Capstone Deisgn2

Environment Sound Classification with CNN

발 표 자 강 명 훈

KYUNG HEE UNIVERSITY

CONTENTS

KYUNG HEE UNIVERSITY

1.연구 배경

주제 변천사

Multiple Speaker Recognition Application with CNN

Gender Sound Classification with CNN

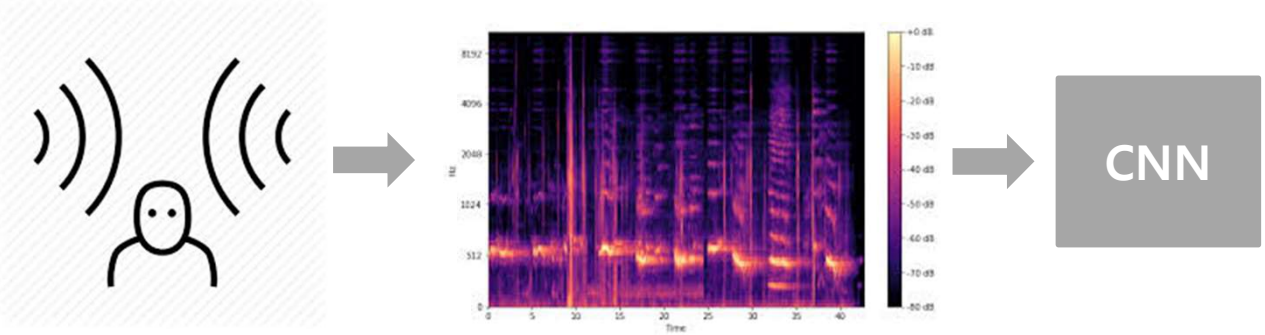
Environment Sound Classification with CNN

01

“ 연구 주제 ”

CNN을 이용 한 Environment Sound Classification

Mel Spectrogram과 CNN을 이용하여 장소를 구분 할 수 있는 Model 구현



기존 연구와 차이

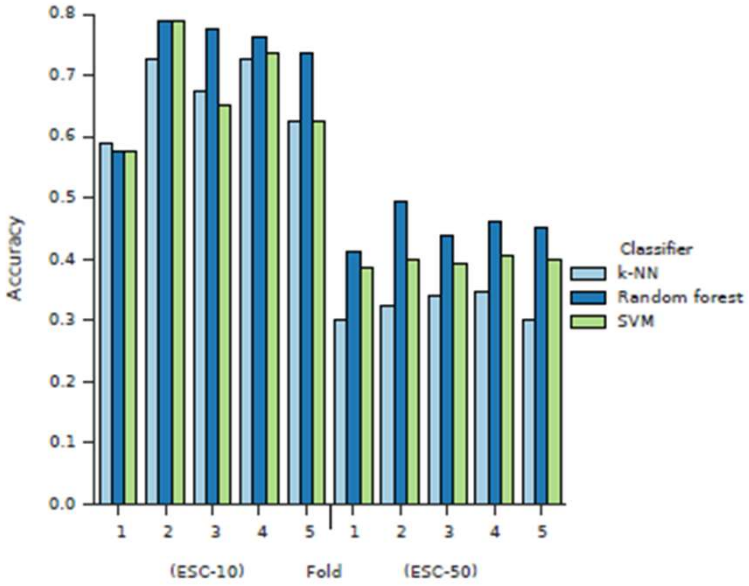
- 단일 Sound가 아닌 복합 Sound에서 Classification 실험

연구 목표

- Classification 정확도 80% 이상

관련 연구 1

ESC: Dataset for Environmental Sound Classification



- Common Environmental Sound
- ESC-10 , ESC-50
- Zero-Crossing Rate, MFCC
- HUMAN, K-NN, Random Forest, SVM

관련 연구 2

Performance Comparison of Acoustic Features for Sound Classification on Ubiquitous Environment

1) Acoustic features

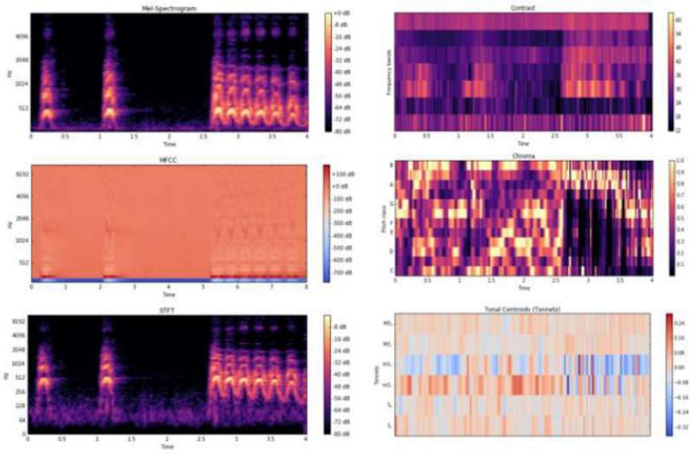


Fig. 1. Converted acoustic features (clockwise from right-top to left-top: Contrast, Chroma, Tonal Centroids, STFT, MFCC, Mel-spectrum)

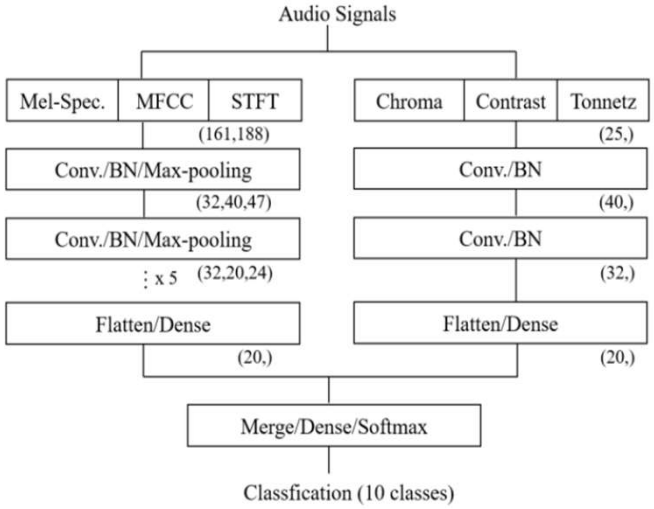


Fig. 2. Training model

- **Mel-Spec, MFCC, STFT, Contrast, Chroma, Tonal Centroids**
- **Freefield1010: 86%, Urban Sound 8K dataset: 78%**

2.연구 과정

02

데이터 수집 및 가공

```python
from pydub import AudioSegment

def cutaudio(inputfile, label, maxtime, start_point):
    t1 = start_point
    # cut the sample by 5 second if you want another second change the plus var
    t2 = t1+5
    max_time = t1 + maxtime

    while t1<max_time:
        #in pydub use ms second so multiple 1000
        #start
        st1 = t1*1000
        #end
        et2 = t2*1000
        newAudio = AudioSegment.from_wav(inputfile)
        newAudio = newAudio[st1:et2]
        #make naming numbering
        number = t1/5
        name = str(number)+label
        newAudio.export('audio8/'+name+'.wav', format="wav")
        t1 = t1+5
        t2 = t1+5
```

- 주변 환경 직접 녹음
- Online Data Sound DB
- Sound Duration: 5 Sec
- Sound Rate: 44100hz

데이터 분류

- 5가지 Category: Nature, Loud, Music, Quiet, ETC
- 15가지 Sub Class

Nature	Loud	Music	Quiet	ETC
Mountain	Factory	Music Concert	Home	Road
Sea	Airport	Classic Concert	Office	Lecture
Cave	Harbor	Club	Library	Cafe

모델 훈련

- 음원 파일 -> Mel Spectrogram 변환
- Train 90%, Validation 10%, Test 10%
- Model의 Accuracy, Loss 확인
- 신뢰도가 낮다고 판단되면 Data Set 추가 및 모델 변경

3.연구 결과

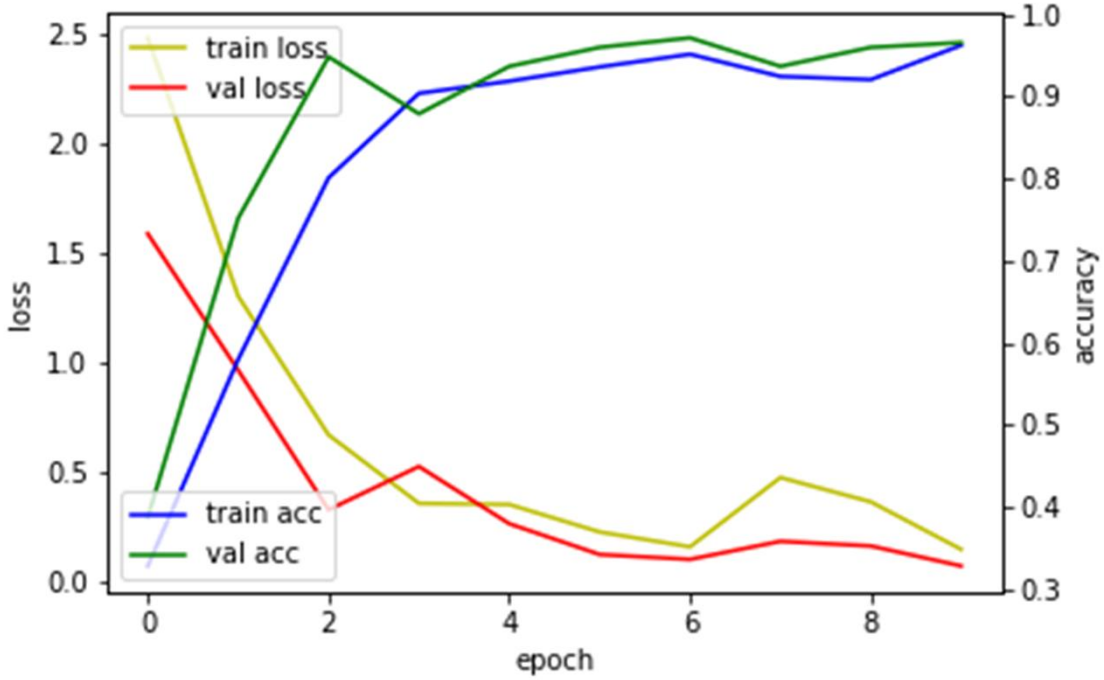
모델

Result3 의 Model

```
model = Sequential()

# add layers
model.add(Conv2D(64, kernel_size=3, activation="relu", input_shape=(SPEC_H, SPEC_W, 1)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(128, kernel_size=3, activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(256, kernel_size=3, activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(256, kernel_size=3, activation="relu"))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(256, activation = 'relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation = 'relu'))
model.add(Dense(15, activation="softmax"))


# compile the model
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy'])

# Training and Evaluation of the model
hist = model.fit(train_x, train_y, batch_size = 30 ,epochs=30,validation_split=0.1)
```

Result1

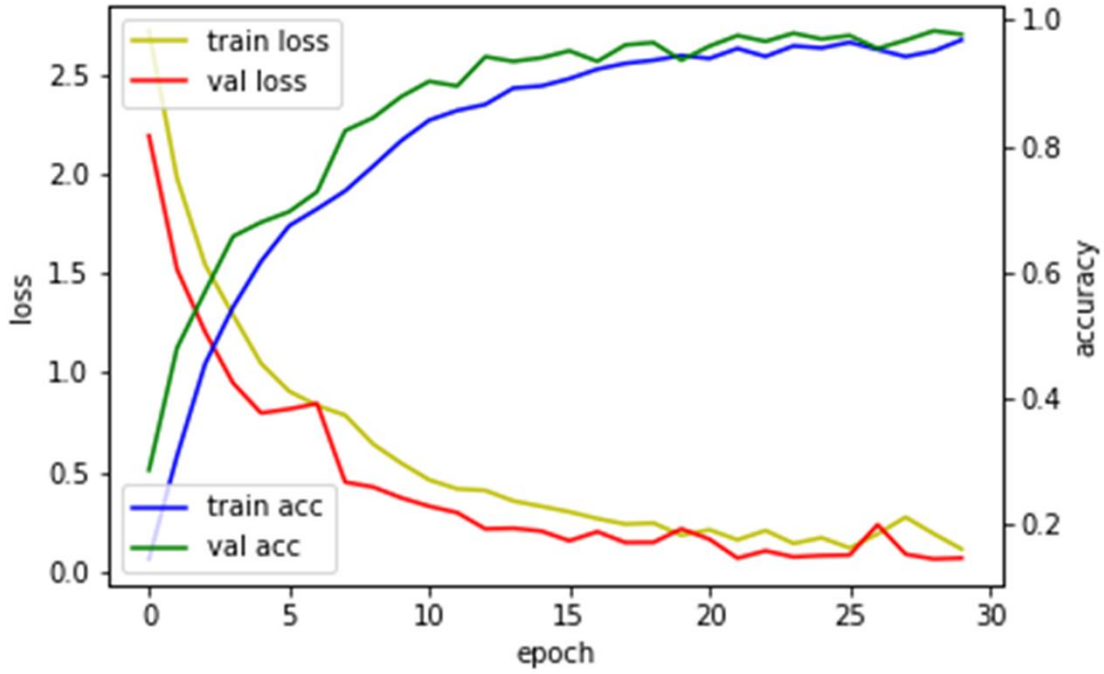
Categories: 3 Classes: 9 Sample Count: 960



- 모델의 신뢰도 확인이 어려움
- 더 많은 데이터를 이용하여 모델을 비교하기로 결정

Result 2

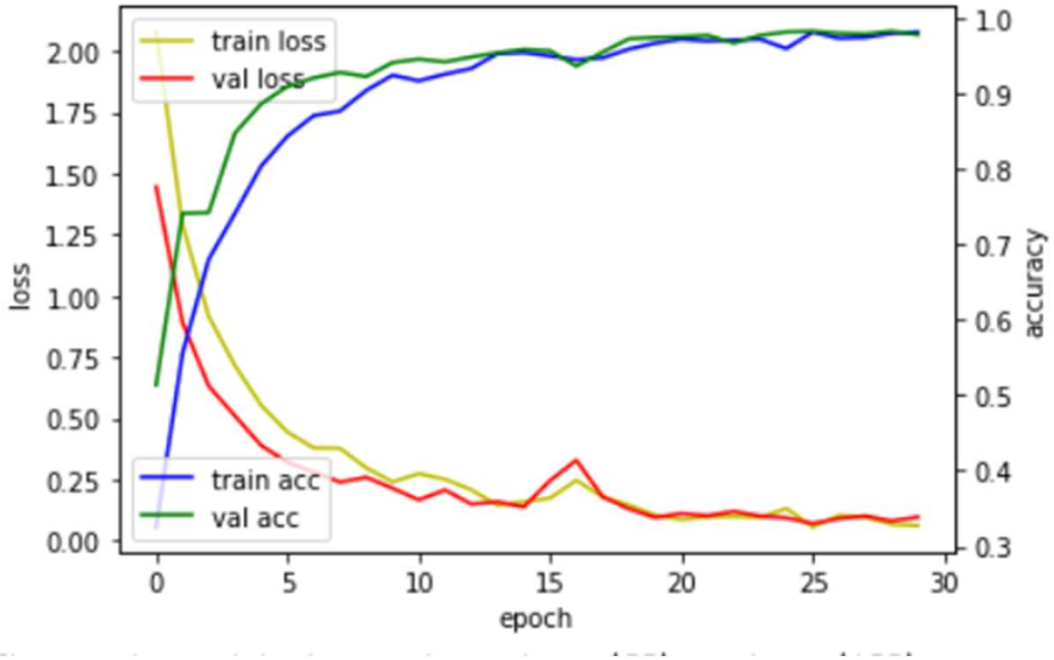
Categories: 5 Classes: 15 Sample Count: 2974



- Train1 보다 더 좋은 형태 확인
- 좀 더 객관화 시키기 위해 데이터를 추가하기로 결정

Result 3

Categories: 5 Classes: 15 Sample Count: 6000



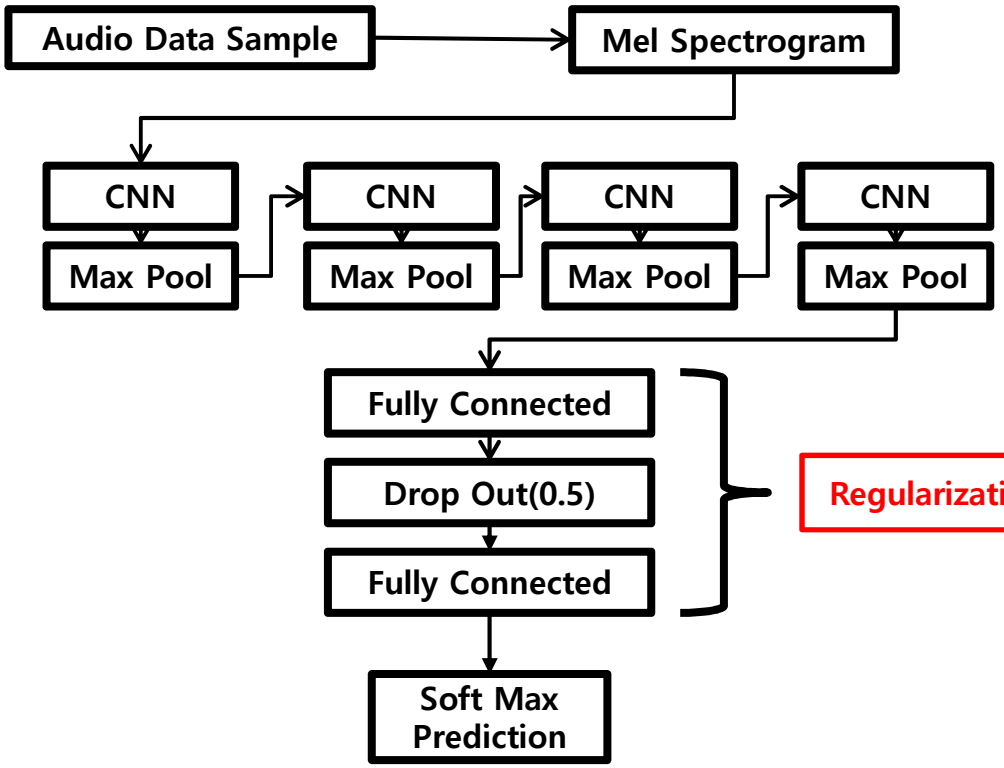
- Train2 보다 더 좋은 형태 확인
- 모델의 구조적 변경 시점이라 판단

Model Evaluate

Train 4에서 가장 좋은 결과

Model	Train Evaluate	Test Evaluate
Train1(960)	0.926	0.932
Train2(2974)	0.987	0.984
Train3(6000)	0.983	0.962
Train4(6000)	**0.992**	**0.985**
Train5(6000)	0.987	0.972

Model Architecture



Model: "sequential_2"

Layer (type)	Output Shape	Param #
conv2d_5 (Conv2D)	(None, 126, 214, 64)	640
max_pooling2d_5 (MaxPooling2	(None, 63, 107, 64)	0
conv2d_6 (Conv2D)	(None, 61, 105, 128)	73856
max_pooling2d_6 (MaxPooling2	(None, 30, 52, 128)	0
conv2d_7 (Conv2D)	(None, 28, 50, 256)	295168
max_pooling2d_7 (MaxPooling2	(None, 14, 25, 256)	0
conv2d_8 (Conv2D)	(None, 12, 23, 256)	590080
max_pooling2d_8 (MaxPooling2	(None, 6, 11, 256)	0
flatten_2 (Flatten)	(None, 16896)	0
dense_3 (Dense)	(None, 256)	4325632
dropout_2 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 128)	32896
dense_5 (Dense)	(None, 15)	1935

Total params: 5,320,207
Trainable params: 5,320,207
Non-trainable params: 0

4.연구 한계점

연구 검증1

- 만든 모델이 실제로 잘 작동하는지를 테스트
- Train, Val, Test에 쓰이지 않은 전혀 새로운 데이터로 Test

Model Valid Accuracy

Valid Count	Category Accuracy	Class Accuracy
33	72%	55%
45	71%	53%
100	65%	52%
150	66%	49%

검증 결과 분석

Category Accuracy

Category	Category Accuracy	Class Accuracy
Nature	73%	66%
Loud	46%	40%
Music	93%	76%
Quiet	63%	20%
ETC	56%	43%

- 정확도가 좋은 경우는 명확한 중심 Feature 존재
- 너무 많은 Feature가 결합된 장소의 경우 판별을 제대로 하지 못함
- Train에 이용된 데이터가 완벽한 객관적 지표가 아님

5.결론

연구 의의

- 목표한 Accuracy를 달성하지는 못했지만 여러 복합적인 Sound가 섞여 있는 환경에서도 구분이 가능 할 수 있다는 가능성을 발견

- Category와 같은 명확하고 공통적인 Feature를 특정 할 수 있다면 Environmental Sound Classification이 이루어질 수 있을 것으로 기대

추후 연구 방향

- 일반화 성능을 위해서 더 많은 데이터 수집
- 분류를 위한 명확한 Environmental Sound Class 정의
- Mel Spectrogram 말고도 다른 음성 Feature를 이용하여 연구
- Mobile, IOT와 연계하여 확장된 Application 개발

THANK YOU

발 표 자 강 명 훈