

Heart Beat

Signal object detection

INDEX

☐ 프로젝트 주제

☐ 데이터셋 선정

☐ 프로젝트 수행 과정

☐ 성능 결과 및 시각화

☐ 후속 연구

☐ Q&A



심음의 신호 내에서 s1 , s2 Detection

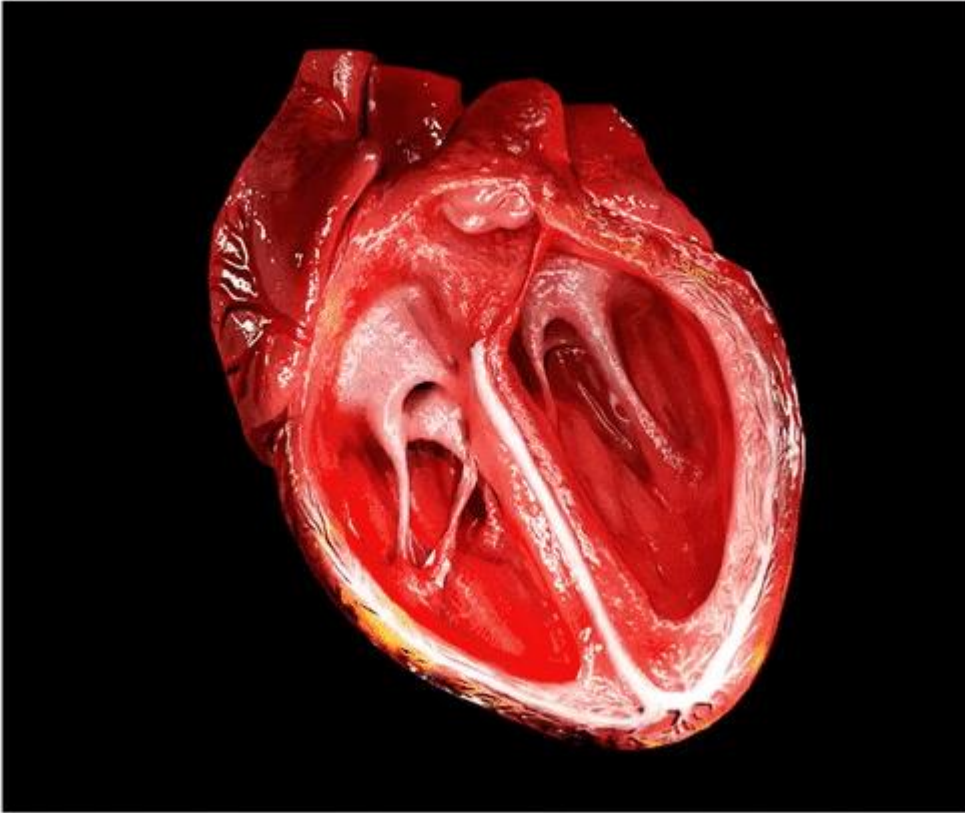


Background

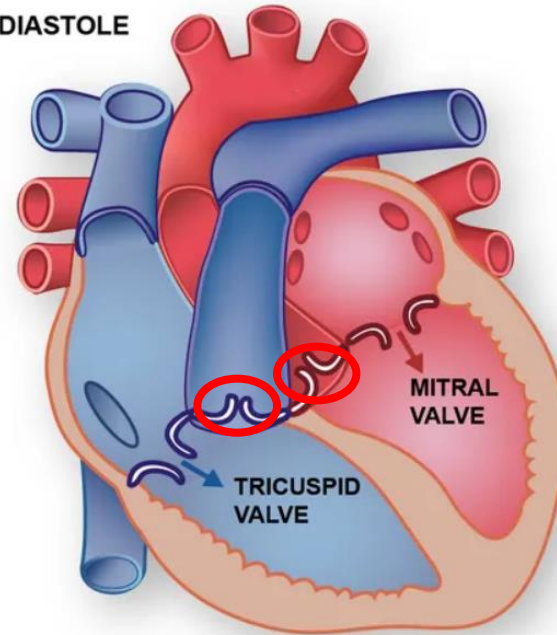
What is Heart Sound?



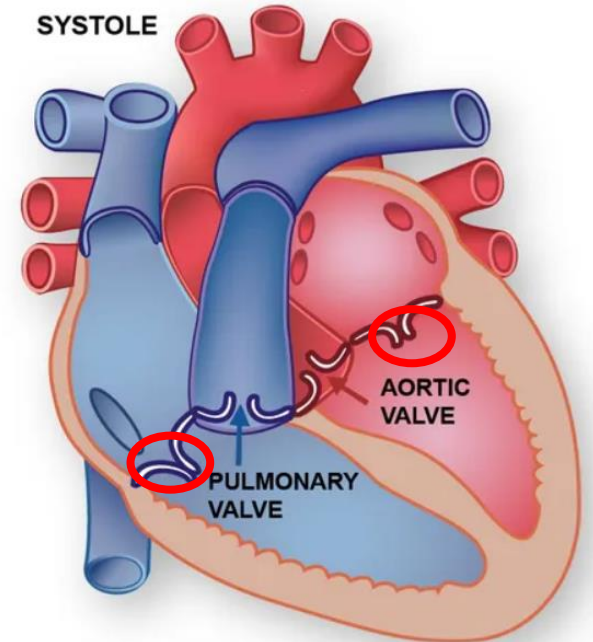
CARDIAC CYCLE



DIASTOLE



SYSTOLE



2

데이터셋 선정

Cicor-heart-sound dataset



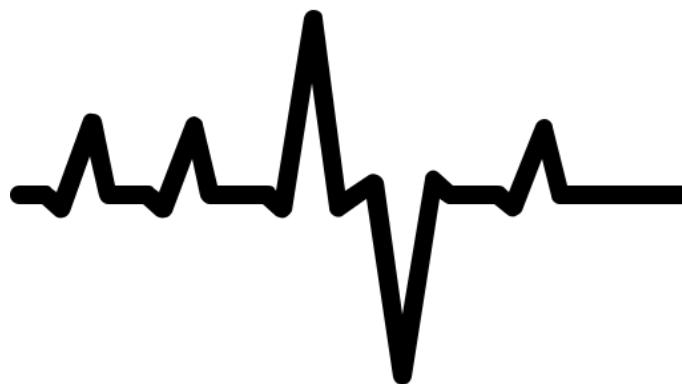
✓ Physionet의 cicor-heart-sound dataset

✓ Data 저장 경로

training_data

- PatientNum.txt
- PatientNum_local.wav
- PatientNum_local.tsv
- PatientNum_local.hef

.wav file



Sample rate = 4000

.tsv file

S	E	class
0.00	5.31	0
5.31	5.42	1
5.42	5.88	2
	•	
	•	
	•	
22.31	24.42	2
24.42	24.88	3
24.88	28.57	0

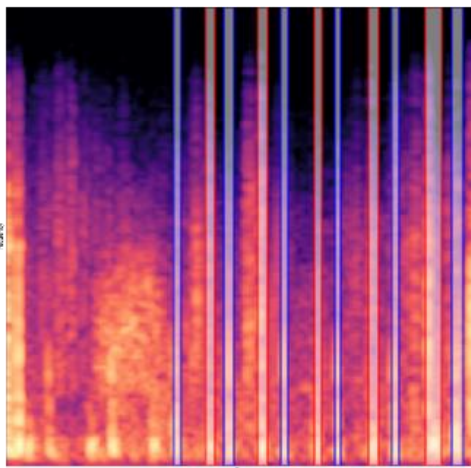
3

프로젝트 수행과정

Overview



Waveform → Mel spectrogram → CNN Objectdetection



3

프로젝트 수행과정

Data EDA



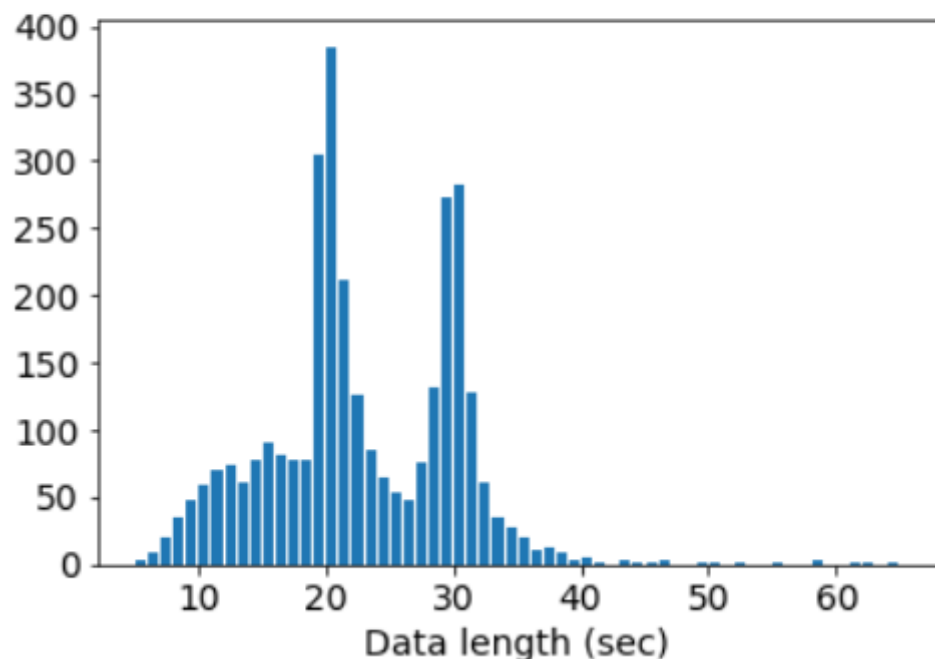
평균 : 22.87030121719886

최대값 : 64.512

최소값 : 5.152

총 데이터 개수: 3163

5~28초 사이의 데이터 개수: 2545



5sec(0.6)

data: 5421

label: 5421

5sec(0.2)

data: 1841

label: 1841

s1 길이 평균 : 0.119164496582365

s1 길이 최대값 : 0.7291699999999999

s1 길이 최소값 : 0.0082090000000000799

s1 표준편차 : 0.02380322459624547

s2 길이 평균 : 0.1053391844957107

s2 길이 최대값 : 0.67974999999999985

s2 길이 최소값 : 0.00087000000000008146

s2 표준편차 : 0.023438662372884743

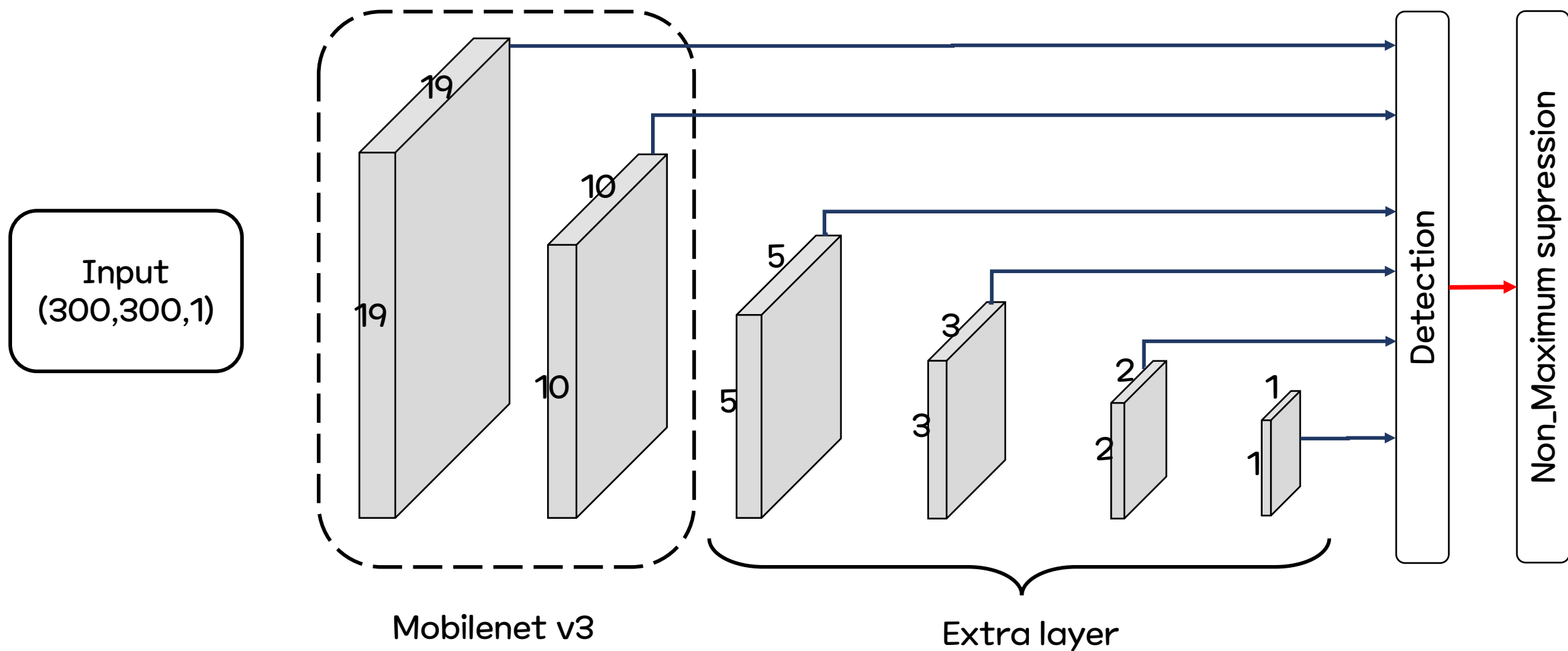
s1 라벨 개수 : 64372

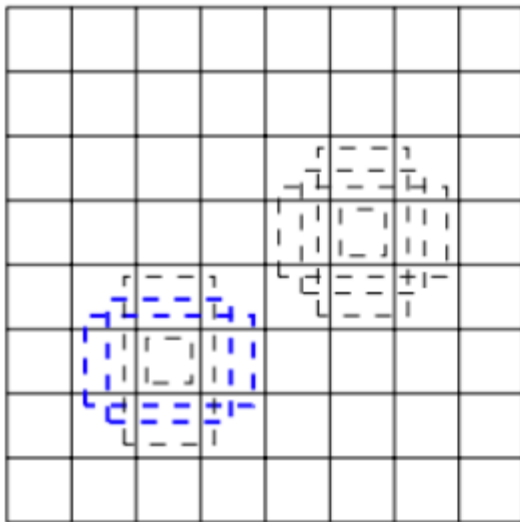
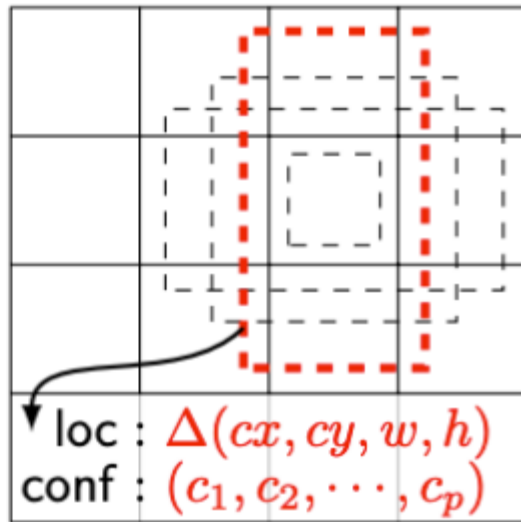
s2 라벨 개수 : 63763

3

프로젝트 수행과정

Model Architecture



(b) 8×8 feature map(c) 4×4 feature map

W, H 계산방식

- ① featuremap 1대 1 비율로 area 크기 설정
- ② 이후 다음 스텝의 비율 area 맞게 너비 조절
- ③ ratio 값을 줘서 추가적인 default box에 대해 같은 넓이를 공유하도록 w, h 설정
- ④ 이후 모든 h에 대해 0.7로 고정

3

프로젝트 수행과정

Loss Function



softmax

{Batch, box_n, class_n}

$$L_{conf}(x, c) = - \sum_{i \in Pos}^N x_{ij}^p \log(\hat{c}_i^p) - \sum_{i \in Neg} \log(\hat{c}_i^0) \text{ where } \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

{Batch, box_n, 4}

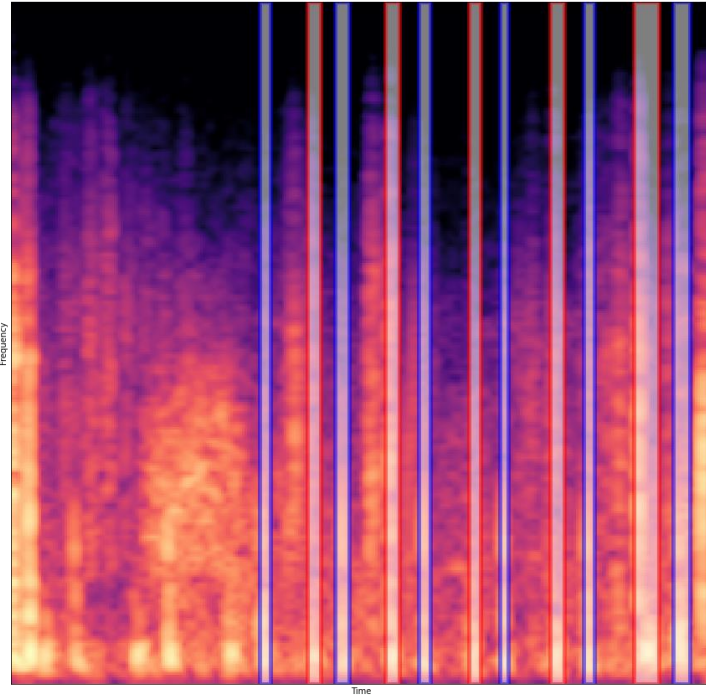
$$L_{loc}(x, l, g) = \sum_{i \in Pos}^N \sum_{m \in cx, cy, w, h} x_{ij}^k \text{smooth}_{L1}(l_i^m - g_j^m)$$

$$L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

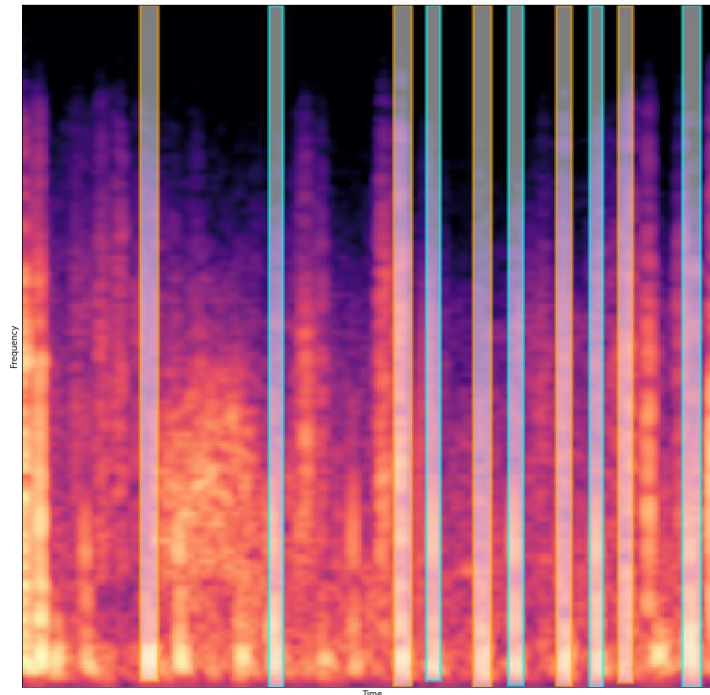
3

프로젝트 수행과정

Base Model Result

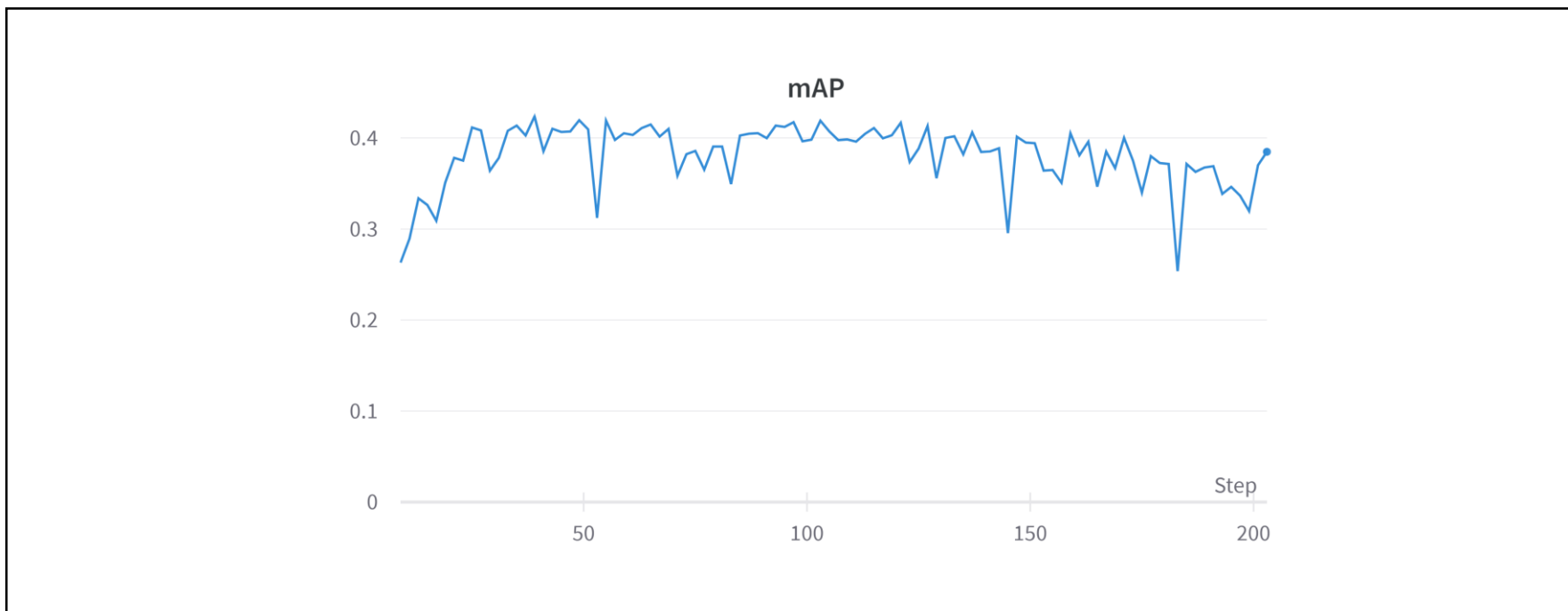


GT



Pred

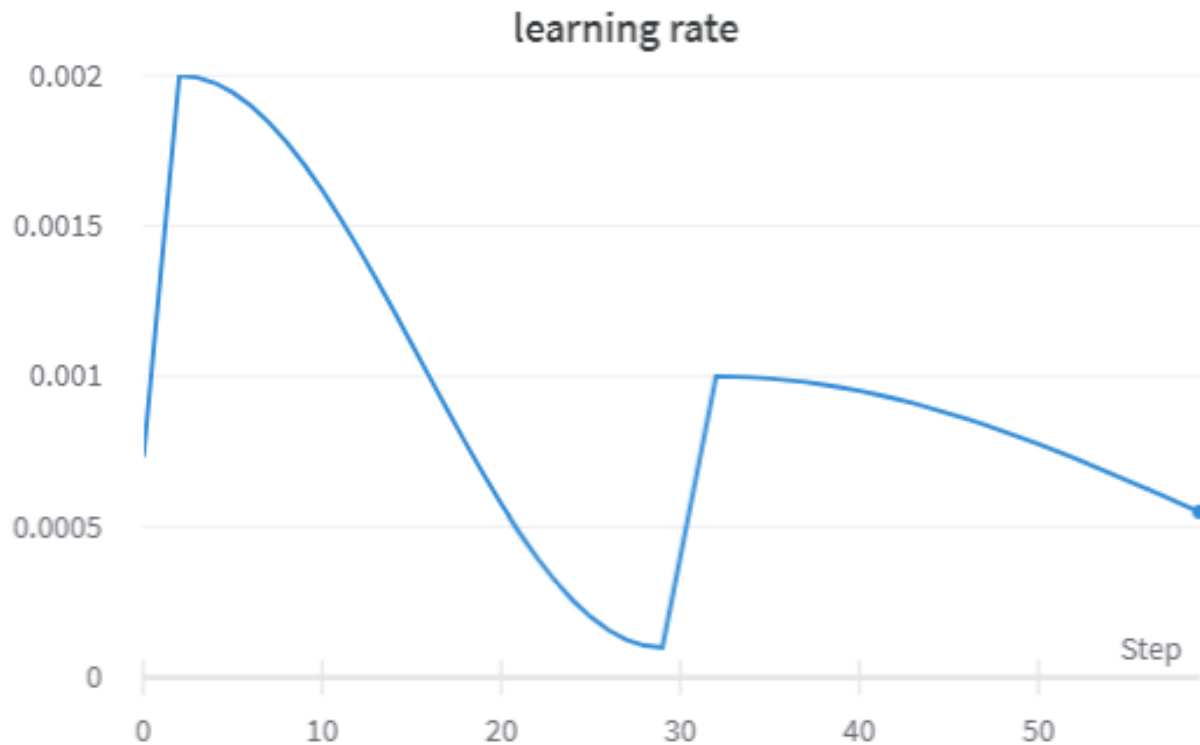
mRecall 61% , mPrecision 72% , mAP 0.528 (iou 0.5 기준)



(CosAnnealing Scheduler, Adam)

3

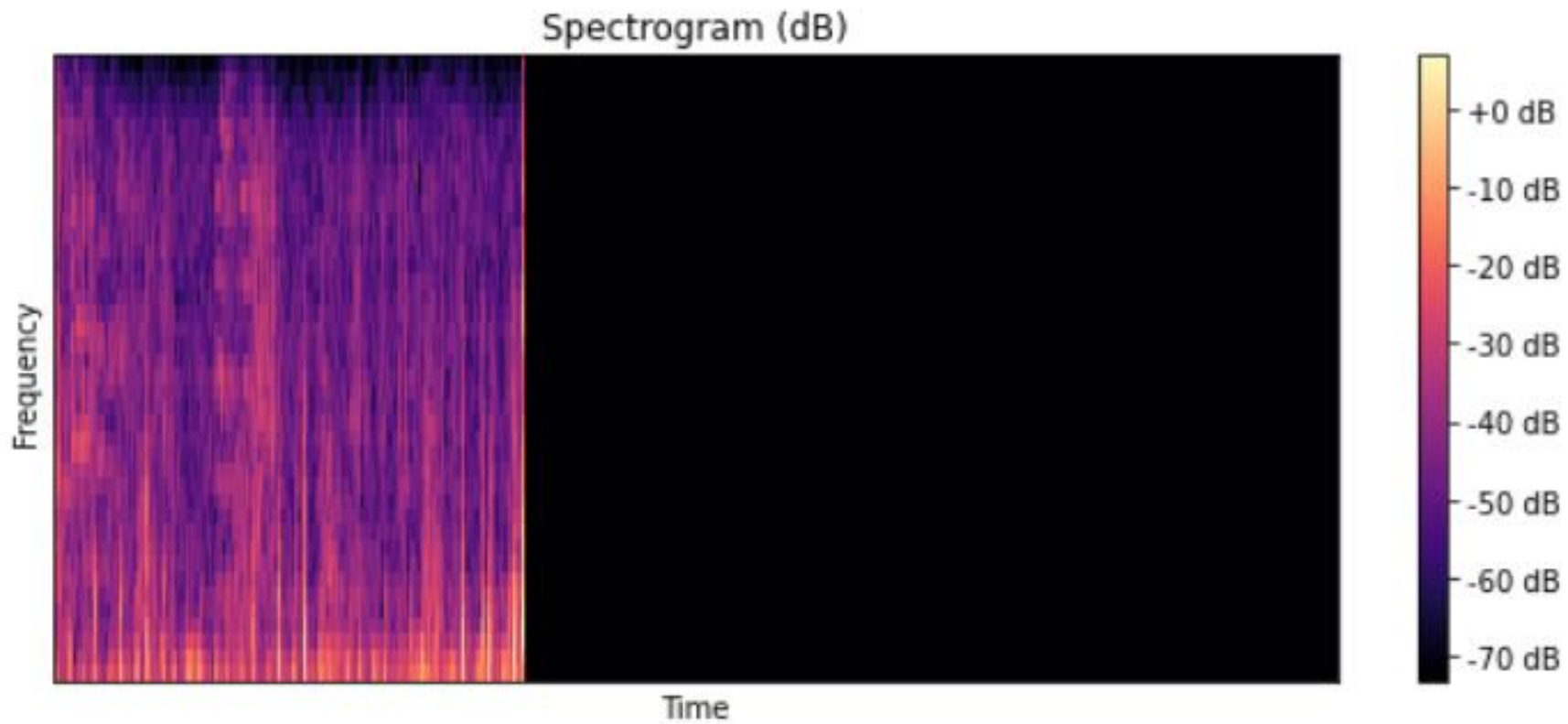
프로젝트 수행과정 Improvements



3

프로젝트 수행과정

Data processing

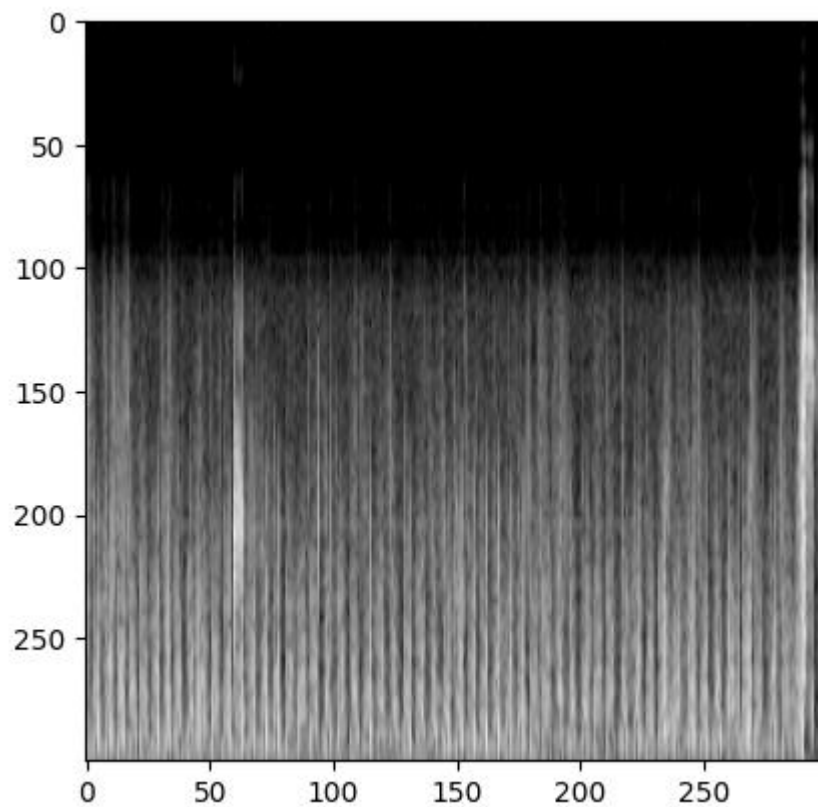


3

프로젝트 수행과정

Data processing

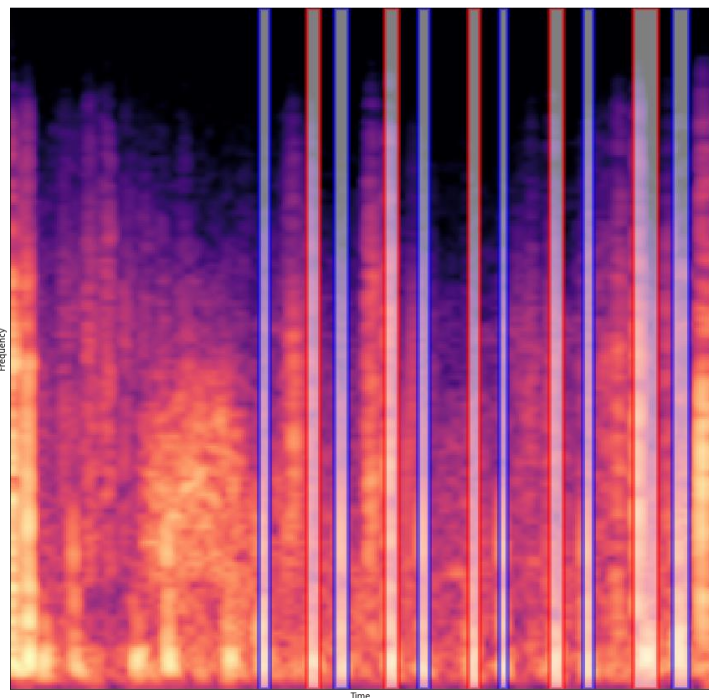
✓ 1채널 이미지 시각화



3

프로젝트 수행과정

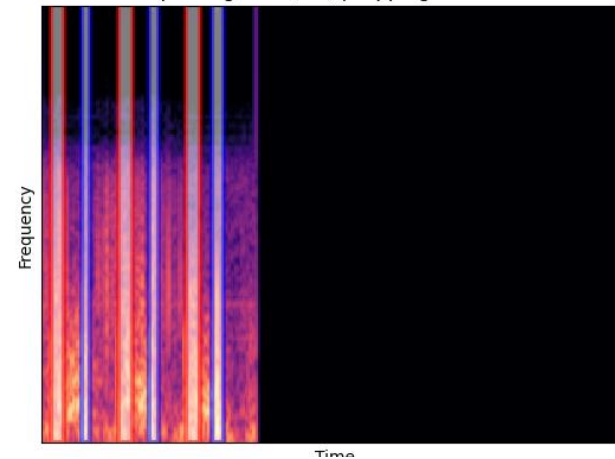
Data processing



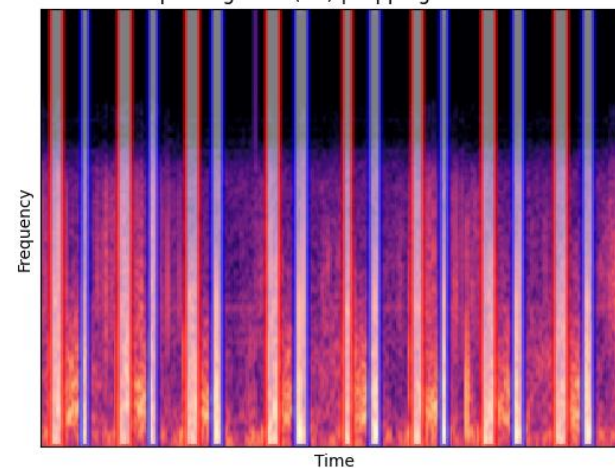
.tsv file

S	E	class
0.00	5.31	0
5.31	5.42	1
5.42	5.88	2
•		
•		
•		
22.31	24.42	2
24.42	24.88	3
24.88	28.57	0

Spectrogram3 (dB) | clipping row=1.0



Spectrogram3 (dB) | clipping row=1.0



3

프로젝트 수행과정

Data processing



✓ 필터링 과정을 통해 주변 노이즈를 최대한 줄인다

✓ LowPass

- 컷오프 주파수 이하의 신호를 통과, 그 이상의 주파수 차단

✓ HighPass

- 컷오프 주파수 이상의 신호를 통과, 그 이하의 주파수 차단

✓ BandPass

- 특정 주파수 범위 내 신호만을 통과, 그 외의 신호는 차단

✓ Peak

- 특정 주파수 주변 신호를 강조하거나 약화

✓ Notch

- 특정 주파수 범위 신호 차단, 그 외의 신호는 통과

✓ Lowshelf

- 컷오프 주파수 이하의 모든 주파수에 대해 신호의 진폭을 증가시키거나 감소

✓ HIGHSELF

- 컷오프 주파수 이상의 모든 주파수에 대해 신호의 진폭을 증가시키거나 감소

3

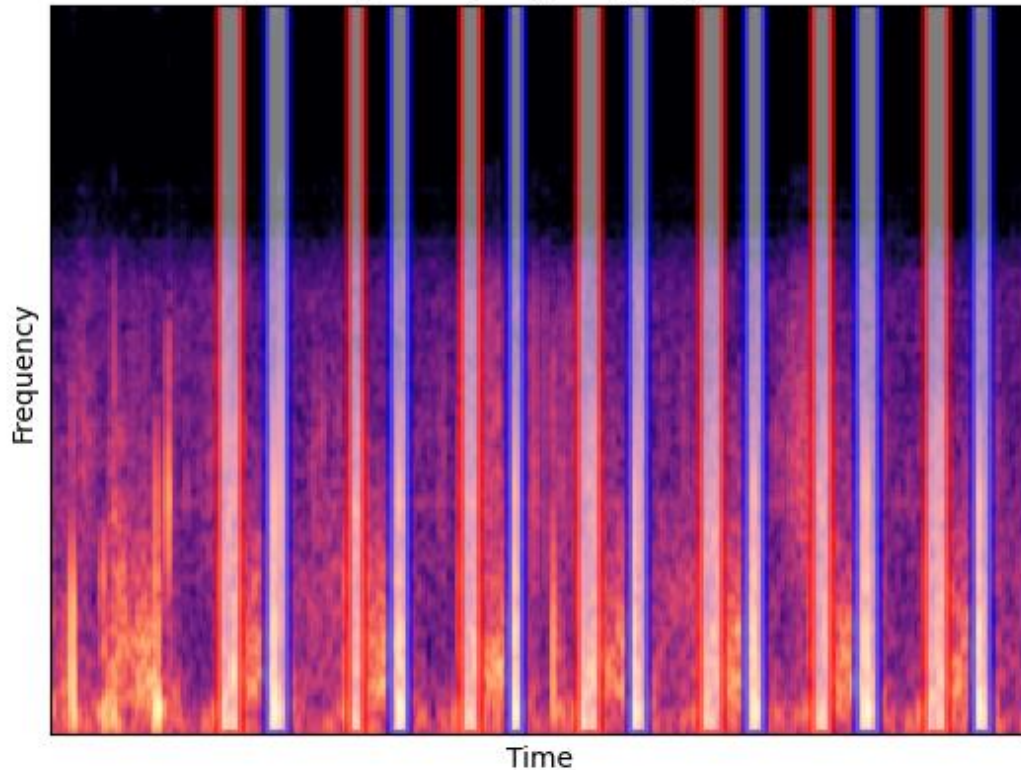
프로젝트 수행과정

Data processing



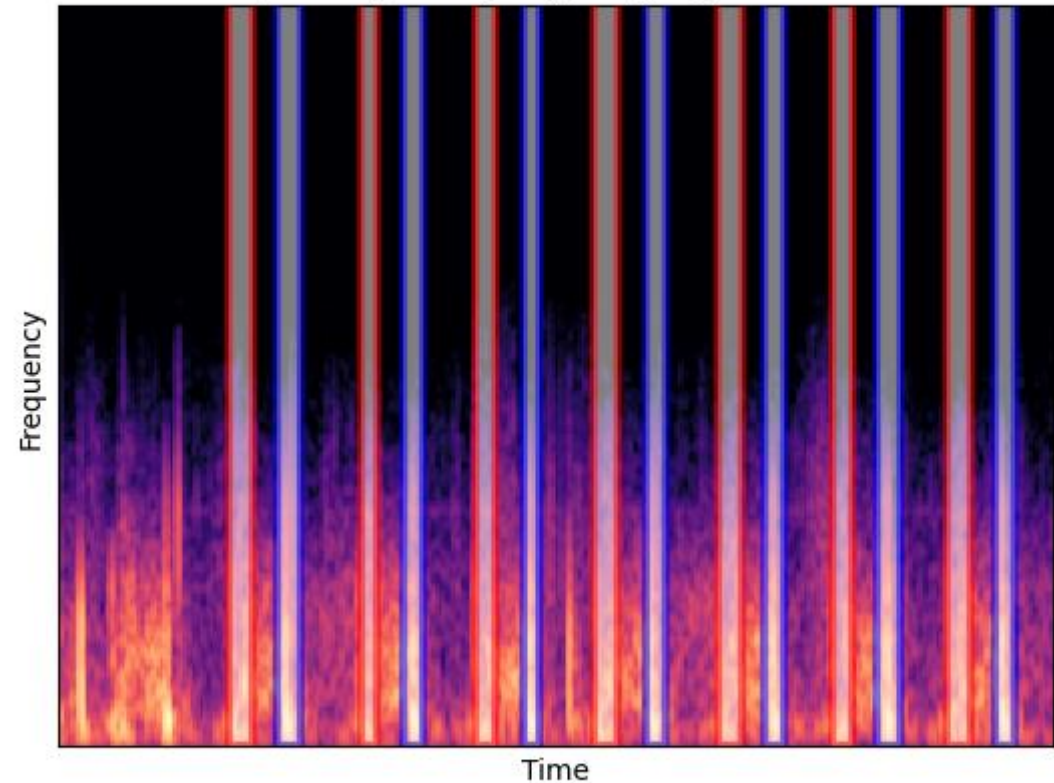
- ✓ 필터링 과정을 통해 주변 노이즈를 최대한 줄인다

Spectrogram1 (dB) | clipping row=1.0



Source

Spectrogram1 (dB) | clipping row=1.0



Highpass & LowPass

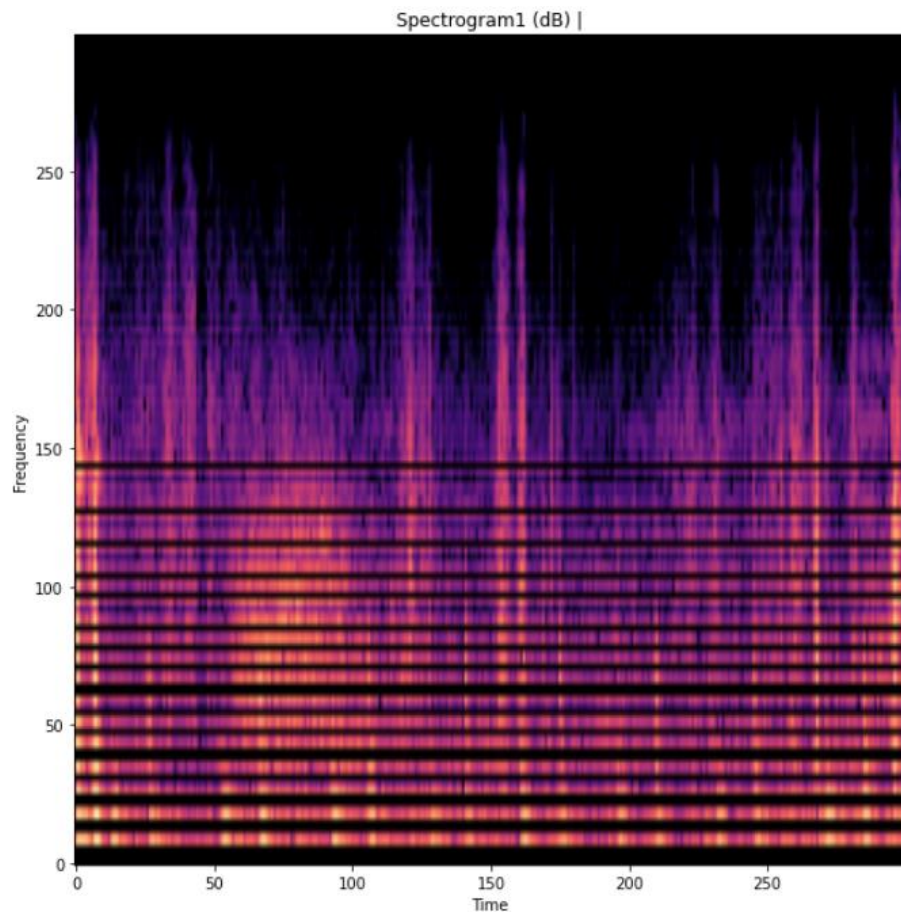
3

프로젝트 수행과정

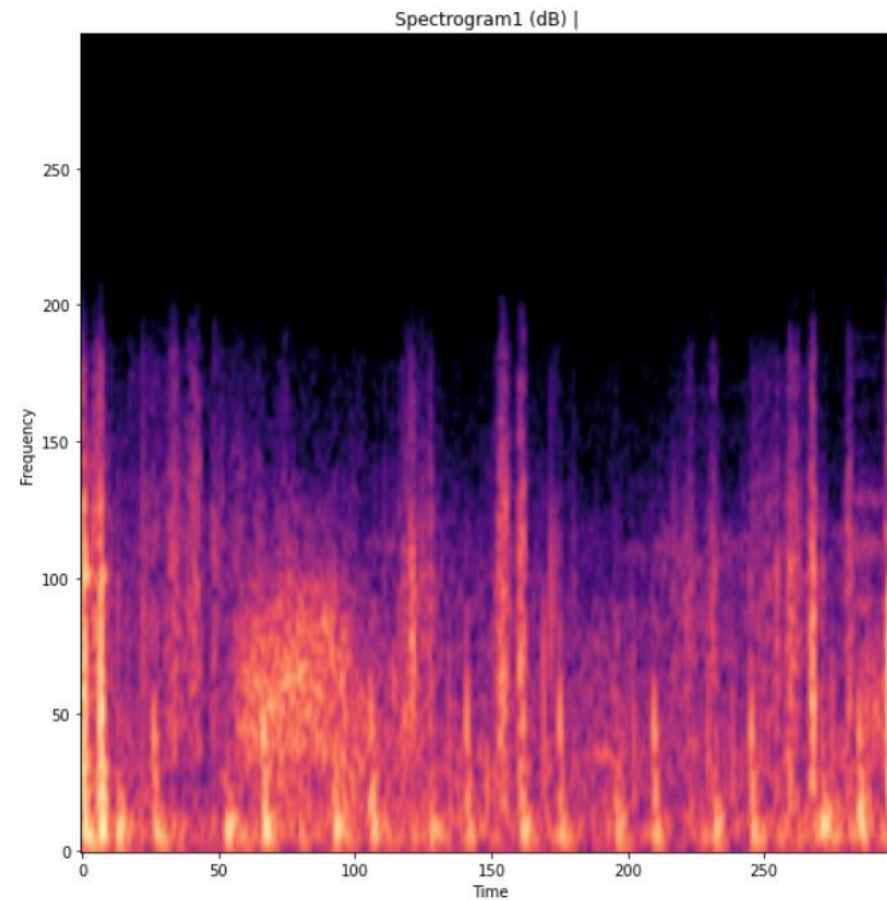
Data processing



- ✓ 주로 오디오 처리에 사용하는 기본인자를 사용 시 해상도와 심한 노이즈가 생기는 문제 발생



Before



After



프로젝트 수행과정

Data processing



NoiseAug
LoudnessAug
PitchAug
Freq_mask
Time_mask

3

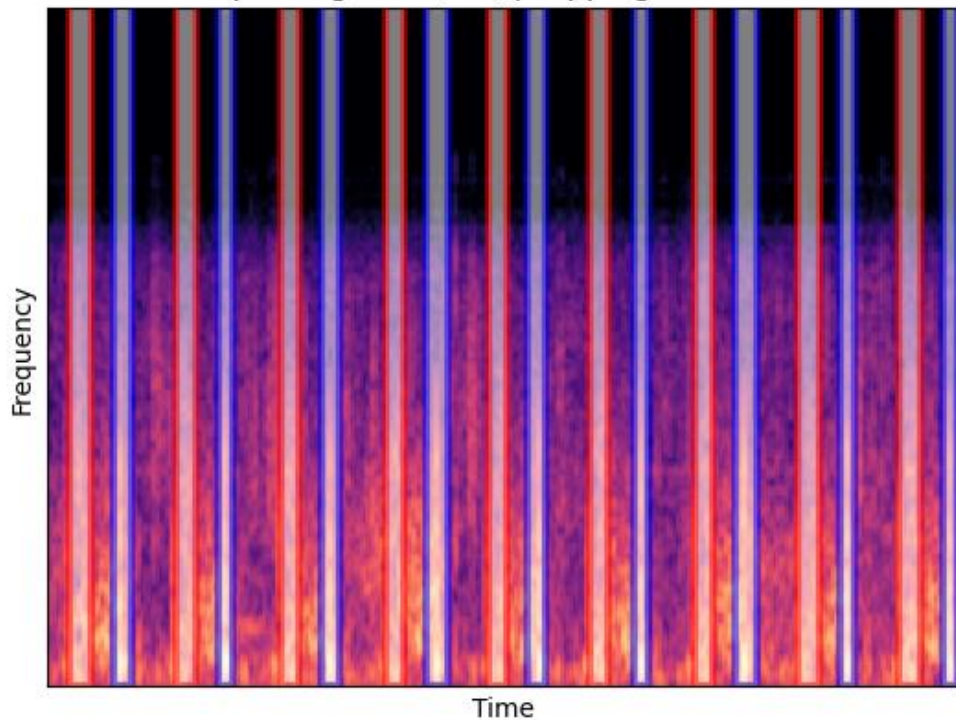
프로젝트 수행과정

Data processing – NoiseAug



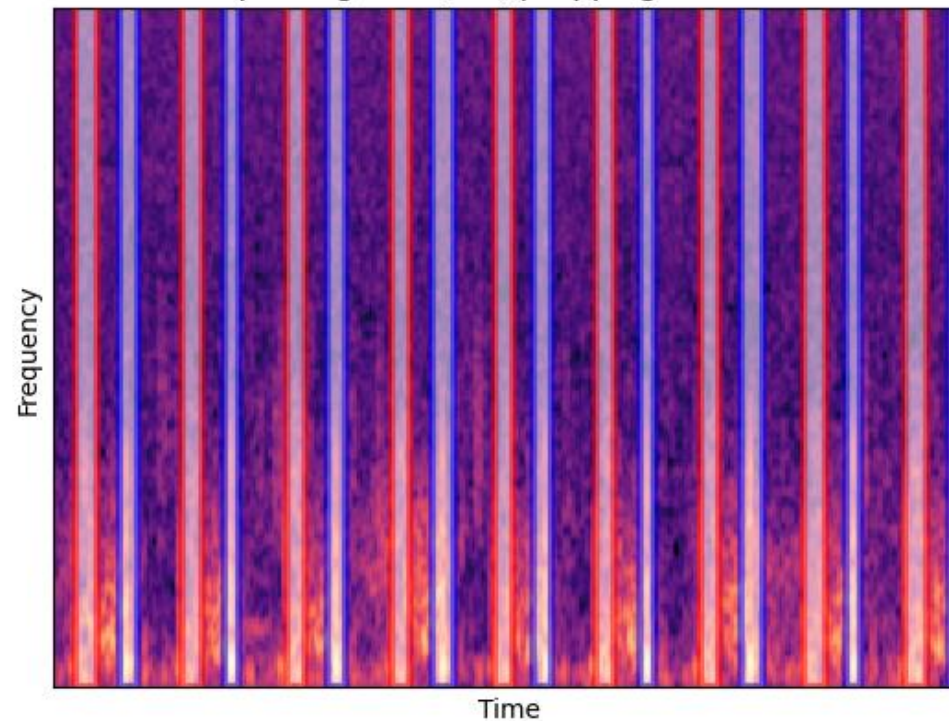
- ✓ NoiseAug 기법은 오디오 신호에 노이즈를 추가한다

Spectrogram2 (dB) | clipping row=1.0



Before

Spectrogram5 (dB) | clipping row=1.0



After

3

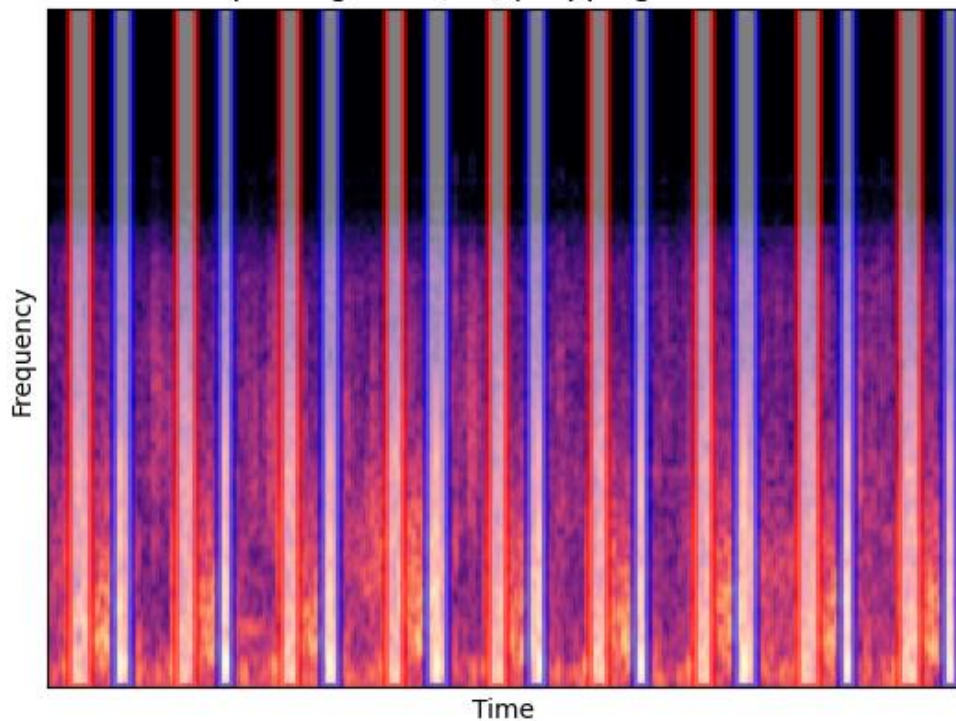
프로젝트 수행과정

Data processing – NoiseAug



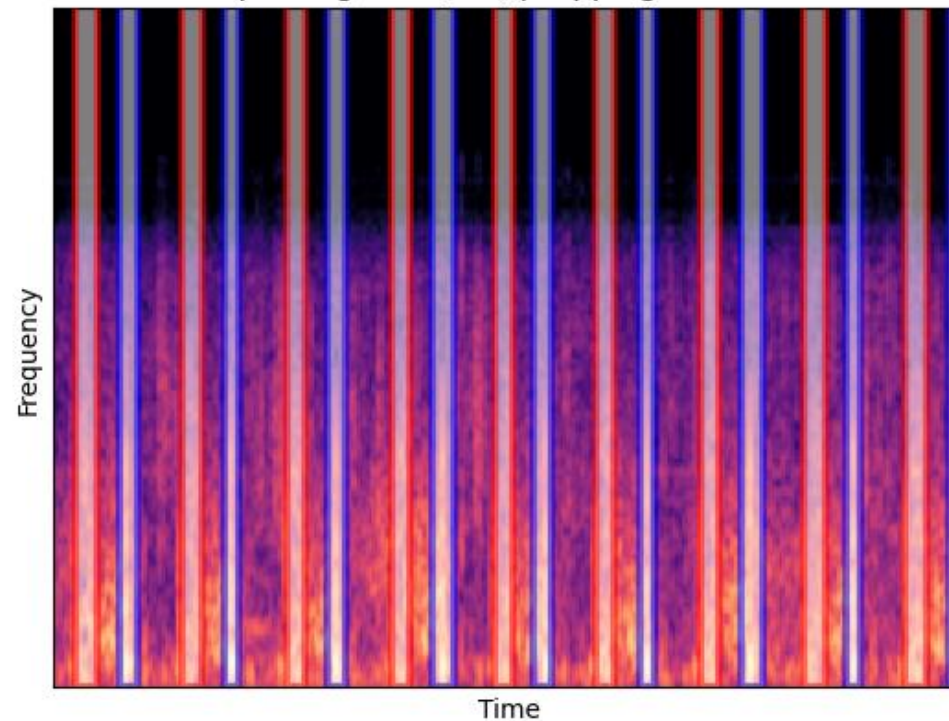
- ✓ LoudnessAug 기법은 오디오 신호에 음량을 조절합니다

Spectrogram2 (dB) | clipping row=1.0



Before

Spectrogram8 (dB) | clipping row=1.0



After

3

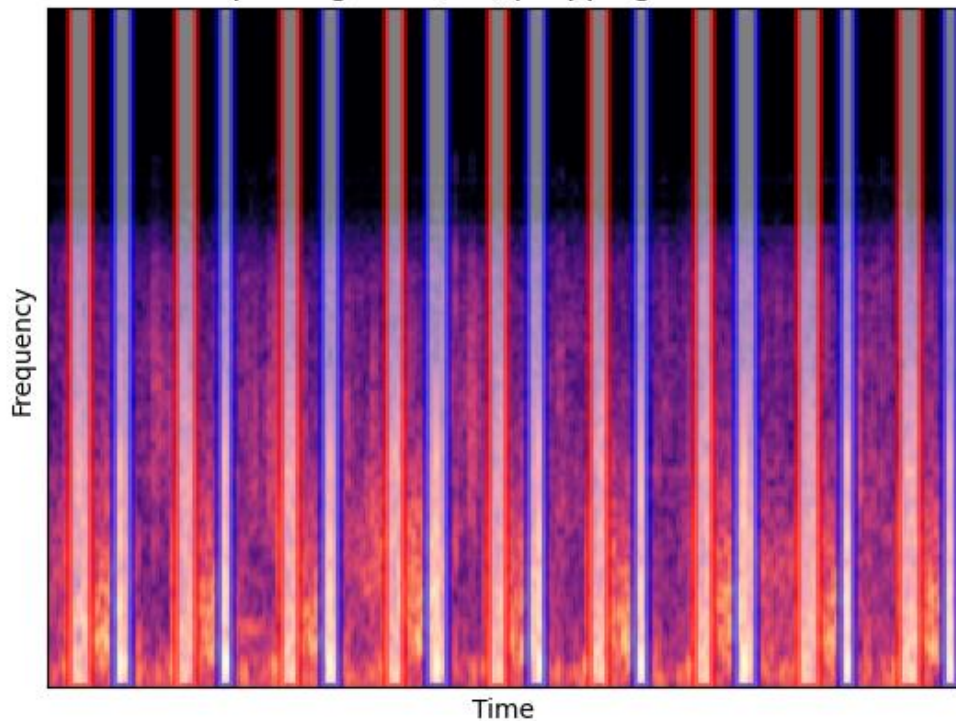
프로젝트 수행과정

Data processing – NoiseAug



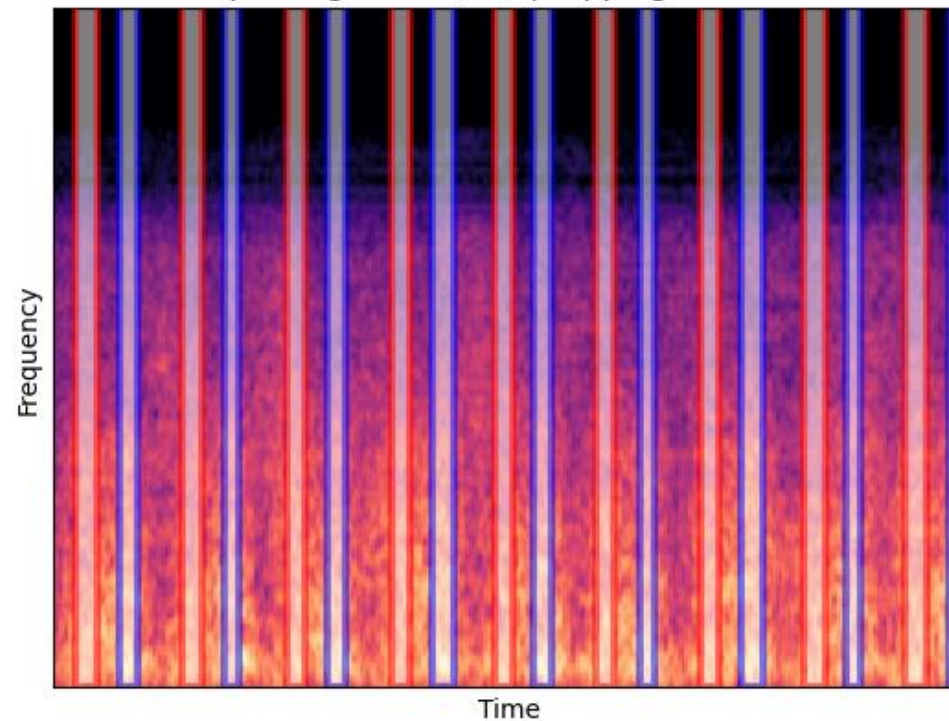
- ✓ PitchAug 기법은 오디오 신호의 음성의 높낮이를 변경합니다

Spectrogram2 (dB) | clipping row=1.0



Before

Spectrogram11 (dB) | clipping row=1.0



After

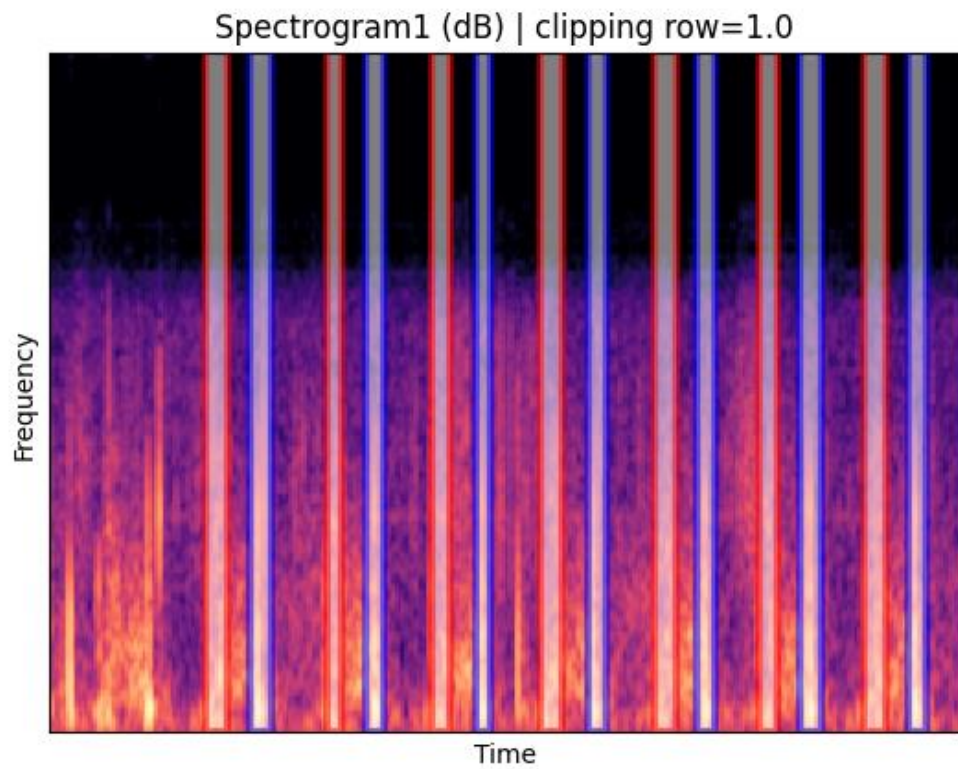
3

프로젝트 수행과정

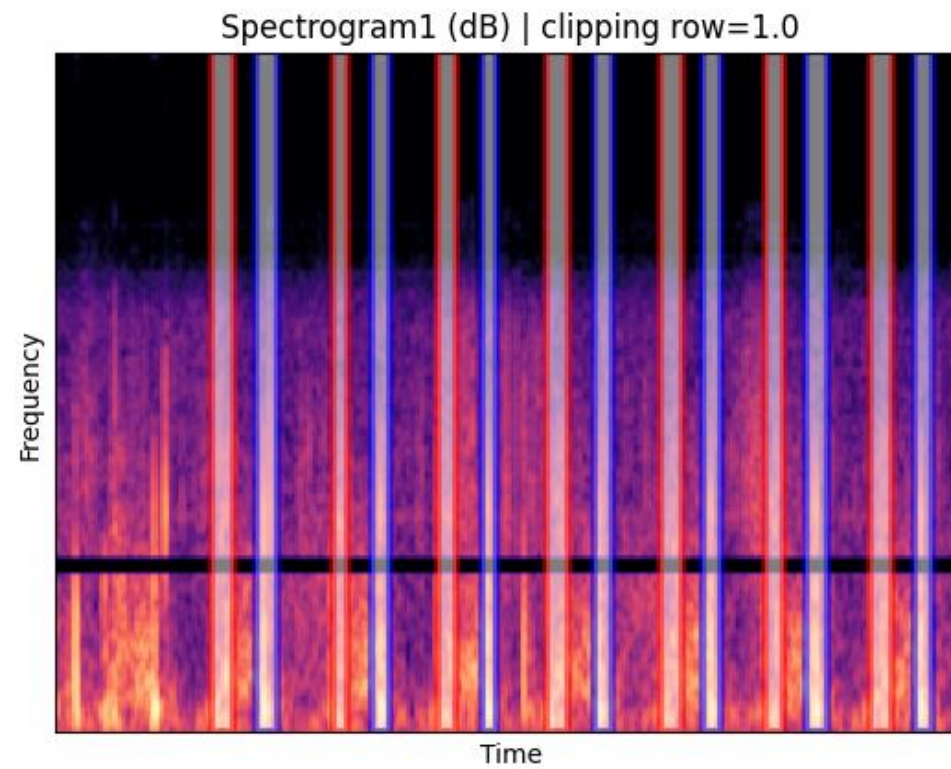
Data processing – NoiseAug



- ✓ Frequency Masking 기법은 시간 축에 평행한 마스크를 덮어 씹습니다



Before



After

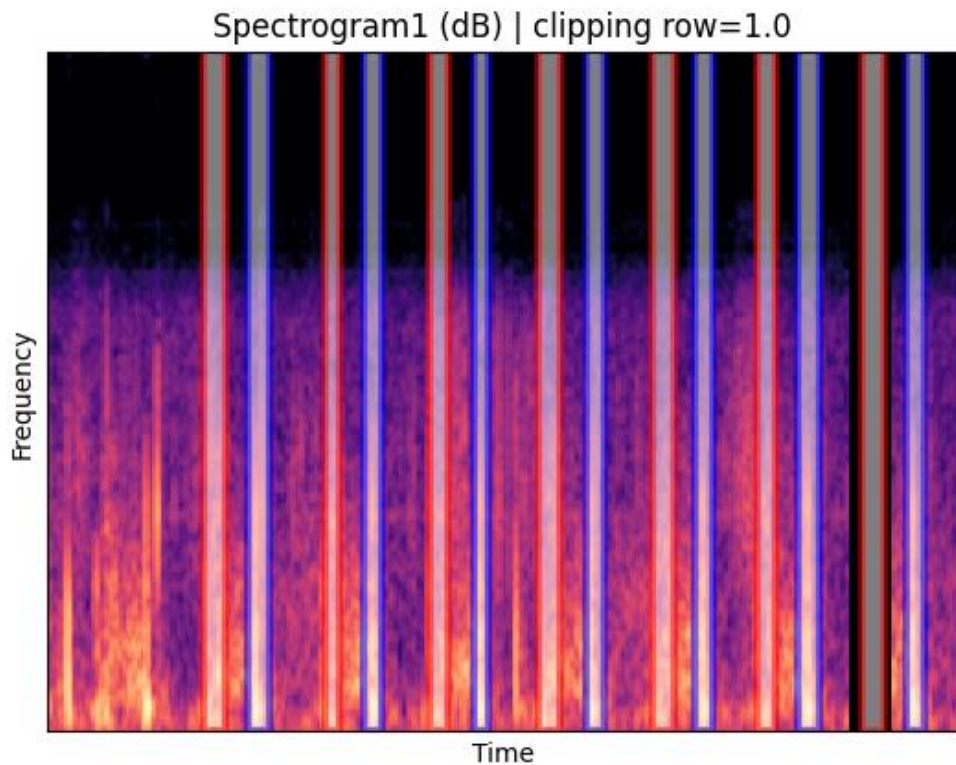
3

프로젝트 수행과정

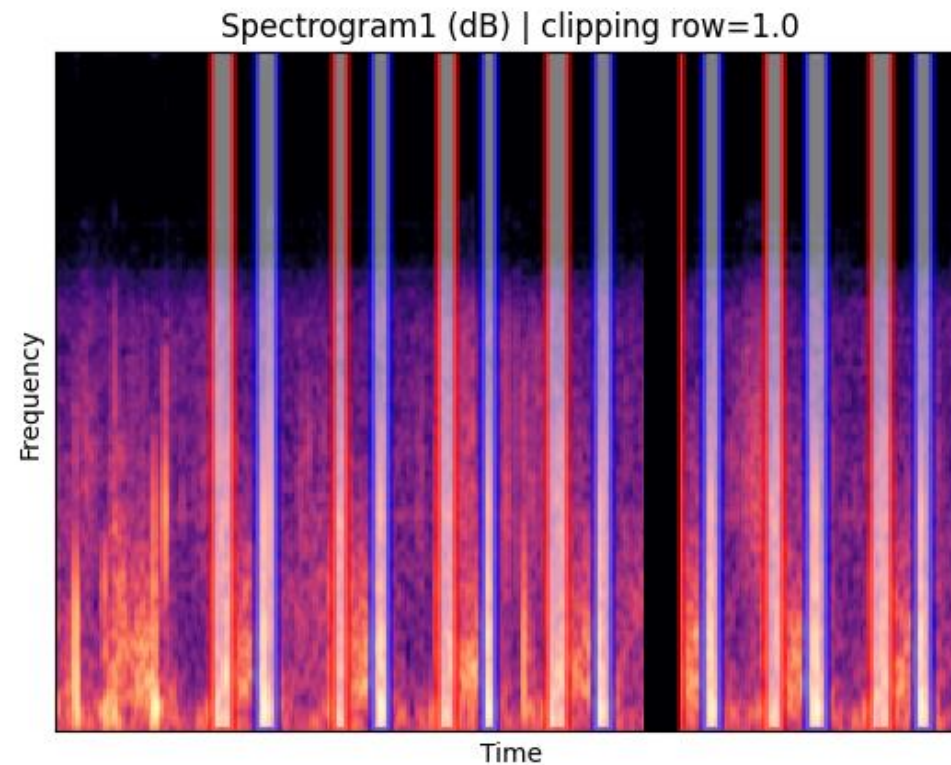
Data processing – NoiseAug



- ✓ Time Masking 기법은 Frequency축에 평행한 마스크를 덮어 씹습니다



Type 1



Type 2



Before

mPrecision 72% , mRecall 61% , mAP 0.528 (iou 0.5 기준, 1채널)

After

mPrecision 87.29% , mRecall 81.8% , mAP 0.79

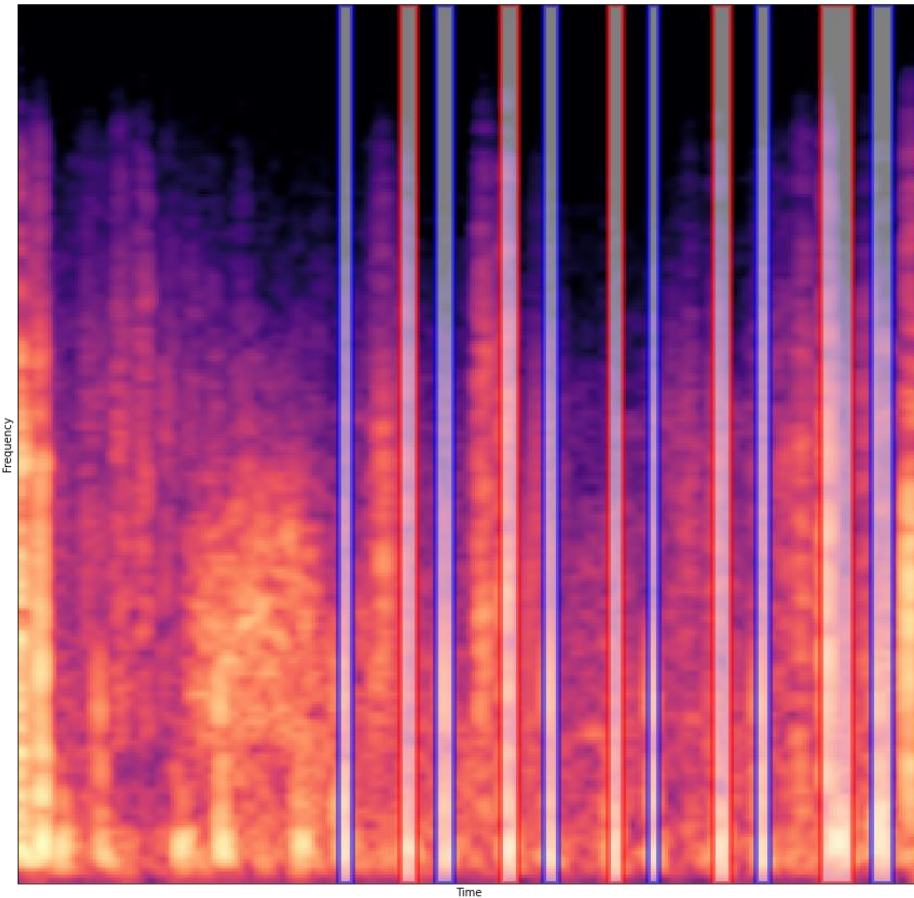
S1 Precison 82.3% , S2 Recall 81.3%

S1 Precison 86.5% , S2 Recall 82.3%

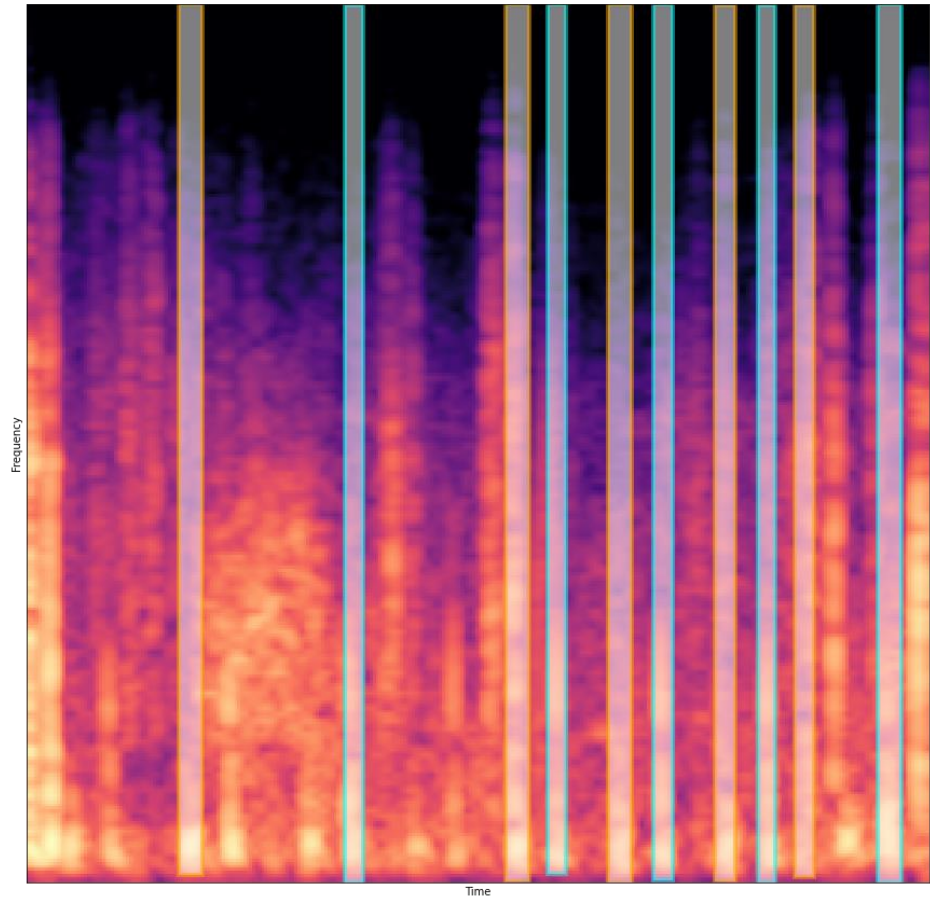
(iou 0.5 기준, 1채널)



source

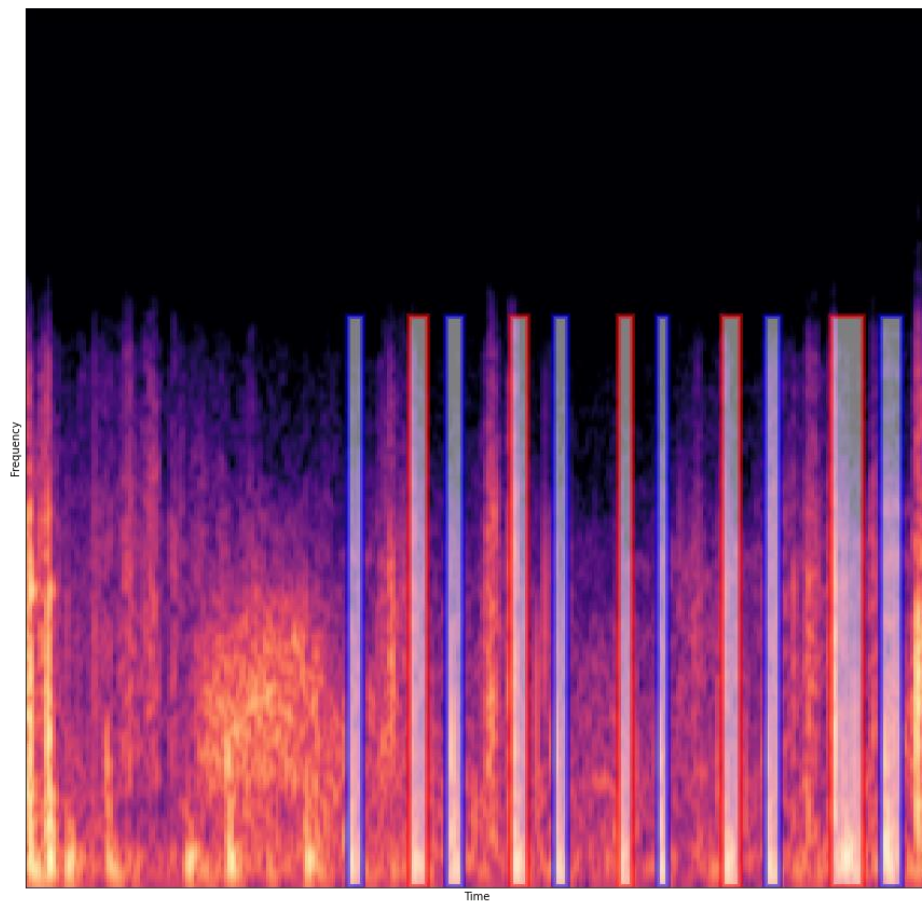


Early

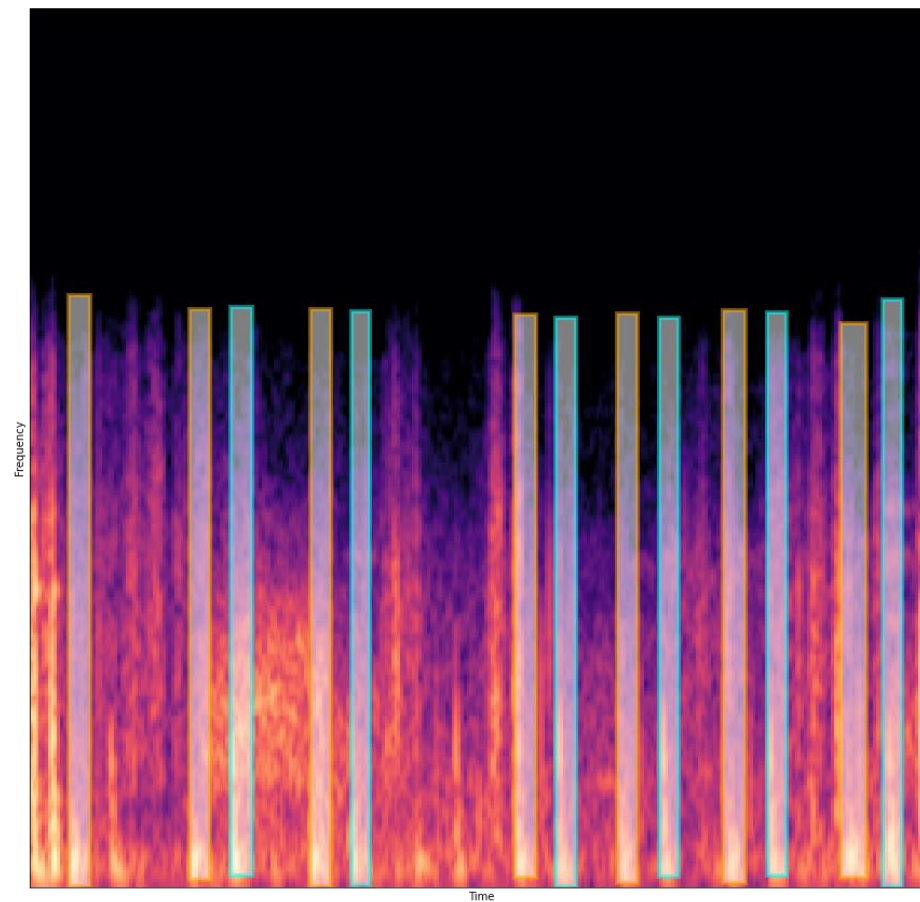




source



Final



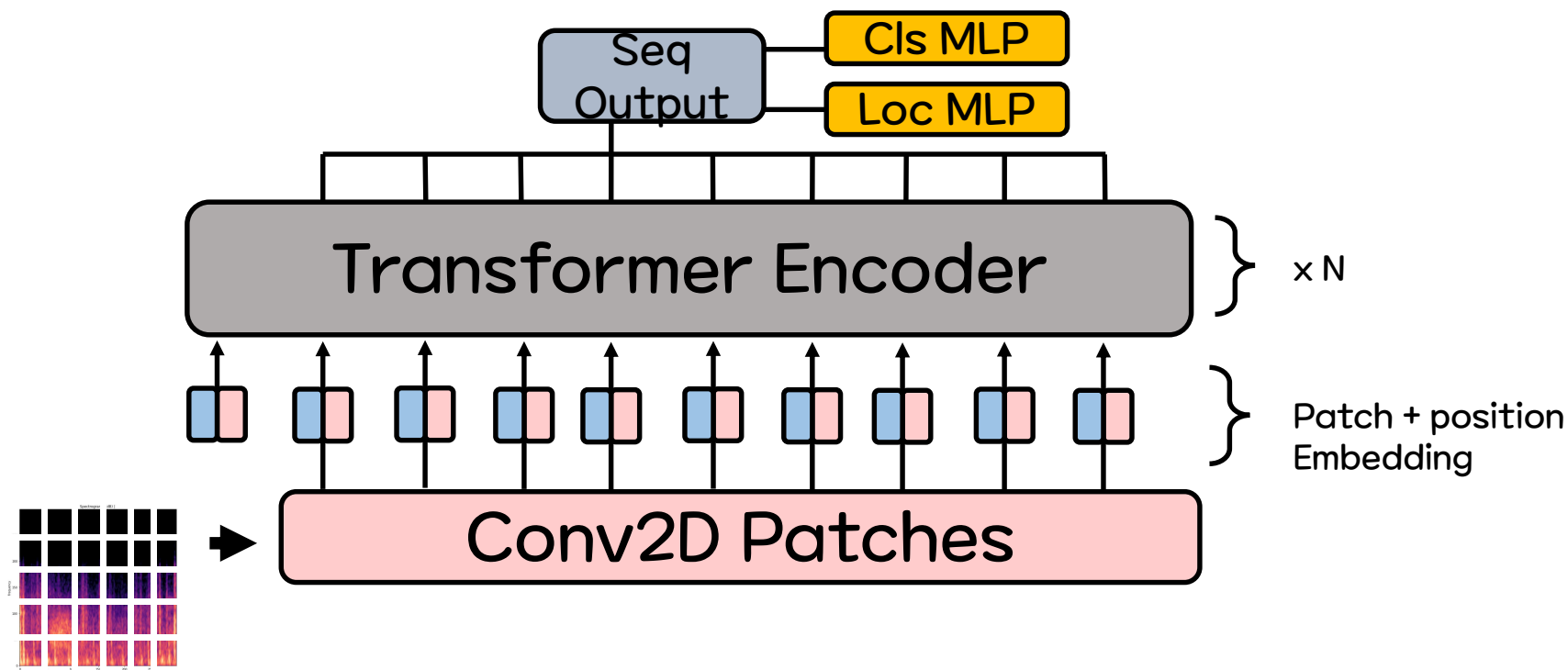
5

후속 연구

ViT



- ① 오디오 이미지를 Conv2D layer를 통해 여러 개의 patch embedding과정을 진행합니다.
- ② 임베딩된 패치에 position 값을 더하고 transformer Encoder Layer를 통과합니다.
- ③ 최종적으로는 layer를 최종적인 output에 대하여 class token를 제거하고 각각 Cls, Loc MLP을 통과시켜 output을 도출합니다



Q&A