Chest X-ray Abnormalities Detection

조원 : 류소리, 최샘이

01-1

주제 및 주제 선정이유

주제

- 의사는 CT, PET scan, MRI , X-ray 와 같은 데이터를 판독하며 환자의 의학적 상태를 진단하고 치료하는데, 흉부 X-ray 의 경우 다른 부위보다 의학적 오진이 생기기 쉬운 부위입니다.
- 생명과 직결된 부위인 흉부 X-ray에서 작은 size의 병변까지 보다 정확하게 식별하고 위치 파악을 할 수 있 다면, 전문의사(방사선과)가 아니더라도 판독에 도움을 줄 수 있을 것입니다.

선정 이유

- 이미지 데이터의 처리 및 분석에 대하여 궁금증이 있었기 때문에 프로젝트를 진행하며 알아보고 싶었습니다.
- 의료 전공 팀원이 있기에 분석 및 결과 해석이 용이 할 것 같아, 의료 이미지를 이용하는 것으로 방향을 잡았습니다.

개발 환경

Tool

Google Colab

language

Python

Library

Scikit-learn
Keras
TensorFlow
etc

I차 분석 목표 및 계획

- ✓ 흉부의 X-ray 이미지만으로 이상 유무를 정확하게 평가 가능한지 알아보려고 한다.

- ✓ 흉부의 X-ray 이미지에서 14가지 유형의 병변과 한가지의 이상 없음을 파악하고 올바른 곳에 바운딩 박스를 그려보려고 한다.

02

Train csv 데이터 분석

Train.csv Data Analysis

[Columns]

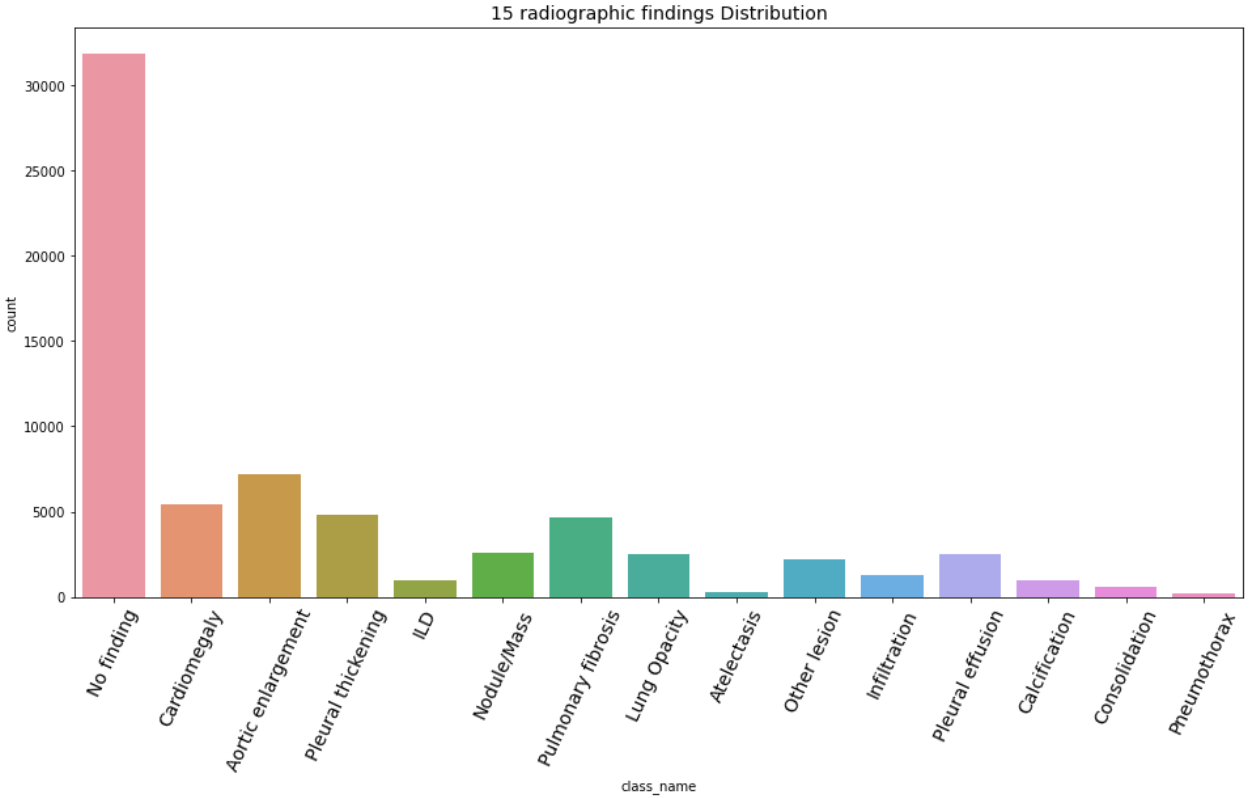
- image_id 이미지와 매칭하기 위한 고유 식별자
- class_name 감지된 개체의 클래스 이름 (혹은 No finding)
- class_id 감지된 개체의 클래스 ID
- rad_id 관찰한 방사선 전문의의 ID
- x_min bounding box의 최소 X좌표
- y_min bounding box의 최소 y좌표
- x_max bounding box의 최대 x좌표
- y_min bounding box의 최대 y좌표

	image_id	class_name	class_id	rad_id	x_min	y_min	x_max	y_max
0	50a418190bc3fb1ef1633bf9678929b3	No finding	14	R11	NaN	NaN	NaN	NaN
1	21a10246a5ec7af151081d0cd6d65dc9	No finding	14	R7	NaN	NaN	NaN	NaN
2	9a5094b2563a1ef3ff50dc5c7ff71345	Cardiomegaly	3	R10	691.0	1375.0	1653.0	1831.0
3	051132a778e61a86eb147c7c6f564dfe	Aortic enlargement	0	R10	1264.0	743.0	1611.0	1019.0
4	063319de25ce7edb9b1c6b8881290140	No finding	14	R10	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
67909	936fd5cff1c058d39817a08f58b72cae	No finding	14	R1	NaN	NaN	NaN	NaN
67910	ca7e72954550eeb610fe22bf0244b7fa	No finding	14	R1	NaN	NaN	NaN	NaN
67911	aa17d5312a0fb4a2939436abca7f9579	No finding	14	R8	NaN	NaN	NaN	NaN
67912	4b56bc6d22b192f075f13231419dfcc8	Cardiomegaly	3	R8	771.0	979.0	1680.0	1311.0
67913	5e272e3adbdaafb07a7e84a9e62b1a4c	No finding	14	R16	NaN	NaN	NaN	NaN

67914 rows × 8 columns

Train.csv Data Analysis

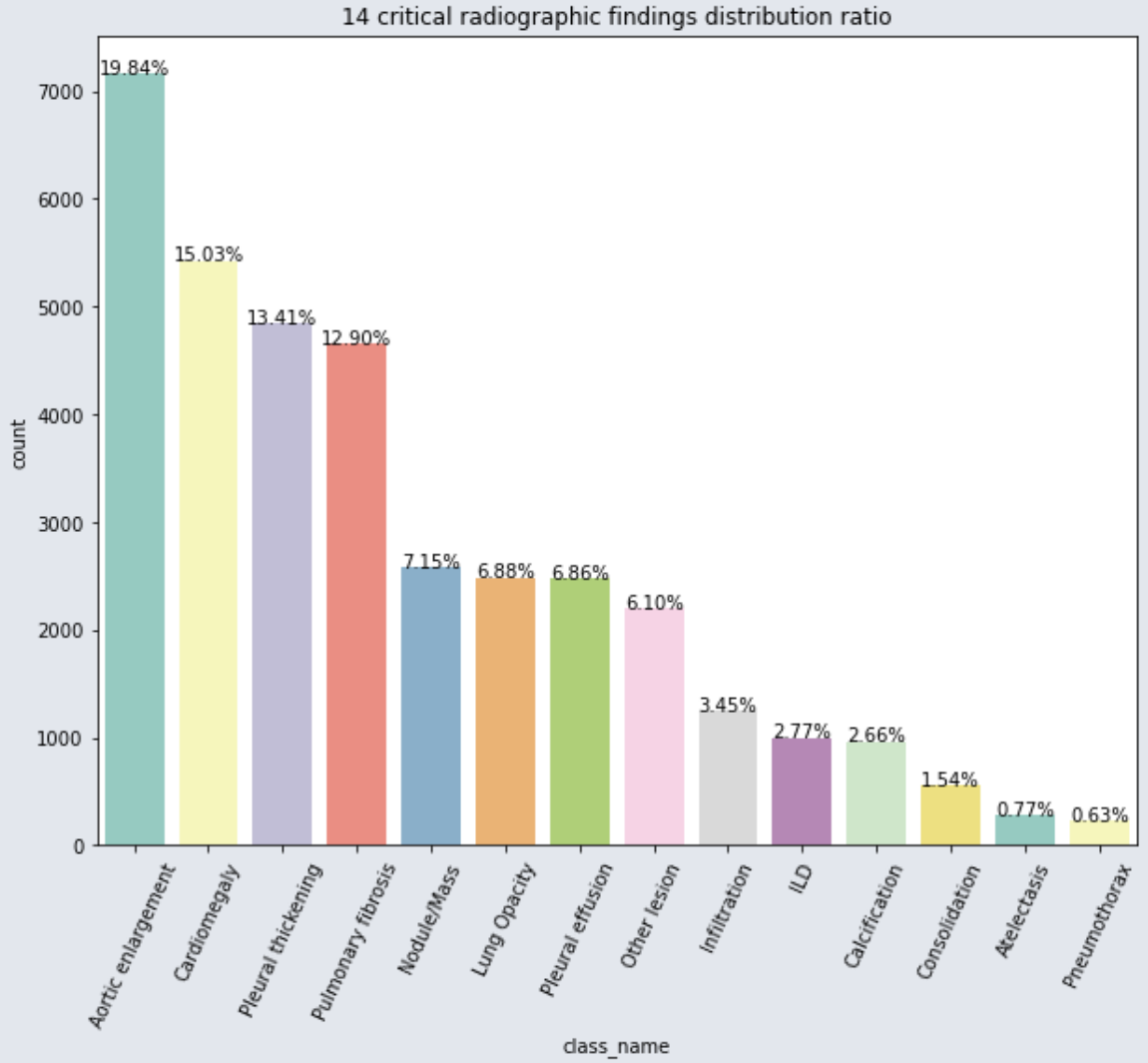
```
No finding              31818
Aortic enlargement       7162
Cardiomegaly             5427
Pleural thickening       4842
Pulmonary fibrosis       4655
Nodule/Mass              2580
Lung Opacity             2483
Pleural effusion         2476
Other lesion             2203
Infiltration             1247
ILD                      1000
Calcification             960
Consolidation             556
Atelectasis               279
Pneumothorax              226
Name: class_name, dtype: int64
```



15가지의 병변 구분에 있고, 데이터 분포 확인.

→ No finding 이 압도적으로 많은 것을 알 수 있습니다.

Train.csv Data Analysis



가장 많이 잡힌 No finding을 제외하고
14가지의 질병 조사 결과만 비율(%)과 함께 확인.

→ Aortic enlargement (대동맥 확장) 19.84%
가장 많았고,

Cardiomegaly (심장 비대) 15.03%

Pleural thickening (흉막 비후) 13.41%
순으로 많은 것을 확인하였습니다.

Train.csv Data Analysis

```
Train Data Size : 67914
Train Data Columns : Index(['image_id', 'class_name', 'class_id', 'rad_id', 'x_min', 'y_min', 'x_max', 'y_max'], dtype='object')
Train Data info :

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 67914 entries, 0 to 67913
Data columns (total 8 columns):
#   Column      Non-Null Count  Dtype
---  ------      --------------  -----
0   image_id    67914 non-null  object
1   class_name  67914 non-null  object
2   class_id    67914 non-null  int64
3   rad_id      67914 non-null  object
4   x_min       36096 non-null  float64
5   y_min       36096 non-null  float64
6   x_max       36096 non-null  float64
7   y_max       36096 non-null  float64
dtypes: float64(4), int64(1), object(3)
```

사이킷런에서는 문자열을 입력 값으로 처리하지 않기 때문에 Dtype의 Object형을 숫자형으로 변환해야 하는 지 살펴 봤으나 Image_id의 경우 이미지 파일과 매칭되는 string값이기 때문에 현재로서는 변환하지 않아도 될 것으로 보이며,
병명 혹은 이상 없음의 이름인 class_name의 경우 int형태의 class_id값으로 구분이 가능하기 때문에 변환하지 않으려 합니다.
방사선 전문의의 ID인 Rad_id의 경우에는 분석방향에 필요 없는 값이라 컬럼을 삭제하려 합니다.

먼저

- 우리가 아는 일반적 이미지 확장자(.jpg .png etc)로 되어 있는 것이 아니라 DICOM이라는 확장자인 것을 확인하였다.
- DICOM은 의료용 디지털 영상 및 통신 표준 확장자로 의료용 기기에서 디지털 영상표현과 통신에 사용되는 여러가지 표준을 통칭하는 확장자이다.
- 이 파일을 다룰 수 있도록 도와주는 Library로는 pydicom, SimpleITK 등이 있는 것으로 보이며 Pydicom의 용례가 가장 많은 것으로 보여 이를 이용하려 한다.

Sample Image

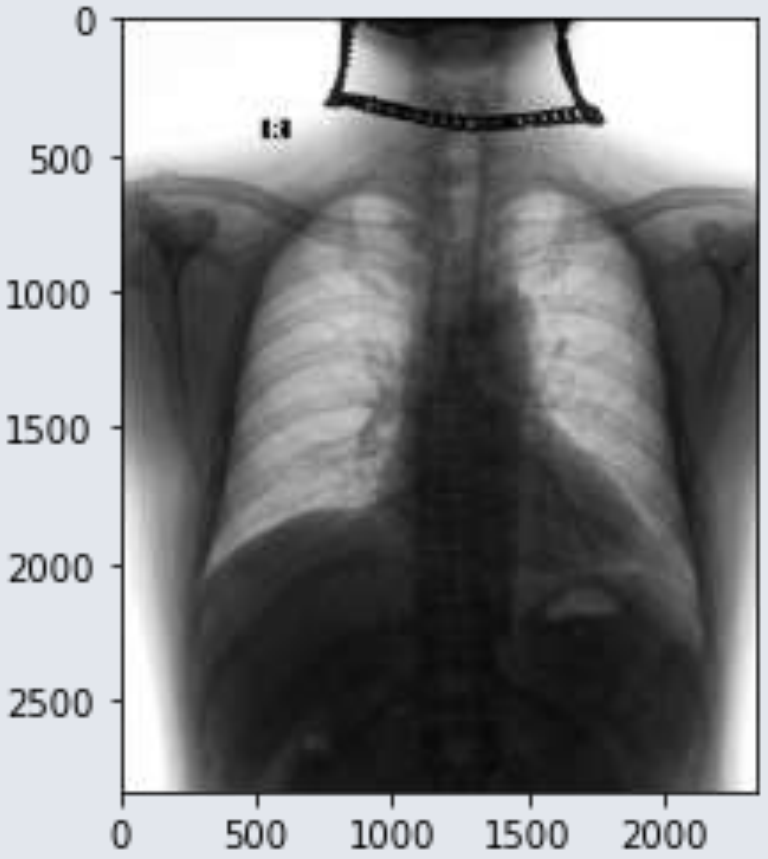
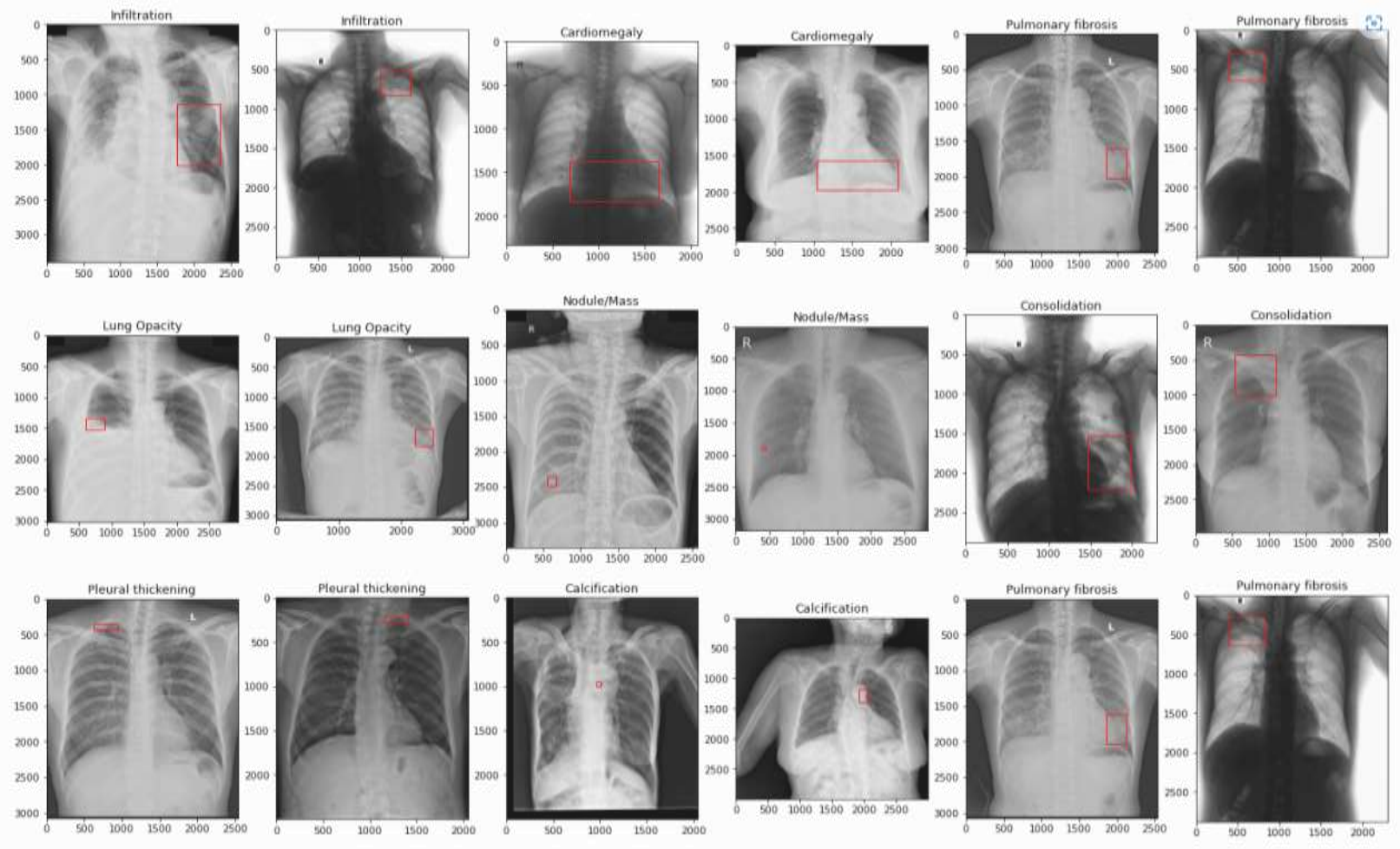


Image 출력 확인 을 위하여
Sample Image Data를 추출해 봤으며

Sample Image



- Image 출력 확인을 위하여 Sample Image Data를 추출.

- 출력이 정상적으로 되는 것으로 보여
  무작위로 Class_id를 골라 2장씩
  총 16장의Sample Image를 추출.

- 그리고 Train Data를 활용하여
  바운딩 박스를 그렸습니다.

→ 올바른 정답 이미지의 모습 확인.
→ 그러나 사진 이미지 크기 다름.

2차 발표 과정

데이터 전처리

Rad_id 칼럼 삭제

img_path 칼럼 추가

Resize

학습 방법 정하기

이중모델

다중 모델

다중모델 **+ bbox**

코드 적용

결과

데이터 전처리

데이터 전처리. 칼럼추가

	image_id	class_name	class_id	x_min	y_min	x_max	y_max	img_path
0	50a418190bc3fb1ef1633bf9678929b3	No finding	14	NaN	NaN	NaN	NaN	/content/drive/MyDrive/과제2/train/50a418190bc...
1	21a10246a5ec7af151081d0cd6d65dc9	No finding	14	NaN	NaN	NaN	NaN	/content/drive/MyDrive/과제2/train/21a10246a5e...
2	9a5094b2563a1ef3ff50dc5c7ff71345	Cardiomegaly	3	691.0	1375.0	1653.0	1831.0	/content/drive/MyDrive/과제2/train/9a5094b2563...
3	051132a778e61a86eb147c7c6f564dfe	Aortic enlargement	0	1264.0	743.0	1611.0	1019.0	/content/drive/MyDrive/과제2/train/051132a778e...
4	063319de25ce7edb9b1c6b8881290140	No finding	14	NaN	NaN	NaN	NaN	/content/drive/MyDrive/과제2/train/063319de25c...
...	...	...	...	...	...	...	...	...
67909	936fd5cff1c058d39817a08f58b72cae	No finding	14	NaN	NaN	NaN	NaN	/content/drive/MyDrive/과제2/train/936fd5cff1c...
67910	ca7e72954550eeb610fe22bf0244b7fa	No finding	14	NaN	NaN	NaN	NaN	/content/drive/MyDrive/과제2/train/ca7e7295455...
67911	aa17d5312a0fb4a2939436abca7f9579	No finding	14	NaN	NaN	NaN	NaN	/content/drive/MyDrive/과제2/train/aa17d5312a0...
67912	4b56bc6d22b192f075f13231419dfcc8	Cardiomegaly	3	771.0	979.0	1680.0	1311.0	/content/drive/MyDrive/과제2/train/4b56bc6d22b...
67913	5e272e3adbdaafb07a7e84a9e62b1a4c	No finding	14	NaN	NaN	NaN	NaN	/content/drive/MyDrive/과제2/train/5e272e3adbd...

67914 rows × 8 columns

→ X_featers값으로 이미지 경로를 가지고 오고 싶어서 **img_path** 칼럼 추가함.

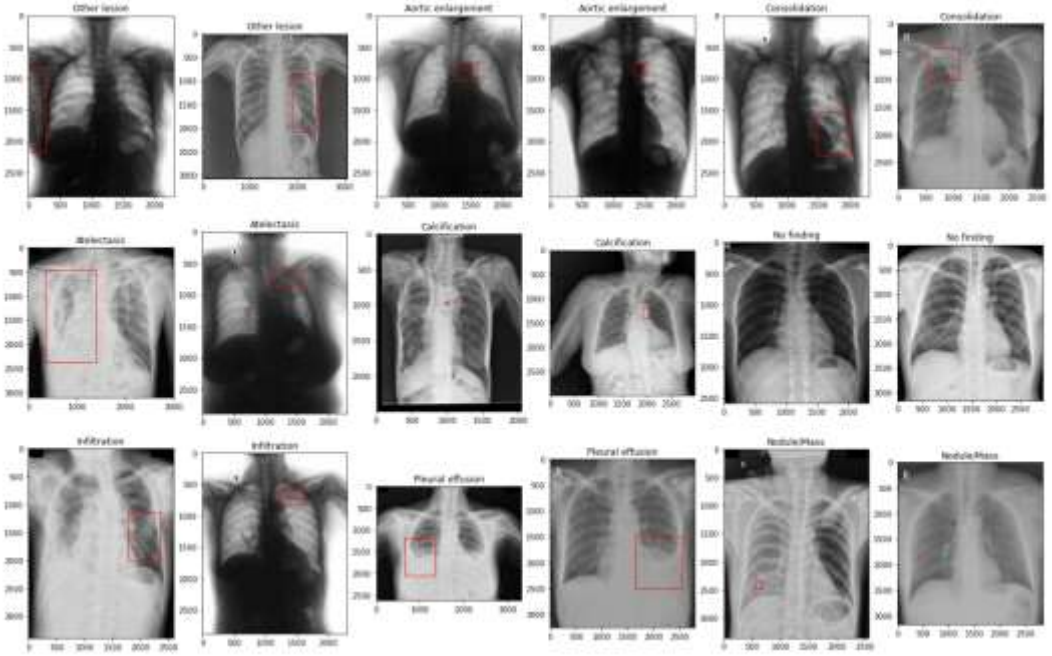
데이터 전처리. **Nan** 값 변경

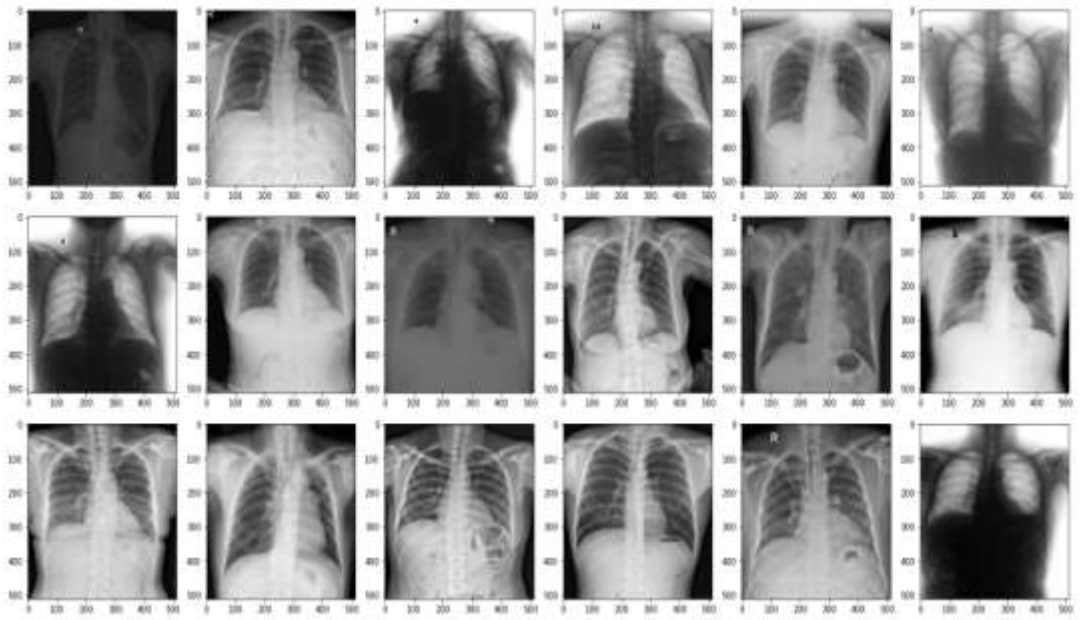
	image_id	class_name	class_id	x_min	y_min	x_max	y_max	img_path
0	50a418190bc3fb1ef1633bf9678929b3	No finding	14	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/50a418190bc...
1	21a10246a5ec7af151081d0cd6d65dc9	No finding	14	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/21a10246a5e...
2	9a5094b2563a1ef3ff50dc5c7ff71345	Cardiomegaly	3	691.0	1375.0	1653.0	1831.0	/content/drive/MyDrive/과제2/train/9a5094b2563...
3	051132a778e61a86eb147c7c6f564dfe	Aortic enlargement	0	1264.0	743.0	1611.0	1019.0	/content/drive/MyDrive/과제2/train/051132a778e...
4	063319de25ce7edb9b1c6b8881290140	No finding	14	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/063319de25c...
...	...	...	...	...	...	...	...	...
67909	936fd5cff1c058d39817a08f58b72cae	No finding	14	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/936fd5cff1c...
67910	ca7e72954550eeb610fe22bf0244b7fa	No finding	14	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/ca7e7295455...
67911	aa17d5312a0fb4a2939436abca7f9579	No finding	14	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/aa17d5312a0...
67912	4b56bc6d22b192f075f13231419dfcc8	Cardiomegaly	3	771.0	979.0	1680.0	1311.0	/content/drive/MyDrive/과제2/train/4b56bc6d22b...
67913	5e272e3adbdaafb07a7e84a9e62b1a4c	No finding	14	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/5e272e3adbd...

67914 rows × 8 columns

→ Nan값 0.1로 바꿔 주었다.

데이터 전처리. **Resize**

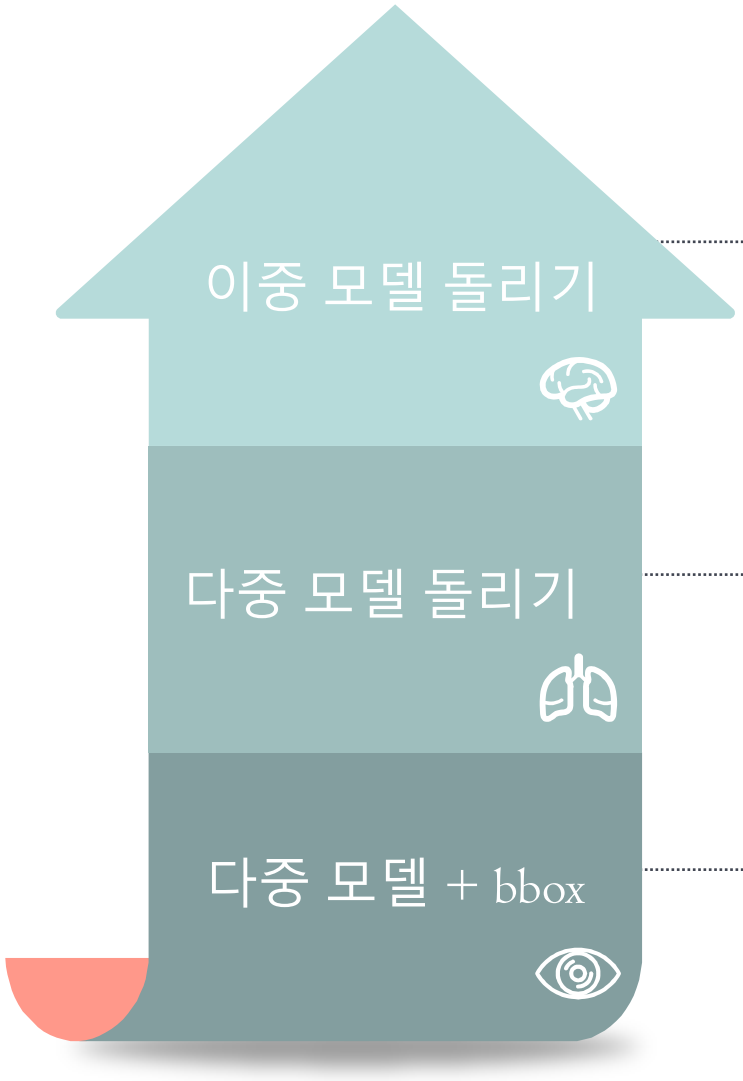




이미지 사이즈를 맞춰주었다.

학습 방법 정하기 & 코드 적용

학습 방법 정하기

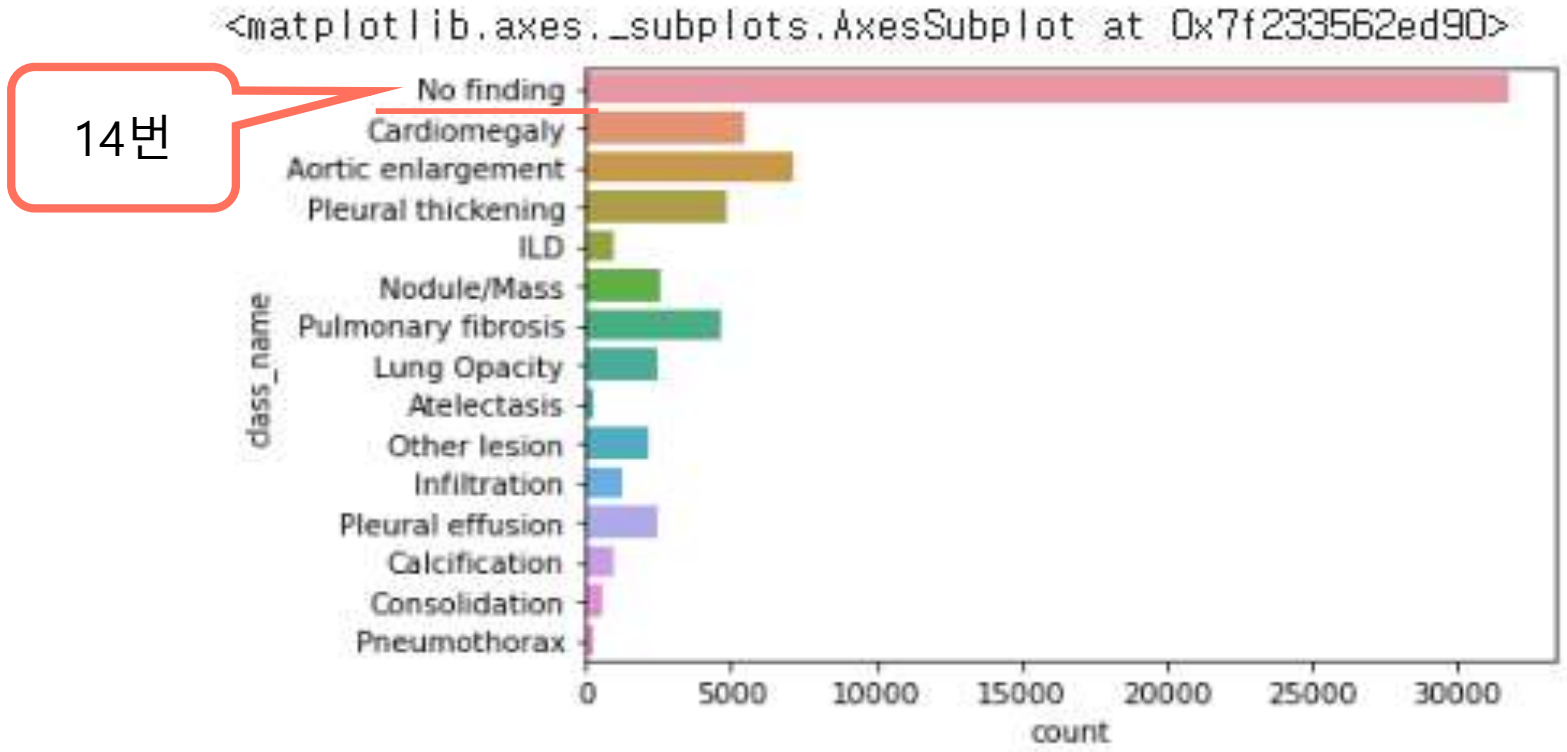


01 정상치와 병변치 구분하기

02 14가지의 병변의 따라 구분하기

03 병변 구분 및 병변에 따른 bbox 확인

I.이중 분류 하기



정상치 14번 → 0 , 병변치 0~13번 → 1

I. 이중 분류 하기

```
chest1.iloc[:,2:3] = np.where (chest1.iloc[:,2:3]== 14  , 0  ,1)
chest1
```

	image_id	class_nam	class_id	min	y_min	x_max	y_max	img_path
0	50a418190bc3fb1ef1633bf9678929b3	No findi	0	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/50a418190bc...
1	21a10246a5ec7af151081d0cd6d65dc9	No findi	0	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/21a10246a5e...
2	9a5094b2563a1ef3ff50dc5c7ff71345	Cardiomega	1	91.0	1375.0	1653.0	1831.0	/content/drive/MyDrive/과제2/train/9a5094b2563...
3	051132a778e61a86eb147c7c6f564dfe	Aortic enlargeme	1	64.0	743.0	1611.0	1019.0	/content/drive/MyDrive/과제2/train/051132a778e...
4	063319de25ce7edb9b1c6b8881290140	No findi	0	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/063319de25c...
...	...		...	...	...	...	...	...
67909	936fd5cff1c058d39817a08f58b72cae	No findi	0	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/936fd5cff1c...
67910	ca7e72954550eeb610fe22bf0244b7fa	No findi	0	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/ca7e7295455...
67911	aa17d5312a0fb4a2939436abca7f9579	No findi	0	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/aa17d5312a0...
67912	4b56bc6d22b192f075f13231419dfcc8	Cardiomega	1	71.0	979.0	1680.0	1311.0	/content/drive/MyDrive/과제2/train/4b56bc6d22b...
67913	5e272e3adbdaafb07a7e84a9e62b1a4c	No findi	0	0.1	0.1	0.1	0.1	/content/drive/MyDrive/과제2/train/5e272e3adbd...

67914 rows × 8 columns

정상치 14번 → 0 , 병변치 0~13번 → 1

I.이중 분류 하기

```
# 렌덤포레스트

from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import pandas as pd
import warnings
warnings. filterwarnings('ignore')

X_featers= chest2.img_path
y_lable = chest2.class_id

# 결정 트리에서 사용한 get_human_dataset() 이용해 학습/ 테스트용 DataFrame 변환
X_train,X_test,y_train,y_test= train_test_split(X_featers, y_lable, test_size=0.3, random_state= 121)

# 랜덤 포레스트 학습 및 별도의 테스트 세트로 예측 성능 평가
rf_clf = RandomForestClassifier(random_state= 0)
rf_clf.fit(X_train,y_train)
pred = rf_clf.predict(X_test)
accuracy = accuracy_score(y_test , pred)
print("랜덤 포레스트 정확도 : {0:.4f}".format(accuracy))
```

```
ValueError: could not convert string to float: '/content/drive/MyDrive/과제2/train/31ad9a7330a3d2abe38bce55c4bf1109.dicom'
```

랜덤포레스트로 모델 확인하려 했으나 X_featers값으로 경로가 들어가 Error 발생.

→ 경로에 가서 이미지를 가져오는 함수를 만들기로 결정.

I.이중 분류 하기

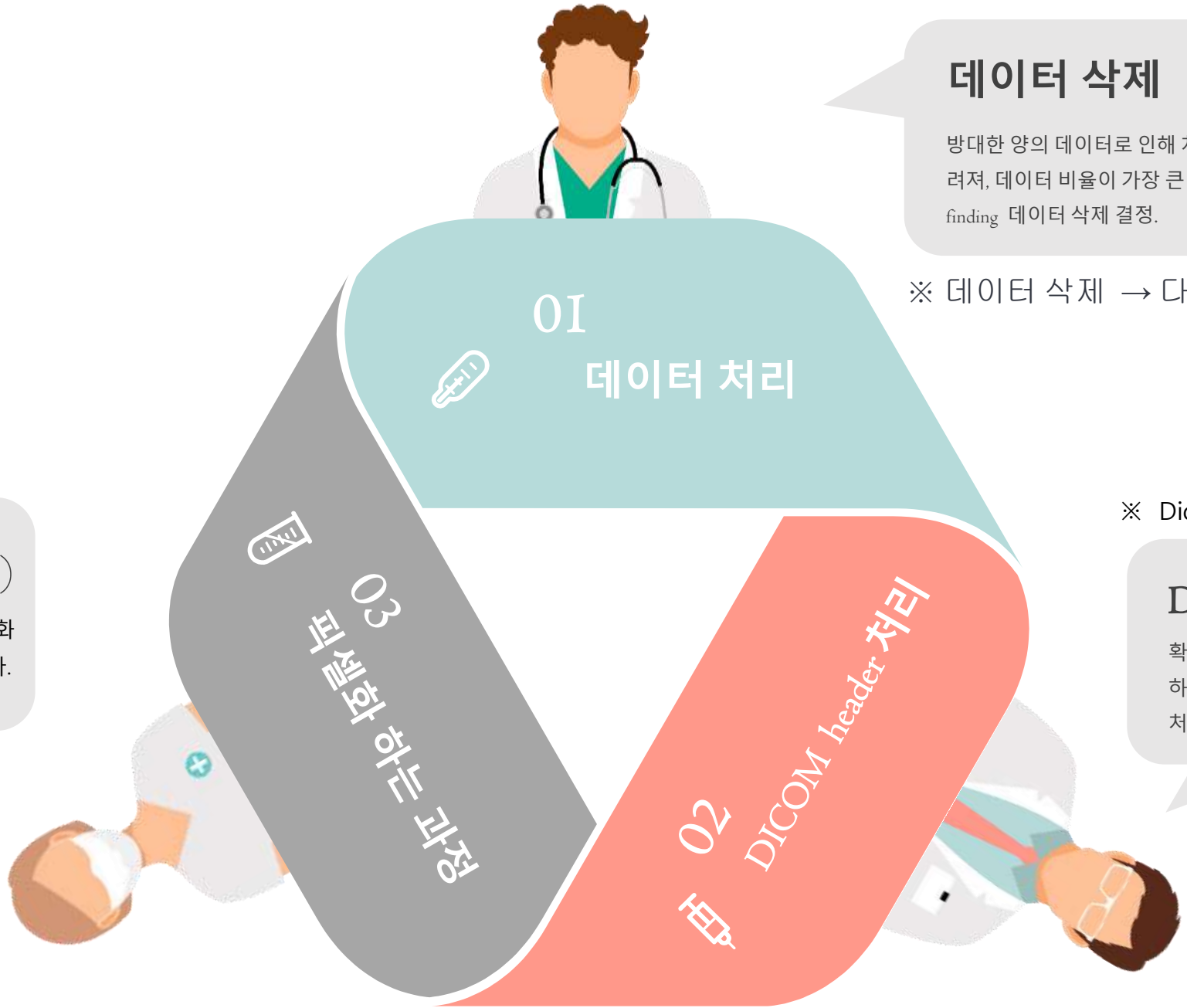
픽셀화 하는 함수

```
# 픽셀화 하는 함수

def loader(path) :
    out = []
    for i in path :
        img_out = pydicom.dcmread(i)
        out.append(img_out.pixel_array)
    return out
```

데이터 양이 많아 함수를 돌릴 수 없었다.

- 데이터 삭제
- Mritopng 라이브러리로 png 변환

문제 해결



데이터 삭제

방대한 양의 데이터로 인해 처리 속도 느려져, 데이터 비율이 가장 큰 no finding 데이터 삭제 결정.

※ 데이터 삭제 → 다중 분류로 전환.

※ Dicom : 의료용 디지털 확장자

DICOM -> PNG (1)

확장자 안에 환자 정보 값이 불필요하게 들어있어 처리과정이 필요.
처리 속도가 느려짐 방지.

DICOM -> PNG (2)

이미지에 접근하기 위해서는 pixel화 해주는 과정을 한번 더 거쳐야 한다.

2.다중 분류 하기

(1) 사이즈 변경

```
# 이미지 사이즈 조정하기 위해서 albumentations Module 설정
# 처음에는 이미지를 512사이즈로 조정하려 했으나 더 빠른 학습을 위해서 214사이즈로 변경

IMAGE_SHAPE = [214, 214]
transforms_train = A.Compose([
    A.Resize(height=IMAGE_SHAPE[0], width=IMAGE_SHAPE[1], p=1.0),
])
```

(2) 원-핫 인코딩, split

```
from sklearn.model_selection import train_test_split
from keras.utils import np_utils

images = train_data_df['img_path']

# 현재 class_id 0~13번까지니까 이걸 원-핫-인코딩(카테고리)화 해줌
labels = list(np_utils.to_categorical(train_data_df['class_id'], 14))

# split해줬더니 trian 이미지 -개, val 이미지 -개로 나뉘어진 모습을 확인
train_images, val_images, train_labels, val_labels = train_test_split(images, labels, test_size = 0.3, random_state=121, stratify=labels)
print(f'train image count : {train_images.shape[0]}')
print(f'val image count : {val_images.shape[0]}')
```

```
train image count : 25267
val image count : 10829
```

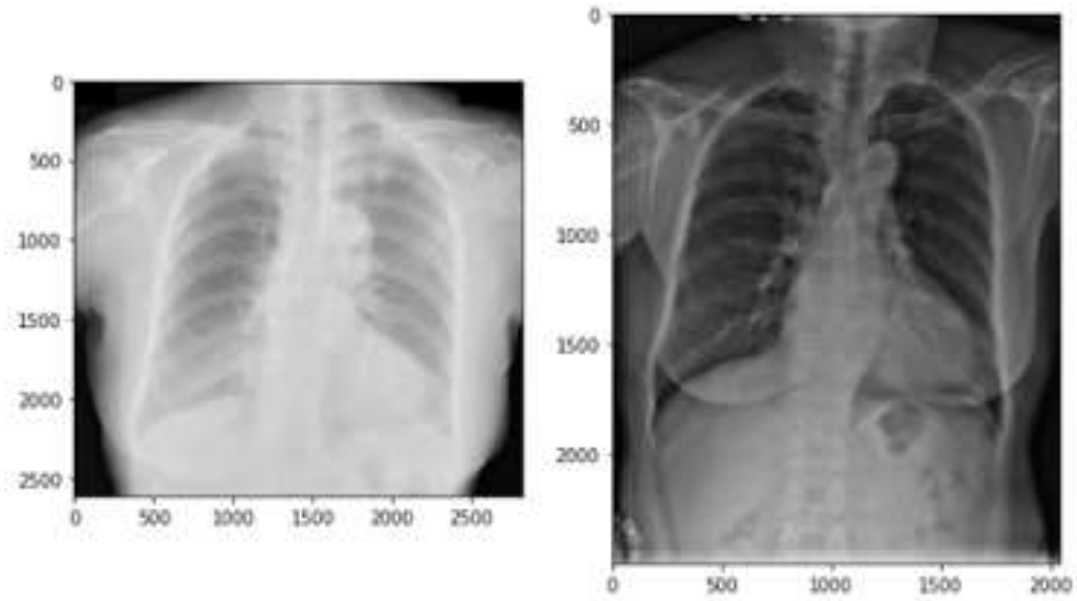
2.다중 분류 하기

(3) 이미지 경로 확인

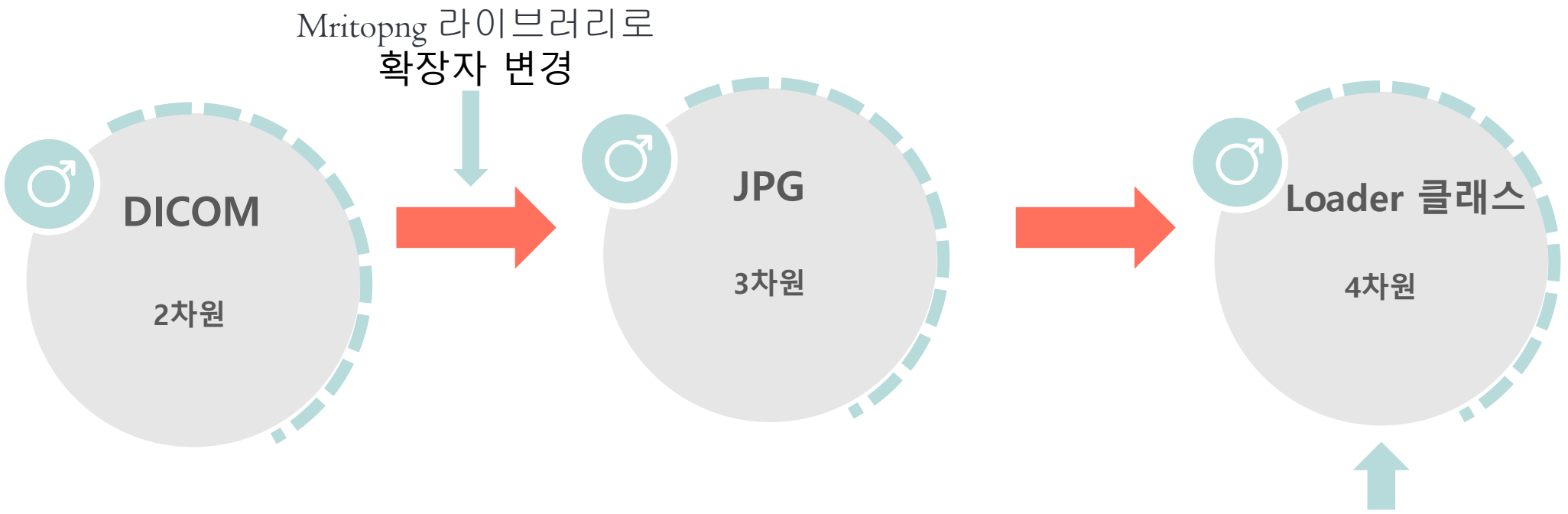
```
# 이미지 확인
fig, axs = plt.subplots(1, 2, figsize=(10, 8))
axs = axs.flatten()

train_img_ck = cv2.imread(train_images.iloc[0])
train_img_ck = cv2.cvtColor(train_img_ck, cv2.COLOR_BGR2RGB)
axs[0].imshow(train_img_ck, cmap='gray')
print(train_img_ck.shape)

test_img_ck = cv2.imread(val_images.iloc[0])
test_img_ck = cv2.cvtColor(test_img_ck, cv2.COLOR_BGR2RGB)
axs[1].imshow(test_img_ck, cmap='gray')
print(test_img_ck.shape)
```

```
(2615, 2625, 3)
(2500, 2048, 3)
```

3차원



잠깐, 차원의 문제


Mritopng 라이브러리로
확장자 변경
DICOM
2차원
JPG
3차원
Loader 클래스
4차원
최종으로 필요한 차원

2.다중 분류 하기

(4) 카테고리화, 이미지 차원 확인

```
tf.Tensor(
[[1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]], shape=(10, 14), dtype=float32)
```

```
(2615, 2625, 3)
(2500, 2048, 3)
```

3차원

```
print(images.shape)
```

```
(20, 512, 512, 3)
```

이미지 데이터 3차원 → 4차원으로 변경

2.다중 분류 하기

(5) 모델 확인

```
model.summary()
```

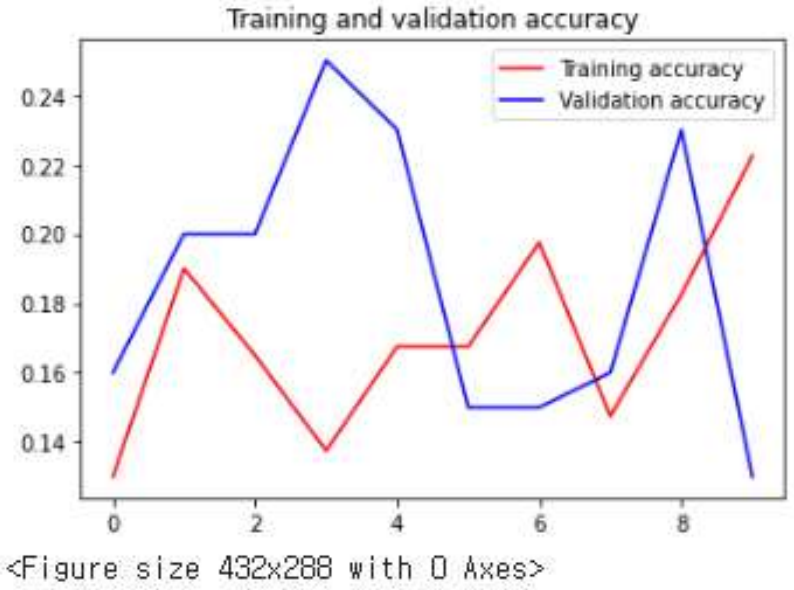
```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                 Output Shape              Param #
=================================================================
 conv2d_14 (Conv2D)           (None, 510, 510, 32)      896

 conv2d_15 (Conv2D)           (None, 508, 508, 64)      18496

 max_pooling2d_5 (MaxPooling  (None, 254, 254, 64)      0
 2D)

 dropout (Dropout)            (None, 254, 254, 64)      0

 flatten (Flatten)            (None, 4129024)           0

 dense_2 (Dense)              (None, 128)               528515200

 dropout_1 (Dropout)          (None, 128)               0

 dense_3 (Dense)              (None, 14)                1806

=================================================================
Total params: 528,536,398
Trainable params: 528,536,398
Non-trainable params: 0
_________________________________________________________________
```

```
from tensorflow import keras

train_step = 40
val_step = 10
test_step = 1
epoch = 20

history = fit_test(model=model,
                   train_gen=train_data_gen,
                   train_steps=train_step,
                   epochs=10,
                   val_gen=valid_data_gen,
                   val_steps=val_step)
```

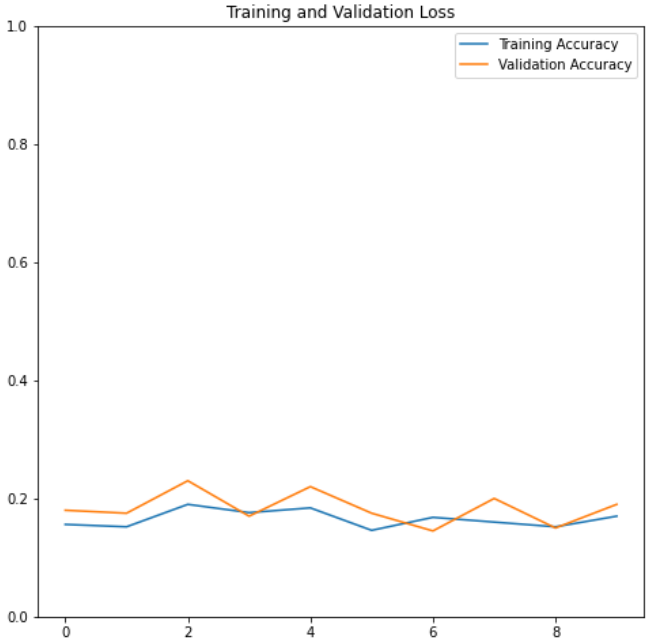
```
/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:6: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.

Epoch 1/10
80/80 [==============================] - 1769s 22s/step - loss: 6.8183 - accuracy: 0.1746 - val_loss: 2.3120 - val_accuracy: 0.2078
Epoch 2/10
80/80 [==============================] - 1133s 14s/step - loss: 2.3904 - accuracy: 0.1742 - val_loss: 2.3300 - val_accuracy: 0.1609
Epoch 3/10
80/80 [==============================] - 908s 11s/step - loss: 2.3434 - accuracy: 0.1840 - val_loss: 2.2771 - val_accuracy: 0.2281
Epoch 4/10
80/80 [==============================] - 909s 10s/step - loss: 2.3396 - accuracy: 0.2059 - val_loss: 2.2127 - val_accuracy: 0.2219
Epoch 5/10
80/80 [==============================] - 790s 10s/step - loss: 2.3407 - accuracy: 0.1908 - val_loss: 2.3093 - val_accuracy: 0.1922
Epoch 6/10
80/80 [==============================] - 799s 10s/step - loss: 2.3305 - accuracy: 0.1632 - val_loss: 2.3434 - val_accuracy: 0.1953
Epoch 7/10
80/80 [==============================] - 735s 9s/step - loss: 2.3374 - accuracy: 0.1937 - val_loss: 2.2924 - val_accuracy: 0.2250
Epoch 8/10
80/80 [==============================] - 807s 10s/step - loss: 2.3218 - accuracy: 0.1746 - val_loss: 2.3220 - val_accuracy: 0.2156
Epoch 9/10
80/80 [==============================] - 781s 10s/step - loss: 2.2981 - accuracy: 0.2022 - val_loss: 2.3703 - val_accuracy: 0.1750
Epoch 10/10
80/80 [==============================] - 757s 9s/step - loss: 2.3067 - accuracy: 0.1945 - val_loss: 2.3251 - val_accuracy: 0.2000
<keras.callbacks.History at 0x7fe6f2e92a90>
```

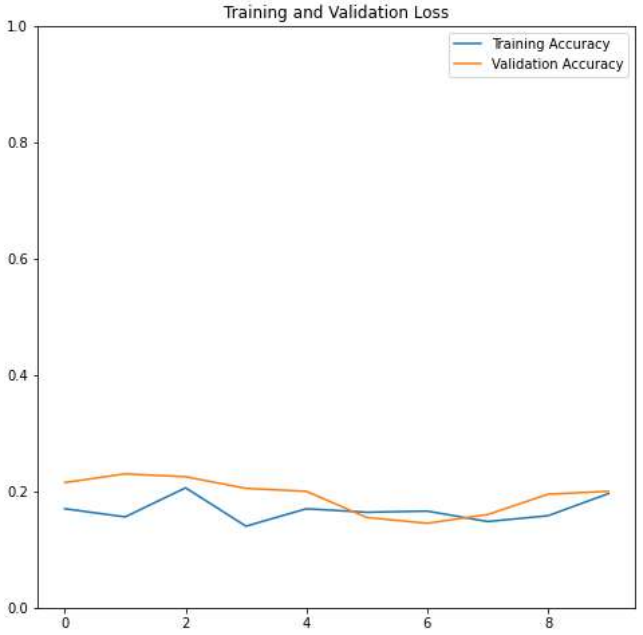
결과

시각화 (그래프)



<Figure size 432x288 with 0 Axes>



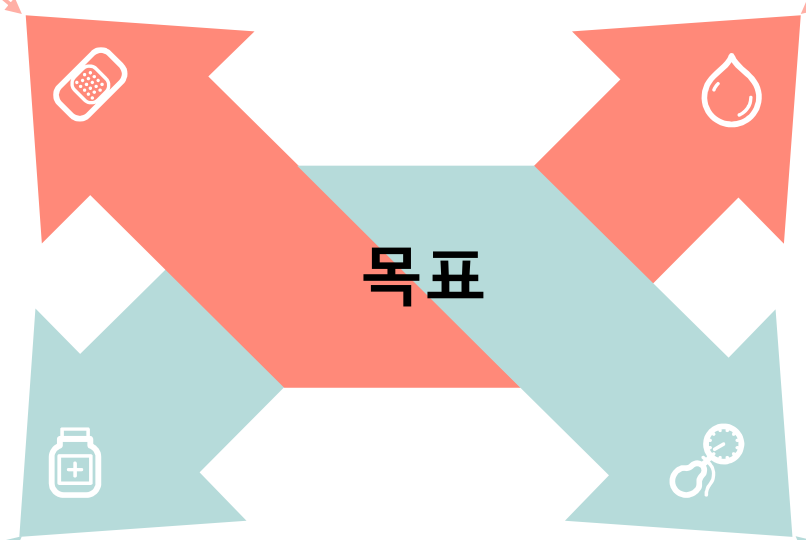


Acc: 80%

정확도 80% 넘기기

병변 부위 bbox

Tset 이미지 넣어서 병변 있는 부위 bbox 그리기.



Acc: 25%

목표에 도달하기 위해 계속 작업 중 이다.

Bbox 이미지 확인

정확도가 낮아서 아직 bbox의 이미지 까지는 확인하지 못한 상태이다.

추후 과정

01 Acc : 80% 이상 끌어올리기

02 Grad-CAM , bbox 그리기

Thank you