

VinBigData

Chest X-ray Abnormalities Detection

21 . 02 . 26



권형주, 김기영, 손원용, 오상수, 이준호

01 대회 소개 및 일정

02 전처리 & EDA

03 알고리즘 소개

INDEX

04 모형 평가

05 한계 및 시사점

06 참고 문헌

대회 소개 및 일정

kaggle



Description

When you have a broken arm, radiologists help save the day—and the bone. These doctors diagnose and treat medical conditions using imaging techniques like CT and PET scans, MRIs, and, of course, X-rays. Yet, as it happens when working with such a wide variety of medical tools, radiologists face many daily challenges, perhaps the most difficult being the chest radiograph. The interpretation of chest X-rays can lead to medical misdiagnosis, even for the best practicing doctor. Computer-aided detection and diagnosis systems (CADe/CADx) would help reduce the pressure on doctors at metropolitan hospitals and improve diagnostic quality in rural areas.

Existing methods of interpreting chest X-ray images classify them into a list of findings. There is currently no specification of their locations on the image which sometimes leads to inexplicable results. A solution for localizing findings on chest X-ray images is needed for providing doctors with more meaningful diagnostic assistance.

In this competition, you'll automatically localize and classify 14 types of thoracic abnormalities from chest radiographs. You'll work with a dataset consisting of 18,000 scans that have been annotated by experienced radiologists. You can train your model with 15,000 independently-labeled images and will be evaluated on a test set of 3,000 images. These annotations were collected via VinBigData's web-based platform, VinLab. Details on building the dataset can be found in our recent paper "VinDr-CXR: An open dataset of chest X-rays with radiologist's annotations".

kaggle



1. 대회 개요

- (1) 주최 목적 : 전문의도 진단하기 어려운 흉부 X-ray의 질병예측을 보조할 질병판단 모델을 딥러닝을 활용하여 개발
- (2) 전체 대회 기간 : 2020.12.31 ~ 2021.03.31 (3개월)
- (3) 실제 대회 참여 기간 : 2021.02.08 ~ 2021.02.26 (3주)
- (4) 대회 참여 인원 : 743 팀

2. 대회 참여 목적

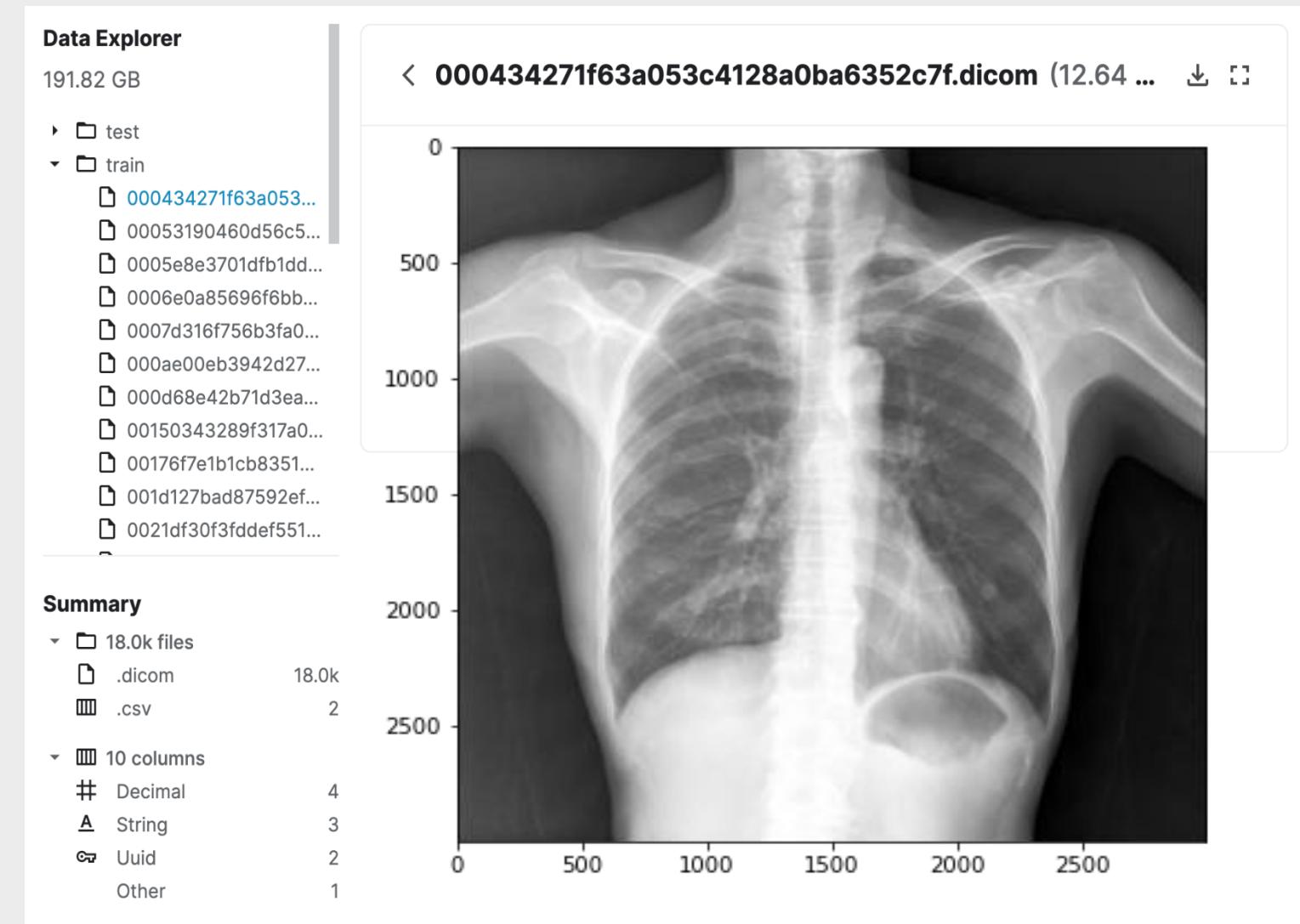
- (1) 의료 영상 이미지 데이터 처리
- (2) 딥러닝 모델 및 알고리즘 적용

데이터 | 요약

1. 총 개수: 18,000개
훈련 : **15,000**
테스트 : **3,000**

2. 크기
총 **191.82GB**

3. 종류
Dicom (이미지 및 고객 정보 포함)
CSV (질병명 및 질병 위치)



데이터 | 의료용 디지털 영상 파일의 통합 관리 표준의 총칭



이미지

```
Dataset.file_meta -----
(0002, 0000) File Meta Information Group Length UL: 154
(0002, 0001) File Meta Information Version OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID UI: Computed Radiography Image Storage
(0002, 0003) Media Storage SOP Instance UID UI: 0005e8e3701dfb1dd93d53e2ff537b6e
(0002, 0010) Transfer Syntax UID UI: Implicit VR LittleEndian
(0002, 0012) Implementation Class UID UI: 1.2.3.4
(0002, 0013) Implementation Version Name SH: 'OFFIS_DCMTK_360'

-----
(0010, 0040) Patient's Sex CS: 'F'
(0010, 1010) Patient's Age AS: '028Y'
(0028, 0002) Samples per Pixel US: 1
(0028, 0004) Photometric Interpretation CS: 'MONOCHROME2'
(0028, 0008) Number of Frames IS: "1"
(0028, 0010) Rows US: 3072
(0028, 0011) Columns US: 3072
(0028, 0030) Pixel Spacing DS: [0.140, 0.140]
(0028, 0034) Pixel Aspect Ratio IS: None
(0028, 0100) Bits Allocated US: 16
(0028, 0101) Bits Stored US: 14
(0028, 0102) High Bit US: 13
(0028, 0103) Pixel Representation US: 0
(0028, 1050) Window Center DS: "8483.0"
(0028, 1051) Window Width DS: "11160.0"
(0028, 2110) Lossy Image Compression CS: ''
(0028, 2112) Lossy Image Compression Ratio DS: None
(0028, 2114) Lossy Image Compression Method CS: ''
(7fe0, 0010) Pixel Data OW: Array of 18874368 elements
time : 3.132694959640503
```

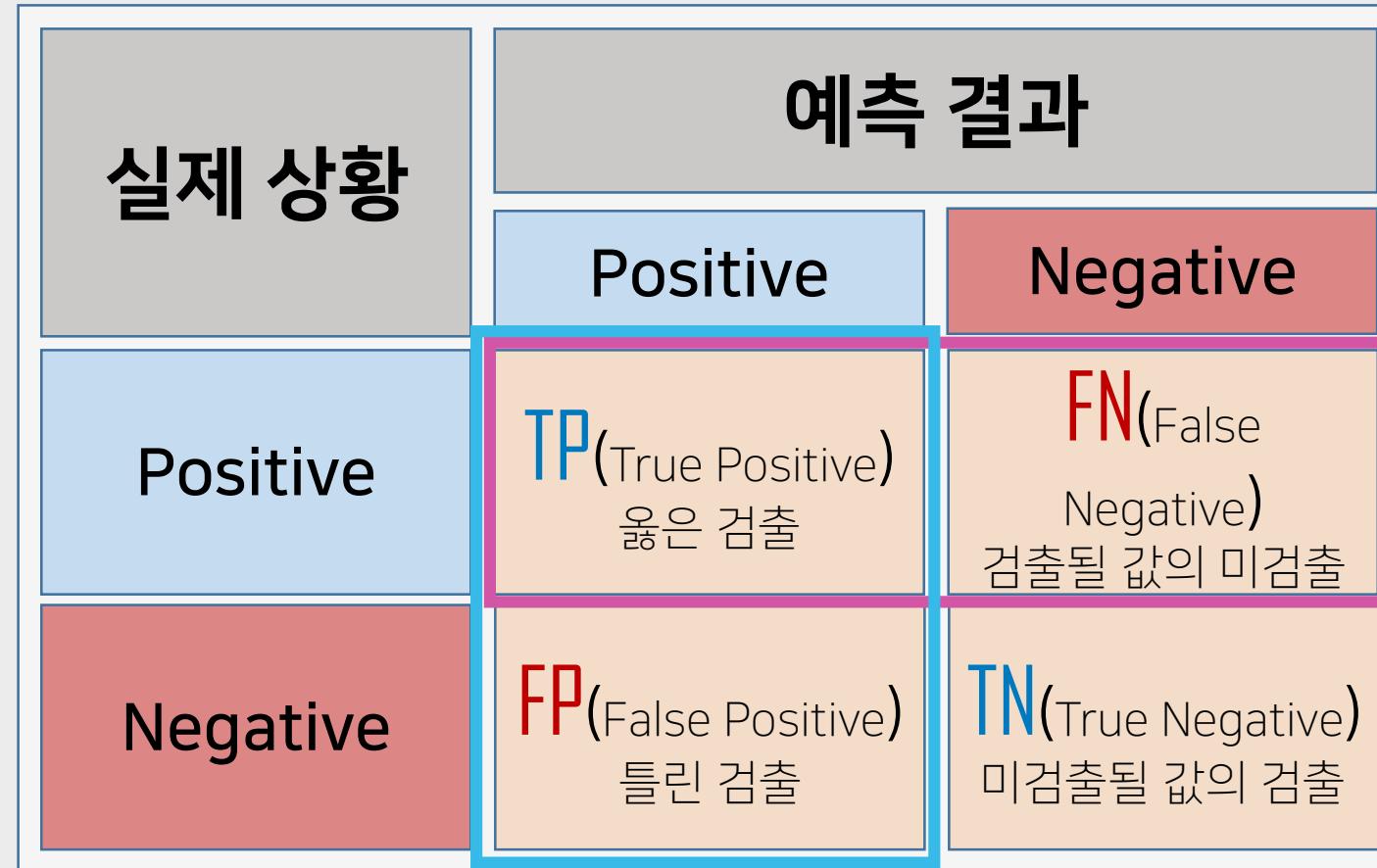
환자 정보(성별, 나이 등)

데이터 | 각 환자가 촬영한 X-ray의 전체적인 내용이 함축적으로 담긴 파일

환자번호	질병 종류	방사선사 번호	질병 위치(이미지 좌표)				
image_id	class_name	class_id	rad_id	x_min	y_min	x_max	y_max
50a418190	No finding	14	R11				
21a10246a	No finding	14	R7				
9a5094b25	Cardiomegaly	3	R10	691	1375	1653	1831
051132a77	Aortic enlargement	0	R10	1264	743	1611	1019
063319de2	No finding	14	R10				
1c32170b4	Pleural thickening	11	R9	627	357	947	433
0c7a38f29	ILD	5	R17	1347	245	2188	2169
47ed17dcb	Nodule/Mass	8	R9	557	2352	675	2484
d3637a193	Aortic enlargement	0	R10	1329	743	1521	958
afb623070	Pulmonary findings	13	R9	1857	1607	2126	2036
7c1add683	Lung Opacity	7	R10	600	1332	903	1523
18a61a07e	Pulmonary findings	13	R9	393	283	822	643
5550a493b	No finding	14	R9				
869f39afbd	No finding	14	R3				
321c11171	Aortic enlargement	0	R8	1292	554	1477	805
f55460fccf2	No finding	14	R6				
cdbacab6bf	No finding	14	R9				
80caa435b	Atelectasis	1	R9	331	462	1384	2365
Eda004778d	Aortic enlargement	0	R0	1025	672	1070	1151

주요 특징 파악

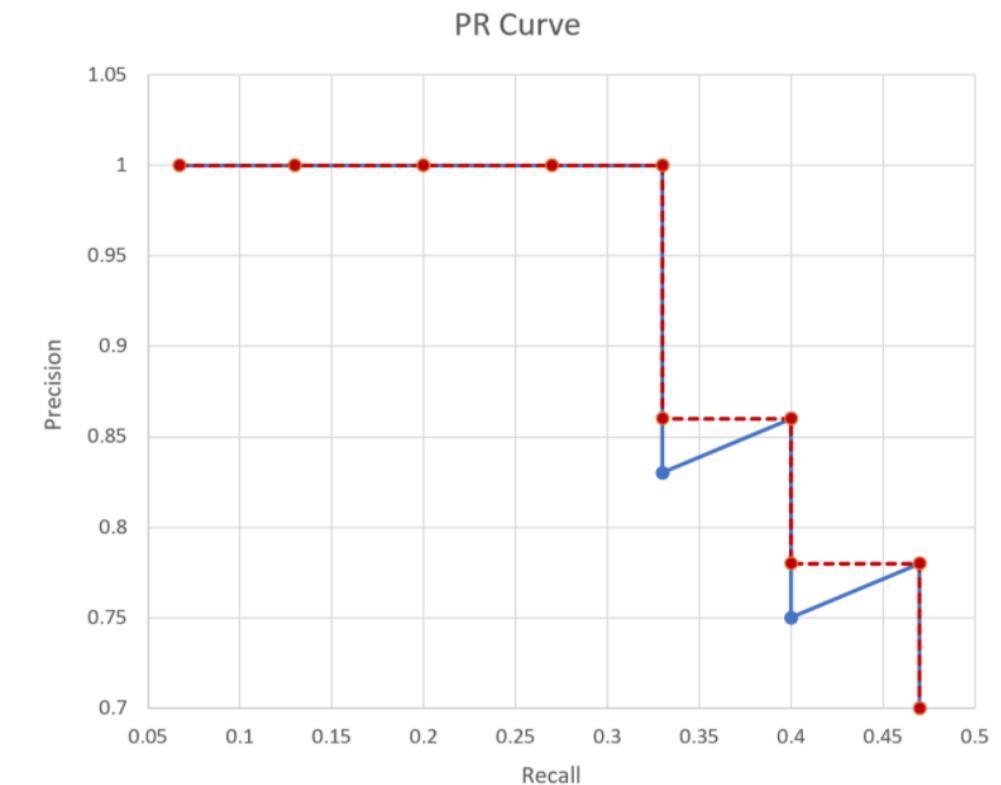
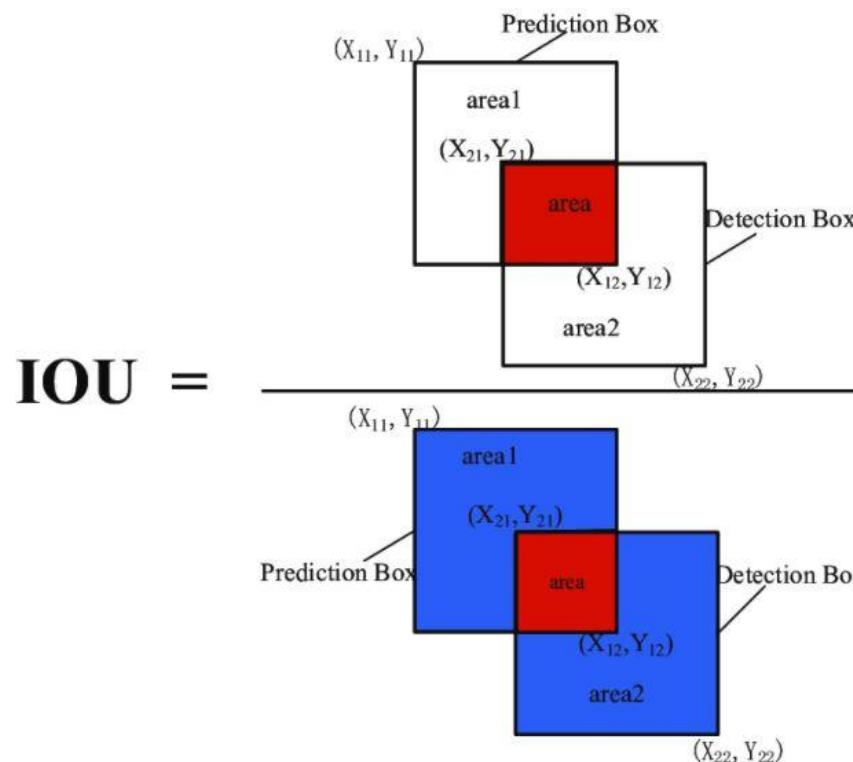
평가 지표 (TP/FN)



$$Precision = \frac{TP}{TP + FP} = \frac{TP}{All\ Detections}$$

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{All\ Ground\ truths}$$

평가 지표 (IOU)

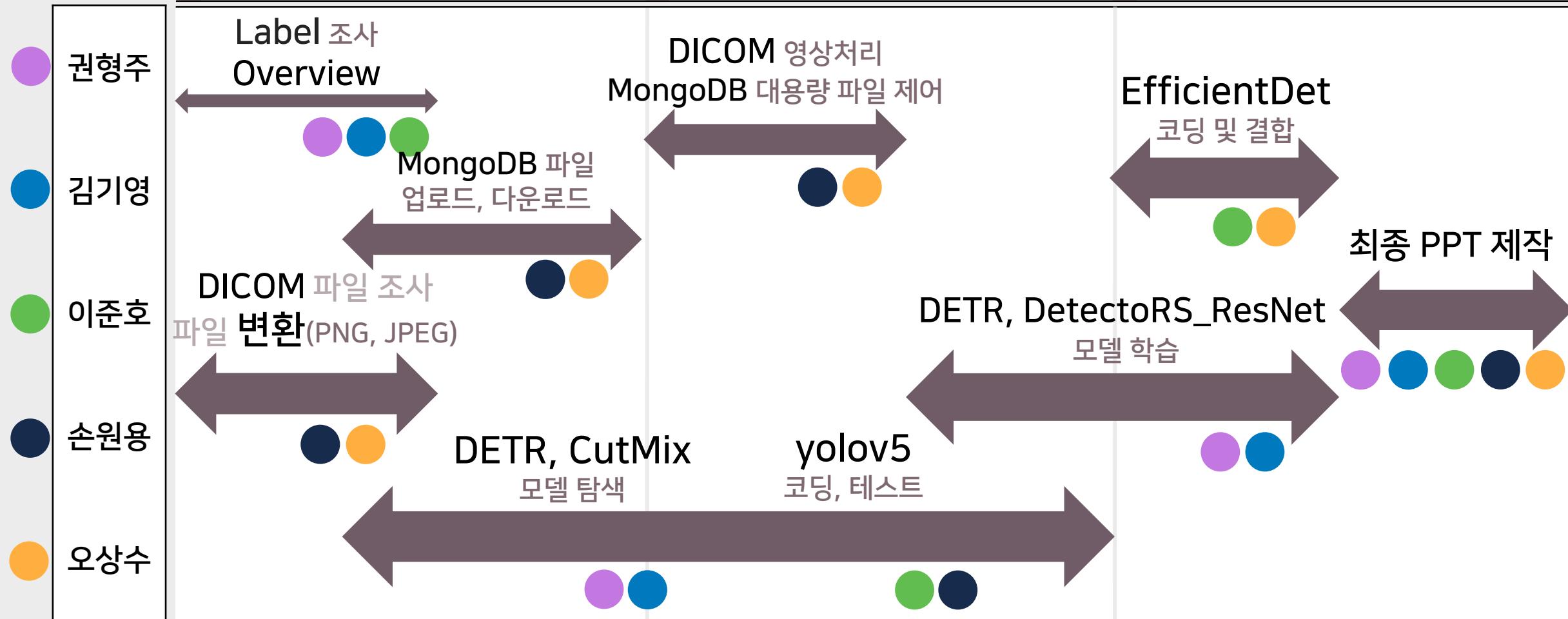


겹치는 면적 / 전체 면적

PR Curve

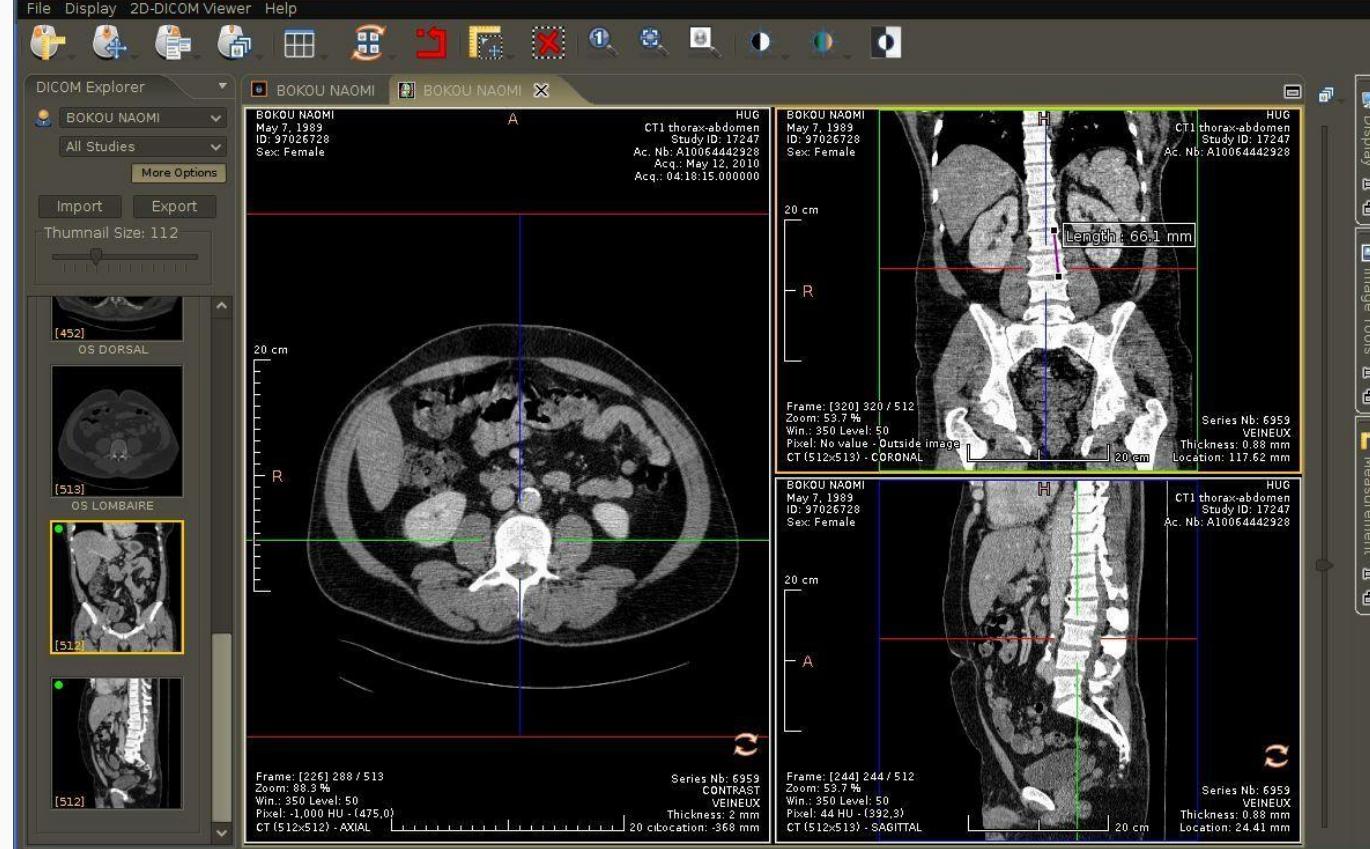
팀 일정

1주차	2주차	3주차
21년 2/8(월) ~ 12(금)	21년 2/15(월) ~ 19(금)	21년 2/22(월) ~ 26(금)



전처리 & EDA

Dicom | 의료용 디지털 영상 파일의 통합 관리 표준의 총칭



Dicom Viewer 화면



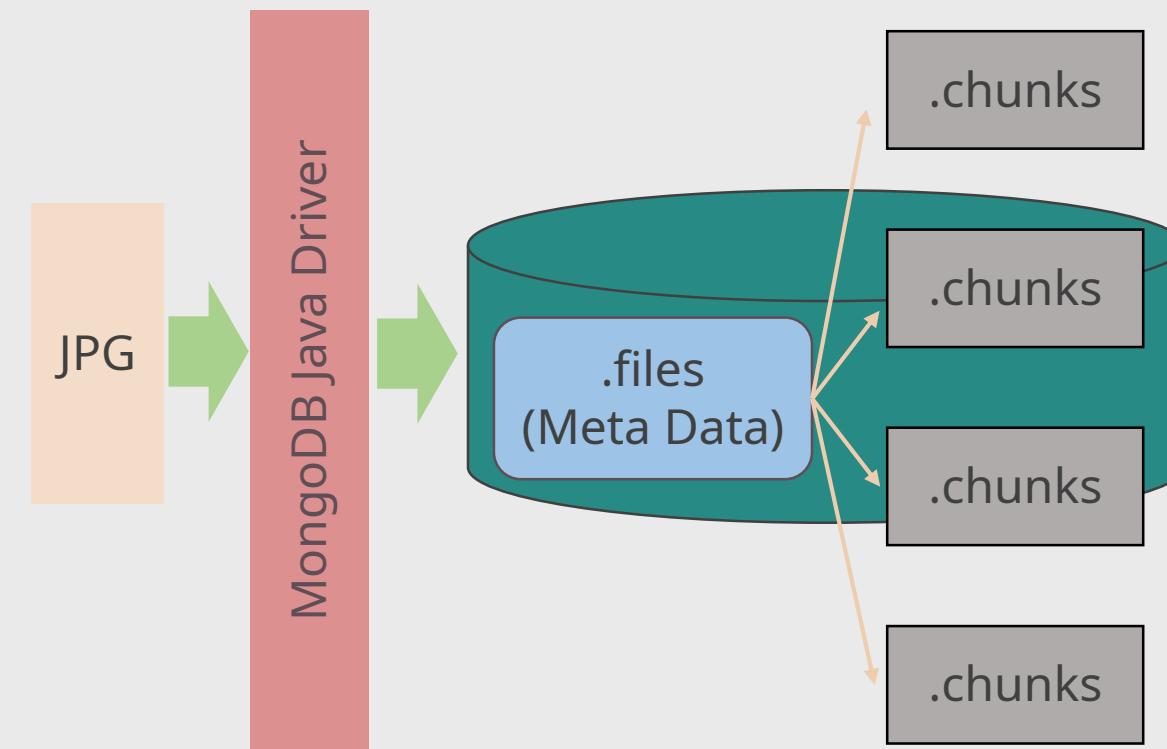
Dicom Viewer

MongoDB Atlas

대용량 이미지 파일을 Cloud에 저장 및 배포
GridFS 파일 시스템 활용

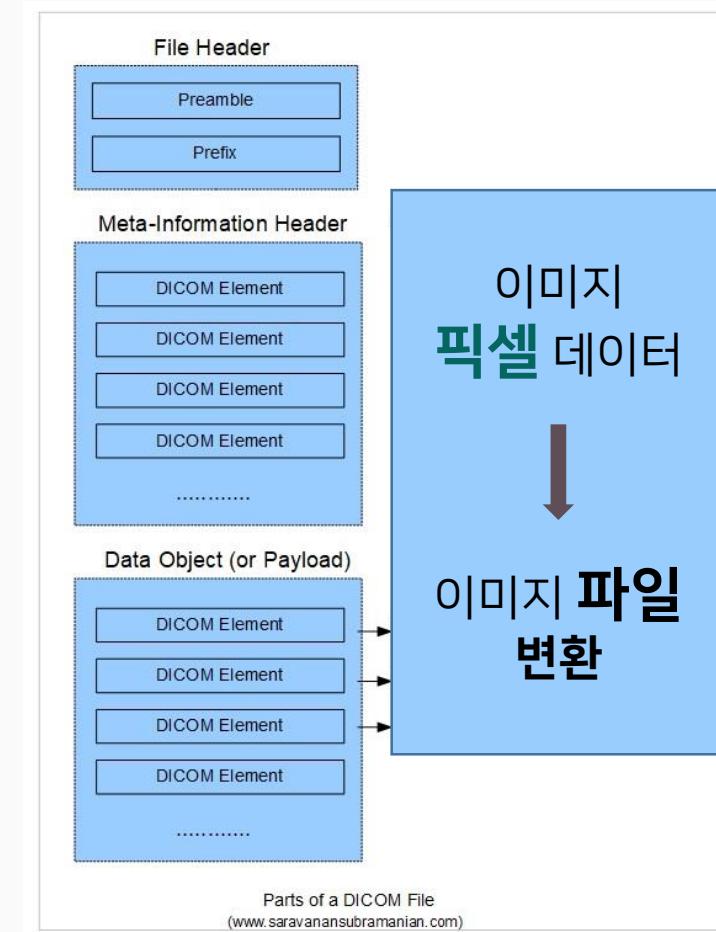
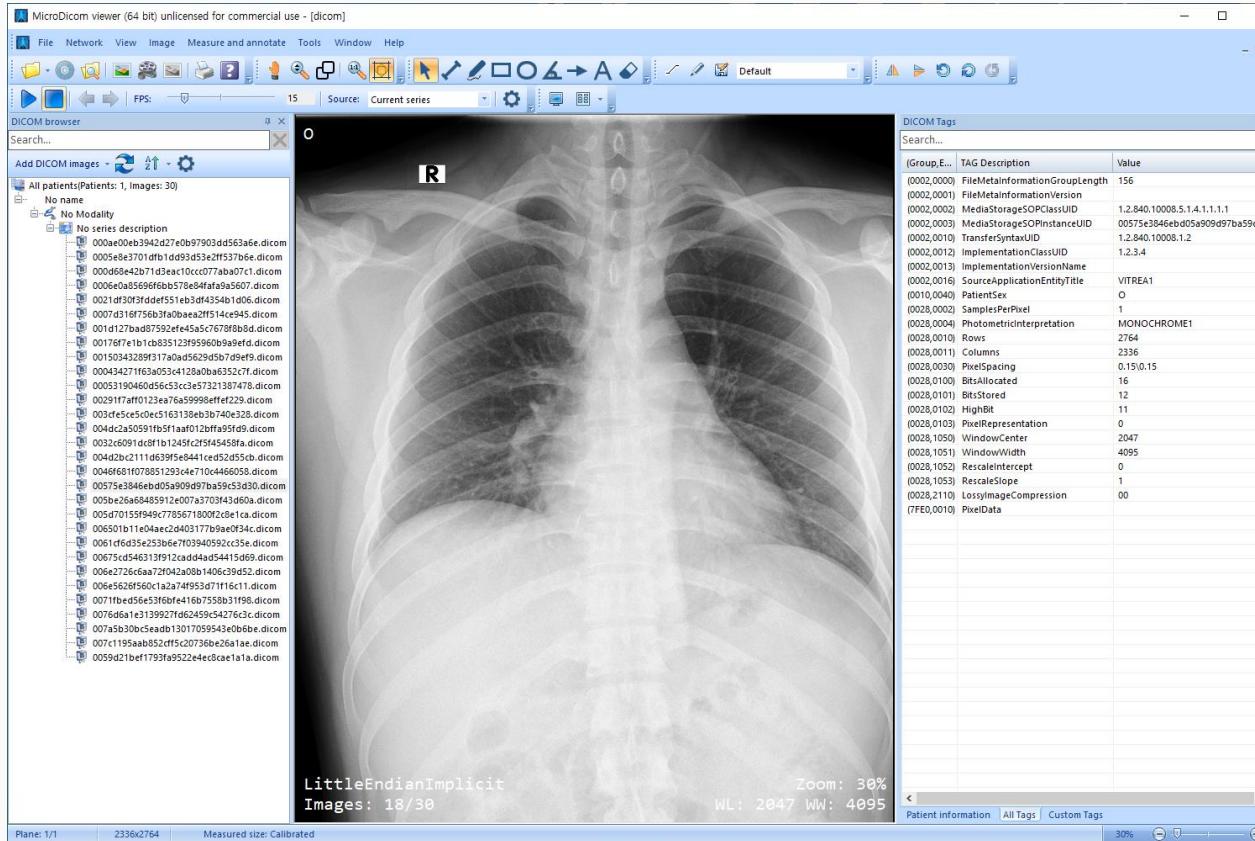
GridFS

MongoDB에서 JSON 문서 크기 제한인 **16MB**를 초과하는 파일을 저장하고 검색하기 위한 기능



몽고DB GridFS 구조

Image | DICOM은 의료용 디지털 영상파일이므로 학습 가능한 이미지가 실제로 필요 convert



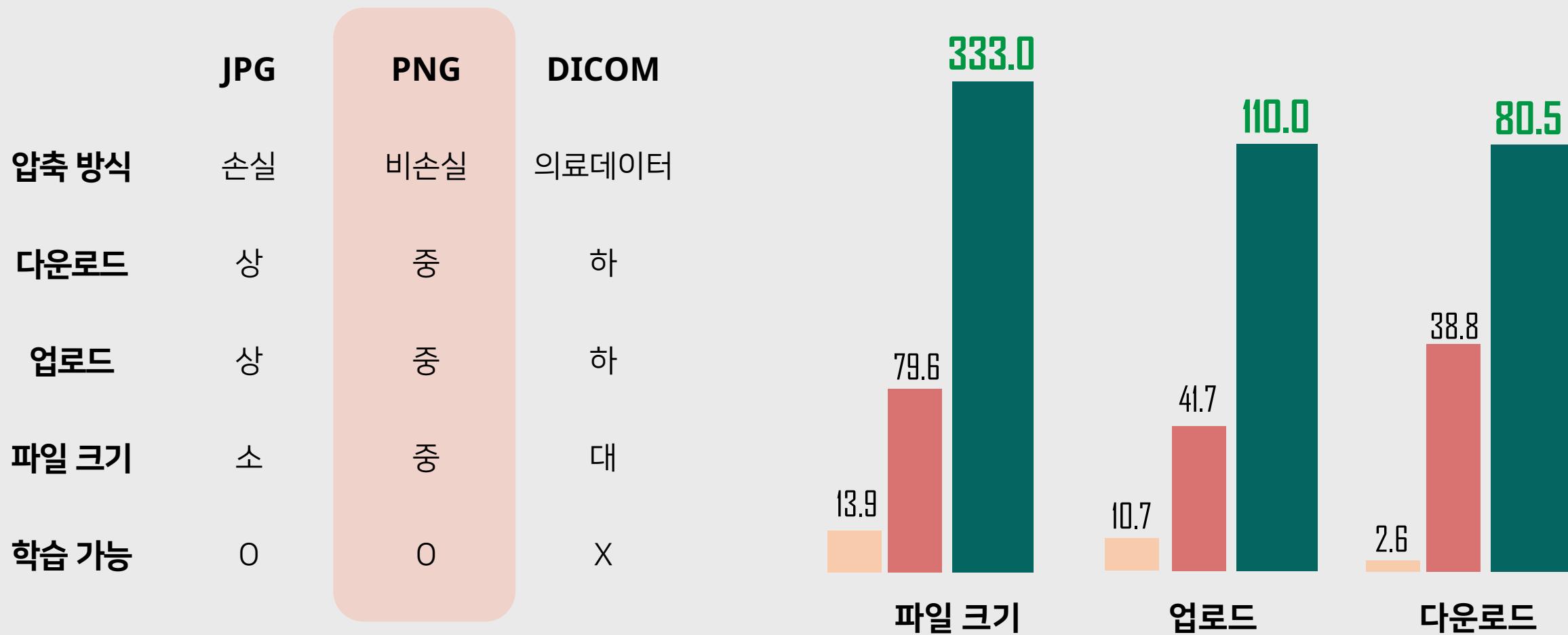
데이터 전송

| 성능 측정 - 최적의 포맷 방식 선정

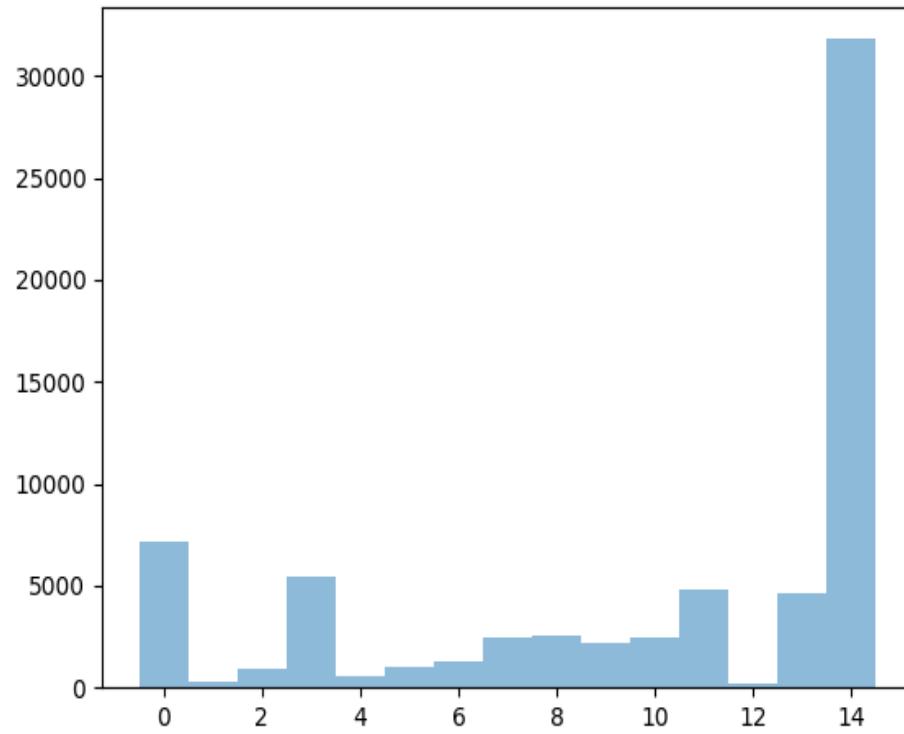
JPG

PNG

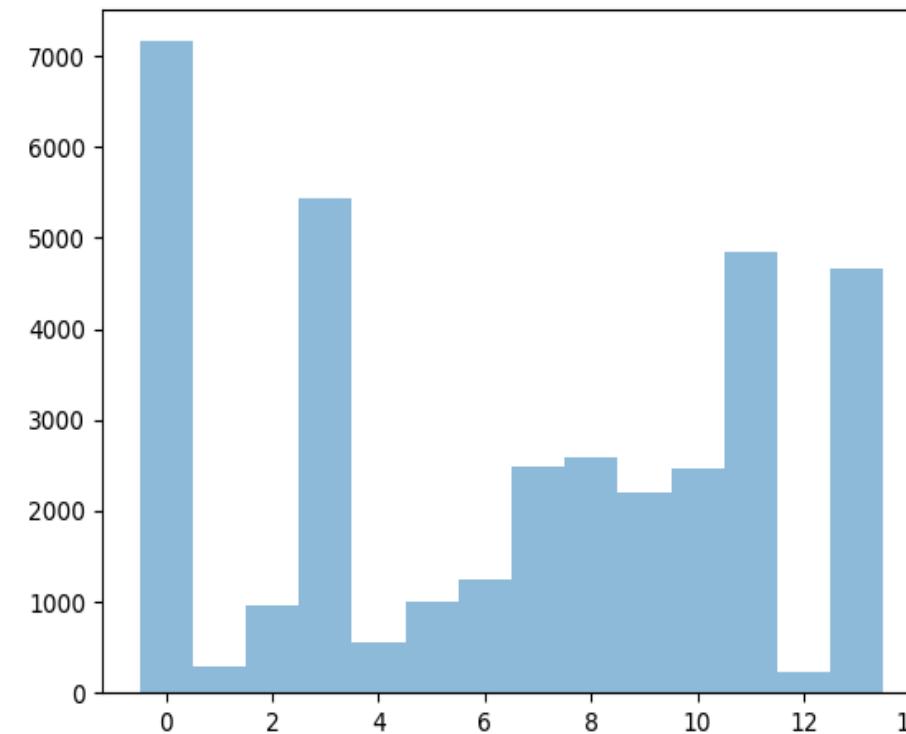
DICOM



EDA | 데이터 중에 정상 데이터가 다수 분포 (약 70.7 퍼센트)



전체 분포



정상을 제외한 분포

라벨 분석 | 총 14개의 질병(병변)에 대한 X-ray 특징 조사

Cardiomegaly 심비대

Aortic enlargement 대동맥 확장

Consolidation 간경화

ILD 간질성 폐질환

Lung Opacity 폐 불투명도

Other lesion 기타 병변

Infiltration 침투

무기폐 Ateletasis

흉막 삼출 Pleural effusion

늑막 비후 Pleural thickening

종양 Nodule/Mass

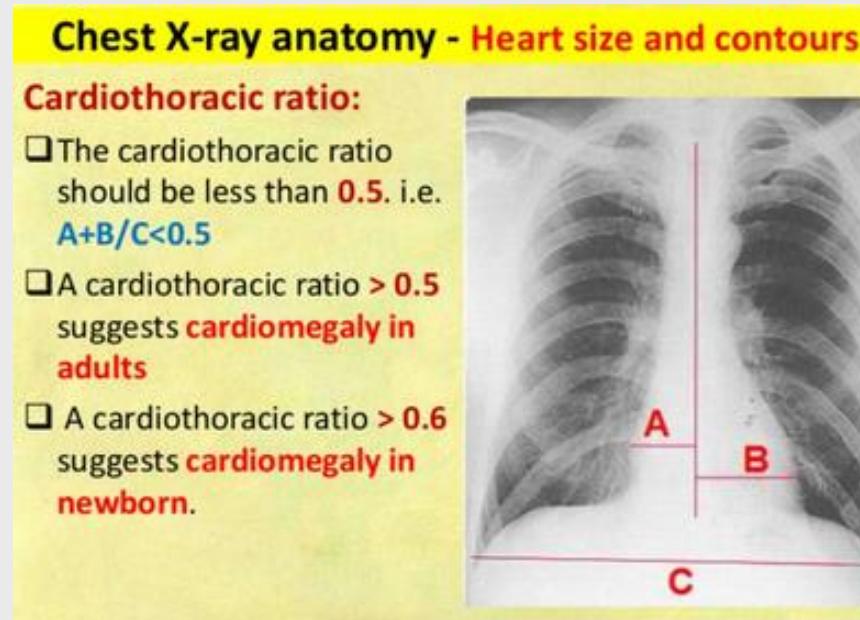
폐 섬유화 Pulmonary fibrosis

석회화 Calcification

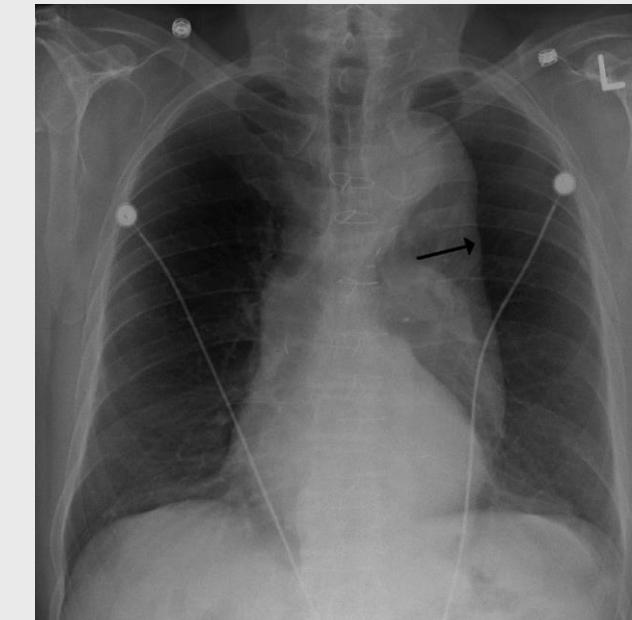
기흉 Pneumothorax

라벨 분석 | 특징 1

데이터 분포 상에서 가장 다수 진단



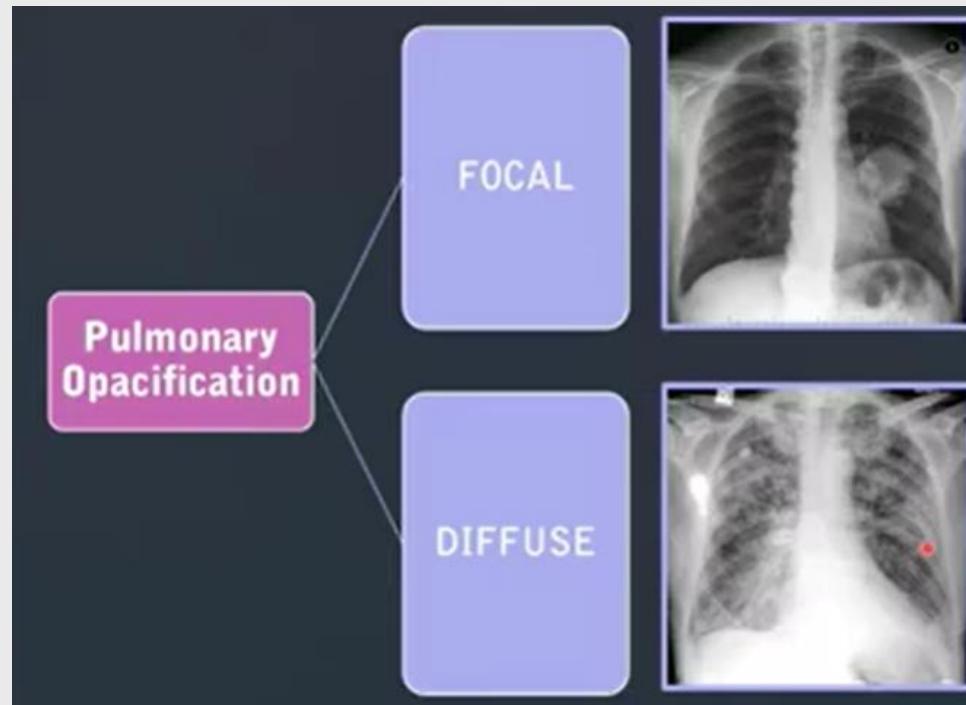
심비대



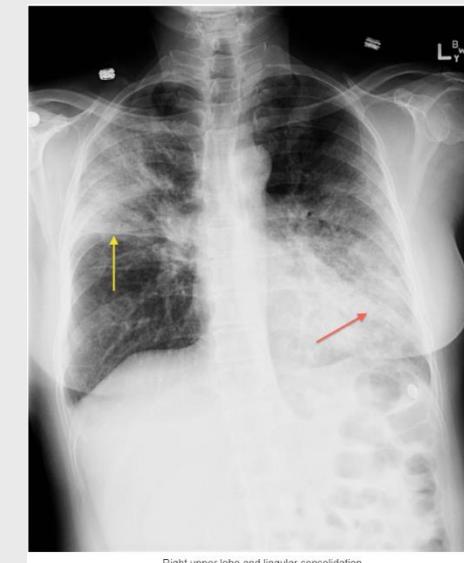
대동맥 확장

라벨 분석 | 특징 2

침투(Infiltration) 하위 개념

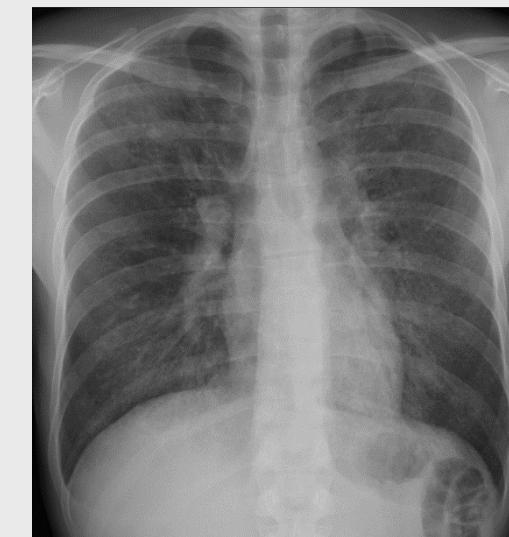


폐 불투명도



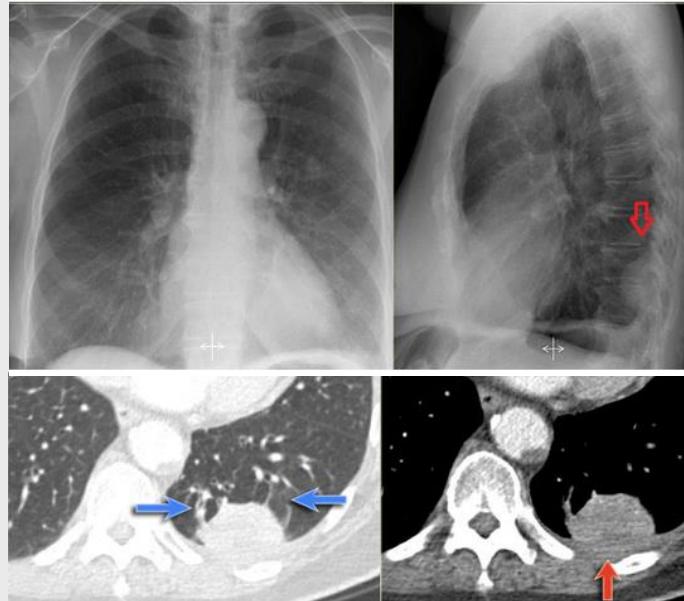
간경화

간질성 폐질환



라벨 분석 | 특징 3

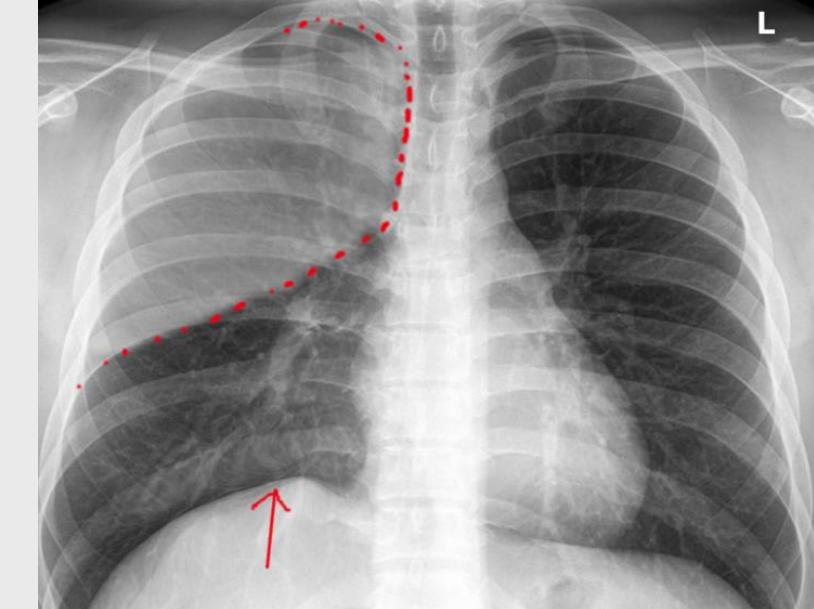
X-ray에서 보이는 모양 흡사



늑막 비후



흉막 삼출

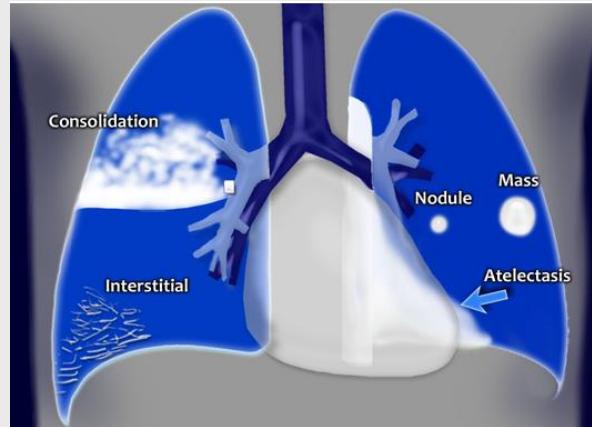


무기폐

라벨 분석 | 나머지 특징

Other lesion 기타 병변

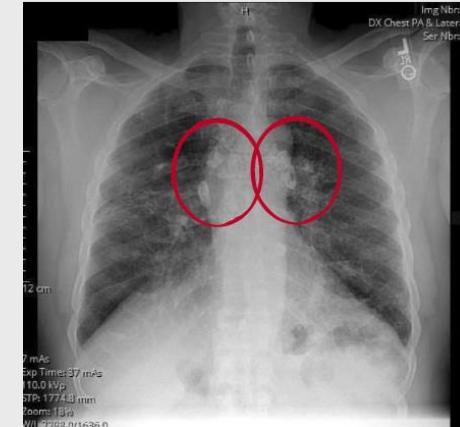
앞서 이야기한 질병과 다른 상이한 이물질이 보일 경우
방사선사가 판단하기에 모호한 질병인 경우 등을 포함



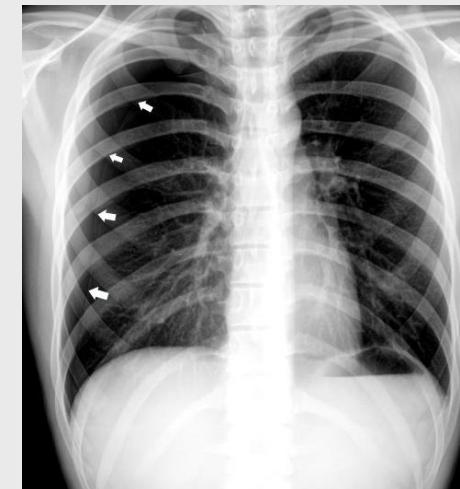
종양



섬유화



석회화



기흉

EDA | 데이터 분석 결과 겹치는 증상이 다수 발견

1. 증상 별 상관관계 분석

(IOU 0.4 기준) 4,352개의 이미지에 증상 중복

2. 겹치는 증상 Top 3

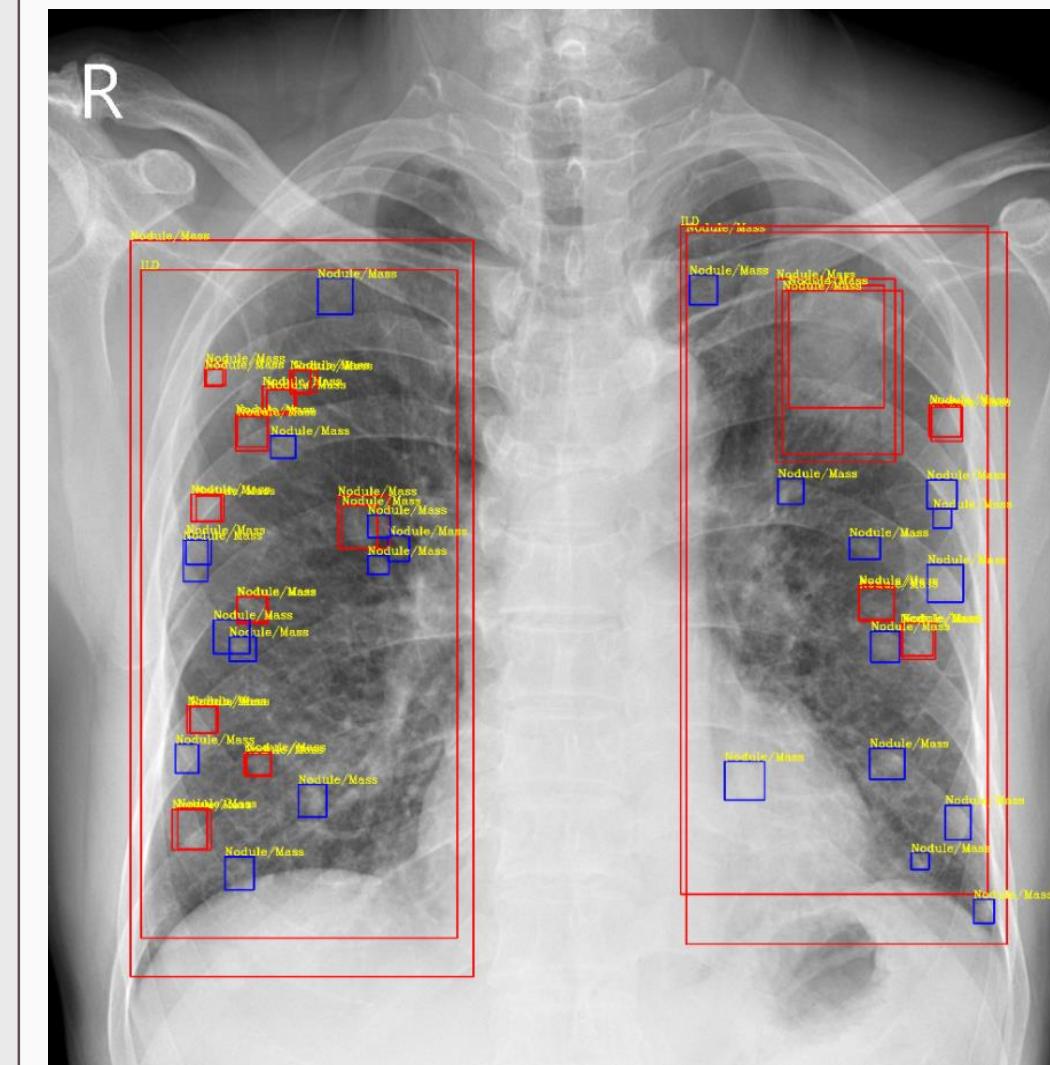
- 1) 대동맥 확장 + 심비대
- 2) 흉막 삼출 + 늑막 비후
- 3) 늑막 비후 + 폐 섬유화

대략 1,252개의 케이스에 대해 증상 중복

	overlap	count
221	Aortic enlargement + Cardiomegaly	3847
94	Pleural effusion + Pleural thickening	479
		⋮

3,847

479



알고리즘 소개

Yolo

모델 소개



You Only Look Once
Unified, Real-Time Object Detection 15.06

특장점 : 빠르고 가볍다.

You Only Look Once
이미지 전체를 단 한 번만 본다.

Unified
통합된 모델

Real-time Object Detection
실시간 객체 탐지

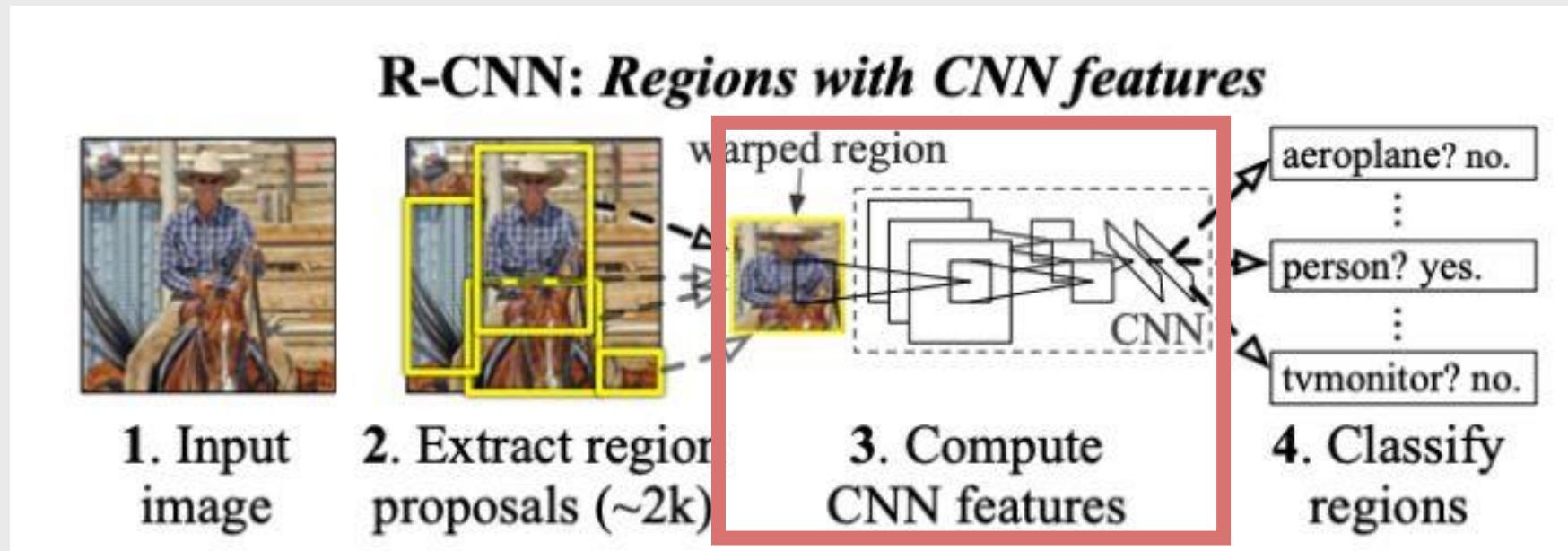
- 조셉 레드몬

Yolo

모델 이론

R-CNN

| 기존의 R-CNN은 높은 정확도를 가진 CNN을 사용



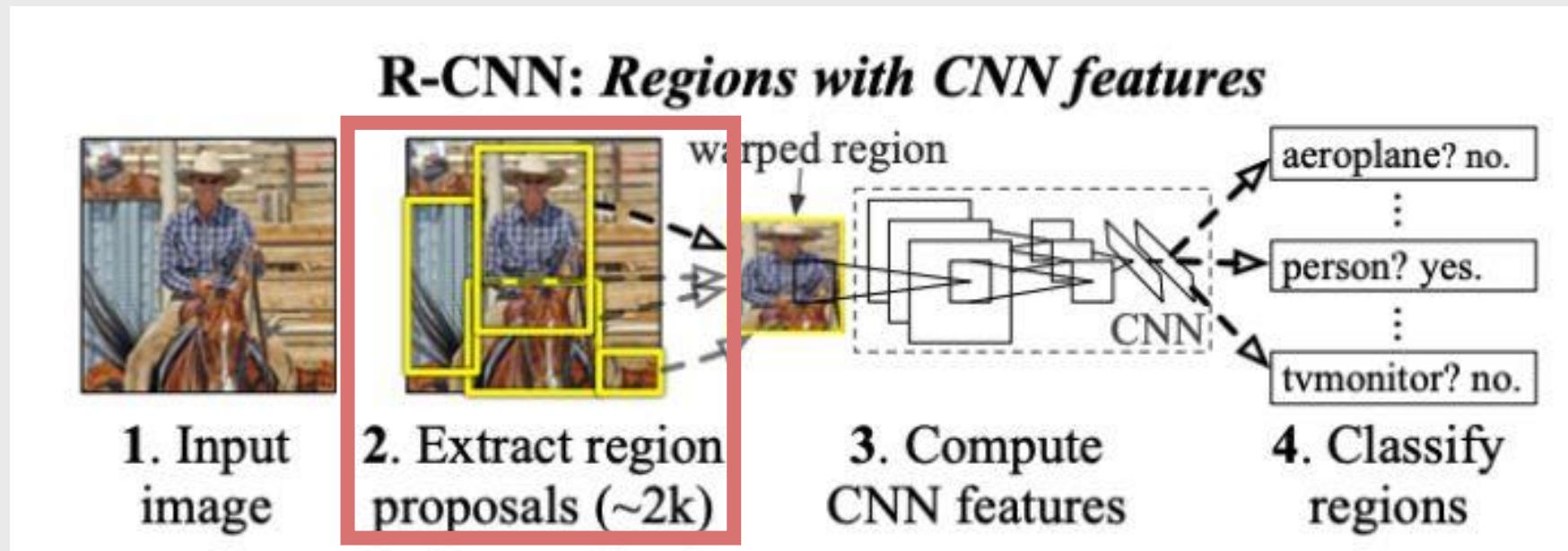
Region-based CNN(R-CNN) 시스템 모델
CNN : 높은 정확도를 가진 image
classification model

Yolo

모델 이론

R-CNN

| 입력한 이미지에 선택적 알고리즘을 적용해 서브 이미지를 추출



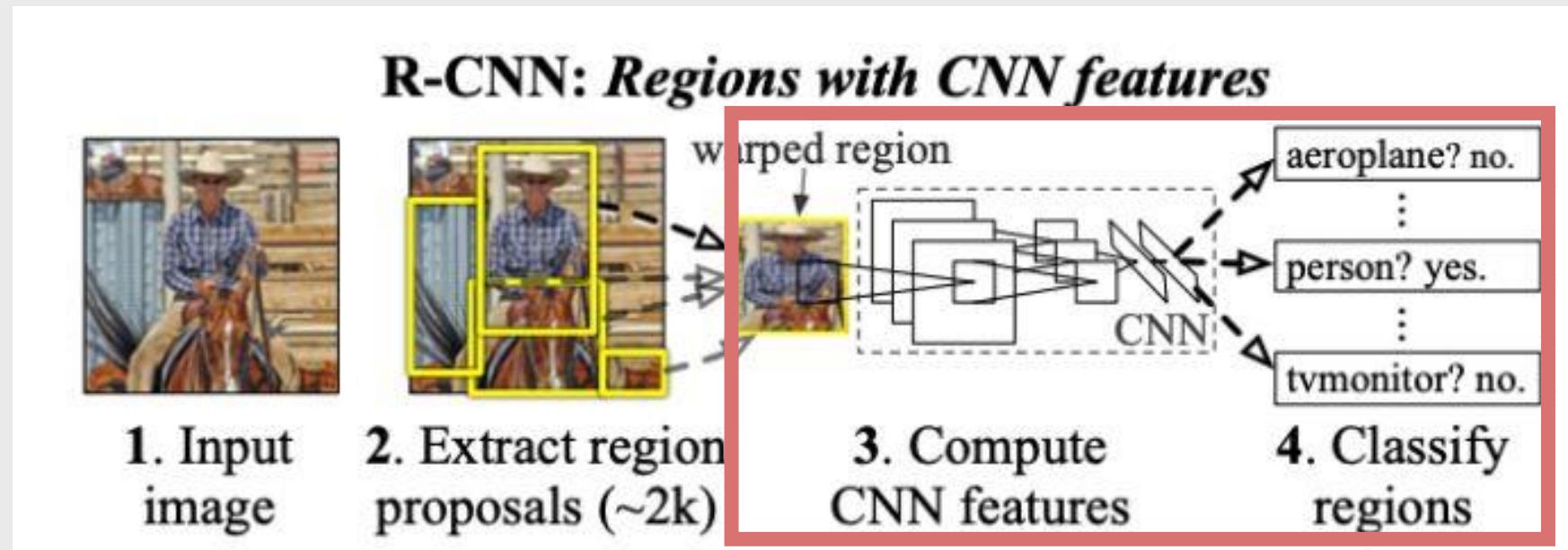
선택적 알고리즘을 이용해
최대 2000개의 서브 이미지 추출

Yolo

모델 이론

R-CNN

추출된 서브 이미지가 CNN을 통하여 특징을 추출
이 추출된 특징들이 SVM을 통해 이미지 별로 클래스를 추출

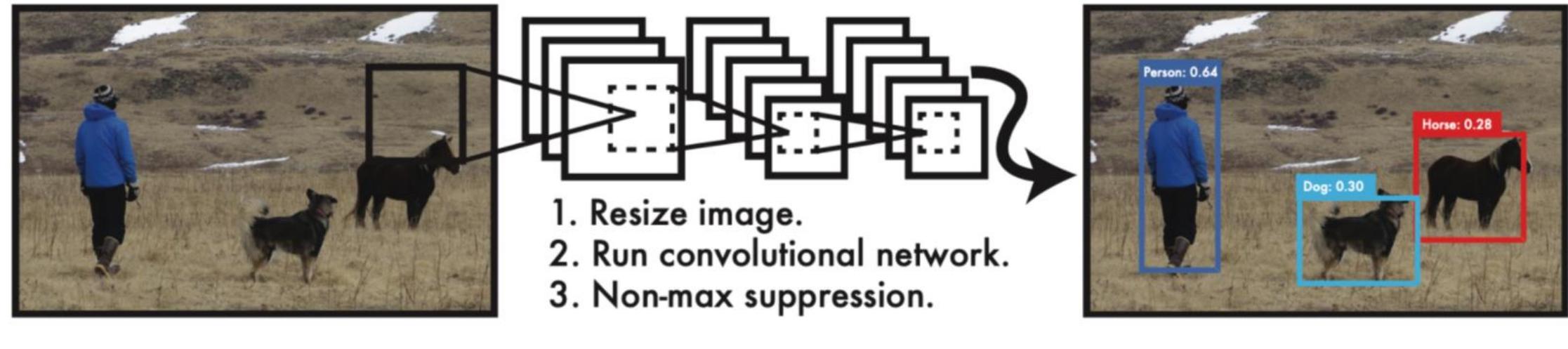


CNN - 특징 추출

SVM - 이미지 별 클래스 추출

Yolo 모델 이론 System

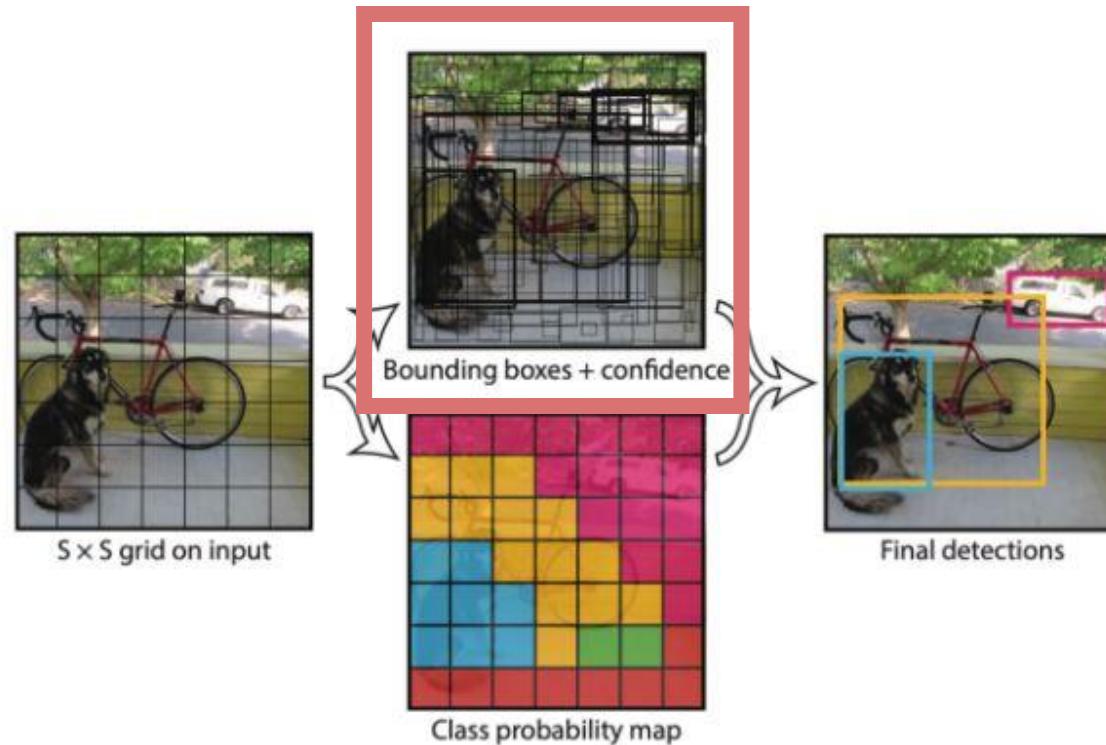
R-CNN과 달리 이미지 1개를 통과
각 객체의 바운딩 박스와 분류 확률이 출력됨



R-CNN과 달리 CNN에 이미지를 단 1개 통과
각 객체의 바운딩 박스, 분류 확률 출력
NMS(Non-max suppression)
최종 위치 결정

Yolo 모델 이론 System

Yolo는 두 가지 작업을 동시에 실행하는 데,
그 중 첫번째가 바운딩 박스와 박스 안에 물체가 존재할 확률을 예측



Hidden Layer 1-1

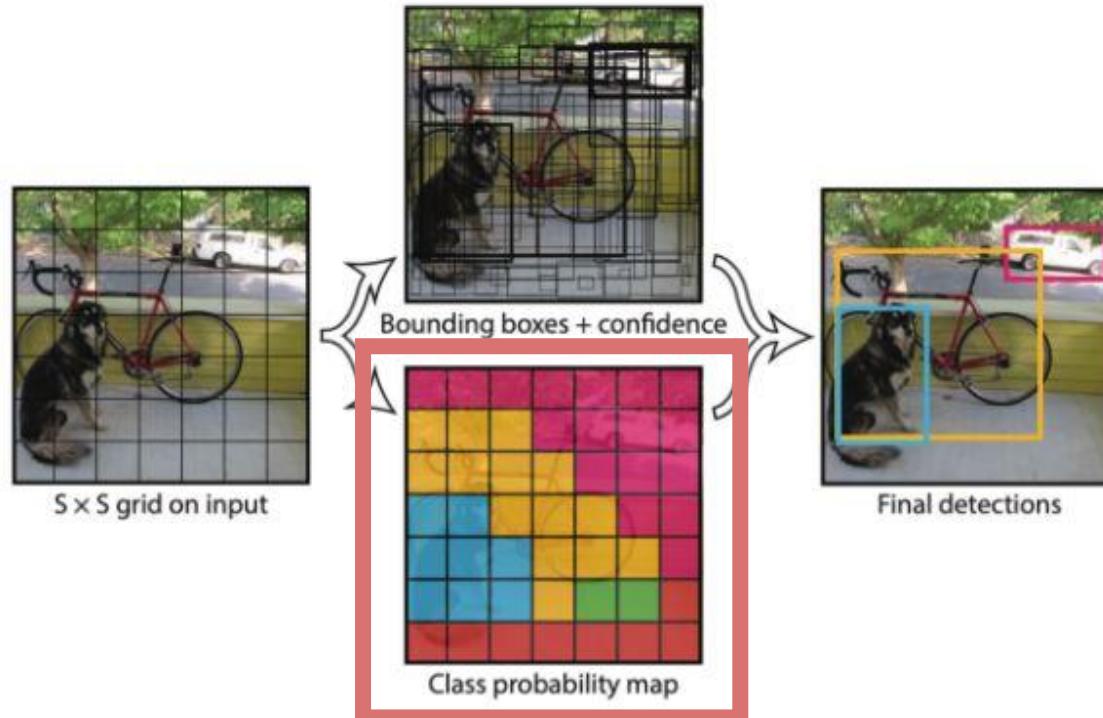
바운딩 박스 + Confidence Score
X,Y 좌표, 가로, 세로 크기
Confidence Score

$$\text{Confidence Score} = \text{Pr}(\text{Obj}) * \text{IOU}$$

$\text{Pr}(\text{Obj})$:
물체가 바운딩 박스 안에 존재할 확률

Yolo 모델 이론 System

| 2번째 작업은 각 그리드 셀 별로 클래스 분류 확률을 예측



Hidden Layer 1-2

셀 별 클래스 분류 확률

$$\Pr(\text{Class}_i \mid \text{Obj})$$

객체를 포함하는 조건부 확률
배경이었다면 0

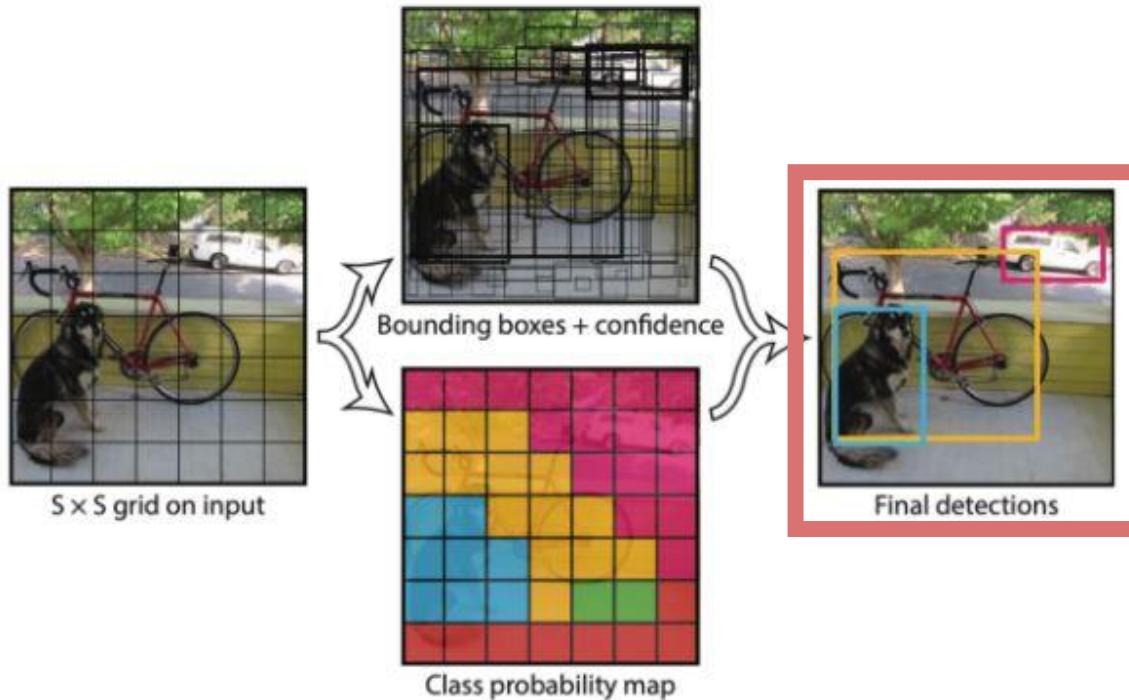
Yolo 모델 이론 System

| 최종으로 예측한 바운딩 박스와 두 확률을 곱하여 만들어진 클래스 확률을 출력

Output Layer

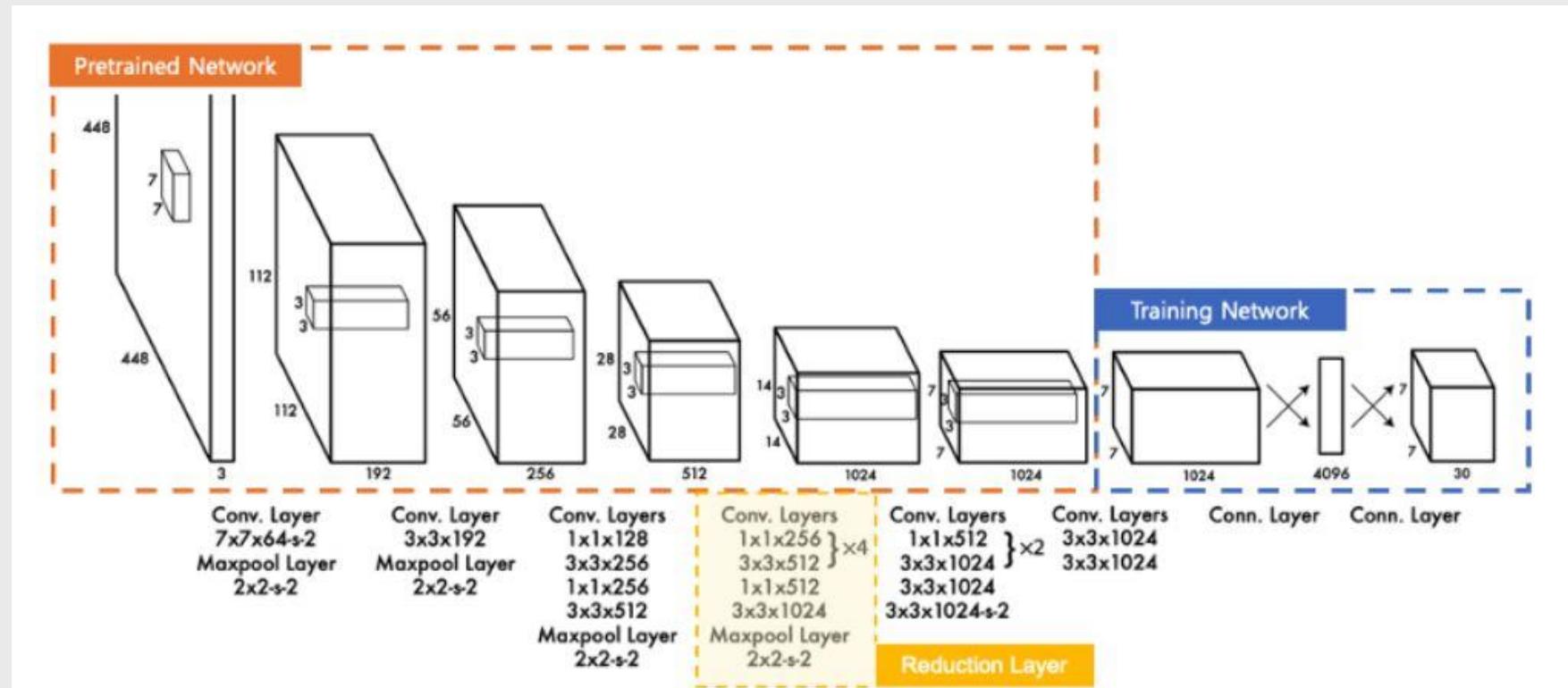
최종 결과값
바운딩 박스 내의 셀을 이용

$$\Pr(\text{Class}_i \mid \text{Obj}) * \Pr(\text{Obj}) * \text{IOU} = \Pr(\text{Class}_i) * \text{IOU}$$



Yolo 모델 이론 Network

| Yolo Network는 선행학습 모델과 학습할 모델이 나뉘어져 있음



**24 Convolutional Layers
2 Fully-Connected Layers
Pre-trained Network & Training Network
Reduction Layer**

Yolo

모델 이론

Loss func

$$\begin{aligned}
 & \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
 & + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
 & + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
 & + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (3)
 \end{aligned}$$

Loss Function 3 Points

1. 이미지 분류 문제를 바운딩 박스를 만드는 **Regression** 문제로 판단 - SSD
2. Localization Error와 Classification Error의 패널티를 다르게 평가
(Localization Error에 더 **높은 패널티**를 부과)
- λ_{coord} , λ_{noobj}
3. 많은 바운딩 박스 중 IOU 수치가 **가장 높게 생성**된 바운딩 박스만 학습에 참여

Yolo

모델 이론

Loss func

실시간 객체 탐지에는 압도적인 성능을 보이며,
이미지 객체 탐지에서는 Faster R-CNN과 유사한 성능이다.

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
<hr/>			
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Real-Time Detectors

압도적인 성능

Less Than Real-Time

Faster R-CNN과 유사한 성능 수치(향상)

Yolov2, 3

모델 버전

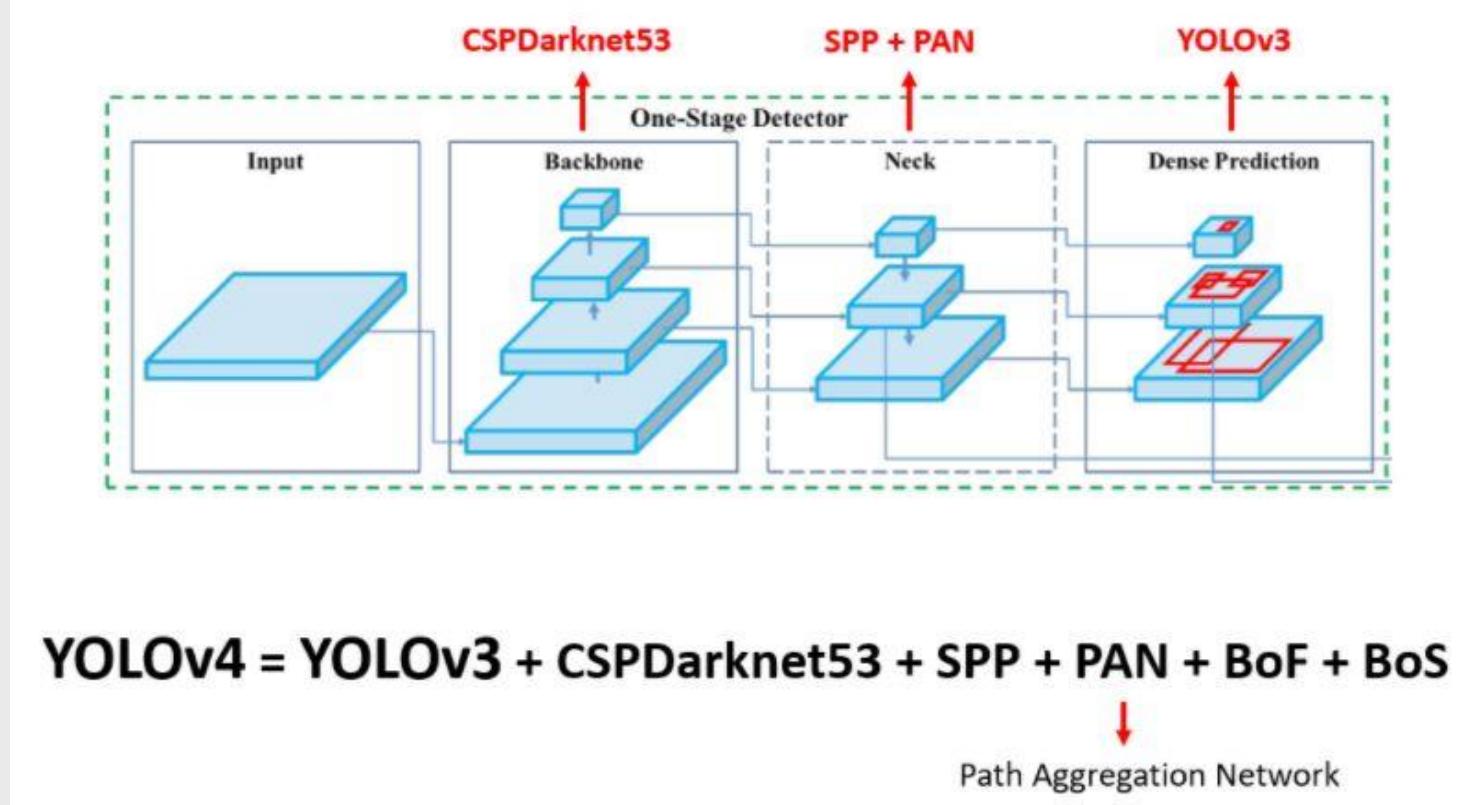
YOLO9000: Better, Faster, Stronger (Yolov2) 16.12
9000개의 이미지를 탐지하여 분류할 수 있는 모델 개발

Yolov3 18.04
ResNet의 Residual Block 기법을 사용
106개의 신경망 층 구성

Yolov4

모델 버전

Yolov4는 Yolov3에서 Darknet, SPP, PAN 등을 결합한 모델



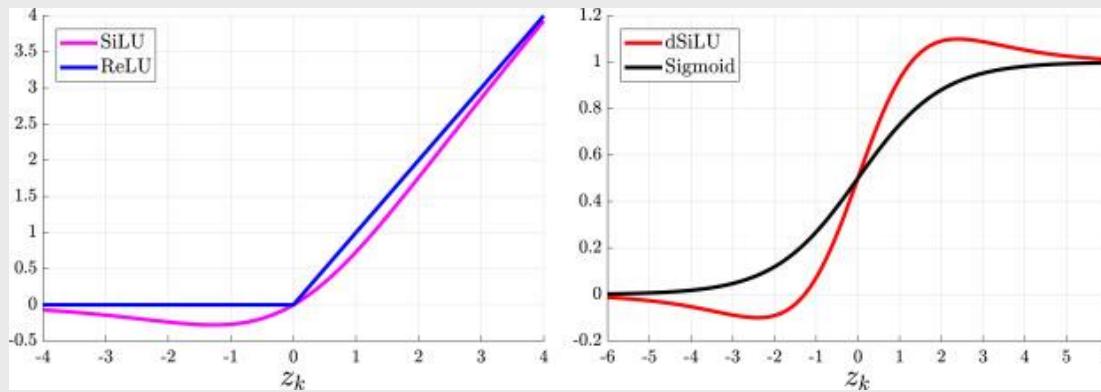
Yolov5

모델 버전

Yolov5는 Yolov4에서 LeakyReLU 함수를 SiLu로 변경한 모델

$$silu(x) = x * \sigma(x), \text{ where } \sigma(x) \text{ is the logistic sigmoid.}$$

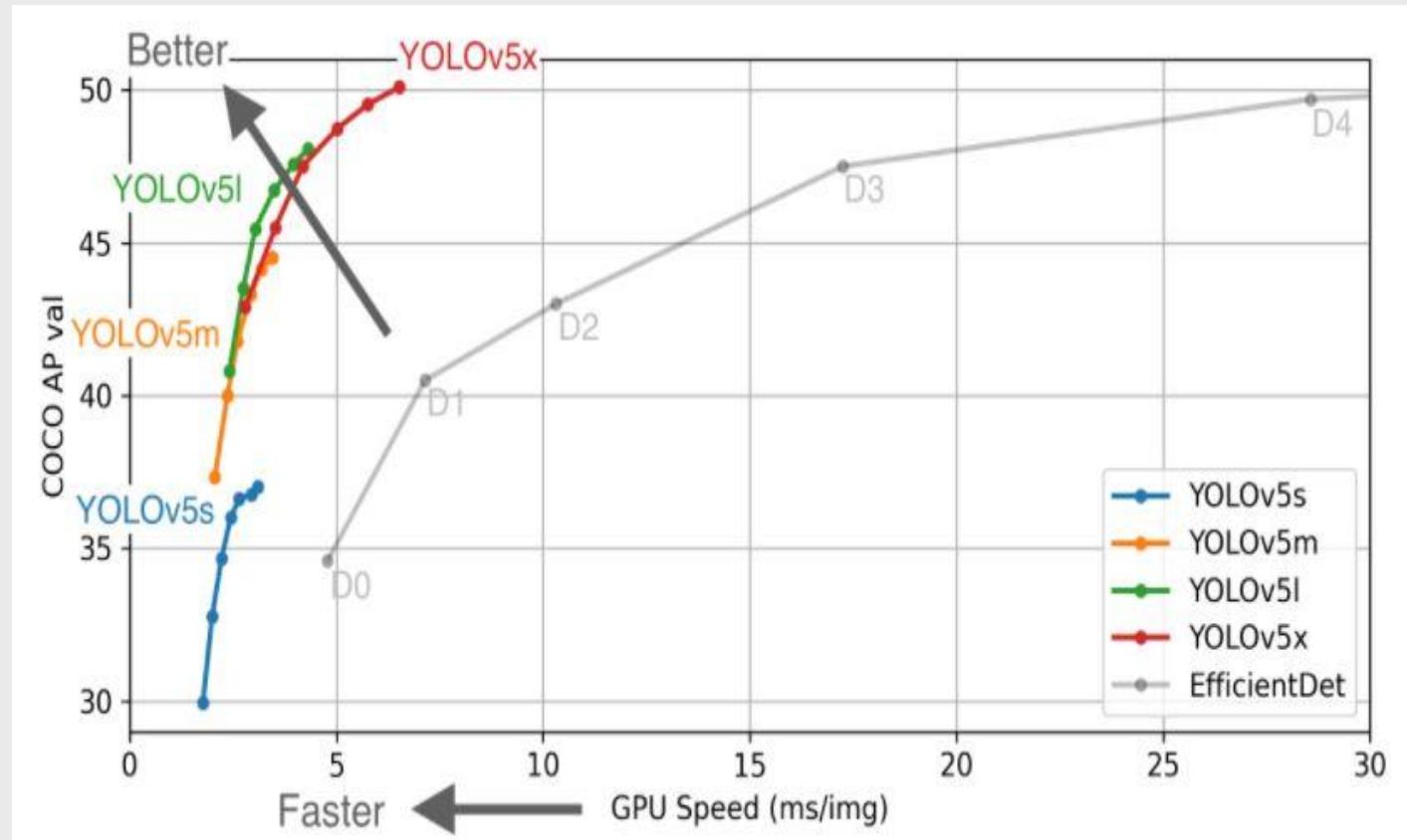
모델 안의 활성화 함수
nn.LeakyReLU
nn.SiLU로 변경



Yolov5

모델 버전

EfficientDet 보다 훨씬 빠른 트레이닝 속도를 가지고 있고,
가장 무거운 모델인 Yolov5x 경우 EfficientDet과의 성능이 비등하다.



Yolov5

모델 종류

모델 별 parameter 개수와 Speed, FPS 차이가 크므로
분석 목적에 따라 올바른 모델을 사용해야 한다.

Model	size	AP ^{val}	AP ^{test}	AP ₅₀	Speed _{V100}	FPS _{V100}	params	GFLOPS
YOLOv5s	640	36.8	36.8	55.6	2.2ms	455	7.3M	17.0
YOLOv5m	640	44.5	44.5	63.1	2.9ms	345	21.4M	51.3
YOLOv5l	640	48.1	48.1	66.4	3.8ms	264	47.0M	115.4
YOLOv5x	640	50.1	50.1	68.7	6.0ms	167	87.7M	218.8
YOLOv5x + TTA	832	51.9	51.9	69.6	24.9ms	40	87.7M	1005.3

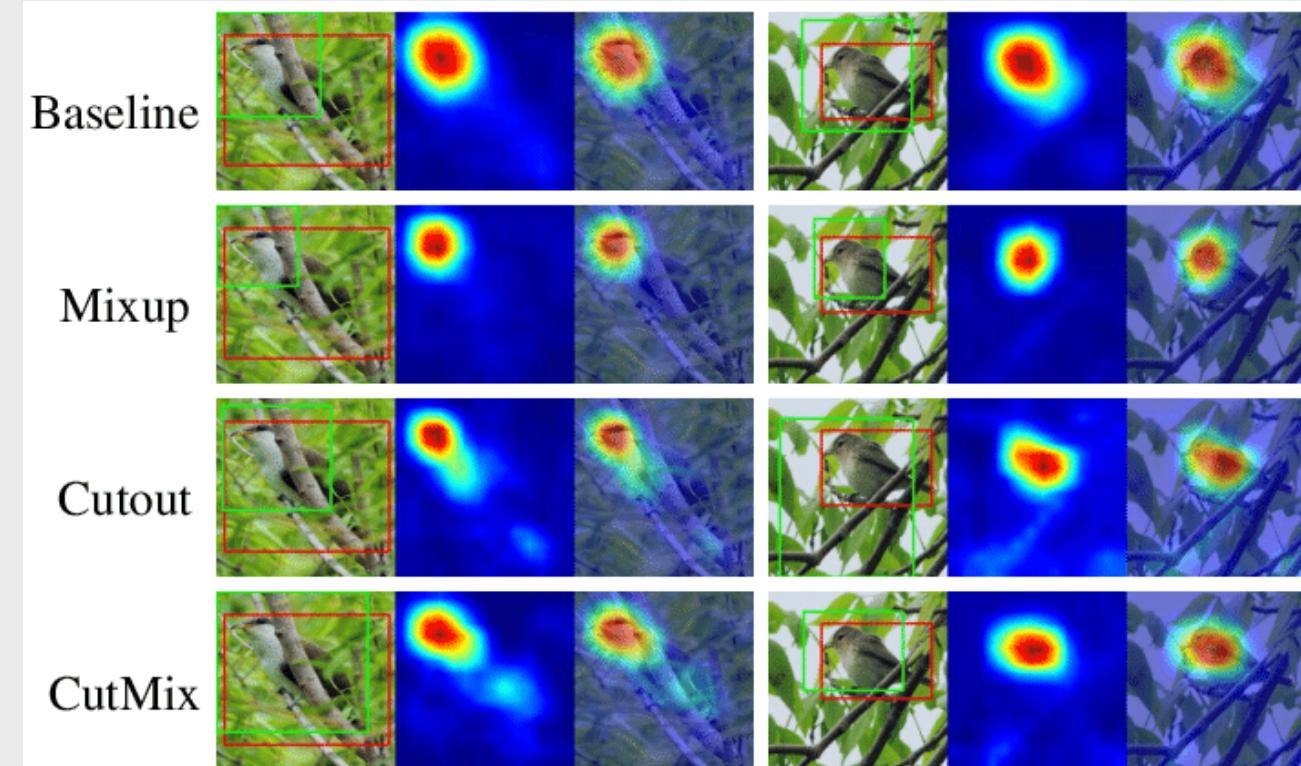
CutMix

모델 설명

분류할 샘플 이미지를 기반으로 임의로 새로운 샘플을 생성하여 기존 분류에서 낮은 특성값을 지닌 부분도 분류 대상의 큰 특징으로 인식하게 도와주는 모델

	ResNet-50	Mixup	Cutout	CutMix
Image				
Label	Dog 1.0	Dog 0.5 Cat 0.5	Dog 1.0	Dog 0.6 Cat 0.4
ImageNet Cls (%)	76.3 (+0.0)	77.4 (+1.1)	77.1 (+0.8)	78.4 (+2.1)
ImageNet Loc (%)	46.3 (+0.0)	45.8 (-0.5)	46.7 (+0.4)	47.3 (+1.0)
Pascal VOC Det (mAP)	75.6 (+0.0)	73.9 (-1.7)	75.1 (-0.5)	76.7 (+1.1)

데이터셋 별 성능 지표 결과 비교



이미지 Box Dropout 방식에 따른 예측 범위 확인

MMDetection

모듈 설명

Object Detection 및 Instance Segmentation을 다루는 모델들을 하나의 toolbox로 표현한 플랫폼 (2018 코코 챌린지 우승 / 2019 논문 발표)

장점

1. 모듈화 - 사용자 제작이 간편
2. 여러 가지 형태로 제공되는 Framework 존재
3. 높은 효율성. Detectron과 같은 다른 플랫폼 대비 학습 속도 ↑

모델

Support backbones (7개)

- ResNet(CVPR'2016), ResNeXt(CVPR'2017), ...

Supported methods (42개)

- RPN(NeurIPS'2015), Faster R-CNN(NeurIPS'2015), ...

General Modules and Methods (15개)

- Mixed Precision Training, M2Det, BN, ...

구조

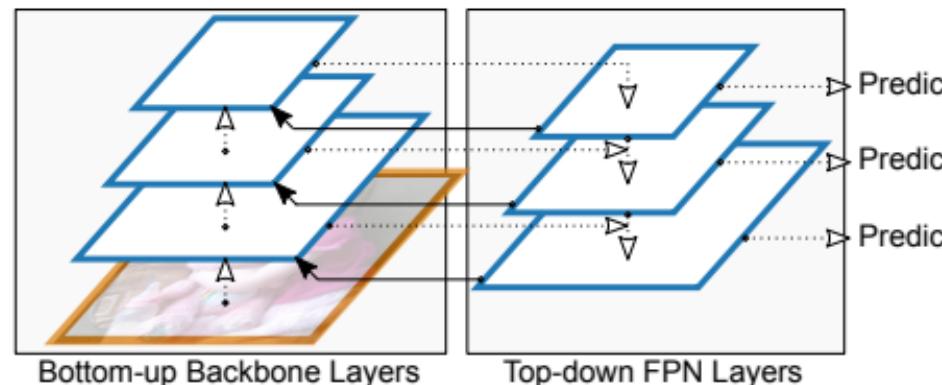
backbone, Neck, DenseHead, ROIExtractor, ROIHead(BBoxHead, MaskHead)



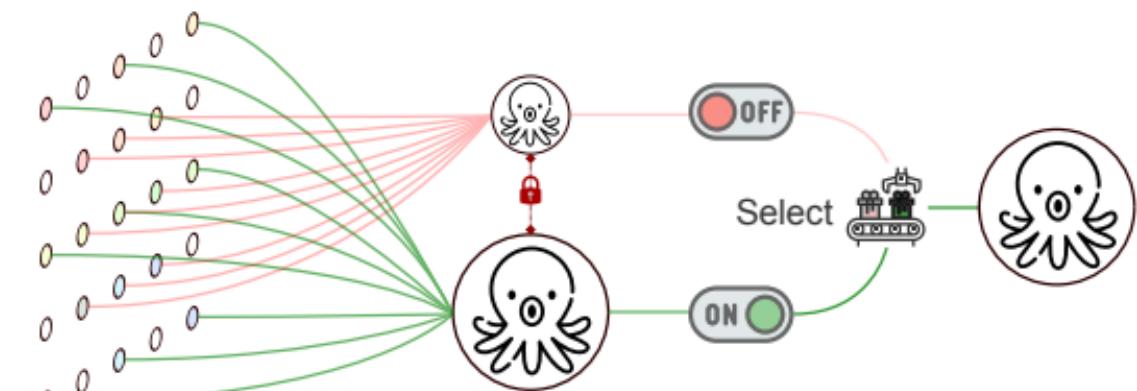
DetectoRS_ResNet

모델 설명

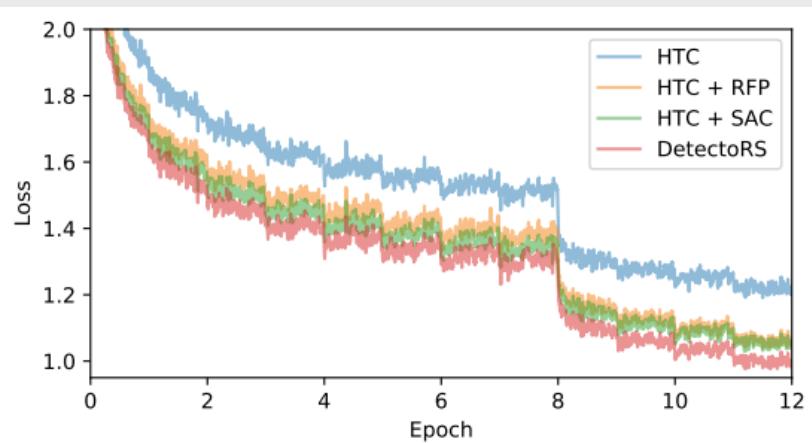
ResNet과 더불어 이미지의 특징 추출에 도움을 주는 Network 모델
피라미드 구조에서의 문제점을 보완하여 큰 이미지의 피쳐값 보장에 유리



(a) Macro Design: Recursive Feature Pyramid.



(b) Micro Design: Switchable Atrous Convolution.



특징

피라미드 & 재귀 형태의 Object Detection 모델

Forward 단계에서,

Switch 역할을 수행하는 함수를 통해 Conv 방식 결정

EfficientNet

모델 소개

Improving Accuracy and Efficiency
정확도와 효율성 향상

through AutoML and Model Scaling
AutoML 과 모델 스케일링 을 통한

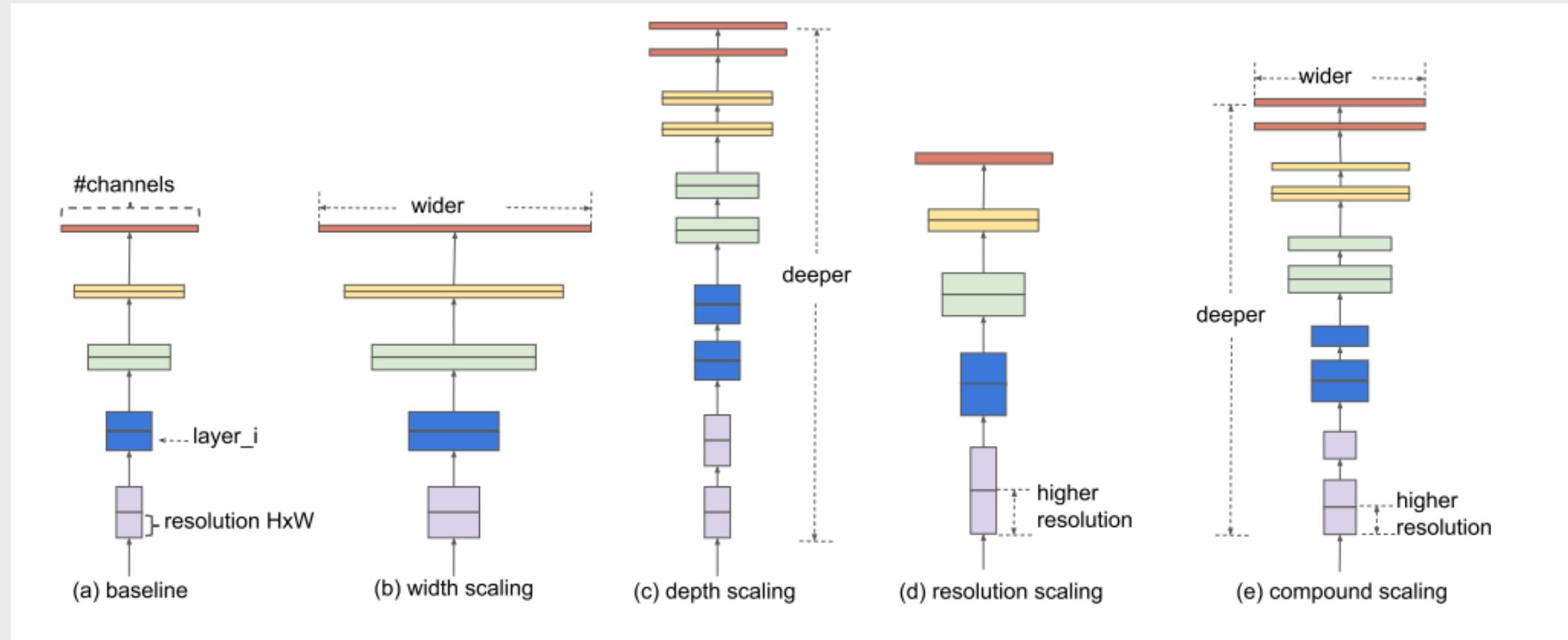
- 특장점 : 이미지 분류 성능이 탁월
- Mingxing Tan, Quoc V.Le

EfficientNet

모델 이론

width, depth resolution scaling 구조

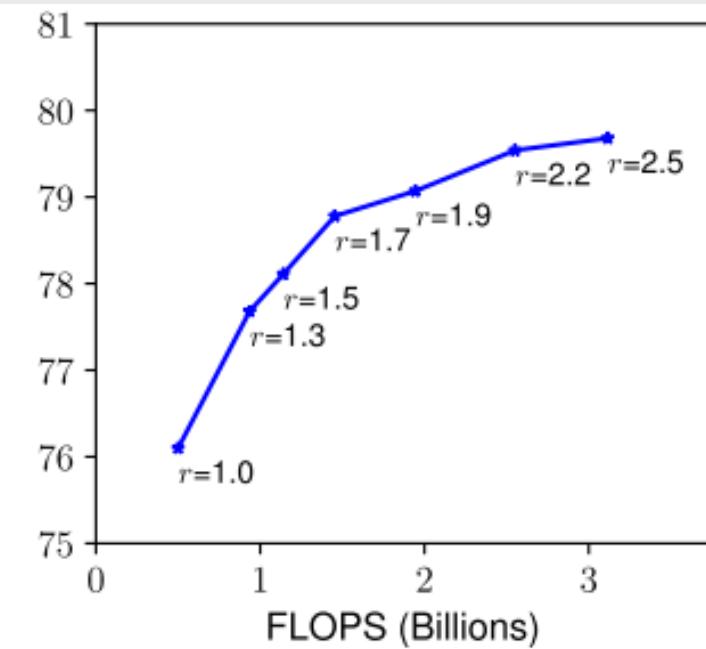
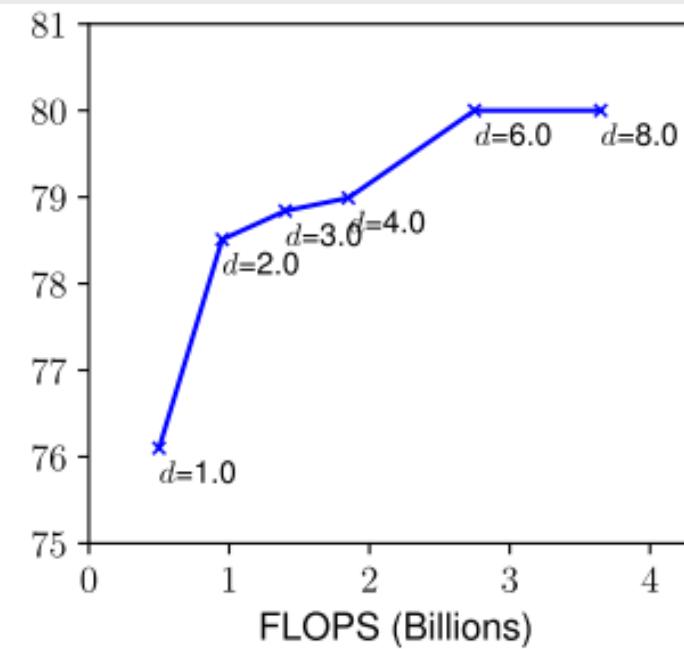
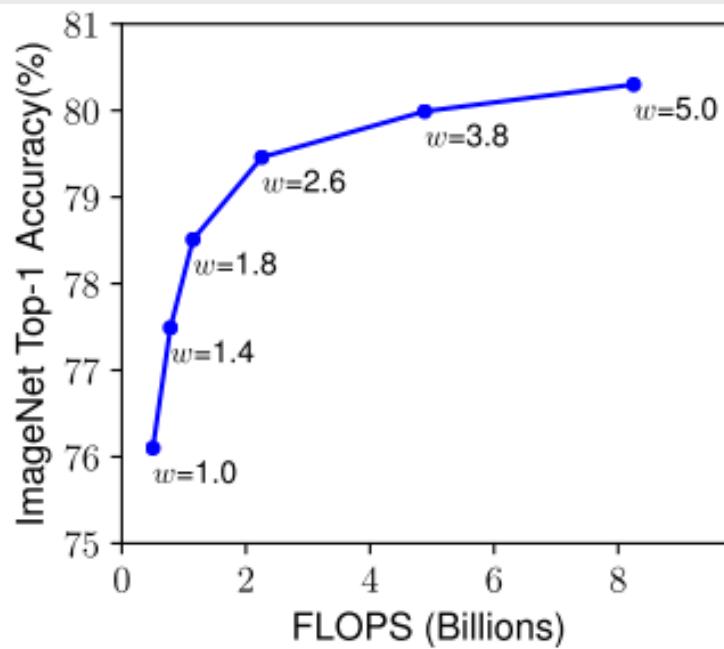
3개를 모두 활용하는 Compound scaling



EfficientNet

기법 별 성능

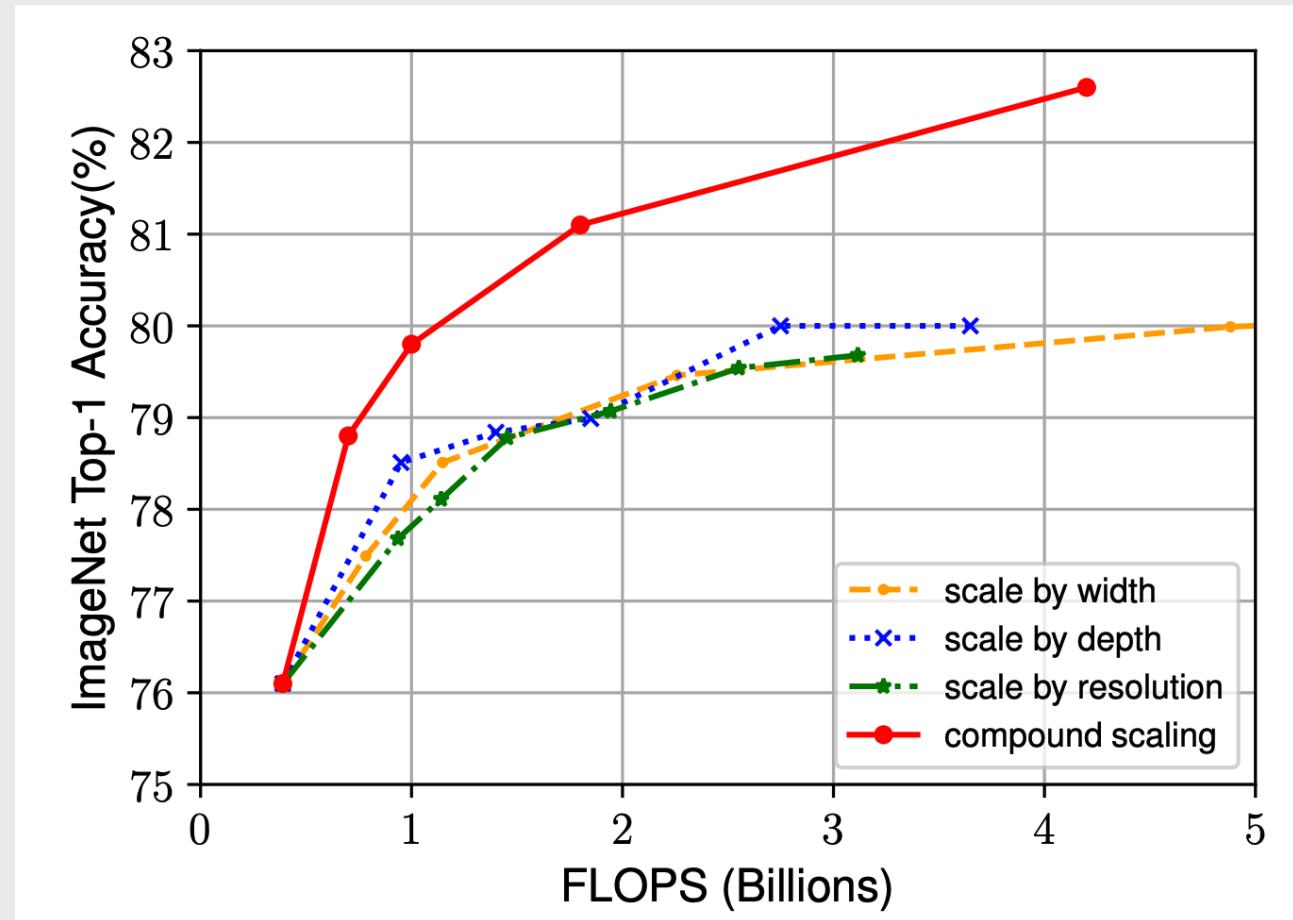
각 기법마다 나머지를 고정해두고 하나의 factor만 키워나가면
성능이 올라가는 것을 알 수 있다.



EfficientNet

| Depth, width, resolution, compound scaling 비교

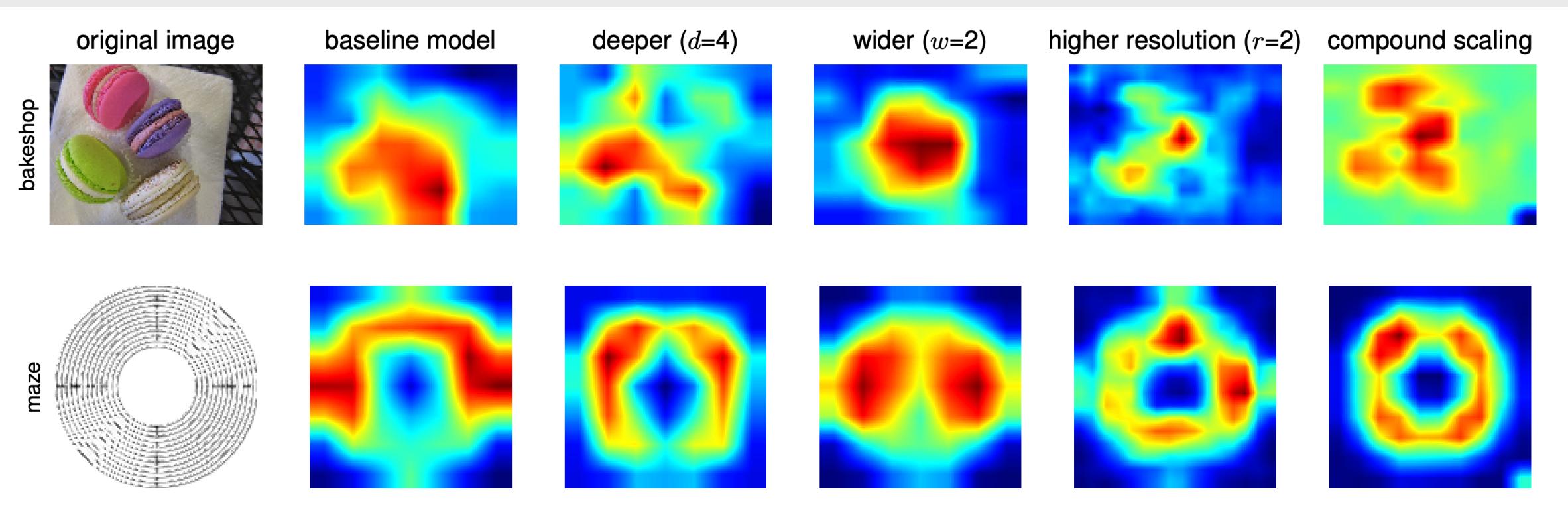
Compound scaling



EfficientNet

CAM

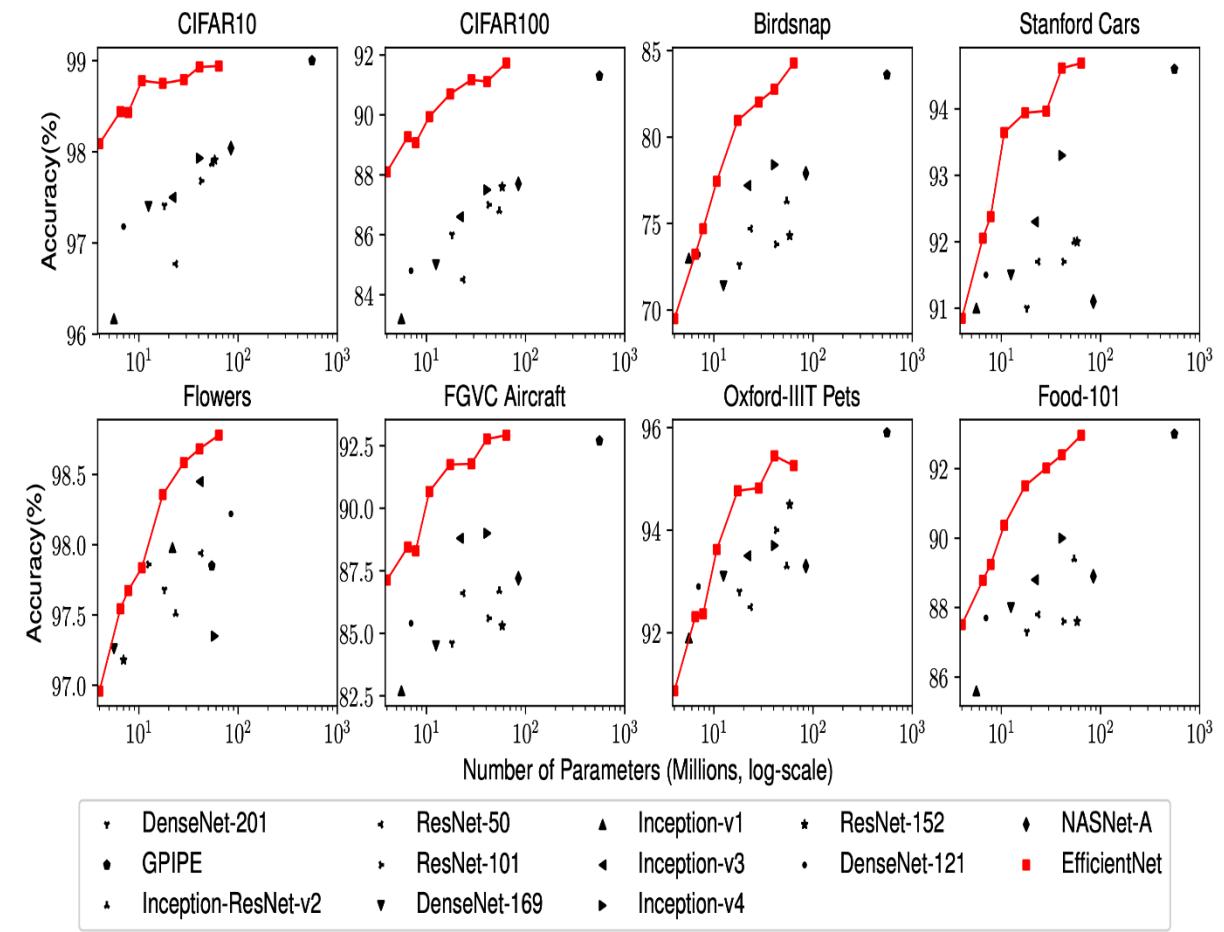
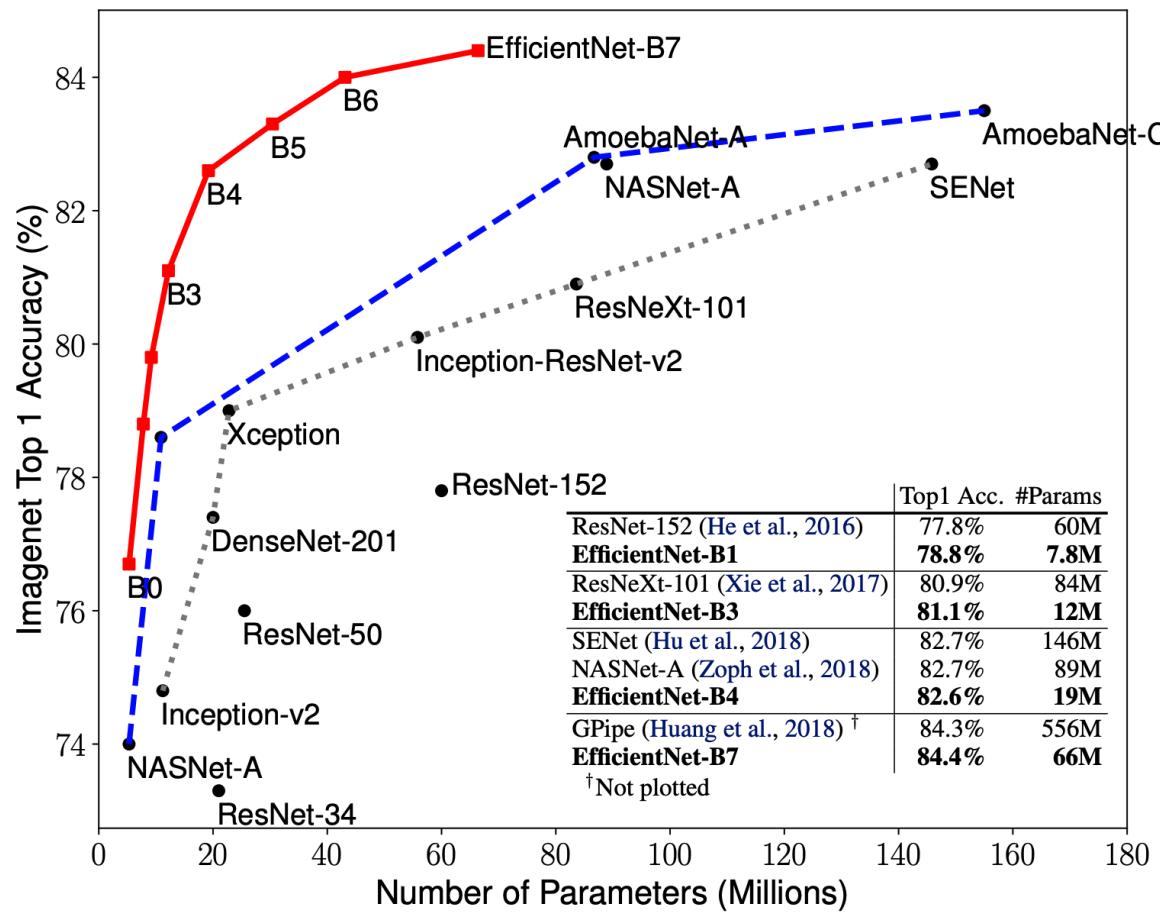
Class activation map을 통한 시각화로 compound scaling이
좀 더 정교하게 Image Detection을 수행한다.



EfficientNet

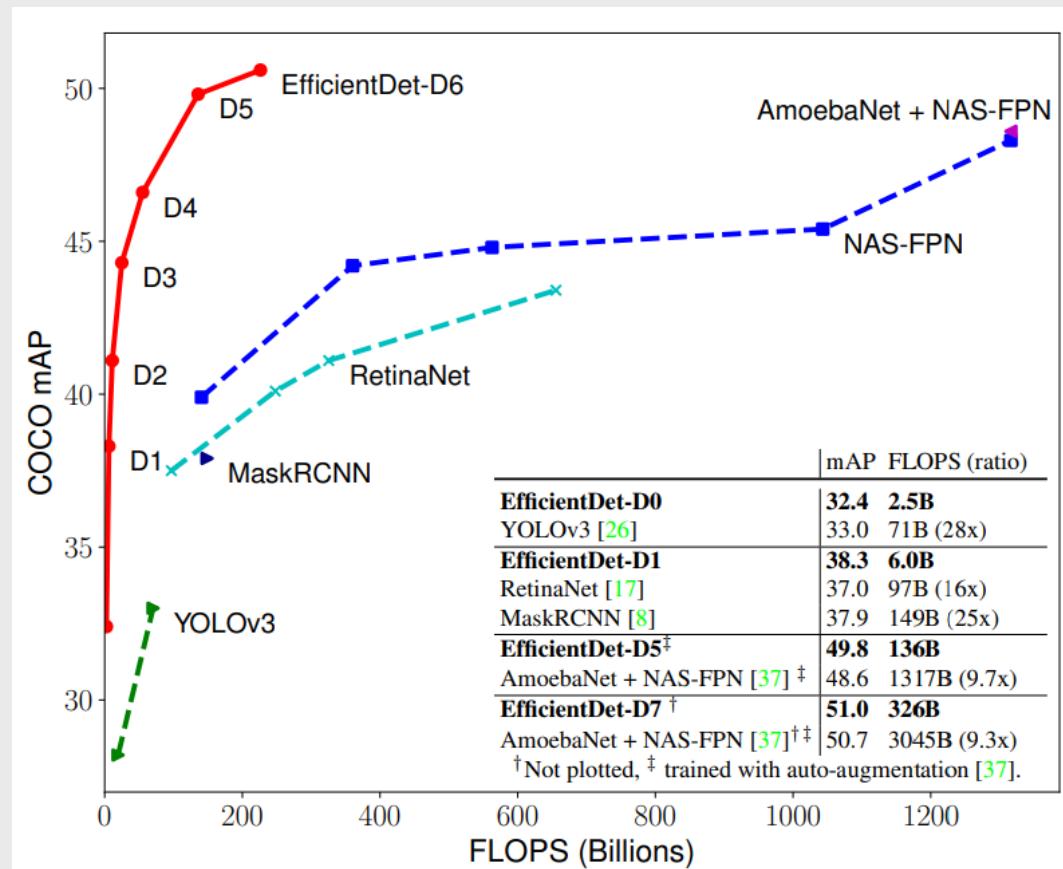
| 타 모델 대비 뛰어난 성능을 가지고 있다.

성능



EfficientDet

| COCO 데이터셋에서 가장 높은 mAP를 달성하여 연산 효율이 좋음 (2019년 11월 기준)
성능

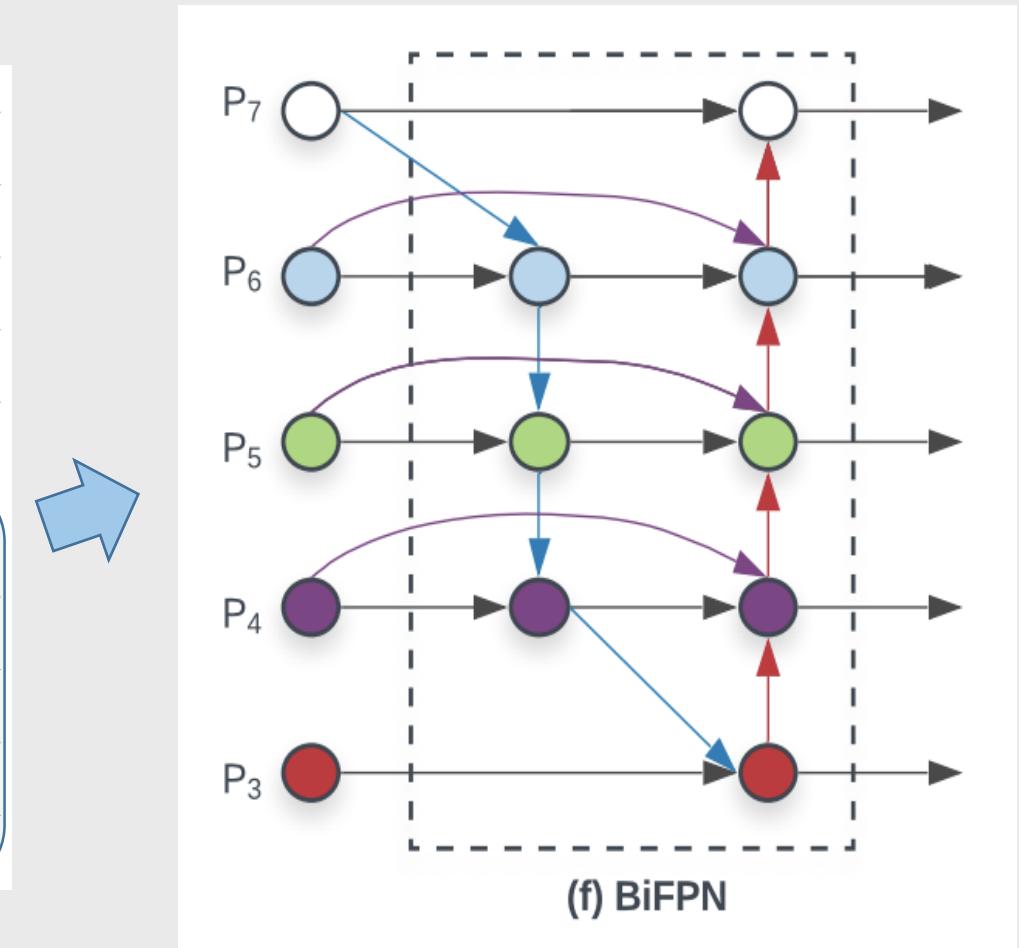
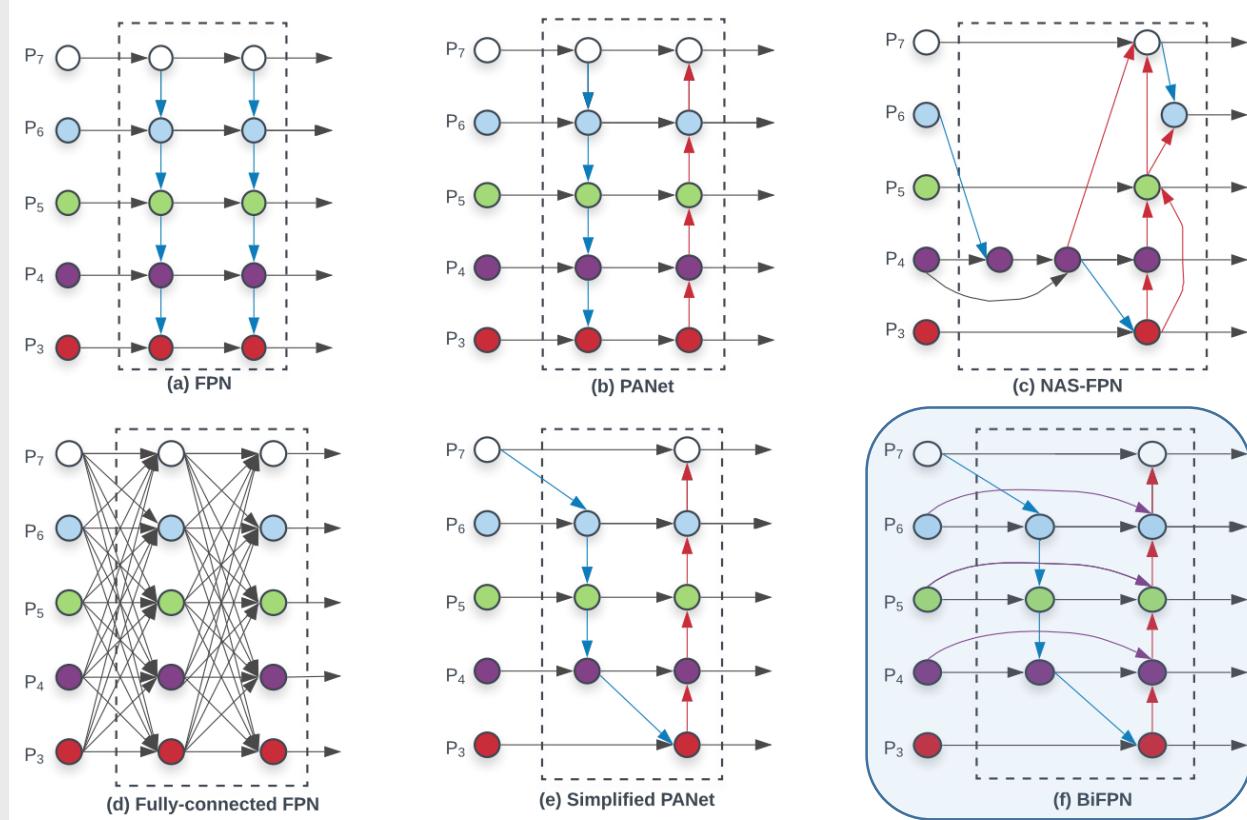


Weighted bidirectional feature network (**BiFPN**) 과 **Compound Scaling**을 접목하여 좋은 성능을 보이는 EfficientDet 구조를 제안하였다.

BiFPN

이론

같은 scale에서 edge를 추가하여 더 많은 feature들이 fusion되도록 구성한 방식이 BiFPN입니다. (보라색 선)



BiFPN

이론

Backbone을 EfficientNet-B3으로 두고 Feature Network를 BiFPN으로 사용할 때 성능이 가장 우수

	mAP	Parameters	FLOPS
ResNet50 + FPN	37.0	34M	97B
EfficientNet-B3 + FPN	40.3	21M	75B
EfficientNet-B3 + BiFPN	44.4	12M	24B

FPN을 BiFPN으로 바꿨을 때의 성능을 측정

	mAP	#Params ratio	#FLOPS ratio
Top-Down FPN [16]	42.29	1.0x	1.0x
Repeated PANet [19]	44.08	1.0x	1.0x
NAS-FPN [5]	43.16	0.71x	0.72x
Fully-Connected FPN	43.06	1.24x	1.21x
BiFPN (w/o weighted)	43.94	0.88x	0.67x
BiFPN (w/ weighted)	44.39	0.88x	0.68x

여러 Feature Network 방식들에 따라 성능을 측정

Weighted Feature Fusion

이론

Input feature에 가중치를 주고, 학습을 통해 가중치를 배울 수 있는 3가지 방식 중 Fast normalized fusion 방식 제안

Unbounded fusion: $O = \sum_i w_i \cdot I_i$

Softmax-based fusion: $O = \sum_i \frac{e^{w_i}}{\sum_j e^{w_j}} \cdot I_i$

Fast normalized fusion: $O = \sum_i \frac{w_i}{\epsilon + \sum_j w_j} \cdot I_i$

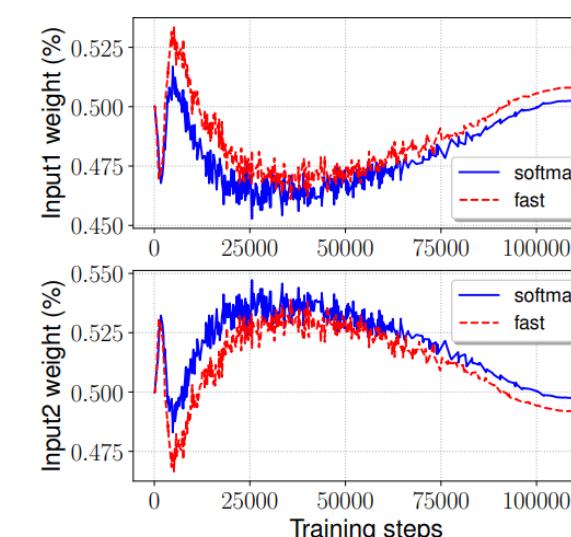
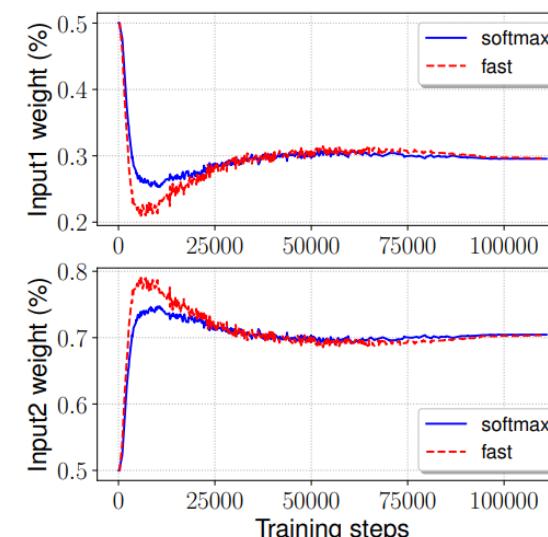
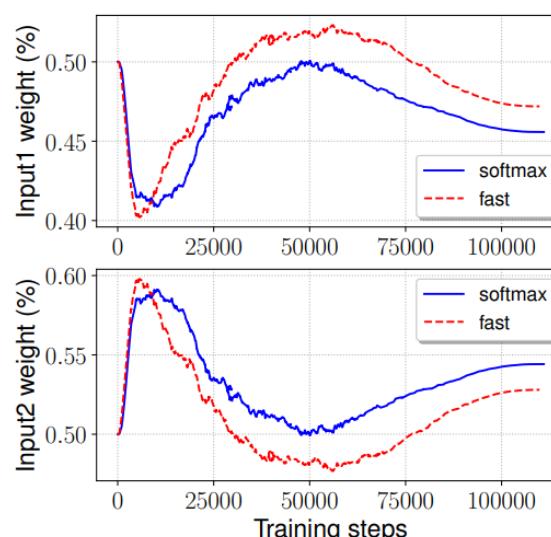
$$P_6^{td} = Conv \left(\frac{w_1 \cdot P_6^{in} + w_2 \cdot Resize(P_7^{in})}{w_1 + w_2 + \epsilon} \right)$$

$$P_6^{out} = Conv \left(\frac{w'_1 \cdot P_6^{in} + w'_2 \cdot P_6^{td} + w'_3 \cdot Resize(P_5^{out})}{w'_1 + w'_2 + w'_3 + \epsilon} \right)$$

– condition : $\epsilon \approx 0.0001$

Weighted Feature Fusion

성능 비교



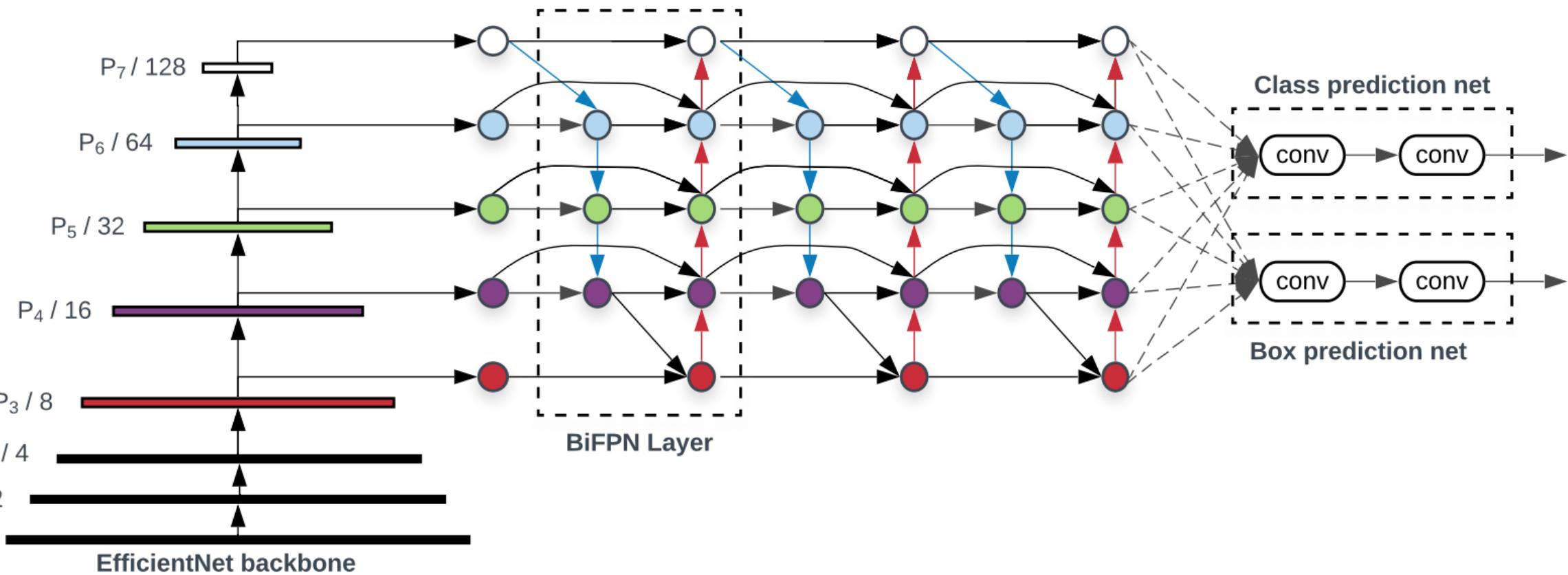
Fast Fusion을 사용하면 약간의 mAP 하락은 있지만
약 30%의 속도 향상을 달성할 수 있습니다

Model	Softmax Fusion mAP	Fast Fusion mAP (delta)	Speedup
Model1	33.96	33.85 (-0.11)	1.28x
Model2	43.78	43.77 (-0.01)	1.26x
Model3	48.79	48.74 (-0.05)	1.31x

EfficientDet Architecture

모델 구조

Backbone으로 EfficientNet 사용, BiFPN을 Feature Net으로 사용하여 top-down & bottom-up 을 반복적으로 수행

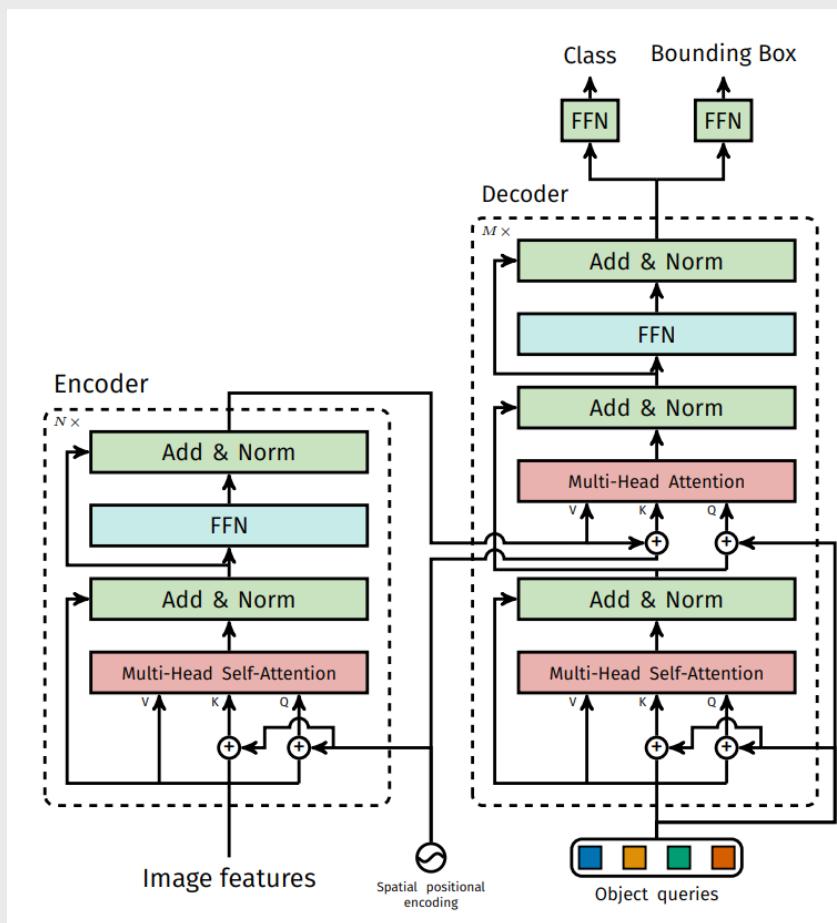


DETR

모델 소개

ECCV 2020, End-to-End Object Detection with Transformers

Anchor나 NMS 단계를 생략한 간단한 모델



Transformer

- Attention을 통해 전체 이미지의 문맥 정보를 이해
- 이미지 내 각 인스턴스의 **상호작용**(interaction) 파악 용이
- 큰 bounding box에서의 거리가 **먼 픽셀 간의 연관성** 파악 용이

Encoder

- 이미지의 특징(Feature) 정보를 포함하고 있는 **각 픽셀 위치 데이터**를 입력받아 인코딩 수행

Decoder

- N개의 **object query**를 초기 입력으로 받아 인코딩된 정보에 활용
- 각 query는 이미지 내의 **서로 다른 고유한 인스턴스**를 구별

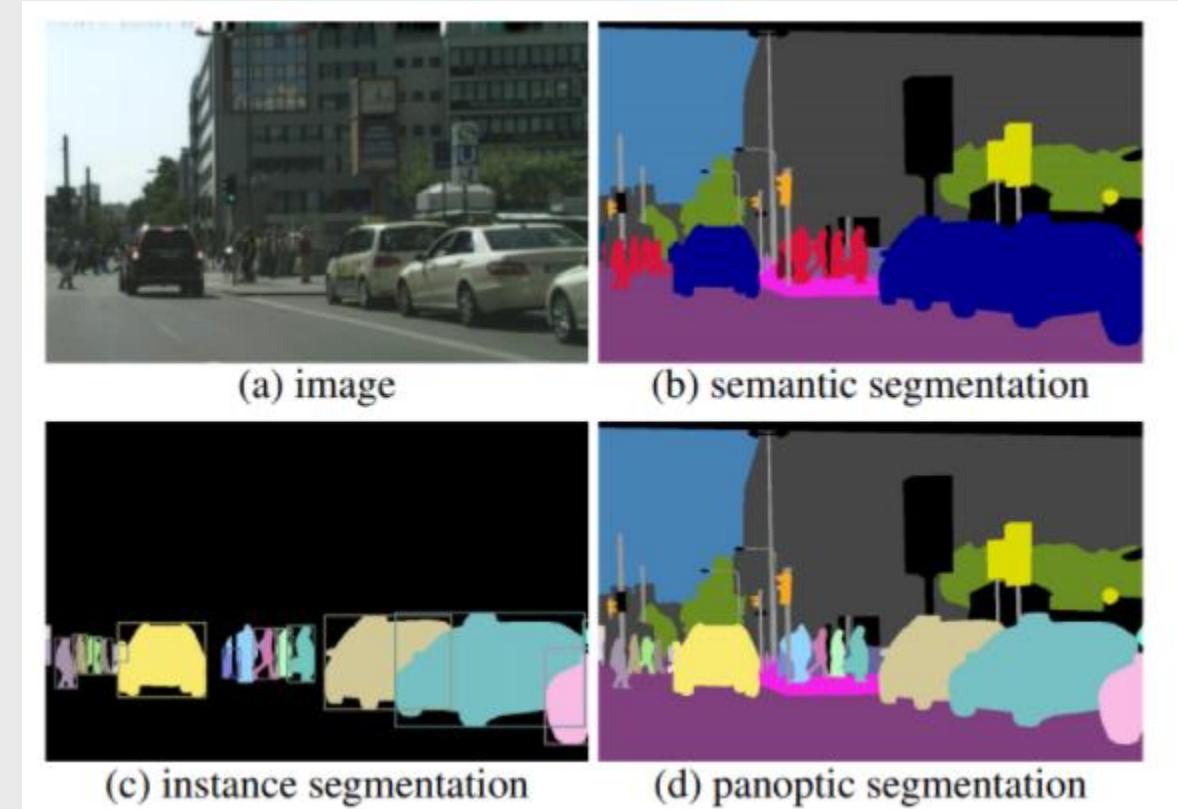
DETR

모델 이론

Transformer

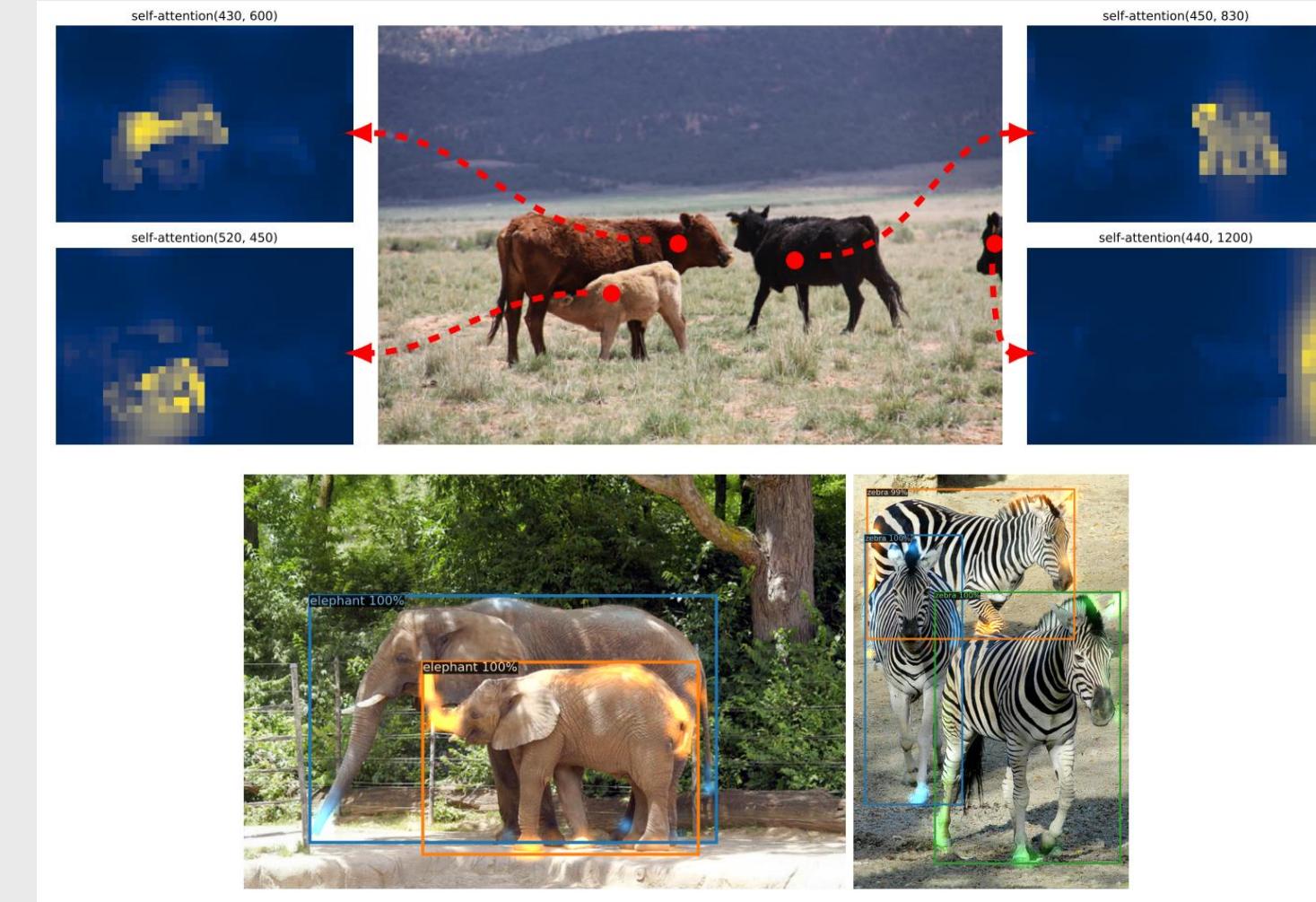
객체를 종류별로 구분하는 것을 넘어서서
하나의 물체를 **고유한 객체**로 인정

| 새로운 Transformer를 사용하는 Detection Framework



DETR

모델 이론
System



DETR

모델 이론

Loss func

이분 매칭 손실 함수(Hungarian loss)

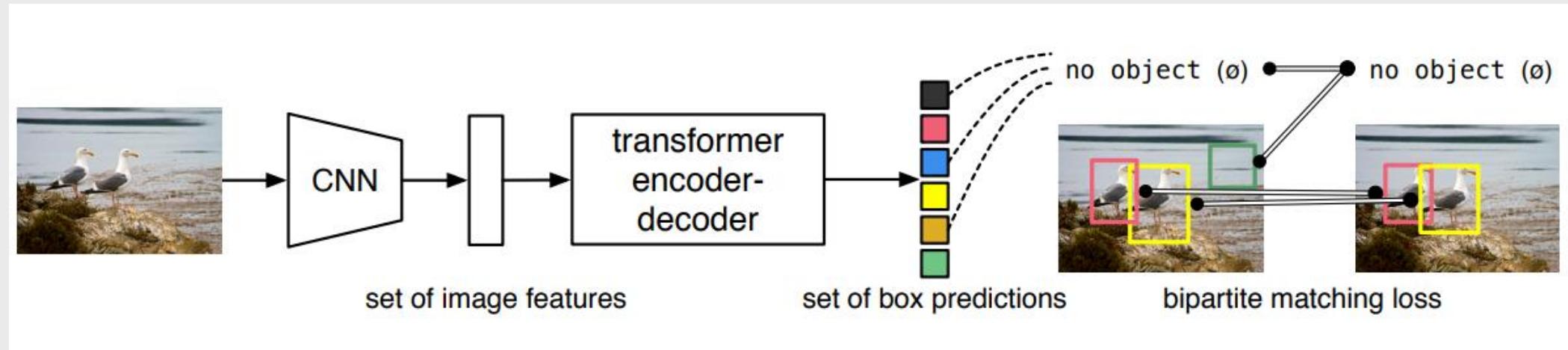
N = 6 일 경우 예측 결과

 $(c_0 = \emptyset, b_0)$ $(c_1 = \text{bird}, b_1 = (180, 180, 150, 240))$ $(c_2 = \emptyset, b_2)$ $(c_3 = \text{bird}, b_3 = (120, 150, 100, 150))$ $(c_4 = \emptyset, b_4)$ $(c_5 = \text{dog}, b_5)$  $(c_0 = \text{bird}, b_0 = (122, 151, 100, 150))$ $(c_1 = \text{bird}, b_1 = (182, 180, 148, 238))$ $(c_2 = \emptyset, b_2)$ $(c_3 = \emptyset, b_3)$ $(c_4 = \emptyset, b_4)$ $(c_5 = \emptyset, b_5)$

DETR

| DETR = Loss func + Transformer

모델 요약



Query의 개수를 정해 출력 개수를 고정

이분 매칭을 통해 중복되지 않는 객체를 탐지하되

Anchor와 NMS 사용 없이 간단하게 해결

대표 모형 | Yolo를 대표 모형으로 선정한 이유

1. 빠른 시간 내 훈련 및 제출 가능
2. 간단하게 parameter 변경 가능
3. 이전 프로젝트 Yolov3를 사용한 경험

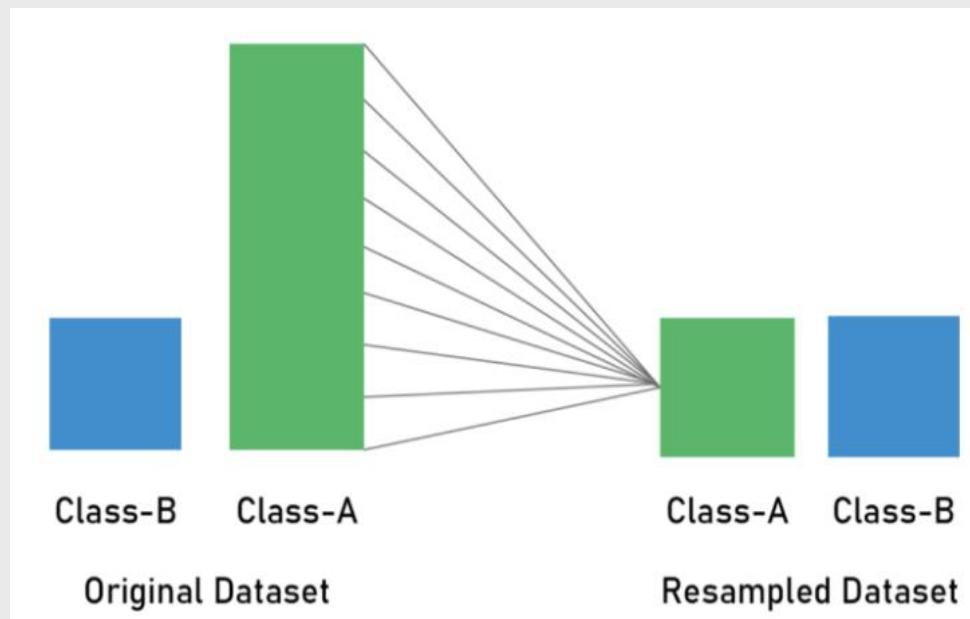
모형 평가

Yolo

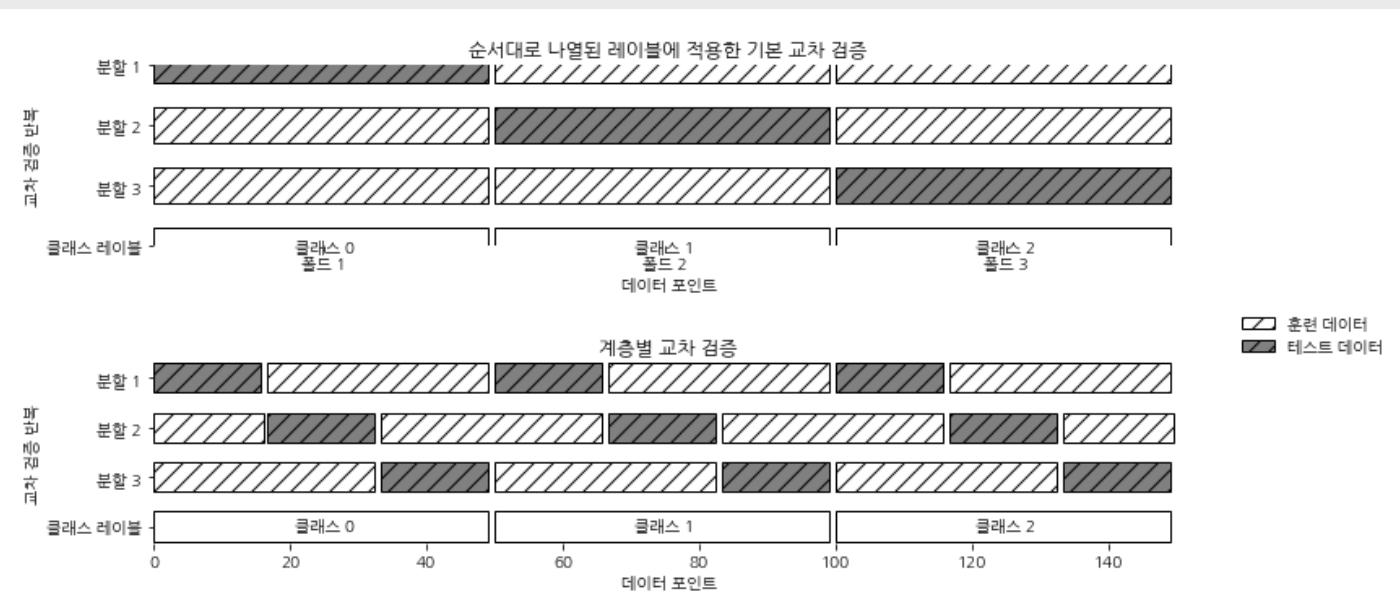
불균형

불균형 데이터이기 때문에 그대로 훈련하기엔 제한이 있어 두 가지 방법을 선택
가장 적은 수로 샘플링, 같은 비율로 Fold 생성

Undersampling



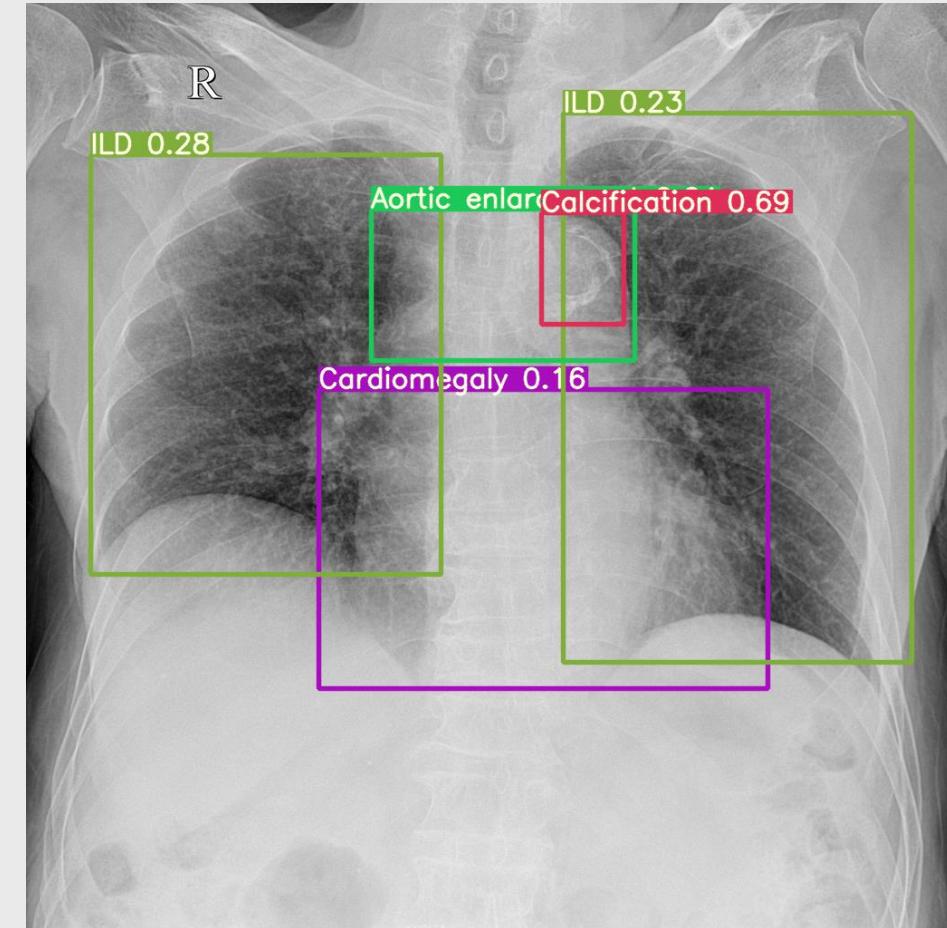
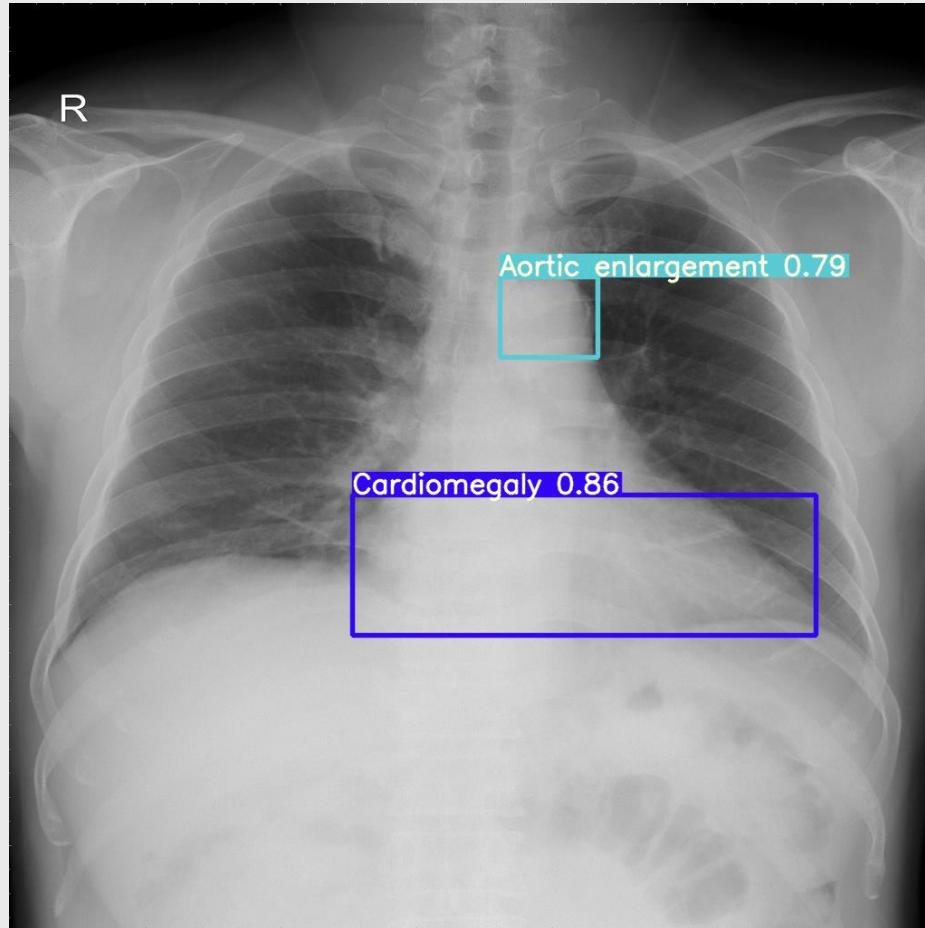
계층적 K-Fold



Yolo

| 실제 Test한 이미지들

모형 평가



Yolo

모형 평가

Yolo기반으로 훈련한 모델들과 모델 별 대표적인 parameter 변경점들이다.

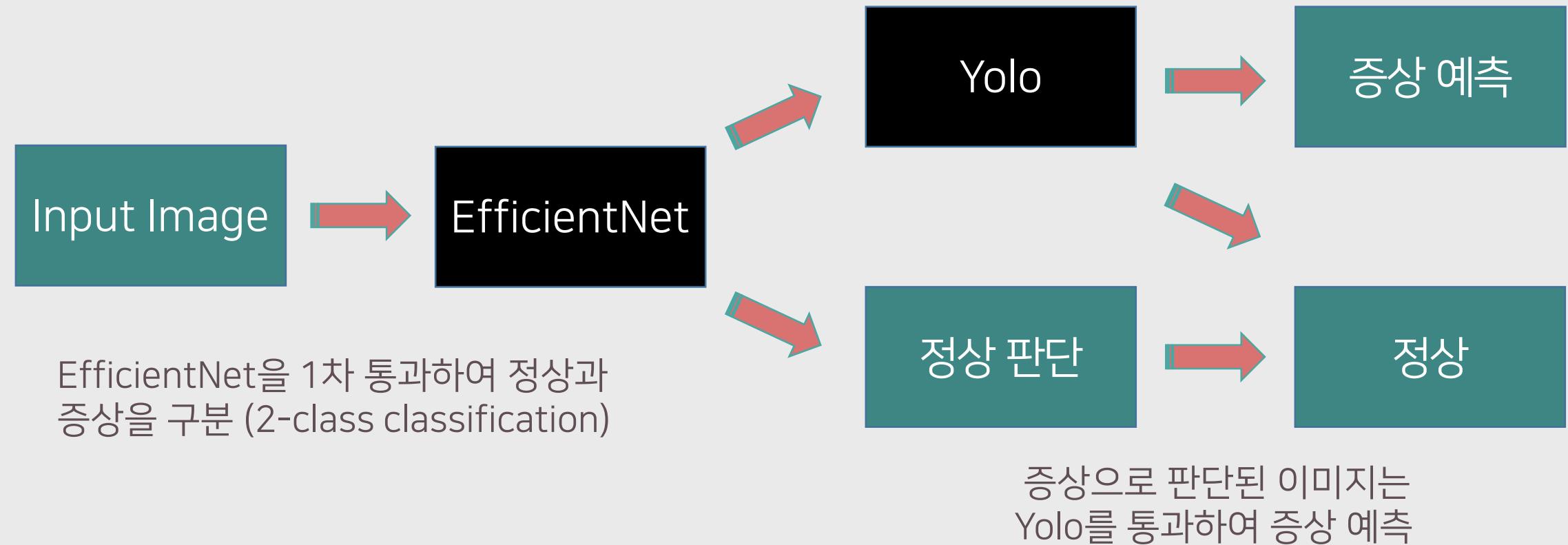
각 모델의 훈련 시간은 평균 59분 30초, Submission detection 시간은 4분 30초였다.

모델명	Submission	Img	Epochs	Confidence	Score
Yolov5s	2021.02.15	832x832	30	0.15	0.024
Yolov5l	2021.02.15	640x640	200	0.15	0.048
Yolov5s	2021.02.16	640x640	40	0.15	0.098
Yolov5s	2021.02.16	832x832	40	0.2	0.126
Yolov5s	2021.02.17	832x832	40	0.4	0.113
Yolov5s	2021.02.17	832x832	40	0.15	0.13
Yolov5m	2021.02.17	1024x1024	50	0.15	0.135
Yolov5m	2021.02.18	1024x1024	50	0.2	0.133
Yolov5m	2021.02.18	1024x1024	50	0.15	0.133
Yolov5m	2021.02.18	1024x1024	50	0.15	0.137

Yolo

모형 평가

Top Score 가 0.137로 모델에 큰 변화를 주는 것이 올바르다고 판단하여 EfficientNet을 활용하는 방법을 채택



Yolo

모형 평가

| 현재 743개의 팀 중 434등이다. 약 1달 간의 시간 안에 Top Score를 넘어 서는 것이 목표이다.

현재

mAP
0.187
434/743



Top

mAP
0.318
1/743

제출 기한 : 03.30

한계 및 시사점

한계

개발환경

Colab, kaggle notebook and GPU issue

환경	특징
Kaggle Notebook	대회 데이터를 바로 이용할 수 있지만, 모델 훈련할 때 GPU 사용이 제한됨.
Google Colab Pro	사용하면 많은 시간 GPU 사용이 가능하지만, 많은 데이터들을 매번 업로드 해야하는 어려움

두 환경에 대해 코딩 변경을 하는 데 시간이 소모

+

공통	모델의 특징과 관련해 의존성이 높음 무거운 모델의 경우 환경과 상관없이 메모리 이슈가 발생
----	----------------------------------------------------------

한계

DICOM

“메타 데이터” - 개인정보 보안 이슈와 관련해 환자 정보를 담고 있는 DICOM 파일이
실질적으로 환자의 성별, 나이를 제외한 정보는 기록되어 있지 않음 (대다수가 이미지 정보)

```
Dataset.file_meta -----
(0002, 0000) File Meta Information Group Length UL: 154
(0002, 0001) File Meta Information Version OB: b'\x00\x01'
(0002, 0002) Media Storage SOP Class UID UI: Computed Radiography Image Storage
(0002, 0003) Media Storage SOP Instance UID UI: 0005e8e3701dfb1dd93d53e2ff537b6e
(0002, 0010) Transfer Syntax UID UI: Implicit VR LittleEndian
(0002, 0012) Implementation Class UID UI: 1.2.3.4
(0002, 0013) Implementation Version Name SH: 'OFFIS_DCMTK_360'

-----
(0010, 0040) Patient's Sex CS: 'F'
(0010, 1010) Patient's Age AS: '028Y'
(0028, 0002) Samples per Pixel US: 1
(0028, 0004) Photometric Interpretation CS: 'MONOCHROME2'
(0028, 0008) Number of Frames IS: "1"
(0028, 0010) Rows US: 3072
(0028, 0011) Columns US: 3072
(0028, 0030) Pixel Spacing DS: [0.140, 0.140]
(0028, 0034) Pixel Aspect Ratio IS: None
(0028, 0100) Bits Allocated US: 16
(0028, 0101) Bits Stored US: 14
(0028, 0102) High Bit US: 13
(0028, 0103) Pixel Representation US: 0
(0028, 1050) Window Center DS: "8483.0"
(0028, 1051) Window Width DS: "11160.0"
(0028, 2110) Lossy Image Compression CS: ''
(0028, 2112) Lossy Image Compression Ratio DS: None
(0028, 2114) Lossy Image Compression Method CS: ''
(7fe0, 0010) Pixel Data OW: Array of 18874368 elements
time : 3.132694959640503
```

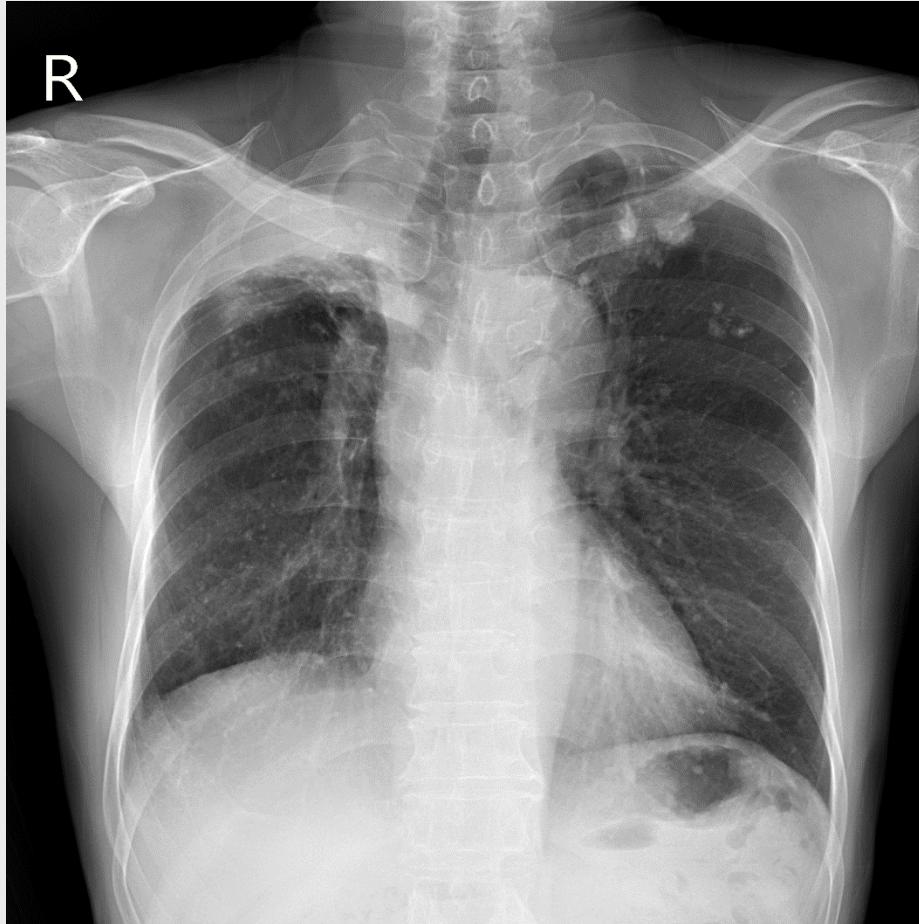
미디어 저장소와 관련된 정보
환자의 성별, 나이
프레임 번호, 이미지 크기
비트 정보 등...

환자 정보를 이미지 파일 특성과 결합시켜
관계성을 이끌어내길 기대했으나 불가

한계

Image

고해상도 이미지 업로드



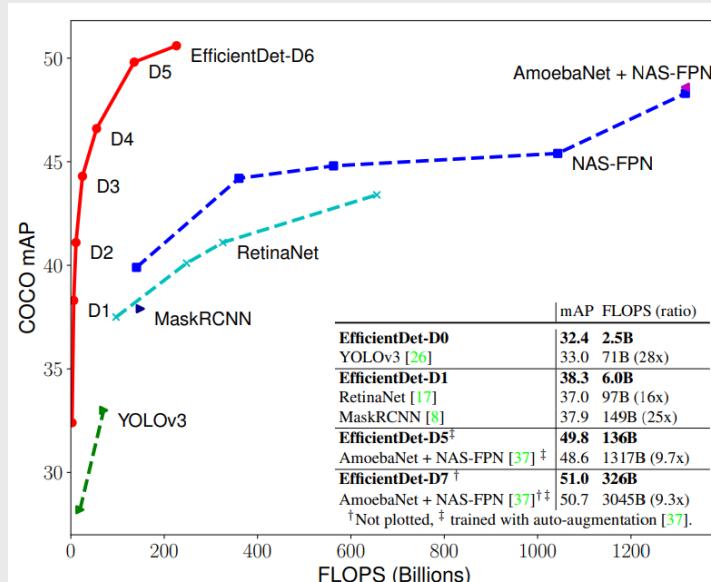
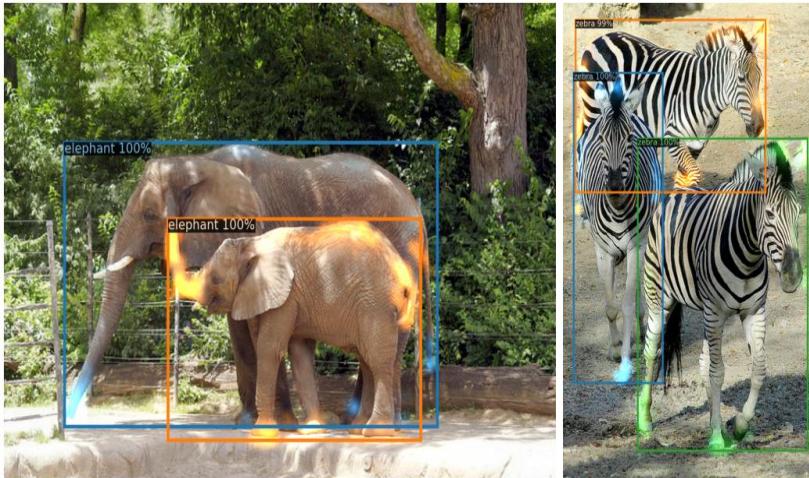
고해상도 이미지 파일(jpg) 생성 성공
총 용량 19.4 GB
이미지 업로드에 제한
비교적 작은 이미지를 활용

연구방향

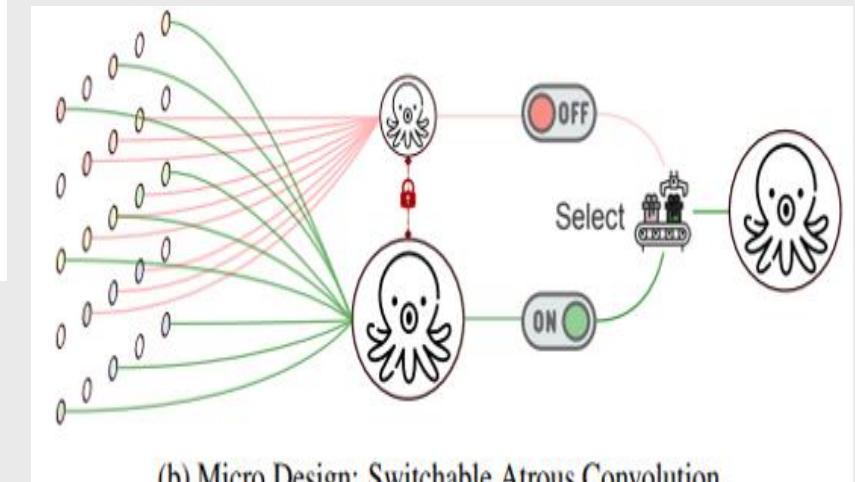
| 연구 중인 모델들

다른 모델

DETR



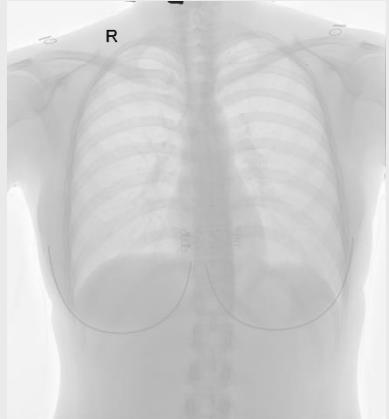
DetectoRS_ResNet



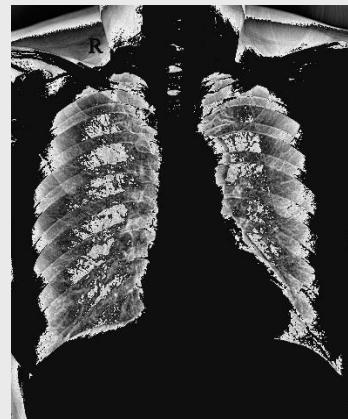
연구방향

전처리

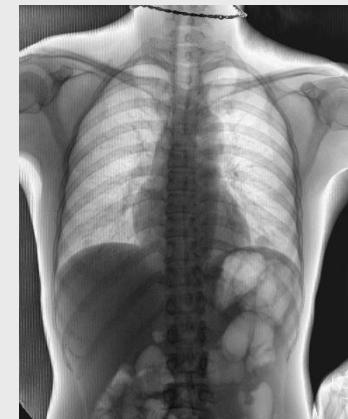
이미지 변환을 통해 여러가지 형태의 이미지 획득 후
높은 정확성을 가진 이미지 변환 방식 사용 필요



1



2



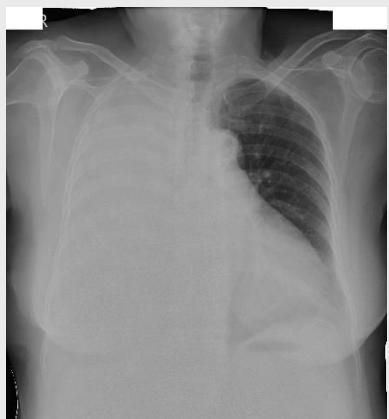
3



4



5



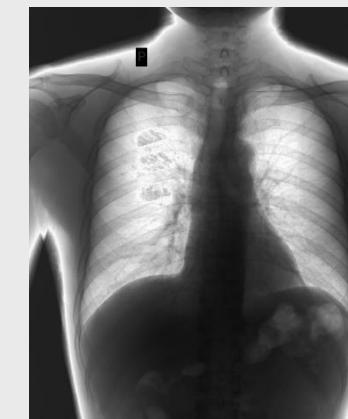
6



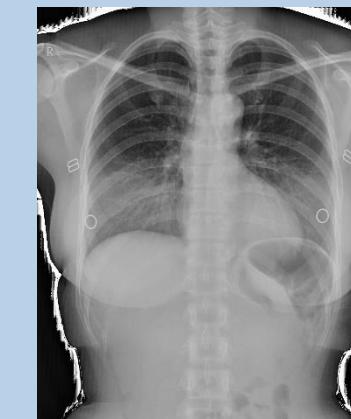
7



8



9



10

참고 문헌

참고문헌

논문

- 김지율, 고성진. (2017). DICOM 영상과 다양한 형식의 영상 비교.
신호처리 시스템 학회 논문지, 18(2), 76-83.
- Redmon, J., Divvala, S., Girshick, R.B., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection.
2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 779-788.
- J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. ArXiv, abs/1804.02767.
- Bochkovskiy, A., Wang, C., & Liao, H. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. ArXiv, abs/2004.10934.
- Tan, M., & Le, Q.V. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. ArXiv, abs/1905.11946.
- Tan, M., Pang, R., & Le, Q.V. (2020). EfficientDet: Scalable and Efficient Object Detection.
2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 10778-10787.
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020).
End-to-End Object Detection with Transformers. ArXiv, abs/2005.12872.
- Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., & Yoo, Y. (2019). CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 6022-6031.
- Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., & Lin, D. (2019). MMDetection: Open MMLab Detection Toolbox and Benchmark.
ArXiv, abs/1906.07155.
- Qiao, S., Chen, L., & Yuille, A. (2020). DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution.
ArXiv, abs/2006.02334.

개발환경



프로젝트 관리



Thank You

