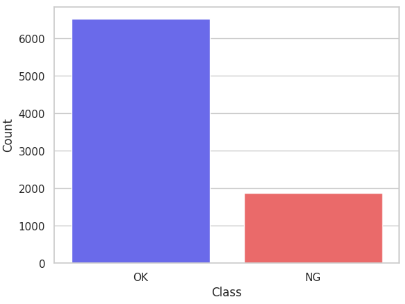
**Abstract**

본 과제는 제조공정에 AI기술을 접목시키고자 한다. Kaggle에서 제공하는 Fan 소음 데이터를 활용해서 정상과 불량을 판정한다. Mel-Spectrogram 변환을 통해 Fan 소음 데이터를 이미지 데이터로 변형하여 CNN 모델을 활용하였다. 또한 각 모델에 하이퍼파라미터 튜닝을 적용하여, 성능을 최적화하였다. CNN모델을 활용한 경우, 지난 과제의 SVM과 FCN을 활용할 때 보다 눈에 띄는 성능 향상이 관찰되었다.

1. **Introduction**

AI기술이 모든 기술 분야에서 핵심 기술로 자리잡고 있다. 특히 제조 공정에서 AI 기술은 결함이 발생하는 제품들을 실제로 사람이 사용해보지 않고, 공정에서 수집되는 데이터를 통해 사전에 검출할 수 있게 한다. 이를 통해 큰 수율 상승을 이끌 수 있다.

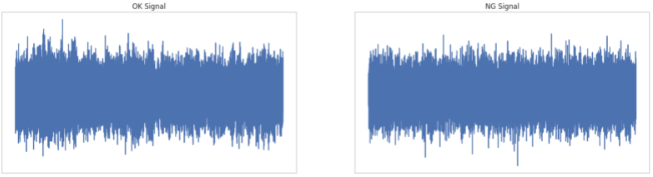
본 과제에서는 Kaggle에서 제공하는 Fan 제조 공정에서 측정된 Audio 데이터를 활용해서 제품의 정상(OK)와 불량(NG)를 판별하는 모델을 개발하고자 한다. <https://www.kaggle.com/datasets/vuppalaadithyasairam/anomaly-detection-from-sound-data-fan>에서 다운받을 수 있는 이 데이터는 6,521개의 정상 데이터와 1,875개의 불량 데이터가 포함되어 있다 (그림 1). 각 Audio 데이터는 Sampling Rate가 16,000인 집음 장치에 의해서 수집되었고, 10초간의 작동 음이 집음 되었다.



**그림 1.**

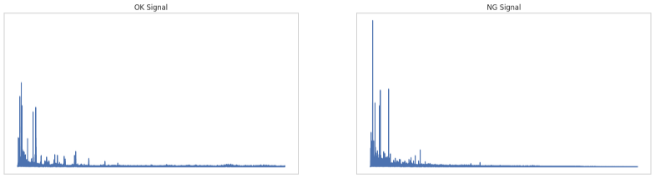
1. **Experiment**
2. **Data EDA**

Audio 데이터는 확장자는 wav이며, 이는 Python에서 제공하는 librosa 패키지를 활용해서 불러올 수 있다. 각 Audio 데이터는 Sampling Rate 16,000에서 10초간 집음 되었으므로 160,000개의 값을 가지는 시계열 데이터로 표현된다. 정상과 불량의 시계열 그래프를 그려보면 다음과 같다 (그림 2).



**그림 2.**

눈으로 보기에는 정상과 불량 시계열에 차이는 거의 나타나지 않는다. Audio 데이터의 경우에는 진동 수가 굉장히 중요한 특징이지만 위의 시계열 그래프에서는 x축이 시간, y축이 진폭으로 진동 수는 표현되지 않는다. Audio 데이터에서 진동 수 정보를 추출하기 위해서 Fourier Transform을 활용할 수 있다. 아래의 그래프는 정상과 불량의 Fourier Transform 결과를 보여준다 (그림 3).

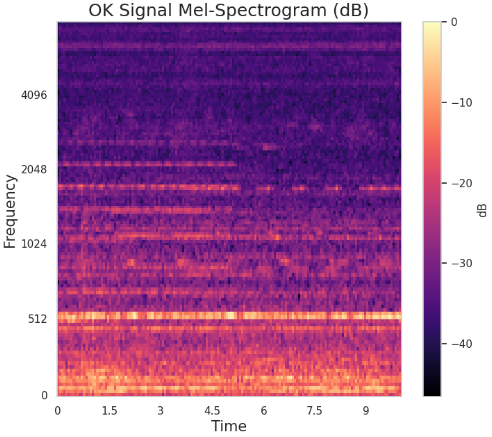


**그림 3.**

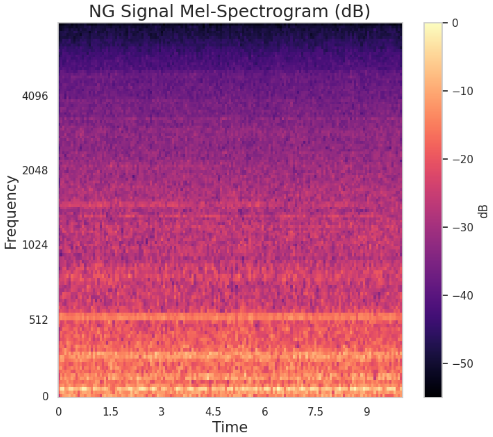
불량의 Fourier Transform의 경우 정상보다 저 주파수가 더 큰 값이 나타나는 것으로 보인다. 여기서 더 나아가 Audio 데이터를 변형하는 방법으로 Spectrogram이 있다. Spectrogram은 Audio 데이터를 x축은 시간, y축은 주파수, z축은 진폭으로 시각화한다. 추가로 Mel Spectrogram은 주파수 영역을 Linear하지 않게 변형을 한다. 이는 사람의 청각의 변화가 주파수 변화를 Linear하게 받아들이지 않기 때문에, 사람의 청각에 맞게 Scale의 변화를 준 것이다. Mel Spectrogram 변화 식은 다음과 같다 (수식 1).

**수식 1. Mel(f) = 2595 log(1+f/700)**

정상과 불량의 Mel Spectrogram은 다음과 같다 (그림 4, 그림5). 이 때, 하이퍼 파라미터로는 n\_fft: 2048, hop\_length: 1024이다. 정상에서의 Mel Spectrogram이 불량에서 보다 가로 선이 좀 더 명확하게 보이고 있다.



**그림 4.**



**그림 5.**

1. **Data Preprocessing**

본 과제에서는 Mel Spectrogram 변환을 통해 얻은 이미지 피처를 활용하고자 한다. 이 때, CNN의 Input으로 활용하기 위해서 Image Size를 128\*128로 맞추도록 한다. 이를 위한 파라미터는 계산식에 따라 n\_mels=128, hop\_length=1250, n\_fft=5000로 결정된다. 최종적으로 Mel Spectrogram으로 변환한 Feature의 차원은 (128,128)이 된다. 이 때 뒤의 128은 주파수를 의미하며, 앞의 128은 시간을 의미한다. 또한 값은 진폭을 의미하며, Scale을 맞추기 위해서 Min Max 정규화를 수행하였다.

데이터를 Train과 Test로 분할할 때, Class Imbalance 문제를 최소화하기 위해서 Train set은 정상 900, 불량 900으로 구성하였고, Test set은 정상 100, 불량 100으로 랜덤 비복원 추출하였다. 이 때 랜덤 시드는 555이다.

1. **Models**

Python의 Pytorch로 CNN 모델을 구성하였다. 성능 향상을 목적으로, CNN Layer의 수를 바꿔가며 실험하였다. 1-Lyaer CNN은 kernel=3, stride=1, padding=1, filter=32의 CNN Layer와 Max-pooling layer를 포함한다. 2-Layer CNN은 동일한 kernel, stride, padding, Max-pooling layer를 포함하며, filter는 32 > 64의 구조를 가진다. 마지막으로 3-Layer CNN은 filter가 32 > 64 >128의 구조를 가진다. 3개의 CNN 모두 CNN Layer를 통과한 후, 2개의 FC Layer를 거쳐 정상과 불량을 판정한다. 또한 학습과정에서 Learning Rate를 조절하기 위해서 Step Scheduler를 활용하였다. 이는 학습 과정이 진행됨에 따라 Learning Rate를 감소시키며, 원활한 학습을 도와준다. 적절한 Layer 수를 결정한 다음에, Optimizer를 Adam, Adagrad, SGD로 변경하여 최적의 Optimizer를 선정한다. 모든 학습과정에서 Loss가 가장 낮았던 Epoch의 모델을 Best 모델로 저장한다.

마지막으로 고정된 파라미터는 다음과 같다. Batch size: 32, Epoch: 20, Initial Learning Rate: 0.001, Weight Decay: 0.000001, Loss: Binary Cross Entropy, Activation: ReLU

1. **Result**

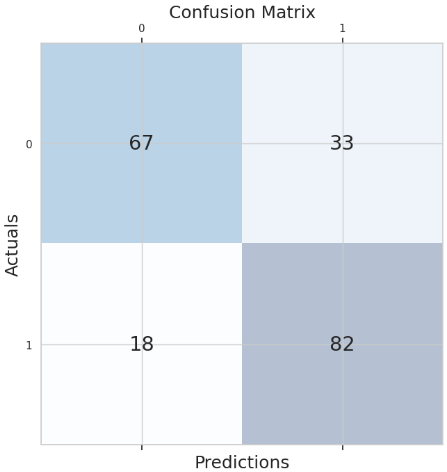
CNN의 레이어 수를 결정하기 위한 실험 결과는 아래와 같다 (표 1). 레이어 수가 2개일 때 가장 좋은 예측 성능을 보여준다. 이제 CNN 레이어의 수를 2개로 고정하고 Optimizer를 변경하여 실험한 결과는 아래와 같다 (표 2). Accuracy 기준으로 Adam이 가장 좋은 예측 성능을 보여준다. 이에 대한 Confusion Matrix 결과는 아래와 같다 (그림 8).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CNN Layers | Accuracy | Precision | Recall | F1 |
| 1 | 0.5 | 0 | 0 | 0 |
| 2 | 0.745 | 0.713 | 0.82 | 0.763 |
| 3 | 0.695 | 0.673 | 0.76 | 0.714 |

**표 1.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Optimizer | Accuracy | Precision | Recall | F1 |
| Adam | 0.745 | 0.713 | 0.82 | 0.763 |
| Adagrad | 0.695 | 0.673 | 0.76 | 0.714 |
| SGD | 0.695 | 0.673 | 0.76 | 0.714 |

**표 2.**



**그림 8.**

1. **Conclusion**

본 과제에서는 Spectrogram을 이미지로 활용하여, CNN 모델로 Fan의 정상과 불량을 판정하는 실험을 진행하였다. 그 결과 FCN과 SVM을 활용했던 이전 과제보다 더 높은 성능을 얻을 수 있었다.