 기술의 핵심 요소로 모델의 정확도, 일반화 능력, 다양성 확보, 사용자 차이 처리, 감정 및 의미 인식, 에러 감소 등 다양한 측면에서 중요한 역할을 한다고 볼 수 있다.

하지만 외부 환경에 대한 영향을 받는 음성의 특성 상 다양한 상황에서의 음성의 학습이 필요로 한데, 직접 음성 데이터를 수집하고 라벨링(labeling)하는 작업은 복잡하며 비용이 많이 든다. 이로 인해 많은 연구자와 기업은 제한된 데이터 셋으로 감정 인식 모델을 개발하는 어려움을 겪고 있으며, 일반화 능력과 정확도에 제약이 생긴다. 또한, 모델이 훈련 데이터에만 적응하고 다양한 감정과 환경에서의 음성을 잘 인식하지 못해 데이터 부족으로 인한 과적합(overfitting)이 발생할 수 있다. 이를 해결하기 위해 기존 데이터를 변형하거나 확장하여 새로운 데이터를 생성하는 방법인 데이터 증강 방법을 사용할 수 있다. 이는 제한된 훈련 데이터로도 감정 인식 모델의 정확도와 일반화 능력을 향상시키는 효과가 있다[5]. 또한, 실제 상황에서의 감정 분석과 다른 음성 처리 작업에 적용할 때 여러 성격의 데이터를 가질 수 있어 학습 간에 이점을 가질 수 있으며 음성 데이터를 증강함으로써 모델이 다양한 발화 속도와 음성 톤의 환경에서 감정을 인식하도록 도움을 준다. 결과적으로 데이터 증강은 데이터 셋 부족의 제약을 해소하고 효율적이고 음성 처리 및 음성 인식 연구를 가능하게 한다.

이에 본 연구에서는 원본 데이터 기반 속도와 노이즈 증강 기법을 적용한 다양한 데이터 셋을 생성하고 두 가지 특징 벡터 추출 방법인 MFCC, Mel-Spectrogram을 활용한 실험을 통해 음성 감정 인식 성능을 향상하는 방법을 제안하고자 한다.

B. 연구 내용 및 구성

본 논문에서는 음성 감정 인식 성능 향상을 목표로 원본 데이터 기반 속도와 노이즈 증강 기법을 통해 다양한 음성 데이터 인식 실험을 진행하였다. 음성의 속도와 노이즈를 여러 배율로 구분하여 다양하고 많은 데이터 셋을 생성하고 MFCC(Mel-Frequency Cepstral Coefficients)와 Mel-Spectrogram 음성 특징 추출 방법을 적용하여 특징을 추출하였다. [6]의 연구와 같이 음성 특징 벡터는 데이터의 일관성과 정규화를 확보하고 분류 능력을 향상한 결과를 도출할 수 있는 SVM(Support Vector Machine) 모델을 사용하여 학습하였다. 학습 간 음성의 라벨된 감정을 분류하고 파라미터를 조정하여 최적의 모델을 생성하였고 Confusion matrix를 지표로 평가하였다.

본 논문의 구성은 다음과 같다. 2장에서는 음성 감정 인식에 대한 멀티 모달과 모델에 기반한 연구, 음성 데이터 증강 기법 및 음성 특징 벡터 추출 연구에서 활용되는 기법과 관련된 내용을 소개한다. 3장에서는 제안하는 음성 감정 인식 성능 향상을 목표로 음성 데이터 증강 기법, 음성 특징 벡터 추출 방법과 음성 데이터 감정 인식 방법에 대한 연구 구성 과정을 상세하게 설명하였다. 4장에서는 3장을 기반으로 주장한한 실험 내용과 결과를 설명하였다. 원본 데이터 셋과 증강 데이터 셋에 대한 내용을 설명하고 감정 인식 정확도와 Confusion matrix 실험 평가 방법으로 실험 결과를 분석하였다. 5장에서는 연구 내용의 결론과 향후 연구 계획을 설명하고 참고 문헌을 소개하며 마무리한다.

Ⅱ. 관련 연구

A. 음성 데이터 감정 인식

1. 멀티 모달 감정 인식

a. 음성과 생체인식 멀티 모달 데이터

최근 인공지능 기술의 발전으로 정확하고 신뢰성 높은 감정 인식을 위한 다양한 방법이 연구되고 있으며 텍스트, 생체 신호 등 다양한 모달의 데이터에서 특징을 추출하여 사용하고 있다. 하지만 외부 환경의 영향을 많이 받아 고려해야 할 요소가 많은 감정 인식 분야에서 이처럼 단일 데이터만을 활용하는 것은 정확도 향상에 한계가 있다.

[7]의 연구에서는 이와 같은 문제의 해결을 위해 심박 수와 음성 데이터를 결합한 멀티 모달 방식의 모델 방법론을 제안하였다. 심박 수는 연구 참여자의 웨어러블 기계를 통해 수집하였으며 데이터 각각의 시간 정보를 사용하고 데이터의 분포 조정이 가능한 Sliding과 Window 알고리즘을 사용하였다. 음성 데이터는 CREMA, SAVEE, TESS, RAVDESS 공개 데이터 셋을 사용하였으며 감정의 종류는 기쁨, 중립, 슬픔, 화남, 두려움, 혐오, 놀람 총7개로 분류하였다. 하지만, 연구에서 사용한 심박 수와 음성 데이터는 서로 다른 특성을 가지며, 수집 시점과 시간 축도 다르기 때문에 입력 데이터 융합과 모델 출력 융합은 어려울 수 있다. 그래서 각 모달에서 추출된 특징을 결합하는 특징 수준 결합과 후기 융합 방법을 사용하였다. 두 방법은 모달 간 특징 형태가 다르더라도 사용 가능하며 결합 공간이 독립적인 여부에 따라 차이가 있다[7]. 또한, 심박 수 데이터를 직접 수집하여 검증 데이터로 활용하였고 단일 데이터 입력을 사용하는 분류 모델에 다양한 기술을 도입하여 정확도와 신뢰도를 향상시켰다. 후기 융합 형태의 멀티 모달 융합 방법은 음성 데이터만을 분석한 해당 모델을 비롯한 텍스트, 영상 모델을 독립적인 형태로 앙상블하여 추후 감정 인식 성능 향상에 도움을 줄 수 있다.

b. 음성과 텍스트 멀티 모달 데이터

정확한 감정 인식을 위해서는 효율적인 특징 추출과 적절한 특징 분류기의 선택이 중요하다. [8]의 논문에서는 음성과 텍스트를 결합하여 감정 인식을 향상시키는 방법을 제안하였으며 한국어 감정 음성 데이터를 통해 Angry, Happy, Sad, Neutral 총 4가지 감정 카테고리로 실험하였다. 데이터 전처리 과정으로는 음성 구간을 추출하고 각 데이터에서 특징을 추출하여 43차원의 특징 벡터를 생성하였다. 텍스트의 경우 한국어의 특성을 고려하여 자모 단위로 토큰화하고 각 토큰을 임베딩하여 특징 벡터를 형성하였기 때문에 이러한 특징 벡터를 LSTM(Long Short-Term Memory) Layer와 Fully Connected Layer를 통해 처리하고 Softmax 함수를 사용하여 감정을 예측하였다. 최종 예측은 각 감정에 대한 Softmax 결과 값의 평균으로 도출하였다. 이 연구 방법은 다른 음성과 텍스트 기반 멀티 모달 감정 인식 방법보다 95.97%의 더 높은 정확도를 나타냈다. 한국어 텍스트 처리와 임베딩 연구 관련 연구는 영어 기반 연구에 비해 연구의 수나 질에서 미비하고 감정 인식 성능도 낮다. 따라서 이 연구와 같은 한국어 텍스트 처리 및 임베딩 연구를 통해 향후 더 높은 감정 인식 성능을 기대할 수 있으며, 음성과 텍스트 정보를 결합하여 대화 인터페이스 기술에도 긍정적인 영향을 미칠 것이다.

2. 학습 모델 기반 감정 인식

음성 인식 기술에서 일반적으로 사용하는 신경망은 연산과 메모리 요구가 많아 휴대용 기기나 실시간 처리에서 어려움이 있었다. [9]의 연구에서는 작은 신경망을 사용하면서도 높은 음성 인식 성능을 얻기 위해 음성의 노이즈 특성을 활용하였다. 심층 신경망을 사용하여 잡음이 포함된 음성 신호로부터 배음 신호를 생성하고 스펙트럼 포락선 신호를 추정하여 음성 인식의 성능을 향상시켰다. 사용되는 심층 신경망은 배음 신호 생성과 스펙트럼 포락선 신호 추정을 위한 두 개의 부분으로 나누었다.

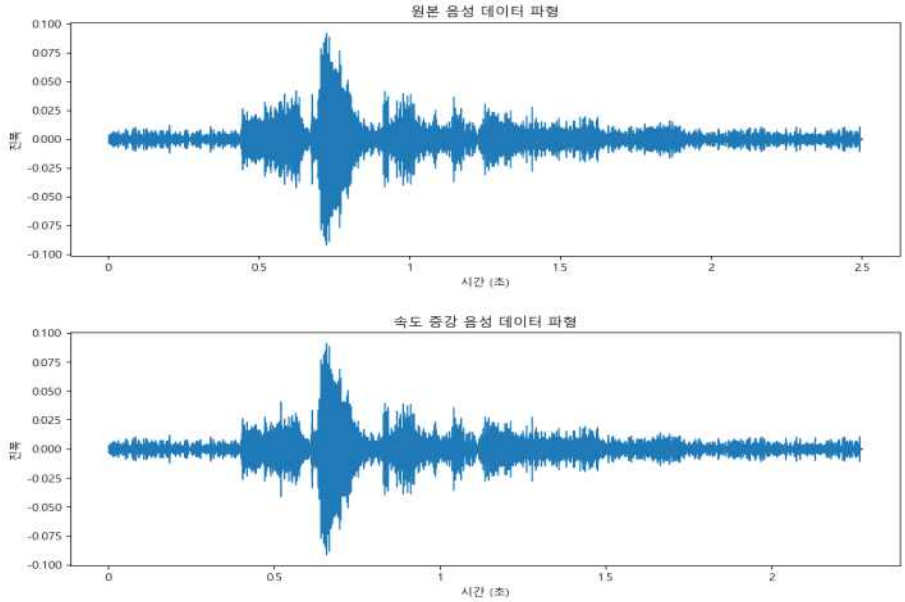
이 신경망은 모두 합성곱 신경망인 CNN(Convolutional Neural Network)으로 구성하였고 잡음이 제거된 최종 스펙트로그램을 생성하기 위해 배음 신호

와 스펙트럼 포락선 신호를 곱하여 사용하였다. 제안된 신경망은 스펙트로그램의 다양한 특성을 고려하며, 합성곱 계층과 합성곱 오토인코더를 사용하여 음성 향상을 수행하였다. 이로써 시간 영역으로 변환된 스펙트로그램을 얻을 수 있다. 이 연구 방법은 성능 평가 과정에서 다른 기존 기술들인 Wiener, SEGAN, Wavenet과 비교했을 때 작은 신경망 크기를 가지면서도 높은 음성 향상 성능을 보여주었으며, 복원된 음성의 품질에서 우수한 결과를 도출하였다. 이처럼 음성 데이터에서 효율적으로 정보를 추출하여 실시간 연산이나 처리하거나 머신러닝이나 파라미터 조절함으로서 연산을 줄여 모델의 경량화를 구현할 수 있다.

B. 음성 데이터 증강 기법

1. 속도 조정

음성 데이터의 속도 증강은 [그림 2-1]과 같이 음성 데이터를 다양한 방법으로 조정하여 데이터를 증강하는 기법이다.



[그림 2-1] 속도 조정 음성 파형 비교

[10]의 연구에서도 적은 데이터 셋 환경에서 효과적인 학습을 위해 방대한 훈련 데이터 셋을 증강한 모델 훈련 간 직관적이고 간단하지만 효과적인 여러 데이터 증강 기법을 소개하였다. 속도를 조정한 증강 기법은 음성 신호의 재생 속도를 조절하여 음성의 지속시간을 변경하는 기술로, 주로 음성 신호의 시간 축을 늘리거나 줄여서 작동한다. 시간을 늘리면 음성은 느리게 재생되고, 시간을 줄이면 빠르게 재생되어 음성 학습 데이터를 다양화하고 모델의 일반화 능력 향상시킬 수 있다.

[11]의 연구에서는 심층 신경망(DNN) 기반 음향 모델링을 사용하여 음성 인식 성능 향상을 위한 음성 증강 데이터 구현에 관한 연구를 진행하였다.

제안된 방법은 음성 신호의 속도를 변경하여 속도 계수를 0.9, 1,0, 1.1로 조정하였다. 다양한 데이터 시나리오에서의 음성 데이터 증강 효과를 제시하기 위해 100-1000시간 범위의 학습 데이터를 사용하여 LVCSR(Large Vocabulary Continuous Speech Recognition) 처리 기술에 대한 연구를 진행하였다. 그 결과, 훈련 데이터의 양에 관계없이 LVCSR 처리 기술에서 평균 4.3%의 성능 향상이라는 긍정적인 효과를 보였다. 해당 연구는 복잡한 방식의 증강 기법이 아닌 속도 변화로 긍정적인 효과를 도출하여 데이터 증강 기법의 효율성을 인지할 수 있었다.

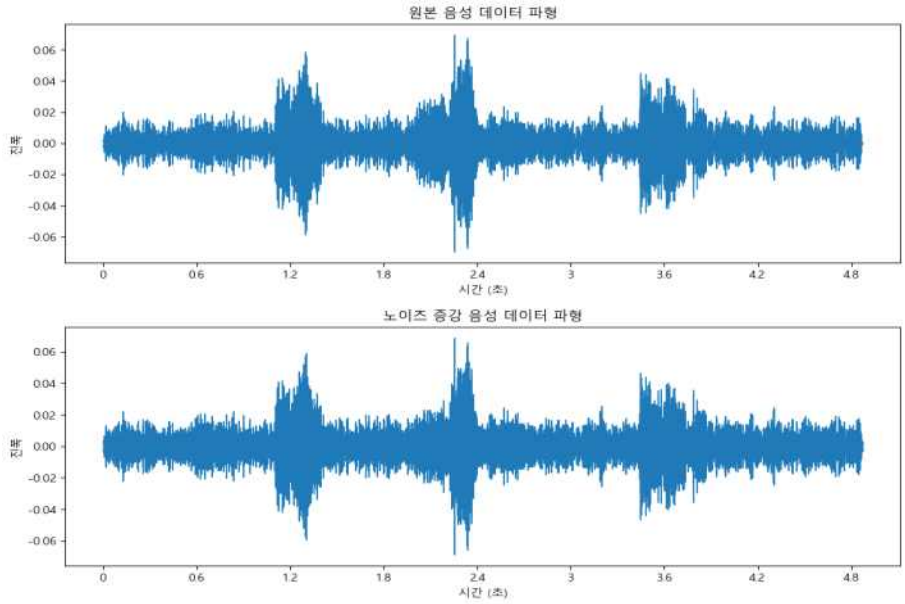
위의 연구와 비슷하게 [12]의 연구에서도 음성 속도와 발화 길이를 모두 조정하여 음성 데이터를 증강하여 기존의 제한된 환경에서보다 많은 학습 데이터를 수집하였다. 음성 속도는 발화 전체의 속도 대신 임의의 스펙트로그램 구간에서 프레임 속도를 조정하여 학습하는 것을 중점으로 하였다. 이는 데이터 증강의 정도를 조절하기 위해 속도 변경 정도나 구간 길이를 가변적으로 조절한 것이다. 그리고 무작위로 선택된 파라미터를 활용하여 특정 구간의 프레임 속도를 조절하였고 발화의 길이에 따라 일관된 증강 정도를 적용하기 위해 발화 길이에 따라 다른 증강 범위를 조정하였다. 그 결과, 하나의 발화 길이의 절반을 최대 증강하였을 때 가장 높은 성능이 도출되었다[12].

위의 연구처럼 다양하게 속도를 조정하여 데이터를 생성함으로써 데이터 증강의 긍정적인 효과를 확인할 수 있었다. 이러한 연구 결과는 데이터가 왜곡 되지 않은 수준에서의 데이터 증강은 효율적이며 복잡한 방식이 아닌 방법으로도 데이터를 수집할 수 있음을 보여주었다.

2. 노이즈 추가

노이즈 추가는 음성 신호에 노이즈를 추가하여 환경 소음이나 녹음 조건의 변화에 대응하는 모델을 훈련하는데 사용되며 노이즈가 있는 환경에서도 정확한 예측을 수행할 수 있도록 도움을 준다. 자연 소음(백색 소음, 바람 소음) 또는 다른 환경 소음(카페, 거리, 음악 등)에서 추출되거나 생성될 수 있으며 이 노이즈는 음성 신호의 주파수 영역에서 생성된다. 노이즈의 특성 및 강도를 조절하여 모델을 일반 환경에서 더 작동하도록 만들어줄 수 있으나 과도할 경우에는 원본 음성 정보가 손실될 수 있으므로 적절한 노이즈 세기

를 선택하여 균형을 유지하도록 해야 한다. [그림 2-2]는 기존 음성 데이터와 노이즈를 추가한 음성의 파형을 비교한 것이다.



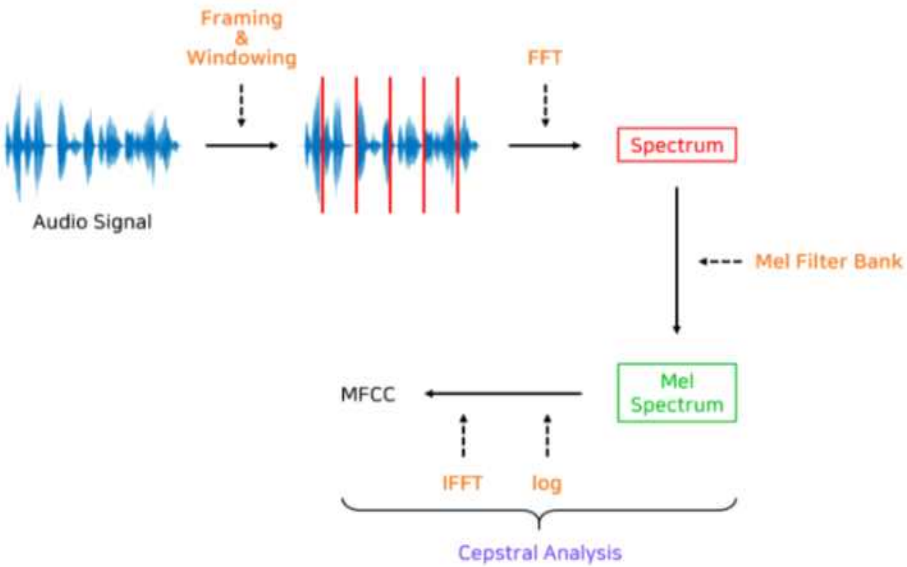
[그림 2-2] 노이즈 추가 음성 파형 비교

[13]의 연구에서는 자연스러운 음성을 합성을 위해 음성합성 단계에서 출력되는 음성 중 각 음소 위치를 결정하는 역할을 하는 어텐션을 기반으로 노이즈를 추가하여 데이터를 합성하는 방법을 제안하였다. 안정적인 모델 개발을 위해 풍부한 학습 데이터에 대해 학습해야 한다고 주장하였으며 딥러닝 분야에서 노이즈 추가를 활용한 학습 데이터 보완 방법이 널리 사용된다고 하였다. 또한, 노이즈가 포함된 입력 데이터로 딥러닝 모델을 학습하면 모델은 정상적인 결과를 출력하도록 학습될 수 있고 이는 학습에 참여하지 않은 입력 패턴에 대해서도 강력한 성능을 보일 수 있음을 주장하였다. 이를 통해 데이터가 왜곡되지 않는 정도의 노이즈 증강은 모델의 일반화를 높여줄 수 있으므로 증강 기법으로 사용이 가능하다.

C. 특징 벡터 추출 방법

1. MFCC

MFCC(Mel-Frequency Cepstral Coefficients)는 음성 및 음향 신호 처리 분야에서 중요한 특징 벡터로 널리 사용되며, 주로 음성 신호의 스펙트럼 특성을 분석하고 모델링 하는 데 활용된다[14]. MFCC는 인간 청각 시스템의 특성을 모방하여 음성 신호의 중요한 정보를 추출하는 데 사용한다. 다음 [그림 2-3]는 MFCC 추출 과정을 나타낸다.



[그림 2-3] MFCC 추출 과정[15]

[그림 2-3]과 같이 MFCC 추출은 음성 신호의 주요 특성을 추출하기 위한 단계적인 과정으로 진행된다. 먼저, 음성 신호에 정보 손실을 줄이기 위해 전처리 필터를 적용하여 높은 주파수 성분을 강조한다. 그리고 음성 신호를 작은 프레임으로 나누고 각 프레임에 윈도우 함수를 적용하는 STFT(Short-Time Fourier Transform) 또는 FFT(Fast Fourier Transform)를 수행하여 주파수 영역으로 변환한다. 주로 FFT가 사용되지만 STFT를 사

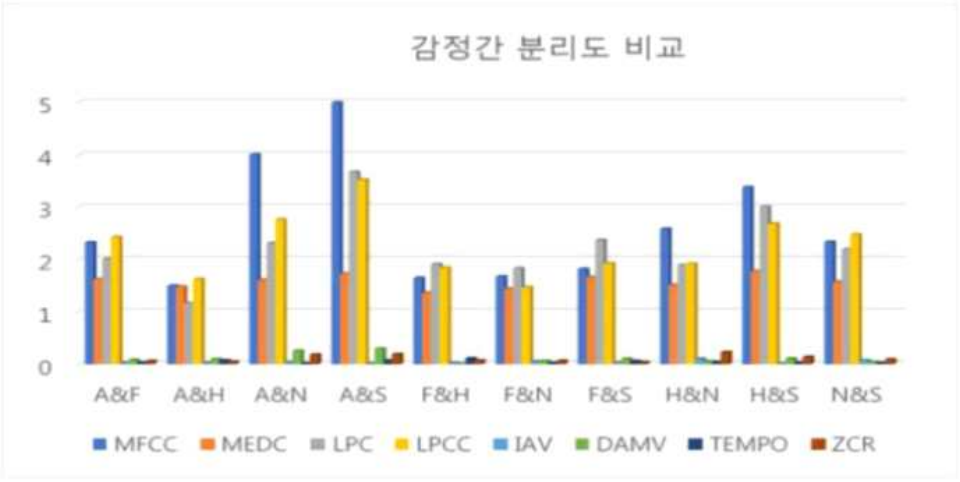
용하여 대신 수행할 수 있다. STFT는 시간 도메인에서 주파수 정보를 추출하는 신호 처리 기술로 시간을 작은 창(프레임) 단위로 나누어 각 창에서의 주파수 구성을 분석한다. 음성 처리, 음악 분석, 이미지 처리 및 다양한 신호 처리 응용에서 사용되며 시간-주파수 표현을 얻을 수 있어 신호의 주파수 특성이 시간에 따라 어떻게 변하는지 분석하고자 할 때 유용하다[16]. 이와 같은 방법으로 주파수 스펙트럼을 얻게 되며, 멜 스케일 필터 뱅크를 적용하여 주파수 스펙트럼을 인간 청각 시스템에 맞게 모델링한다. 그 후, 로그 스케일로 변환하여 높은 주파수 성분에서의 에너지 차이를 줄이고, 이산 코사인 변환을 적용하여 최종 MFCC 계수를 얻는다[6]. 이러한 계수는 각 프레임에 대한 음성 신호의 중요한 특징을 나타내며, 전체 음성 신호에 대한 MFCC 특징 벡터를 생성한다.

음성 특징 벡터를 사용하여 감정 인식 효과적으로 분류하기 위한 연구는 꾸준히 진행되고 있다. [17]의 연구에서는 유사 감정 음성 데이터를 효과적으로 분류하기 위한 최적의 특징 벡터를 조사하였다. 그리고 Bhattacharyya 거리 측정을 활용하여 각 감정 그룹에 가장 적합한 특징 벡터를 결정하였다. Russell이 제안한 Circumplex Model 방법을 사용하여 유사 감정 그룹을 선정하였다. 아래 [그림 2-4]는 2차원 감정 분류에 대한 그림으로, 선정된 그룹은 (Happy, Surprise), (Fear, Anger), (Sad, Disgust)이며, 각 그룹 간 적합한 특징 벡터 추출을 위해 분리도를 계산하여 분류 정확도를 측정하였다.

[그림 2-4] 2차원 감정 분류[17]

[그림 2-4]와 같이 실험 결과로 특정 감정 그룹에 대한 특정 특징 벡터 조합이 감정 분류의 정확도를 향상시키는 것을 확인하였다. 특히, (Fear, Anger) 그룹에서는(MFCC + Chroma + Spectral Contrast) 조합이 가장 적합한 특징 벡터로 나타났으며 (Disgust, Sad) 그룹과 (Surprise, Happy) 그룹에서 독립적인 MFCC 적용 방법이 가장 효과적인 것으로 나타났다. 이 연구 결과는 유사한 감정 간 인식 기술의 발전 가능성을 제시하였으며 향후 연구에서 서로 다른 감정 그룹에 따라 다른 특징 벡터를 사용하는 것이 실제로 어떤 영향을 미치는지에 대한 더 깊은 연구가 필요하다고 제언하였다.

이와 비슷한 [3]의 연구에서도 Bhattacharyya 거리 측정법을 사용하였으며, IEMOCAP_DB 데이터 셋에서 주로 사용되는 대표적인 감정인 화남(Anger), 즐거움(Happy), 두려움(Fear), 평범(Neutral), 슬픔(Sad)의 5가지 감정을 정의하고 분석하였다. 각 감정이 혼합되어 있는 동일한 구간에서 여러 특징 벡터 추출 방법을 사용하여 특징 값을 추출하였다. 이렇게 추출한 특징 벡터들의 분리 정도를 비교하기 위해 Bhattacharyya 거리 측정법을 사용하여 평균값을 분석한 결과 [그림 2-5]과 같이 결과를 도출하였다.



[그림 2-5] 감정 카테고리 간의 분리도 비교[3]

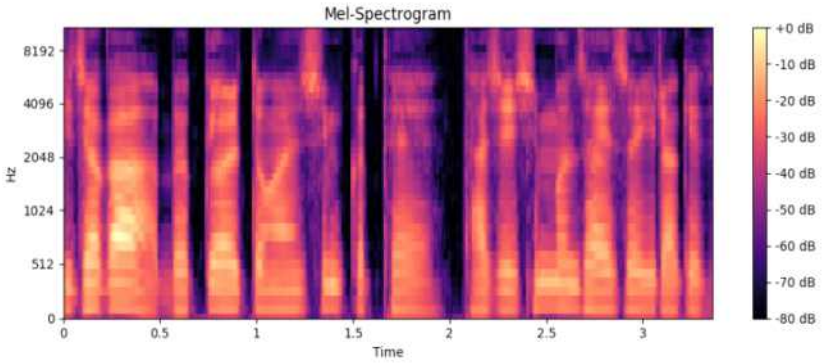
[그림 2-5]과 같이 MFCC를 활용한 감정간 분리도 비교 실험에서 전반적으로 높은 비교 결과가 나타났으며 화남과 중립, 화남과 슬픔처럼 대립되는

결과의 분리도가 가장 높은 것을 알 수 있다. 향후 이 연구의 실험 결과를 기반으로 적절한 특징 벡터 조합과 정확한 분류 엔진의 병렬 사용을 통해 더 정확한 감정 인식 성능을 기대할 수 있다.

위의 연구들을 통해 음성 벡터를 추출하는 여러 방법을 사용하여 정확도를 개선하는 방법을 알 수 있었으며 음성 속도, 길이 변화 등 다양한 환경의 증강 데이터를 생성하고 여러 특징 벡터 추출 방법으로 사용하여 결과를 비교하고 음성 인식 성능을 개선할 수 있을 것이다.

2. Mel-Spectrogram

Mel-Spectrogram(멜 스펙트로그램)은 음성 데이터에서 특징 벡터를 추출하는 주요 방법 중 하나이다. 음성 신호는 시간에 따라 변화하는 복잡한 주파수 성분을 가지고 있기 때문에 이 주파수 성분을 더 효과적으로 분석하기 위해 주파수 스케일을 멜 스케일로 변환하여 [그림 2-6]과 같이 나타낼 수 있다. 이는 인간 청각 시스템의 특성을 반영한 것으로, 높은 주파수에서는 세세한 주파수 변화를 미미하게 감지하고, 낮은 주파수에서는 주파수 변화를 상대적으로 더 정확하게 감지하도록 한다.



[그림 2-6] Mel-Spectrogram 예시[18]

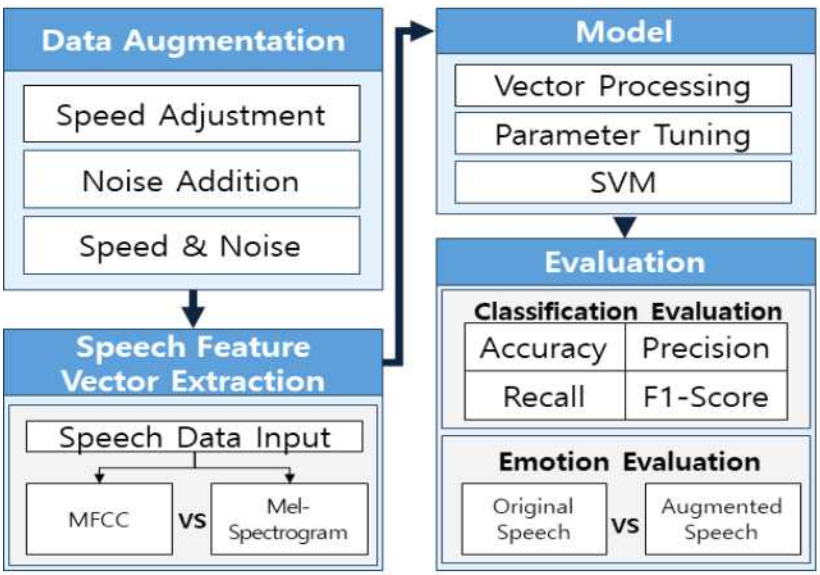
이와 같은 단계는 먼저, 음성 데이터를 20-30 ms 길이의 프레임 단위로 분할하며 이 프레임은 Mel-Spectrogram의 시간 축을 형성한다. 각 프레임 내의 음성 신호를 주파수 영역으로 변환한다. 대표적으로 FFT를 사용하여 시

간 도메인의 신호를 주파수 도메인으로 변환하면 해당 프레임에서의 주파수 성분을 얻을 수 있다. FFT를 통해 얻은 주파수 스펙트럼을 멜 스케일로 변환하고 멜 필터 뱅크를 적용한다. 멜 스케일 변환은 주파수를 멜 스케일로 매핑하는데 사용되는데 이는 인간 청각 시스템의 주파수 분해 능력을 모방한 것이다. 멜 필터 뱅크는 멜 스케일 상에서 주파수 대역을 에너지로 매핑하며, 여러 개의 삼각형 모양 필터로 구성되며 이 필터들은 주파수 스펙트럼을 여러 가지 주파수 대역으로 분할된다. 마지막으로, 각 멜 필터의 에너지 값을 로그 변환하여 음성 신호의 주파수 성분의 에너지 차이를 줄이고 더욱 민감한 특징을 추출할 수 있게 된다. MFCC 방식과 Mel-Spectrogram은 비슷한 성격을 띄고 있으나 주파수 스펙트럼을 멜 스케일로 변환하여 표현한 것으로 Mel-Spectrogram은 주파수 스펙트럼의 표현으로 구분할 수 있다. 이와 마찬가지로 [19]의 연구에서는 CNN 기반 학습 모델을 통해 감정 인식 성능을 평가하였으며 Mel-Spectrogram과 유사한 ERB(Equivalent Rectangular Bandwidth) Spectrogram과 Log Mel-Spectrogram을 활용하였다. 두 방법 모두 필터 뱅크와 스펙트로그램을 활용하여 음성을 처리하고 비교하여 Resnet 모델을 기반으로 음성 감정 인식 연구를 진행할 때는 더 긍정적인 결과를 나타내는 방법은 Log를 사용한 것이라는 연구를 진행하였다.

Ⅲ. 속도와 노이즈 증강 데이터를 통한 음성 감정 인식 성능 향상

A. 연구 구성도

음성은 사람의 감정을 인식하는 중요한 정보를 제공하며 최근 사람의 감정 인식에 관한 연구가 다양하게 진행되고 있다. 외부 환경에 대해 영향을 많이 받는 음성의 특성 상 다양한 상황에서의 많은 음성 학습이 필요하기 때문에 기존 데이터가 부족한 문제를 해결하기 위해 데이터가 왜곡되지 않는 범위에서 데이터 증강을 통해 음성 감정 인식 성능을 향상시키고자 한다. 따라서 본 연구에서는 음성 감정 인식 성능 향상을 목표로 음성 데이터에 적합한 다양한 데이터 증강과 특징 벡터 추출 방법을 제안하였다. 제안하는 방법은 아래 [그림 3-1]과 같이 모델이 실제 일반화 환경을 대처할 수 있도록 속도 조정, 노이즈 추가와 같은 데이터 증강 기법을 적용하였으며 MFCC와 Mel-Spectrogram 특징 벡터 추출 방법으로 정확도를 비교·분석하였다.



[그림 3-1] 연구 구성도

위의 [그림 3-1]과 같이 먼저, 원본 데이터를 기반으로 여러 증강 데이터를 생성하고자 속도 조정, 노이즈 추가, 원본·증강 데이터 간 병합, 증강 데이터 병합 총 4개의 데이터 셋을 구성하여 실험하였다. 특징 벡터 추출 방법으로는 MFCC와 Mel-Spectrogram을 적용하였다. 두 가지 방법으로 벡터를 추출하면 데이터 분석 작업 전에 데이터를 준비하고 가공하여 모델이 이해할 수 있는 형태로 전처리하였다. 전처리 과정으로는 크게 음성 벡터 최대 프레임 제한, Zero-Padding, 차원 축소로 총 세 단계의 과정을 진행하였다. 전처리가 완료된 데이터는 SVM을 사용하여 학습을 진행하였으며 머신러닝 모델의 선택은 연구 환경과 데이터 셋의 특성에 따라 결정하였다. SVM을 선택한 이유는 데이터 셋의 크기와 자원의 제한 때문이다. 실험에서 사용할 데이터 셋이 다른 대규모의 데이터 셋보다 상대적으로 적었으며, 한정된 컴퓨터 자원을 활용하였기 때문에 적은 규모의 데이터 셋에서도 강력한 성능을 보이고 감정 인식과 같은 데이터 포인트가 다른 감정 카테고리에 속할 확률을 정확하게 분류할 수 있는 모델이 SVM이기 때문이다. SVM은 분류 작업에 가장 적합한 알고리즘 중 하나로 전처리에서 진행했던 3가지의 과정 모두 연관이 있다. 데이터의 일관성을 확보하고 정규화하여 SVM이 모든 특징 벡터를 고르게 다룰 수 있도록 하였다. 또한, SVM은 데이터를 잘 분리하는 초평면(Hyperplane)을 찾는 데 강력한 능력이 있다. 정규화된 데이터에 대해 SVM을 사용하면 감정 분류 작업에서 높은 정확도를 얻는 것과 더불어 커널 종류 및 C 매개변수와 같은 하이퍼 파라미터를 조정하여 모델의 성능을 개선 가능한 여러 옵션을 사용할 수 있다.

따라서 전처리를 통해 데이터 일관성과 정규화를 확보하고 모델을 학습함으로써, 음성 감정 인식 모델은 강력한 분류 능력과 일반화 능력을 갖출 수 있다. 이 연구는 음성 인식 분야에 효과적으로 적용 가능한 방법론을 제시하며 환경 변화에 민감한 응용 분야에서 좋은 성능을 발휘함을 목표로 하였다.

B. 음성 데이터 증강

본 연구에서는 원본 데이터를 기반으로 다양한 증강 데이터를 생성하고자 한다. 증강 기법으로 속도 조정, 노이즈 추가 방법을 사용하였으며 각 증강 기법의 데이터를 비롯하여 원본·증강 데이터 병합, 증강 데이터 병합으로 크게 4종류의 데이터 셋을 구성하였다. 범위는 원본 데이터의 배율은 1.0배율을 제외한 0.9-1.1 배율로, 각 간격은 0.02로 설정하여 0.9-0.98, 1.02-1.1이다. 예를 들어 0.9-0.98 데이터 셋에 포함되는 데이터 배율은 0.9, 0.92, 0.94, 0.96, 0.98으로 각 배율을 원본 데이터에 적용하여 각 배율의 데이터 셋에서 총 5종을 수집하였다. 마찬가지로 노이즈 추가는 진폭(Amplitude)을 2-10까지 2세기 간격으로 설정하였으며 노이즈의 세기 범위는 2, 4, 6, 8, 10에 해당한다. 다음은 제안하는 네 가지 주요 데이터 증강 기법을 상세한 설명하고자 한다.

1. Numpy 기반 속도 증강

첫 번째는 Numpy 기반 속도 증강 기법이다. 일반적으로 데이터의 자연스러움을 유지하고 다양성 확보를 위해 데이터가 왜곡되지 않는 범위에서 배율을 선택하였다. 많은 속도 증강 연구에서 자주 사용하며 연구의 실험과 검증을 기반으로 [11]의 연구와 같이 음성 신호의 속도를 변경하여 데이터를 증강했던 점과 유사한 방법을 참고하여 음성 속도는 0.9, 1.1 배율로 조정하였하였다. 그리고 더 세분화하여 음성 속도를 원본 데이터 셋의 배율인 1.0을 제외하고 0.9-1.1배율까지 0.02 간격으로 총 10가지의 종류로 설정하였다. 다음 [표 3-1]은 속도 증강 코드 예시이다.

[표 3-1] 속도 증강 코드 예시

```
# 시작 배율 및 종료 배율 설정
start_Scale = 0.9
end_Scale = 1.1
Scale_interval = 0.02

# 입력 폴더 내의 모든 오디오 파일 찾기
for filename in os.listdir(input_folder):
    if filename.endswith(".wav"):
        # 입력 오디오 파일 경로
        input_audio_path = os.path.join(input_folder, filename)
        # 오디오 파일 이름에서 확장자 제거
        name, ext = os.path.splitext(filename)
        # 배율을 조정하면서 저장
        for Scale in np.arange(start_Scale, end_Scale, Scale_interval):
            # 배율을 파일 이름에 추가
            Scaled_filename = f"{Scale:.2f}_{name}.wav"
            # 출력 오디오 파일 경로
            output_audio_path = os.path.join(output_folder, Scaled_filename)
            # 입력 오디오 파일 열기
            audio, sample_rate = sf.read(input_audio_path)
            # 음성 속도 조절
            resampled_audio = np.interp(
                np.arange(0, len(audio), Scale),
                np.arange(0, len(audio)),
                audio
            )
            # 결과 저장
            sf.write(output_audio_path, resampled_audio, sample_rate)
```

[표 3-1]과 같이 먼저, Os, Soundfile, Numpy 모듈을 Import 한다. Os 모듈은 파일 및 폴더 관리를 위한 함수를 제공하며 Soundfile 모듈은 오디오 파일을 읽고 쓰는 데 사용되고 Numpy 모듈은 숫자 배열 조작을 위한 기능을 제공한다. 다른 라이브러리나 방법을 사용하여 음성 속도를 변화시킬 수도 있으나 Numpy 모듈을 활용한 증강은 다음과 같은 장점이 있다. 첫째, 간단하고 가볍다. Numpy를 사용하면 추가적인 라이브러리나 의존성을 도입할 필요가 없으며 코드가 가벼워진다. 또한, 데이터 조작을 빠르게 처리할 수 있어 데이터를 증강하는 시간을 단축시킬 수 있다. 둘째, 선형 보간 기능을 제공한다. Np.Interp 함수를 사용하여 선형 보간을 수행할 수 있다. 이는 기존 오디오 데이터를 새로운 배율에 맞게 조정하는 데 유용한 방법 중 하나로, 선형 보간은 간단하면서도 효과적으로 음성 속도를 조절할 수 있다.

다음은 음성 파일 변환을 수행하는 스크립트로, 입력 및 출력 폴더 설정과 함께 시작 배율부터 종료 배율까지 정의한 배율 차이로 음성 파일을 변환하는 작업을 수행한다. 이후 시작 배율(Start_Scale), 종료 배율(End_Scale), 그리고 배율 간격을 설정하여 오디오 배율을 조절할 범위와 간격을 정의한다.

다음으로 입력 폴더 내의 모든 파일을 반복하며 .WAV 확장자를 가진 입력 음성 파일을 열고 음성 데이터의 속도를 적용한다. Np.Interp 함수를 사용하여 음성 속도를 조절하고 완료 시 지정된 출력 경로에 저장하도록 한다. 모든 파일에 대한 변환 작업이 완료되면 메시지가 출력된다. 원본 데이터는 6742개로, 0.9배율과 1.1배율에 포함되는 데이터 수는 원본 데이터의 5배씩 증강하여 각각 33,710개에 해당한다.

Numpy 기반 속도 증강 기법 이외에도 대표적으로 Pydub 라이브러리의 Audiosegment 클래스가 있다. 주로, Audiosegment의 Speedup 메서드는 주어진 속도 배율에 따라 음성 스트림의 길이를 조절하고 이 과정에서 소리의 높낮이와 음성의 빠르기를 동시에 조절할 수 있다. 음성 데이터를 빠르게 만들 때, 음성 스트림의 시간을 압축하는 것이며, 주파수와 음의 높낮이가 동시에 상승한다. 상대적으로 작은 속도 상승에서 주파수 변화가 인지하기 어렵고 피치 변화에도 덜 두드러지기 때문에 1.0 이상의 속도 배율에서는 문제가 발생하지 않는다. 그러나, 0.9 이하의 느린 배속에서는 시간을 늘리는 과정에서 주파수와 음의 높낮이가 낮아질 수 있으며, 이로 인해 소리가 이상하게 늘어진다. 따라서 느린 배속에서 발생하는 길이 확장을 보완하기 위해서는

중간에 추가적인 소리를 삽입해야하며 이러한 추가적인 소리는 원본 소리와 일관성 있게 조절되어야 한다. 그래서 속도 배율 간에 자연스러운 소리를 생성할 수 있도록 Numpy 기반의 증강 기법을 사용했다.

2. Audiosegment 기반 노이즈 증강

두 번째 방법은 노이즈를 추가하는 방법으로 진폭(Amplitude)과 길이(Length) 차이를 통해 얼마나 강렬하게 음성에 덧붙이는지, 얼마나 오랫동안 영향을 주는지에 따라 다르게 생성할 수 있다. 노이즈 증강에 사용하고자 하는 방법은 Pydub 라이브러리의 Audiosegment 기반이다. Audiosegment는 음성 데이터를 쉽게 다룰 수 있는 라이브러리로서, 음성 자르기, 볼륨 조절, 음성의 높낮이 조절 등 다양한 작업을 수행할 수 있다. 또한, 특정한 형태의 노이즈를 생성하고 음성에 오버레이가 가능하고 노이즈의 강도와 지속 시간을 조절할 수 있어서 증간 시 유리하다.

데이터 증강 간에 노이즈 세기와 노이즈의 지속 시간 정도를 선택할 수 있다. 노이즈 추가 간에 지속 시간과 강도에 따라 여러 데이터 셋을 구성할 수 있어 가장 효율적이고 왜곡되지 않는 방법을 고려하고자 하였다. 먼저, 노이즈 세기 설정 간 높은 세기의 노이즈를 추가하게 되면 원본 음성 데이터의 학습을 방해할뿐더러 진폭의 크기가 증가하기 때문에 음성 데이터의 소리가 작거나 소리가 낮은 형태의 대화문은 잘 인식하지 못하는 데이터 왜곡이 발생하지 않도록 주의하였다. 데이터 자체가 왜곡되지 않으면서 정확도가 낮아지지 않는 정도의 데이터를 주고자 최대한 낮은 세기의 노이즈를 추가하여 데이터를 생성하였다. 예를 들어, 노이즈 세기를 0.2 이상부터 적용했을 때 음의 진폭이 높은 'Anger'와 'Happiness'의 전체적인 예측 값이 상승하여 높은 음의 감정은 물론 낮은 음의 감정도 잘 구분하지 못하였다. 그리고 노이즈 추가는 세기를 0부터 10까지 2세기 간격으로 설정하였으며 2, 4, 6, 8, 10의 세기로 증강하였다. 그리고 노이즈 지속 시간은 각 음성 데이터의 지속 시간과 일치시켜 데이터 왜곡을 최소화하였다. 물론, 노이즈를 특정 부분에만 주는 것은 실제 환경처럼 노이즈 패턴을 더 잘 학습하도록 돕거나 음성의 품질을 덜 훼손하여 품질을 보존할 가능성이 있다. 하지만, 노이즈의 시작 및

끝 지점, 강도 및 블렌딩을 조절하지 않으면 데이터 양이 적어 노이즈가 갑자기 시작하거나 끝나면 부자연스러울 수 있어 음성 전체에 노이즈를 적용하였다. 다음 [표 3-2]는 노이즈를 증강하여 데이터를 생성하는 예제 코드이다.

[표 3-2] 노이즈 증강 코드 예시

```
# 화이트 노이즈 생성 함수
def generate_white_noise(duration_ms, amplitude=''):
    noise = WhiteNoise().to_audio_segment(duration=duration_ms)
    # 노이즈에 진폭 조절 적용
    noise = noise - amplitude
    return noise

# 원본 음성 파일 목록 가져오기
audio_files = os.listdir(original_folder)

# 화이트 노이즈 생성
# noise_duration_ms = 2000  # 노이즈의 길이 (밀리초)
# generated_noise = generate_white_noise(noise_duration_ms)

amplitude = ''

# 노이즈 증강
for audio_file in audio_files:
    audio_path = os.path.join(original_folder, audio_file)
    original_audio = AudioSegment.from_file(audio_path)
    noise_audio = original_audio.overlay(generate_white_noise(len(original_audio)))
    augmented_file_path = os.path.join(augmented_folder, f"noise{amplitude}_{audio_file}")
    noisy_audio.export(augmented_file_path, format="wav")
```

[표 3-2]처럼 먼저, Csv 라이브러리를 사용하여 Csv 파일을 다루고, Pydub

라이브러리의 Audiosegment 메서드를 이용하여 음성 처리 작업을 수행한다. Generate_White_Noise 함수는 Duration_Ms, Amplitude 인자 값을 받아 노이즈를 생성하는 함수로 첫 번째 인수인 Duration_Ms는 백색 노이즈의 지속 시간을 밀리 초 단위로 설정하며, 두 번째 인수인 Amplitude는 백색 노이즈의 세기를 설정하는 매개변수로, 설정 값은 90-98이다.. 다음, Amplitude 변수는 백색 노이즈의 세기를 나타내며 백색 노이즈를 생성하고 원본 음성 파일에 추가하고자 사용하였다. 마지막으로, 원본 음성 파일에 백색 노이즈를 추가하는 노이즈 추가 증강 작업을 수행하며 Audio_Files 목록을 반복하여 각 원본 음성 파일에 대해 백색 노이즈를 생성하고 추가한 후, 그 결과를 파일로 저장하고 각 파일의 저장 경로는 백색 노이즈의 Amplitude 세기와 원본 파일 이름을 조합하여 생성하였다.

3. 음성 데이터 병합

a. 원본·증강 데이터 병합

데이터 셋을 구성할 때 원본·증강 데이터를 병합하면 다음과 같은 장점이 있다. 첫째, 데이터의 다양성을 확보할 수 있다. 원본 데이터와 증강 데이터를 병합하여 원본 데이터의 특징과 증강 데이터의 속도 배율 조절, 노이즈 추가 등 다양한 환경에서 수집된 데이터를 모델 학습에 활용할 수 있다. 둘째, 로버스트한 모델 훈련이 가능하다. 실제 음성 데이터는 다양한 환경에서 발생하며 소음, 다양한 발화 속도, 강도 등이 발생한다. 이러한 다양성과 비슷한 환경을 구성하기 위해 증강 데이터를 함께 사용하면 로버스트한 성격을 띄는 모델을 만들 수 있다. 셋째, 과적합을 방지할 수 있다. 현재 실험에서 사용하는 데이터 셋의 경우 데이터 양 자체가 많지 않아 적은 데이터 셋의 경우에 과적합 될 위험이 있다. 실제로 노이즈 크기를 올렸을 때 과적합되어 Anger와 Happiness의 경우 검증 단계에서 높은 정확도가 나왔으나 테스트 데이터로 정확도를 테스트 하였을 때 과적합 된 결과가 도출되기도 하였다. 따라서 다양한 데이터 셋을 구성하여 모델의 성능을 높이기 위해 각 데이터를 병합하고자 한다.

구성한 데이터 셋은 네 가지로 크게 음성 특징 추출 벡터 기반으로 구분하

였다. MFCC와 Mel-Spectrogram을 각각 사용하여 속도 배율 기반의 원본 데이터 셋과 노이즈 배율 기반의 원본 데이터 셋으로 구성하였다. 속도 배율의 경우, 0.9배율부터 1.1 배율까지 0.02 배율 간격의 데이터 셋에 원본 데이터를 포함하였으며, 노이즈 세기의 경우, 2세기부터 10세기까지 2세기 간격으로 데이터 셋에 원본 데이터를 포함하였다.

b. 증강 데이터 병합

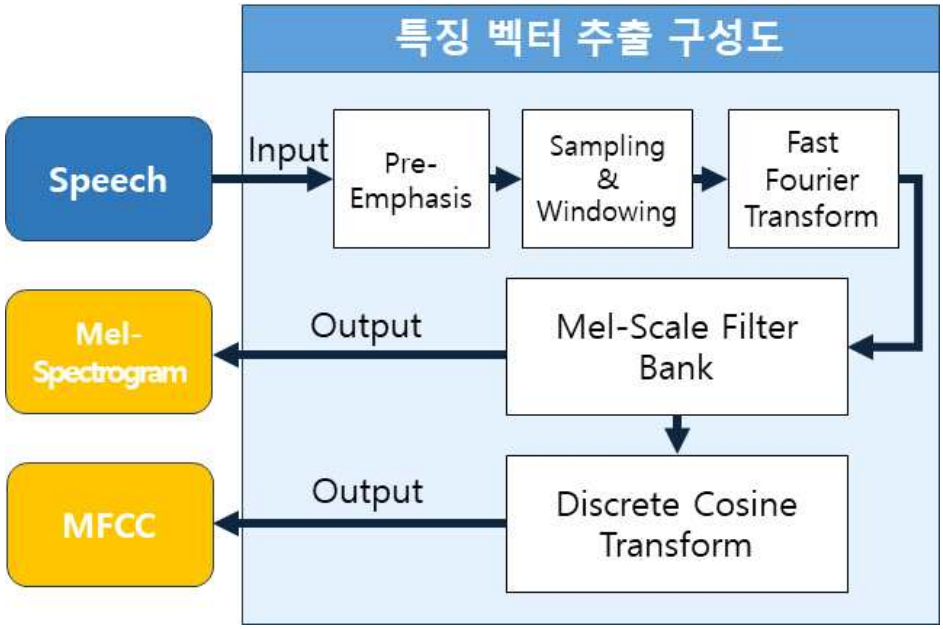
다음은 속도 배율과 노이즈 배율의 증강 데이터 병합을 통해 데이터 셋을 구성하였다. 증강 데이터 병합은 속도 및 노이즈 조건을 모두 데이터에 적용하여 기존의 단일 증강 기법이 적용된 데이터 셋보다 더 다양한 환경을 모델이 학습하게 하여 일반화 능력을 향상시키고자 하였다. 즉, 원본·증강 데이터 병합을 비롯한 기존 데이터 셋처럼 속도나 노이즈 증강 기법을 한 가지 적용한 데이터가 아닌 모두 적용하여 두 증강 데이터의 특징을 모두 포함하는 데이터 셋이다.

C. 음성 특징 벡터 추출

1. 데이터 전처리

데이터를 증강하였다면 음성 특징 벡터를 추출하기 위해서는 다음과 같은 데이터 전처리 과정이 필요하다. 먼저, 감정 레이블이 저장된 CSV 파일 경로와 음성 데이터가 들어있는 폴더의 경로를 설정한다. 감정 CSV 파일에서는 데이터 명(Wav_id) 칼럼과 감정(Emotion) 칼럼을 기준으로 화남, 행복, 중립, 슬픔, 두려움의 감정을 각각 리스트에 담는다. 그리고 감정 레이블을 리스트에 입력한 후 각 감정별 데이터의 개수와 종류를 확인하였다. 다음, 음성 데이터 폴더에서는 폴더 내 모든 WAV 파일의 데이터명을 추출하여 CSV 내 파일명과 동일한 음성 데이터명인지 비교한 다음 해당하는 WAV 파일만 리스트에 담는다. CSV 파일과 WAV 파일의 데이터명을 비교하여 리스트에 담는 것은 데이터 정제 및 필터링 과정이라 할 수 있다. CSV 파일과 WAV 파일 간 데이터와 레이블 간의 일치성이 유지되며, 데이터 분석 및 모델 학습 과정에서 정확한 감정 레이블을 사용할 수 있다. 또한, 레이블이 일치하지 않는 데이터를 걸러내고 필요한 데이터만 선택할 수 있다. 이는 추후 최대 프레임을 설정하여 최대 프레임 이상의 데이터를 걸러내는 작업에서 사용할 수 있다. 만약 데이터 명을 비교하지 않는다면 데이터가 걸러지면서 중간에 제외된 데이터의 위치에 다른 감정 데이터가 입력될 수 있으므로 반드시 확인해야 한다.

2. 특징 벡터 추출



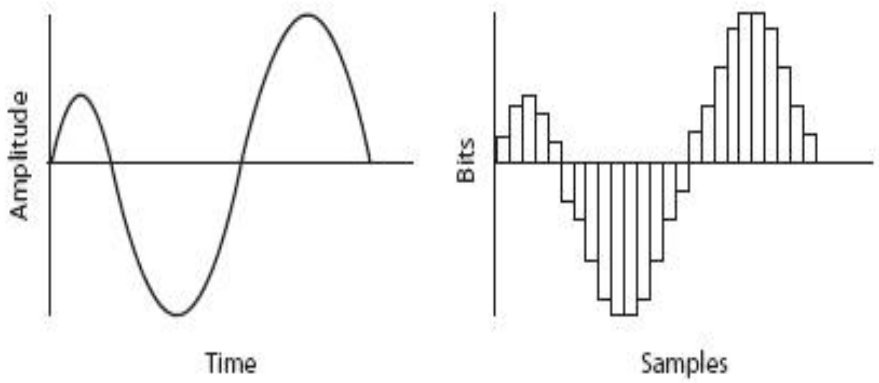
[그림 3-2] 특징 벡터 추출 구성도

a. MFCC

MFCC는 음성 신호 처리에서 주로 사용되는 기술로 음성 신호의 주파수 특성을 컴팩트한 형태로 변환하여 음성 처리 및 인식 작업을 수행하는 데 도움을 준다. 다음 [그림 3-2]는 각 특징 벡터의 추출 과정을 나열한 구성도이다. [그림 3-2]처럼 먼저, MFCC 추출 프로세스의 첫 번째 단계는 음성 데이터 로드로 음성 파일을 로드하여 이후의 신호 처리 및 특징 추출을 수행하기 위한 데이터를 얻는 과정이다. Librosa 라이브러리의 Load 함수를 사용하여 데이터를 로드하며 파일의 샘플링 레이트, 웨이브폼 및 기타 관련 정보를 반환한다. 샘플링 레이트(Sr)는 초당 샘플링된 점의 수로 음성 신호를 이산적인 형태로 표현하는 데 사용된다. MFCC 추출 및 다양한 음성 처리 작업에서 사용되는 파라미터 중 하나이다. 예를 들어, 44.1kHz의 샘플링 레이트는 초당 44,100개의 샘플을 의미하며, 이는 고음질 CD 오디오의 표준 샘플링 레이트이다. 즉, 시간의 흐름을 나타내며 얼마나 많은 데이터 포인트가 초당 오디오 신호에서 기록됐는지를 의미한다. 웨이브폼(Wf)은 시간에 따른 음성 신호의 진폭 정보를 포함하는 배열로, 이 데이터는 음성 신호의 시간 도메인 표현으

로 사용된다. 일반적으로 1차원 배열로 표현되며, 각 원소는 특정 시간의 오디오 신호 진폭을 나타내기 때문에 음성 신호의 모양과 변화를 나타낸다.

다음으로 프레임 분할 단계이다. 프레임 분할은 [그림 3-3]과 같이 음성 데이터를 작은 프레임으로 나누는 과정을 의미한다.

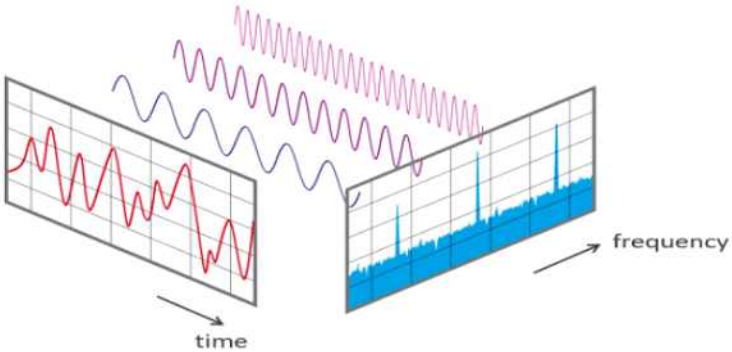


[그림 3-3] 웨이브폼과 샘플링 레이트 예시(좌:Wf, 우:Sr)

이는 MFCC 추출 및 다른 음성 특징 추출 작업을 수행하기 위한 필수적인 단계로 이전에 사용된 샘플링 레이트와 웨이브폼을 사용하여 아래 그림과 같이 프레임을 분할한다. Librosa 라이브러리의 Feature.MFCC 함수를 사용하였으며, 이 인자 값으로 n_fft, n_MFCC, hop_legnth가 필요하다.

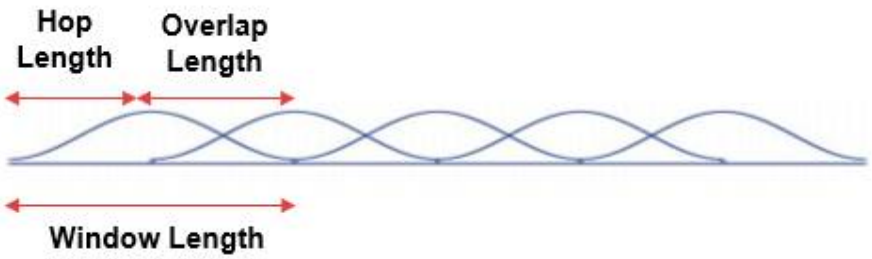
프레임 분할 과정에서 유의할 점으로 프레임크기(Frame size, n_fft)는 음성 신호를 분할 할 때 사용되는 윈도우의 크기로서 일반적으로 작은 값에서 시작하여 큰 값으로 사용하게 된다. 작은 프레임 크기는 빠른 변화를 포착할 수 있으나 주파수 해상도가 낮으며 반대로 큰 프레임 크기는 주파수 해상도를 높일 수 있지만 빠른 변화를 놓칠 수 있다. n_fft는 1024로 설정하였으며, 주파수 도메인에서 해상도와 계산 효율성을 고려하였기 때문이다. FFT는 주로 2의 제곱수 크기의 윈도우를 사용하는 것이 계산 효율성과 구현 편의성 측면에서 일반적이다. FFT 함수는 입력된 시계열 데이터에 FFT를 사용하여 주파수 성분으로 변환한다. 이는 주파수 도메인에서의 주파수 특성을 강조하는 도메인이다. 예를 들어, 음성 신호는 주파수로 구성된 파형으로 표현되지만 FFT를 사용하면 음성 신호의 주파수 성분을 추출하고 분석할 수 있다.

FFT의 앞 자리인 Fast라는 의미의 '고속'에서 알 수 있듯 신속한 연산을 통해 도메인 변환을 수행하며 기존의 푸리에 변환 알고리즘보다 훨씬 효율적이므로 계산 복잡성을 줄일 수 있다. FFT는 아래 [그림 3-4]와 같이 시간 영역의 신호인 입력 데이터, 주파수 도메인으로 변환된 데이터인 출력 데이터, FFT의 크기를 나타내는 변환 크기, 주파수 도메인에서 주파수 성분의 간격을 결정하는 샘플 주파수로 구성되어 있다. FFT는 음성 처리, 스펙트럼 분석 등 신호를 분석하고 필요한 정보를 추출하는데 필수적이다.



[그림 3-4] 시간 및 주파수 영역의 신호[20]

오버랩(Overlap, hop_length)는 연속적인 프레임 간의 겹침을 나타내는 값으로 일반적으로 프레임 크기보다 작은 값을 가진다. 다음 [그림 3-5]는 Hop length 예시이다.



[그림 3-5] hop_length 예시[21]

[그림 3-5]과 같이 겹치는 정도는 프레임 간의 정보 공유를 의미하며, 음성

신호의 부드러운 전환을 보장하는 역할을 한다. hop_length는 프레임 간의 간격을 나타내므로 예를 들어 큰 hop_length 값을 사용하면 프레임들 사이에 더 많은 겹침이 발생하고, 작은 hop_length 값을 사용하면 겹침이 줄어든다. 이렇게 겹침을 조절함으로써 프레임 간의 연속성 및 시간적인 정보를 제어할 수 있으며 음성 데이터의 종류나 분석 목적에 따라 다르다. hop_length를 150으로 설정하여 연속적인 프레임 간의 겹침을 많게 하여 오디오 신호의 부드러운 전환을 더 잘 캡처하고 적당한 시간 해상도를 유지하면서 효율적인 연산 자원을 고려하였다.

n_MFCC는 MFCC의 수를 나타내는 파라미터로 MFCC 특징을 추출할 때 몇 개의 계수를 생성할지 결정하는 데 사용된다. n_MFCC의 값이 클수록 더 많은 MFCC 계수가 추출되며, 음성 신호의 더 상세한 주파수 특성을 포착할 수 있다. 이 값은 MFCC 특징 벡터의 차원을 결정하며, 일반적으로 13인 MFCC 계수를 사용한다. n_MFCC가 13으로 설정되는 이유는 음성 신호의 특징성과 차원 감소로 설명할 수 있다. 음성 신호는 자연어를 나타내는 복잡한 신호로 이 신호의 주파수 특성을 추출하는 데 MFCC와 같은 특징 벡터가 사용된다. 차원 감소로는 13개의 MFCC 계수로 구성된 특징 벡터는 음성 신호의 주요 특성을 잘 포착하면서도 차원이 상대적으로 낮기 때문에 모델 학습 및 추론 시에 계산 비용을 낮추고 메모리 사용을 감소시키는 데 도움이 된다. 따라서 MFCC 추출 간에 음성 처리 분야에서 널리 사용되는 기본적인 설정 중 하나인 n_MFCC 값을 13으로 설정하였다. 다음 [표 3-3]은 특징 벡터의 의미 및 특징을 나타낸다.

[표 3-3] 특징 벡터의 의미 및 특징

계수 (Index)	MFCC 벡터	Mel-Spectrogram 벡터	의미 및 특징
1	C1	에너지(낮은 주파수 대역)	기본 음성 특성, 성별 및 나이와 관련
2	C2	에너지(약간 높은 주파수 대역)	음성의 전반적인 특성 보완
3-12	C3-C12	에너지(높은 주파수 대역)	고주파 성분에 민감하게 반응하며 음성의 상세한 주파수 성분 특징화
13	C13	에너지(고주파 성분)	감정이나 말하는 사람의 독특한 음성 특징 반영

[표 3-3]처럼 사용한 MFCC 계수는 13으로 각 프레임에서 'MFCC = [C1, C2 ,C3 ... C13]'처럼 13개의 행을 담고 있다. C1은 낮은 주파수 대역에서의 에너지를 나타내며 발화의 저음 부분을 특징화한다. C2는 조금 더 높은 주파수 대역에서의 에너지를 나타내며 음성의 전반적인 특성을 보완하고 C3부터는 높은 주파수 대역으로 가는 에너지를 나타내며 고주파 성분에 민감하게 반응하며 음성의 상세한 주파수 성분을 특징화한다. 이러한 특징들은 주로 음성 인식 및 화자 인식과 같은 음성 처리 작업에서 사용된다. 예를 들어, 낮은 주파수의 C1은 화자의 성별과 같은 기본 음성 특성을 나타내며, 고주파 성분인 C13은 감정이나 화자의 독특한 음성 특징을 반영할 수 있다. 이러한 음성 신호를 이용한 MFCC 벡터의 특징으로 음성을 처리하였다.

MFCC를 이용한 특징 추출로 다음과 같은 방법들이 사용될 수 있다. 첫째는 주파수 성분으로 낮은 주파수인 C1과 C2는 음성의 기본 주파수 특성을 의미하며 화자의 성별과 나이와 관련이 있고 감정 표출에도 영향을 미칠 수 있다. 두 번째는 에너지와 강도로 낮은 주파수 대역의 에너지와 강도는 감정적인 음성의 특성을 나타낼 수 있다. 반면, 높은 강도는 강한 감정을 나타낼 수 있으며, 에너지의 분포는 감정의 성격을 나타낼 수 있다. 세 번째는 감정은 발음, 강도, 음조 등에서 나타나는 동적인 특성에도 영향을 받으므로 이를 활용한 MFCC 벡터를 시간에 따라 변하는 동적인 특성으로 확장하여 감정의 변화를 더 잘 포착할 수 있다. 네 번째는 MFCC 계수의 변화율로 감정 변화는 주로 음성 특성의 동적인 변화와 관련이 있으며 MFCC 벡터 각 요소의 변화율을 계산하여 이러한 동적인 특성을 추출할 수 있다. 마지막은 고차원의 MFCC 계수로 고차원의 MFCC 계수, 특히 C13 이상은 고주파 성분에 민감하게 반응하며 이 고주파 성분은 감정의 세세한 특성을 나타낼 수 있다. 이러한 특징들은 주로 머신러닝 모델에 입력으로 사용되어 감정을 인식할 수 있다. 예를 들어, 높은 주파수 성분이나 에너지의 분포 등 특정 감정과 관련이 있을 수 있으며 특징들을 통해 감정 상태를 구분하고 인식하였다.

b. Mel-Spectrogram

Mel-Spectrogram(멜 스펙트로그램)은 음성 처리와 음향 신호 분석에서 사용되는 다른 주파수 모델인 특징 벡터로 MFCC와 마찬가지로 음성 신호의

주파수 특성을 나타내지만 MFCC와 달리 로그 메일 스케일로 변환하지 않는다. 위의 [그림 3-2]와 같이 Mel-Spectrogram은 MFCC와 마찬가지로 데이터를 로드하여 웨이브폼과 샘플링 레이트를 추출한 후 프레임을 분할한다. 다음, Librosa 라이브러리의 Mel-Spectrogram 함수를 사용하여 추출한다. 이 함수는 MFCC와 비슷하게 n_fft, hop_length, n_Mels 인자를 필요로 한다. MFCC와 다른 인자인 n_Mels는 n_MFCC와 같은 역할을 한다. n_Mels는 멜 필터의 수를 나타내는 파라미터로 주파수 영역의 신호를 작은 밴드로 분리하는 데 사용된다. 멜 필터는 주파수 영역을 멜 스케일로 변환하여 인간의 청각 특성을 모델링하고 인간의 청각 시스템의 반응에 기반한 스케일로 낮은 주파수에서는 더 밀접하게 나누며 높은 주파수에서는 덜 나누어진 밴드를 생성한다. n_Mels 값은 일반적으로 13을 사용하는데 이는 일반적으로 음성 처리 작업에서 출력 차원과 주파수 영역의 해상도를 조절하는데 잘 작동되는 값으로 알려져 있다.

다음은 Mel-Spectrogram을 데시벨 스케일로 변환하는 과정으로 적절한 척도로 변환하여 주파수 영역의 특성을 강조하고 인간의 청각 특성과 일치시키는 역할을 한다. 이전 단계에서 추출한 Mel-Spectrogram을 Librosa 라이브러리의 Power_to_db 함수를 통해 [그림 3-6]과 같이 일상 생활에서 자주 사용하는 데시벨 스케일(db Scale)로 변환한다.

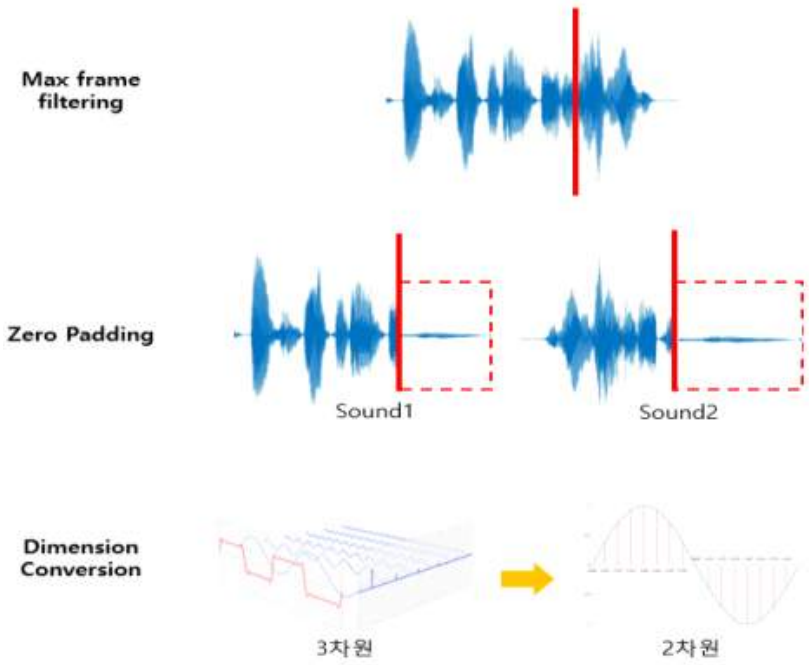
[그림 3-6] 데시벨 스케일

[그림 3-6]과 같이 데시벨 스케일은 주파수 도메인에서 선형적인 스케일에서 로그 스케일로 변환된 스케일로 음향 신호 처리에서 사용되는 주파수 영

역의 에너지나 파워를 나타내는 데 사용된다. 로그 스케일을 사용하면 작은 값과 큰 값 사이의 차이를 뚜렷하게 보여주며, 큰 범위의 값에 대한 상대적인 비교가 용이하다. 또한, 인간의 청각 시스템은 주파수 영역에서 에너지를 로그 스케일로 인식한다. 즉, 높은 주파수에서의 에너지 변화에 민감하며, 낮은 주파수에서는 에너지 변화를 미미하게 인식하기 때문에 데시벨 스케일을 사용하면 주파수 영역의 에너지 분포를 인간의 청각 특성과 일치시켜 더 효과적으로 분석할 수 있다.

다음은 음성 데이터에서 Mel-Spectrogram을 사용하여 추출한 특징 벡터에 대한 설명이다. 사용한 계수 값은 13으로 예시로 각 프레임에서 [200, 50, 300, 80, 120, 25, 180, 10, 15, 5, 8, 20, 150]과 같이 벡터를 담고 있다고 가정하면, 이 벡터는 해당 프레임에서의 주파수 성분의 에너지를 나타내고 있다. 벡터 예시를 보면 300 (세 번째 값)은 해당 프레임에서의 주파수 성분 중에서 Mel 필터 3번에서 높은 에너지로 이는 음성에서 3번 Mel 필터에 해당하는 주파수 대역에서 강한 활동이 있음을 나타낸다. 따라서 해당 프레임에서는 주로 해당 주파수 성분이 강조되는 어떤 소리 또는 음성의 특정 부분이 있음을 의미한다. 50 (두 번째 값)과 25 (여섯 번째 값)은 Mel 필터 2번에서 낮은 에너지로 해당 주파수 대역은 상대적으로 약한 신호임을 알 수 있다. 상대적으로 약한 신호가 있는 것은 화음이나 배경 소음과 같이 중요하지 않은 소리임을 생각해볼 수 있다. 일곱 번째 값인 180은 Mel 필터 7에서의 주파수 성분에서 중간 정도의 에너지를 나타낸다. 이는 해당 주파수 대역에서는 중요한 신호 또는 음성의 일부가 있는 것으로 판단할 수 있으며 이 주파수 성분은 발음의 다양한 특징을 포함할 수 있다. 또한, 마지막 값인 150은 마지막 Mel 필터에서의 주파수 성분에서 상대적으로 높은 에너지를 나타낸다. 예를 들어 해당 주파수 대역에서는 말소리의 일부가 강조되는 강한 신호나 활동이 있을 수 있다. 이처럼 Mel-Spectrogram 벡터를 자세하게 분석하면 음성에서의 주파수 대역별 에너지 분포를 이해할 수 있다. 높은 값은 해당 주파수 대역에서의 강한 신호 또는 발음의 중요한 성분을 나타내고, 낮은 값은 상대적으로 약한 신호를 나타낸다. 이 정보를 활용하여 음성 데이터에서의 특정 주파수 성분의 통해 데이터를 처리하였다.

D. 음성 데이터 감정 인식 방법



[그림 3-7] 음성 데이터 감정 인식 방법 구성도

데이터의 일관성을 유지하고 모델의 성능을 최적화하기 위한 특징 벡터 전처리는 필요하다. 추출된 특징 벡터는 음성과 텍스트 데이터를 이용하여 감정 인식을 수행한 [8]의 연구와 같이 위의 [그림 3-7]의 순서에 따라 음성 데이터 감정 인식 과정을 거친다. [그림 3-7]의 구성도를 보면 첫째, Max_frame filtering 작업에서는 특징 벡터가 최대 프레임 수를 넘지 않도록 데이터를 처리한다. 이는 데이터의 크기를 관리하고 일관성 있게 유지하기 위한 중요한 단계로, 데이터의 길이를 제한하는 역할을 한다. 둘째, 데이터 길이를 동일하게 유지하기 위해 Zero-Padding을 수행한다. 모델에 입력되는 데이터의 길이를 통일함으로써, 모델이 일관된 형태의 입력을 받을 수 있게 되며 예측 결과를 더 정확하게 만들어준다. 셋째, 2차원의 특징 벡터 데이터를 하나로 담은 3차원의 리스트를 다시 2차원으로 변환하는 작업을 수행한

다. 이는 데이터의 형태를 모델이 처리하기 쉬운 형태로 변환하는 단계로 2차원 데이터 형태로 모델에 입력된다. 넷째, 데이터 전처리 과정이 끝나면 추출된 특징 벡터는 모델 학습을 위해 학습 데이터와 테스트 데이터 셋으로 나누어지며, SVM 모델을 사용하여 감정 분류 작업을 수행한다. 먼저, 전처리된 특징 벡터는 학습 데이터와 테스트 데이터로 분할된다. 이 데이터의 분할을 통해 학습 데이터와 모델 평가를 위한 테스트 데이터를 나누게 된다. 분할된 데이터는 SVM 모델을 통해 특징 벡터를 입력받아 감정 분류를 수행하며 음성 데이터와 감정 레이블 간의 관계를 학습하고 이를 기반으로 테스트 데이터에서 감정을 예측한다. 모델 훈련 후 SVM 모델의 성능 향상을 위해 하이퍼파라미터 튜닝이 수행된다. 이 작업은 최적의 모델 설정을 찾아내기 위해 다양한 하이퍼파라미터 조합을 평가하고 모델의 정확도를 높이는 데 도움을 준다. 이와 같은 순서로 구성된 음성 데이터 감정 인식 방법을 통해 특징 벡터를 활용하여 감정 분류 모델을 학습하고 성능을 평가하였다.

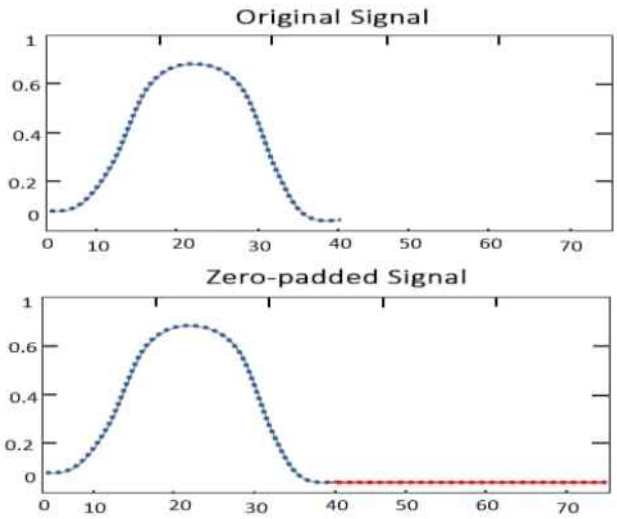
1. 모델 학습을 위한 데이터 처리

a. Max frame filtering

Max frame filtering은 음성 데이터에서 추출된 특징 벡터의 프레임 수를 제한하는 작업이다. 일반적으로 음성 데이터는 길이가 다양하며, 특징 벡터 기법으로 추출된 벡터의 프레임 수는 음성 데이터의 길이와 파라미터 값에 비례하기 때문에 각 데이터마다 다른 프레임 수를 갖게 되므로 길이를 제한할 필요가 있다. 최대 프레임을 제한하게 되면 다음과 같은 효과를 얻을 수 있다. 첫째, 계산 효율성 측면에서 유리하다. 각 프레임의 크기가 동일하면 모델 학습 및 추론 과정에서 연산 량을 줄일 수 있으며, 이는 모델의 속도를 향상시키고 메모리 사용을 최적화하는 데 도움이 된다. 둘째, 대부분의 모델은 고정된 입력 크기를 입력해야 하므로 모델 호환성과도 연관이 있다. 만약 입력 데이터의 크기가 가변적이면 모델을 구축하고 학습시키기가 복잡해진다. 따라서 MFCC 특징 벡터 프레임 수를 제한함으로써 모델에 대한 입력 크기를 표준화할 수 있다. 셋째, 더 짧은 음성 데이터를 길게 변환하는 것은 정보 손실을 초래할 가능성이 있어 정확도 향상을 위해서 사용할 수 있다.

단, 감정 분류와 같은 작업에서 각 음성 데이터는 특정 감정을 나타내는 정보를 포함하고 있을 수 있으므로 너무 큰 프레임으로 제한하지 않도록 주의해야 한다. 감정 데이터의 대부분이 2-3초 정도의 길이이며 hop_length 등 파라미터 값을 고려하여 특징 벡터의 프레임 수를 1000 프레임으로 설정하였다. 이는 음성 데이터를 잘 처리하고 모델을 효과적으로 학습시키기 위함이다. 예를 들어, 프레임 수가 크면 데이터의 빈 공간에 들어가는 Zero-Padding의 값이 많아져 데이터가 훼손되거나 작으면 데이터의 정보 손실이 많이 일어날 수 있으므로 데이터의 훼손이 적고 정확도가 다른 수치에 비해 높은 1000 프레임으로 설정하였다.

b. Zero-Padding



[그림 3-8] Zero-Padding 예시[22]

위의 [그림 3-8]과 같이 Padding은 데이터의 길이를 동일하게 유지하거나 원하는 길이로 조정하는 작업으로 목적은 모든 데이터가 동일한 크기를 가지도록 하여 모델이 데이터를 처리하기 쉽도록 만드는 것이다. [14]의 연구에서는 Zero-Padding과 데이터의 패턴을 고려하여 선형 또는 다항식을 사용하여 보간(Interpolation)을 적용했다. Zero-Padding은 간단하고 쉬운 방법이지만, 0으로 Padding된 부분은 모델에게 정보를 제공하지 않아 일부 정보의 손실이 발생할 수 있다는 특징이 있어 연구 목적에 따라 보간은 Padding보다 더 유

용한 정보를 제공할 수 있다. 따라서 어떤 Padding 방식을 선택할지는 데이터 및 모델에 따라 다를 수 있으며, 데이터의 특성을 고려하여 Padding 방식을 선택해야 한다.

본 실험에서는 다음과 같은 이유로 Zero-Padding을 사용하였다. 첫째, 대부분의 딥러닝 모델은 고정된 입력 크기를 기대하므로 모델 호환성에 유리하다. 입력 데이터의 크기가 가변적이면 모델을 구축하고 학습시키기가 더 복잡하다. 둘째, 각 음성 샘플의 길이가 동일하면 모델 학습 및 추론 시에 연산량을 줄일 수 있어서 효율적이다. 셋째, 원본 음성 데이터의 정보를 손상시키지 않고 유지할 수 있다. 0 값으로 채워진 부분은 실제 정보를 의미하지 않아 기존 음성 데이터의 정보를 보존하면서 동일한 길이의 입력을 생성할 수 있다. 또한, 보간을 선택하지 않은 이유는 각 음성 데이터의 길이를 맞추는 과정에서 보간을 사용하면 음성 데이터의 뒷부분에 데이터가 채워져서 음성의 실제 종료 지점을 왜곡할 수 있기 때문이다. 이로 인해 감정 분류 모델에게 잘못된 정보가 입력될 수 있으며 모델의 성능을 저하시킬 수 있다. 따라서 음성 데이터를 동일한 길이로 조정함으로써 모델이 일관된 데이터를 받아들이고 각 음성 데이터의 실제 종료 지점을 왜곡하지 않도록 고려하였다.

c. Dimension Conversion

추출된 특징 벡터는 일반적으로 2차원 배열로 추출되며, 각 음성 데이터 샘플은 특징 벡터 계수(특징)와 프레임 길이로 구성된다. 다수의 음성 데이터를 다룰 때, 이러한 2차원 특징 벡터 배열들을 리스트나 배열로 담게 되면 다음 [표 3-4]과 같은 3차원 형태로 표현할 수 있다.

[표 3-4] 데이터 셋 차원 예시

	축0(1차원)	**축1(2차원)**	**축2(3차원)**
변수 및 형태	음성 샘플 (Sample 1, Sample 2, ..., Sample n)	특징 벡터 계수	프레임 길이

[표 3-4]과 같이 (580, 13, 1000)의 Shape을 가진 데이터 셋이 있다고 가정하면 첫 번째 차원 580(축 0)은 각각 580개인 음성 데이터의 인덱스를 나타내며, 각 음성 데이터마다 다른 특징 벡터 시퀀스가 존재하며, 감쩡 레이블과 함께 표현된다. 두 번째 차원(축 1) 13은 각 특징 벡터 시퀀스 내에서 각 특징 벡터 계수를 표현한다. 특징 벡터 계수는 MFCC 추출 전에 사람의 청각과 가장 유사한 정도인 13으로 설정한 변수이다. 세 번째 차원(축 2) 1000은 프레임 길이를 나타낸다. 최대 프레임으로 설정한 1000 프레임으로 1000 프레임을 넘는 데이터 셋은 제외되거나 넘지 않는 데이터 셋은 Zero-Padding 된 데이터만 해당된다. 이 3차원 데이터는 각 음성 데이터의 MFCC 계수와 프레임 길이에 대한 정보를 포함하며, 다수의 음성 데이터 샘플을 나타낸다.

하지만 3차원 특징 벡터에서 2차원으로 변환하는 작업은 대부분의 머신러닝 모델에서 필요한 과정이다. 예를 들어, 주로 2차원으로 변환하는 방법으로 각 음성 샘플에 대해 특징 벡터 시퀀스를 하나의 벡터로 펼치는 방법을 사용할 수 있다. 이렇게 하면 각 음성 샘플에 대해 하나의 2차원 데이터 포인트가 생성된다. 3차원 데이터를 2차원으로 변환하는 과정으로 MFCC 계수와 프레임의 길이를 곱하여 데이터의 형태를 유지하면서 2차원 배열로 재구성한다. 예를 들어 (580, 13, 1000)인 3차원의 형태를 (580, 13000)으로 변환할 수 있다. 580은 음성 데이터의 인덱스를 나타내며, 13000은 MFCC 계수 13과 프레임 길이 1000을 곱하여 생성한 값으로 데이터의 개수와 2차원 배열로 변환된 MFCC 데이터의 열 수를 나타낸다. 따라서 3차원의 값을 2차원으로 변환해도 동일한 결과를 얻을 수 있다.

2. SVM 기반 감정 인식 방법

다음은 SVM 모델의 하이퍼파라미터 튜닝 설정과 모델 학습을 수행하였다. 먼저, train_test_split 함수를 사용하여 데이터 셋을 무작위로 학습 데이터(X_train, y_train)와 테스트 데이터 (X_test, y_test)로 분할한다. 테스트 데이터의 비율을 나타내는 test_size는 전체 데이터 셋의 20%를 테스트 데이터로 할당하였다. 다음, SVM 모델의 하이퍼파라미터를 최적화하기 위해 Grid SearchCV(그리드 서치) 방법을 활용하였다. Grid SearchCV는 여러 가지 하

이퍼파라미터 조합을 시도하고, 가장 우수한 조합을 선택하는 데 주로 사용된다. 특히, SVM 모델의 C 값, 커널 함수(kernel), 그리고 gamma 값이 튜닝에 주로 사용되며 다음 [표 3-5]는 실험에서 사용한 파라미터와 그 후보 값을 나타낸다.

[표 3-5] SVM 파라미터

파라미터	후보값
'C'	[0.1, 1, 10]
'Kernel'	['Linear', 'RBF', 'Poly']
'Gamma'	['Scale', 'Auto', 0.01, 0.1, 1]

[표 3-5]와 같이 C 파라미터 후보 값은 [0.1, 1, 10]으로 얼마나 많은 오분류를 허용할지를 조절하는 정규화 파라미터이다. 작은 값일수록 많은 오분류를 허용하고, 큰 값일수록 오분류를 허용하지 않는다. Kernel 함수는 데이터를 고차원 공간으로 매핑하는 데 사용되며 Kernel 값으로 선형 커널('Linear'), Radial Basis Function 커널('RBF'), 다항식 커널('Poly')을 사용하였다. 이 파라미터는 어떤 종류의 결정 경계를 생성할지에 영향을 미친다. Gamma는 Kernel 함수의 스케일(Scale) 파라미터로, 커널 함수의 모양을 제어하고. 'Scale'과 'Auto'는 자동으로 스케일링을 조정하는 옵션이다. 또한 [0.01, 0.1, 1] 세 가지 다른 값으로 파라미터를 선정하였다.

Ⅳ. 실험 및 결과

A. 데이터 셋

1. 원본 데이터 셋

본 연구에서는 Kaggle에서 제공하는 CREMA[23] 음성 데이터 셋을 실험 데이터로 선정하였다. CREMA 데이터 셋은 전문 배우들은 대상으로 영화 대사를 연기하도록 설계되었으며, '화남(Anger), 중립(Neutral), 슬픔(Sad), 행복(Happiness), 두려움(Fear), 실망(Disappointed)'과 같은 다양한 감정을 표현하며 이러한 감정을 여러 감정 톤과 발음으로 전달한다. 각 음성 데이터는 음성 톤, 발음, 음성 세기 등 음성을 담고 있어 감정 인식 및 음성 처리 연구에 유용하다. CREMA 데이터 셋은 비디오 데이터와 표정 데이터도 포함하고 있어 멀티모달 연구에도 활용 가능하며 이를 통해 추후 음성, 비디오 및 표정 간의 상호 연관성을 연구하고 다양한 감정을 다루는 실험도 가능하다. 데이터 셋에는 다양한 배우들이 녹음에 참여하였으며 이로 인해 여러 연령, 성별, 백인, 아프리카계 미국인, 아시아계 미국인과 같은 배우들의 감정 표현을 포함하고 있다. 데이터는 감정 클래스에 해당하는 음성 데이터를 각각 약 1200개씩 포함하고 있으며 각 데이터 당 대략 2-3초 정도의 길이를 가지고 있다. 감정 클래스는 6개의 감정 중 '실망' 감정 클래스를 제외한 5가지 감정을 선택했다. '실망' 감정 클래스는 음성 데이터에서 'Sad'와 음성 톤을 듣고 비교하였을 때 의미를 듣지 않고는 사람 간에도 잘 분류되지 않는 어려운 감정 중 하나로 알려져 있다. 이는 감정의 주관적이고 다양한 특성 때문에 발생하는 어려움으로 이해된다. 따라서 해당 클래스를 실험에서 제외함으로써 모델 학습의 안정성을 높이고 데이터 불균형으로 인한 문제를 방지하였다.

데이터 셋은 파일명 구조를 통해 각 음성 데이터에 대한 고유한 식별 정보를 제공하고 있다. 파일명은 아래 [표 4-1]로 구성하였다.

[표 4-1] CREMA 데이터 셋의 음성 파일명 구조

항목	설명
Actor id (배우 식별번호)	- 파일명의 시작 부분에 위치 - 각 배우를 식별하는 데 사용되며, 4 자리 숫자로 표현
Sentences (문장)	- 음성 샘플이 어떤 문장을 포함하는지를 나타냄
Emotion (감정)	- 괄호 안에 감정 유형을 나타내는 세 글자 코드가 포함
Emotion level (감정 수준)	- 괄호 안에 두 글자 코드가 포함 - 해당 음성 샘플의 감정 수준 - 낮음 (Low), 중간 (Medium), 높음 (High)

[표 4-1]과 같이 음성 파일명 구조에 따라 다음과 같이 데이터의 예시를 들 수 있다. 각 배우를 식별하는 Actor_id, 샘플 문장인 Sentences, 감정을 나타내는 Emotion, 감정의 수준을 나타내는 Emotion level로 구성되어 있다. 다음 [표 4-2]는 [표 4-1]을 항목을 기반으로 만든 음성 파일명 구조 예시이다.

[표 4-2] 음성 파일명 구조 예시

Actor_id	Identifier	Sentences	Emotion	Emotion level
1005	IWW	I wonder what this is about (IWW)	Disgust (DIS)	Unspecified (XX)
1007	IWL	I would like a new alarm clock (IWL)	Happy (HAP)	Unspecified (XX)

[표 4-2]을 참고하여 음성 데이터를 다음과 같이 설명할 수 있다. Actor_id가 1005번이며, 문장의 종류를 나타내는 Identify는 IWW, Identify를 풀어서 문장으로 나타내는 Sentences는 'I wonder what this is about', 감정은 Disgust, 감정 세기는 Unspectified로 이해할 수 있다.

2. 증강 데이터 셋

본 연구에서는 음성 감정 인식에 적합한 데이터 처리를 위해 원본 데이터를 기반으로 다양한 증강 데이터를 생성하였다. 증강 기법으로는 속도 조정과 노이즈 추가를 적용하였으며, 크게 네 종류의 데이터 셋을 형성하였다. 첫 번째는 Numpy 기반 속도 증강 데이터, 두 번째는 Audiosegment 기반 노이즈 증강 데이터, 세 번째는 원본 데이터와 증강 데이터를 병합한 데이터 셋, 네 번째는 증강 데이터 간 병합한 데이터 셋으로 구성되어 있다. 속도 범위는 원본 데이터의 배율을 기준으로 하며, 1.0 배율을 제외한 범위에서 0.9 배율부터 1.1 배율까지의 범위를 포함한다. 이 범위를 0.02 간격으로 나누어 0.9에서 0.98까지와 1.02에서 1.1까지의 각 데이터 셋을 형성하였다. 예를 들어, 0.9-0.98 데이터 셋에는 0.9, 0.92, 0.94, 0.96, 0.98과 같이 다양한 배율을 가진 데이터가 있으며 총 10개의 서로 다른 배율을 포함하고 있다. 노이즈 추가 증강 역시 2-10세기까지 2세기 간격으로 설정하였다. 노이즈 데이터 셋은 세기 간격으로 2, 4, 6, 8, 10과 같이 다양한 노이즈 세기를 가진 데이터를 생성하였다. 다음은 설정한 네 가지 주요 데이터 증강 기법을 상세하게 설명하고자 한다.

a. 속도 증강 실험

음성 데이터의 속도 배율을 0.9-0.98, 1.02-1.1로 분류하였으며 0.9-0.98 데이터 셋, 1.02-1.1 데이터 셋, 0.9-1.1 데이터 셋으로 구성하였다. 데이터 셋은 원본 데이터를 제외하고 67,420개이며 File Name과 Emotion 칼럼을 기준으로 데이터를 라벨링하였다. 다음 [표 4-3]과 같이 MFCC와 Mel-Spectrogram으로 나누어 속도 증강 데이터 별로 실험을 진행하였다.

[표 4-3] MFCC와 Mel-Spectrogram 속도 증강 실험

Extraction method	Dataset	Speed rate
MFCC	Speed(67420)	0.9-0.98, 1.02-1.1, 0.9-1.1
Mel-Spectrogram		

[표 4-3]과 같이 속도 증강 실험은 추출 방법으로 MFCC와 Mel-Spectrogram을 사용하였으며 사용된 데이터 셋은 67,420개이다. 그리고 적용된 속도 배율은 0.9-0.98, 1.02-1.1, 0.9-1.1로 크게 3종류로 구분하여 실험하였다.

b. 노이즈 증강 실험

음성 데이터의 노이즈 세기 범위를 2-10세기로 하였으며, 데이터를 2세기 간격으로 구성하였다. 데이터 셋은 원본 데이터를 제외하고 33,710개이며 File Name과 Emotion 칼럼을 기준으로 데이터를 라벨링하였다. 다음 [표 4-4]와 같이 MFCC와 Mel-Spectrogram으로 나누어 속도 증강 데이터 별로 실험을 진행하였다.

[표 4-4] MFCC와 Mel-Spectrogram 노이즈 증강 실험

Extraction method	Dataset	Noise rate
MFCC	Noise(33710)	2-10
Mel-Spectrogram		

[표 4-4]처럼 속도 증강 실험은 추출 방법으로 MFCC와 Mel-Spectrogram을 사용하였으며 사용된 데이터 셋은 33,710개이다. 그리고 적용된 노이즈 세기는 2-10이며, 2세기 간격으로 구분하여 실험하였다.

c. 원본·증강 데이터 병합

다음은 원본·증강 데이터를 병합한 데이터 셋으로 속도 증강 데이터는 0.9-1.1 배율 데이터 전체와 원본 데이터를 병합하여 총 데이터는 74,162개로, 노이즈 증강 데이터는 2-10 배율 데이터와 원본 데이터를 병합하여 40452개로 구성되었다. 다음 [표 4-5]는 각 추출 방법에 대한 병합 실험을 나타낸다.

[표 4-5] MFCC, Mel-Spectrogram 기반 원본·증강 데이터 셋 병합 실험

Extraction method	MFCC, Mel-Spectrogram	Extraction method	MFCC, Mel-Spectrogram
Dataset 1	Speed + Raw (74162)	Dataset 2	Noise + Raw (40452)
Speed rate	0.9-1.1	Noise rate	2-10

위의 [표 4-5]는 각각 MFCC와 Mel-Spectrogram을 기반으로 원본·증강 데이터 셋의 병합 실험이다. Dataset은 원본·속도 데이터를 병합한 데이터 셋 1인 74,162개와 원본·노이즈 데이터를 병합한 데이터 셋 2인 40,452로 구성된다. 속도와 노이즈 세기는 각각 0.9-11, 2-10에 해당한다.

d. 증강 데이터 병합

다음은 증강 데이터를 병합한 데이터 셋으로 속도 증강 수치와 노이즈 증강 수치를 병합하여 총 데이터는 6,742개로 구성하였으며 다음 [표 4-6]은 각 실험에 대한 내용이다.

[표 4-6] MFCC, Mel-Spectrogram 기반 증강 데이터 병합 실험

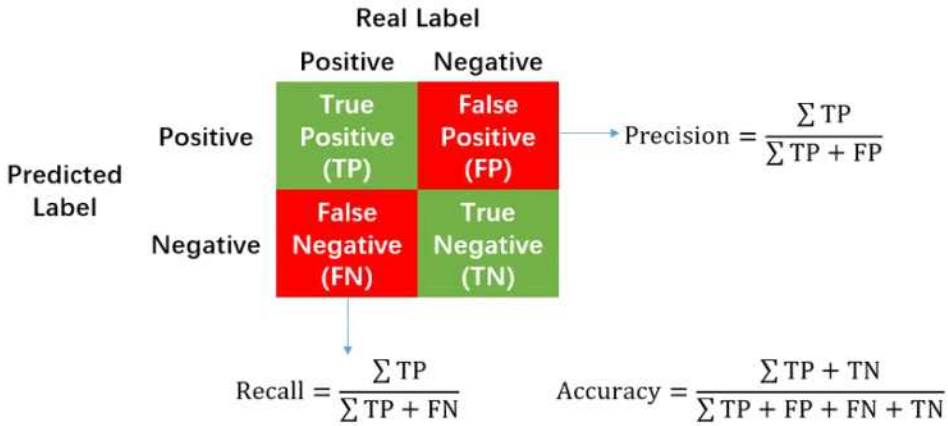
Extraction method	Dataset	Speed rate	Noise rate
MFCC	Speed + Noise (6,742)	0.9	10
Mel-Spectrogram			

[표 4-6]은 MFCC와 Mel-Spectrogram을 기반으로 증강 데이터를 병합한 실험이다. Dataset은 속도와 노이즈를 모두 적용한 데이터인 6,742개로 구성되어있으며 속도 배율은 0.9, 노이즈 세기는 10으로 설정하였다. 기존의 원본·증강 데이터 셋처럼 증강 기법 하나를 적용한 데이터가 아닌 하나의 음성 샘플에 속도와 노이즈를 동시에 적용하여 두 증강 데이터의 특징을 모두 포함하는 데이터 셋을 추가로 생성하여 실험하였다.

B. 실험 및 분석

1. 실험 평가 방법

실험 평가 방법으로는 SVM(Support Vector Machine) 모델을 사용하여 도출된 테스트 데이터에 대한 예측을 수행한다. 실제 감정 레이블과 모델의 예측 결과를 비교하여 정확도를 계산하였으며 정확도는 예측된 레이블 중 실제 레이블과 일치하는 비율을 나타낸다. 또한, Scikit-learn 라이브러리에서 제공하는 metrics의 Classification report 함수를 사용하여 각 감정마다 정밀도(Precision), 재현율(Recall), F1 점수(F1-Score), 지원(Support) 값을 계산하고 Confusion matrix를 생성하였다. 다음 [그림 4-1]은 사용한 Confusion matrix에 관한 그림과 수식이다.



[그림 4-1] Confusion matrix

[그림 4-1]과 같이 Confusion Matrix은 머신러닝 및 통계 분야에서 모델의 성능을 평가하고 분류 작업의 결과를 시각화하는 데 사용되는 중요한 도구로서 주로 이진 분류(예: 양성/음성, 참/거짓)을 다룰 때 사용되지만, 다중 클래스 분류에도 확장 가능하다. 정밀도는 양성(Positive) 클래스로 예측한 데이터 중 실제로 양성인 샘플의 비율을 나타낸다. 정밀도는 거짓 양성(FP)을 최소화하고 양성 데이터를 정확하게 예측하는 데 중점을 둔다. 정밀도는 높을

수록 모델이 양성으로 예측한 데이터 중 실제로 양성인 비율이 높아짐을 의미한다. 예를 들어, 'Anger'의 정밀도가 0.74였을 때 100개 중 74개는 실제로 양성으로 예측했다는 것이다. 재현율은 실제 양성 클래스에 속한 데이터 중 모델이 양성으로 예측한 샘플의 비율을 나타낸다. 거짓 음성(FN)을 최소화하고 모든 실제 양성 데이터를 식별하는 데 중점을 둔다. 재현율은 높을수록 모델이 실제 양성 데이터를 놓치지 않고 잘 식별함을 의미한다. 예를 들어, 'Anger' 감정 클래스에 대한 재현율은 실제 'Anger' 감정을 가진 샘플 중 약 75%를 모델이 'Anger'로 정확하게 예측했음을 나타낸다. F1 점수는 정밀도와 재현율의 조화 평균으로, 정밀도와 재현율을 모두 고려하기 때문에 불균형한 데이터 셋에서 모델의 성능을 정확하게 평가하는 데 도움이 된다. 예를 들어, 'Anger' 감정 클래스에 대한 F1 점수는 정밀도와 재현율의 조화 평균으로, 모델의 성능을 하나의 숫자로 나타낸다. F1 점수가 약 0.74이므로 모델은 'Anger' 감정 클래스에 대해 균형 있는 성능이라고 볼 수 있다. 마지막으로 지원은 각 클래스에 대한 샘플 수를 나타내며, 이 값은 각 클래스가 얼마나 많은 샘플을 가지고 있는지를 보여준다. 이와 같은 특성으로 Classification report는 주로 다중 클래스 분류 모델에서 사용되며, 모델의 각 클래스에 대한 정보를 제공하여 모델이 어떤 클래스를 얼마나 잘 분류하는지를 평가하는 데 도움을 준다. 'Anger' 감정 클래스에 대한 지원 값은 데이터에서 'Anger' 감정을 가진 데이터의 수이며, 실험에서 사용한 데이터 셋에서 'Anger' 감정을 가진 데이터가 1377개임을 의미한다.

위의 Confusion matrix 결과에 따라 각 실험에 대한 감정 인식 성능 향상의 기준은 원본 데이터의 감정 인식 결과 및 정확도로 설정하였다. 원본 데이터에 MFCC 벡터를 사용하여 추출한 정확도는 0.443, Mel-Spectrogram 벡터를 사용하여 추출한 정확도는 0.452로 원본 데이터 셋에서는 Mel-Spectrogram이 소폭 높았다. 이처럼 도출된 Confusion matrix를 기준으로 여러 증강 데이터와 벡터 추출 방법을 활용한 다양한 종류의 실험을 통해 음성 감정 인식에 적합한 방법이 무엇인지, 어떤 방식이 감정 인식 정확도가 더 높은지 분석하였다.

2. 실험 결과 분석

a. 속도 증강 실험

음성 속도 증강 실험에서는 음성 데이터의 속도를 증강하여 MFCC와 Mel-Spectrogram 두 가지 음성 특징 추출 방법의 결과를 비교하였다. 실험은 속도 증강 범위가 0.9-1.1 배율까지의 데이터 셋으로 수행하였으며 어떤 음성 특징 추출 방법이 가장 효과적인지 평가하고자 하였다. 다음 [표 4-7]은 MFCC와 Mel-Spectrogram으로 나누어 속도 증강 데이터 별로 진행한 실험이다.

[표 4-7] MFCC와 Mel-Spectrogram 속도 증강 실험 결과

Extraction method	Dataset	Speed rate	Validation (Accuracy)
MFCC	Speed (67420)	0.9-0.98	0.54(0.458)
		1.02-1.1	0.51(0.43)
		0.9-1.1(1.0 제외)	**0.561(0.467)**
Mel-Spectrogram		0.9-0.98	0.568(0.465)
		1.02-1.1	0.573(0.446)
		0.9-1.1(1.0 제외)	**0.582(0.475)**

[표 4-7]과 같이 Mel-Spectrogram이 MFCC 보다 전반적으로 더 우수한 성능을 보였으며 이는 음성 데이터의 스펙트럼 정보를 놓치지 않는다는 것이다. 또한, 주파수-시간 도메인의 정보를 포함하기 때문에 음성 데이터의 속도 변화에도 대응할 수 있었다. 특히, 음성 속도를 0.9-1.1 배율까지 다양하게 증강한 상황에서 안정적인 성능을 보였다. 이 실험을 통해 Mel-Spectrogram이 음성 데이터 속도 증강 상황에서 더 효과적인 특징 추출 이고 음성 데이터의 속도 변화를 고려하여 모델을 훈련시키는 것이 실제 음성 처리 응용에서 중요하다는 것을 알 수 있었다. 다음 [표 4-8]은 해당 실험에서 가장 정확도가 높은 Mel-Spectrogram의 0.9-1.1배의 Confusion Matrix이다.

[표 4-8] MFCC와 Mel-Spectrogram 속도 증강 Best Confusion Matrix

	Precision	Recall	f1-score	Support
Anger	0.74	0.75	0.74	1377
Fear	0.48	0.34	0.40	1383
Happiness	0.56	0.53	0.55	1403
Neutral	0.53	0.56	0.55	1209
Sad	0.58	0.73	0.65	1338
Accuracy	0.58	-	-	6710
최적 모델 정확도 : 0.58 하이퍼 파라미터 : 'C': 1, 'Gamma': 'Scale', 'Kernel' : 'RBF'				

[표 4-8]의 결과와 같이 'Anger' 클래스가 각 메트릭 중에서도 가장 높고 안정적인 수치로 나타났으며 다른 클래스와 Support 비교해도 클래스 불균형이 있지는 않는 것으로 보인다. 다만, 'Fear' 클래스의 정밀도가 0.48로 다른 클래스에 비해 낮고, 재현율 역시 0.34로 굉장히 낮다. 이는 'Fear' 클래스의 예측이 상대적으로 부정확하며 실제 클래스에 속하는 데이터도 놓치고 있다는 것을 의미한다. 이는 'Fear' 클래스와 비슷한 특성을 가질 수 있는 'Sad' 클래스와 'Neutral' 클래스가 있으며 다른 클래스에 비해 'Fear' 클래스만의 감정 특징이 없는 것으로 볼 수 있다. 따라서 'Fear' 클래스의 데이터 수를 늘려 모델을 학습시키거나 더 나은 특징 추출이 가능한 모델 및 데이터를 확보해야한다. 그리고 하이퍼파라미터의 경우 모든 실험에서 C, Gamma, Kernel가 동일한 수치가 나왔다. 동일한 하이퍼파라미터가 모든 실험에서 좋은 결과로 도출된다면, 이는 해당 하이퍼파라미터가 해당 데이터와 모델 아키텍처에 대해 안정적으로 잘 작동한다고 볼 수 있다. 또한, 모든 실험에서 일관되게 유지된 것으로 보아 실험의 안정성을 나타내며 결과의 신뢰성을 향상시킬 수 있다.

b. 노이즈 증강 실험

노이즈 증강 실험에서는 음성 데이터의 노이즈를 증강하여 MFCC와

Mel-Spectrogram 두 가지 음성 특징 추출 방법을 비교하였으며 노이즈 범위를 2-10세기까지 적용하여 실험을 진행하였다. MFCC와 Mel-Spectrogram으로 나누어 노이즈 증강 데이터 별로 실험을 진행하였으며 결과는 다음 [표 4-9]와 같다.

[표 4-9] MFCC와 Mel-Spectrogram 노이즈 증강 실험 결과

Extraction method	Dataset	Noise rate	Validation (Accuracy)
MFCC	Noise (33710)	2-10	0.576 (0.429)
Mel-Spectrogram			**0.611 (0.439)**

[표 4-9]와 같이 MFCC는 정확도는 0.576(0.429), Mel-Spectrogram의 경우 0.611(0.439)로 검증과 테스트 모두 후자의 추출 방법이 더 높게 나타났다. 이는 주파수 표현 방식과 정보를 유지하는 방식에 차이가 있으며 주파수 정보를 더 효과적으로 보존할 수 있고 설정된 하이퍼파라미터에서 더 잘 작동하는 기법이라 할 수 있다고 볼 수 있다. 다음 [표 4-10]은 해당 실험에서 가장 정확도가 높은 Mel-Spectrogram의 2-10 세기의 Confusion Matrix이다.

[표 4-10] MFCC와 Mel-Spectrogram 노이즈 증강 Best Confusion Matrix

	Precision	Recall	f1-score	Support
Anger	0.75	0.76	0.76	1146
Fear	0.55	0.36	0.43	1168
Happiness	0.61	0.63	0.62	1152
Neutral	0.55	0.56	0.55	1000
Sad	0.58	0.75	0.65	1125
Accuracy	0.61	-	-	5591
최적 모델 정확도 : 0.611 하이퍼 파라미터 : 'C': 1, 'Gamma': 'Scale', 'Kernel' : 'RBF'				

[표 4-10]과 같이 노이즈 증강 실험에서도 'Anger' 클래스의 정밀도, 재현율, f1-score 모두 높은 값을 보여 다른 클래스 중에서도 가장 확실한 특징을 가진 감정인 것을 알 수 있었다. 또한, 'Sad' 클래스는 재현율이 매우 높으며, 다른 메트릭도 상대적으로 높게 나타났다. 다만, 'Fear' 클래스의 정밀도와 재현율 모두 낮아 특별히 다른 클래스와 구분할만한 감정 특징이 없는 것으로 보인다.

c. 원본·증강 데이터 병합

원본·증강 데이터를 병합한 실험에서는 [표 4-11]과 같이 추출 방법 별, 데이터 셋별로 실험을 진행했다. 두 기법 모두 속도 증강 데이터에서 검증 간 더 낮은 정확도가 도출되었지만 테스트 간에는 검증보다 더 높은 정확도가 나왔다. 이는 검증 단계에서 노이즈 데이터가 특정 감정에서 과적합이 되었다는 것을 고려할 수 있다.

[표 4-11] 원본·증강 데이터 셋 병합 실험 결과

Extraction method	Dataset	Speed rate	Noise rate	Validation (Accuracy)
MFCC	Speed + Raw (74162), Noise + Raw (40452)	0.9-1.1	-	0.576(0.465)
		-	0.1	0.584(0.455)
Mel-spectrogram		0.9-1.1	-	0.596(0.484)
		-	0.1	0.624(0.455)

아래 [표 4-12] 해당 실험에서 가장 정확도가 0.484로 높은 Mel-Spectrogram의 속도 0.9-1.1배의 Confusion Matrix이다. 각 감정 클래스의 결과를 종합하면, 'Anger' 클래스와 'Sad' 클래스에서 비교적 좋은 성능을 보이고 다른 클래스에서는 성능이 떨어지는 것으로 나타났다. 대체로 이전의 실험과 동일한 결과가 도출되었으며 마찬가지로 'Fear' 클래스의 감정 특징을 분석하고 분류하는 것은 어려울 것으로 보인다.

[표 4-12] MFCC와 Mel-Spectrogram 원본·증강 병합 Best Confusion Matrix

	Precision	Recall	f1-score	Support
Anger	0.72	0.76	0.74	2516
Fear	0.52	0.35	0.42	2512
Happiness	0.57	0.55	0.56	2584
Neutral	0.53	0.60	0.56	2101
Sad	0.61	0.72	0.66	2588
Accuracy	0.59	-	-	12301
최적 모델 정확도 : 0.59 하이퍼 파라미터 : 'C': 1, 'Gamma': 'Scale', 'Kernel' : 'RBF'				

d. 증강 데이터 병합

증강 데이터 병합 실험에서는 아래 [표 4-13]과 같이 추출 방법별로 실험을 진행하였다. 다른 실험의 경우 모두 Mel-Spectrogram이 검증과 테스트 간 정확도가 높게 나왔으나 해당 데이터 셋의 경우 테스트 간에 MFCC의 정확도가 더 높게 나타났다. 기존의 데이터 셋의 경우 각각의 증강 특성을 가진 데이터 셋을 따로 병합하였으나 두 증강 기법의 특징을 모두 가진 데이터 셋에서 MFCC 기법의 정확도가 높게 나타난 것은 증강 데이터 간 병합의 경우 MFCC가 음성 데이터 분석을 더 잘한다고 볼 수 있다.

[표 4-13] 증강 데이터 병합 실험 결과

Extraction method	Dataset	Speed rate	Noise rate	Validation (Accuracy)
MFCC	Speed + Noise (6,742)	0.9	10	**0.519 (0.467)**
Mel-spectrogram				0.537 (0.437)

아래 [표 4-14]는 해당 실험에서 가장 정확도가 높은 MFCC의 Confusion

Matrix이다.

[표 4-14] MFCC와 Mel-Spectrogram 증강 병합 Best Confusion Matrix

	Precision	Recall	f1-score	Support
Anger	0.63	0.70	0.66	204
Fear	0.36	0.22	0.28	232
Happiness	0.50	0.42	0.46	226
Neutral	0.48	0.44	0.46	212
Sad	0.54	0.81	0.65	245
Accuracy	0.50	-	-	1119
최적 모델 정확도 : 0.519 하이퍼 파라미터 : 'C': 1, 'Gamma': 'Scale', 'Kernel' : 'RBF'				

'Anger' 클래스는 대부분의 메트릭에서 상대적으로 높았으며, 'Sad' 클래스의 재현율이 굉장히 높아 상대적으로 가장 잘 감지하고 있다는 것을 알 수 있다. 다만, 'Fear' 클래스는 굉장히 낮은 수치를 보여 기존의 다른 데이터 셋보다 더 분류하지 못하는 것을 알 수 있다.

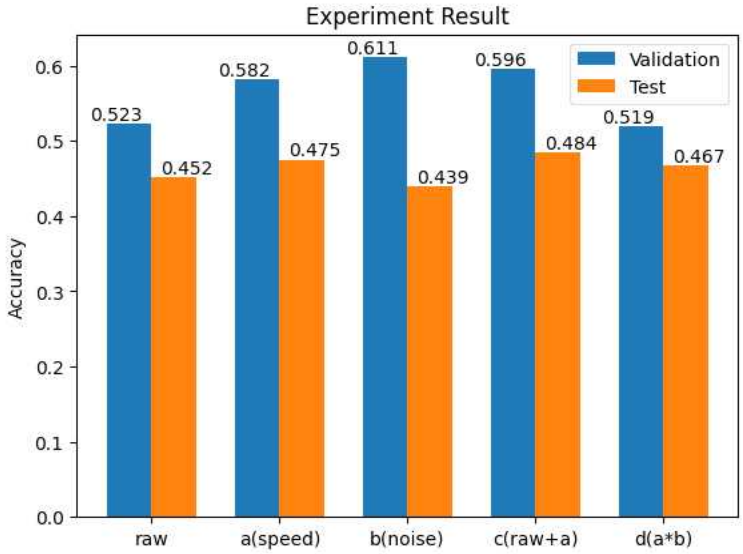
e. 실험 결과 비교

본 연구는 음성 감정 인식에 적합한 데이터 증강 및 특징 벡터 추출 방법을 활용하여 다양한 종류의 데이터 셋에 관한 실험을 진행하였다. 그중에서도 각 실험 데이터 셋 중에서 가장 높은 정확도를 도출한 실험 환경을 [표 4-15]와 [그림 4-2]를 통해 비교하고자 한다.

[표 4-15] 전체 실험 결과 비교표

Experiment	Extraction method	Speed rate	Noise rate	Validation (Accuracy)
raw	Mel-spectrogram	-	-	0.523 (0.452)
a (speed)		0.9-1.1 (1.0 제외)	-	0.582 (0.475)
b (noise)		-	2-10	0.611 (0.439)
c (raw+a)		0.9-1.1	-	**0.596 (0.484)**
d (a*b)	MFCC	0.9	10	0.519 (0.467)

[표 4-15]와 같이 실험 'raw', 'a(speed)', 'b(noise)', 'c(raw+a)', 'd(a*b')에서 가장 높은 정확도를 도출한 실험 결과를 기준으로 데이터 셋을 선택하였다. 원본 데이터를 포함한 'a','b','c' 데이터 셋에서 Mel-Spectrogram이 MFCC보다 나은 성능을 보였다. 또한, 속도를 조절한 증강 데이터 셋이 원본 데이터의 정확도보다 높았으며, 노이즈를 추가한 증강 기법은 원본보다 정확도가 낮았다. 이는 속도 증강 기법은 실제로 빠르거나 느린 속도로 발화할 때 나타나는 언어 특성을 잘 재현하였으며, 증강 과정에서 데이터에 대한 왜곡이 노이즈 증강보다 적었음을 알 수 있다. 이는 b의 데이터 셋에서 관찰되는 특징으로 검증 정확도가 0.611로 가장 높았으나 테스트 정확도는 0.439로 원본 데이터의 정확도보다 낮았으며 실험 중 가장 낮은 정확도를 기록했다. 이는 노이즈 증강 기법을 적용한 데이터 셋은 과적합이 되었다고도 볼 수 있다. 따라서 속도 증강 데이터가 노이즈 증강 데이터보다 정확도가 높은 이유는 데이터의 자연스러움과 정보 손실이 적은 증강 기법을 사용했고, 모델의 특성과 데이터 양이 향상에 기여했을 가능성이 있다고 볼 수 있다. 다음 [그림 4-2]는 비교 실험 그래프를 나타낸다.



[그림 4-2] 전체 실험 결과 정확도 그래프

[그림 4-2]와 같이 전체 실험 중에서 가장 높은 정확도를 보인 데이터 셋은 c(raw+a)이다. 원본 데이터와 속도 증강 데이터를 병합한 실험 환경이 데이터 증강과 벡터 추출 방법을 적용하였을 때 가장 적합하였으며 원본 데이터의 정확도보다 3% 높은 0.484의 결과를 도출하였다. 또한, 공통적인 감정 특징으로는 'Anger'와 'Sad'의 정밀도와 재현율 등 메트릭 수치가 높고 'Fear'는 정밀도와 재현율 모두 낮아 감정 특징을 잘 감지하지 못하였다. MFCC의 경우, 속도와 노이즈 증강 기법의 특징을 모두 가진 데이터 셋에서 더 높은 성능을 보였다. 그러므로 MFCC는 Mel-Spectrogram보다 증강 데이터 간 병합 환경에서 음성 데이터 분석에 더 유리한 환경임을 알 수 있다.

V. 결론 및 향후 연구

본 연구에서는 감정 인식 성능 향상을 목표로 음성 데이터의 속도와 노이즈 증강 기법을 통해 다양한 음성 데이터 처리 실험을 진행하였다. 5가지 감정(Anger, Fear, Happiness, Neutral, Sad)으 구성된 음성을 효과적으로 분류하기 위해 속도와 노이즈의 배율을 적용하고 음성의 특징을 MFCC, Mel-Spectrogram으로 벡터화한 후 3차원 데이터를 2차원으로 변환하였다. 이러한 음성 데이터 셋에 SVM 모델을 활용하였고 GridsearchCV를 통해 최적의 하이퍼파라미터를 조정하였으며 모델의 성능은 Confusion Maxtrix와 테스트 셋으로 검증 및 평가하였다. 실험 데이터 셋은 속도 증강, 노이즈 증강, 원본·증강 데이터 병합, 증강 데이터 병합인 4종류로 구성된다. 실험은 MFCC와 Mel-Spectrogram 특징 추출 방법을 기반으로 속도와 노이즈 증강 배율 등 데이터 조합에 따라 다양하게 진행되었다. 데이터 성능 향상은 음성 원본 데이터의 정확도를 기준으로 하였으며, 3%의 성능 향상을 도출하였다.

실험 결과 각각의 증강 특성을 가진 데이터 셋을 따로 병합한 속도 증강, 노이즈 증강, 원본 데이터와 증강 데이터 병합 실험에서 Mel-Spectrogram이 성능이 더 높았다. 또한, 속도와 노이즈를 증강한 데이터에 원본 데이터를 추가한 실험에서 Mel-Spectrogram이 성능이 더 높았다.

공통적인 감정 특징으로는 'Anger'와 'Sad'의 정밀도와 재현율 등 메트릭 수치가 높고 감정을 잘 분류하였다. 다만, 'Fear' 클래스는 정밀도와 재현율 모두 낮아 해당 감정은 비슷한 감정 클래스가 있거나 특징을 잘 감지하지 못하는 것으로 나타났다. MFCC의 경우, 두 증강 기법의 특징을 모두 가진 데이터 셋에서 더 높은 성능을 보였으며 이는 증강 데이터 간 병합 환경에서 음성 데이터 분석에 더 유리하다고 볼 수 있다.

본 연구는 음성 감정 인식을 위한 다양한 실험을 수행하여 증강 데이터와 특징 벡터 추출 방법을 적용하였다. 향후 연구에서는 특정 감정들을 더 잘 분류할 수 있는 특징 벡터 추출 방법과 다양한 증강 기법을 통해 성능을 보완하고 멀티모달 융합방법을 활용한 감정 인식 연구 간에 다양한 음성 분석 모델을 앙상블하여 감정 인식에 대한 정확도 향상을 목표로 한다.

참고 문헌

[1] 김병건, Improvement Methods of Speech Emotion Recognition with Small Amount of Dataset, 한양대학교 대학원 석사 학위논문, 2021

[2] 최지원 외 3명, 인간의 기본 감정에 따른 어조 탐색과 스펙트럼 분석, Journal of Speech, Media and Communication Research. vol18, no 4. 121-157, 2019

[3] 신보라, 이석필, A Comparison of Effective Feature Vectors for Speech Emotion Recognition, The Transactions of the Korean Institute of Electrical Engineers, Vol. 67, 1364-1369, 2018

[4] Moataz El Ayadi 외 2명, Survey on speech emotion recognition: Features, Classification schemes and databases, Pattern Recognition 44, 572-587, 2011

[5] 김영준 외 3명, Increasing Accuracy of Stock Price Pattern Prediction through Data Augmentation for Deep Learning, The Korea Journal of BigData, 제 4권 2호, 1-2, 2019

[6] Yashpalsing Chavhan 외 2명, Speech Emotion Recognition Using Support Vector Machine, VIT, Pune India, 7, 2010

[7] 문이선, 김형석, Multi-modal Emotion Recognition Using Physiological Sensor and Speech Data, Institute of Control, Robotics and Systems, 635-636, 2023

[8] 김주희, 이석필, Multi-modal Emotion Recognition using Speech Features and Text Embedding, The Transactions of the Korean Institute of Electrical Engineers, Vol. 70, No. 1, 108-113, 2021

[9] Jeongchan Yu 외 4명, Speech Enhancement based on Machine Learning using Speech Features, 한국방송미디어공학회 2023 하계학술대회, 70-71, 2023

[10] KwanYeol Park, Il-Youp Kwakm, Comparative study of data augmentation methods for fake audio detection, The Korean Journal of

APPlied Statistics, Vol.36, No.2, 101-114, 2023

[11] Tom Ko 외 3명, Audio Augmentation for Speech Recognition, Huawei Noah's Ark Research Lab, Hong Kong, China, 2021

[12] 임성수, Data augmentation by local frame rate changes for enhanced speech recognition, 충북대학교 대학원 석사 학위논문, 2022

[13] 양진혁, 김인중, Deep Learning-based Speech Synthesis with Noised Attention, 한국컴퓨터종합학술대회 논문집, 904-906 2019

[14] 윤원정, A Study on Deep Learning-based Voice Recognition Model Using MFCC, 한세대학교 대학원 석사 학위논문, 2021

[15] https://ratsgo.github.io/speechbook/docs/neuralfe/pase

[16] 백승인 외 3명, Effective Noise Reduction using STFT-based Content Analysis, Journal of The Institute of Electronics and Information Engineers Vol.52, NO.4, 2015

[17] 노경민, A study on feature vector for effective emotion Classification of similar emotional speech, 2023년도 대한전기학회 하계학술대회 논문집, 1895-1896, 2023

[18] https://xangmin.tistory.com/61

[19] 조아현, 곽근창, Emotion Recognition Using Deep Learning Based on Transfer Learning from Speech Emotion Signals, 2021년도 대한전기학회 산업전기위원회 추계학술대회 논문집, 196-198 ,2021

[20] https://www.nti-audio.com/en/support/know-how/fast-fourier-transform-fft

[21] https://kr.mathworks.com/help/dsp/ref/dsp.stft.html

[22] https://www.educba.com/matlab-zero-Padding/

[23] https://www.kaggle.com/datasets/ejlok1/CREMAd

감사의 글

신주현 교수님과 함께해주신 존경하는 모든 분들게,

먼저, 저의 석사 학위 논문을 마무리하게 되어 깊은 감사의 말씀을 전하고자 합니다. 신주현 교수님, 김판구 교수님, 최준호 교수님과 모든 박사님들, 가족과 동료들 그리고 학문의 길을 함께해주신 모든 분들께 감사드립니다.

신주현 교수님, 여러 달간 보여주신 지도와 조언에 감사드립니다. 열정적인 지도 덕분에 논문 작성 과정에서 끊임없는 도전과 성장을 경험할 수 있었습니다. 교수님의 헌신적인 가르침 아래에서, 새로운 아이디어를 탐험하고 발전시킬 수 있어 뜻깊었습니다.

그리고 함께 논문 지도에 힘써주신 모든 박사님들께도 감사의 말씀을 전합니다. 각 분야에서 나눠주신 전문 지식과 소중한 피드백은 논문의 방향에 큰 도움이 되었습니다. 여러분의 깊은 이해와 의견 감사드립니다.

또한, 가족과 동료들 그리고 끊임없는 지지와 격려를 보내주신 모든 이들에게도 감사의 인사를 전합니다. 학업과 업무를 동시에 병행하면서 지난 2년간 예민하고 지친 모습을 많이 보여드렸습니다. 그럼에도 항상 넓은 마음으로 이해해주시고 안아주셔서 감사합니다. 여러분의 따뜻한 응원이 없었다면 이 순간에 도달하지 못했을 것입니다.

앞으로의 여정에서도 여러분의 가르침을 잊지 않고 성장하고 발전하는 모습을 보여드리겠습니다. 다시 한번 감사의 인사를 드리며, 앞으로의 도전에 기대와 열정을 가지고 나아가겠습니다.

안 진 성 배상