

CHEST X-RAY

Object Detection

김수현 김현나 이봉학 이소정

01.

주제 소개

- 아이디어 선정 이유
- 기대효과

02.

모델 소개

- 모델 선정 이유
- 모델 소개
 - Faster R-CNN
 - YOLOX
 - EfficientDet

03.

데이터 / 전처리

- 데이터 소개
- 데이터 이슈
- 데이터 이슈 해결방안
 - 양 : augmentation
 - 불균형 : up-sampling
 - down-sampling
- workflow

04.

결과

- Inference
- Kaggle 점수 및 Insight

01

주제 소개

1-1. 아이디어 선정 이유

「삼성서울병원_기본건강검진_공통검사항목」

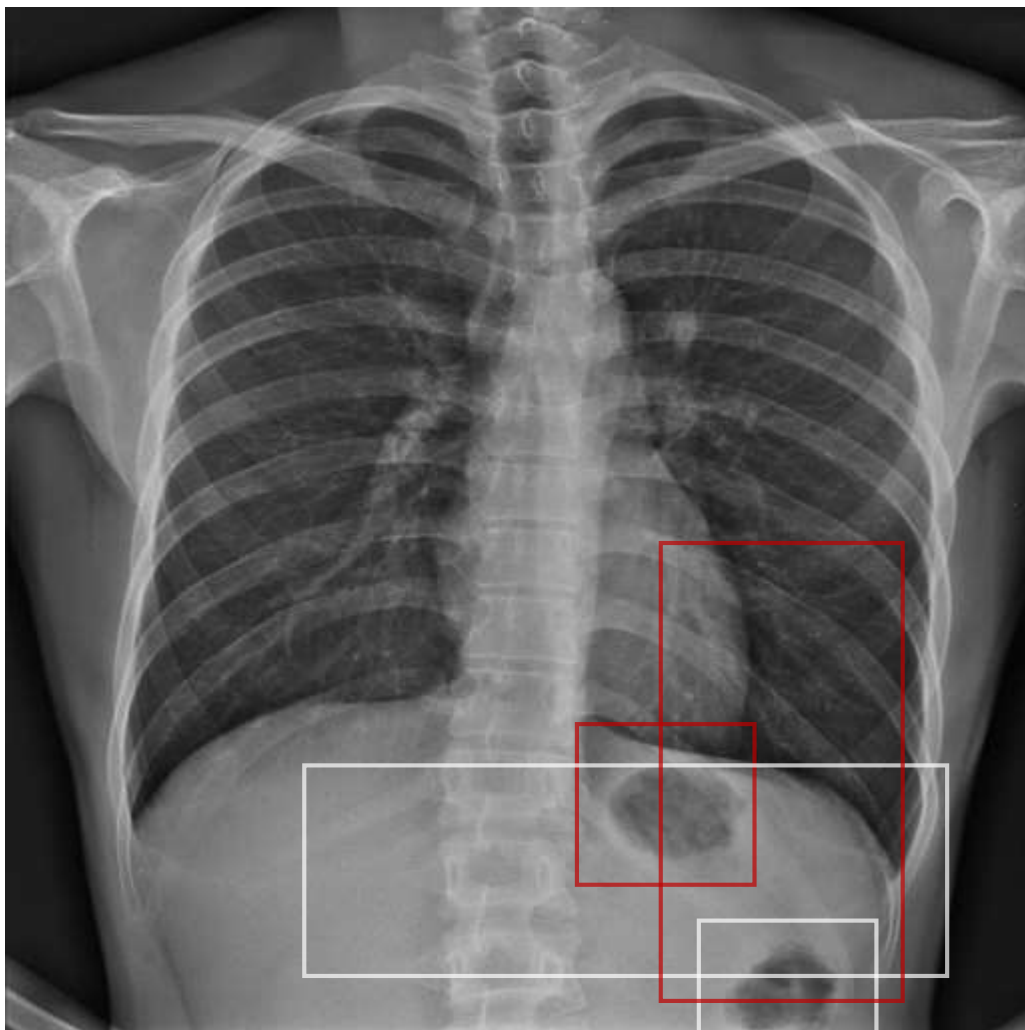
공통 검사항목

- 신체측정 (신장, 체중, 비만도, 혈압)
- 상복부초음파
- 심전도 **흉부촬영 (정면)**
- 위내시경
- 소변, 대변검사
- 안과 (시력, 안압, 안저 촬영)
- 식생활평가/영양상담
- 혈액검사 (일반혈액, 혈액응고, 간기능, 신장기능, 요산, 갑상선기능, 전해질, 고지혈, 당뇨, 혈액형, 간염, CRP, 매독, 에이즈, 류마티스인자, 비타민D, 종양표지자(간,대장,췌장,담도))

에이즈, 류마티스인자, 비타민D, 종양표지자(간,대장,췌장,담도))

• 혈액검사 (일반혈액, 혈액응고, 간기능, 신장기능, 요산, 갑상선기능, 전해질, 고지혈, 당뇨, 혈액형, 간염, CRP, 매독,

1-2. 목표



프로젝트 목표 : *CXR을 object detection을 통해 폐 관련 14가지 질병을 detection.

학습 목표 : Data augmentation, 모델에 따른 성능 비교.

02

모델 소개

2-1. 모델 선정 이유

YOLOX

One-Stage model (dense prediction)

Faster R-CNN

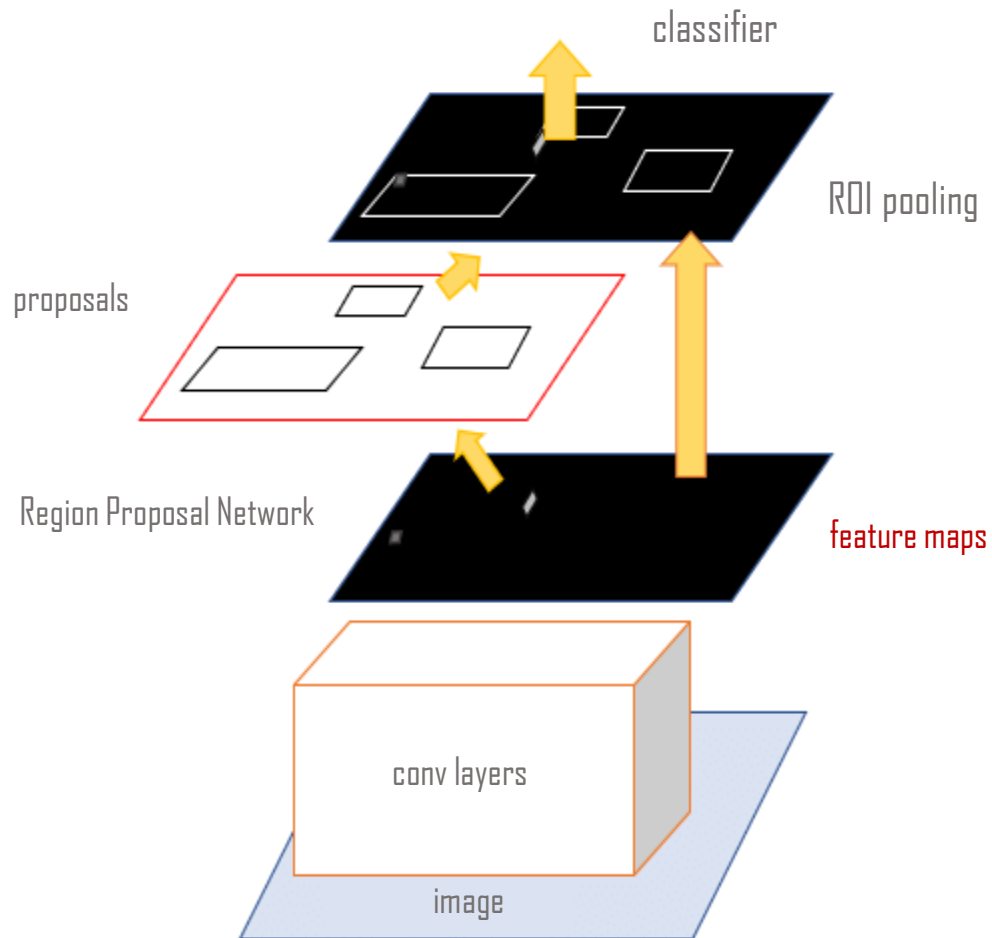
Two-Stage model (sparse prediction)

EfficientDet

width ,depth ,resolution에 따른 효율적인 모델

2-2. Faster R-CNN

>>> Architecture



- RPN (Region Proposal Network) 도입 및 FM(Feature Map) 공유.
→ 속도가 빨라지고 end-to-end 학습 가능해짐.
- ROI-pooling
→ FM 상에서 물체의 위치 예측하고 ROI-pooling을 통해 고정된 크기의 FM 출력.
- Classification & Bounding Box Regression
→ Fast R-CNN과 동일.

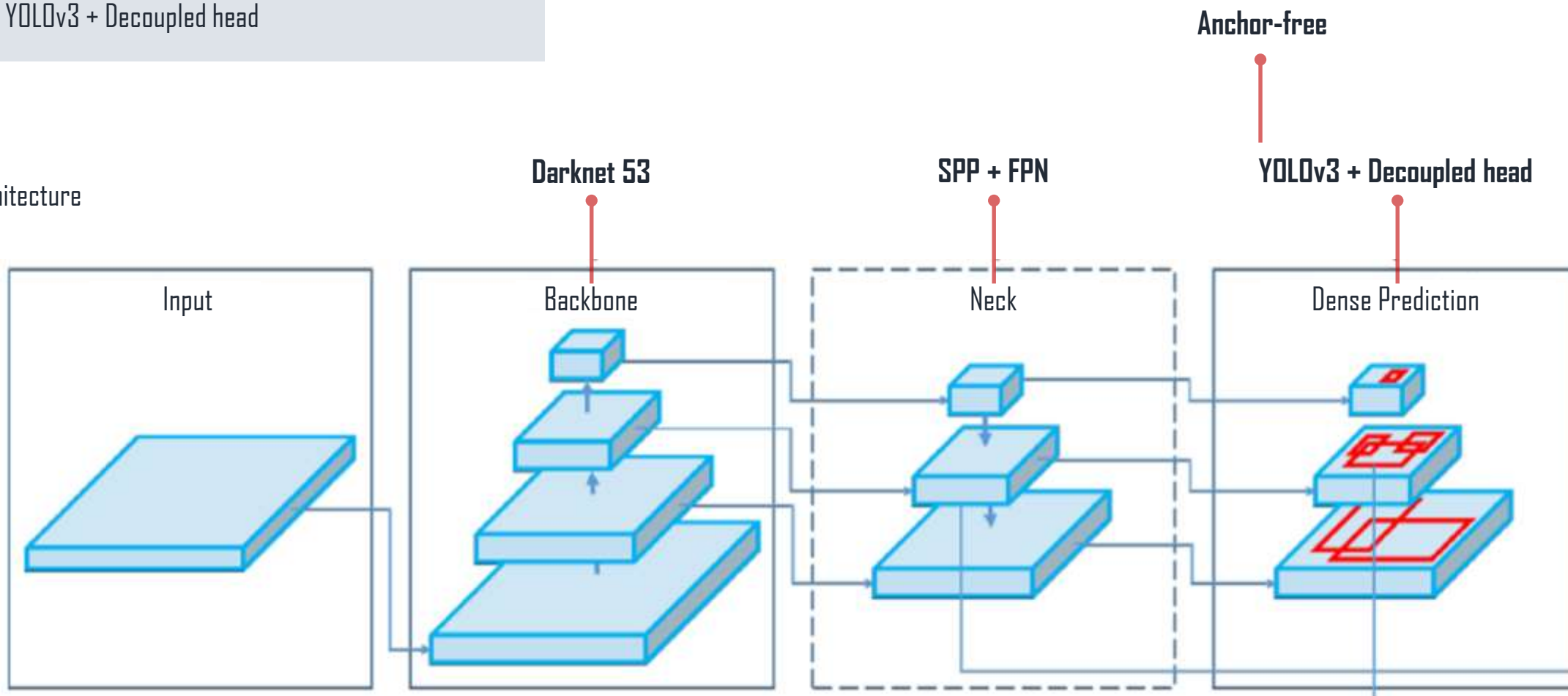
2-2. YOLOX

Backbone : Darknet53

Neck : SPP (Spatial Pyramid Pooling)

Head : YOLOv3 + Decoupled head

>>> Architecture



One - Stage Detector

2-2. EfficientDet

- 최적의 depth, width, resolution

→ 그리드 서치를 통해 최적의 α, β, γ 의 연관성을 찾아냈고,

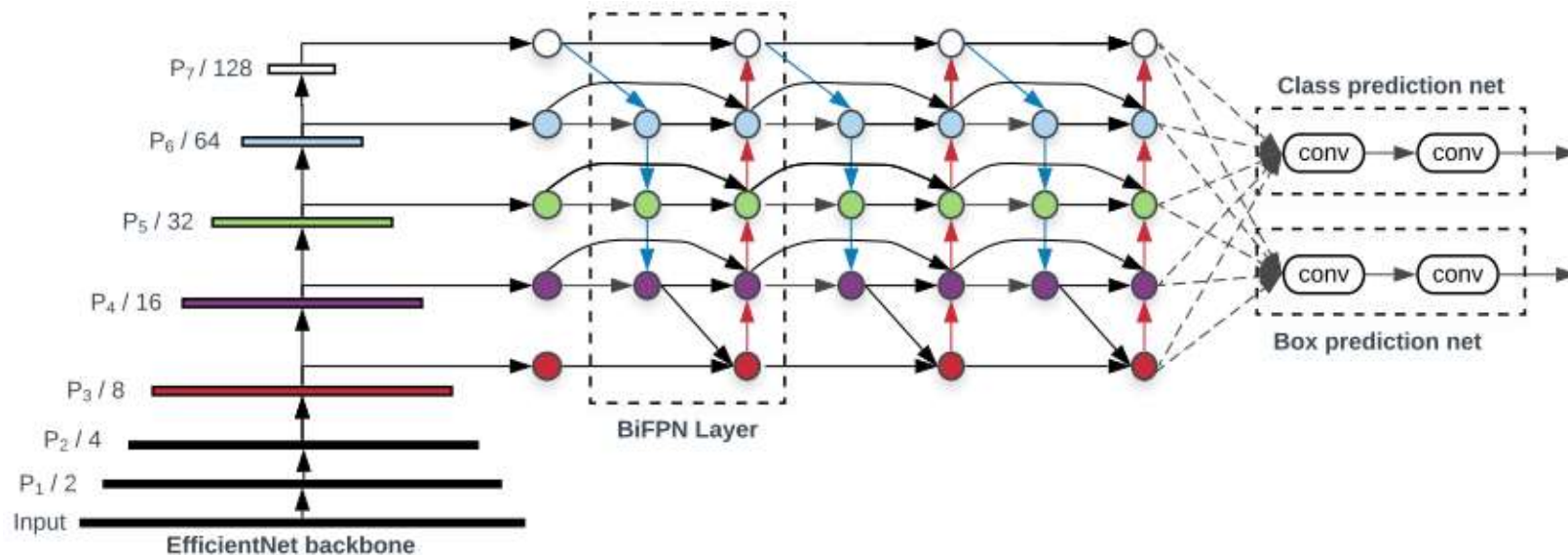
$\alpha=1.2, \beta=1.1, \gamma=1.15$ 의 값으로 고정한 뒤, 파이(ϕ)를 키워주며 모델
사이즈를 키웠다.

$1.2^\phi, 1.1^\phi, 1.15^\phi$

- BiFPN

→ 기존의 FPN에서 top-down, bottom-up 방식을 반복 사용하고,
레이어마다 가중치를 주어 각각의 층에 대한 해상도 정보가 잘
들어
갈 수 있도록 함.

>>> Architecture



Backbone : EfficientNet

Neck : BiFPN

Head : Box Prediction Net

03

데이터 / 전처리

3-1. 데이터 소개



Data Explorer

205.96 GiB

- ▶ test
- ▶ train
- sample_submission.csv
- train.csv

Kaggle의 'VinBigData Chest X-ray Abnormalities Detection' 대회에서 베트남의 두 병원에서 제공해준 환자들의 흉부 x-ray 데이터셋을 가져왔다. 데이터는 Dicom 파일의 **test images 3000장**, **train images 15,000장**, 그리고 train image의 **image_id, class_id, x_min, y_min, x_max, y_max** 정보가 들어있는 **train.csv** 파일로 이루어져있다.

01. 적은 양의 데이터

최종 데이터:

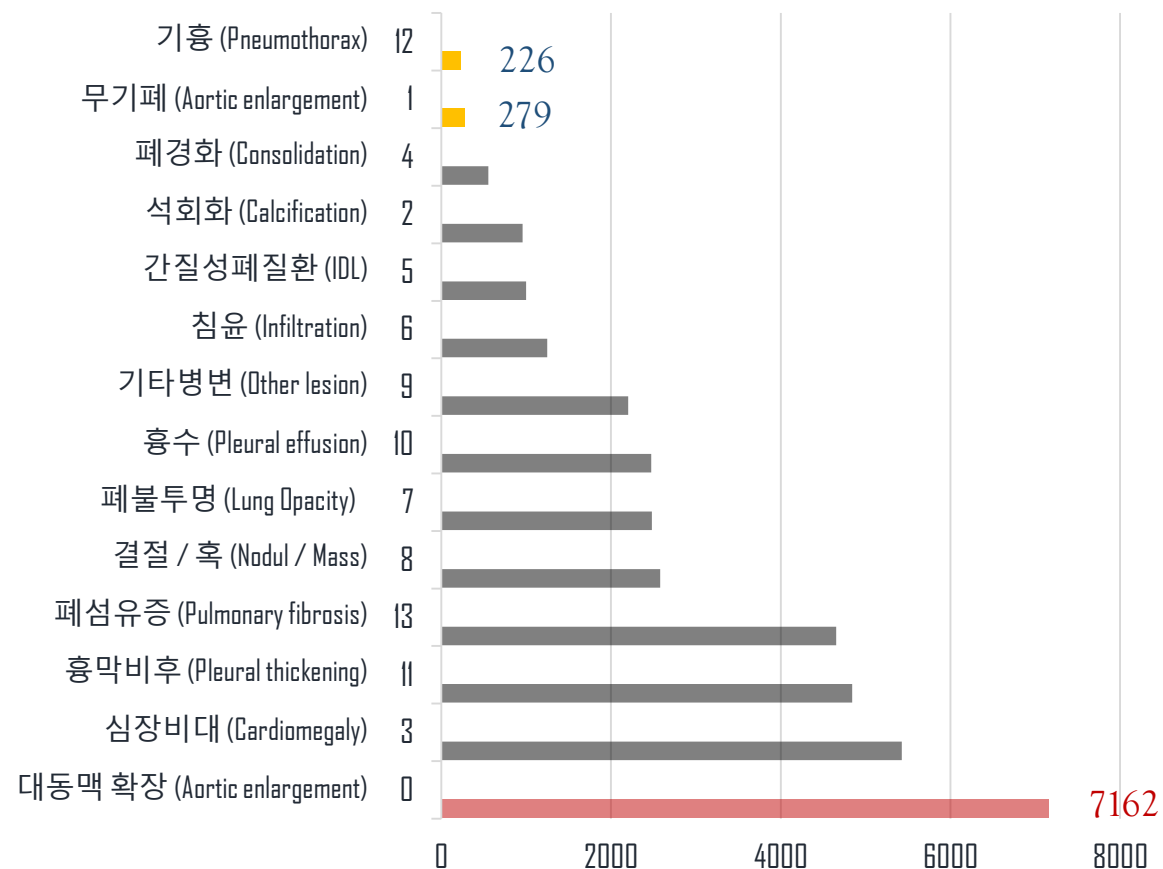
14번 라벨(정상인) ↖
15000장 - 10606장 = 4394장

이미지 4394 장, 라벨 36096개

데이터 불균형 해결이 가장 어려웠던 이유:

- 라벨 간의 극단적인 양의 차이
- 단일 이미지: 다중 라벨

02. 데이터 불균형



3-3. 데이터 전처리 방안

CATEGORY A

이미지 : 4394 장

전처리 x

CATEGORY B

이미지 : 17576 장

Rotation : 90°

Flip : 수평

Zoom : 10%

CATEGORY C

이미지 : 30758 장

Rotation : 90°

Flip : 수평

Zoom : 10%

CutMix

CLAHE

Equalization

Mosaic

Cutmix + Mosaic

CATEGORY D

이미지 : 5999 장

Rotation : 90°

Flip : 수평

Zoom : 10%

CutMix

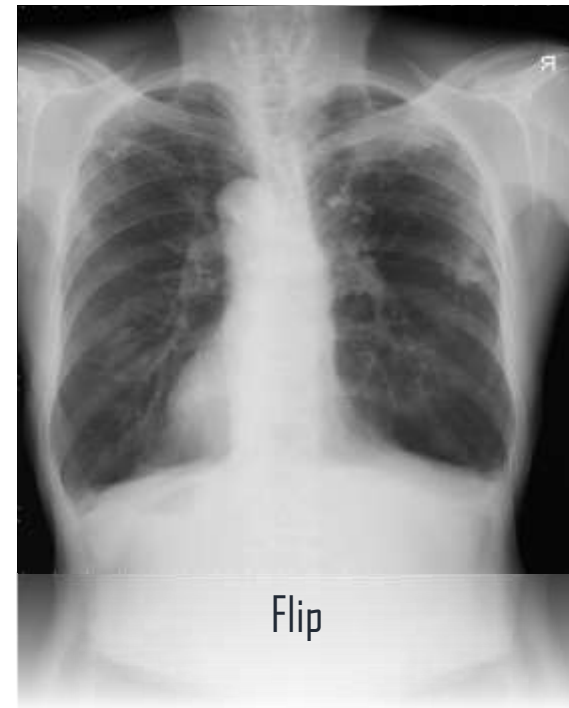
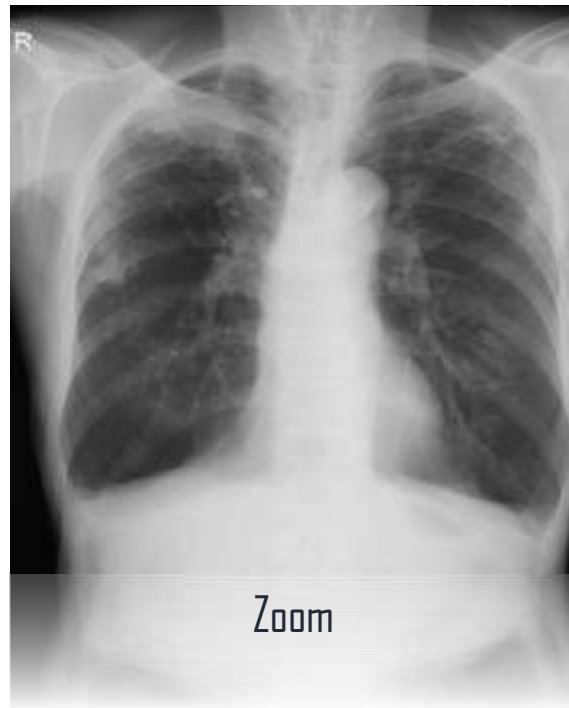
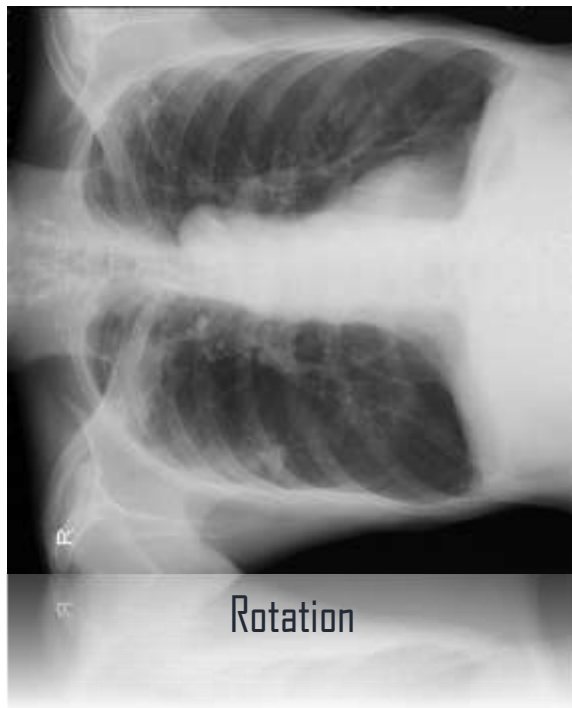
CLAHE

Equalization

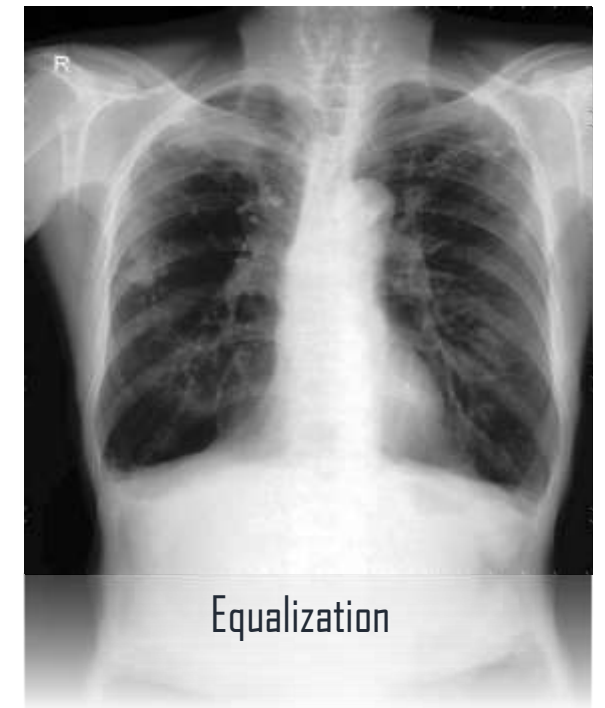


라벨 1과 12에만 적용

Category B

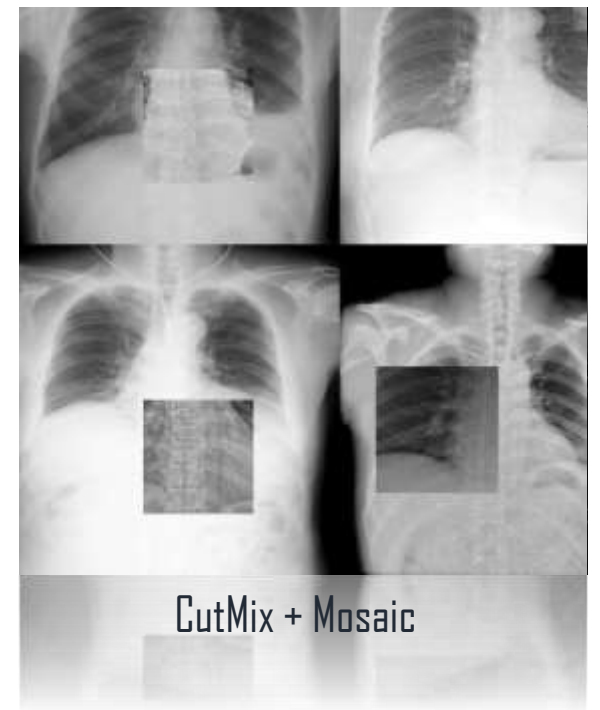
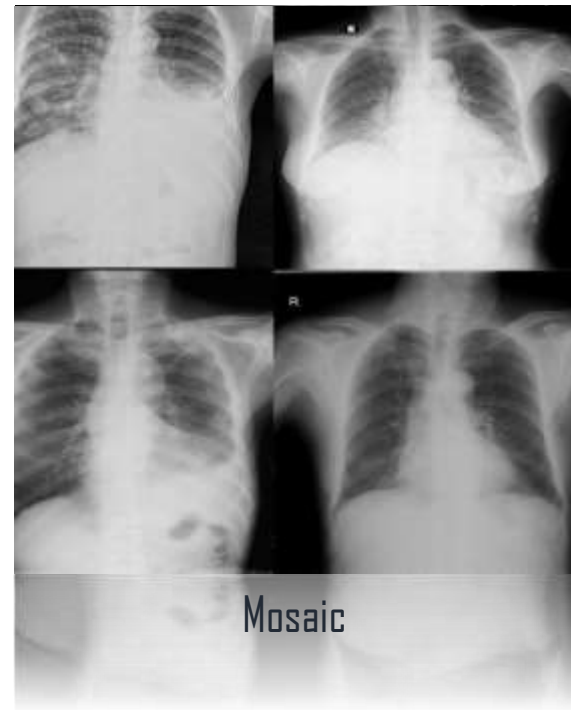
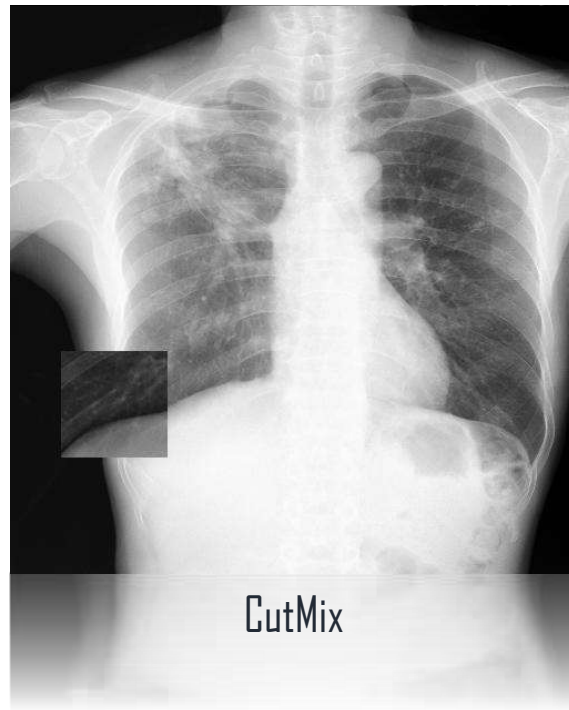


Category C



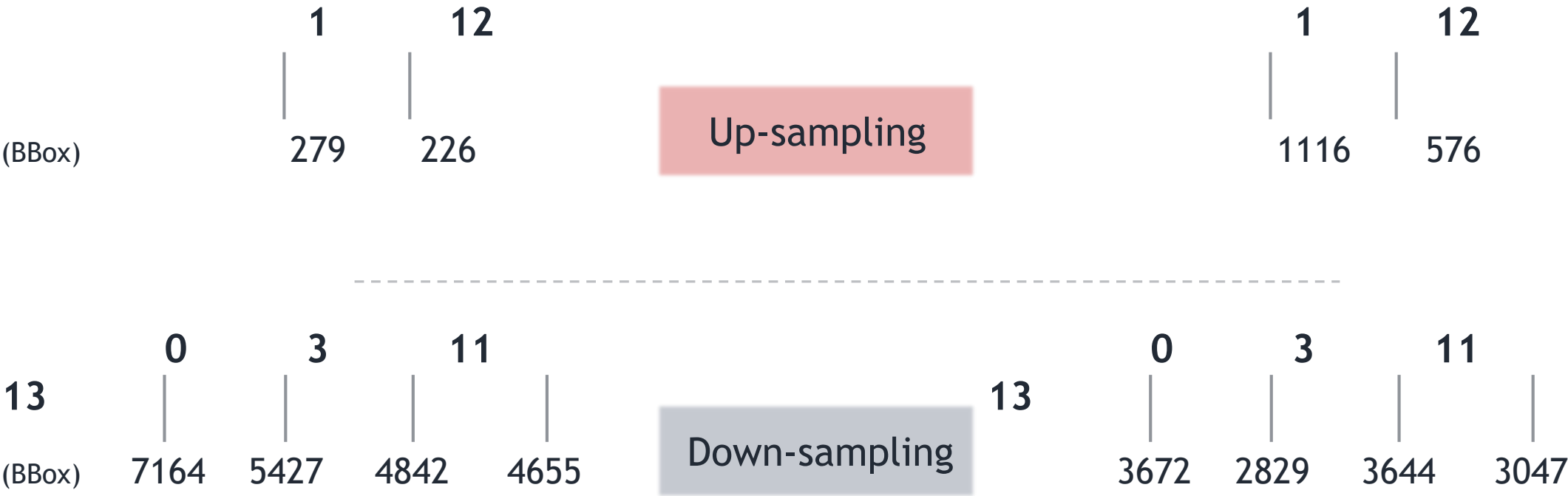
변경 후

Category C

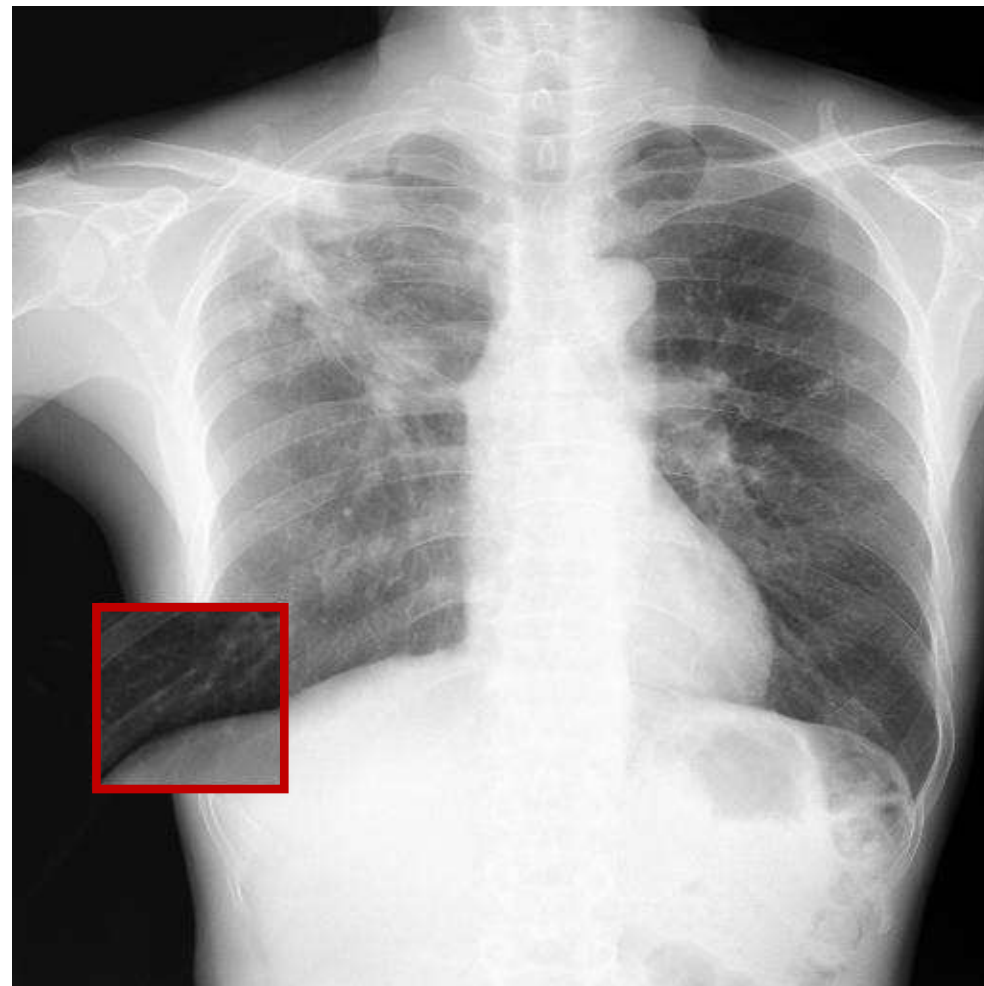


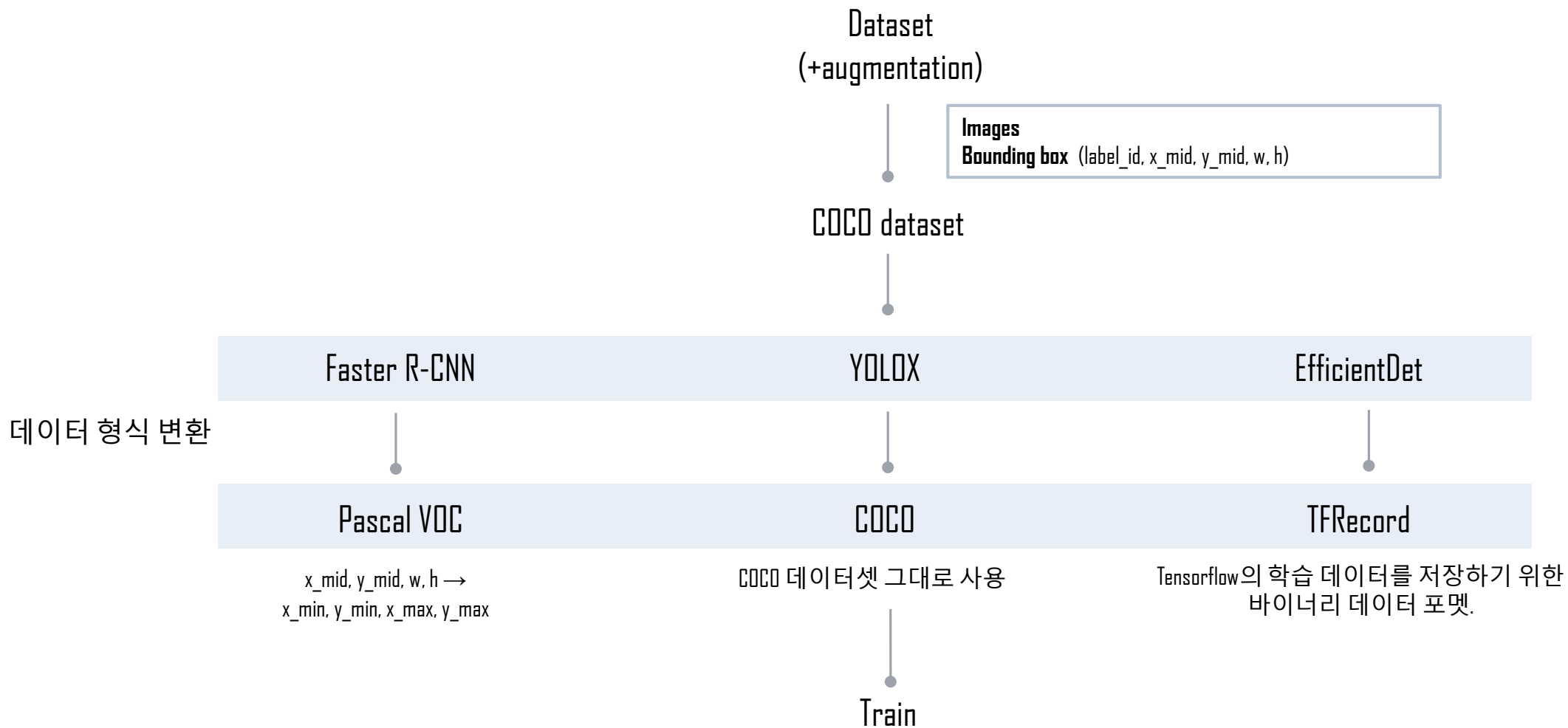
변경 후

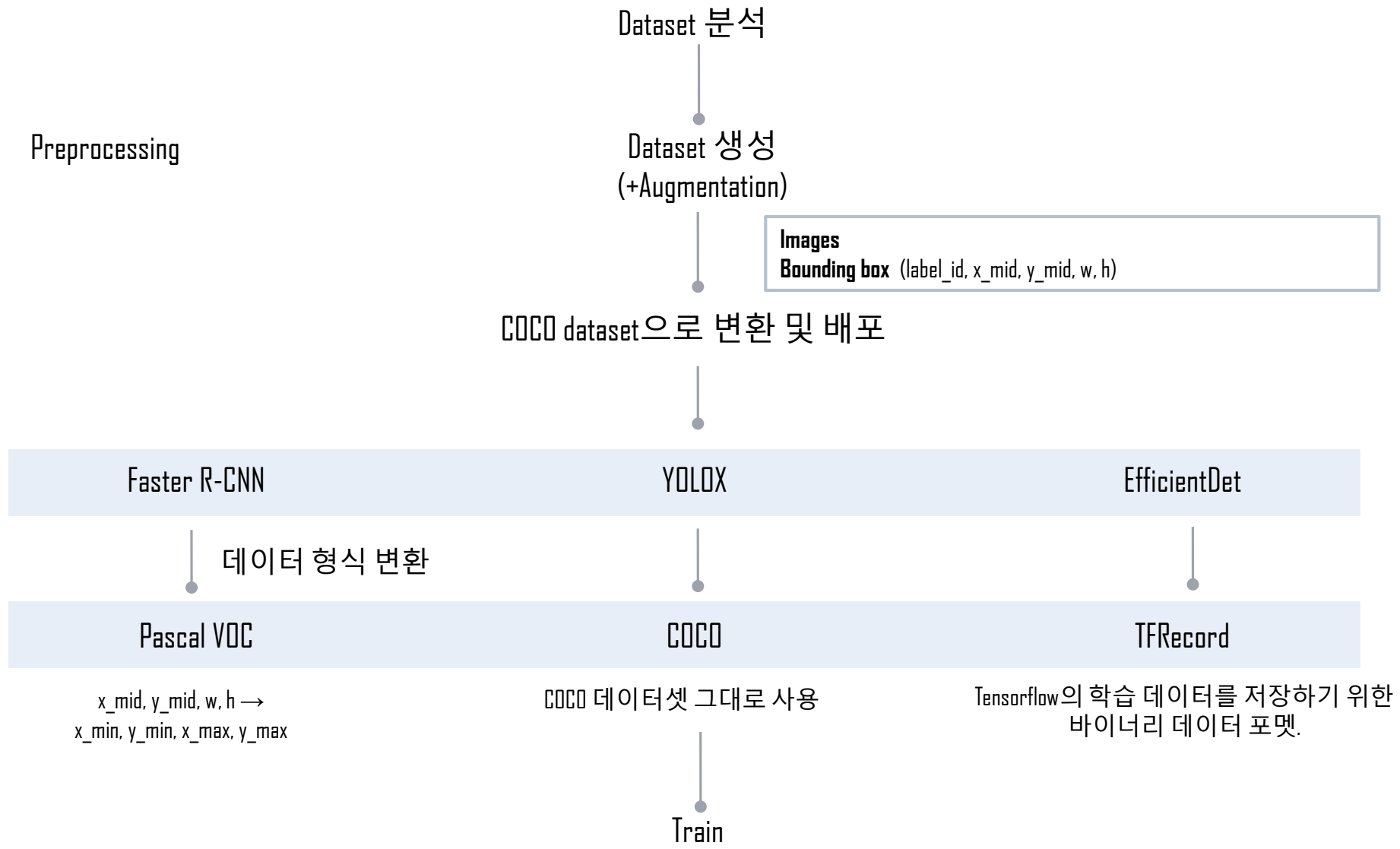
Category D



3-3. 데이터 전처리 이슈 - CutMix



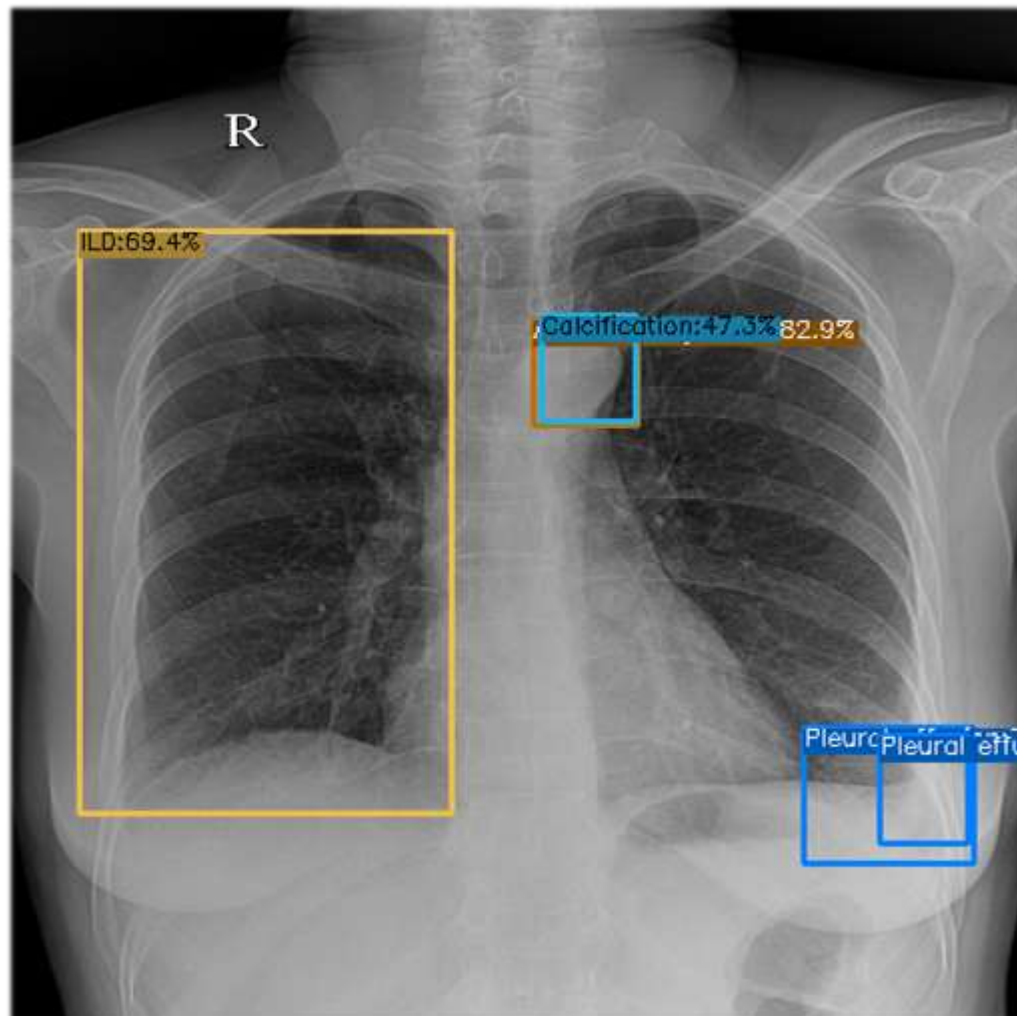




04

결과

4-1. inference



- **Kaggle 점수**

	CATEGORY A	CATEGORY B	CATEGORY C	CATEGORY D
YOLOX	0.021	0.068	0.147	0.070
Faster R-CNN	0.012	0.098	0.013	--
EfficientDet	0.038	0.046	0.052	--

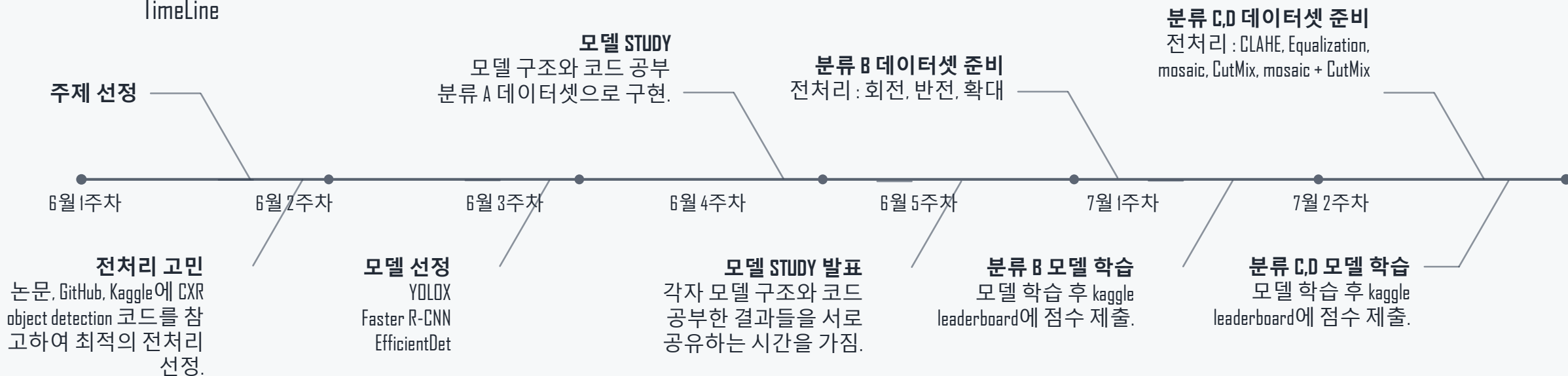
- **Insight**

- 데이터 augmentation에 따른 성능 향상 확인.
- 데이터 불균형 해소에 따른 성능 향상 확인.

05

타임라인 / 이슈사항 / 추후과제

TimeLine



5-2. 이슈사항

Faster R-CNN

Inference시 입출력 이슈 학습할 때 저장한 파라미터가 들어있는 .pt파일로 테스트 이미지 inference중에 에러 발생.

```
RuntimeError: Error(s) in loading state_dict for FasterRCNN:
  size mismatch for roi_heads.box_predictor.cls_score.weight: copying a param with shape torch.Size([15, 1024]) from
  checkpoint, the shape in current model is torch.Size([91, 1024]).

  size mismatch for roi_heads.box_predictor.cls_score.weight: copying a param with shape torch.Size([15, 1024]) from
  checkpoint, the shape in current model is torch.Size([91]).
```

↔ solution: 훈련 시킬 때와 같은 모델 구조를 갖도록 모델의 일부층을 데이터에 맞게 수정시켜서 해결.

```
num_classes 15 #14classes + background

#get number of input features for the classifier
in_features = model.roi_heads.box_predictor.cls_score.in_features

#replace the pre-trained head with a new one
model.roi_head.box_predictor = FastRCNNPredictor(in_feature, num_classes)

model.load_state_dict(torch.load('/content/drive/MyDrive/dataset_B(20epoch)fasterrcnn_model_0.pt'))
```

5-2. 이슈사항

Faster R-CNN

훈련 중에 멈추는 문제 이미지 사이즈는 512x512인데 바운딩박스 좌표가 이미지 크기를 벗어남.

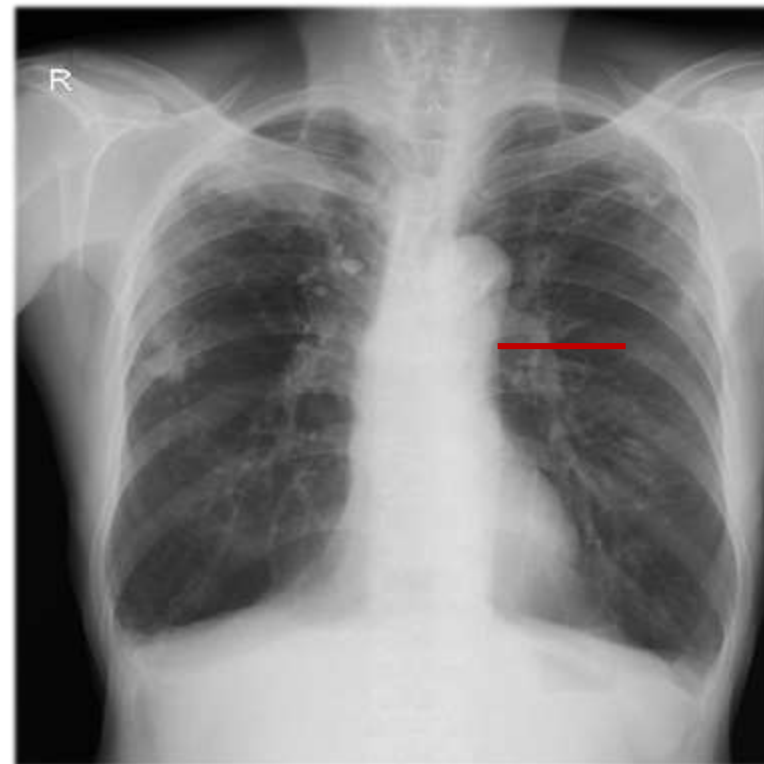
```
"All bounding boxes should have positive height and width. Found invalid box [459.375, 593.75, 500.0, 593.75] for target at index2"
```

↪ solution : 데이터의 y_{min} 과 y_{max} 값이 같은 데이터 삭제.

바운딩박스 오류 예시

$x_{min} = 342.0, x_{max} = 348.0$

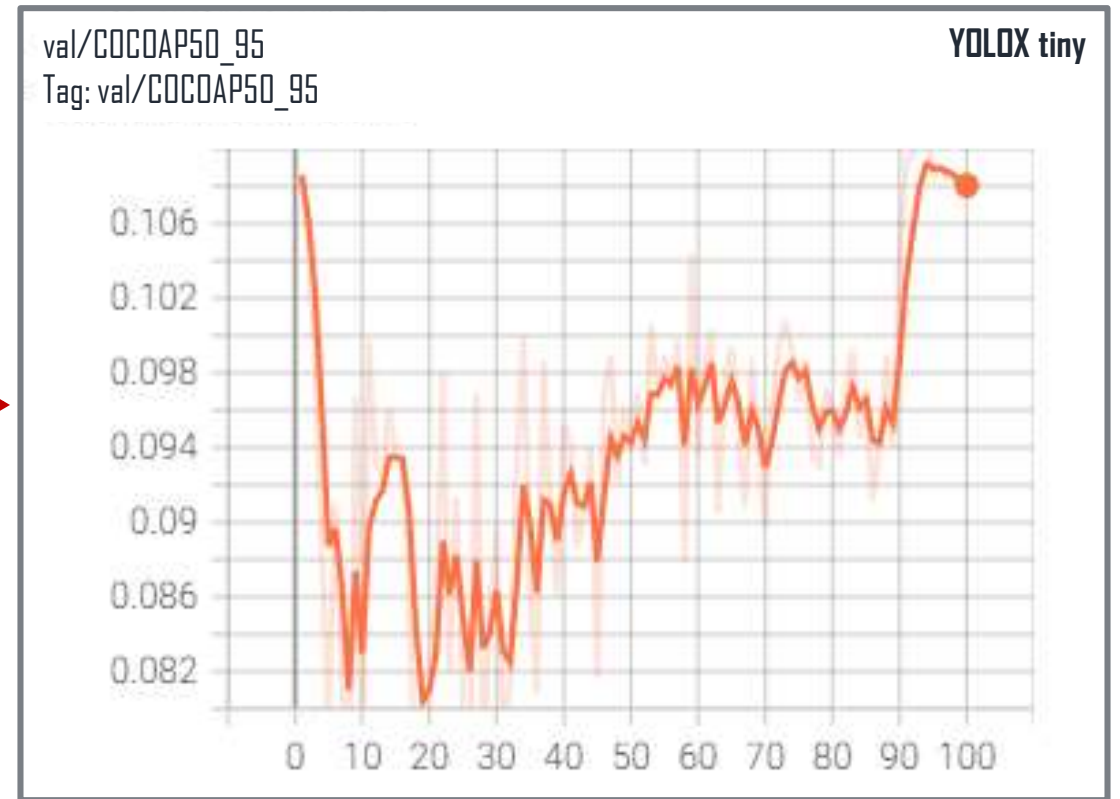
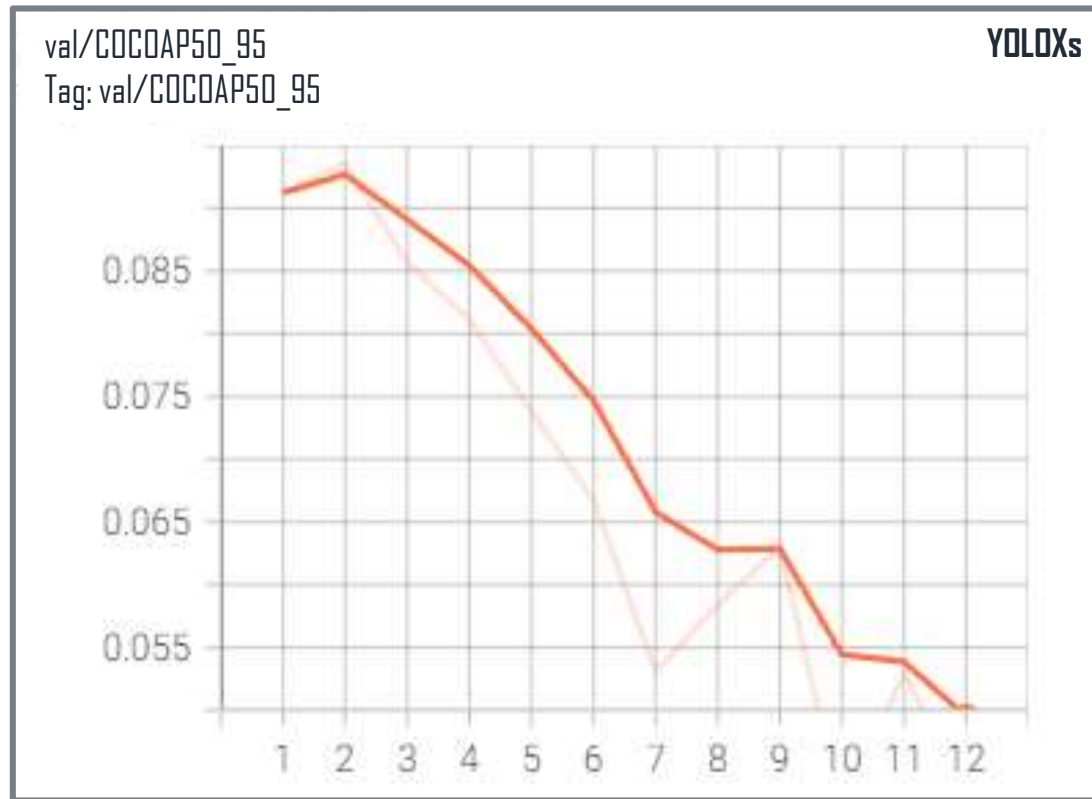
$y_{min} = 270.0, y_{max} = 270.0$



5-2. 이슈사항

YOLOX

정상적인 학습이 안되는 문제



↔ solution : YOLOX s → YOLOX tiny 로 변경.

5-2. 이슈사항

YOLOX

재학습 문제 학습이 후 best 체크포인트 (best.ckpt)로 이어서 학습시켰더니, 전부 재학습이 됨.

↪ solution : 모델 파라미터 중 start epoch 값을 줌.

EfficientDet

배치사이즈 : 8이상의 값을 주면 "falied to allocate memory" 에러 발생

ERROR:

(0) RESOURCE_EXHAUSTED: failed to allocate memory

```
[[{{node EfficientDet-D0/model/stack_4/block_1/expand_activation/Sigmoid}}]]
```

↪ solution : GPU의 VRAM 용량이 크지 않아서 computing 비용을 고려해 **배치사이즈를 8로 고정**시키고 학습.

Q & A