

[프로젝트 #2]

YOLO를 이용한 안전모 검출

2022. 04. 20

김 현 용

충북대학교 산업인공지능학과

조편성 결과 (4/20 현재)

20 학번 조편성

조	성명	발표여부
20-1	전일우, 이효중, 박성범	○
20-2	임동민, 신정환, 안건호	
20-3	최원희, 손의걸	○
20-4	윤재웅, 김성웅	○
20-5	강윤구, 김병근	○
20-6	박민우	○
20-7	원형일, 장민우	
20-8	고정재, 유용주	○

총 8조 / 17명

21 학번 조편성

조	성명	발표여부
21-1	이용규, 유대건	
21-2	김대훈	○
21-3	최준혁, 이지연	
21-4	김상순, 정수현	
21-5	방창현, 정원용	
21-6	우상진, 김준태	
21-7	봉은정, 김원우	
21-8	이충현, 이지호	
21-9	정준영, 윤범희	

총 9조 / 17명

조별 발표 주제 (4/20 현재)

주차	날짜	발표 주제	발표조
1	3/2	오리엔테이션 / 조 편성	도규원
2	3/16	Project #1 : CNN을 이용한 불량 검출	김현용
3	3/23	OpenCV 활용 - 주요 명령어 사용법, 영상파일 입출력 및 변환 등	20-3, 20-4, 20-8
4	3/30	CNN- 컨벌루션 레이어 개념 및 레이어 코딩 등	20-5, 20-6
5	4/6	데이터 증량 - 영상 회전, 스케일링, 대칭 등	20-1, 21-2
6	4/13	Project #1 발표평가	모든 조
7	4/20	Project #2 : YOLO를 이용한 안전모 검출	김현용, 도규원
8	4/27	검출과 분할 - Object Detection vs. Segmentation, Mask-RCNN	21-3, 21-4
9	5/4	AI특강1(19:00~20:30)	초청강사
10	5/11	RCNN계열-R-CNN, fast R-CNN, faster R-CNN의 비교(구조, 성능) 1-Stage detector - YOLO 외 SSD, RetinaNet 등	21-6, 21-9 20-7, 21-1
11	5/18	AI특강2(19:00~20:00)	초청강사
12	5/25	YOLOv5 사용법 - 각종 파라미터 설정, 학습/검증/추론 방법 등 주석 - 레이블링 파일형식(xml, json, yolov5) 변환 코드 평가지표- mAP(예제, 코드), 프로젝트 중간점검 : 진행상황 발표	20-2, 21-7 21-5 21-8
13	6/1	[휴무 대보강] 동영상 강의 시청	윤성철, 이광연
14	6/8	AI특강3(19:00~20:00) , 프로젝트 최종점검 : 테스트 데이터 공개 → 검출결과 제출	초청강사
15	6/15	프로젝트 #2 발표평가	모든 조

대면수업
전환

목 차

1. 객체검출의 개요
2. R-CNN 계열
3. YOLO 계열
4. 데이터셋과 주석
5. 평가지표 mAP
6. 프로젝트 #2 설명

객체검출이란?

- Classification(분류) : 객체의 종류(class, label)를 구분하는 것
- Localization(위치찾기) : 하나의 객체의 위치를 찾아 Bounding box로 표시
- Object Detection(객체검출) : 여러 개의 객체에 대한 Localization + Classification
- Instance Segmentation(객체분할) : 픽셀 단위로 분류하는 작업 → polygon으로 표시

Classification



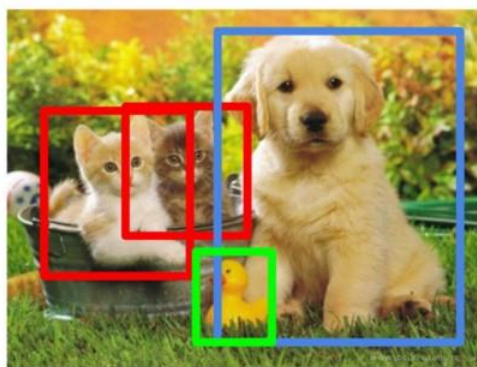
CAT

**Classification
+ Localization**



CAT

Object Detection



CAT, DOG, DUCK

**Instance
Segmentation**



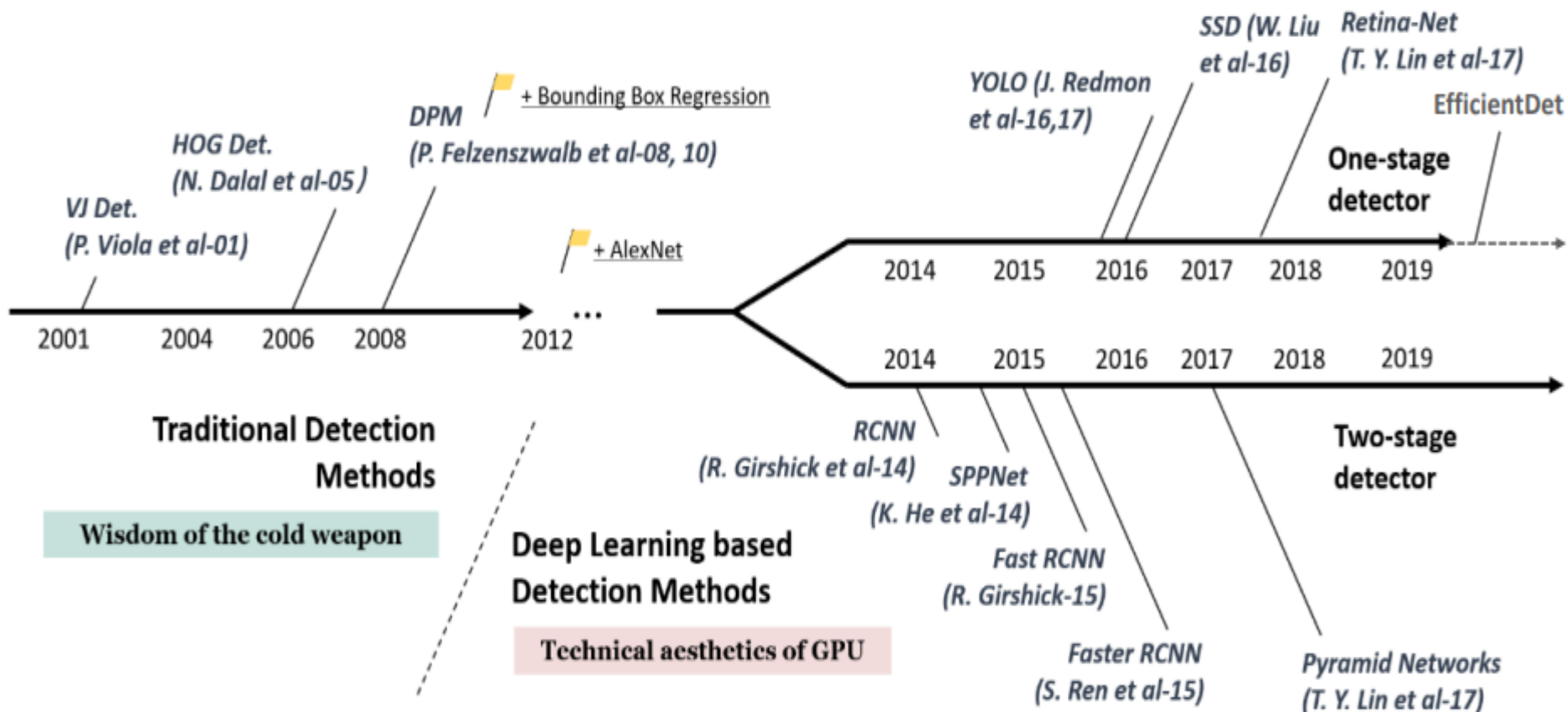
CAT, DOG, DUCK

Single object

Multiple objects

객체검출의 역사


- 2014년부터 딥러닝을 이용한 객체검출 기술이 발표됨
- **Object Detection = Localization + Classification**의 조합
 - 2-stage Detector는 순차적으로 수행 → 정확도는 높지만 속도가 느림
 - 1-stage Detector는 이 두 가지를 동시에 수행
 - 실시간 검출이 가능할 정도로 빠르지만 정확도가 낮음

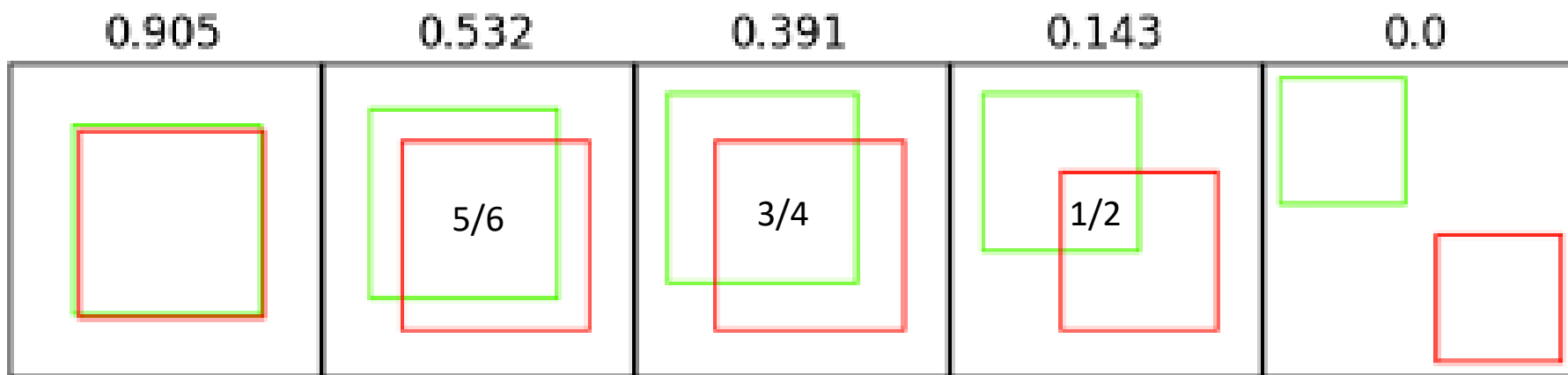
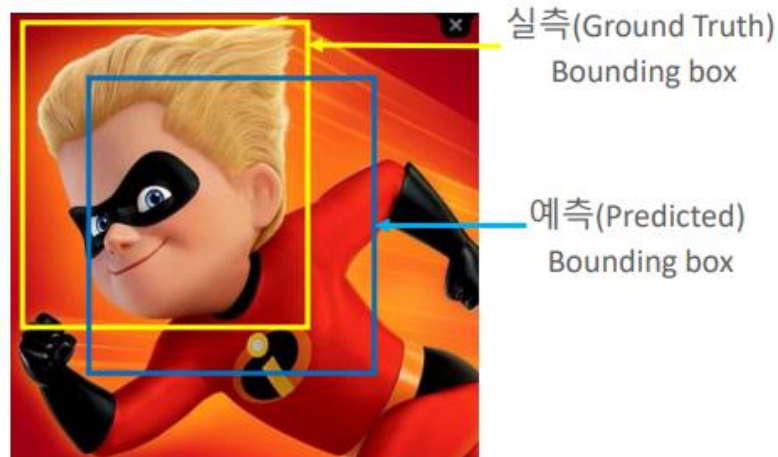


객체검출의 주요 용어

- **IoU (Intersection over Union)**

- 교집합/합집합
- Bounding box의 실측값(Ground Truth)과 예측값이 **얼마나 겹치는가의 지표**
- 일반적으로 $\text{IoU} > 0.5$ 이면 제대로 검출한 것으로 인정

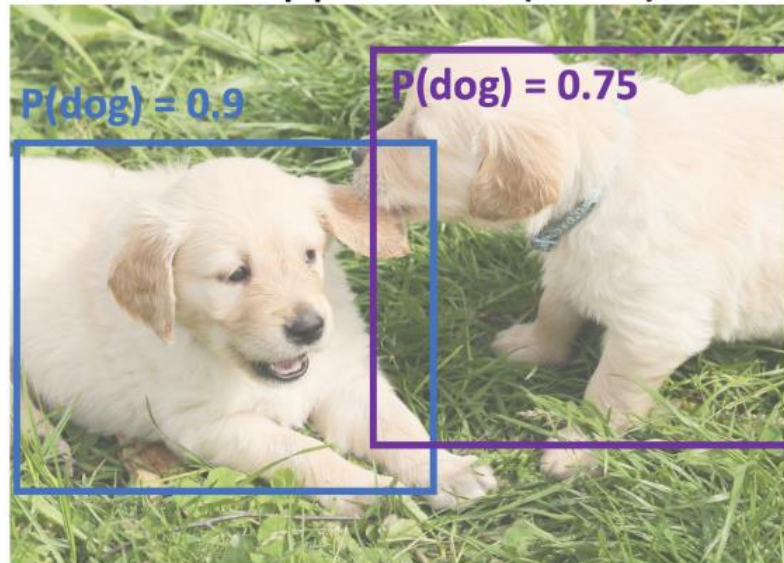
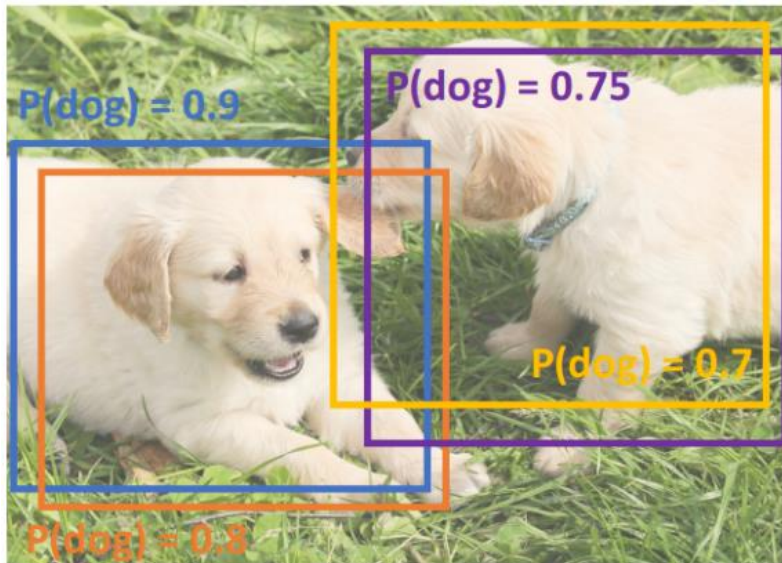
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$




객체검출의 주요 용어

- 비최대 억제(Non-Max Suppression, NMS)

- Object Detection 알고리즘은 객체가 있을 만한 위치에 수많은 Bbox를 출력 (845, 2K개)
- 클래스별로 검출된 객체의 Bounding box 중에 confidence가 가장 높은 box만 남기고, '비슷한 위치'에 있는 box는 중복검출로 간주하여 제거하는 기법
- 중복여부는 box간 IoU를 계산하여 ' $\text{IoU} > \text{threshold}(0.5)$ '이면 중복으로 판단
- 객체가 밀집되어 서로 겹쳐있는 경우에는 'good' 검출을 제거하는 오류가 발생

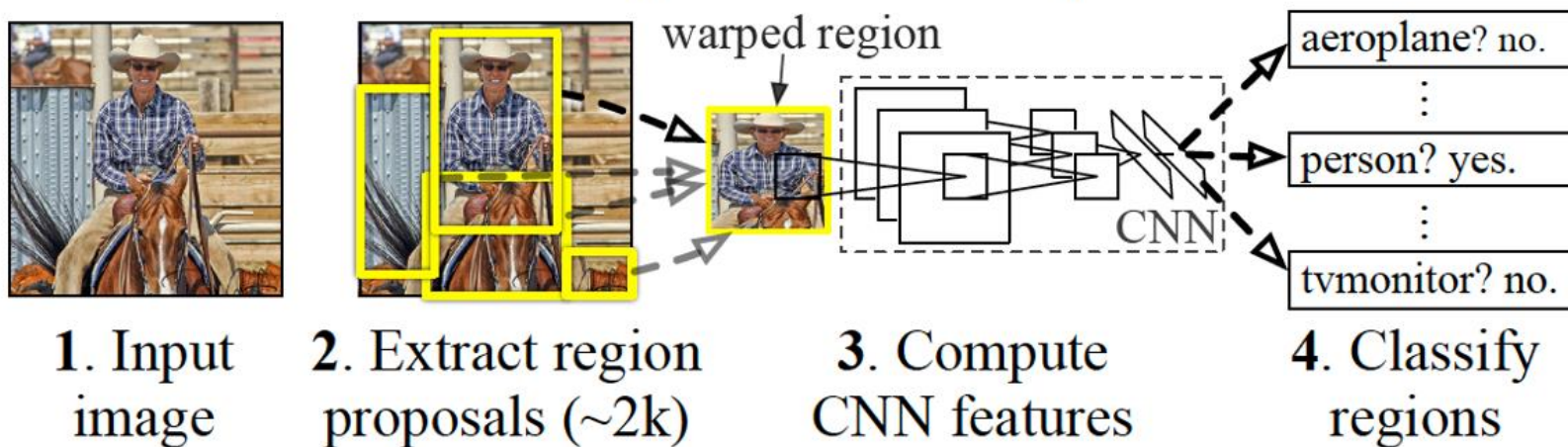


R-CNN (Region-based CNN)

- R-CNN (Region with CNN)

- 객체검출 분야에 최초로 딥러닝을 적용한 모델

R-CNN: *Regions with CNN features*



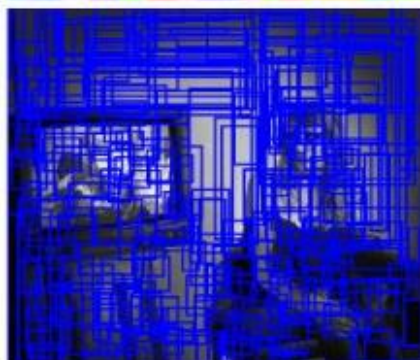
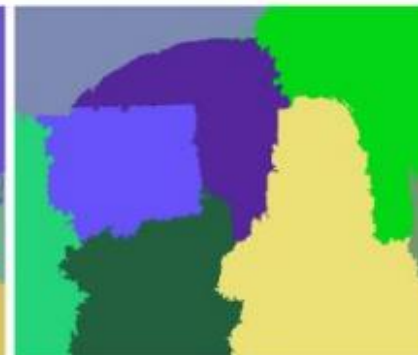
- **Selective search**를 이용해 2,000개의 **ROI(Region of Interest, 관심영역)**을 추출
- 각 ROI에 대해서 동일한 크기의 입력 이미지로 변경
- Warped(변형된) image를 **CNN**에 넣어서 (forward) 영상의 **피쳐맵을 추출 (2,000회)**
- 해당 피쳐맵을 이진분류기인 **SVM**에 넣어 분류 → Class 출력
 - 각 클래스에 대해 독립적으로 훈련된 이진(binary) SVM을 사용
- 해당 피쳐를 regressor에 넣어 **위치(bounding box)**를 예측

R-CNN (Region-based CNN)

- **Selective Search (선택적 탐색)을 이용한 영역제안(Region proposal)**
 - 컬러, 무늬(Texture), 크기(Size), 형태(Shape)에 따라 유사도가 비슷한 영역들을 반복적으로 그룹핑하여 **2,000개의 영역(Bounding box)을 제안**
- R-CNN의 문제점
 - Selective search로 이미지당 2,000 영역제안에 많은 연산시간 소요
→ 사전 추출 필요, 종단(end-to-end) 학습 불가
 - Selective search, CNN 특징추출, SVM과 Bounding box regressor로 복잡한 구성



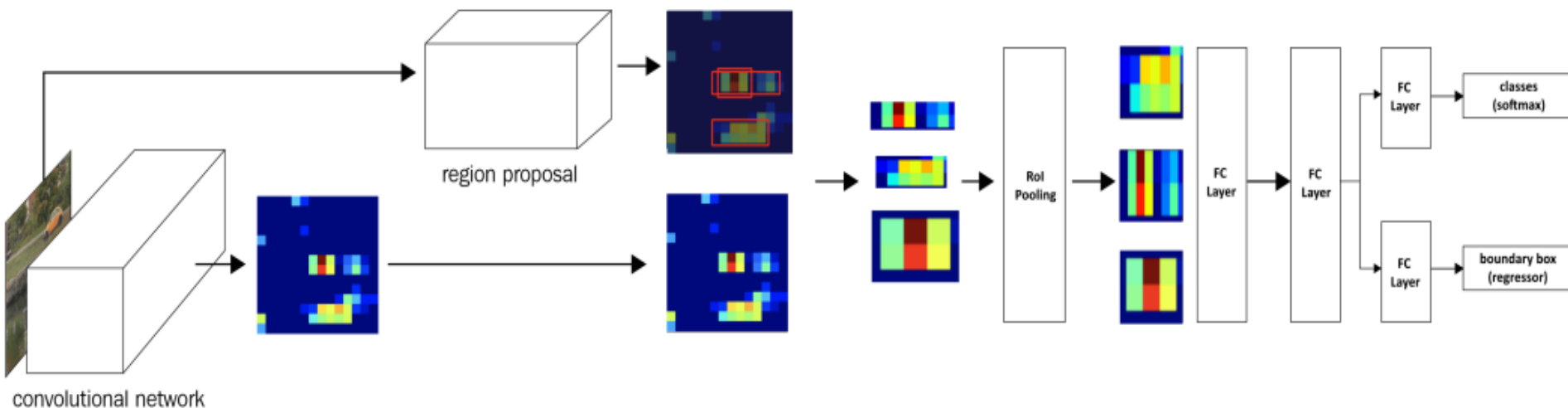
Input Image



Fast R-CNN

- ROI + CNN (2,000회) → CNN (1회) + ROI

- SVM을 softmax 함수로 변경 → Multi-task loss 함수로 분류와 회귀 동시 최적화
- 동일한 Region Proposal을 이용하되 이미지당 한 번만 CNN 연산(forward) 수행

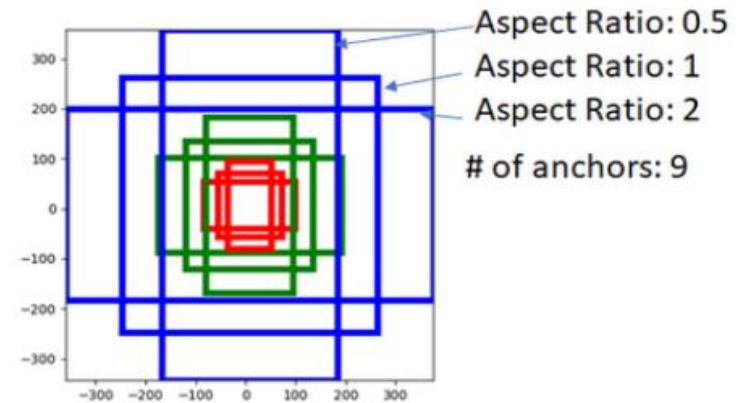


- **Selective Search**로 원본 이미지에서 객체에 해당하는 영역을 제안
- CNN 모델을 통해 **피쳐맵 생성 (1회)**
- 영역 제안에 해당하는 특징맵 영역을 추출 → 연산량 감소
예) 원본 이미지에서 (40, 32, 200, 240) 영역은 특징맵에서는 (5, 4, 25, 30)에 해당
- 영역 제안에 해당하는 특징맵을 **RoI 풀링**으로 (7x7) 특징맵으로 만듦.
- RoI pooling layer 출력값을 FC 레이어에 전달
- 각 영역 제안에 해당하는 클래스 및 오프셋을 예측하도록 모델을 학습시킴.

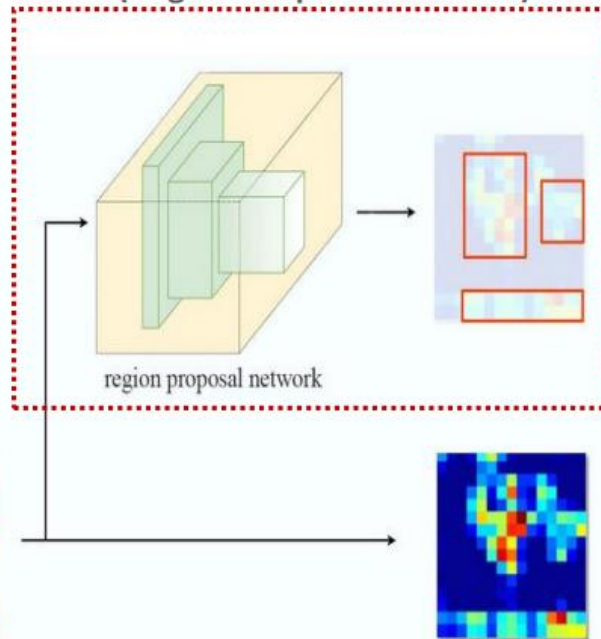
Faster R-CNN

- **Fast RCNN + RPN → Faster RCNN**

- RPN (Region Proposal Network) : GPU 수행
- 피쳐맵 픽셀당 9개의 **Anchor Box** 적용
(size 3 x ratio 3)

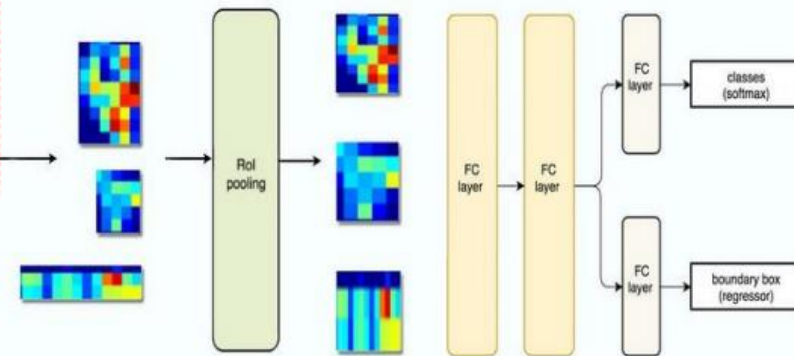


RPN (Region Proposal Network)



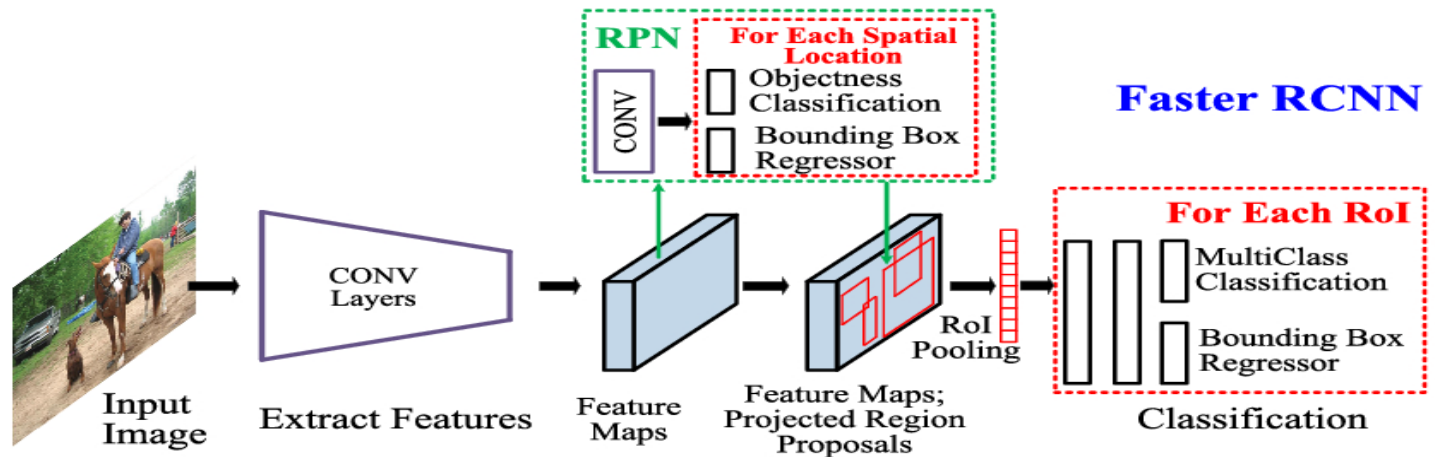
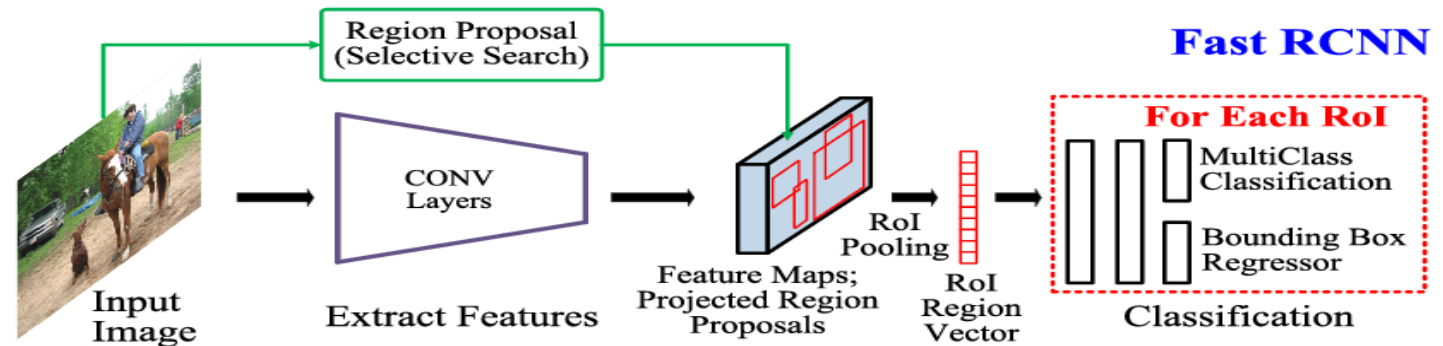
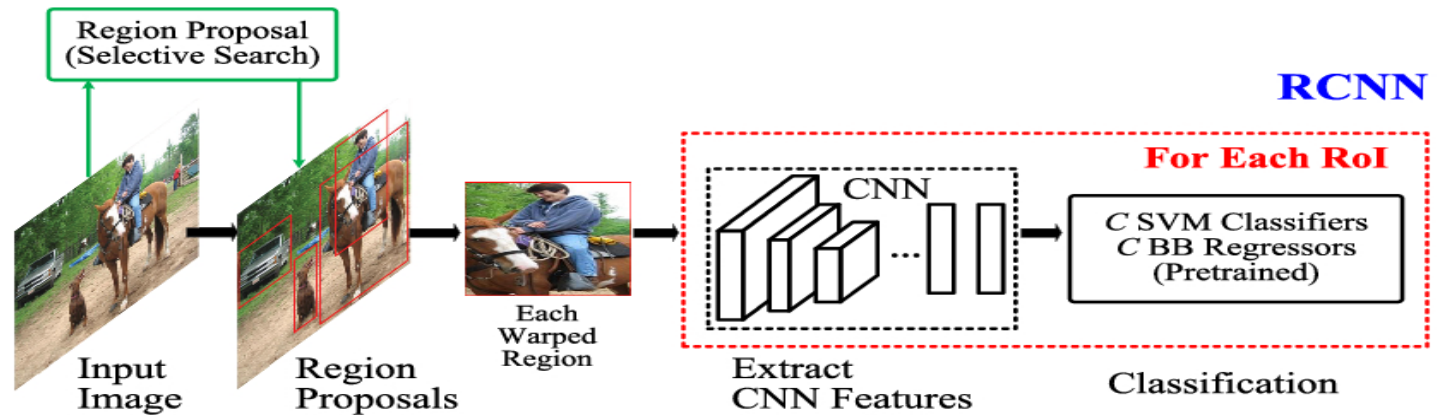
Selective Search를 Neural Network 구조로 변경

- GPU 사용으로 빠른 학습/Inference
- End to End Network 학습



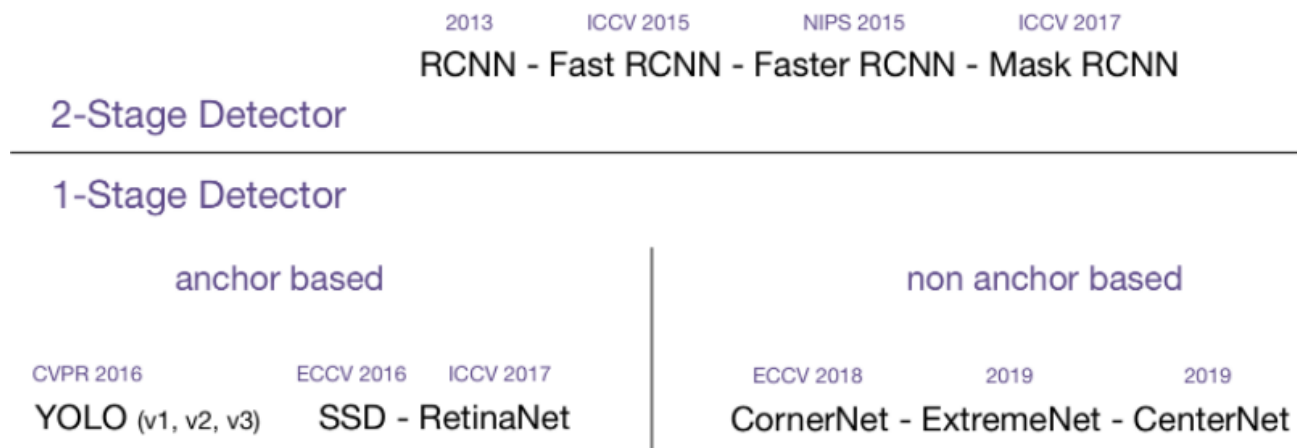
Faster R-CNN

- R-CNN의 발전



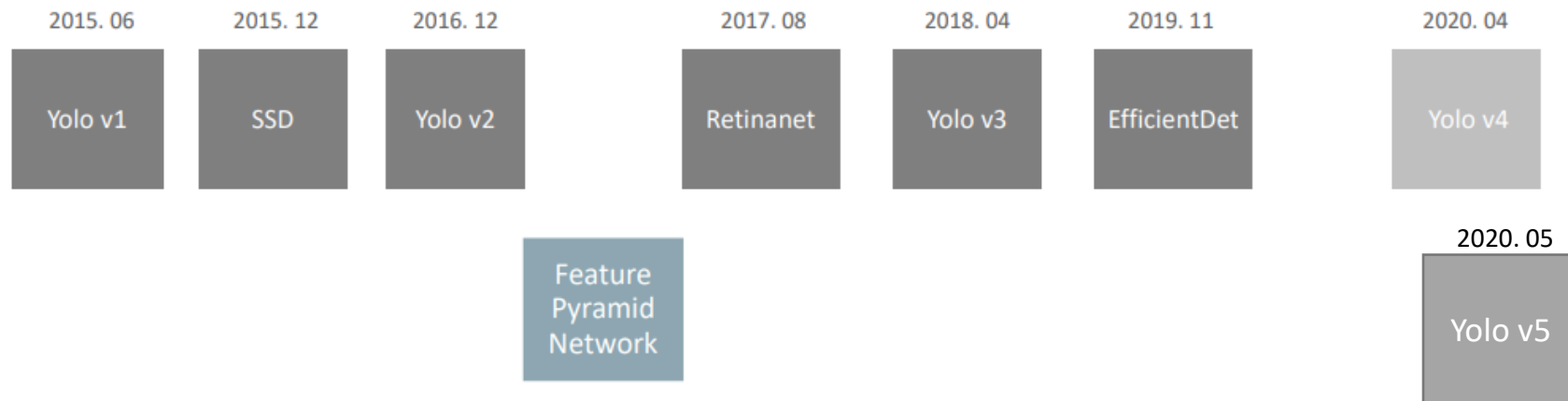
1-Stage Detector

- One stage detector



- YOLO (You Only Look Once)

- Darknet (C로 작성된 딥러닝 프레임워크) 기반 (v1~v4) → pytorch 기반 (v5)
- Joseph Redmon (v1~v3) → Alexey Bochkovskiy (v4) → Ultralytics/Glenn Jocher (v5)



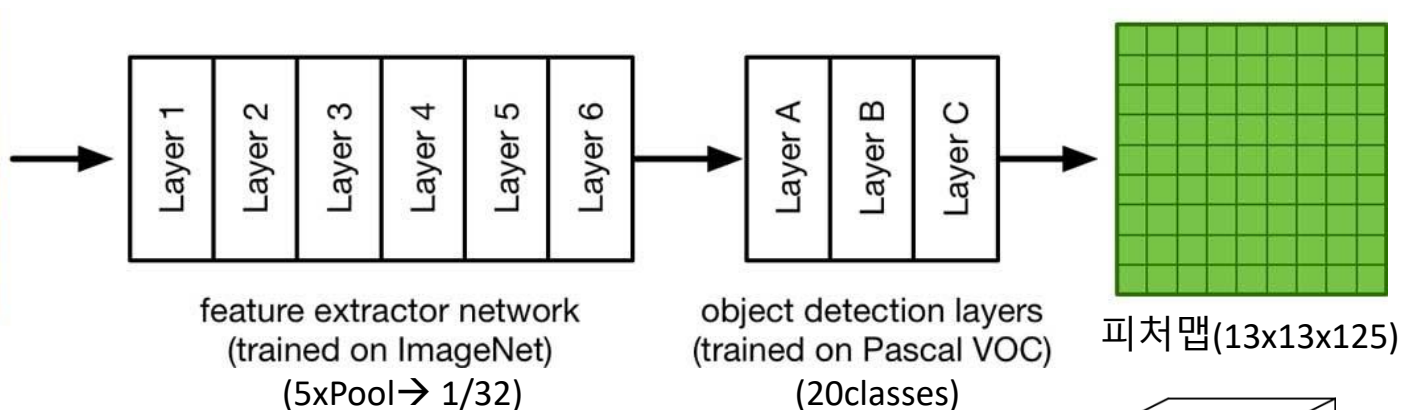
YOLO 계열

- YOLO의 구조

- 후보영역을 추출하기 위한 별도의 네트워크가 없음 → 고속
- 입력 영상을 $s \times s$ 그리드(grid)로 분할한 후 그 셀(cell)에 대해 객체검출을 수행
- 입력영상 : 416x416 이미지 → 출력 피쳐맵 : 13x13 생성



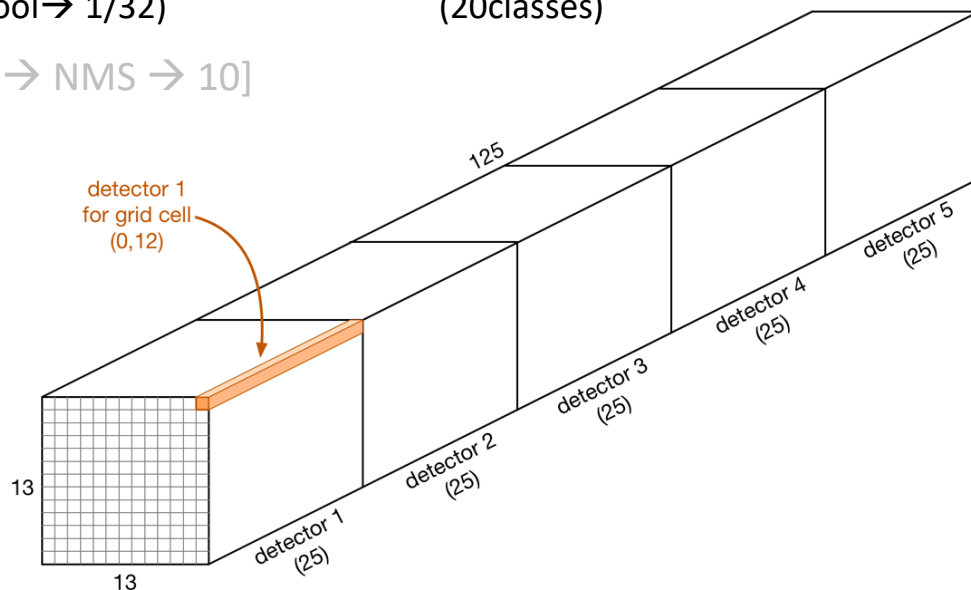
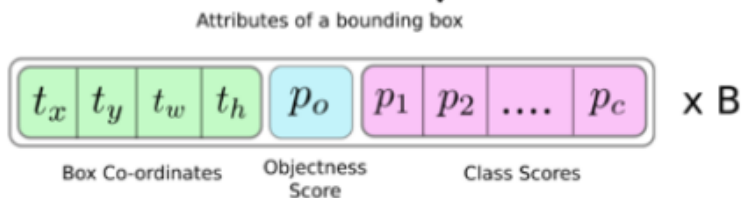
영상(416x416)



- 검출기(Detector) 수 = $13 \times 13 \times 5 = 845$ [→ NMS → 10]

- (4) Box 좌표(cx, cy, w, h)
- (1) Objectness(Confidence) score
- (20) Class score
- (5) Anchor

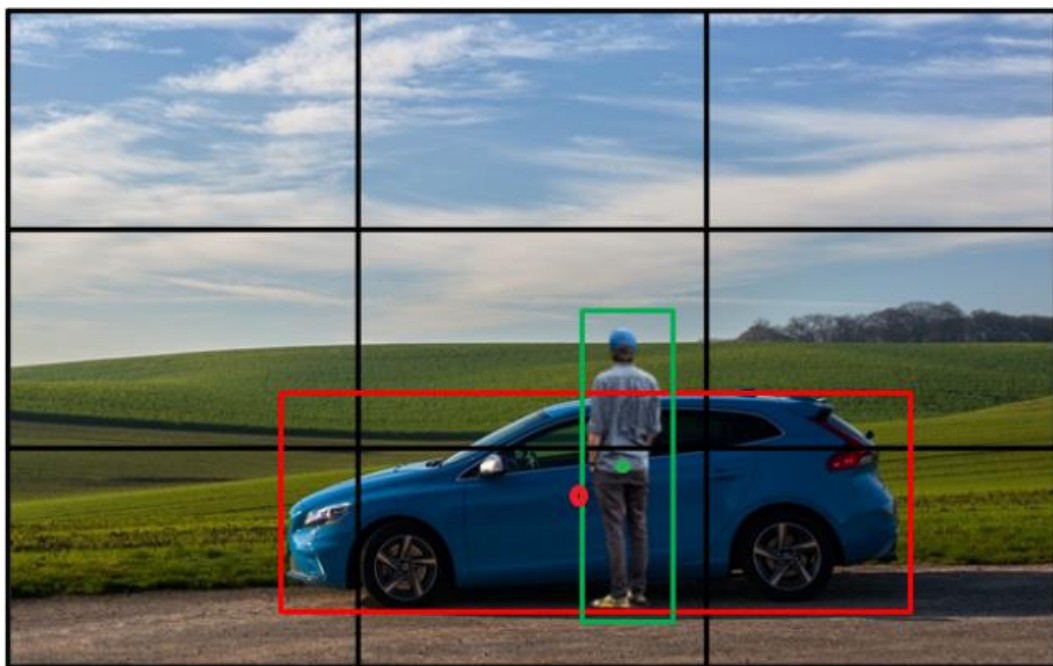
Bounding Box별 25개 정보



YOLO 계열

- YOLO 객체검출 예

- 3x3 Grid, 2x Anchors, 3 Classes → $3 \times 3 \times [pc, (bx, by, bh, bw), (c1, c2, c3)] \times 2$
- if $pc < \text{threshold}$ → not calculate the regression loss and classification loss
- if $pc > \text{threshold}$ → calculate the regression loss and classification loss



Anchor box 1



Anchor box 2



- 모델의 정확도, 복잡성
- pc : objectness score
- bx, by : position offset
- bh, bw : scaling factor
- Confidence in class = $pc \times c$

y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3
	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

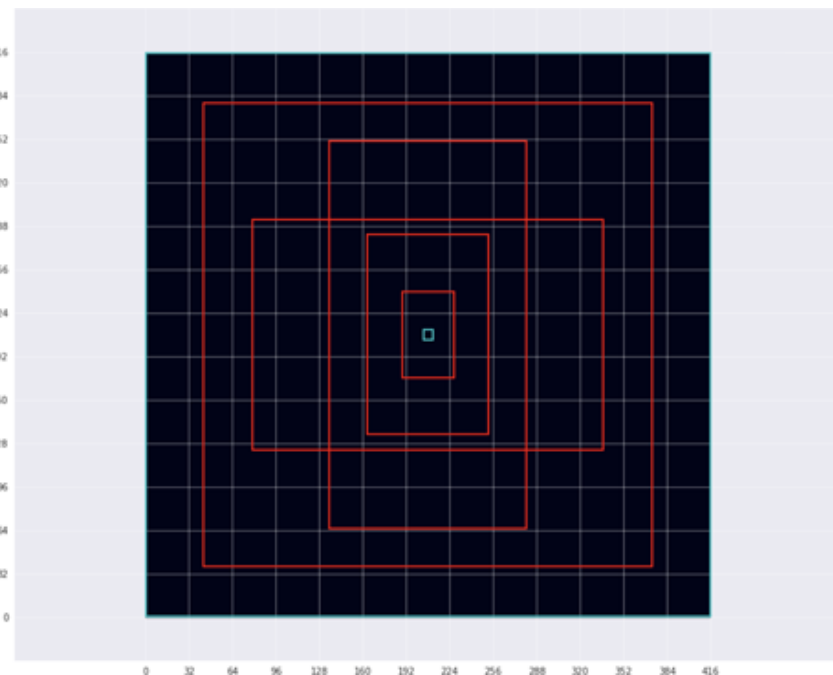
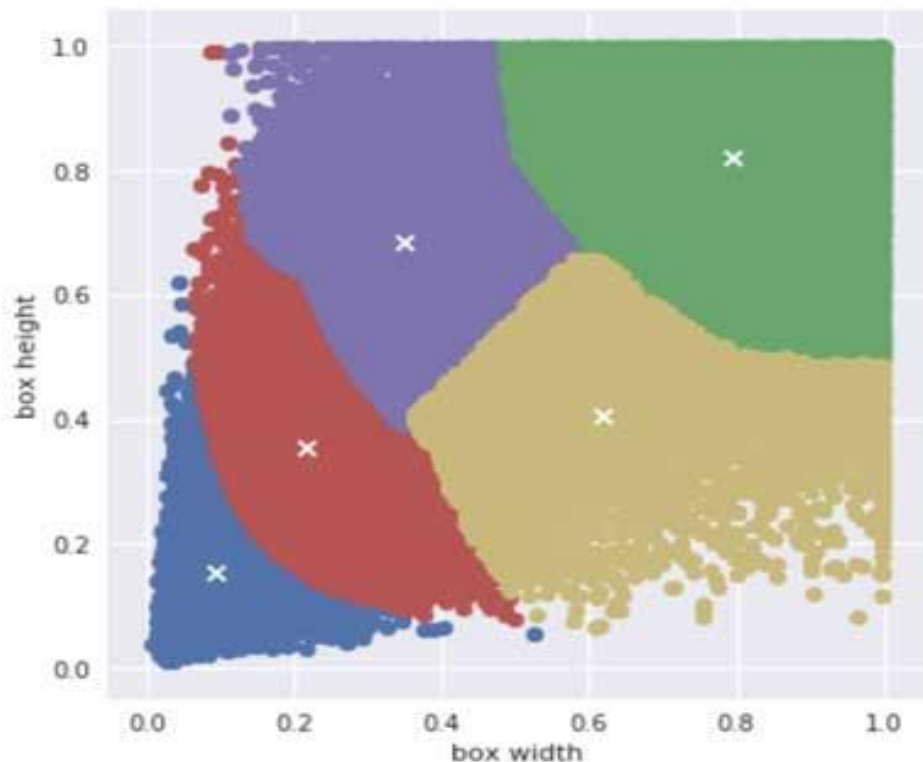
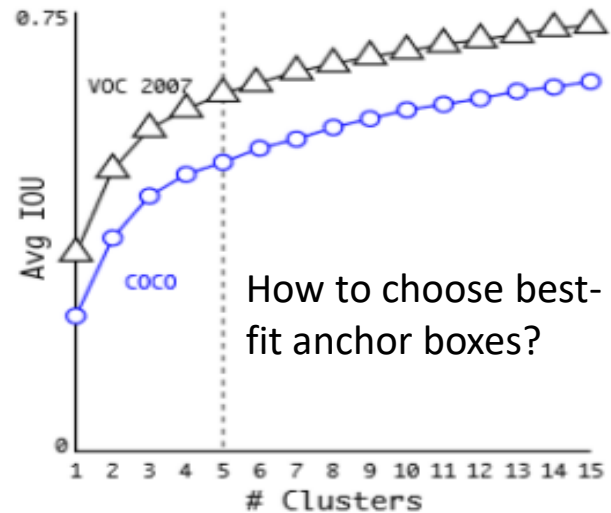
YOLO 계열

- YOLO v1, v2, v3의 비교
 - You only look once: Unified, real-time object detection
 - YOLO9000: Better, Faster, Stronger
 - YOLOv3: An Incremental Improvement
 - YOLOv4: Optimal Speed and Accuracy of Object Detection
 - YOLOv5: <https://www.github.com/ultralytics/yolov5>
 - [YOLO의 역사](#)

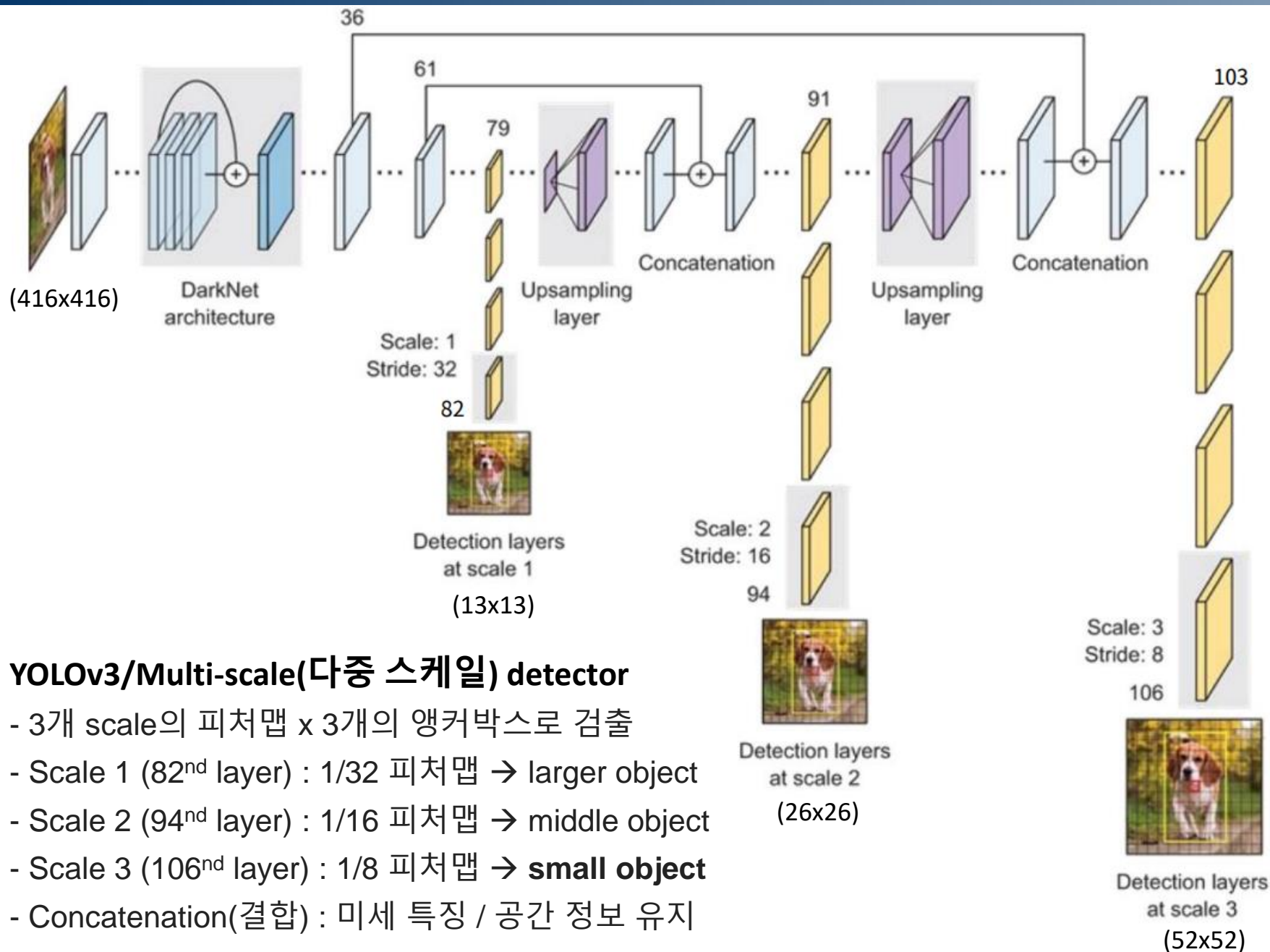
항목	V1	V2	V3
원본 이미지 크기	446 X 446	416 X 416	416 X 416
Feature Extractor	Inception 변형	Darknet 19	Darknet 53
Grid당 Anchor Box 수	2개(anchor box는 아님)	5개	Output Feature Map당 3개 서로 다른 크기와 스케일로 총 9개
Anchor box 결정 방법		K-Means Clustering	K-Means Clustering
Output Feature Map 크기 (Depth 제외)	7 x 7	13 x 13	13 x13, 26 X 26, 52X52 3개의 Feature Map 사용
Feature Map Scaling 기법			FPN(Feature Pyramid Network)

YOLO 계열

- 앵커(Anchor, 닳) 박스 = 미리 정의한/default Bbox
 - 대부분의 물체는 비슷한 모양을 가짐
 - 학습 데이터의 Bbox를 대상으로 K-means clustering 수행
 - 가장 대표적인 5개의 앵커 **box** 설정
 - grid당 앵커수 이상의 객체는 검출 불가



YOLO 계열



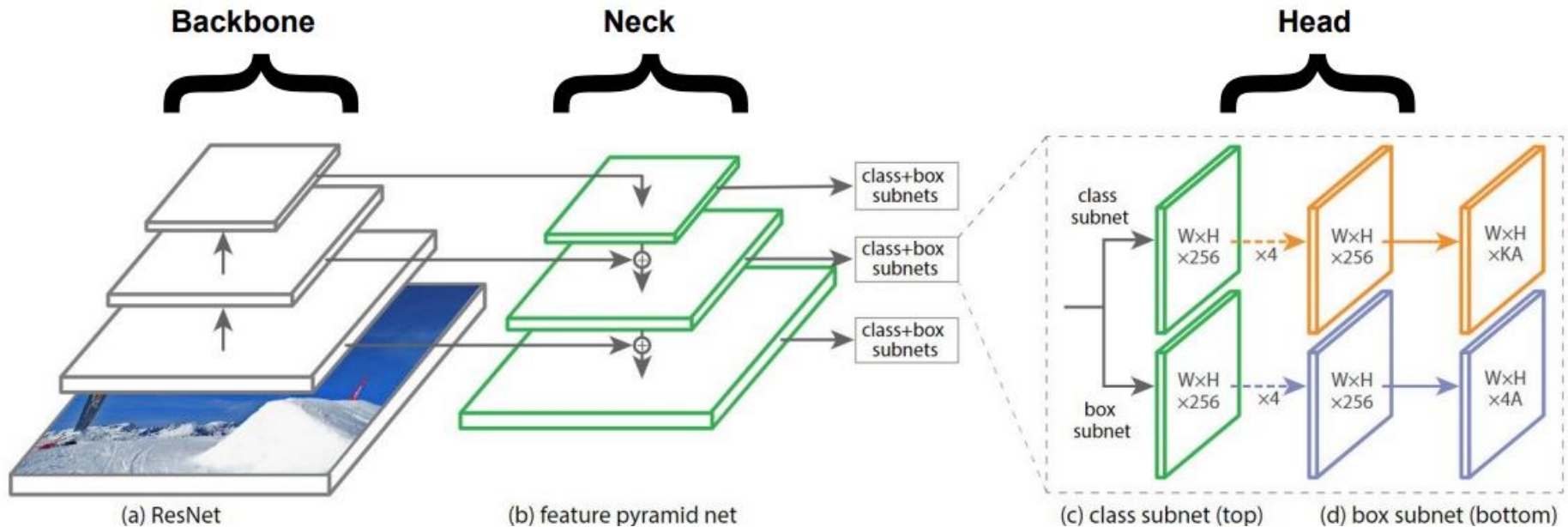
- YOLOv3/Multi-scale(다중 스케일) detector

- 3개 scale의 피쳐맵 x 3개의 앵커박스로 검출
- Scale 1 (82nd layer) : 1/32 피쳐맵 → larger object
- Scale 2 (94nd layer) : 1/16 피쳐맵 → middle object
- Scale 3 (106nd layer) : 1/8 피쳐맵 → **small object**
- Concatenation(결합) : 미세 특징 / 공간 정보 유지

YOLO 계열

- YOLOv3/객체검출 네트워크의 구조

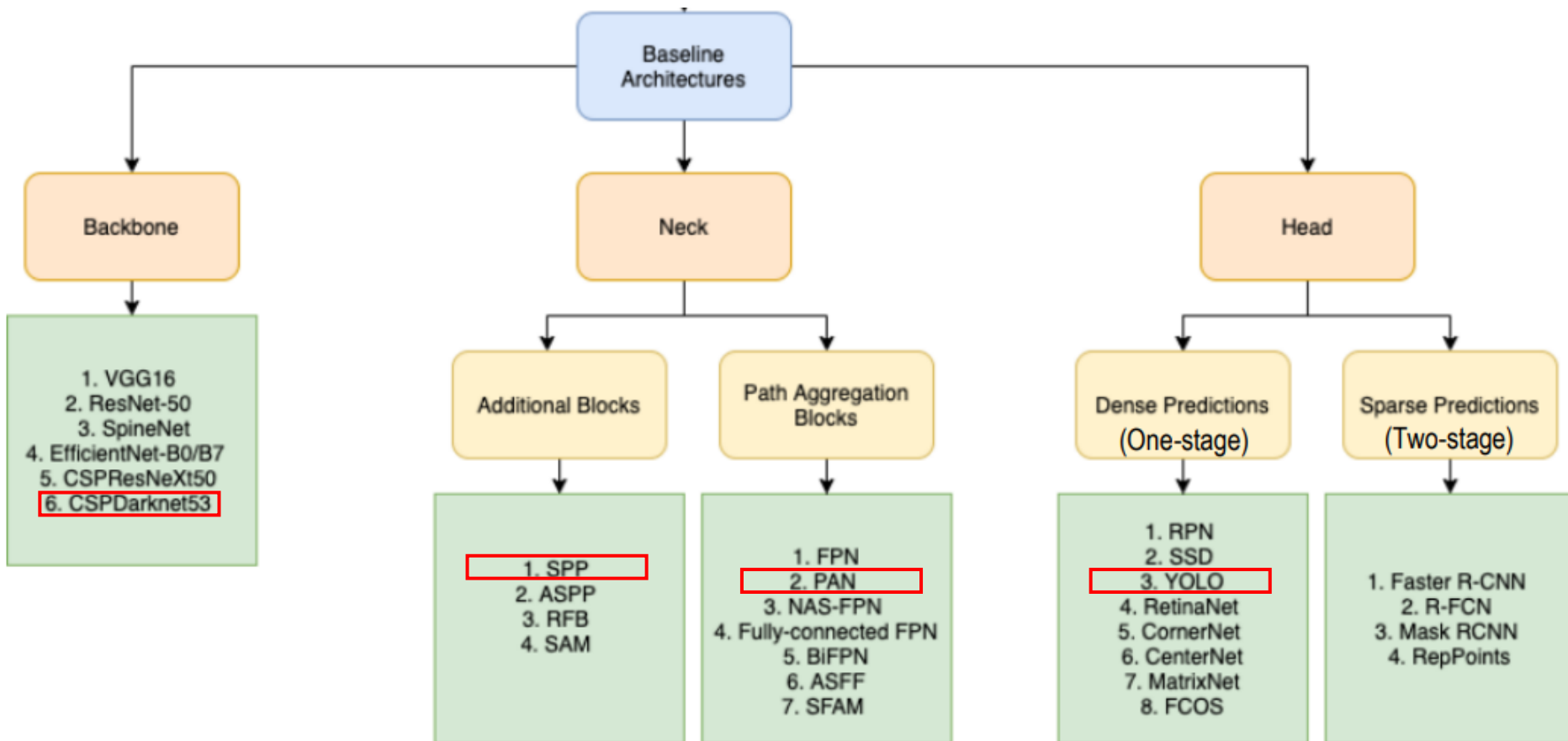
- Backbone(Base) : 입력영상에서 특징맵을 추출 (feature extractor)
- Neck : Backbone에서 생성된 특징을 융합(mix + combine)
- Head : 각 bounding box에 대해 검출(classification + localization)을 수행



YOLO 계열

- YOLOv4

- 아키텍처 각 부분에 대해 컴퓨터 비전의 가장 진보된 기법으로 수많은 실험을 수행
- 정확도와 속도를 최적화

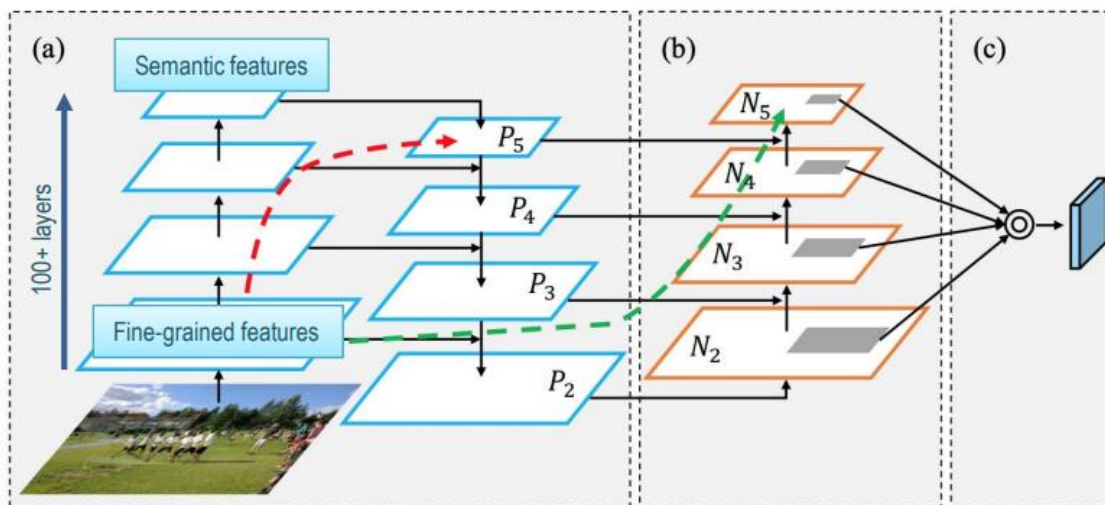


- CSP (Cross Stage Partial)
- SPP(Spatial Pyramid Pooling) block
- PAN(Path Aggregation Network) : advanced version of FPN(Feature Pyramid Network)

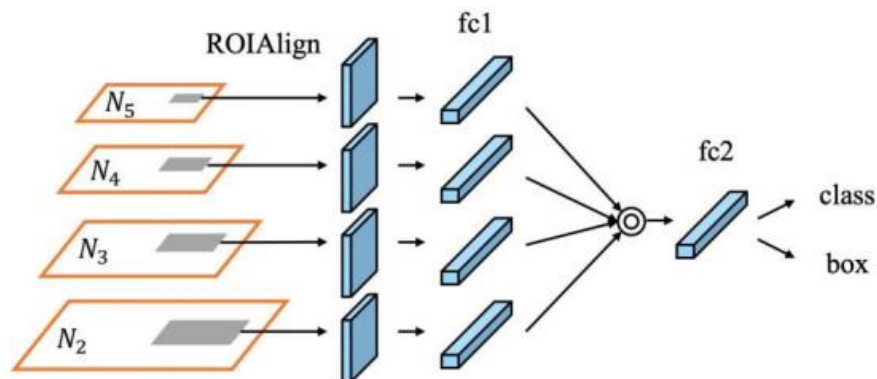
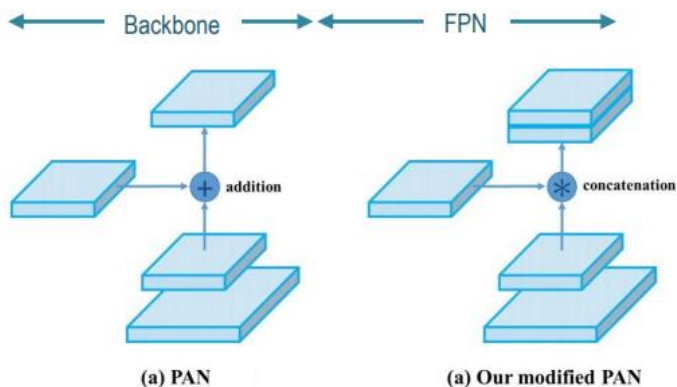
YOLO 계열

- (v3) FPN (Feature Pyramid Network) → (v4) **PAN (Path Aggregation Network)**

- 작은 객체를 보다 잘 검출하기 위해서 다양한 크기의 피쳐맵을 활용
- (a) 하위 low level 피쳐맵의 상세정보와 상위 high level 피쳐맵의 추상화된 정보를 융합
- (b) 융합의 효율성을 위해 **bottom-up path를 추가** (red long path → blue 'shortcut' path)
- (c) (v3) Scale별 검출 수행 → (v4) 모든 피쳐맵을 융합/통합한 후 검출 수행



(a) FPN backbone
(b) bottom-up path augmentation
(c) adaptive feature pooling



YOLO 계열

- Glenn Jocher의 YOLOv5

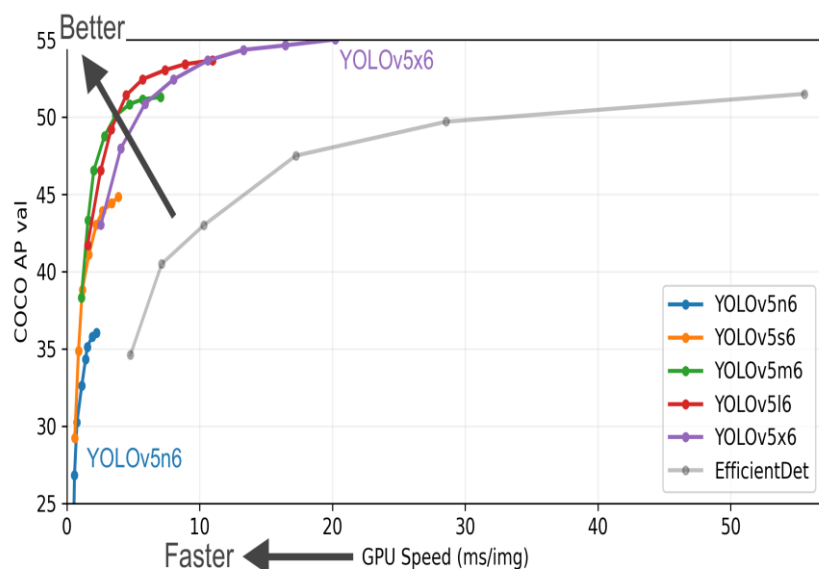
- Ultralytics LLC의 CEO, Mosaic data augmentation의 발명자
- YOLOv5와 거의 비슷하지만 개발환경의 편의성, 우수한 성능으로 대세가 됨
- **Adaptive anchor boxes** 선정

(V2) COCO dataset (80 classes) → 5 best-fit anchors → 다른 클래스/데이터에 대해서?

(V5) 자동적으로 최적의 앵크 박스를 선택하는 과정을 YOLOv5에 통합

- Ultralytics GitHub : <https://github.com/ultralytics/yolov5>

- YOLOv5의 모델 선정 (COCO val)



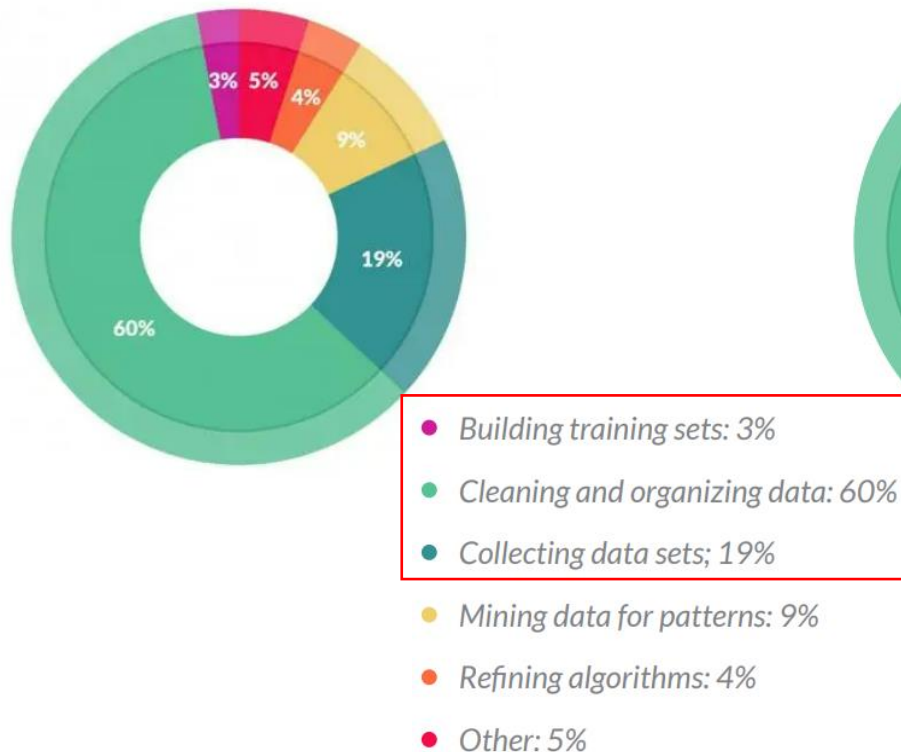
Model	size (pixels)	mAP ^{val} 0.5:0.95	mAP ^{val} 0.5	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	45	6.3	0.6	1.9	4.5
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5	76.8	111.4
YOLOv5x6	1280	55.0	72.7	3136	26.2	19.4	140.7	209.8
+ TTA	1536	55.8	72.7	-	-	-	-	-

주요 데이터셋과 주석(annotation) 파일

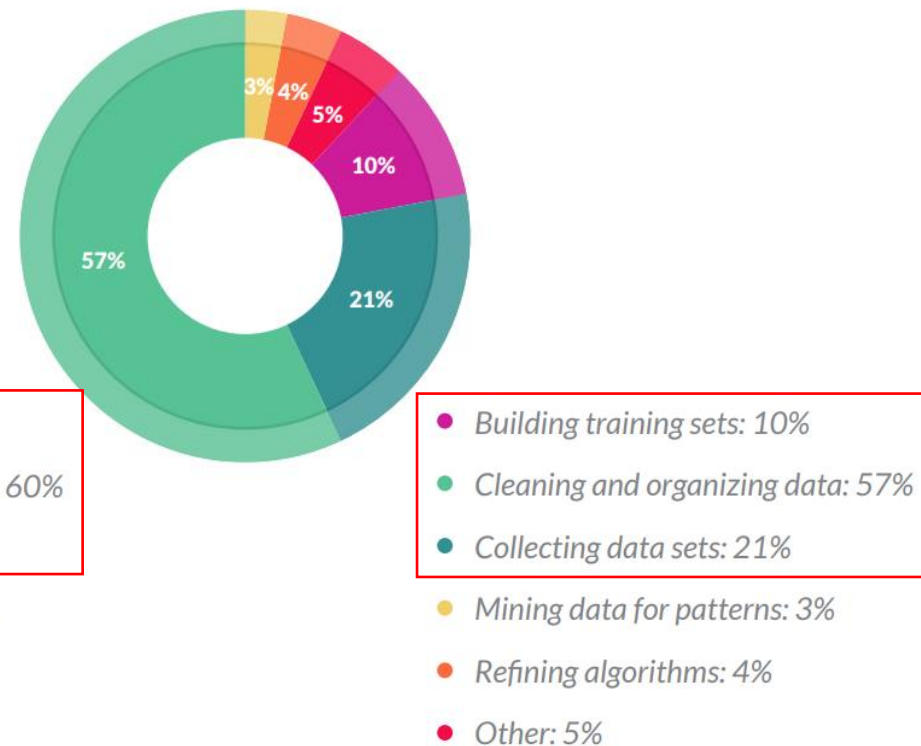
- 데이터셋의 중요성 : **garbage in, garbage out (GIGO)**

- 딥러닝 등 '학습' 기반 모델의 성능은 학습시킨 데이터에 크게 의존 → 데이터의 양과 질이 중요
- 모델의 성능을 향상시키기 위해서는 매우 다양하고 균형잡힌(balanced) 많은 데이터가 필요
 - * 빅데이터 4V : **대용량(Volume)**, **다양성(Variety)**, **정확성(Veracity)**, **고속(Velocity)**
- 빅데이터를 직접 수집하는 데에는 한계가 있으므로 **데이터 증량(augmentation)**을 사용
- 데이터셋 구축은 많은 비용과 시간이 필요하며 실제 데이터 과학 연구의 80% 이상을 차지

What data scientists spend the most time doing



What's the least enjoyable part of data science?



주요 데이터셋 및 주석(annotation) 형식

- **PASCAL VOC (Visual Object Classes) - XML format**
 - 20개 오브젝트 카테고리 / 27,450개의 영상
 - 모델 평가(AP)에 있어 표준이 되었던 데이터셋이지만 최근에는 더 큰 데이터셋의 등장으로 조금씩 밀려나는 추세
- **MS COCO – JSON format**
 - 80개 오브젝트 카테고리 / 30만 개의 영상과 개당 평균 5개의 인스턴스로 구성
 - ImageNet에서 물체가 주로 가운데에 위치하고 큰 경향이 있다는 문제를 개선하여 실생활과 유사하게 만들어진 데이터셋으로 현재 객체검출 **성능평가의 표준**이 되는 데이터셋
 - COCO Challenge : 이미지 크기(대/중/소) 및 IoU별 mAP 계산 → 까다로운 대회
- **Google Open Images – csv format**
 - 600개 오브젝트 카테고리(계층화) / 190만 개의 영상을 갖는 최대 규모의 데이터셋
 - 클래스의 계층구조 (Hierarchy for the 600 boxable classes)
- **Kaggle**
 - Competitions(경진대회) : 다양한 분야의 머신러닝 도전 과제
 - Datasets : 데이터를 탐색하고 분석, 공유
 - Code, Kernel : 코드를 탐색하고 실행 (cf. Colab)

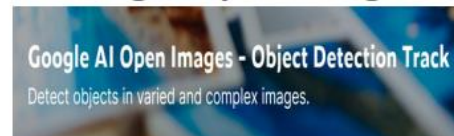
PASCAL VOC



MS COCO



Google Open Images



주요 데이터셋 및 주석(annotation) 형식

• (국내) AI Hub

- 지능정보산업인프라조성 사업으로 추진한 AI 학습용 데이터와 국내외 기관/기업에서 보유한 AI 학습용 데이터를 공개
- 국내 중소벤처기업, 연구소, 개인 등이 높은 비용과 투입시간으로 인해 자체적으로 확보하기 어려운 **양질의 대용량 인공지능 학습용 데이터**를 누구에게나 공개하여 지능정보사회와 인공지능 산업생태계를 조성하는데 기여

음성/자연어

데이터 48종

헬스케어

데이터 35종

안전

데이터 21종

농축수산

데이터 15종

비전

데이터 36종

자율주행

데이터 23종

국토환경

데이터 12종

로 고장 예지 센서

2020

안전

노후 시설물 이미지

이미지

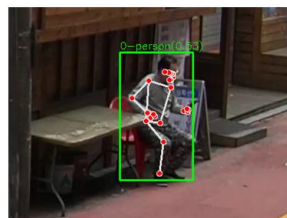
2020



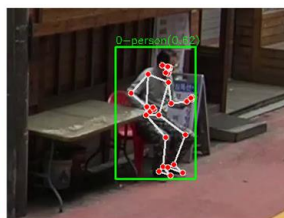
(a) 흰 연기



(b) 검은 연기와 불꽃



(c) 오류2 - 자세 추정 부분 실패



(d) 성공 - 자세 추정 성공

안전
시니어 이상행동 영상

이미지 비디오

2020

안전
안면인식 영상

이미지

2020

안전
위급상황 음성/음향

텍스트 오디오

2020

안전
이상행동 CCTV 영상

비디오

2019

안전
전력 설비 에너지 패턴 및 고장 분석 센서

센서

2020

안전
졸음운전 예방을 위한 운전자 상태 정보 ...

이미지

2020

안전
해상 객체 이미지

이미지

2020

안전
화재 발생 예측 영상

이미지

2020

안전
어린이 보호구역 내 도로보행 위험행동 ...

이미지 비디오

2020

안전
자동차 차종/연식/번호판 인식용 영상

이미지

2020

안전
지하철 역사 내 CCTV 이상행동 영상

이미지

2020

안전
화주로 내 이상물체 감지 객체 이미지

이미지

2020

주석/레이블 파일 형식

- Annotation(주석) / label(레이블)

- 영상에 대한 위치, 클래스, 속성 등의 정보를 갖고 있는 텍스트 설명 파일
- 주석 파일의 형식은 데이터셋에 따라 다르며 주로 xml, json, csv, txt를 사용

원본 이미지



plane.jpg



Annotation

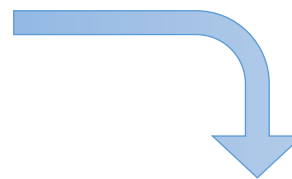
```
<annotation>
  <folder>VOC2012</folder>
  <filename>2007_000032.jpg</filename>
  <source>
    <database>The VOC2007 Database</database>
    <annotation>PASCAL VOC2007</annotation>
    <image>flickr</image>
  </source>
  <size>
    <width>500</width>
    <height>281</height>
    <depth>3</depth>
  </size>
  <segmented>1</segmented>
  <object> ← 개별 Object 정보
    <name>aeroplane</name>
    <pose>Frontal</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>104</xmin>
      <ymin>78</ymin>
      <xmax>375</xmax>
      <ymax>183</ymax>
    </bndbox>
  </object>
  <object> ← 개별 Object 정보
    <name>aeroplane</name>
    <pose>Left</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>133</xmin>
      <ymin>88</ymin>
      <xmax>197</xmax>
      <ymax>123</ymax>
    </bndbox>
  </object>
</annotation>
```

개별 Object의
bounding box 정보 →

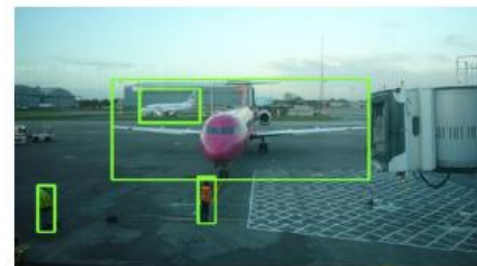
개별 Object의
bounding box 정보 →

plane.xml

- 개별 영상 주석파일
 - PASCAL VOC (*.xml)
 - YOLOv5 (*.txt)
- 통합 주석파일
 - MS COCO (annotations.json)
 - Open Images Dataset (ann.csv)



원본 이미지에 Bounding Box 시각화



주석/레이블 파일 형식

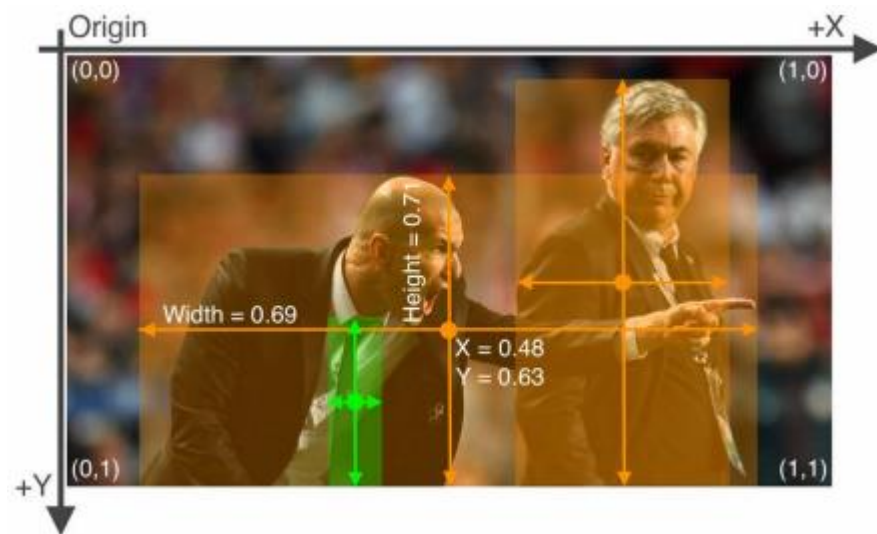
- YOLOv5의 주석 파일 형식 (*.txt)
 - 5개의 값이 공백으로 분리
 - (1열) Class ID : 0~79 for COCO dataset
 - (2~5열) Bounding box 좌표 : ncx, ncy, nw, nh [0~1]



.../images/.../zidane.jpg

```
0 0.481719 0.634028 0.690625 0.713278
0 0.741094 0.524306 0.314750 0.933389
27 0.364844 0.795833 0.078125 0.400000
```

.../lables/.../zidane.txt

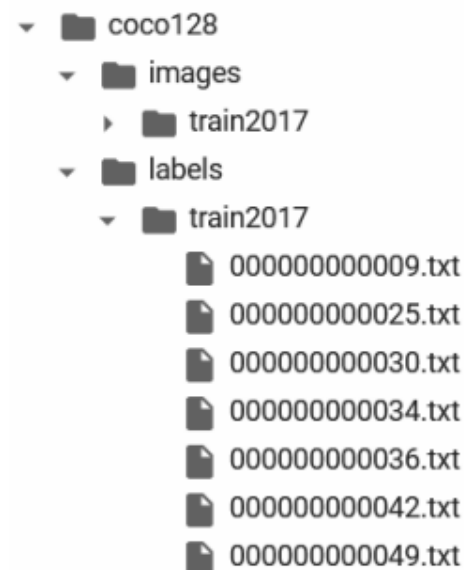
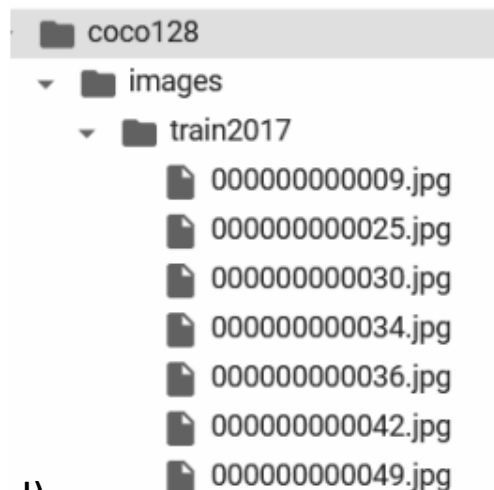


주석/레이블 파일 형식

- YOLOv5의 Dataset 구조

- 영상별로 주석파일(*.txt)이 매칭

영상 폴더



- 학습용 환경설정파일 (coco128.yaml)

- 영상 폴더는 반드시 'images' 문자열을 포함하도록 지정
- 주석 폴더는 자동적으로 영상폴더에서 'images'만 'labels'로 대체하여 지정됨

```
# download command/URL (optional)
download: https://github.com/ultralytics/yolov5/releases

# train and val data as 1) directory: path/images/, 2)
train: ../coco128/images/train2017/ # 128 images
val: ../coco128/images/train2017/ # 128 images

# number of classes
nc: 80
```

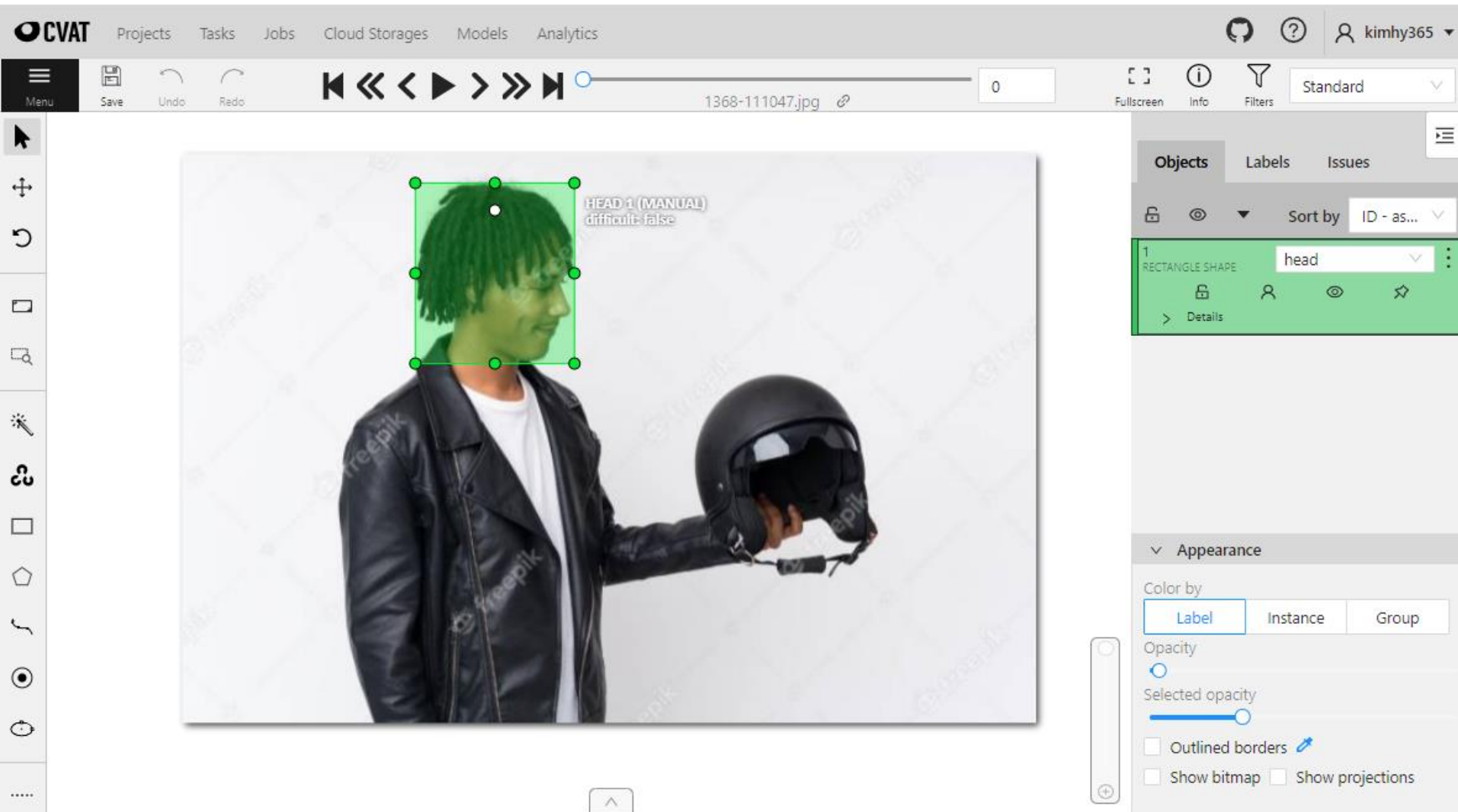

주석툴(Annotation Tool)

Annotation tool	Annotation types	Output formats
Label me	Bounding boxes polygons	Labelme format, but provides conversion to COCO and PASCAL VOC
LabelIMG	Bounding boxes	PASCAL VOC and YOLO
Microsoft VoTT	Bounding boxes polygons	PASCAL VOC, TFRecords, specific CSV, Azure Custom Vision Service, Microsoft Cognitive Toolkit (CNTK), VoTT
Computer Vision Annotation Tool (CVAT)	Bounding boxes polygons	COCO, CVAT, Labelme, PASCAL VOC, TFRecord, YOLO, etc
VGG Image Annotation Tool (VIA)	Bounding boxes polygons	COCO and specific CSV and JSON

- DarkLabel (Video/Image Labeling and Annotation Tool) <https://darkpgmr.tistory.com/16?category=460964>

주석툴(Annotation Tool)

- CVAT (Computer Vision Annotation Tool)



주석 Tool

• ImLabeler (S/W 등록)

ImLabeler Ver 1.0

파일(File) 테스트(Test) 도구

레이블 파일 변환 레이블 편집 데이터 전처리 유틸리티

그림 폴더 지정 D:\52-Python\3-ObjectDetection\datasets\helmet_shwd\images 레이블 폴더 지정 D:\52-Python\3-ObjectDetection\datasets\helmet_shwd\labels

7581 Images ☒ 레이블 폴더 자동 설정 간격 10 << < > >> X [2 / 7581] 000002.jpg

추가(A) 삭제 파일 저장(S) 데이터셋 저장

000002.txt

ClsId	Left	Top	Width	Height
1	37	32	39	52
1	97	44	28	38
1	165	103	43	55
1	178	71	35	42
1	221	44	30	44
1	249	61	34	51
1	335	60	41	52
1	344	107	41	56
1	372	59	30	51
1	409	77	45	59
0	9	75	37	49

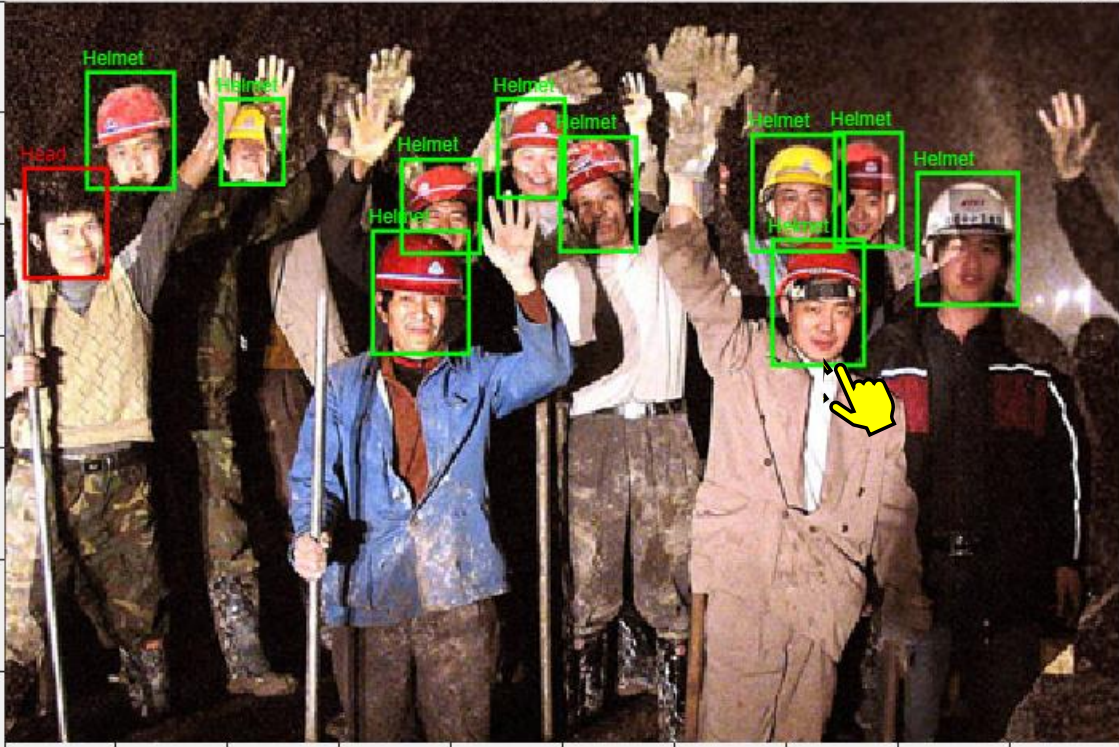
레이블 지정

추가

삭제

ClsId	Clsname
0	Head
1	Helmet

X (500) * Y (332)

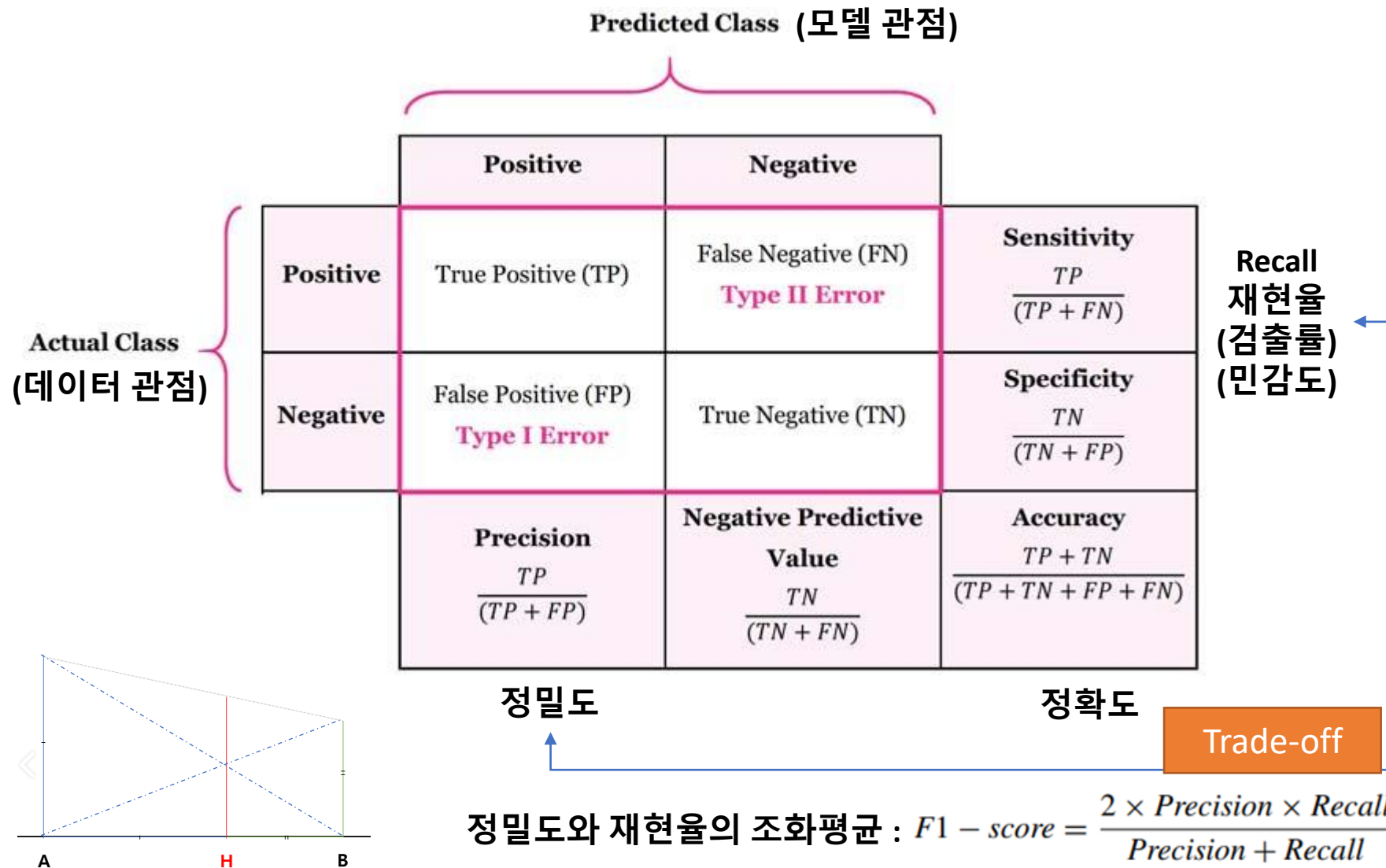


Y

X

평가지표 mAP

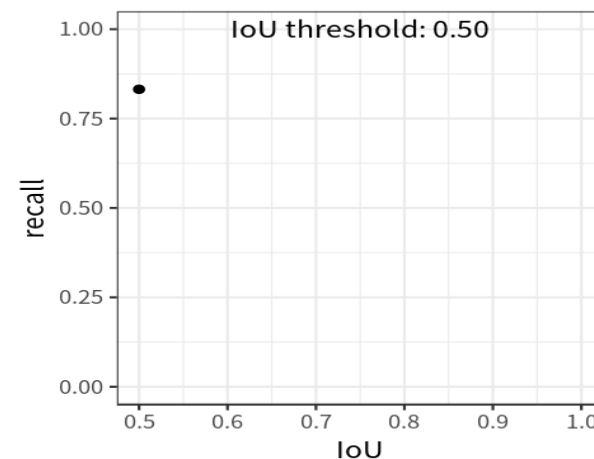
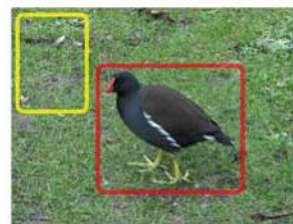
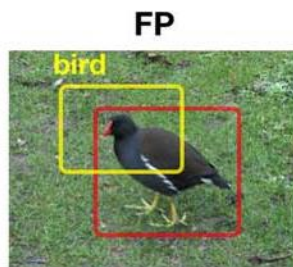
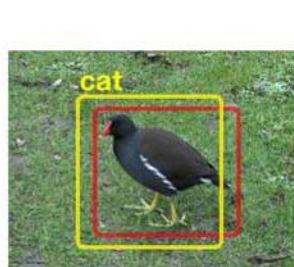
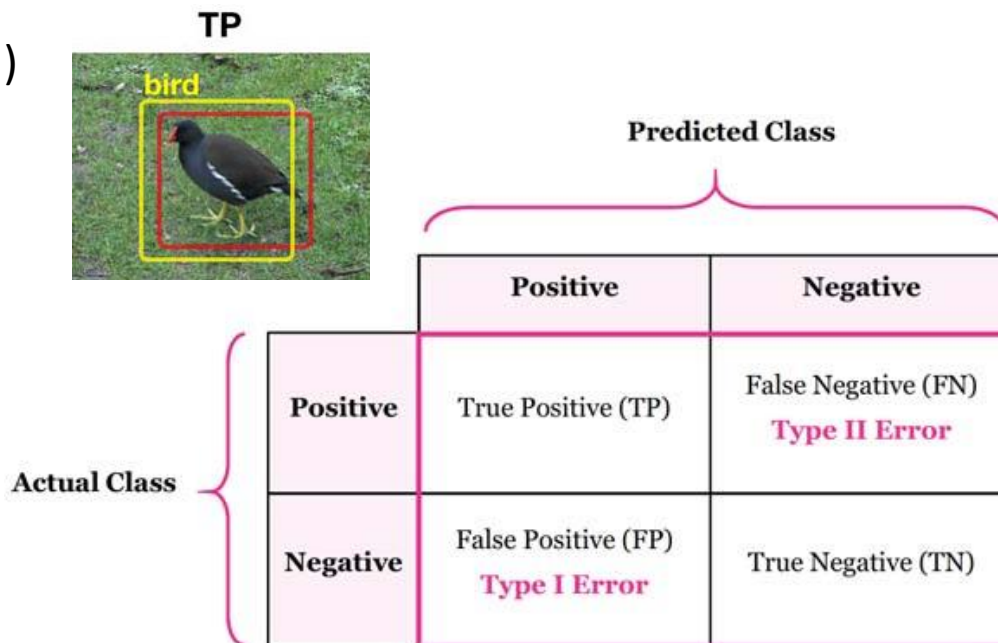
- **Confusion matrix** (혼동/오차 행렬) : 분류기의 성능을 측정하는 도구



평가지표 mAP

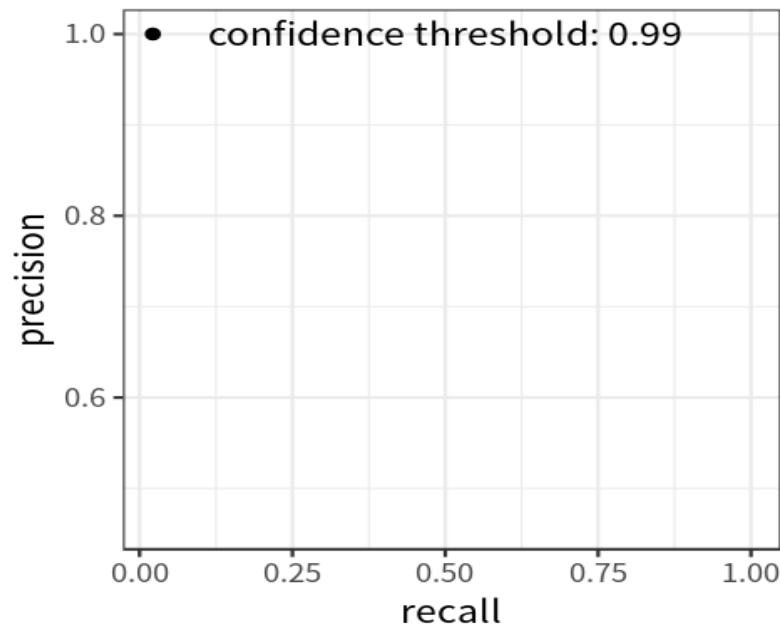
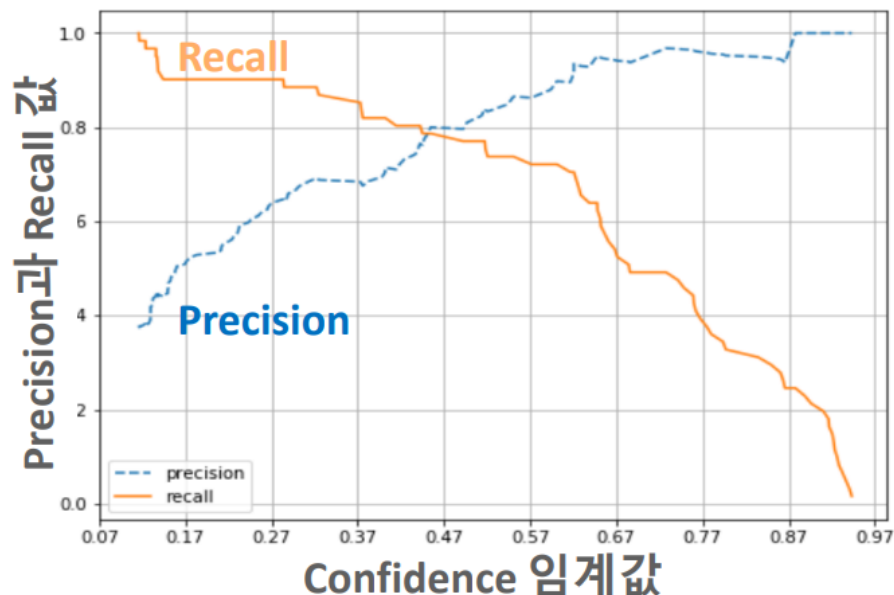
- 객체검출의 분류 판정법

- Classification
- Localization(IoU)



평가지표 mAP

- Confidence 임계값에 따른 Precision/Recall의 변화 → Trade-off 관계
- Precision-Recall(PR) 곡선



- AP (Average Percision)
 - 모델의 성능을 '하나의 값'으로 정량적으로 비교하기 위한 평가지표
 - PR 곡선의 면적 (AUC, Area Under Curve)
- mAP (mean AP)
 - 클래스별 AP를 평균한 값
 - mAP@0.5 – IoU=0.5 (PASCAL VOC)
 - mAP@0.5:0.95 – IoU=0.5, 0.55, 0.6, ... 0.95 (MS-COCO)

평가지표 mAP

3.4.1 Average Precision (AP)

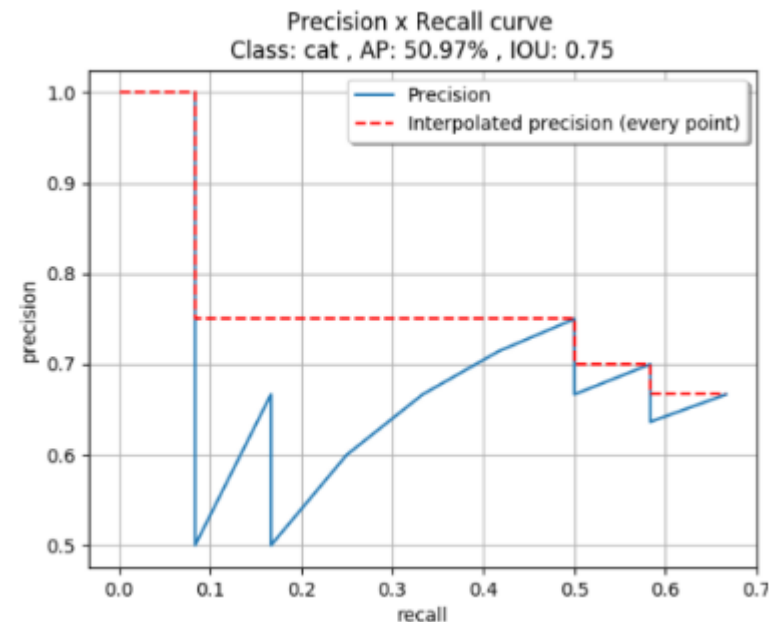
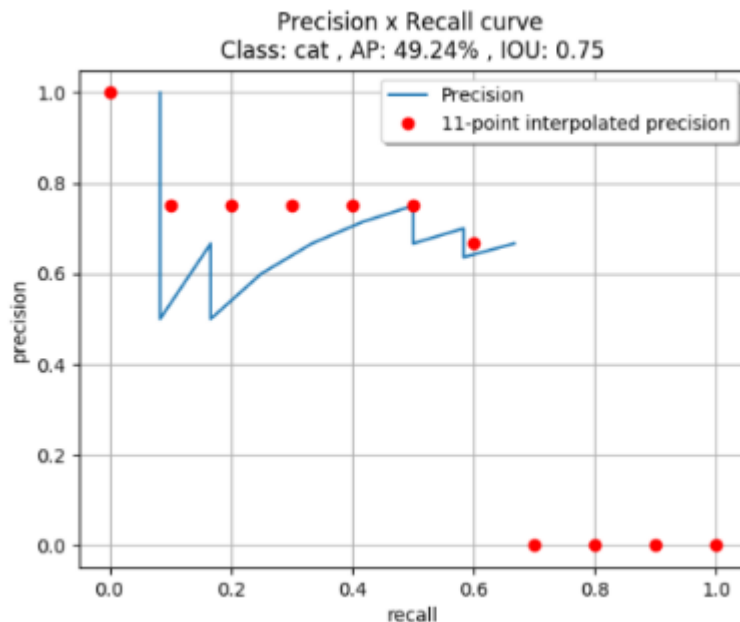
The computation of the average precision (AP) measure was changed in 2010 to improve precision and ability to measure differences between methods with low AP. It is computed as follows:

1. Compute a version of the measured precision/recall curve with precision monotonically decreasing, by setting the precision for recall r to the maximum precision obtained for any recall $r' \geq r$.
2. Compute the AP as the area under this curve by numerical integration. No approximation is involved since the curve is piecewise constant.

Note that prior to 2010 the AP is computed by sampling the monotonically decreasing curve at a fixed set of uniformly-spaced recall values 0, 0.1, 0.2, ..., 1. By contrast, VOC2010–2012 effectively samples the curve at all unique recall values.

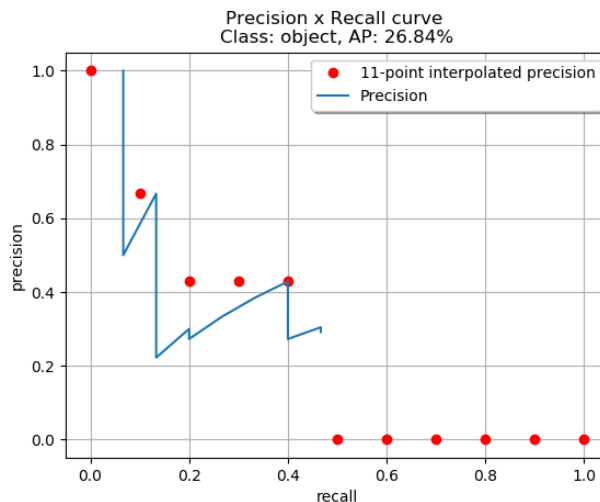
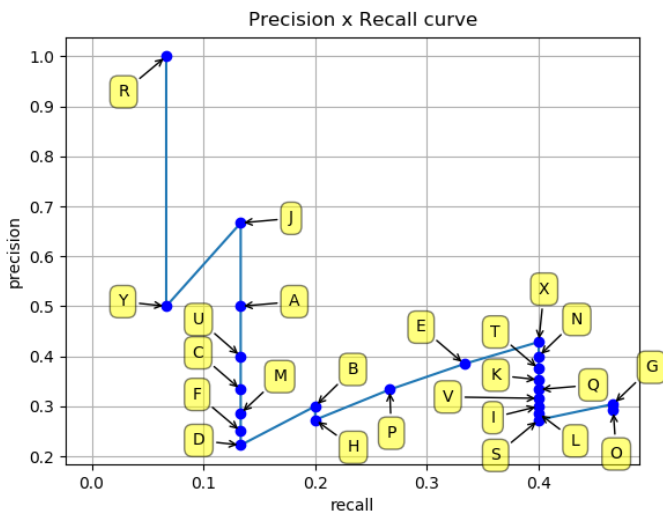
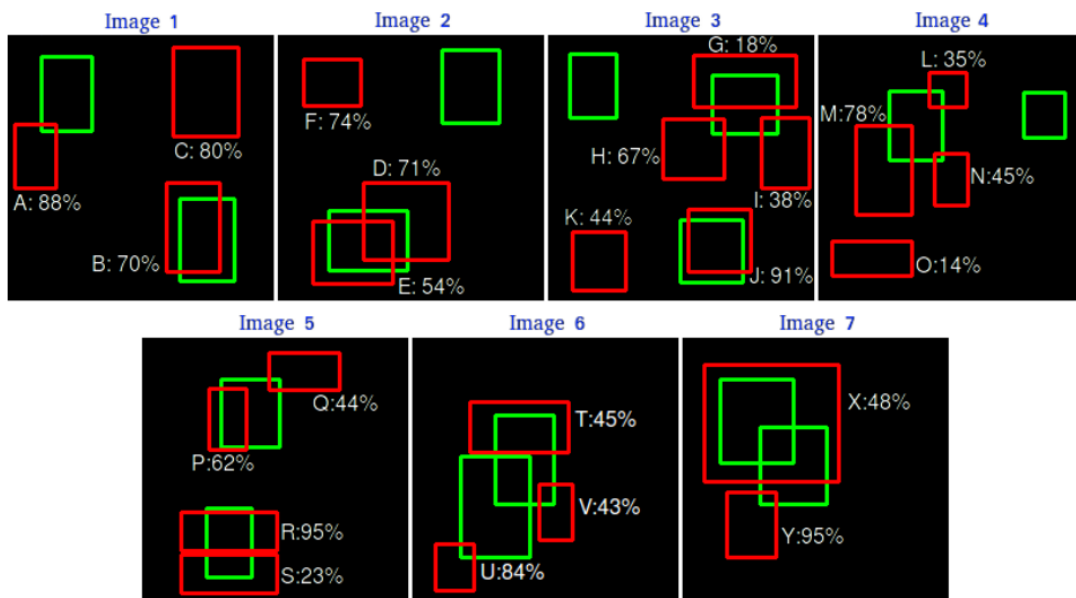
- AP 계산 코드
 - 11-point 보간법(interpolation)
 - **all-point 보간법 (2010년~)**
 - <https://github.com/Cartucho/mAP>

- 출처: PASCAL VOC Challenge 2012 devkit_doc.pdf



평가지표 mAP

- AP 계산 예제/코드 (출처: <https://herbwood.tistory.com/2>)



Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333
Image 2	D	71%	0	1	2	7	0.2222	0.1333
Image 1	B	70%	1	0	3	7	0.3	0.2
Image 3	H	67%	0	1	3	8	0.2727	0.2
Image 5	P	62%	1	0	4	8	0.3333	0.2666
Image 2	E	54%	1	0	5	8	0.3846	0.3333
Image 7	X	48%	1	0	6	8	0.4285	0.4
Image 4	N	45%	0	1	6	9	0.4	0.4
Image 6	T	45%	0	1	6	10	0.375	0.4
Image 3	K	44%	0	1	6	11	0.3529	0.4
Image 5	Q	44%	0	1	6	12	0.3333	0.4
Image 6	V	43%	0	1	6	13	0.3157	0.4
Image 3	I	38%	0	1	6	14	0.3	0.4
Image 4	L	35%	0	1	6	15	0.2857	0.4
Image 5	S	23%	0	1	6	16	0.2727	0.4
Image 3	G	18%	1	0	7	16	0.3043	0.4666
Image 4	O	14%	0	1	7	17	0.2916	0.4666

- AP 계산 코드

- <https://github.com/Cartucho/mAP>

- mAP for YOLOv5 → detect.py
- conf_thres = default (0.001)
- 추론시 : conf_thres=0.25

제2회

+AI 메이커톤 설명회

2021. 10. 5.

INDAI INDUSTRIAL AI
RESEARCH CENTER

진행순서

- 대회 개요
- 출제문제 설명
- 경진대회 진행방식 및 규정
- 결과 제출방법
- 평가지표 및 심사방법
- 대회 일정
- 질의응답
- 서버 사용법 (충북테크노파크)

대회 개요

1. 대회 소개

- 충북대학교 산업인공지능연구센터 주관
- 산업현장의 문제를 인공지능 기술을 적용(+AI)하여 해결하는 경진대회
- AI 기술의 대중화와 이해증진을 목표로 **최적의 AI 모델을 개발하는 대회**

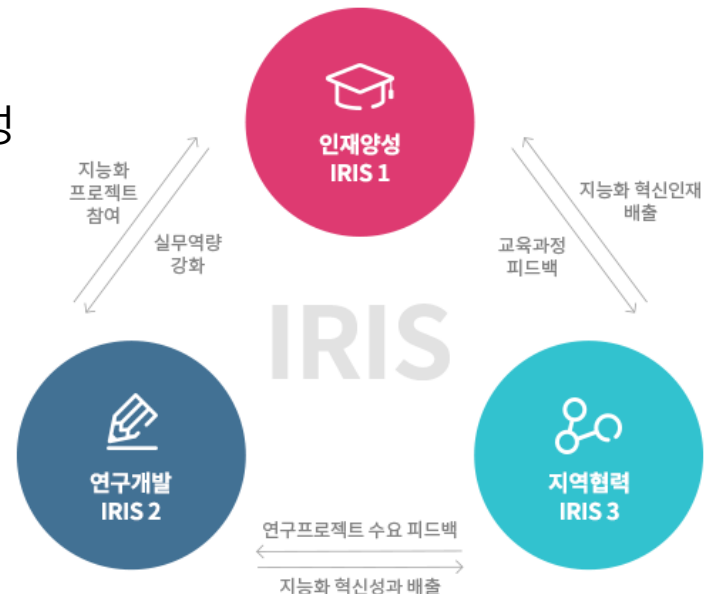
2. 충북대학교 산업인공지능연구센터 소개

- 2020년 과학기술정보통신부 정보통신기획평가원의 Grand ICT연구센터로 선정 및 설립
- 주요 역할

4차 산업혁명을 선도하는 지능화 기술 분야 인재양성
산업현장의 애로기술 지능화 공동연구개발

지역협력 프로그램을 통한 지능화 혁신문화 확산 등

- 홈페이지 <https://indai.cbnu.ac.kr>



출제문제 설명

공사장의 안전모(safety helmet) 착용여부 검출

• 출제 배경

- 2020년 산업재해 사망자(882명) 중 **건설현장** 사고사망자가 51.5% 차지
- 건설현장의 69.1%가 가장 기본적인 보호장비인 **안전모**도 착용하지 않는 실정
- 부실한 안전관리를 강화하기 위한 시스템이 요구됨



출처: 기술인TV 건설안전 교육자료 (1안전모 미착용으로 인한 사고)

출제문제 설명

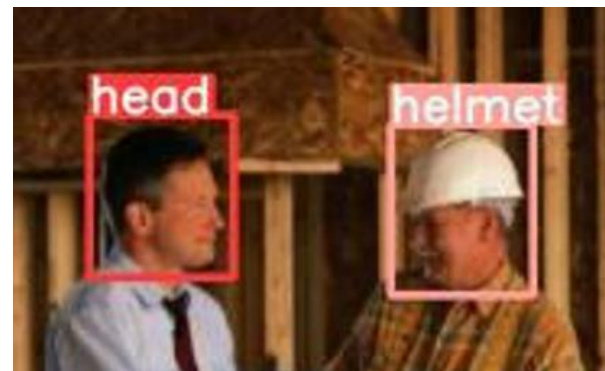
공사장의 안전모(safety helmet) 착용여부 검출

• 정의

- CCTV 등에 의한 **공사장** 영상 속에서 작업자의 **안전모 착용상태**에 대해 Object detection을 수행하여 Class와 Bounding box 정보를 도출

• 데이터셋

- 훈련데이터 : 공사장 영상 5,000개 기본 제공
- 성능향상을 위한 **데이터 수정/추가는 자유**
- Class 종류 : head, helmet
- 예측(출력)값 : Class와 Confidence, Bounding box 좌표



• 평가지표

- mAP@IoU=0.5

- mAP (Mean Average Precision) = $\frac{1}{n} \sum_{i=1}^n AP_i$

- IoU (Intersection Over Union) : 객체 검출 여부 판별을 위한 threshold 0.5 설정

출제문제 설명

공사장의 안전모(safety helmet) 착용여부 검출

- 클래스의 정의
 - head : 안전모를 착용하지 않은 경우
 - helmet : 안전모를 '머리에 착용'한 경우
- 안전모로 인정되는 것 : 공사장 안전모와 머리 보호용 헬멧류



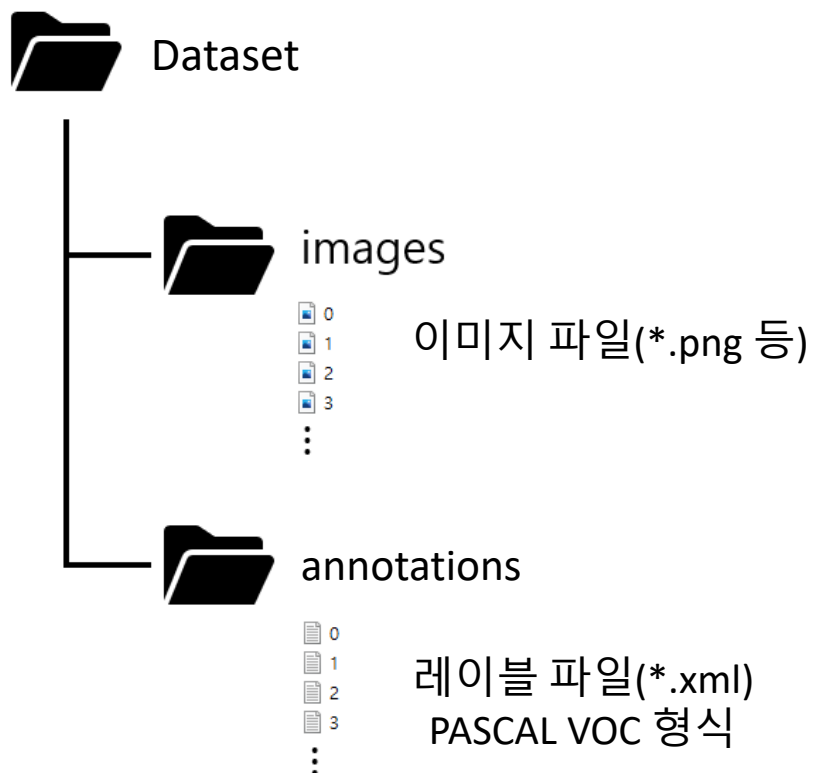
- 안전모로 인정되지 않는 것 : 모자, 선캡, 두건, 수영모 등



출제문제 설명

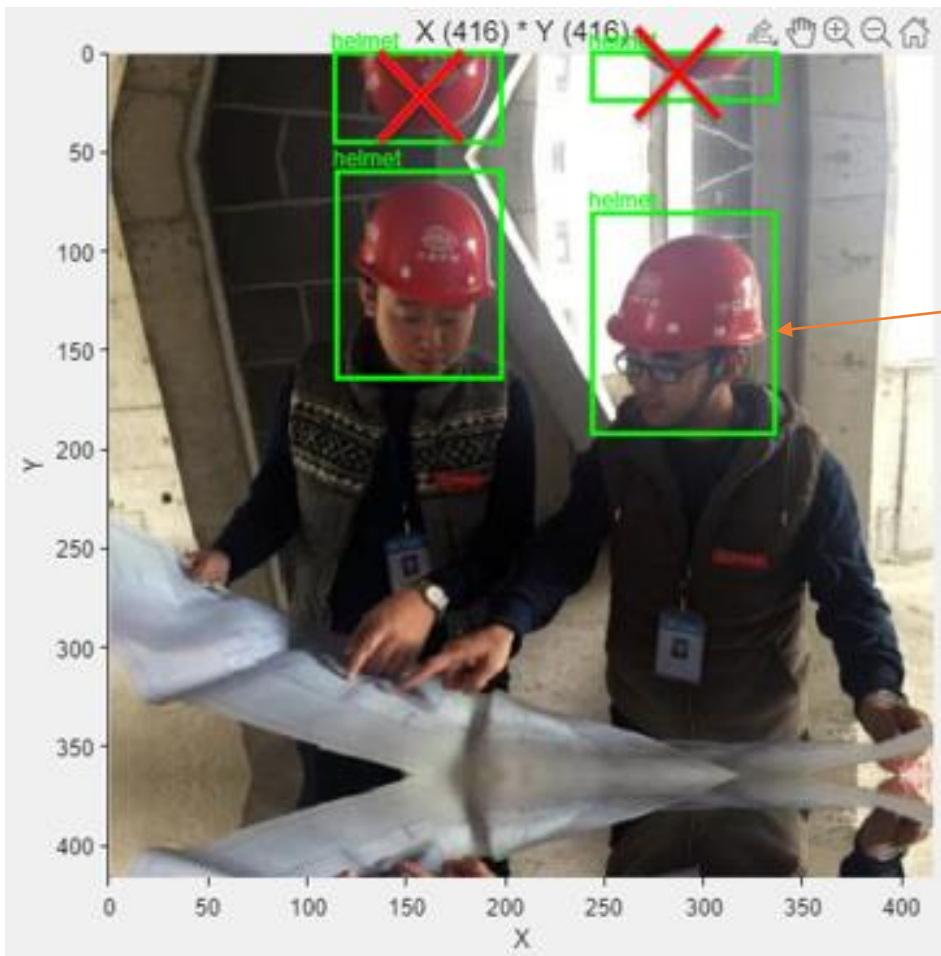
공사장의 안전모(safety helmet) 착용여부 검출

- 데이터셋의 구성



출제문제 설명

- 훈련데이터셋의 이미지-레이블 예
 - 문제 취지와 다른 레이블링(2개) 포함



Hard_hat_workers1002.png

```
<annotation>
  <folder>images</folder>
  <filename>hard_hat_workers1002.png</filename>
  <size>
    <width>416</width>
    <height>416</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>helmet</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <occluded>0</occluded>
    <difficult>0</difficult>
    <bndbox>
      <xmin>244</xmin>
      <ymin>81</ymin>
      <xmax>336</xmax>
      <ymax>192</ymax>
    </bndbox>
  </object>
  <object>
    ...
  </object>
  <object>
    ...
  </object>
  <object>
    ...
  </object>
</annotation>
```

Hard_hat_workers1002.xml

출제문제 설명

공사장의 안전모(safety helmet) 착용여부 검출

• 검출 오류 사례 (YOLOv5s로 학습시)



착용하지 않은 안전모 검출



스케일 오류

- 큰 이미지의 헬멧은 검출 못함
- 10x10px 이하는 평가시 제외



오검출
- 유사색상 등

출제문제 설명



(a) 일반 모자를 잘 구분 [TP]



(b) 가려져서 못 찾은 경우 [FP]



(c) 작아서 못 찾은 경우 [FP]



(d) 모자를 헬멧으로 오인한 경우 [FN]

출제문제 설명

공사장의 안전모(safety helmet) 착용여부 검출

- 검출 오류 사례 (YOLOv5s로 학습시)

머리상태(대머리, 두건)에 따른 오류

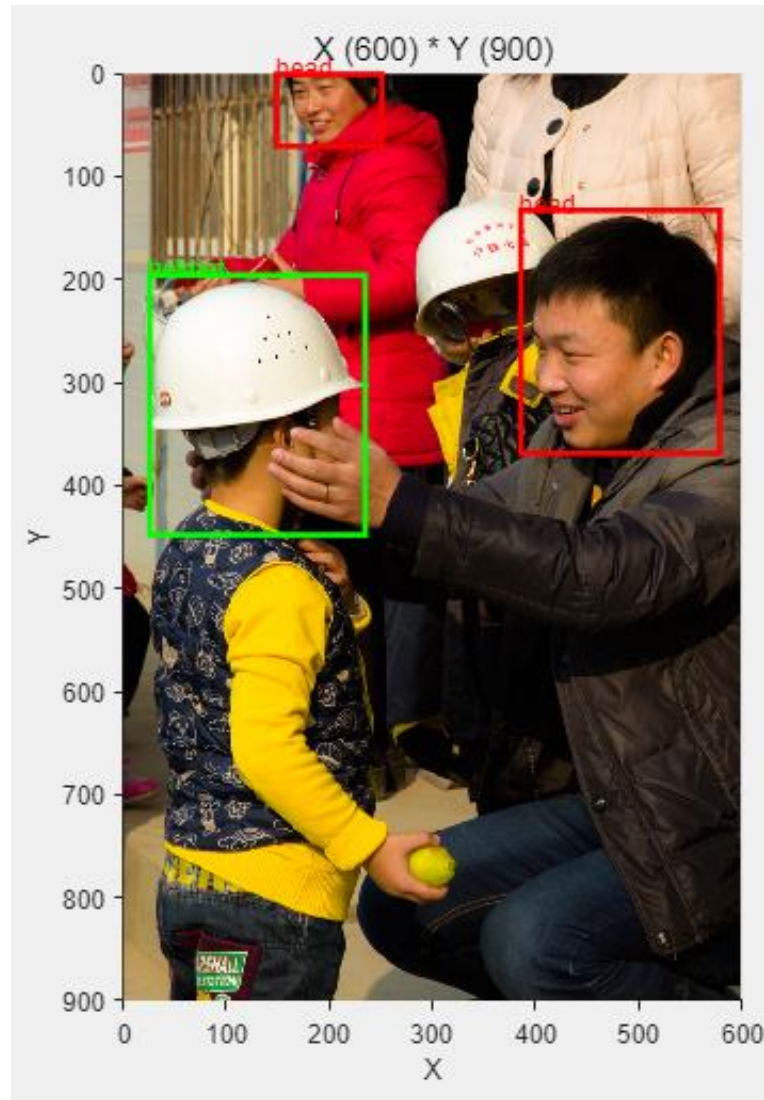


- 테스트셋 준비

- 총 데이터수 : 1,000개
- 이상의 검출 오류 사례로 15% 정도 구성할 예정

출제문제 설명

- 테스트 데이터셋의 이미지-레이블 예



결과 제출방법

- 개발과정에 대한 코드 및 설명자료 제출방법
 - 모델을 학습시키기 위한 **학습코드**
 - 별도의 수정없이 예측값을 재현할 수 있는 **추론코드**
 - **모델(가중치) 파일**
 - 개발 전반에 걸친 **설명자료**(ppt)
 - 코드는 주석을 넣어 가독성 있게 정리하고, 사용한 pretrained model, 추가 dataset 명시할 것

- 테스트 데이터셋에 대한 **예측값** 제출방법
 - 각 이미지와 동일한 이름의 텍스트 파일로 저장 (예, abc.png → abc.txt)
 - 텍스트(.txt)파일의 예측값은 다음 형식으로 작성

<class name> <confidence(0~1)> <xmin(px)> <ymin(px)> <xmax(px)> <ymax(px)>

예측값 파일의 예 (**모든 confidence의 결과를 포함할 것**)

```
head 0.460851 429 219 528 247
helmet 0.287150 336 231 376 305
helmet 0.292345 0 199 88 436
head 0.269833 433 260 506 336
```



- 산출물은 indaidept@cbnu.ac.kr로 압축하여 제출

평가지표 및 심사방법

- 평가지표 : $mAP@IoU=0.5$
 - Object detection에서 주로 사용하는 평가방법([PASCAL VOC 2012 competition](#))
 - $IoU > 0.5$ 인 경우에만 검출되었다고 판단
 - 계산방법 참조 : <https://github.com/Cartucho/mAP>
- 심사방법
 - 심사위원장 1명 + 심사위원 2명
 - 황영배 (충북대학교 지능로봇공학과 교수)
 - 김현용 (충북대학교 산업인공지능연구센터 초빙교수)
 - 김현호 (충북대학교 산업인공지능연구센터 초빙교수)
 - 평가지표를 계산한 후 순위권 팀에 대해 **사후검증(예측값 재현여부)** 실시
 - 입상팀은 시상식(11.2) 때 모델 개발에 대한 **브리핑(5~10분)** 실시 예정

대회 일정

행사	일시	세부 내용
접수	9.1(수) ~ 10.4(월)	홈페이지 접수
참가팀 선정 및 통보	-	팀별 개별 통보 (이메일, 문자)
설명회 및 온라인 경진대회	10.5(화) 10:00~11:00	진행방식, 규정, 서버 사용법 등 설명
	10.5(화) ~ 10.15(금)	훈련 데이터 및 문제 공개
산출물 제출	10.18(월) 10:00	테스트 데이터 공개 (1시간 이내 결과 제출)
	10.18(월) 11:00	소스 코드, 모델, 결과, 설명자료
사후 검증 및 평가	10.18(월) ~ 10.22(금)	수상자에 한해 재현성 및 부정행위 검증
결과 발표	10.27(수) 16:00	홈페이지 게시
결과 보고 및 시상식	11.2(화) (시간 추후공지)	결과 보고, 심사평, 시상 및 수상작 발표

문의처 : 043-249-1257, indaidept@cbnu.ac.kr

참가자 현황

- 참가현황 : 9개 기관 소속, 16개 팀, 38명 참가

접수번호	참가팀	소속(인원)	비 고
1	달복	충북대(1), 카톨릭대(1)	
2	바라쿠다	카이스트(1)	
3	Tensor	충북대(3)	금상
4	고양이키우고싶다	충북대(1)	
5	팀다엘	무소속(4)	
6	GOGO	충남대(1)	
7	MSIS	충북대(4)	동상
8	MSISLAB	충북대(3)	은상
9	투피스	충북대(4)	은상
10	JH	충북대(1)	
11	퍼스트	충북대(2)	
12	LUCID	새롬고(1)	
13	쌍둥	충북대(2)	
14	LAB423	충북대(3)	동상
15	WeMake	고려대(1), 경기대(1)	
16	선일	(주)선일(4)	

수상팀 및 특징

순위	팀명	정확도 (mAP@0.5)	개발 방법	
			데이터	모델
금	Tensor	80.98%	추가 ¹⁾ , 증량 ²⁾	YOLOv5l
은	MSISLAB	80.29%	증량	YOLOv4tiny
	투피스	79.67%	재라벨링 ³⁾ , 추가, 증량	YOLOv5x
동	MSIS	76.13%	재라벨링	YOLOv5l
	LAB423	76.04%	추가, 증량	YOLOv5x + VGG

1) 추가 : 대회에서 제공한 데이터 외의 데이터를 추가한 경우

2) 증량 : 데이터에 Augmentation 기법을 적용하여 데이터 수를 증가시킨 경우

3) 재라벨링 : 대회에서 제공한 데이터의 라벨링을 대회 취지에 맞게 수정한 경우

Q & A