Pusan National University

Computer Science and Engineering

Technical Report 2023-10

심장질환 환자 ECG 데이터 분석을 위한 딥러닝 기법 설계 및 경량화 모델 구축

신다윗

안주현

이재현

지도교수 송길태 교수님

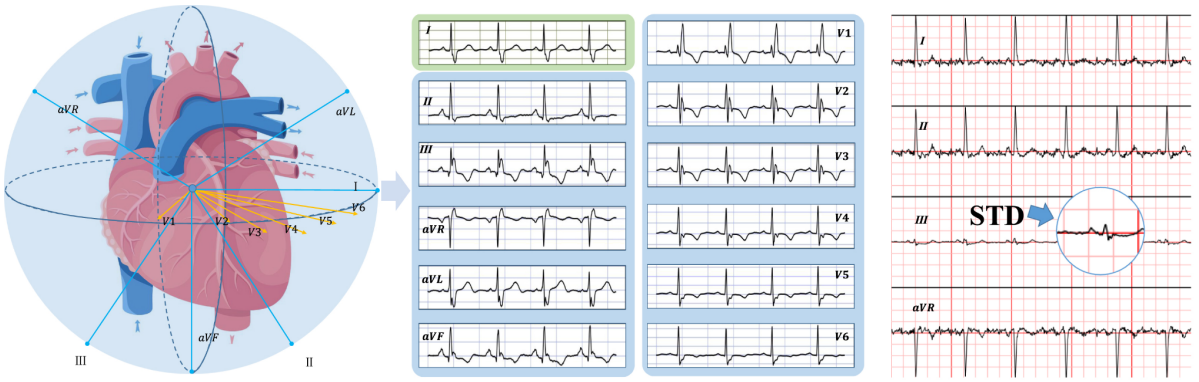
목 차

1. 서론

1.1. 연구 배경

심장질환 환자의 심장 박동 패턴을 분석하고 비정상적인 패턴을 감지하여 분류 대상이 되는 심전도가 어떤 종류의 심장질환에 속하는 지 결정하는 딥러닝 모델을 구축하는 것이다. 부정맥과 관상동맥질환(심장동맥질환)의 진단에 많은 검사들이 이용되고 있으나, 그 중에서도 심전도(Electrocardiogram : ECG)를 이용한 검사는 많은 장점(비침습적, 빠르고 간편한 수행)을 가지며 임상에서 가장 많이 사용되는 검사이다.[1] 이러한 모델은 심장질환 환자의 심장 박동 패턴을 실시간으로 모니터링하는 장치에 설치되어 환자의 건강 상태를 지속적으로 추적하고 긴급한 조치를 취할 수 있는 기회를 제공한다.

더 넓은 응용 범위와 다양한 환경에서 이 모델을 활용하기 위해서는 모델을 경량화하여 임베디드 시스템에도 적용할 수 있어야 한다. 이를 위해 CNN을 이용하여 심전도 데이터를 분석하고 분류할 수 있는 딥러닝 모델을 개발하고, 이를 최적화하여 경량화하여 보는 것은 많은 도움이 될 것으로 생각하여 연구를 진행하게 되었다.



(a) 12-lead ECG from 12 different viewpoints

(b) An ECG waveform with arrhythmia STD (ST-segment depression)

2

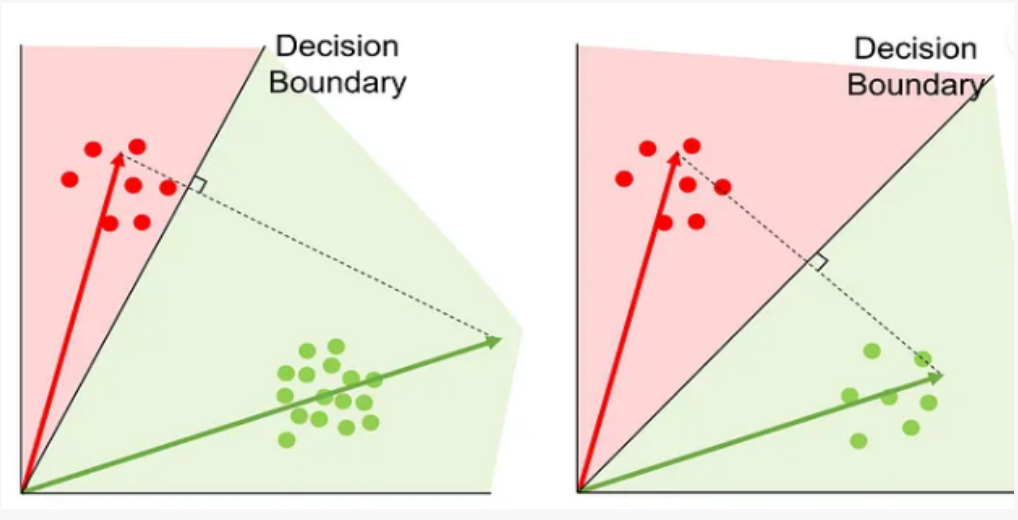
[1] https://www.kci.go.kr/kciportal/landing/article.kci?arti_id=ART002948086

2

https://www.clinicalkey.com/#!/content/playContent/1-s2.0-S001048252300968X?scrollTo=%23hl0002378

1.2. 기존 문제점

딥러닝 모델을 효과적으로 학습하기 위해서는 대량의 ECG 데이터가 필요하다. 기존에는 데이터 수집 및 정제에 많은 시간이 들 수 있으며, 품질 좋은 데이터 확보가 어려울 수 있다. 실제 수집한 데이터에서 불균형이 발생할 수 있다. 이는 일반화를 어렵게 하고 과적합의 위험이 있을 수 있다. 또한 데이터에서 개인 정보를 포함할 수 있으므로, 데이터 수집 및 저장 과정에서 개인정보 보호에 대한 주의가 필요하다.



CNN 모델의 성능을 높이기 위해 구조와 하이퍼 파라미터를 최적화하는 것은 어려운 문제이다. 여러 layer, kernel의 크기, pooling 유형, activation 함수, dropout 등 수많은 모델을 생성할 수 있다. 또한 복잡한 모델은 훈련 데이터에 overfitting 될 가능성이 크고 과도한 계산이 있어야 한다. 적절한 Architecture 선택과 모델의 복잡성을 관리하려면 실험적인 접근과 실험 결과를 통한 지속적인 평가가 필요하다.

딥러닝 모델을 이용한 심전도 데이터 분석은 성능이 좋아도 모델의 크기와 연산량이 크기 때문에 모델의 배포와 실행에 어려움을 겪고 있다. 이를 위해 경량화가 필요하며, 다양한 심장질환 패턴을 고려해야 한다. 또한, 이러한 모델은 모바일 기기에서의 사용에는 적합하지 않다. 베이스 모델을 만들고[3] 리소스가 제한된 장치에 모델을 넣기 위해 특징 기반 지식 증류**(FKD)** 경량화를 적용해 보려고 인터넷을 통한 자료 조사를 진행하였다.[4]

[3] https://www.degruyter.com/document/doi/10.1515/jisys-2023-0002/html

[4] https://www.sciencedirect.com/science/article/abs/pii/S0020025523012136

1.3. 연구 목표

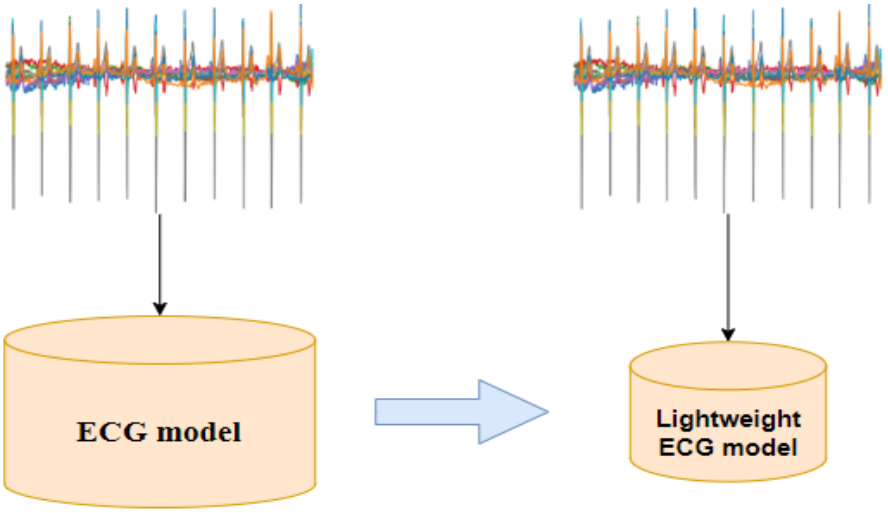


그림 1. ECG 분석 모델의 경량화

ECG (심전도) 데이터를 처리하기 위한 CNN(합성곱 신경망)을 활용하여 심전도 데이터를 분석하고 분류할 수 있는 모델을 설계 및 개발하고, 모델 경량화를 통해 모델의 크기를 줄이고 메모리 및 연산 요구 사항을 최소화하여 실제 제한된 환경에서 동작할 수 있는 가벼운 모델을 학습시키는 것이 주요 목표이다.

2. 연구 배경

2.1. 사용한 데이터

PTB-XL 데이터 세트[5]를 사용했다. PTB-XL ECG 데이터 세트는 10초 길이의 18,869명의 환자로부터 얻은 임상 12-lead ECG로 구성된 데이터 세트이다.

데이터의 유용성과 균일한 성별 데이터 분포는 이 모델을 학습시키는 과정에서 매우 중요한 역할을 한다. 이 데이터 세트는 남성과 여성 데이터가 거의 동등한 비율로 포함되어 있기 때문에 두 성별에 대한 정보를 골고루 학습할 수 있다.

또한, 이 데이터 세트는 0세부터 95세까지의 연령 범위를 포함하고 있으며, 중앙값이 62세이고 분위간 범위가 22세인데, 이는 다양한 연령 그룹을 포함하고 있음을 나타낸다. 이 다양한 연령대를 포함한 데이터는 모델이 다양한 연령 그룹의 특성과 패턴을 학습할 수 있도록 도와주며, 결과적으로 모델의 성능을 향상하고 편향을 방지하는 데 기여할 수 있다.

[5] 데이터 출처: https://physionet.org/content/ptb-xl/1.0.3/

기록 번호 **#Records**	분류할 **Superclass**	설명
9514	NORM	Normal ECG
5469	MI	Myocardial Infarction
5235	STTC	ST/T Change
4898	CD	Conduction Disturbance
2649	HYP	Hypertrophy

표 1. PTB-XL 데이터 세트에 대한 표

NORM (Normal ECG): 이 슈퍼클래스는 정상적인 ECG 결과를 의미하고, 이 범주에 속하는 기록들은 심장 활동이 정상적으로 기록되었으며 심전도에 이상이 없음을 나타낸다.

MI (Myocardial Infarction): MI 슈퍼클래스는 심근경색(Myocardial Infarction)으로 인한 ECG 결과를 나타낸다. 이 범주에 속하는 기록들은 심근경색의 증상을 보이며, 이는 심장 근육에 손상이 있음을 나타낸다.

STTC (ST/T Change): ST/T 변화 슈퍼클래스는 ST 세그먼트와 T 웨이브에 변화가 있는 ECG 결과를 나타낸다. 이러한 변화는 심장의 전기적인 활동에 영향을 미치며 다양한 의료 상황을 나타낼 수 있다.

CD (Conduction Disturbance): 이 슈퍼클래스는 전도 장애(Conduction Disturbance)에 기인한 ECG 결과를 나타낸다. 심장의 전기적인 전달에 이상이 있는 경우, 이러한 변화가 나타난다.

HYP (Hypertrophy): Hypertrophy 슈퍼클래스는 심근 비대(Hypertrophy)로 인한 ECG 결과를 나타낸다. 심근 비대는 심장 근육의 증대를 의미하며, 이러한 변화가 ECG에서 확인될 수 있다.

파형 파일은 1μV/LSB 분해능과 500Hz 샘플링 주파수에서 16비트 정밀도를 갖춘 WFDB(WaveForm DataBase) 형식으로 저장된다(records500/). 사용자의 편의를 위해 100Hz의 샘플링 주파수(records100/)로 다운샘플링 된 파형 데이터 버전도 있다. 본 연구에서는 코드를 구현할 때 다운샘플링 된 파형 데이터를 기반으로 학습을 진행하였다. 그래서 코드에서 데이터를 읽을 때 100Hz로 읽도록 설정했다.

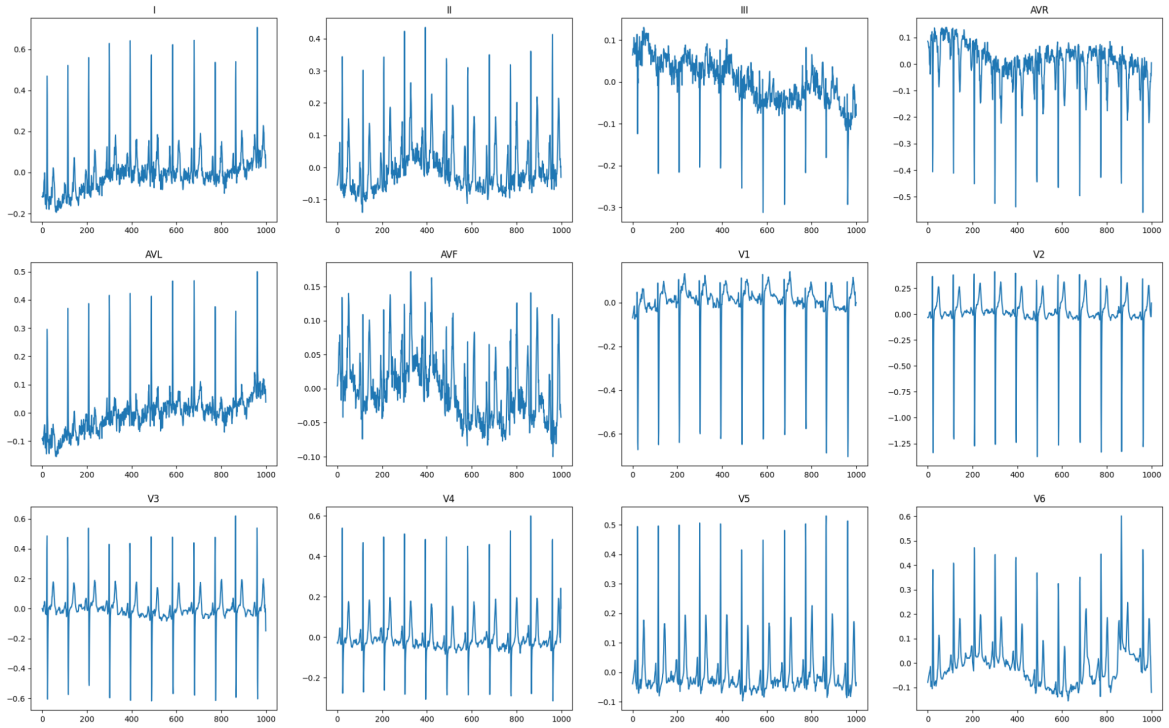


그림 2. ECG 데이터 샘플 1에 대한 각 신호를 출력한 그래프

심전도(ECG)는 발목 및 손목에 있는 12개의 전극에서 측정한 파동 정보를 바탕으로 나타낸 파형으로, 학습에 사용할 데이터는 파동의 정보가 [1000, 12]의 2차원 형태로 저장되어 있다. 이 데이터는 전극에서 얻은 파형 정보를 시계열적으로 포함하고 있다.

2.2. 데이터 전처리

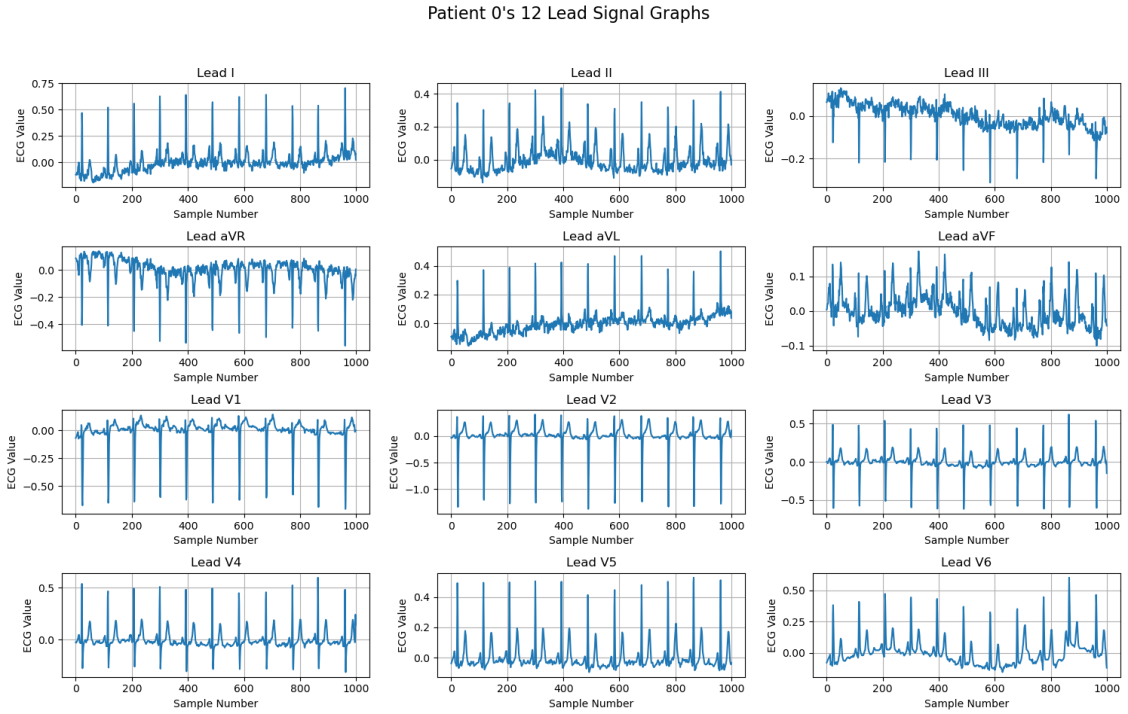


그림 3. 12-lead 데이터의 환자 0번의 그래프

약 2만 개의 임상 12-리드 ECG를 전부 그래프로 출력하여 데이터가 잘못 기록된 것을 살펴보았지만 문제는 없었고, 정답 레이블이 없는 인스턴스들이 있어서 모델을 학습할 때 제외하여 문제가 생기지 않도록 하였다.

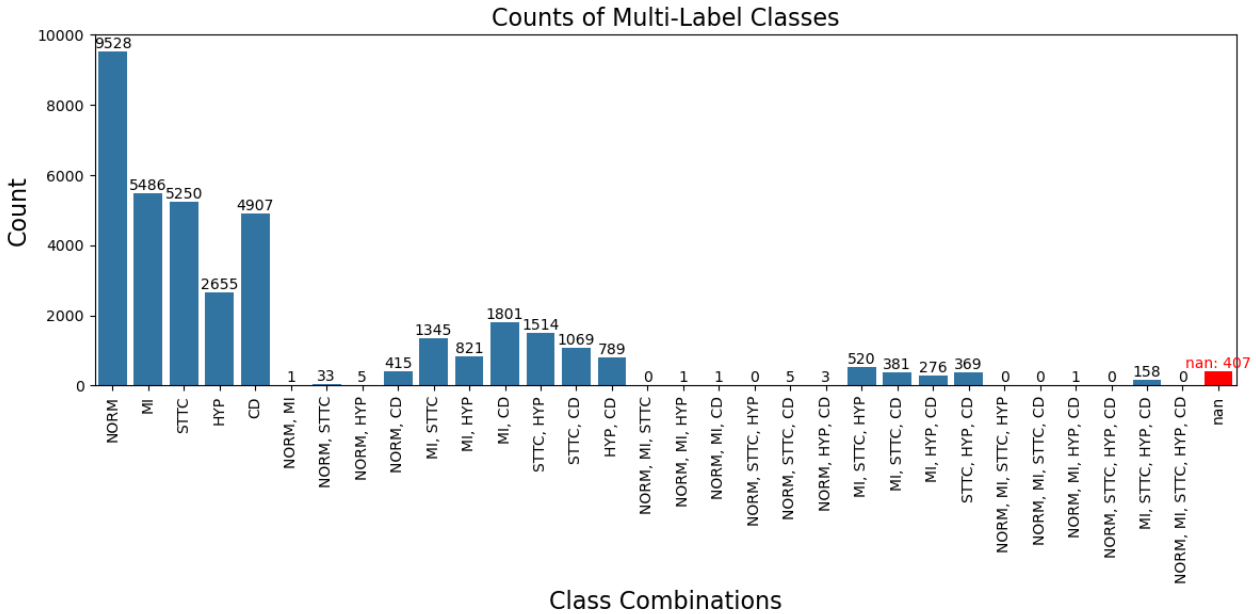


그림 4. ECG 데이터 세트에 대한 Label Counts를 나타낸 그래프

"웨이블릿 임계값 방법과 웨이블릿 분해 재구성 알고리즘"을 적용한 예시 사이트가 있지만 ECG 신호의 잡음(노이즈)은 제거하지 않은 상태로 진행하였다.

⋆데이터 손실 없음 : 원본 ECG 데이터를 그대로 사용하기 때문에 어떠한 정보도 손실하지 않는다.

⋆정보 손실 가능성 : 노이즈 제거 과정에서 신호의 일부 정보가 손실될 수 있으므로 핵심 정보가 사라질 우려가 있다.

위 내용을 고려하여 제거하지 않는 방향으로 진행하였다.

하지만 노이즈를 제거하면 모델이 신호의 핵심 특성에 더 집중할 수 있으며, 분류 성능이 향상을 노려볼 수 있으나 ECG 데이터에 관련된 의학적 지식이 없으므로 안 하는 게 좋다고 판단하여 그렇게 진행하였다.

3. 연구 내용

3.1. 모델 설계 및 학습

Multi-label 분류:

Multi-label 분류는 하나의 입력 샘플이 여러 개의 클래스 레이블에 속할 수 있는 경우에 사용된다. 예를 들어, 이미지에 대한 태그를 예측하거나 문서에 대한 주제를 예측하는 것 등이 있다.

활성화 함수: **Sigmoid** 함수를 사용한다. 시그모이드 함수는 각 클래스에 대한 확률을 독립적으로 계산할 수 있도록 한다.

손실 함수: **Binary Cross-Entropy** 손실 함수를 사용한다. 이 함수는 각 클래스에 대한 예측과 실제 레이블 간의 손실을 계산하는 데 사용된다.

Multi-class 분류:

Multi-class 분류는 하나의 입력 샘플이 여러 개의 클래스 중 하나에만 속하는 경우에 사용된다. 예를 들어, 손 글씨 숫자 분류나 각 이미지에 대한 단일 클래스 레이블 할당 등이 있다.

활성화 함수: **Softmax** 함수를 사용한다. Softmax 함수는 여러 클래스 중 하나를 선택하기 위한 확률 분포를 생성한다.

손실 함수: **Categorical Cross-Entropy** 손실 함수를 사용한다. 이 함수는 모델의 출력과 실제 클래스 레이블 간의 손실을 계산하며, 다중 클래스 분류 작업에 적합하다.[6]

Multi-label data 분류를 위해 기존에 사용되던 K-Fold 방식을 적용하는 것 대신에 iterative-stratification[7]을 이용하여 학습을 진행하였다. 그리고 모델의 과적합을 방지하고 학습 시간을 단축하기 위해서 모델의 훈련을 중간에 중단하는 Early stopping을 사용했다.

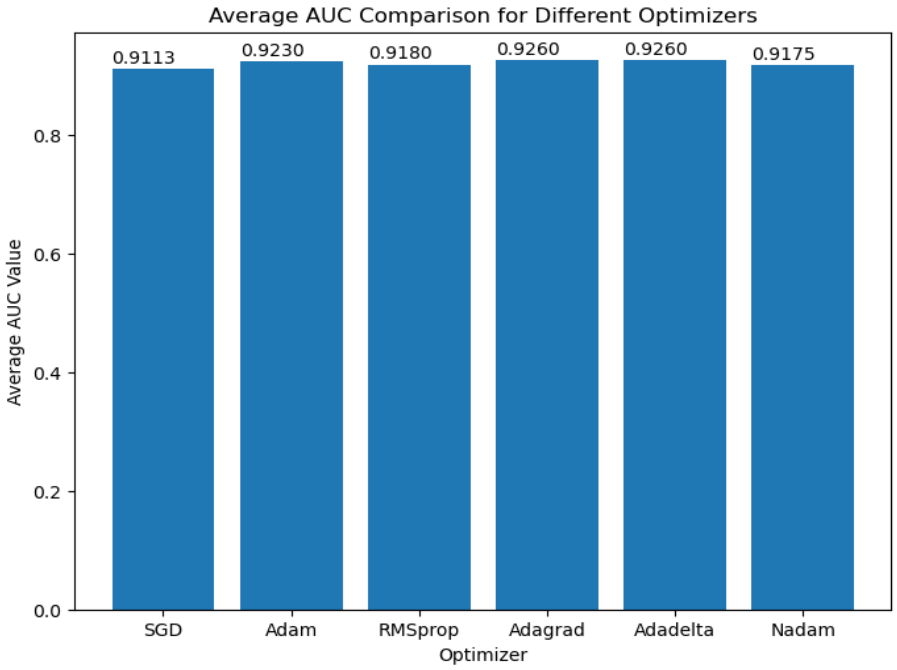


그림 5. ECG 데이터 세트에 대한 Label Counts를 나타낸 그래프

기계 학습 및 딥 러닝 모델에서 모델 파라미터를 조정하여 손실 함수를 최소화하는 데 사용되는 알고리즘을 다양하게 사용해 보았다. Optimizer의 함수 중에서 Adamdelta가 손실함수 중에서 가장 좋았기 때문에 Adamdelta를 사용했다. "Adamdelta"는, 학습률(learning rate)을 동적으로 조절하고 모멘텀(Momentum)을 사용하여 학습 과정을 최적화한다.

[6] https://wordbe.tistory.com/46

[7] https://github.com/trent-b/iterative-stratification

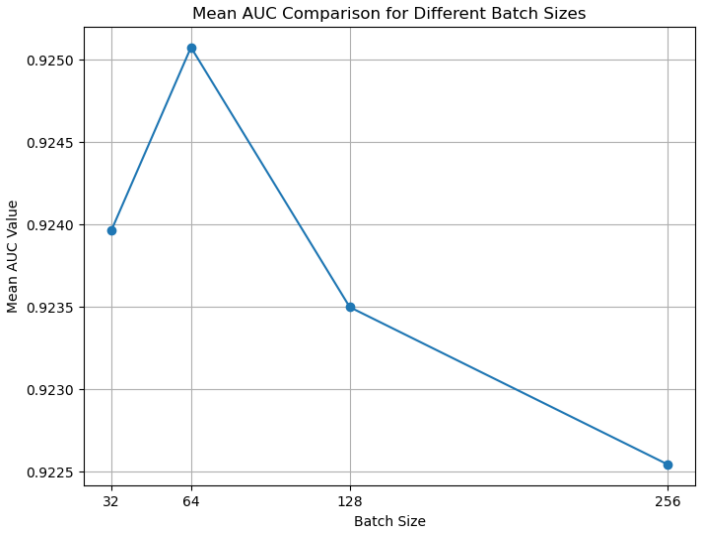


그림 6. ECG 데이터 세트에 대한 Label Counts를 나타낸 그래프

배치 크기는 64에서 가장 높은 성능을 가졌기 때문에 64로 설정했다.

Inception을 이용한 **CNN** 모델

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	[(None, 1000, 12)]	0	[]
conv1d (Conv1D)	(None, 1000, 32)	2720	['input_1[0][0]']
batch_normalization (BatchNorm alization)	(None, 1000, 32)	128	['conv1d[0][0]']
conv1d_1 (Conv1D)	(None, 1000, 128)	20608	['batch_normalization[0][0]']
max_pooling1d (MaxPooling1D)	(None, 500, 128)	0	['conv1d_1[0][0]']
batch_normalization_1 (BatchNo rmalization)	(None, 500, 128)	512	['max_pooling1d[0][0]']
conv1d_2 (Conv1D)	(None, 500, 64)	57408	['batch_normalization_1[0][0]']
max_pooling1d_1 (MaxPooling1D)	(None, 250, 64)	0	['conv1d_2[0][0]']
batch_normalization_2 (BatchNo rmalization)	(None, 250, 64)	256	['max_pooling1d_1[0][0]']
conv1d_3 (Conv1D)	(None, 250, 64)	20544	['batch_normalization_2[0][0]']
max_pooling1d_2 (MaxPooling1D)	(None, 125, 64)	0	['conv1d_3[0][0]']
batch_normalization_3 (BatchNo rmalization)	(None, 125, 64)	256	['max_pooling1d_2[0][0]']
conv1d_5 (Conv1D)	(None, 125, 64)	4160	['batch_normalization_3[0][0]']
conv1d_7 (Conv1D)	(None, 125, 64)	4160	['batch_normalization_3[0][0]']
max_pooling1d_3 (MaxPooling1D)	(None, 125, 64)	0	['batch_normalization_3[0][0]']
conv1d_4 (Conv1D)	(None, 125, 64)	4160	['batch_normalization_3[0][0]']
conv1d_6 (Conv1D)	(None, 125, 64)	12352	['conv1d_5[0][0]']
conv1d_8 (Conv1D)	(None, 125, 64)	20544	['conv1d_7[0][0]']
conv1d_9 (Conv1D)	(None, 125, 64)	4160	['max_pooling1d_3[0][0]']
concatenate (Concatenate)	(None, 125, 256)	0	['conv1d_4[0][0]', 'conv1d_6[0][0]', 'conv1d_8[0][0]', 'conv1d_9[0][0]']
flatten (Flatten)	(None, 32000)	0	['concatenate[0][0]']
dense (Dense)	(None, 128)	4096128	['flatten[0][0]']
batch_normalization_4 (BatchNo rmalization)	(None, 128)	512	['dense[0][0]']
dense_1 (Dense)	(None, 128)	16512	['batch_normalization_4[0][0]']
batch_normalization_5 (BatchNo rmalization)	(None, 128)	512	['dense_1[0][0]']
dense_2 (Dense)	(None, 5)	645	['batch_normalization_5[0][0]']

Total params: 4,266,277
Trainable params: 4,265,189
Non-trainable params: 1,088

그림 7-1. Inception 모델을 이용한 1차원 CNN 모델의 구조

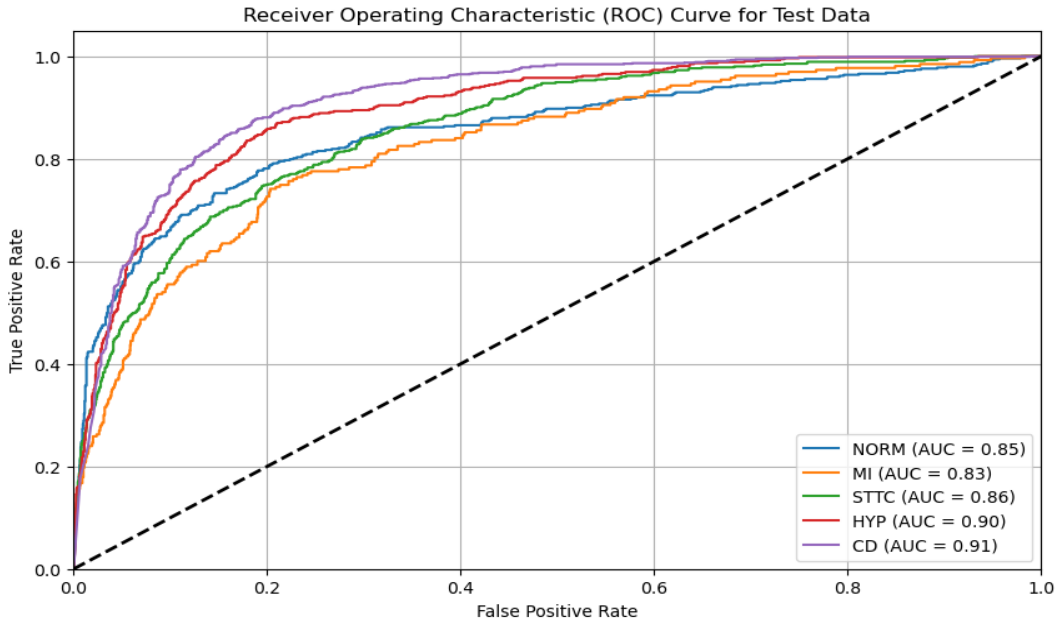


그림 7-2. Inception을 이용한 CNN 모델의 Validation data에 대한 ROC Curve

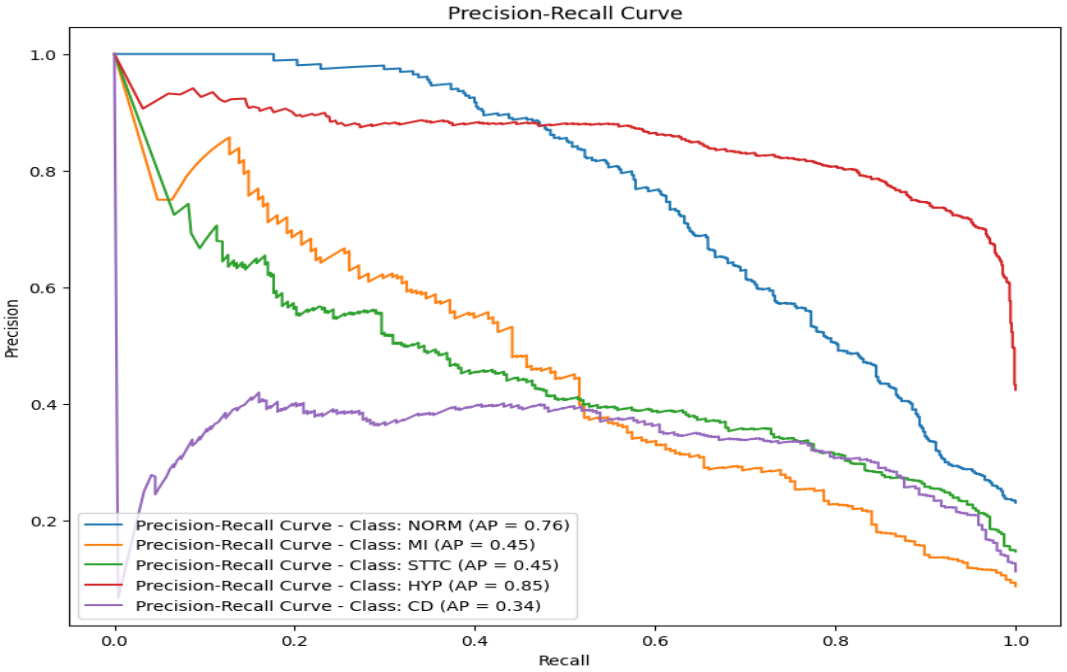


그림 7-3. Inception 을 이용한 CNN 모델의 Validation data에 대한 Precision-Recall Curve

인셉션을 이용한 모델의 파라미터 수는 약 426만개이고, AUC의 평균은 87%이다. 이는 인셉션을 이용하지 않고 1차원 CNN 모델만 사용한 경우보다 AUC의 평균이 낮으므로 모델 설계 과정에서 인셉션을 사용하지 않기로 했다.

일반 **CNN** 모델

Layer (type)	Output Shape	Param #
conv1d_52 (Conv1D)	(None, 996, 50)	3050
max_pooling1d_42 (MaxPooling1D)	(None, 498, 50)	0
conv1d_53 (Conv1D)	(None, 489, 100)	50100
max_pooling1d_43 (MaxPooling1D)	(None, 244, 100)	0
dropout_10 (Dropout)	(None, 244, 100)	0
conv1d_54 (Conv1D)	(None, 240, 150)	75150
max_pooling1d_44 (MaxPooling1D)	(None, 120, 150)	0
conv1d_55 (Conv1D)	(None, 111, 200)	300200
max_pooling1d_45 (MaxPooling1D)	(None, 55, 200)	0
global_average_pooling1d_9 (GlobalAveragePooling1D)	(None, 200)	0
dropout_11 (Dropout)	(None, 200)	0
dense_9 (Dense)	(None, 5)	1005

Total params: 429505 (1.64 MB)
Trainable params: 429505 (1.64 MB)
Non-trainable params: 0 (0.00 Byte)

그림 8-1. 일반적인 1차원 CNN 모델의 구조

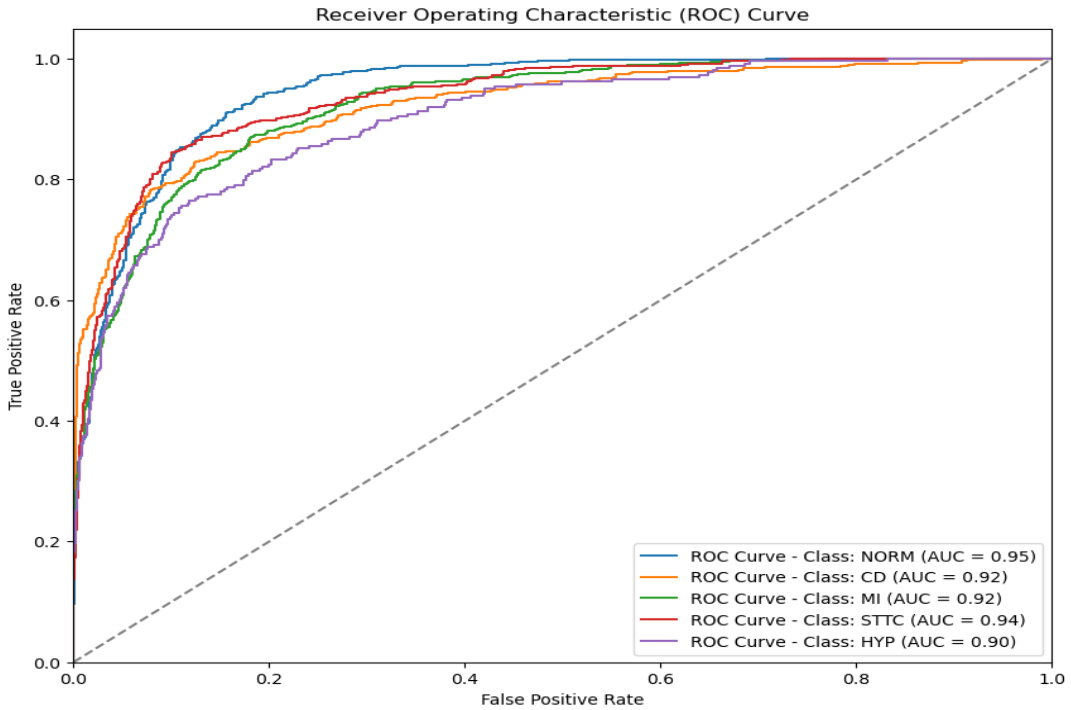


그림 8-2. 일반적인 1차원 CNN 모델의 Validation data에 대한 ROC Curve

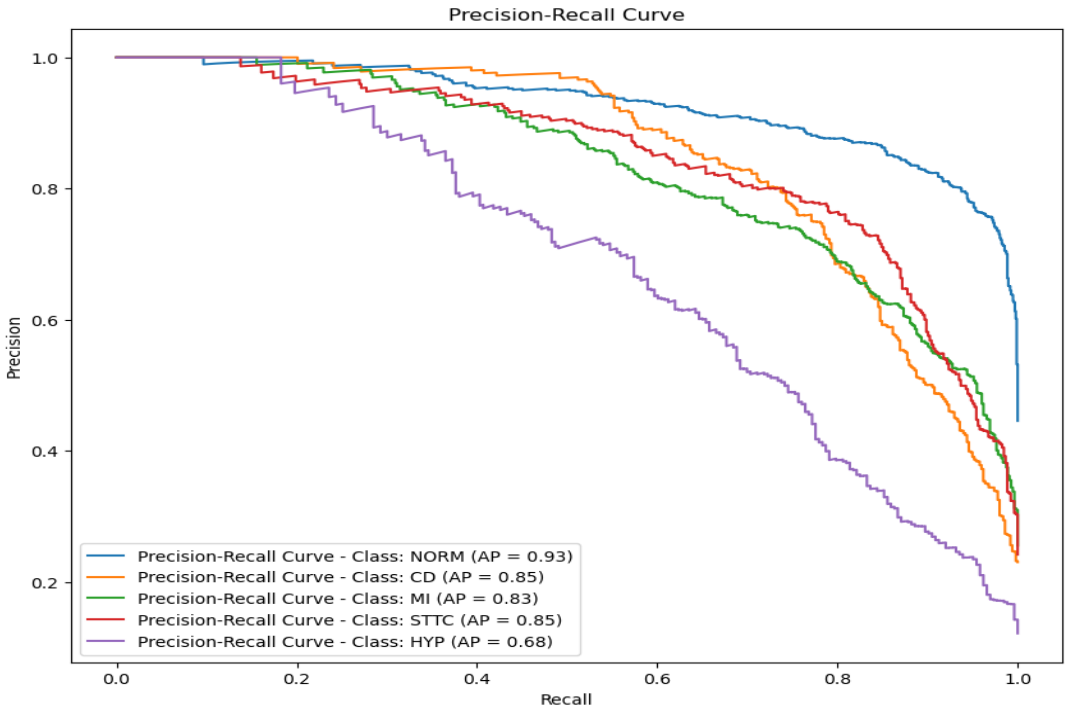


그림 8-3. 일반적인 1차원 CNN 모델의 Validation data에 대한 Precision-Recall Curve

인셉션을 이용하지 않은 일반적인 1차원 CNN 모델의 파라미터 수는 약 43만 개이고, AUC의 평균은 92.6%이다. 이는 인셉션을 이용한 모델보다 AUC의 평균이 높았고, 파라미터 수가 약 50만 개에서 100만 개 사이의 1차원 CNN 모델을 바탕으로 경량화를 진행하기로 했다.

```
Size of the saved model file 'inception_cnn_model_with_batchnorm.h5': 48.99 MB
```

모델의 크기를 보면 48.933MB로 상당히 많은 공간을 차지함을 알 수 있다.

3.2. 모델 경량화

모델 경량화(Model Lightweighting)는 머신 러닝 모델의 크기와 연산량을 줄여서 모델을 더 가볍게 만드는 기술이다. 주로 모델을 배포하거나 모바일 장치, 에지 디바이스, 임베디드 시스템 등의 리소스가 제한된 환경에서 사용하기 위해 모델을 최적화하는 데 사용된다.

경량화를 통해 모델의 파라미터 수와 필요한 연산량을 줄이는 방식으로 모델이 필요로 하는 메모리 공간을 줄인다. 그리고 경량화된 모델은 더 빠른 추론 속도를 가지므로 실시간 응용 프로그램에서 더 빠른 결과를 얻을 수 있으며, 모바일 장치나 배터리가 있는 장치에서 모델을 실행할 때 에너지 소비를 줄이고 배터리 수명을 연장할 수 있다.

3.2.1. 가지치기(Pruning)

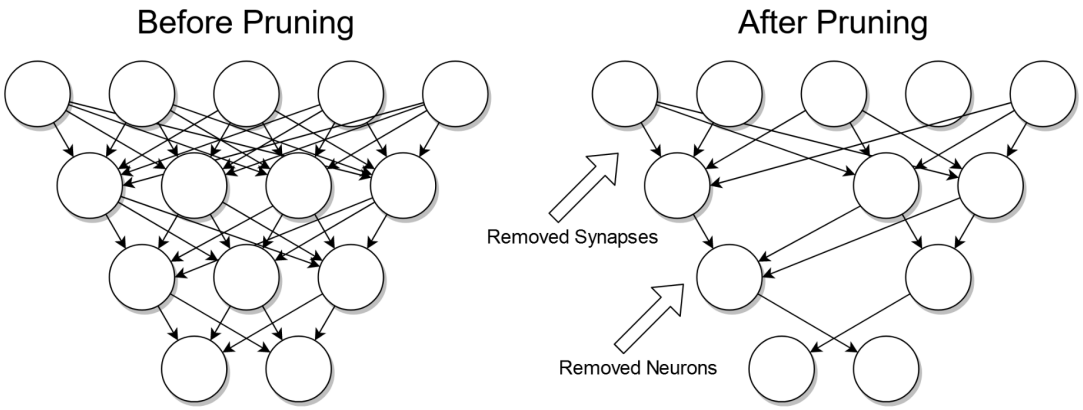


그림 9. 가지치기(Pruning)를 나타낸 그림[8]

가지치기(Pruning)는 모델의 크기와 복잡성을 줄이고 과적합(Overfitting)을 방지하기 위한 정규화(Regularization) 기법으로, 모델을 학습하는 과정 중에 불필요한 가중치(weight) 및 연결(connection)을 제거하여 모델을 더 간단하게 만들 수 있다. 이 과정을 통해 모델의 크기를 줄일 수 있어서 모델을 경량화하는 효과도 가질 수 있다.

3.2.2. 양자화(Quantization)

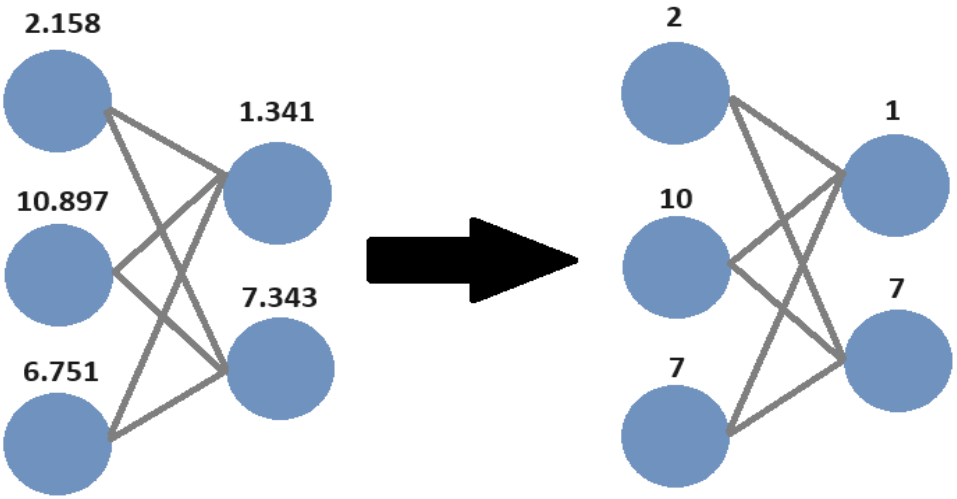


그림 10. 양자화(Quantization)를 나타낸 그림

[8] 출처: https://commons.wikimedia.org/wiki/File:Before_after_pruning.png

양자화(Quantization)는 머신 러닝 모델의 가중치 및 활성화(activation)와 같은 부동 소수점 수치를 더 작은 비트 수의 정수나 다른 형식의 수치로 근사화하는 프로세스를 말한다. 양자화는 특히 Edge Devices 또는 모바일 기기와 같이 제한된 하드웨어 리소스를 가진 환경에서 머신 러닝 모델을 배포하고 실행하는 데 유용하다. 그러나 양자화를 진행할 때, 양자화 수준과 정밀도를 적절하게 선택해야 한다. 너무 과도하게 양자화하면 모델의 성능이 저하될 수 있기 때문이다.

3.2.3. 지식 증류(Knowledge Distillation)

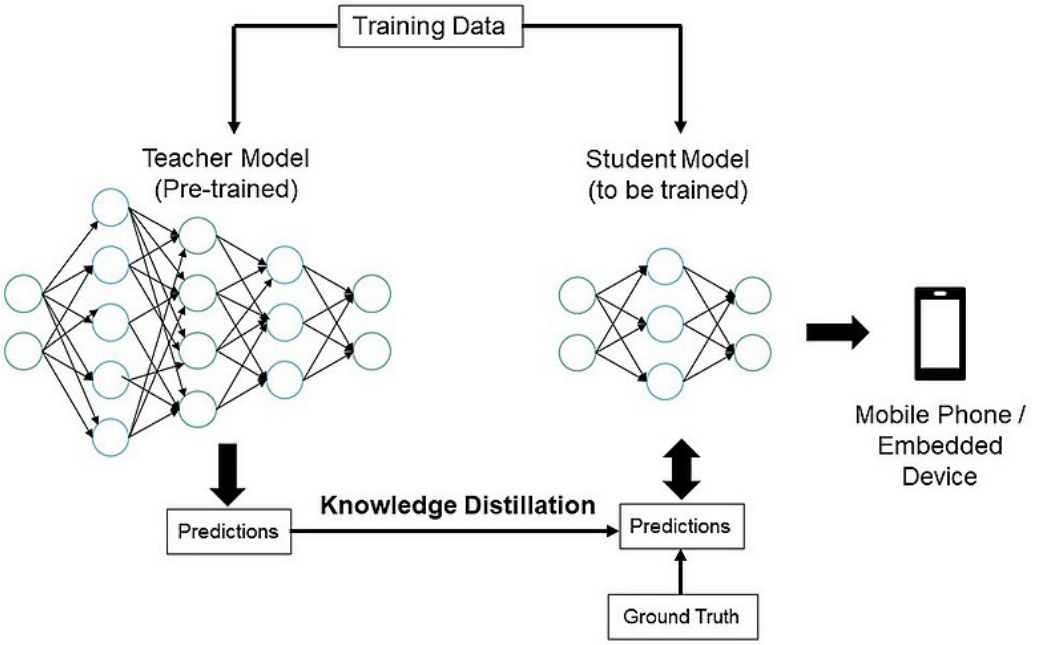


그림 11. 지식 증류(Knowledge Distillation))를 나타낸 그림[9]

머신러닝에서 지식 증류(Knowledge Distillation)란 큰 모델(Teacher)의 지식을 작은 모델(Student)에게 전달하여 작은 모델을 훈련시키는 기술이다. 이 방법은 작은 모델을 더 빠르게 실행하고 메모리 효율성을 향상시키면서, 큰 모델의 성능과 일반화 능력을 작은 모델에게 전달하는 데 사용된다.

[9] 출처 : https://commons.wikimedia.org/wiki/File:Knowledge_Distillation_Flow_Chart.jpg

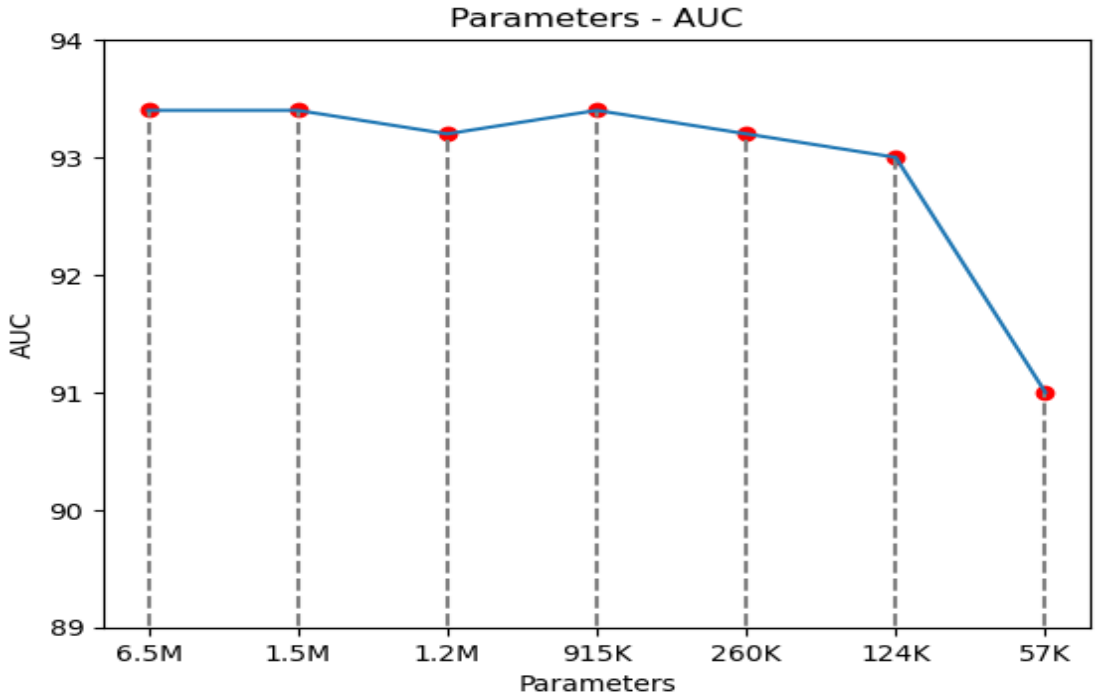


그림 12. 파라미터 수에 따른 AUC 값을 나타낸 그래프

파라미터 수	6.5M	1.5M	1.2M	915K	260K	124K	57K
AUC (%)	93.4	93.4	93.2	93.4	93.2	93.0	91.0

표 2. 파라미터 수에 따른 AUC 값을 나타낸 그래프

위 그림 10과 표 2는 각각 2개에서 4개의 Convolutional Layer로 구성된 기본적인 1차원 CNN 모델에 대하여 파라미터 수에 따른 AUC 값의 변화를 나타내는 그래프와 표이다. AUC 값은 각 label의 AUC 값들의 평균이다. 파라미터 수가 약 650만 개에서 260만 개까지는 약 93.4%의 비슷한 AUC 값을 보여주다가 파라미터 수가 57만 개인 경우 AUC 값이 약 91%로 떨어졌다. 따라서 파라미터 수가 약 650만 개 정도인 모델을 Teacher 모델로, 파라미터 수가 약 57만 개 정도인 모델을 Student 모델로 지식 증류를 진행하였다.

4. 연구 결과 분석 및 평가

4.1. 가지치기(Dropout)

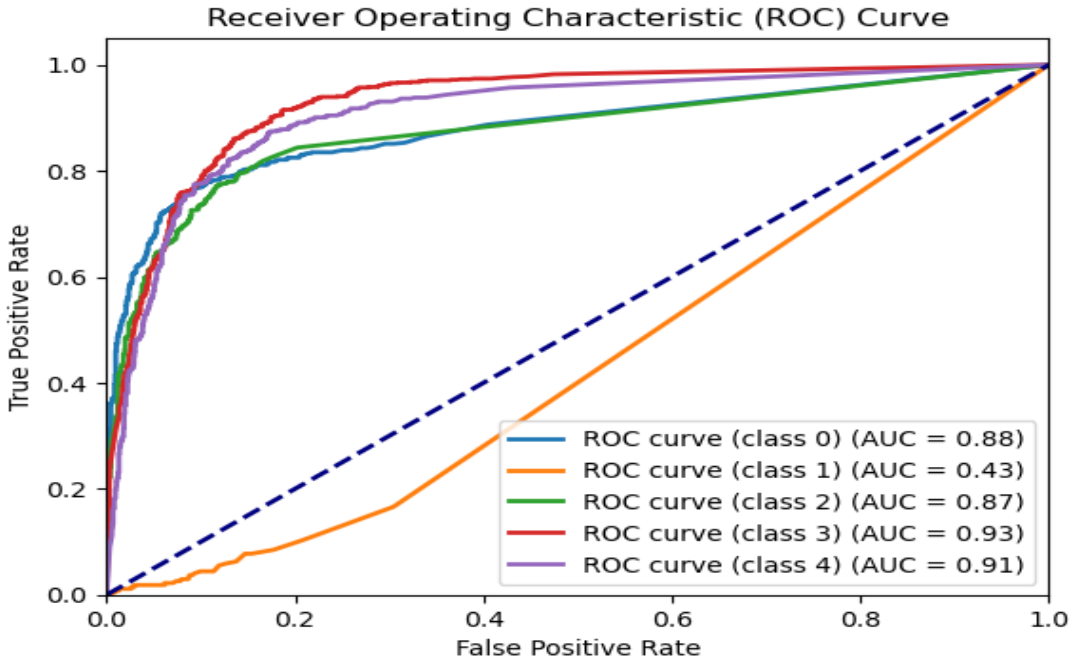


그림 13-1. 가지치기(Pruning) 적용 후 Validation data에 대한 ROC Curve

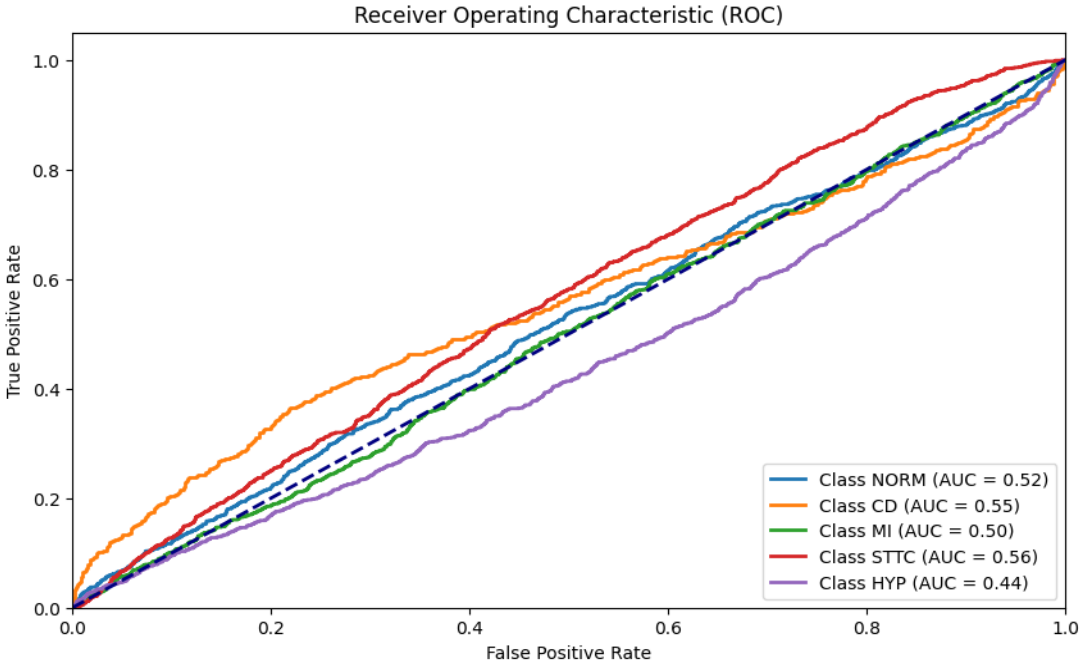


그림 13-2. 가지치기(Pruning) 및 양자화 적용 후 Validation data에 대한 ROC Curve

prune의 하이퍼 파라미터 값을 0.2로 설정한 결과로 ROC 값 평균 51.4%의 값을 얻었다.

Pytorch에서 가지치기(pruning)는 가중치 텐서의 값들 중에서 pruning 된 가중치의 값을 별도의 masking 텐서를 이용하여 0으로 만들기 때문에 파라미터 수에 영향이 없으므로 경량화의 효과를 얻을 수 없다. 그리고 모델의 추론 과정에서 원래 가중치 텐서와 별개로 masking 텐서를 사용해야 하기 때문에 메모리 사용량은 오히려 늘어날 수 있다고 판단된다. 따라서 모델의 경량화와는 거리가 먼 방법이라고 할 수 있다.[10]

4.2. 양자화(Quantization)

파라미터 수가 약 61만 개인 일반적인 CNN 모델을 학습시키고, 학습된 모델을 양자화하는 것으로 진행하였다.

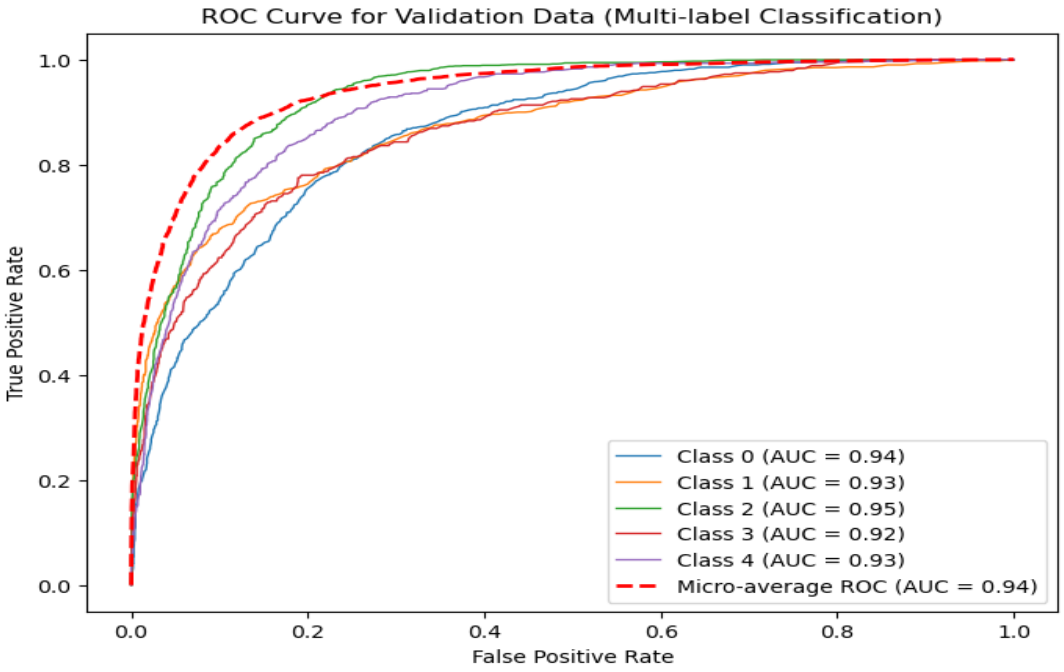


그림 14-1. 일반적인 CNN 모델의 양자화 전 Validation data에 대한 ROC Curve

[10] https://pytorch.org/tutorials/intermediate/pruning_tutorial.html

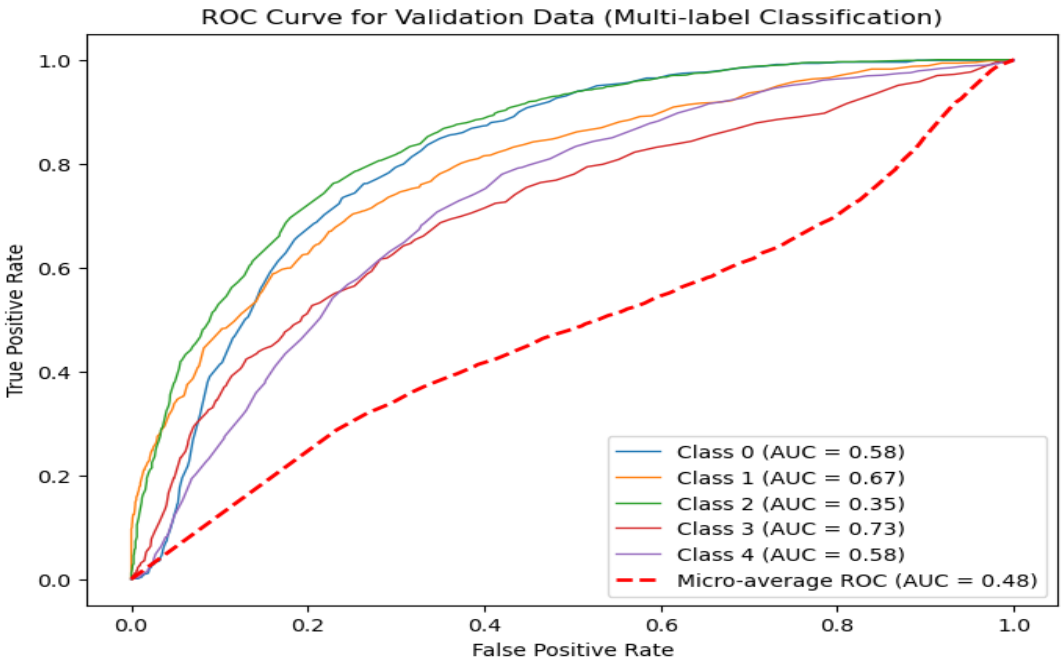


그림 14-2. 일반적인 CNN 모델의 양자화 후 Validation data에 대한 ROC Curve

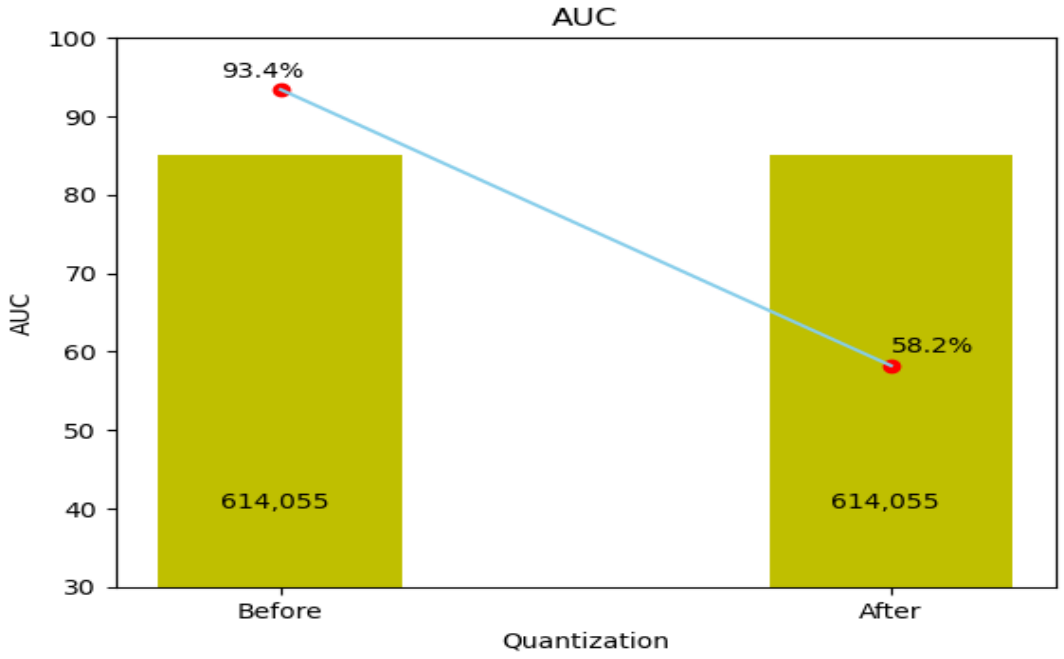


그림14-3. 양자화를 적용하기 전과 후의 파라미터 수와 AUC를 나타내는 그림

```
Flops of Original Model: 606493500.00
Flops of Quantized Model: 606492000.00
```

그림14-4. 양자화를 적용하기 전과 후의 Flops 수 비교

일반적인 CNN 모델의 양자화 전 Validation set에 대한 AUC의 평균은 93.4%였지만 양자화 후 Validation set에 대한 AUC의 평균은 58.2%로 정확도가 매우 낮아졌다. 그리고 양자화 전과 후의 모델에 대한 크기나 연산에 필요한 flops 수의 유의미한 변화를 찾지 못했다.

4.3. 지식 증류**(Knowledge Distillation)**

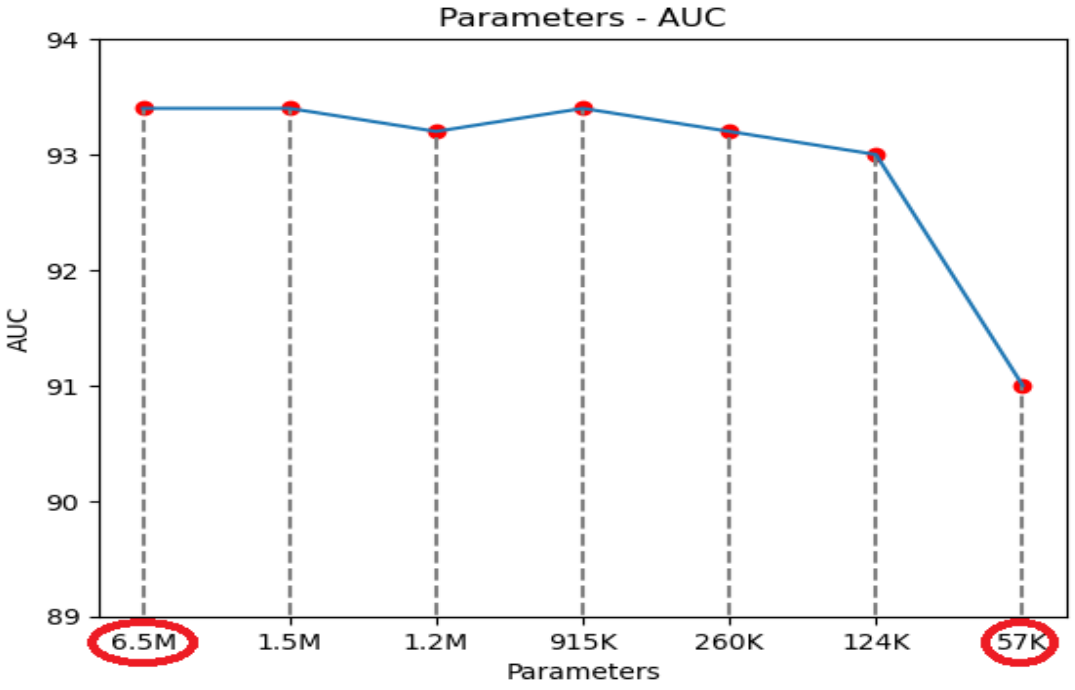


그림 15. 파라미터 수에 따른 AUC 값을 나타낸 그래프

위 그래프를 바탕으로 일반적인 CNN 모델 중에서 파라미터의 수가 약 650만 개인 모델을 Teacher 모델로, 파라미터 수가 약 53만 개인 모델을 Student 모델로 설정하고 지식 증류를 이용하여 Student 모델을 학습시켰다.

Teacher 모델

Layer (type)	Output Shape	Param #
Conv1d-1	[-1, 100, 991]	12,100
BatchNorm1d-2	[-1, 100, 991]	200
ReLU-3	[-1, 100, 991]	0
MaxPool1d-4	[-1, 100, 495]	0
Dropout-5	[-1, 100, 495]	0
Conv1d-6	[-1, 250, 486]	250,250
BatchNorm1d-7	[-1, 250, 486]	500
ReLU-8	[-1, 250, 486]	0
MaxPool1d-9	[-1, 250, 243]	0
Dropout-10	[-1, 250, 243]	0
Conv1d-11	[-1, 500, 234]	1,250,500
BatchNorm1d-12	[-1, 500, 234]	1,000
ReLU-13	[-1, 500, 234]	0
MaxPool1d-14	[-1, 500, 117]	0
Dropout-15	[-1, 500, 117]	0
Conv1d-16	[-1, 1000, 108]	5,001,000
BatchNorm1d-17	[-1, 1000, 108]	2,000
ReLU-18	[-1, 1000, 108]	0
MaxPool1d-19	[-1, 1000, 54]	0
AdaptiveAvgPool1d-20	[-1, 1000, 1]	0
Dropout-21	[-1, 1000]	0
Linear-22	[-1, 5]	5,005
Sigmoid-23	[-1, 5]	0

Total params: 6,522,555
Trainable params: 6,522,555
Non-trainable params: 0

Input size (MB): 0.05
Forward/backward pass size (MB): 13.20
Params size (MB): 24.88
Estimated Total Size (MB): 38.13

그림 16-1. 파라미터 수가 약 650만개인 Teacher 모델의 구조

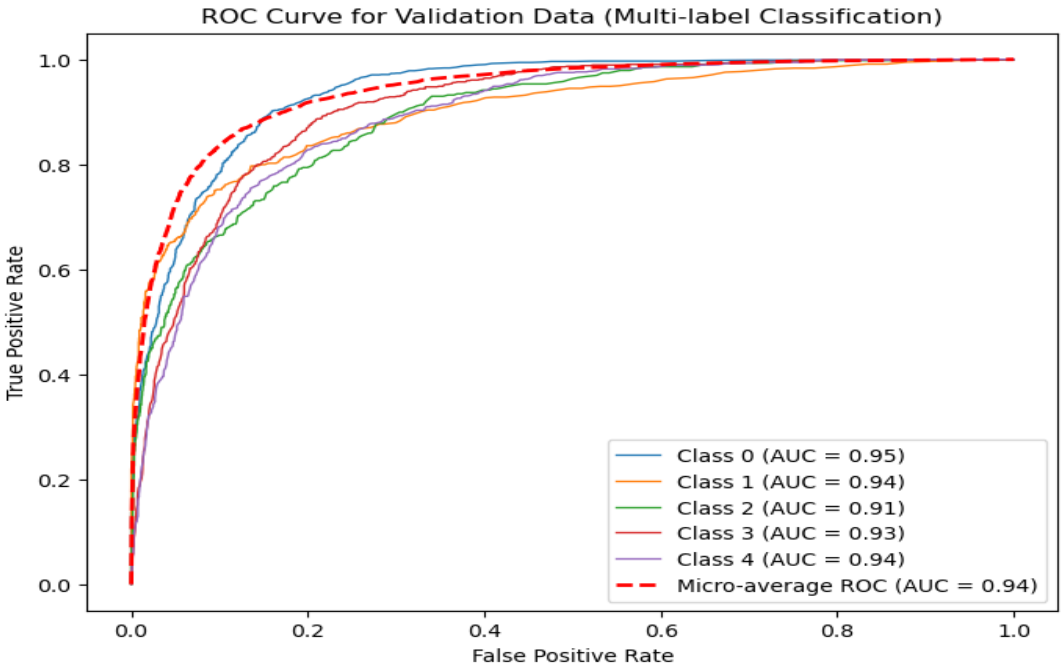


그림 16-2. 파라미터 수가 약 650만개인 Teacher 모델의 Validation data에 대한 AUC 그래프

Teacher 모델의 Estimated Total Size는 약 38MB이고, Validation AUC 평균은 93.4%이다.

Student 모델

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv1d-1              [-1, 50, 991]           6,050
       BatchNorm1d-2              [-1, 50, 991]             100
              ReLU-3              [-1, 50, 991]               0
         MaxPool1d-4              [-1, 50, 495]               0
           Dropout-5              [-1, 50, 495]               0
            Conv1d-6             [-1, 150, 486]          75,150
       BatchNorm1d-7             [-1, 150, 486]             300
              ReLU-8             [-1, 150, 486]               0
         MaxPool1d-9             [-1, 150, 243]               0
          Dropout-10             [-1, 150, 243]               0
           Conv1d-11             [-1, 300, 234]         450,300
      BatchNorm1d-12             [-1, 300, 234]             600
             ReLU-13             [-1, 300, 234]               0
        MaxPool1d-14             [-1, 300, 117]               0
AdaptiveAvgPool1d-15               [-1, 300, 1]               0
          Dropout-16                  [-1, 300]               0
           Linear-17                    [-1, 5]           1,505
          Sigmoid-18                    [-1, 5]               0
================================================================
Total params: 534,005
Trainable params: 534,005
Non-trainable params: 0
----------------------------------------------------------------
Input size (MB): 0.05
Forward/backward pass size (MB): 5.62
Params size (MB): 2.04
Estimated Total Size (MB): 7.70
----------------------------------------------------------------
```

그림 17-1. 파라미터 수가 약 53만개인 Student 모델의 구조

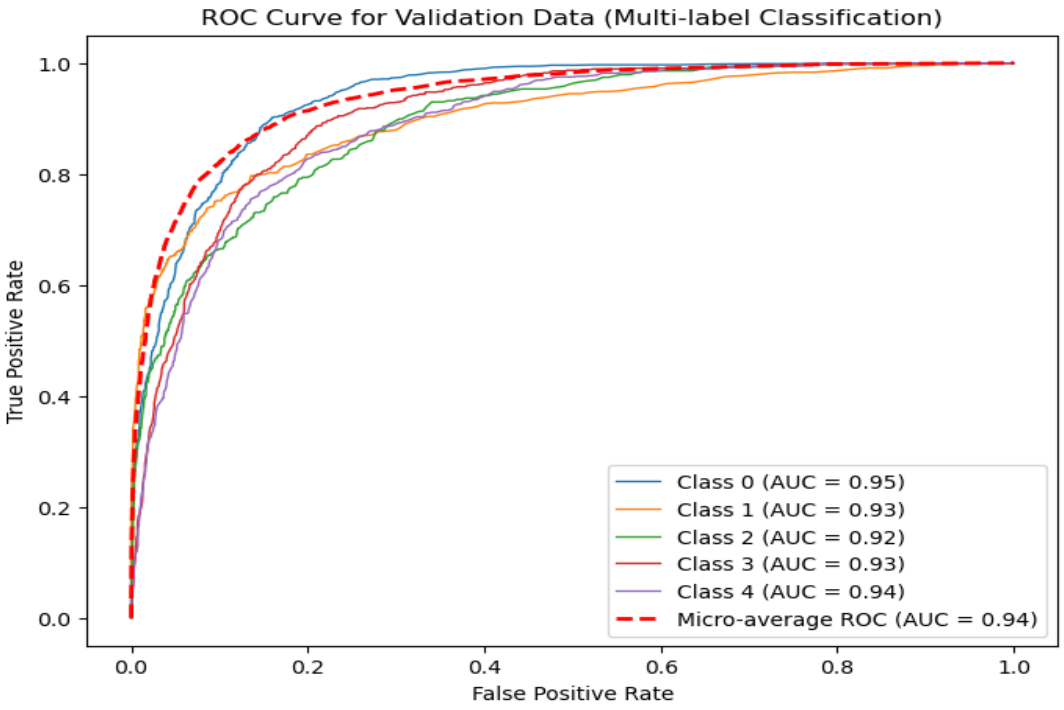


그림 17-2. 파라미터 수가 약 53만개인 Student 모델의 Validation data에 대한 AUC 그래프

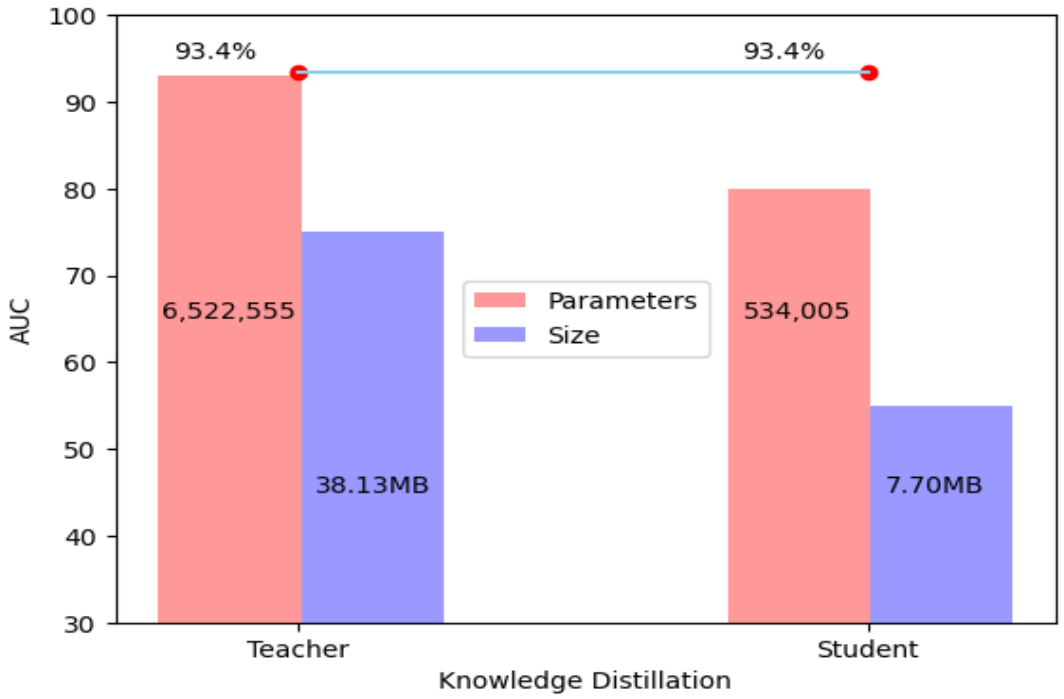


그림 18. 지식 증류의 Teacher 모델과 Student 모델의 비교

Student 모델의 Estimated Total Size는 약 8MB이고, Validation data에 대한 AUC 평균은 93.4%이다. 파라미터 수는 약 650만 개(Teacher)에서 약 53만 개(student)로 600만 개가량 줄었지만, Teacher 모델과 동일한 성능을 보여준다.

Teacher 모델의 지식 증류없이 학습한 모델과 **Student** 모델의 비교

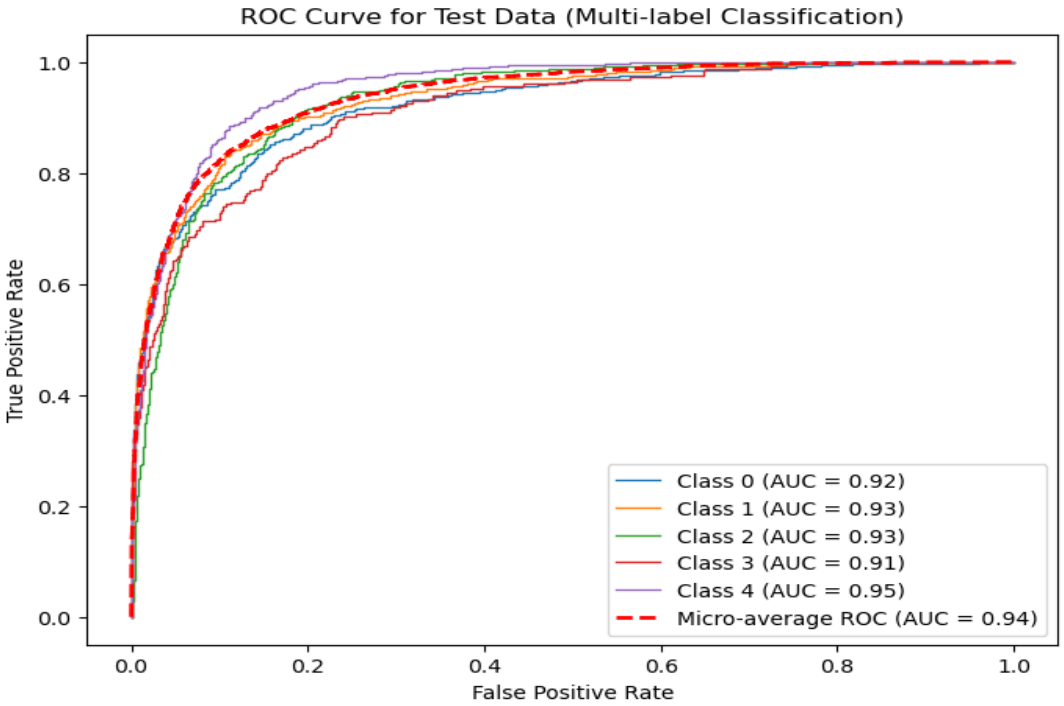


그림 19-1. Teacher 모델의 지식 증류 없이 학습한 Student 모델의 Test AUC

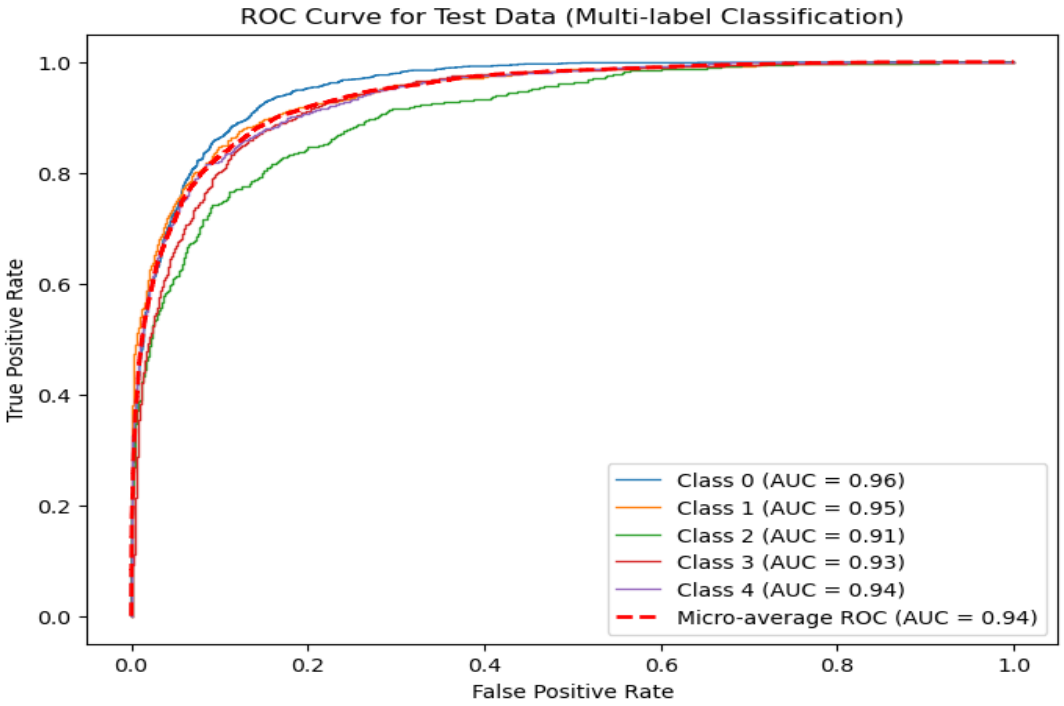


그림 19-2. Teacher 모델의 지식 증류를 통해 학습한 Student 모델의 Test AUC

파라미터의 수가 약 53만 개인 동일한 모델을 지식 증류 없이 학습한 모델과 지식 증류를 통해 학습한 모델의 Test data에 대한 AUC 그래프이다. 지식 증류 없이 혼자 학습한 모델의 AUC 평균은 92.8%이고, 지식 증류를 통해 학습한 모델의 AUC 평균은 93.8%이다.

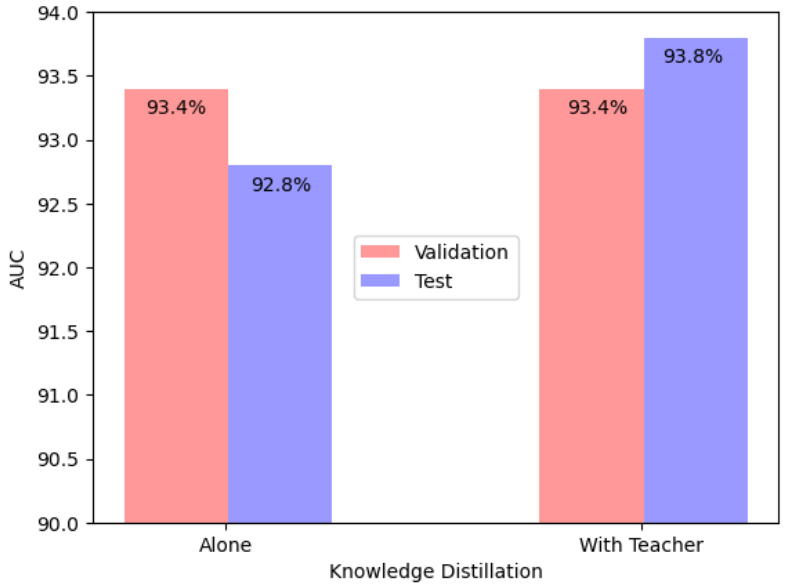


그림 19-3. 지식 증류 없이 학습한 Student 모델과 지식 증류로 학습한 Student 모델의 비교

지식 증류 없이 학습한 Student 모델과 지식 증류로 학습한 Student 모델의 Validation data에 대한 AUC는 93.4%로 동일하지만, Test data에 대한 기존 모델의 AUC가 지식 증류를 통하여 92.8%에서 93.8%로 1% 정도 향상되었다.

따라서 지식 증류를 통해 학습한 Student 모델은 Teacher 모델보다 파라미터 수와 크기가 줄었지만, Teacher 모델과 비슷한 성능을 가지고, 지식 증류 없이 혼자 학습한 동일한 모델에 보다 Test data에 대한 AUC가 1% 향상되었음을 알 수 있다.

5. 결론 및 향후 연구 방향

5.1. 결론

PTB-XL 데이터 세트를 이용하여 ECG 데이터를 처리하는 1차원 CNN 모델을 학습하고 가지치기(Pruning), 양자화(Quantization), 지식 증류(Knowledge Distillation)를 적용하여 경량화를 진행했다. 가지치기와 양자화를 적용한 모델은 flops 수와 크기에 대해서 경량화와 관련된 유의미한 변화를 찾지 못했지만, 지식 증류를 통하여 파라미터 수와 크기를 줄이고 최대한 높은 성능을 가지는 모델을 학습시켰다.

5.2 향후 연구 방향

현재 스마트 워치의 ECG 분석 기능은 single-lead ECG를 기준으로 판단한다. 본 과제의 12-lead ECG는 스마트 워치로 분석할 수 없다. 스마트 워치에서 분석할 수 있는 single-lead ECG에 대한 모델을 학습시키고 경량화하는 것이 좋을 것 같다.

데이터 차원의 부족으로 인해 single-lead ECG 신호에서 신경망에 의해 추출된 질병 정보는 12-lead ECG 신호에 비해 적지만 여전히 부분적인 질병 정보를 포함하고 있고 이 부분적인 질병 정보를 single-lead ECG와 12-lead ECG 간의 상호 정보라고 한다. 또 다른 측면에서, single-lead ECG에 대해 사용할 수 있는 데이터가 없으면 잘못된 진단을 유발할 수도 있다.

12-lead ECG 분석 모델의 정보를 single-lead ECG 분석 모델로 지식 증류를 하는 방법으로 single-lead ECG와 12-lead ECG 간의 상호 정보를 포함한 single-lead ECG 분석 모델을 학습할 수 있다. 그 방법으로 MKD(Multi-label Knowledge Distillation)를 사용하는 방법이 있다.[11]

MKD를 이용하여 12 lead에서 얻은 정보를 단일 리드 ECG로 전이하는 방법을 연구하고, 모델의 가중치 및 특성 추출 방법을 최적화하여 가벼운 모델을 구축할 수 있을 것이다.

[11] https://www.clinicalkey.com/#!/content/playContent/1-s2.0-S001048252300968X?scrollTo=%23fig3

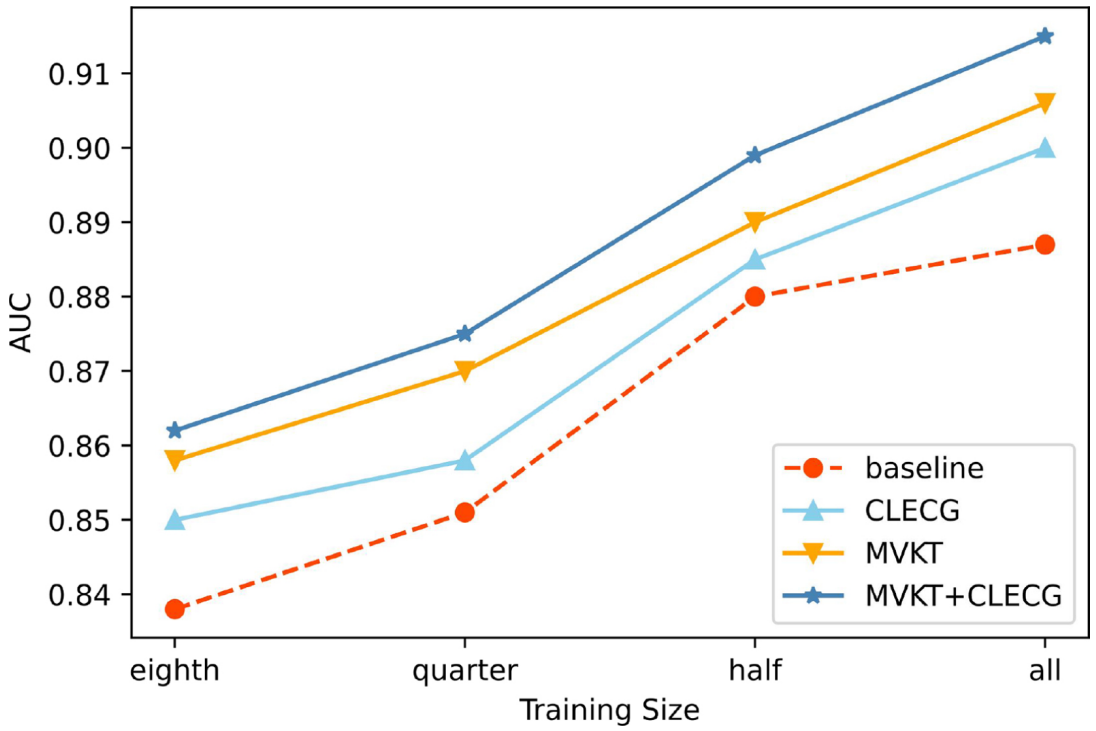


그림 20. MKD를 적용한 MVKT-ECG의 AUC

MKD를 적용한 경량화된 single-lead ECG 분석 모델을 스마트 워치에서 사용한다면 기존의 baseline 모델의 성능이 향상될 것이다.

6. 참고 문헌

[1] Shin Hansu, Roh Youngmin, & Seo Jiwoo. The Relationship between Accessibility to Local Cardio-Cerebrovascular Disease Centers and Mortality Ratio. Health and Social Welfare Review, 43(1), 85-100. (2023)

[2] Yuzhen Qin, Li Sun, Hui Chen, Wei-qiang Zhang, Wenming Yang, Jintao Fei, Guijin Wang. MVKT-ECG: Efficient Single-lead ECG Classification on Multi-Label Arrhythmia by Multi-View Knowledge Transferring (2023)

[3] Rawi, Atiaf A., Elbashir, Murtada K. and Ahmed, Awadallah M.. "Deep learning models for multilabel ECG abnormalities classification: A comparative study using TPE optimization" Journal of Intelligent Systems, vol. 32, no. 1, pp. (2023)

[4] Tao, R.; Wang, L.; Wu, B. “A resource-efficient ECG diagnosis model for mobile health devices”. Information Sciences, 648p (2023)

[5] Wagner, Patrick, et al. "PTB-XL, a large publicly available electrocardiography dataset" (version 1.0.3). PhysioNet (2022)

[6] D. Li, J. Zhang, Q. Zhang and X. Wei, "Classification of ECG signals based on 1D convolution neural network," 2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom), Dalian, China, pp. 1-6, (2017)

[7] Wordbe, [Classification] Cross entropy의 이해, 사용 방법(Categorical, Binary, Focal loss), https://wordbe.tistory.com/46 (2019.08.01.)

[8] Github. Available : https://github.com/trent-b/iterative-stratification

[9] CrumpledBenito, Visual representation of Pruning to see the difference before and after Pruning, https://commons.wikimedia.org/wiki/File:Before_after_pruning.png, (2020.08.25)

[10] Vijendra, Knowledge Distillation Flow Chart, https://www.analyticsvidhya.com/blog/2022/01/knowledge-distillation-theory-and-end-to-end-case-study/, (2022.01.04)

[11] Pytorch. Available : https://pytorch.org/tutorials/intermediate/pruning_tutorial.html

[12] Yuzhen Qin, Li Sun, Hui Chen, Wei-qiang Zhang, Wenming Yang, Jintao Fei, Guijin Wang. MVKT-ECG: Efficient Single-lead ECG Classification on Multi-Label Arrhythmia by Multi-View Knowledge Transferring (2023)

2023 년 전기 산학협력 프로젝트 멘토 의견서

1. 지도개요

팀 명	풀악셀		
과 제 명	심장질환 환자 ECG 데이터 분석 모델 개발		
협력기관	울산테크노파크		
참여학생	이름	전화번호	이메일
	신다윗	010-9217-4004	sin002277@gmail.com
	안주현	010-6800-3948	ajmok8703@naver.com
	이재현	010-5170-9970	jaeback2435@naver.com
참여교수명	송길태		

2. 세부 지도 내용

프로젝트 검토의견, 개선사항, 발전 방향 제안, 프로젝트 수행에 참고할 자료 등을 포함해 자유롭게 작성해주세요.

- 심전도는 심장운동으로 미세하게 변하는 심장의 전위차를 신체외부의 피부에서 측정한 것으로 최근 의료, 금융, 보안, 오락 등 서비스에서 기존의 생체신호시스템의 대안으로 많은 연구가 되고 있어 적합한 실험 주제로 판단됨
- 위 프로젝트에서 적용된 분석모델은 단수개의 모델을 기반으로 연구를 진행하였음 (Inception module 기반 CNN 모델) * inception module 참고자료 (https://gist.github.com/joelouismarino/a2ede9ab3928f999575423b9887abd14)
- 해당 분석 모델의 정확도는 Training 데이터의 88% 으로, 좀 더 정확한 모델을 설계할 필요성이 있음
- 실제, CNN 을 이용하여 심전도의 깊은 특징을 추출한 실험 사례 참고 (중국과학원, 전문가 특징과 신경망의 특징을 결합한 후 여러 개의 독립적인 XGBoost 분류기들을 학습시켜 예측결과를 평균하여 융합하였음)
- 딥러닝을 이용한 심전도 특징추출과 웨이블릿 분석과 딥러닝의 조화를 통해 분류성능을 더 높이려는 국내외 연구 동향에 발맞추어 의미 있는 실험으로 보임

위 내용을 부산대학교 정보컴퓨터공학부 2023 학년도 전기 산학협력프로젝트 지도내용으로 제출합니다.

멘토링 일시	2023 년 9 월 22 일	시작시간	19:00	종료시간	21:00

소속: 울산테크노파크 **직급**: 전문위원 **성명**: 박승남 (서명)