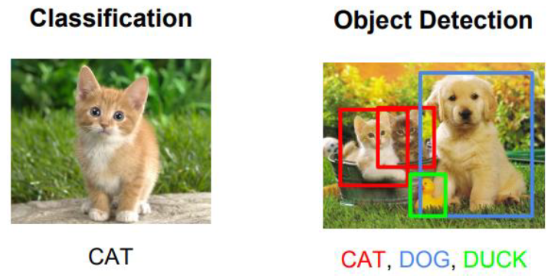
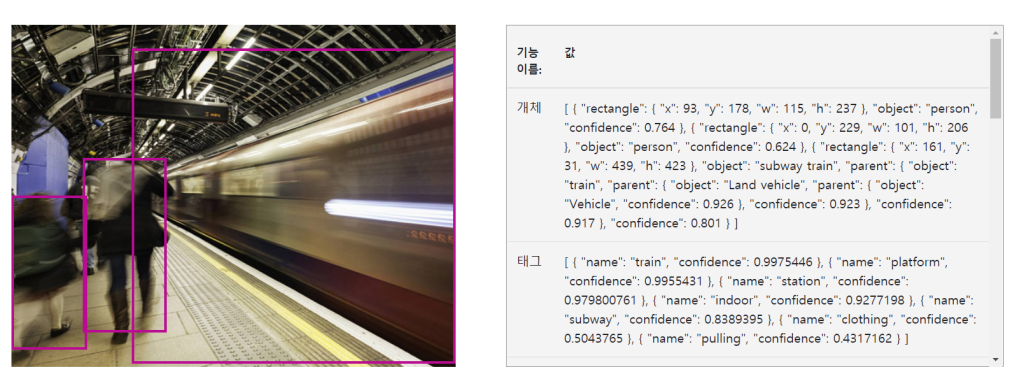
**영상분석기술 및 시장조사 CJ 제일제당 디지털추진부 BIO semester-ING 인턴 정아연**

* Object Detection = Multi-labeled image classification + bounding box regression task
* Localization : 이미지당 하나의 object에 대해 bounding box를 얻는 문제
* Classification : 개인지 고양이인지 분류하는 문제



**<MS Azure API를 이용한 영상분석>**

* 이미지 분류 : 총 86개의 범주 개념으로 이미지를 분류합니다.
* 이미지에 콘텐츠 태그 적용



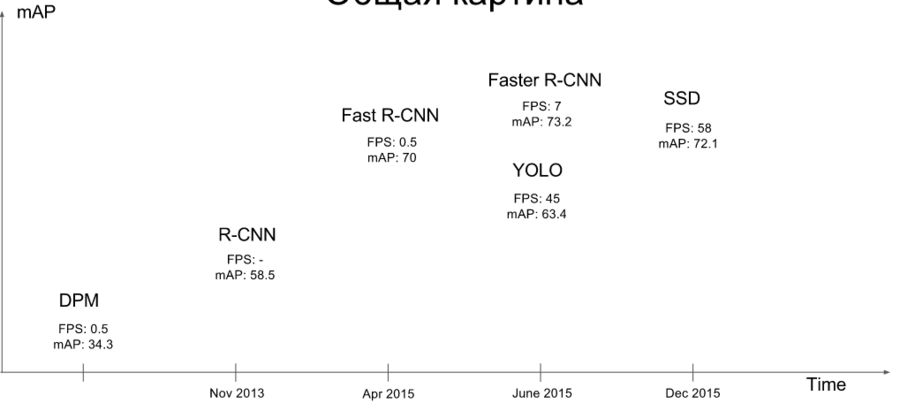
**Azure Media Service Video Indexer :** Azure 미디어 분석, cognitive services API를 기반하여 구축된 클라우드 애플리케이션입니다. 이 애플리케이션에서 Video Indexer 비디오 및 오디오 모델을 사용하여 비디오의 인사이트를 추출할 수 있습니다.

**Object Detection 딥러닝 기술 흐름도**

Object detection을 위한 딥러닝 모델은 일반적으로 다음과 같은 파이프라인을 갖는다.

1. Region Proposal Generation
2. Proposal Feature Extraction
3. Proposal Classification

R-CNN (2013) 🡪 OverFeat(2014) 🡪 Fast R-CNN(2015) 🡪 Faster R-CNN(2015) 🡪 YOLO v1(2016) 🡪 SSD(2016) 🡪 YOLO v2(2017) 🡪 Mask R-CNN(2017) 🡪 YOLO v3(2018) 🡪 M2Det(2019)

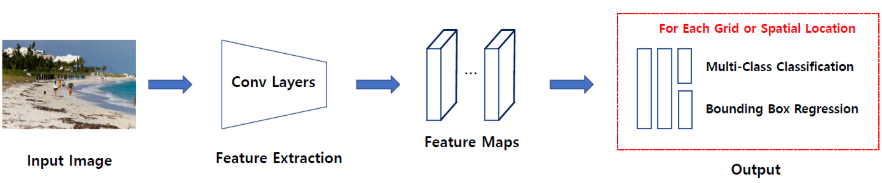


\*mAP : mean average precision (얼마나 잘 검출했는지, 얼마나 정확한지의 평균값)

Deep Learning을 이용한 Object detection은 2가지 방향으로 나눌 수 있다.

* **1- Stage Object Detector (YOLO, SSD)**

탐색영역을 찾는 Region Proposal과 해당 영역을 분류하는 Detection이 한번에 진행되며 빠르지만 정확도가 약간 떨어진다.



* **2- Stage Object Detector (R-CNN계열)**

Region Proposal과 Detection 두 가지 과정이 순차적으로 진행되며 1-Stage Object Detector에 비해 비교적 느리지만 정확하다



**R-CNN** : Selective Search 와 CNN을 결합한 방법론

\*Selective Search : 입력 이미지의 후보영역 ROI(interest of region)을 그려주는 superpixel기반의 알고리즘

대략 R-CNN은 약 1초에 0.2 프레임, Fast-R-CNN은 0.5프레임, Faster R-CNN은 7프레임을 최대로 처리하였으나 YOLO가 등장하여 45프레임을 보여주었고 버전이 올라가면서 155프레임을 기록하고 있다.

**YOLO(You Only Look Once)** : window 별로 sliding하면서 Convnet에 적용하는 것이 아니라 input 값을 한번에 Convnet에 적용하고 출력 값을 grid로 나누어 한번에 object에 classification과 location을 진행한다.

YOLO v1 예시 🡪 <https://www.youtube.com/watch?v=EJy0EI3hfSg>



**Mask R-CNN** : 기존의 Object detection을 수행하는 것을 발전시켜 pixcel-level segmentation으로 확장시켜 보다 정교한 localization이 가능하게 하였다. ROIPool기법에서 bilinear interpolation을 적용한 ROIAlign방법을 사용하였다.



Mask R-CNN 예시 🡪 <https://www.youtube.com/watch?v=OOT3UIXZztE>

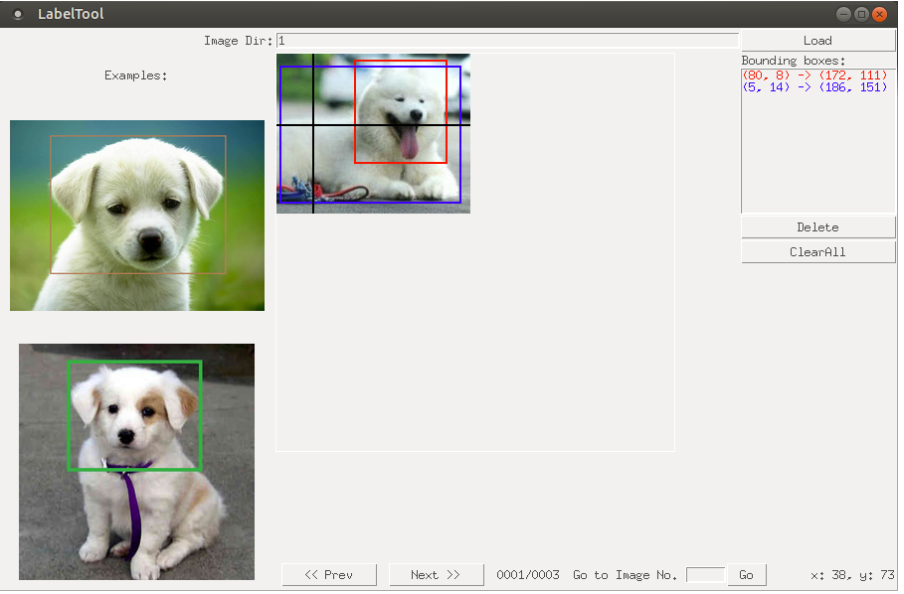
**Object Detection의 Labeling이란?**

PASCAL VOC, COCO, lmageNet등의 유명 데이터 셋은 이미지마다 bounding box annotation이 존재하여 이러한 데이터 셋을 사용하는 경우에는 별도의 Labeling 작업이 필요 없습니다. 그러나 이미지를 직접 취득하는 경우 각 이미지마다 bounding box annotation을 제작해야 합니다. Bounding box정보는 5가지입니다.

* Class index = category index = class name
* Bounding box의 좌표 = (x\_min, x\_max, y\_min, y\_max) or (x\_center, y\_center, width, height)



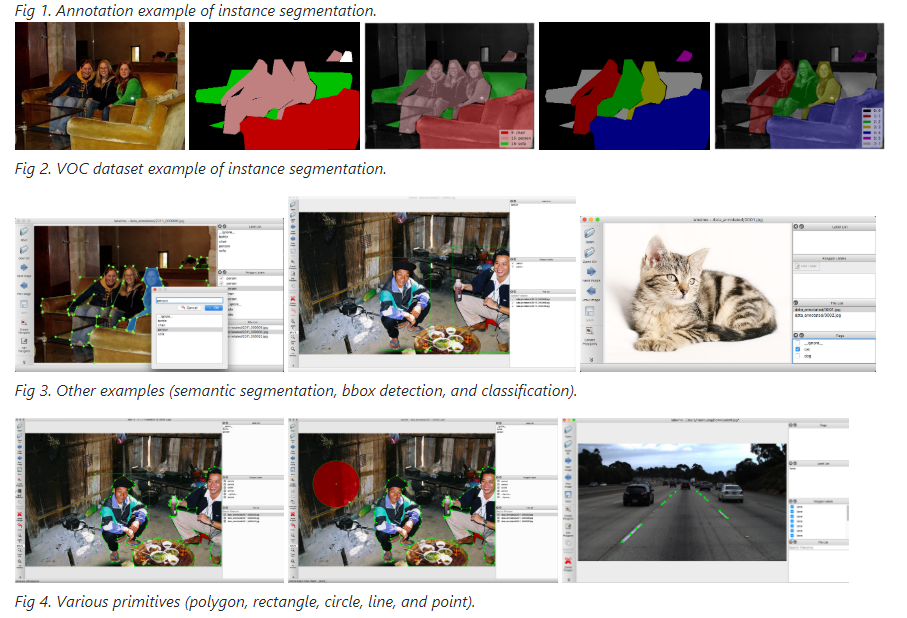
이렇게 이미지의 bounding box마다 5가지 정보를 생성하는 과정을 ‘annotation을 제작한다’ 혹은 ‘labeling한다’ 라고 표현하며 이 과정을 도와주는 **BBox-Label-Tool**이라는 프로그램이 있습니다.



**LabelMe**

<https://github.com/wkentaro/labelme>

이 프로그램은 BBox-Label-Tool과 다르게 사각형모양 뿐만 아니라 선, 점, 원 등 다양한 형태의 도형을 labeling할 수 있으며 classification, segmentation 등 다양한 작업의 labeling도 가능하다.



이외에 **Labelbox**, **RectLabel** 등의 프로그램이 있습니다.

<대표적인 Object Detection 적용사례>

* 자율주행 자동차
* CCTV Surveillance
* 지하철 같은 곳에서 갑자기 사람이 쓰러지거나 싸움이 일어나는 등 이상현상 감지
* 백화점에서 재방문 고객 감지
* 화재발생 감지
* OCR(Optical Character Recognition)
* 체크카드, 신용카드 등 카드번호 추출
* 차량 번호판 인식
* Aerial Image 분석
* 상공에서의 지역 영상에서 차량, 집, 나무 등 검출
* 주차장 차량 수 감지
* 제조업
* 제품의 결함 검사
* 스포츠 경기 분석
* 선수들 위치를 통해 경기 내용 분석 및 전술 활용
* 무인 점포