호흡음 데이터 이용한 호흡기 질환 진단

발표자: 202133885 최윤주

1. 사용한 데이터

VBOOKSHELF · UPDATED 5 YEARS AGO

463 | New Notebook | Download (4 GB)

Respiratory Sound Database

Use audio recordings to detect respiratory diseases.

- kaggle의 호흡음 음성데이터
- 청진음이란 의사나 의료 전문가가 환자의 폐를 청진할 때 들을 수 있는 소리
- 920개의 데이터 - 각 20초의 .wav 데이터
- 포르투갈과 그리스의 두 연구팀이 직접 제작
- 126명의 환자가 표본
- 클래스는 건강한 사람, 만성 폐쇄성 폐질환, 하기도감염, 상기도감염, 폐렴, 천식, 세기관지염의 총 7가지

2. 데이터 선정 이유

호흡기 주요질환

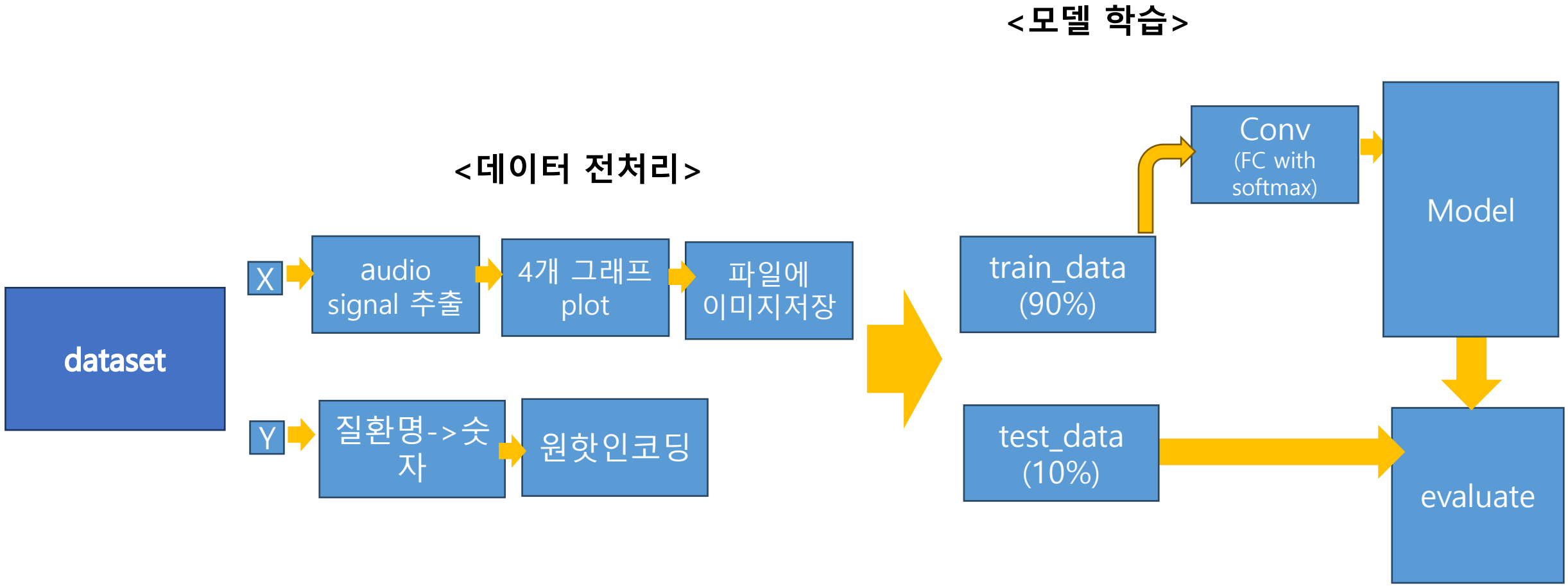
- 만성폐쇄성폐질환(COPD)
- 천식
- 결핵
- 종양(폐암)
- 폐렴(성인)
- 인플루엔자
- 직업성폐질환
- 급성**호흡**곤란증후군

- 호흡기 주요 질환의 수는 상당하다.

  - 정확한 진단의 중요성

- 의사들은 청진음으로 호흡기 질병을 판단하는데 이를 머신 러닝으로 자동화하고 좀 더 정확한 진단을 목표로 하였다.
- 또한 이 모델을 가지고 질환에 대해 중요한 진단을 내릴 때 참고가 될 수 있다.
- 데이터가 많지는 않지만 클래스가 7가지로 다양하여 선택

3. 모델학습 프로젝트 flowchart



4. 데이터 가공 기술

- 주파수와 진폭이라는 소리의 독립적인 속성을 이용하여 오디오 데이터를 다각적으로 분석.
- 스펙트로그램: 시간 축과 주파수 축의 변화에 따라 진폭의 차이를 인쇄 농도 / 표시 색상의 차이로 나타낸다.->주파수 크면 헤르츠 값도 크다.
- 진폭이란? - 소리의 강세를 의미한다. 진폭의 절대값이 크면 소리가 크고 작으면 소리가 작다는 것을 뜻한다. 진폭이 양수이면 소리는 양의 방향으로, 진폭이 음수이면 소리는 음의 방향으로 난다는 의미이다.
- audio_signal, audio_sample_rate =librosa.load(): 소리의 진폭을 구하는 함수. 진폭의 변화를 2차원 그래프의 점으로 표현하고 분포의 모양새를 통해 오디오 데이터 식별.
- 샘플링 속도는 44100이므로 1초에 가지고 있는 데이터의 수는 44100이다. 800개의 간격을 주고 x,y 점을 찍을 것.

5. 데이터 추출 소스1 (진폭)

◆ 이미지 생성에 앞서 라벨링에 쓸 데이터 처리

```
import pandas as pd

df = pd.read_csv("patient_diagnosis.csv")
p_diagnosis = {}
print(df.shape[0])

for i in range(df.shape[0]):
    key = df.iloc[i,0 ]
    value = df.iloc[i,1]
    p_diagnosis[key] = value

print(p_diagnosis)
diagnosis=p_diagnosis[int("109.png"[0:3])]
print(diagnosis)
```

```
126
{101: 'URTI', 102: 'Healthy', 103: 'Asthma', 104: 'COPD', 105: 'URTI', 106: 'COPD', 107: 'COPD', 108: 'LRTI', 109: 'COPD', 110: 'COPD',
111: 'Bronchiectasis', 112: 'COPD', 113: 'COPD', 114: 'COPD', 115: 'LRTI', 116: 'Bronchiectasis', 117: 'COPD', 118: 'COPD', 119: 'URT
I', 120: 'COPD', 121: 'Healthy', 122: 'Pneumonia', 123: 'Healthy', 124: 'COPD', 125: 'Healthy', 126: 'Healthy', 127: 'Healthy', 128: 'C
OPD', 129: 'URTI', 130: 'COPD', 131: 'URTI', 132: 'COPD', 133: 'COPD', 134: 'COPD', 135: 'Pneumonia', 136: 'Healthy', 137: 'URTI', 138:
'COPD', 139: 'COPD', 140: 'Pneumonia', 141: 'COPD', 142: 'COPD', 143: 'Healthy', 144: 'Healthy', 145: 'COPD', 146: 'COPD', 147: 'COPD',
148: 'URTI', 149: 'Bronchiolitis', 150: 'URTI', 151: 'COPD', 152: 'Healthy', 153: 'Healthy', 154: 'COPD', 155: 'COPD', 156: 'COPD', 15
7: 'COPD', 158: 'COPD', 159: 'Healthy', 160: 'COPD', 161: 'Bronchiolitis', 162: 'COPD', 163: 'COPD', 164: 'URTI', 165: 'URTI', 166: 'CO
PD', 167: 'Bronchiolitis', 168: 'Bronchiectasis', 169: 'Bronchiectasis', 170: 'COPD', 171: 'Healthy', 172: 'COPD', 173: 'Bronchioliti
s', 174: 'COPD', 175: 'COPD', 176: 'COPD', 177: 'COPD', 178: 'COPD', 179: 'Healthy', 180: 'COPD', 181: 'COPD', 182: 'Healthy', 183: 'He
althy', 184: 'Healthy', 185: 'COPD', 186: 'COPD', 187: 'Healthy', 188: 'URTI', 189: 'COPD', 190: 'URTI', 191: 'Pneumonia', 192: 'COPD',
193: 'COPD', 194: 'Healthy', 195: 'COPD', 196: 'Bronchiectasis', 197: 'URTI', 198: 'COPD', 199: 'COPD', 200: 'COPD', 201: 'Bronchiectas
is', 202: 'Healthy', 203: 'COPD', 204: 'COPD', 205: 'COPD', 206: 'Bronchiolitis', 207: 'COPD', 208: 'Healthy', 209: 'Healthy', 210: 'UR
TI', 211: 'COPD', 212: 'COPD', 213: 'COPD', 214: 'Healthy', 215: 'Bronchiectasis', 216: 'Bronchiolitis', 217: 'Healthy', 218: 'COPD', 2
19: 'Pneumonia', 220: 'COPD', 221: 'COPD', 222: 'COPD', 223: 'COPD', 224: 'Healthy', 225: 'Healthy', 226: 'Pneumonia'}
COPD
```

5. 데이터 추출 소스1 (진폭)

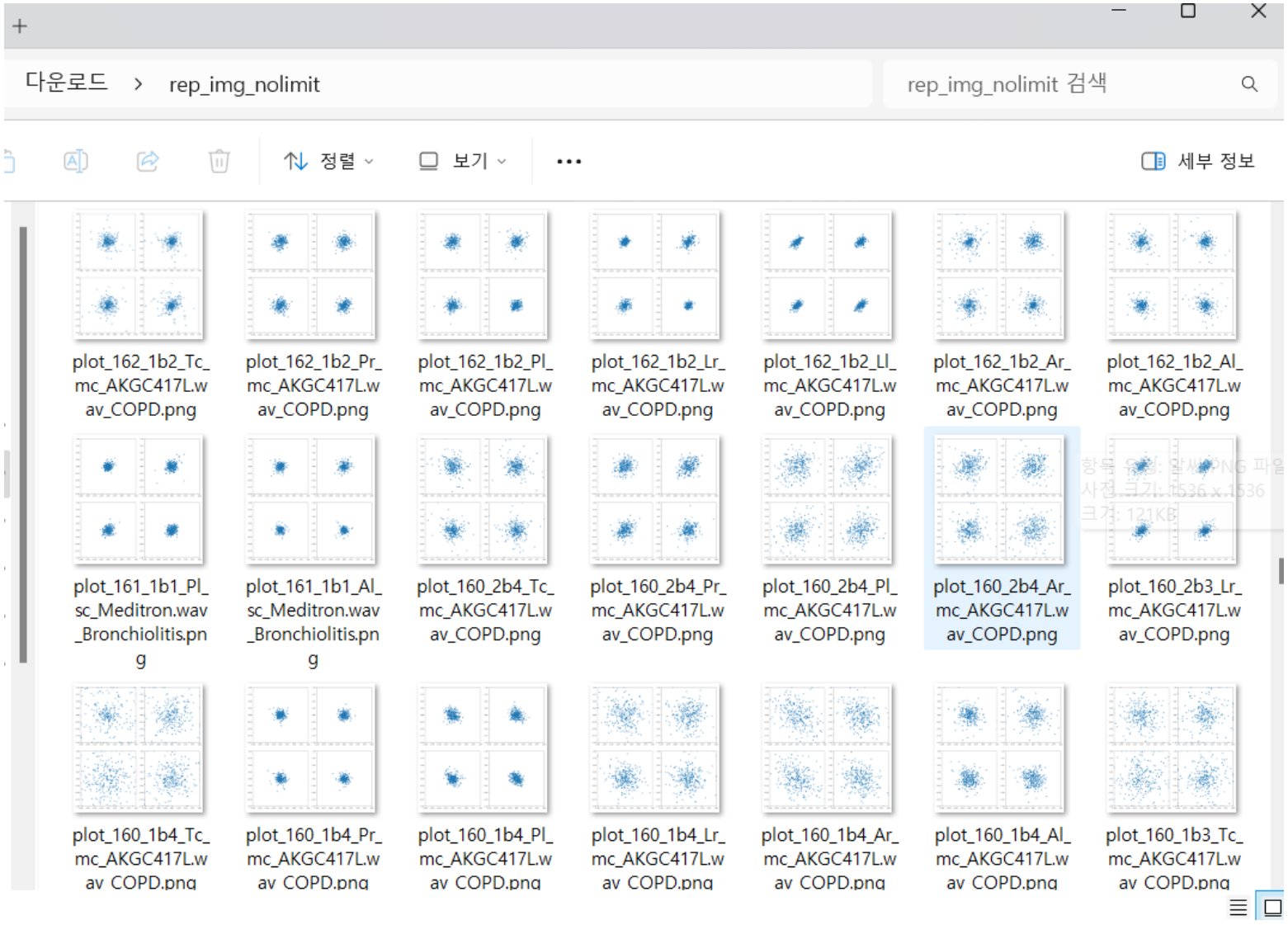
◆ 이미지 생성 코드

In [6]:
```
import os
import librosa
import numpy as np
import matplotlib.pyplot as plt
import librosa.display

path = 'C:WWUsersWWcyj42WWDownloadsWWrep_soundWW'
file_list = os.listdir(path)
file_list_py = [file for file in file_list if file.endswith('.wav')]

print("file_list_py len:", len(file_list_py))
k=0
for i in file_list_py:
    audio_path = path + i
    audio_signal, audio_sample_rate = librosa.load(audio_path, sr=None)
    k +=1

    print(k,"번째 파일")

    diagnosis = p_diagnosis[int(i[0:3])]

    plt.figure(figsize=(12, 12))
    index = 0
```

5. 데이터 추출 소스1 (진폭)

```
    for j in range(4):
        plt.subplot(2, 2, j+1, aspect='equal') #사이즈를 정사각형으로 맞춰준다.
        signal = audio_signal[index:index + int(5 * audio_sample_rate)]
        index += int(5 * audio_sample_rate)

        signalX = signal[:-800:800]
        signalY = signal[800::800]
        #점의위치정보도 중요할 수 있으므로 상대적으로 어디 찍히는지 보이도록 구간을 제한한다.
        plt.xlim([-1, 1])
        plt.ylim([-1, 1])

        plt.scatter(signalX, signalY, s=30)


    plt.tight_layout()
    #plt.savefig(f"C:\Users\cyj42\Downloads\rep_img_nolimit\plot_{i}_{diagnosis}.png", dpi=128)

    plt.show()
```

5. 데이터 추출 소스1 - 추출 결과를 모은 파일



5. 데이터 추출 소스1 - 결과

- 7가지 증상에 따른 청진음 그래프 특징

1) heathy

건강한 사람의 청진음에서는 별

다른 특징적인 소리가 들리지

않고 조용한 편이다.

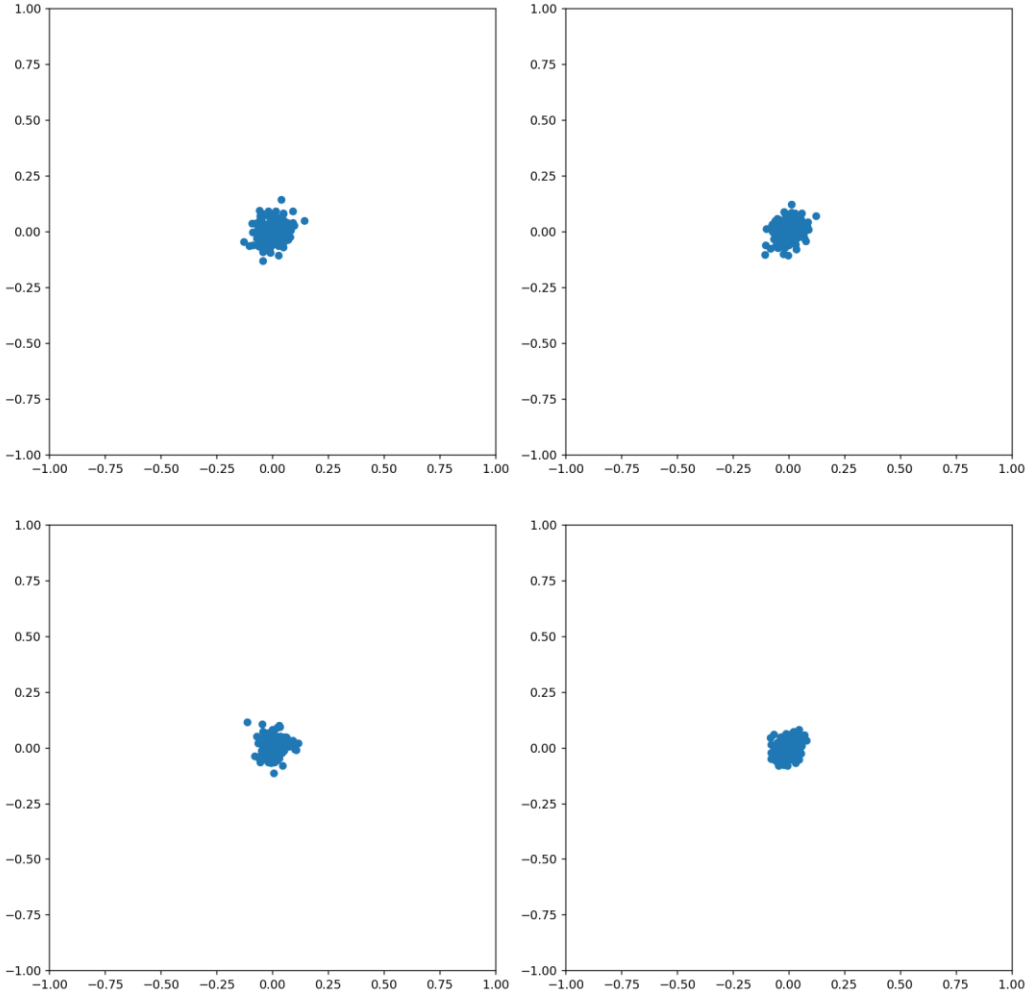
또 숨소리가 일관되고 조용한

특징 때문에 그래프로 진폭의

변화를 찍어 보면 진폭 값의 절

대값이 낮기에 가운데 (0,0)지점

주위로 많이 찍히는 것을 알 수 있다.



5. 데이터 추출 소스1 - 결과

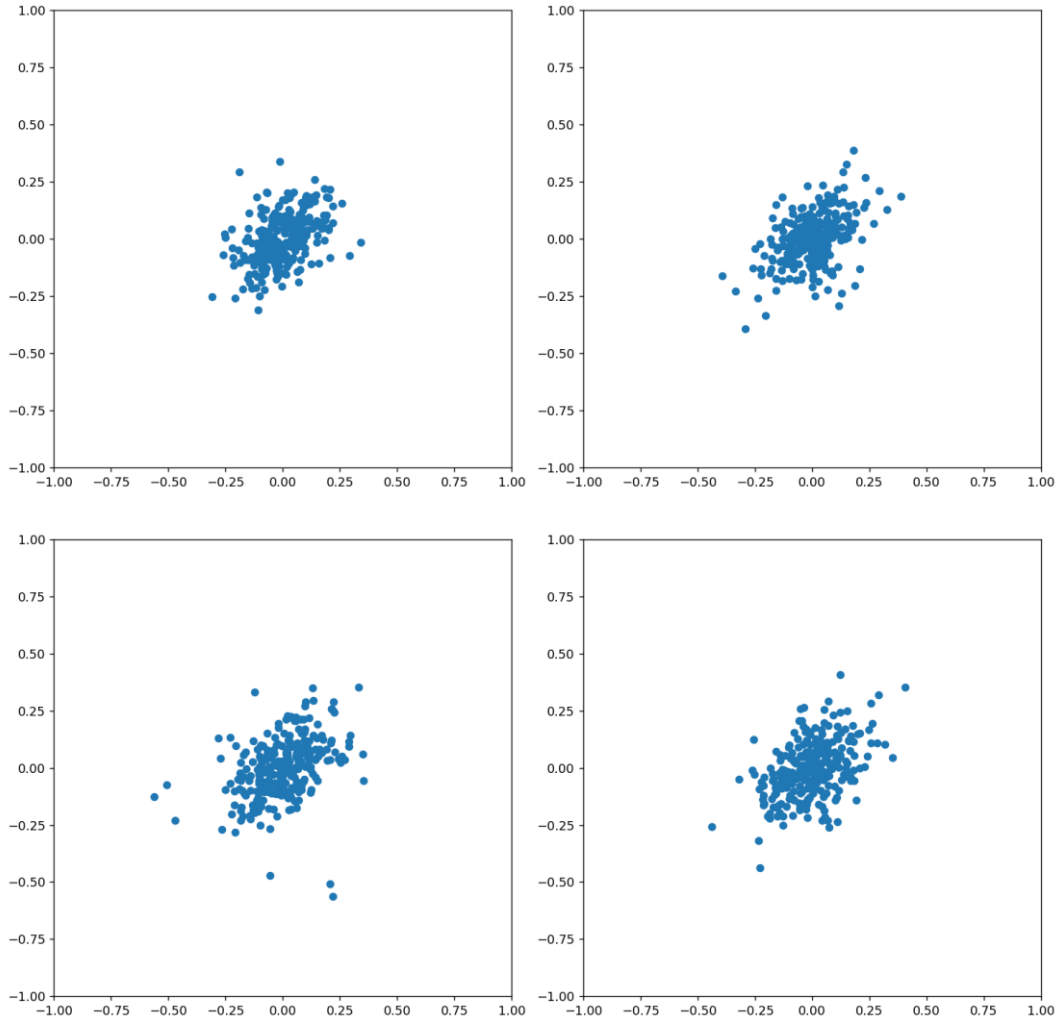
- 7가지 증상에 따른 청진음 그래프 특징

2) 폐렴(Pneumonia)

폐렴은 폐에 염증이 생겨 호흡에 장애가 생기는 질병이다.

폐에 염증이나 분비물 등이 폐포에 끼어서 나는 거친 소리가 특징적이다. 거친 소리가 간헐적으로 나기 때문에 소리의 강도가 갑자기 커지는 부분이 있다.

이것은 소리 세기의 변화로 연결되어 건강한 사람의 호흡보다 원점 주위에서 멀리 떨어져서 점이 찍히는 것을 알수 있다.



5. 데이터 추출 소스1 - 결과

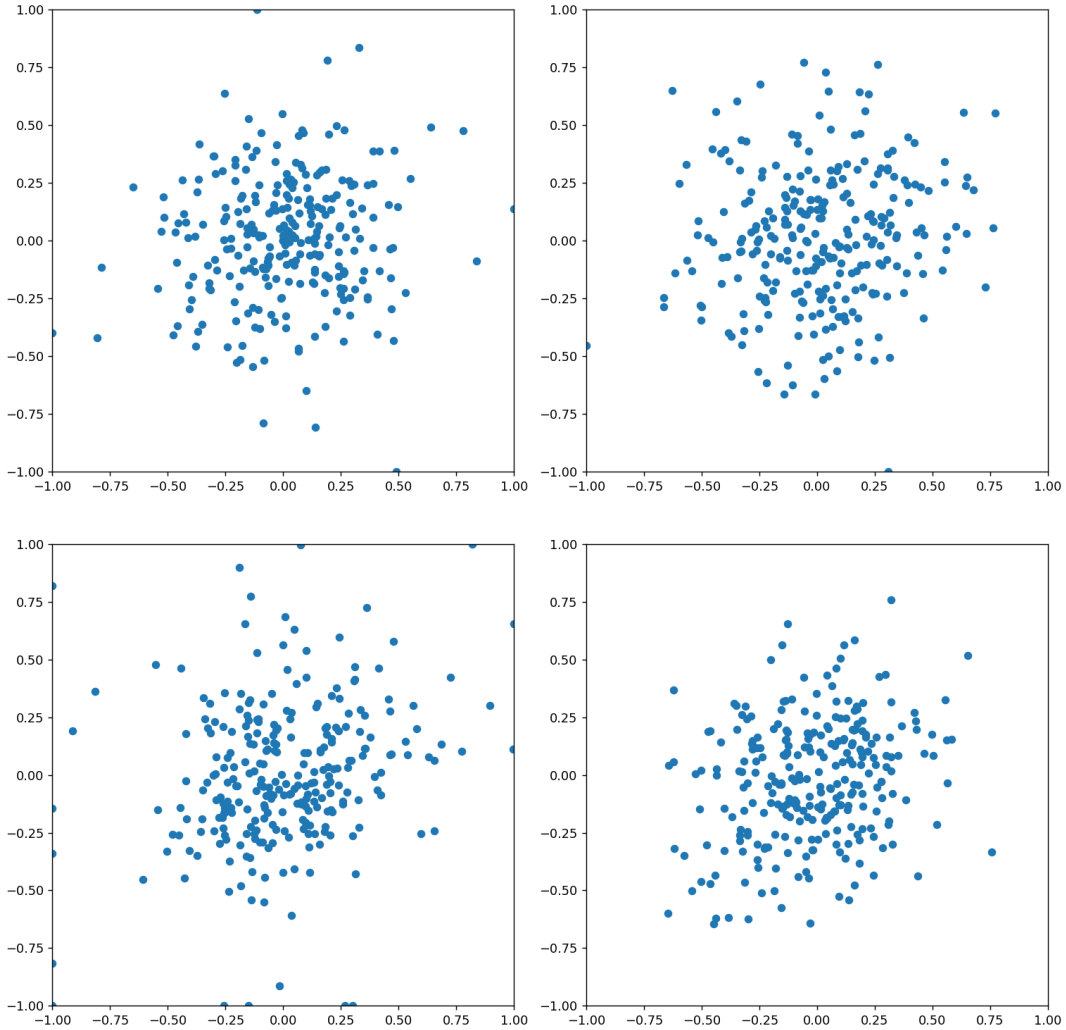
- 7가지 증상에 따른 청진음 그래프 특징

3) 만성폐쇄성폐질환(COPD)

만성폐쇄성폐질환이란 담배, 대기오염 또는 독성흡입물질에 의해 기도에 염증이 지속돼 기도가 좁아지면서 서서히 기도폐쇄가 일어나는 질환이다.

COPD 환자에게서는 폐쇄음이 난다.

-폐쇄음이란 기도의 부분적인 폐쇄로 인해 흡기 및 내기의 호흡 소리에 변화가 나타나는 것이다. 기도에서 나는 소리로, 고르지 않고 불규칙한 특징 때문에 점이 매우 산발적으로 찍히는 것을 알 수 있다.

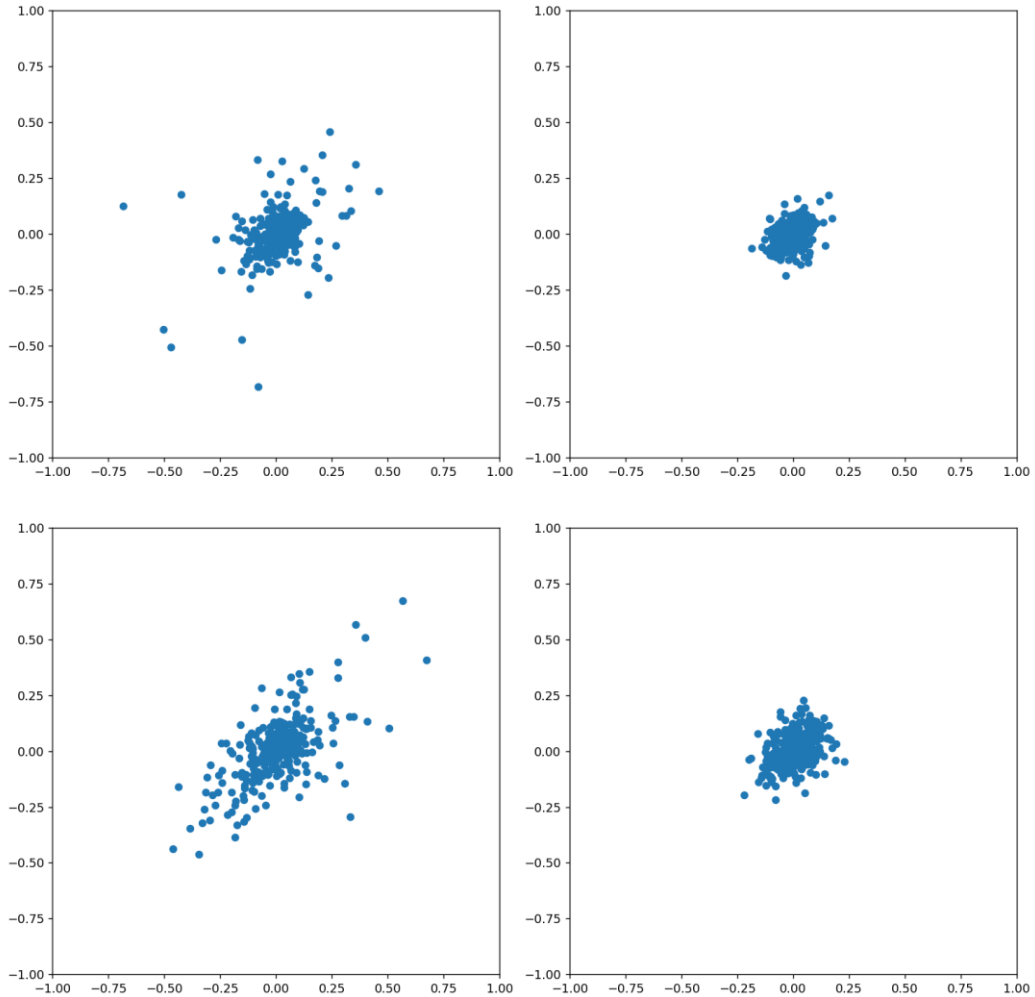


5. 데이터 추출 소스1 - 결과

- 7가지 증상에 따른 청진음 그래프 특징

4) LRTI

하기도감염(영어 약자로 LRTI)은 폐렴의 동의어로 자주 쓰이나, 폐농양, 급성 기관지염 등 다른 질병들에도 적용될 수 있는 용어이다.
하부호흡기감염증이라 부르기도 하고 호흡곤란이 특징이다.
일종의 폐렴이기에 폐의 염증으로부터 나는 소리가 특징이다. 폐에 분비물이 쌓여 호흡 소리가 거칠게 나기 때문에 소리의 강도가 갑자기 커지는 부분이 있어 건강한 사람의 그래프 양산에 비해 원점 주위로 산발적인 모습을 볼 수 있다.



5. 데이터 추출 소스1 - 결과

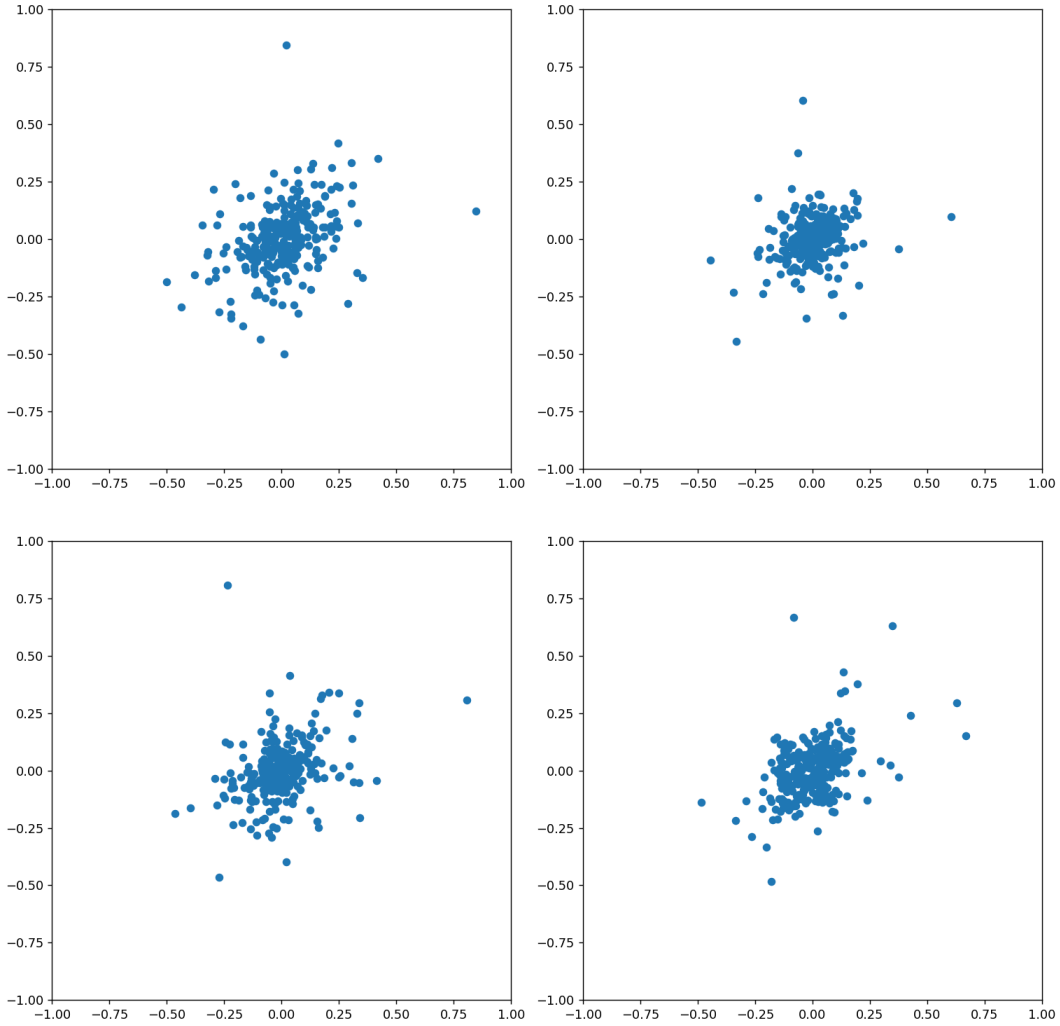
- 7가지 증상에 따른 청진음 그래프 특징

5) URTI

상기도 감염(영어 약자로 URTI)은 감기와 동의어로 쓰인다.

특징적인 소리로는 코막힘과 콧물로, 코와 기도를 통해 들려오는 소리다.

숨 쉴때 코에 분비물이 쌓이거나 기도가 감염되면 불규칙적인 소리가 나게 하는 원인이 되어 소리의 강세가 커졌다 작아졌다 하기 때문에 점이 건강한 사람에 비해 산발적으로 찍히는 것을 알 수 있다.



5. 데이터 추출 소스1 - 결과

- 7가지 증상에 따른 청진음 그래프 특징

6) asthma

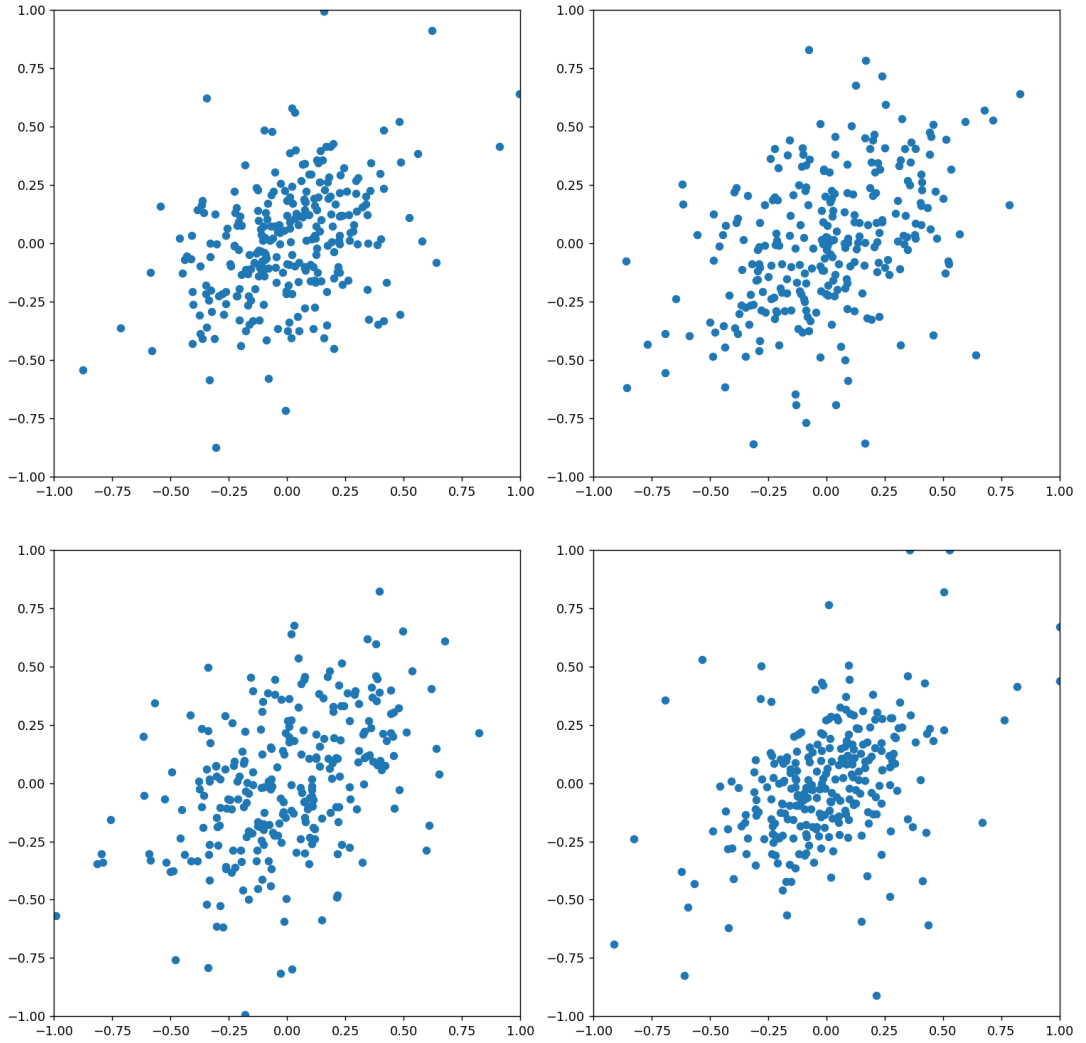
천식(Asthma)은 기도의 염증으로 인해 기도가 좁아지고 수축되면서 호흡곤란을 유발하는 만성 호흡기 질환이다.

천식 환자 청진음의 특이한 점은 천명이다.

-천명은 기도가 좁아지고 수축될 때, 공기가 제한되어 통과하면서 발생하는 고음의 소리다.

소리의 강세로 설명해 보면 소리가 커졌다 작아졌다 하는 특징이 두드러진다는 점이다.

따라서 점은 원점 주위로 산발적이게 찍힌다.

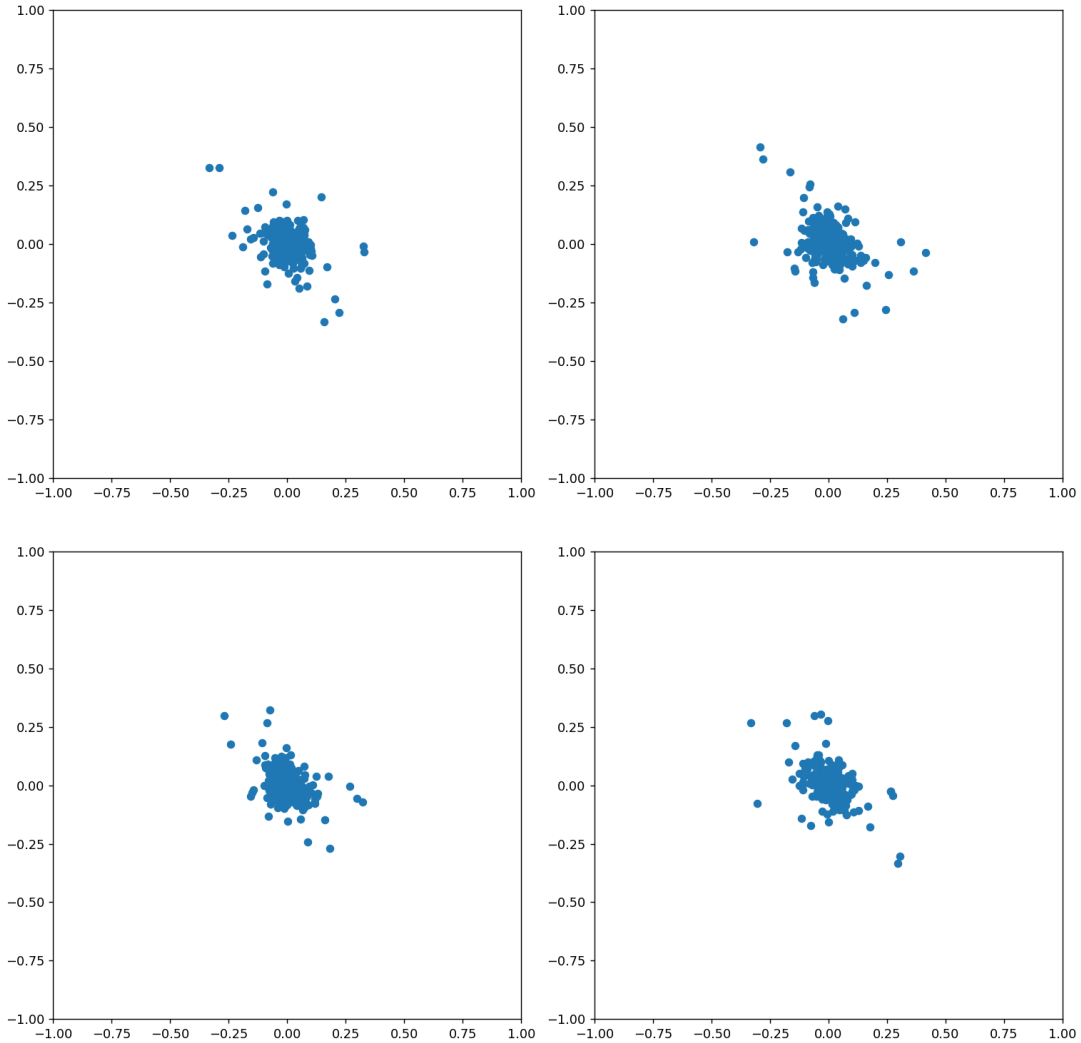


5. 데이터 추출 소스1 - 결과

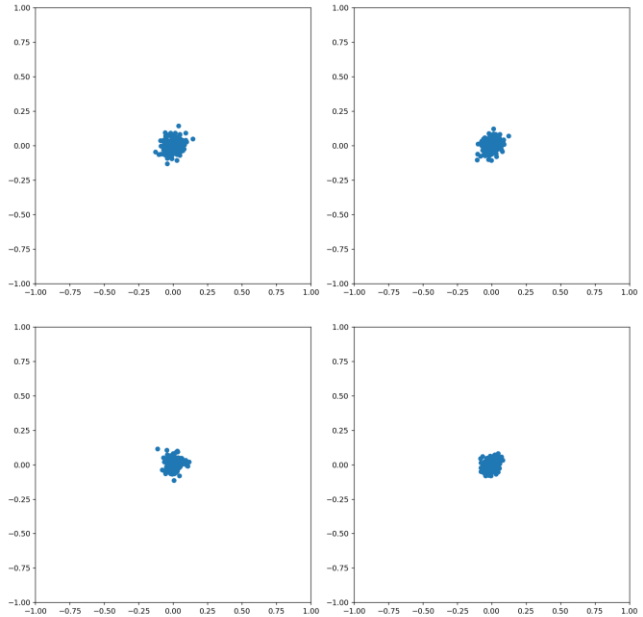
- 7가지 증상에 따른 청진음 그래프 특징

7) Bronchiolitis

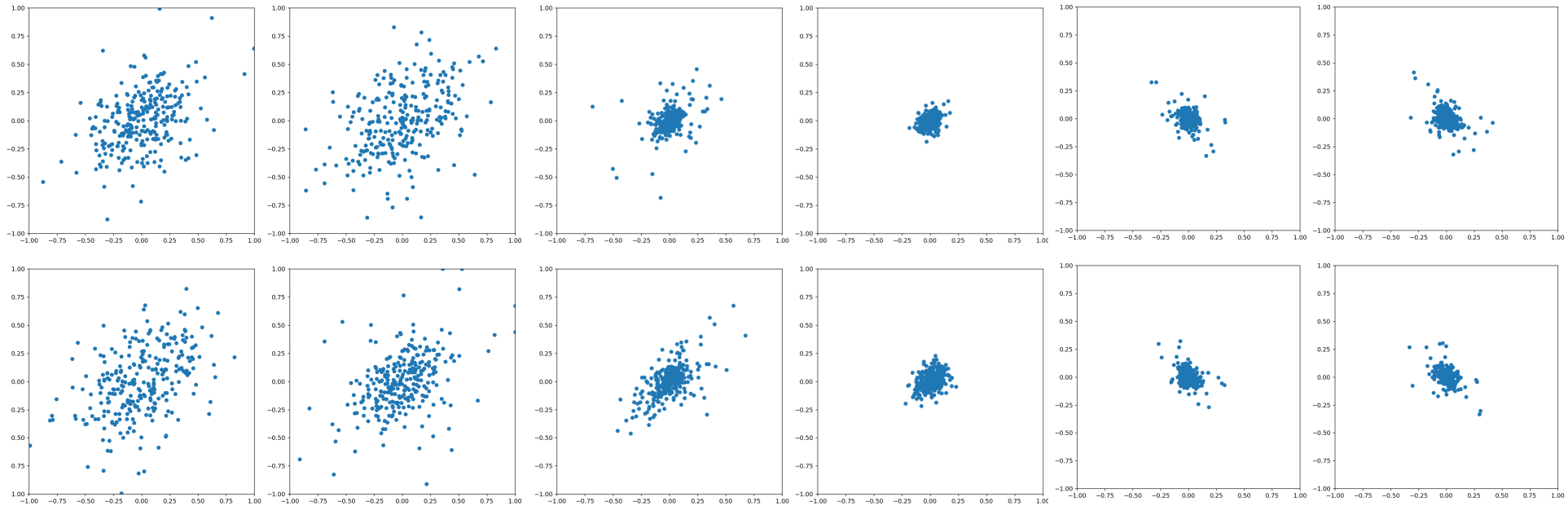
세기관지염(Bronchitis)은 기관지의 염증으로 인한 호흡기 질환으로, 급성 세기관지염과 만성 세기관지염이 있다.
각각의 세기관지염에 따라 청진음의 특징이 다를 수 있지만, 일반적인 특징으로는 기관지 소리와 건성수포음을 들 수 있다.
기관지염은 기관지의 염증으로 인해 기관지에서 특이한 소리가 나고, 건성 수포음은 기도 안에 가래, 혈액 등과 같은 이물질로 인해 호흡을 함에 따라 기도 벽으로부터 떨어지거나 움직여서 생기는 소리다. 딱딱거리는 음이 가끔 나기에 원점 주위로 종종 떨어져 찍힌 점들을 확인 할 수 있다.



정상

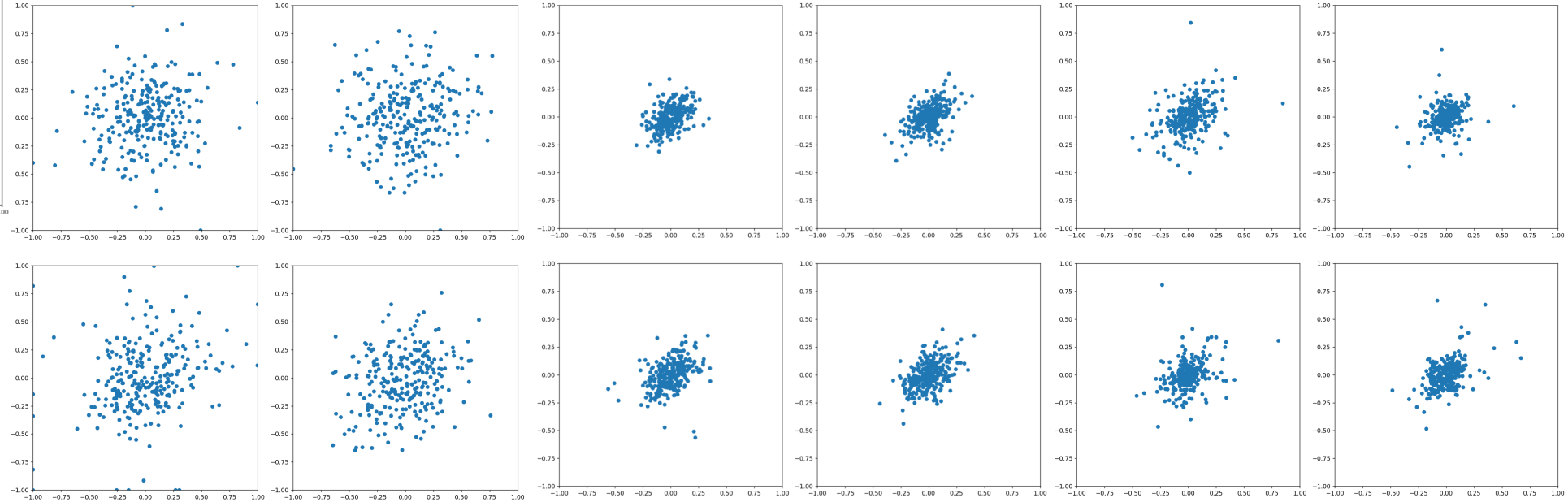


천식



LRTI

세기관지염



COPD

폐렴

URTI

5. 데이터 추출 소스2 (스펙트로그램)

◆ 스펙트로그램: 시간에 따른 주파수의 변화를 시각적으로 표현한 것.

◆ 주파수: 초당 진동의 횟수로 표현되며, 주로 헤르츠 단위로 측정. 주파수가 높을수록 고음이며, 주파수가 낮을수록 저음을 뜻한다.

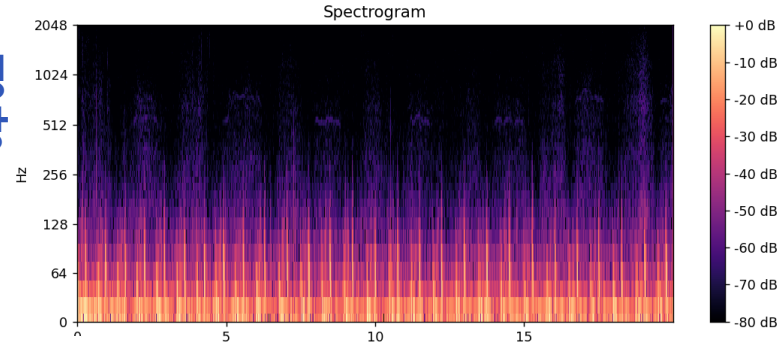
◆ 모델 학습에 직접적으로 사용은 안하고, 오디오 데이터를 추가적으로 설명하는 용도로 사용.

```python
import os
import librosa
import numpy as np
import matplotlib.pyplot as plt
import librosa.display

path = 'C:₩₩Users₩₩cyj42₩₩Downloads₩₩rep_sound₩₩'
file_list = os.listdir(path)
file_list_py = [file for file in file_list if file.endswith('.wav')]

print("file_list_py len:", len(file_list_py))
for i in file_list_py:
    audio_path = path + i
    diagnosis = p_diagnosis[int(i[0:3])]
    print(diagnosis)
    audio_signal, audio_sample_rate = librosa.load(audio_path, sr=None)

    D = librosa.amplitude_to_db(np.abs(librosa.stft(audio_signal)), ref=np.max)

    plt.figure(figsize=(10, 4))
    librosa.display.specshow(D, sr=audio_sample_rate, x_axis='time', y_axis='log')
    plt.colorbar(format='%+2.0f dB')

    #y축 상한을 2048로 설정해서 더 그래프를 잘 식별 가능하게 했다.
    plt.ylim([0, 2048])

    plt.title('Spectrogram')
    plt.savefig(f"C:₩₩Users₩₩cyj42₩₩Downloads₩₩rep_img_specta2₩₩plot_{i}_{diagnosis}.png", dpi=128)
    plt.show()
```

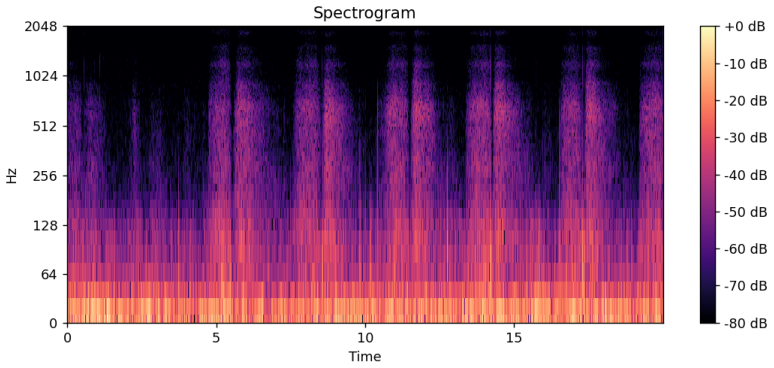
5. 데이터 추출 소스2 - 결과

- 7가지 증상의 스펙트로그램 그래프 (음의 높낮이에 주목)

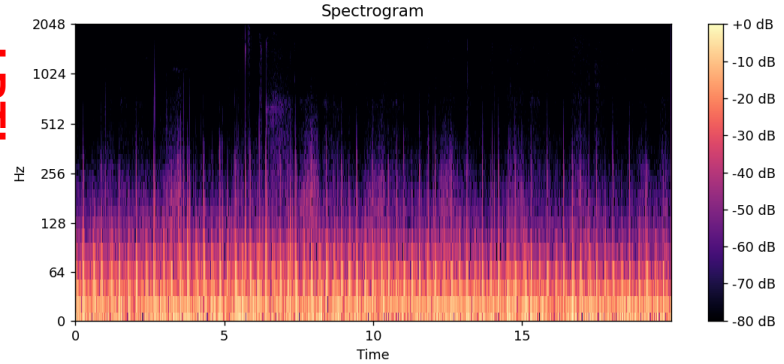
정상



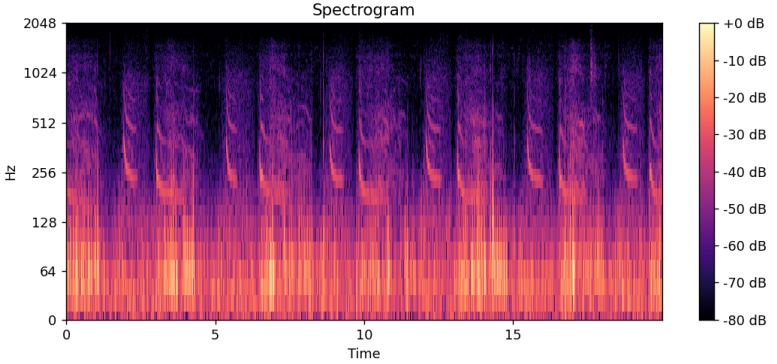
폐렴



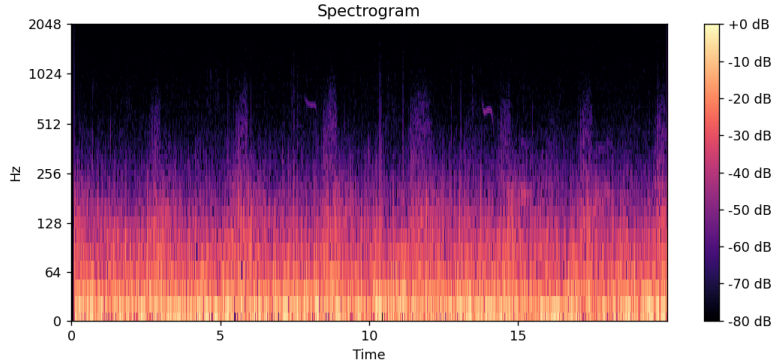
LRTI



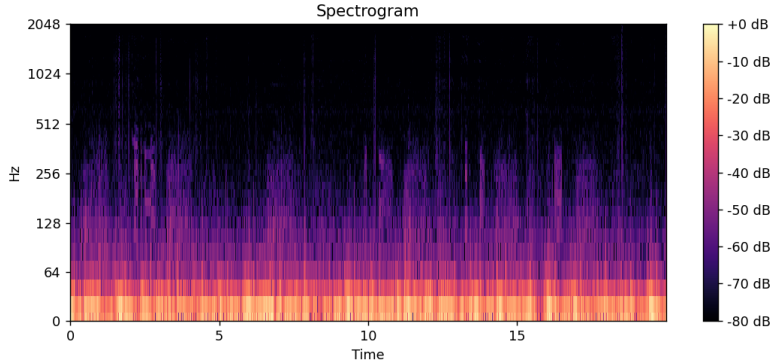
세기관지염



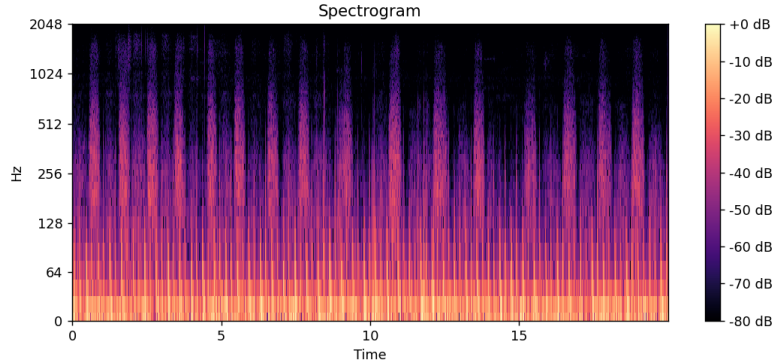
COPD



천식



URTI



6. 데이터 추출1 학습

◆ X에 이미지, Y에 라벨값을 저장하는 코드

In [8]:

```
import cv2
path = 'C:₩₩Users₩₩cyj42₩₩Downloads₩₩rep_img_nolimit₩₩'
file_list = os.listdir(path)
file_list_py = [file for file in file_list if file.endswith('.png')]

print("file_list_py len:", len(file_list_py))

image_w = 128
image_h = 128
num_classes = len(file_list_py)

X = []  # to store images
Y = []  # to store labels
for i in file_list_py:
    print(path+i)
    img = cv2.imread(path+i)
    img = cv2.resize(img, (image_w, image_h))
    img = img / 255.0
    label=i[32:36]
    X.append(img)
    Y.append(label)
    #plt.imshow(img)
    #plt.show()

X = np.array(X)
print(X)
print(X.shape)

Y = np.array(Y)
print(Y)
print(Y.shape)
```

6. 데이터 추출1 학습

◆원 핫 인코딩을 위해 Y리스트의 값을 int로 변환해주고,

◆그 결과와 X를 train set과 test set으로 나눈다.

In [9]:
```
vy_list = []
for i in Y:
    if i == 'Heal':
        vy_list.append(0)
    elif i == 'URTI':
        vy_list.append(1)
    elif i == 'Asth':
        vy_list.append(2)
    elif i == 'COPD':
        vy_list.append(3)
    elif i == 'LRTI':
        vy_list.append(4)
    elif i == 'Bron':
        vy_list.append(5)
    elif i == 'Pneu':
        vy_list.append(6)

vy_list = np.array(vy_list)

from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test = train_test_split(X, vy_list, test_size=0.1, random_state=42)

from keras.utils import to_categorical
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
print(y_test[0])
```

6. 데이터 추출1 학습의 결과

◆ CNN모델로 앞서 전처리한 이미지들을 학습시킨다.

◆ 다중 분류이므로 softmax함수를 사용한다.

◆ModelCheckpoint, EarlyStopping을 사용해 과적합을 막고 적절한 타이밍에 학습을 종료시킨다.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dropout, Dense, Conv2D, MaxPooling2D, Flatten
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping

model = Sequential()
model.add(Conv2D(32, kernel_size=3, padding='same', activation='relu'))
model.add(Conv2D(64, kernel_size=3, activation='relu'))
model.add(MaxPooling2D(pool_size=2))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(7, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])

modelpath = "best_model.hdf5"
checkpointer = ModelCheckpoint(filepath=modelpath, monitor='val_loss', verbose=1, save_best_only=True)
early_stopping_callback = EarlyStopping(monitor='val_loss', patience=10)

history = model.fit(x_train, y_train, validation_split=0.2, epochs=30, batch_size=200,
                    callbacks=[early_stopping_callback, checkpointer])
```

6. 데이터 추출1 학습의 결과

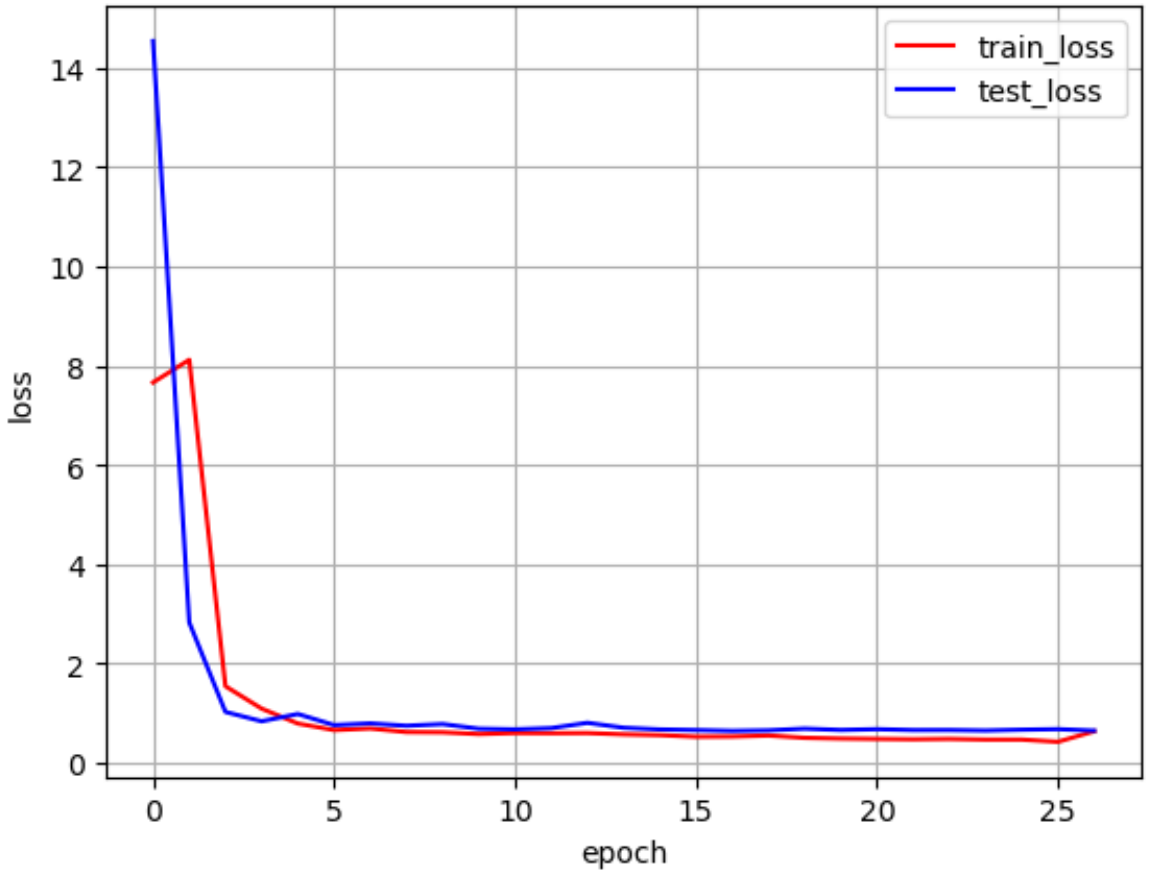
◆ 학습 결과, 86%의 정확도가 나왔다.
- 노이즈나 리버브 제거 작업을 한다면 더 높은 결과가 나올 것으로 생각된다.

```
3/3 [==============================] - ETA: 0s - loss: 0.4839 - accuracy: 0.8591
Epoch 23: val_loss did not improve from 0.64680
3/3 [==============================] - 11s 3s/step - loss: 0.4839 - accuracy: 0.8591 - val_loss: 0.6635 - val_accuracy: 0.7867
Epoch 24/30
3/3 [==============================] - ETA: 0s - loss: 0.4714 - accuracy: 0.8680
Epoch 24: val_loss did not improve from 0.64680
3/3 [==============================] - 12s 3s/step - loss: 0.4714 - accuracy: 0.8680 - val_loss: 0.6534 - val_accuracy: 0.7867
Epoch 25/30
3/3 [==============================] - ETA: 0s - loss: 0.4710 - accuracy: 0.8591
Epoch 25: val_loss did not improve from 0.64680
3/3 [==============================] - 13s 4s/step - loss: 0.4710 - accuracy: 0.8591 - val_loss: 0.6700 - val_accuracy: 0.7867
Epoch 26/30
3/3 [==============================] - ETA: 0s - loss: 0.4259 - accuracy: 0.8658
Epoch 26: val_loss did not improve from 0.64680
3/3 [==============================] - 13s 4s/step - loss: 0.4259 - accuracy: 0.8658 - val_loss: 0.6812 - val_accuracy: 0.7867
Epoch 27/30
3/3 [==============================] - ETA: 0s - loss: 0.6371 - accuracy: 0.8591
Epoch 27: val_loss did not improve from 0.64680
3/3 [==============================] - 12s 3s/step - loss: 0.6371 - accuracy: 0.8591 - val_loss: 0.6512 - val_accuracy: 0.7800
```

```
In [10]: result=model.evaluate(x_test,y_test)
         print("model accuracy:",result[1]*100,"%")

         7/7 [==============================] - 1s 136ms/step - loss: 0.5236 - accuracy: 0.8600
         model accuracy: 86.00000143051147 %
```

6. 데이터 추출1 학습의 결과

◆ 학습 진행중의 train data의 loss, test data의 loss는 다음과 같았다.
- 27 epoch에서 val_loss가 더이상 줄지 않아 학습 중단.



7. 최종 결론

- 음성 데이터의 2차원 그래프화 학습 방법은 기존의 1차원 데이터 속성값들을 학습시키는 것에 비해 사람이 알아보기도 편하고 기계학습시에도 효과적.
- 의학계의 XAI(설명가능한 AI)수요에도 부합.
- 만약 상용화 한다면, 더 체계적인 데이터 수집과 데이터 양을 늘려(한 클래스당 1000개 이상) 정확성을 높일 수 있을 것이다.