
SW 중심대학 디지털 경진대회

AI 부문



DMS 팀

박진성, 김남규, 김성연, 윤은옥, 조수아



창원대학교

Contents

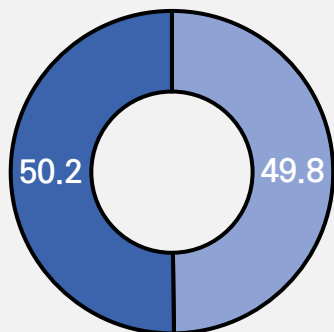
1. 데이터 분석
2. 알고리즘
3. 모델 구축 및 검증
4. 기대효과 및 적용 가능성
5. 참고 문헌 및 출처

1. 데이터 분석

train

- 55,438개의 32kHz로 샘플링 된 오디오 샘플
- 방음 환경에서 녹음된 Real 목소리 샘플
- 방음 환경을 가정한 Fake 목소리 샘플
- 샘플 당 한 명의 Real / Fake 목소리 존재
- Real / Fake Label 제공

Train Data Set



□ Real ■ Fake

test

- 50,000개의 32kHz로 샘플링 된 평가용 오디오 샘플 (5초 분량)
- 방음 환경 / 방음 환경이 아닌 환경 모두 존재
- 각 샘플 당 최대 2명의 목소리 존재

Ex)

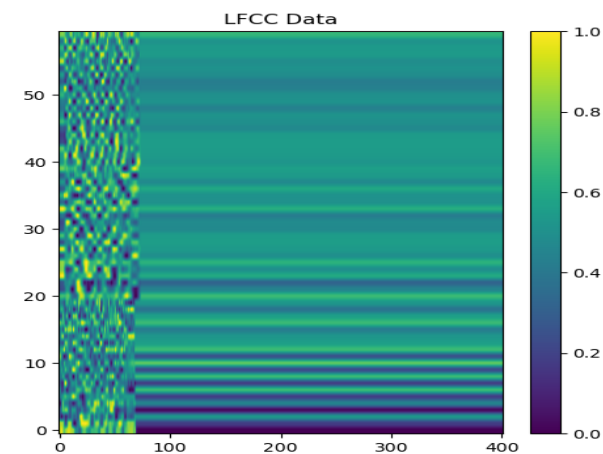
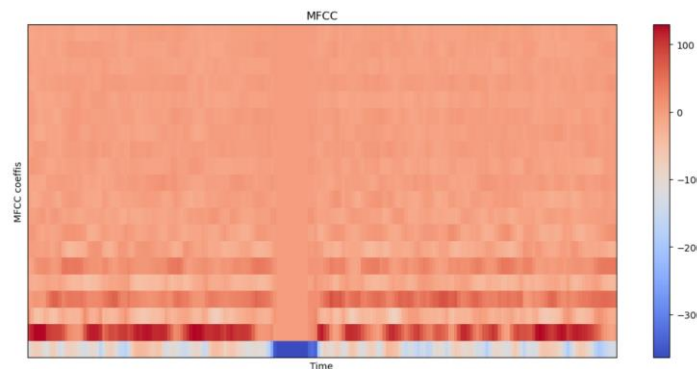
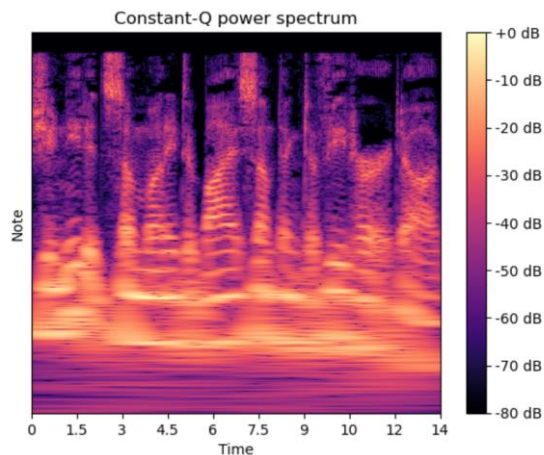
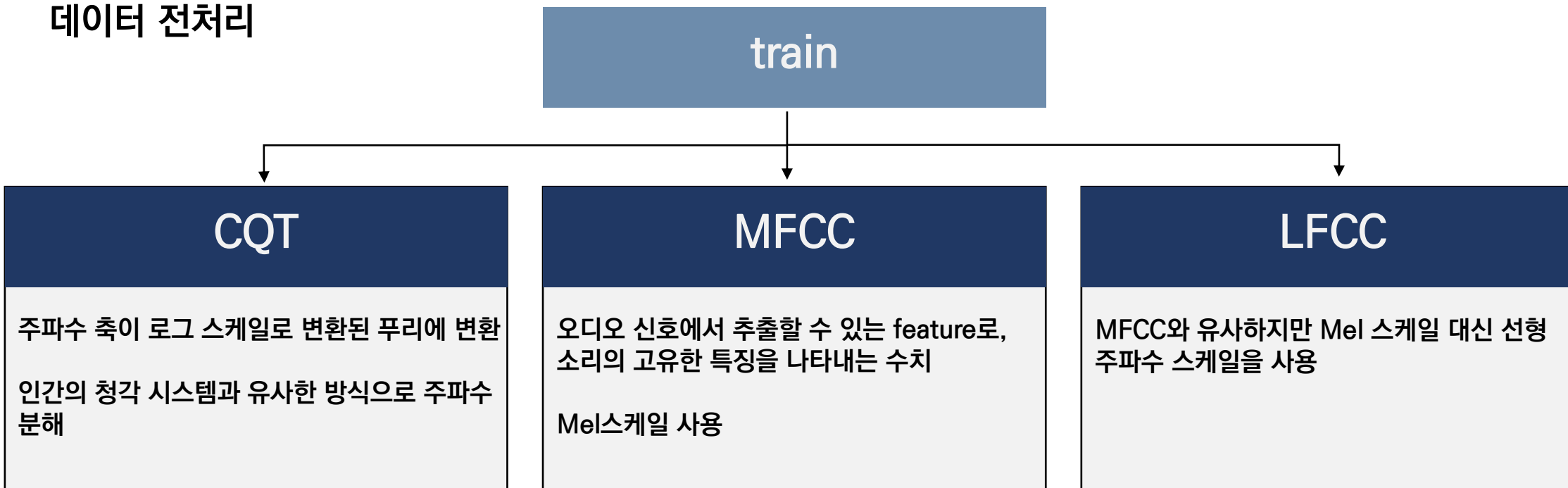
- 1) 1명의 진짜 목소리만 존재
- 2) 1명의 가짜 목소리만 존재
- 3) 1명의 진짜 목소리와 1명의 가짜 목소리가 존재
- 4) 2명의 진짜 목소리가 존재
- 5) 2명의 가짜 목소리가 존재
- 6) 아예 목소리가 없는 경우

unlabeled_data

- 1,264개의 32kHz로 샘플링 된 오디오 샘플 (5초 분량)
- test 오디오 샘플과 동일한 환경이나 Label은 없음

1. 데이터 분석

데이터 전처리



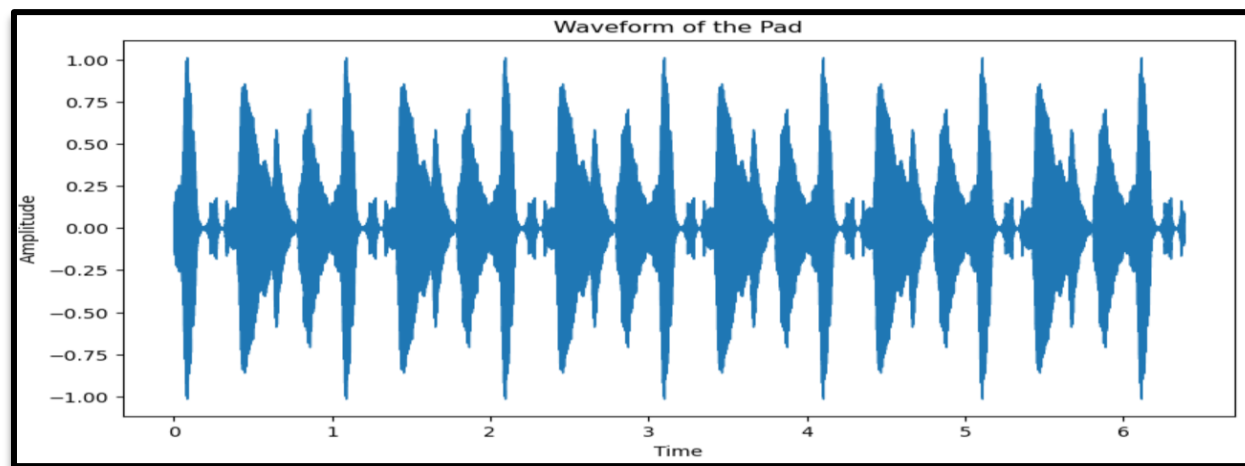
1. 데이터 분석

데이터 전처리

train



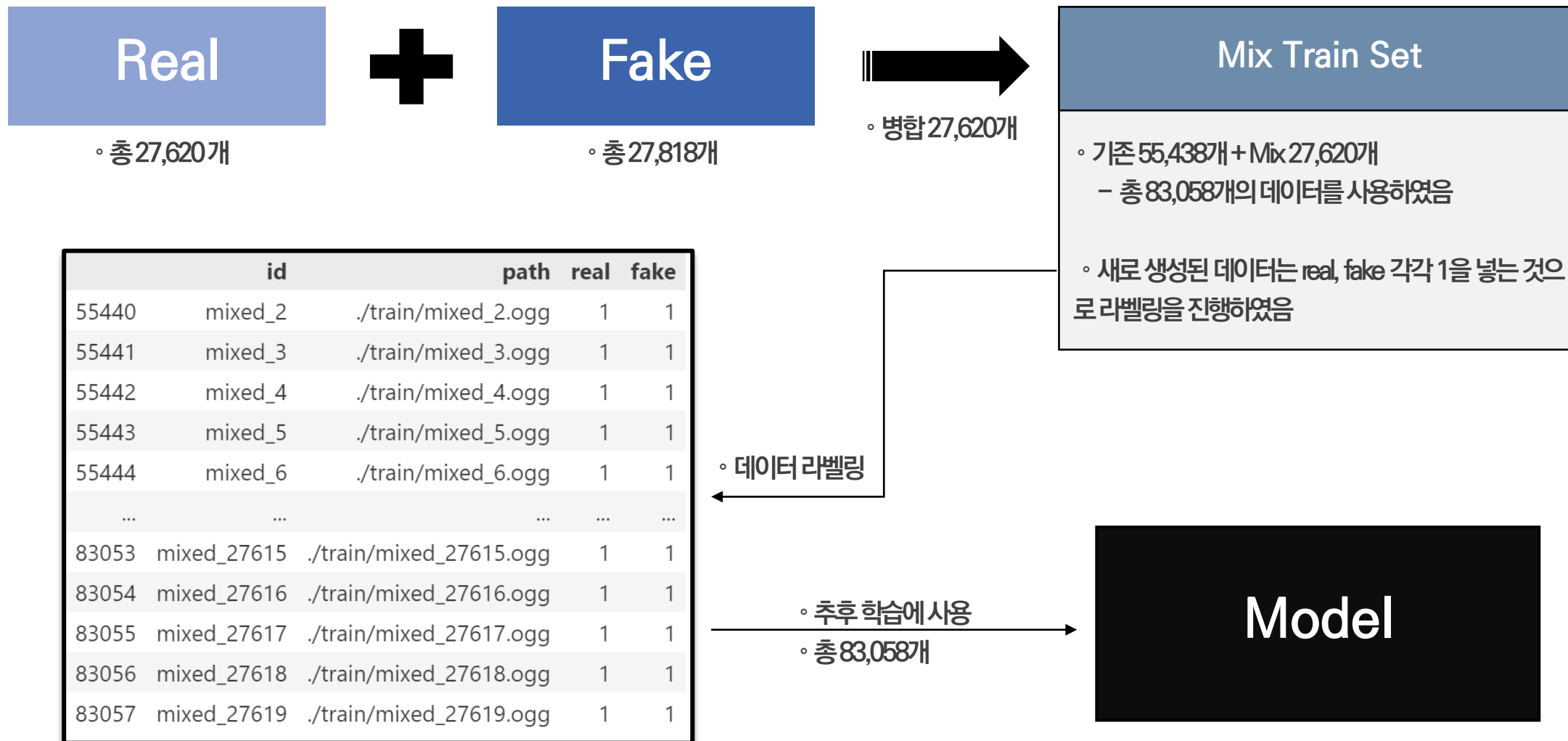
- 1~38초 가량의 다양한 길이의 오디오 샘플들로 구성
- 학습을 위해 6.4초 단위로 잘라 전처리
- 6.4초 미만의 오디오들은 반복을 통해 6.4초로 길이를 늘림



- 전처리 결과물 (기존 1초 샘플)

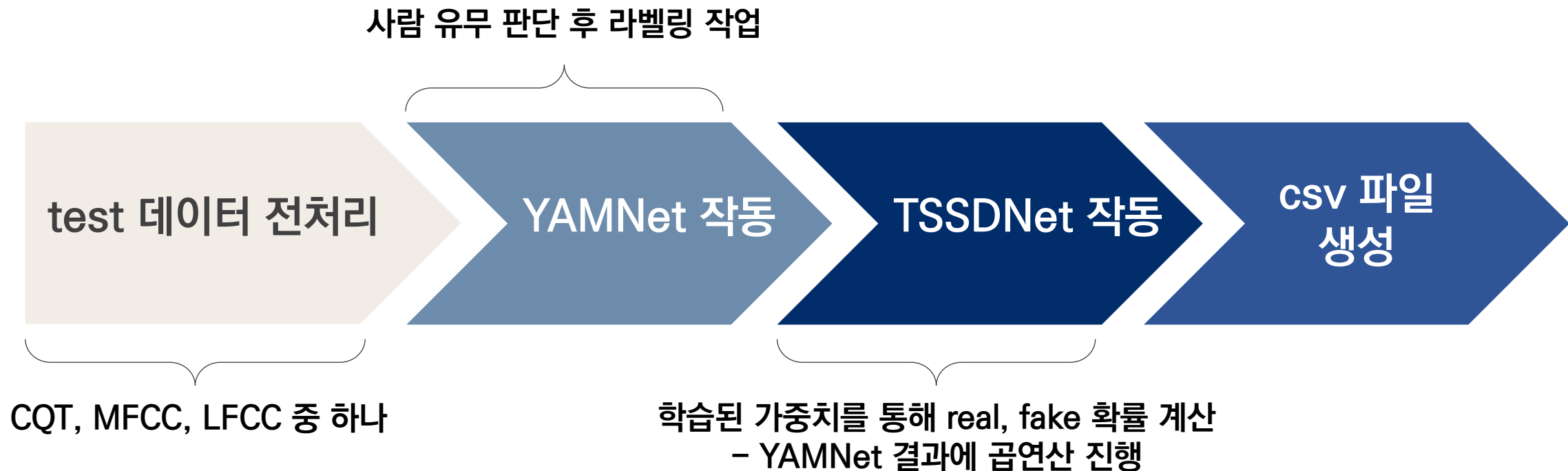
1. 데이터 분석

데이터 추가 제작



2. 알고리즘

모델 작동 순서



3. 모델 구축 및 검증



모델 선택

Towards End-to-End Synthetic Speech Detection

Publisher: IEEE

Cite This


PDF

Guang Hua  ; Andrew Beng Jin Teoh  ; Haijian Zhang  [All Authors](#)

Published in: IEEE Signal Processing Letters (Volume: 28)

Page(s): 1265 - 1269

DOI: 10.1109/LSP.2021.3089437


Date of Publication: 15 June 2021 

Publisher: IEEE

※TSSDNet모델 관련 논문, 16p5-3



TensorFlow > 학습 > TensorFlow Core

도움이 되었나요?  

환경 소리 분류를 위한 YAMNet을 사용한 전이 학습 



Google Colab에서 실행



GitHub에서 보기



노트북 다운로드



TF 허브 모델 보기

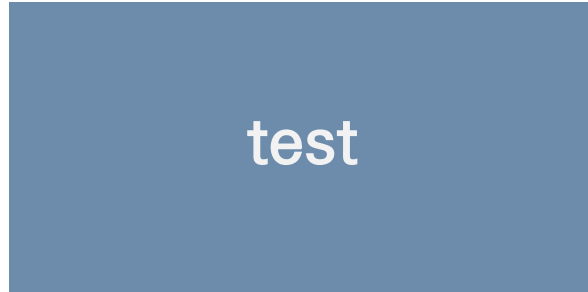
YAMNet은 웃음, 짓음 또는 사이렌과 같은 521개 클래스의 오디오 이벤트를 예측할 수 있는 사전 훈련된 심층 신경망입니다.

※YAMNet모델 관련 페이지, 16p5-4

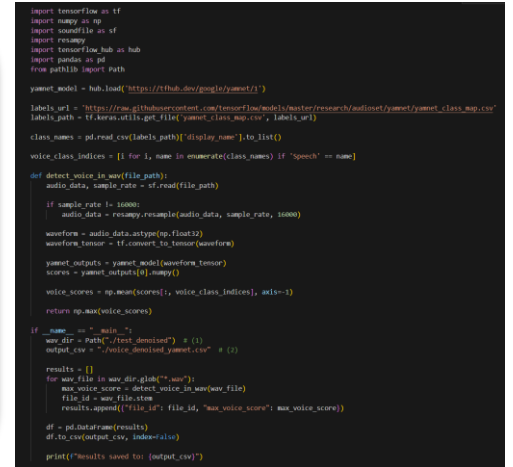
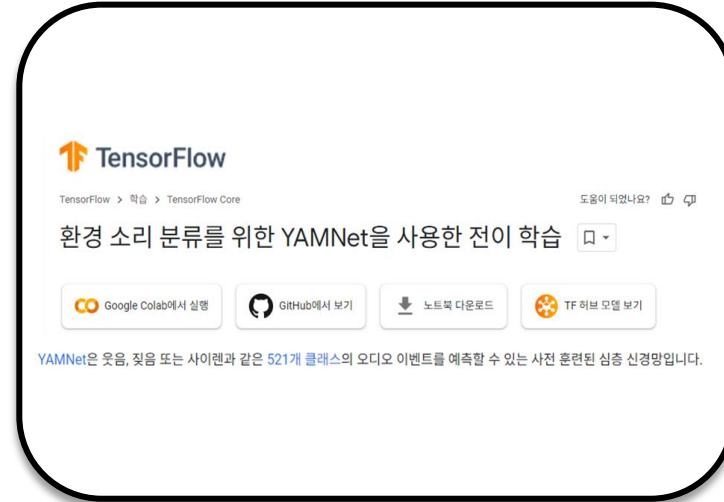
- 총 2개의 모델로, 해당 논문에서 게재된 개선된 TSSDNet (Time-domain Synthetic Speech Detection Net) 모델을 기반으로 수정한 모델과 TensorFlow에서 제공하는 YAMNet 모델을 사용하였음
- 논문에 포함된 git 링크를 통해 TSSDNet 모델을 사용할 수 있음
 - <https://github.com/ghua-ac/end-to-end-synthetic-speech-detection>

3. 모델 구축 및 검증

YAMNet



test 음성 사람 유무 판단



- test 음성의 상황 중 아예 목소리가 없는 경우가 존재
- 이를 구별하기 위해 YAMNet 모델을 사용
- Speech(사람 목소리)인 확률을 score로 계산하여 라벨링

```
yamnet_model = hub.load('https://tfhub.dev/google/yamnet/1')

labels_url = 'https://raw.githubusercontent.com/tensorflow/models/master/research/audioset/yamnet/yamnet_class_map.csv'
labels_path = tf.keras.utils.get_file('yamnet_class_map.csv', labels_url)

class_names = pd.read_csv(labels_path)['display_name'].to_list()

voice_class_indices = [i for i, name in enumerate(class_names) if 'Speech' == name]
```

◦ Speech인 경우를 지정하는 코드

	id	max_voice_score
0	TEST_47315	0.999834
1	TEST_44051	0.999922
2	TEST_43880	0.996847
3	TEST_22088	0.999973
4	TEST_13543	0.948049
...
49995	TEST_39122	0.000025
49996	TEST_06183	0.999711
49997	TEST_42428	0.685903
49998	TEST_44637	0.999853
49999	TEST_20414	0.999977

데이터 라벨링

3. 모델 구축 및 검증

YAMNet

test

◦ 방음 환경/방음 환경이 아닌 환경 모두 존재

◦ 정확도를 높이기 위해 노이즈 제거 작업 진행

Noise removed

```
1 import soundfile as sf
2 import torch
3 from pathlib import Path
4 from df.enhance import enhance, init_df, load_audio, save_audio
5
6 # DeepFilterNet을 사용한 노이즈 제거 함수
7 def denoise_audio_from_ogg(input_path, output_path):
8     data, samplerate = sf.read(input_path)
9     data_tensor = torch.from_numpy(data.astype('float32')).unsqueeze(0)
10    model, df_state, _ = init_df()
11    enhanced_audio = enhance(model, df_state, data_tensor)
12    enhanced_audio = enhanced_audio.squeeze(0).numpy()
13    save_audio(output_path, enhanced_audio, samplerate)
14    print(f"Saved denoised audio to: {output_path}")
15
16 if __name__ == "__main__":
17     # 경로 설정
18     ogg_dir = Path("./test") # OGG 파일이 있는 디렉토리
19     output_dir = Path("./test_denoised") # 소음 제거된 파일을 저장할 디렉토리
20     output_dir.mkdir(exist_ok=True) # 출력 디렉토리가 없으면 생성
21
22     # 디렉토리 내의 모든 OGG 파일을 순회
23     for ogg_file in ogg_dir.glob("*.ogg"):
24         output_file = output_dir / (ogg_file.stem + "_denoised.wav")
25         denoise_audio_from_ogg(ogg_file, output_file)
26
```

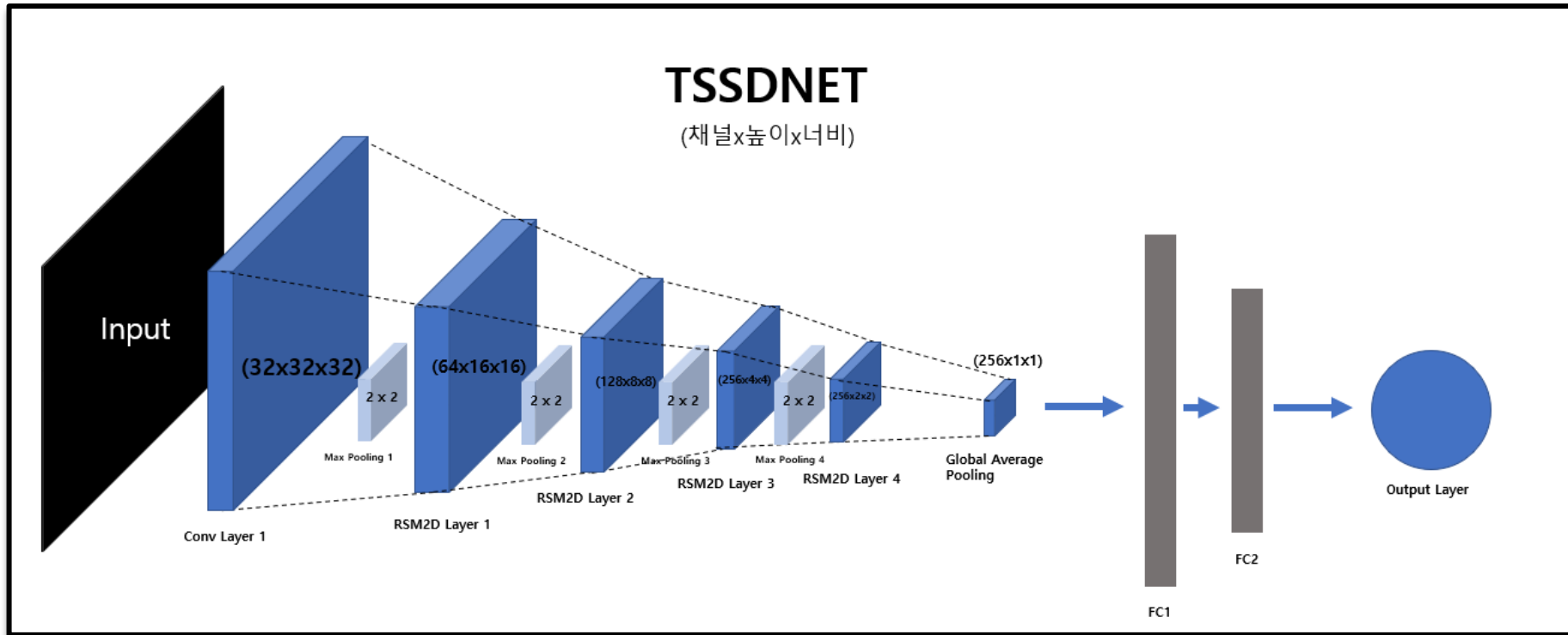
◦ df.enhance 라이브러리 사용

```
TEST_00000_denoised.npy
TEST_00001_denoised.npy
TEST_00002_denoised.npy
TEST_00003_denoised.npy
TEST_00004_denoised.npy
TEST_00005_denoised.npy
TEST_00006_denoised.npy
TEST_00007_denoised.npy
TEST_00008_denoised.npy
TEST_00009_denoised.npy
TEST_00010_denoised.npy
```

◦ 노이즈 제거 결과물

3. 모델 구축 및 검증

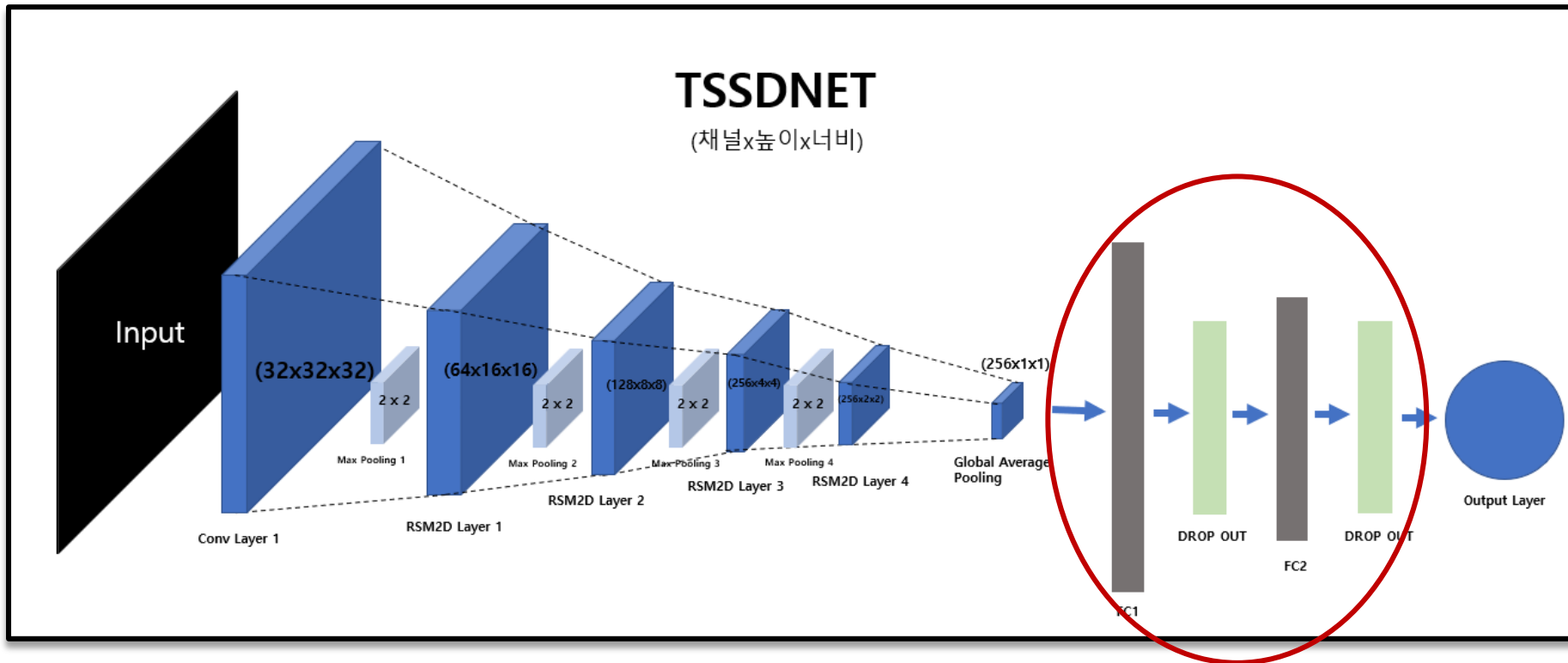
TSSDNet



- TSSDNet은 주로 합성곱 신경망 (CNN), 순환 신경망 (RNN)을 사용하여 음성 신호의 시간적 특성과 공간적 특성을 동시에 학습

3. 모델 구축 및 검증

수정된 TSSDNet

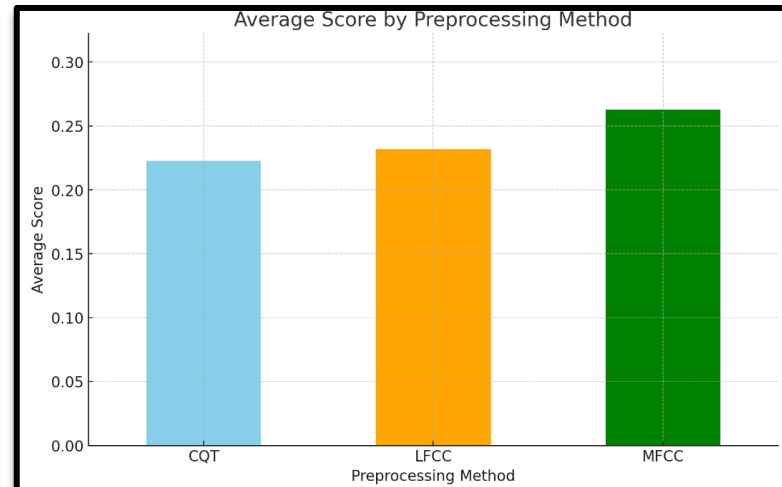
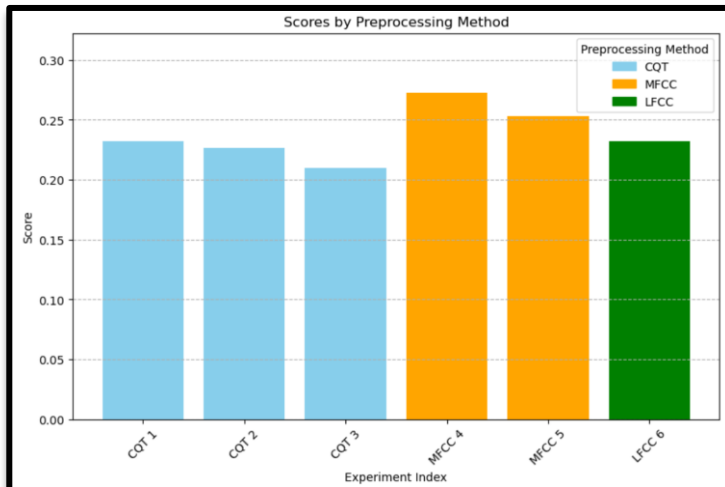


- 기존의 TSSDNet의 학습 과정에 Drop Out 과정을 추가하여 학습하도록 진행

3. 모델 구축 및 검증

하이퍼 파라미터 튜닝

전처리 방식	num_channel	batch_size	learning_rate	dropout_rate	score
CQT	2	16	4.00E-05	0.5	0.231921559
	8	32	3.00E-04	0	0.226711613
	32	16	1.25E-05	0.4	0.209840704
MFCC	2	16	4.00E-05	0.3	0.272444179
	8	16	4.00E-05	0.3	0.253182822
LFCC	32	16	4.00E-04	0.4	0.2318054091



- 전체적으로 MFCC, LFCC 방식보다 CQT 방식이 더 좋은 결과를 보임
- 가장 좋은 결과를 보여준 세 번째가 중치 값 사용

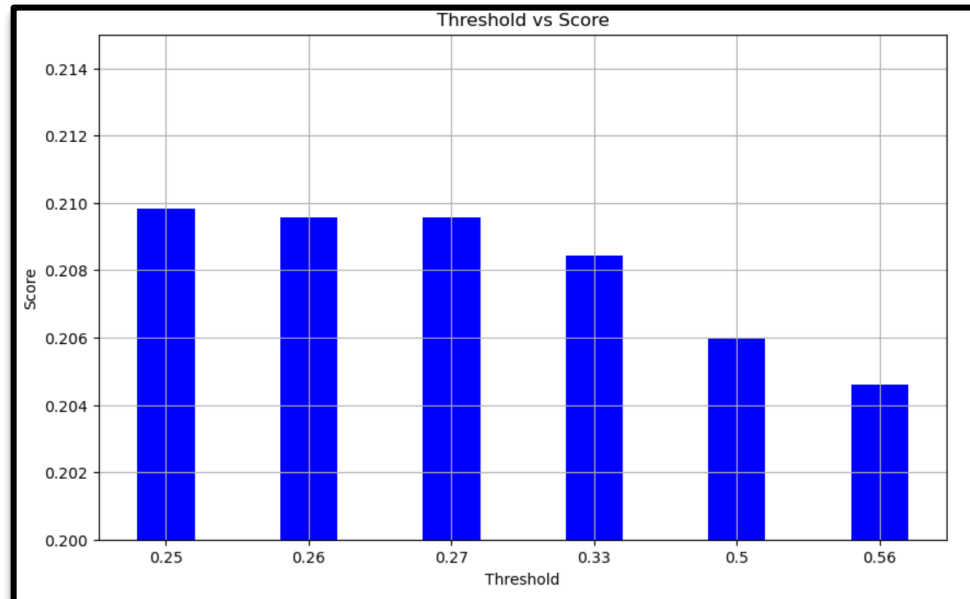
3. 모델 구축 및 검증

파라미터 튜닝

- 가장 좋은 결과를 만들어낸 가중치를 기준으로 YAMNet 파라미터 수정 후 비교

◦ threshold: 사람이라고 인식하는 기준 값

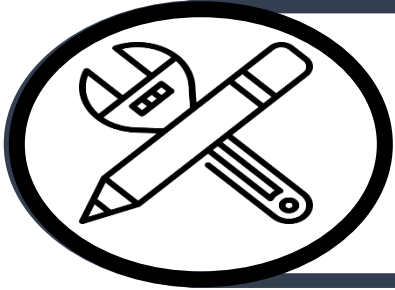
threshold	score
0.25	0.209840704
0.26	0.209583706
0.27	0.2095624849
0.33	0.208444557
0.50	0.2059961945
0.56	0.2046123819



- 사용된 가중치 하이퍼 파라미터

전처리 방식	num_channel	batch_size	learning_rate	dropout_rate	score
CQT	32	16	1.25E-05	0.4	0.209840704

4. 기대효과 및 적용 가능성



- 확장성 및 커스터마이징

특정 목적성에 맞게 특정한 데이터에 맞춰 모델을 커스터마이징하고 추가 학습을 통해 성능을 향상시킬 수 있음



- 딥페이크 영상 분석 가능성

영상 내의 음성을 통해 영상을 분석하여 진위 여부를 판별할 수 있는 모델로 가능성이 있음



- 음성 인증 시스템 보호

금융기관에 사용하는 인증시스템으로 부터 안전성을 보장할 수 있으며 사기와 신원도용 방지



- 법적 증거

음성 증거의 진위 여부를 판별하여, 조작된 증거로 인한 오판 방지 및 공정한 재판 지원 가능

5. 참고 문헌 및 출처

1. SW중심대학 디지털 경진대회_SW와 생성AI의 만남 : AI 부문 데이터
<https://dacon.io/competitions/official/236253/data>
2. test 데이터 상세 질문
<https://dacon.io/competitions/official/236253/talkboard/412040?page=1&dtype=recent>
3. Towards End-to-End Synthetic Speech Detection
G. Hua, A. Teoh, and H. Zhang, "Towards End-to-End Synthetic Speech Detection," IEEE Signal Processing Letters, vol. 28, pp. 1265–1269, 2021, doi: 10.1109/LSP.2021.3089437
4. 환경 소리 분류를 위한 YAMNet을 사용한 전이 학습
https://www.tensorflow.org/tutorials/audio/transfer_learning_audio?hl=ko

Q&A

감사합니다