



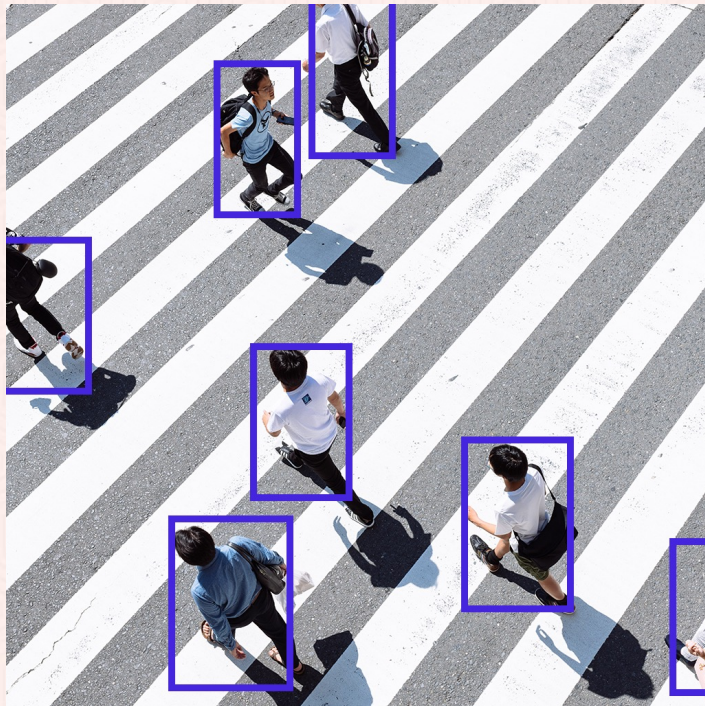
Mask RCNN Fine-Tuning을 이용한 Instance Segmentation

오늘의 학습내용

- Instance Segmentation
- Mask R-CNN
- Mask R-CNN Fine-Tuning
- PyTorch를 이용한 Mask R-CNN Fine-Tuning 구현

▲ 컴퓨터 비전 문제 영역 – Object Detection

❖ **Object Detection** : 물체가 있는 영역의 위치정보를 Bounding Box로 찾고 Bounding Box 내에 존재하는 물체의 레이블(Label)을 분류하는 문제 영역



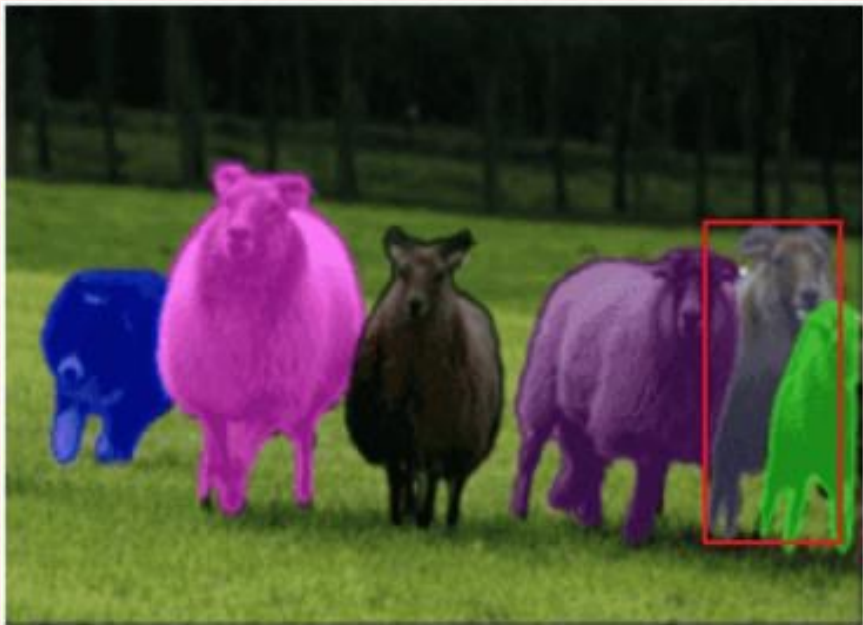
◆ 컴퓨터 비전 문제 영역 – Semantic Image Segmentation

❖ **Semantic Image Segmentation** : 이미지의 전체 픽셀에 대한 분류를 수행하는 문제 영역



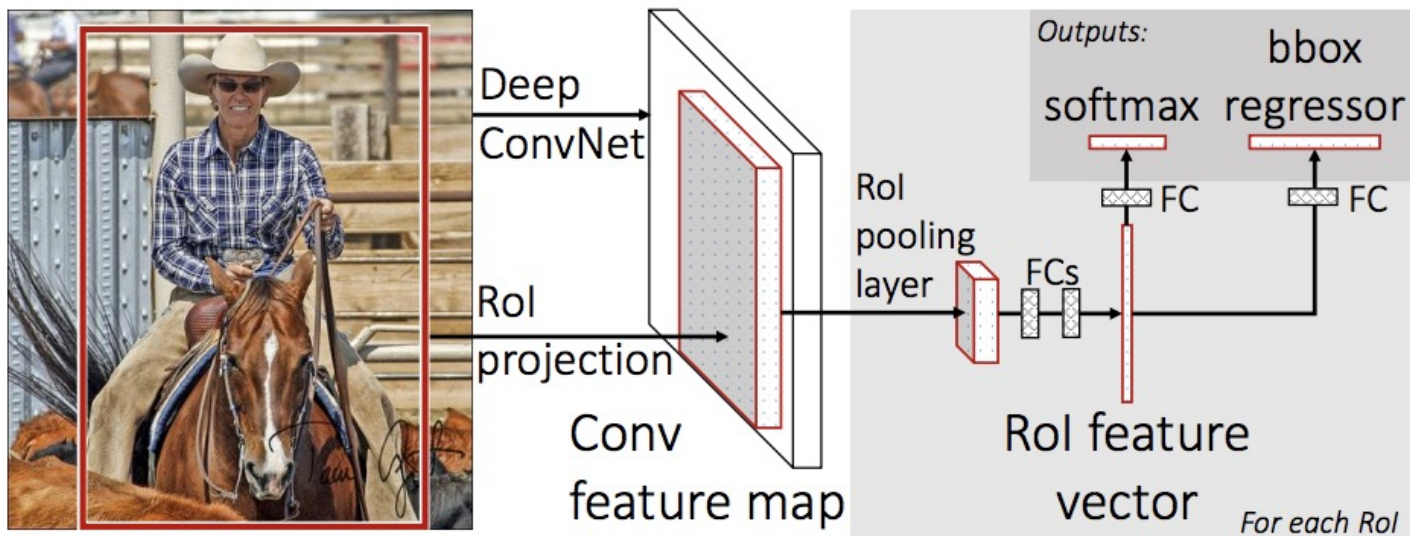
◆ 컴퓨터 비전 문제 영역 – Instance Segmentation

❖ **Instance Segmentation** : Object Detection과 Semantic Image Segmentation을 결합한 문제 영역



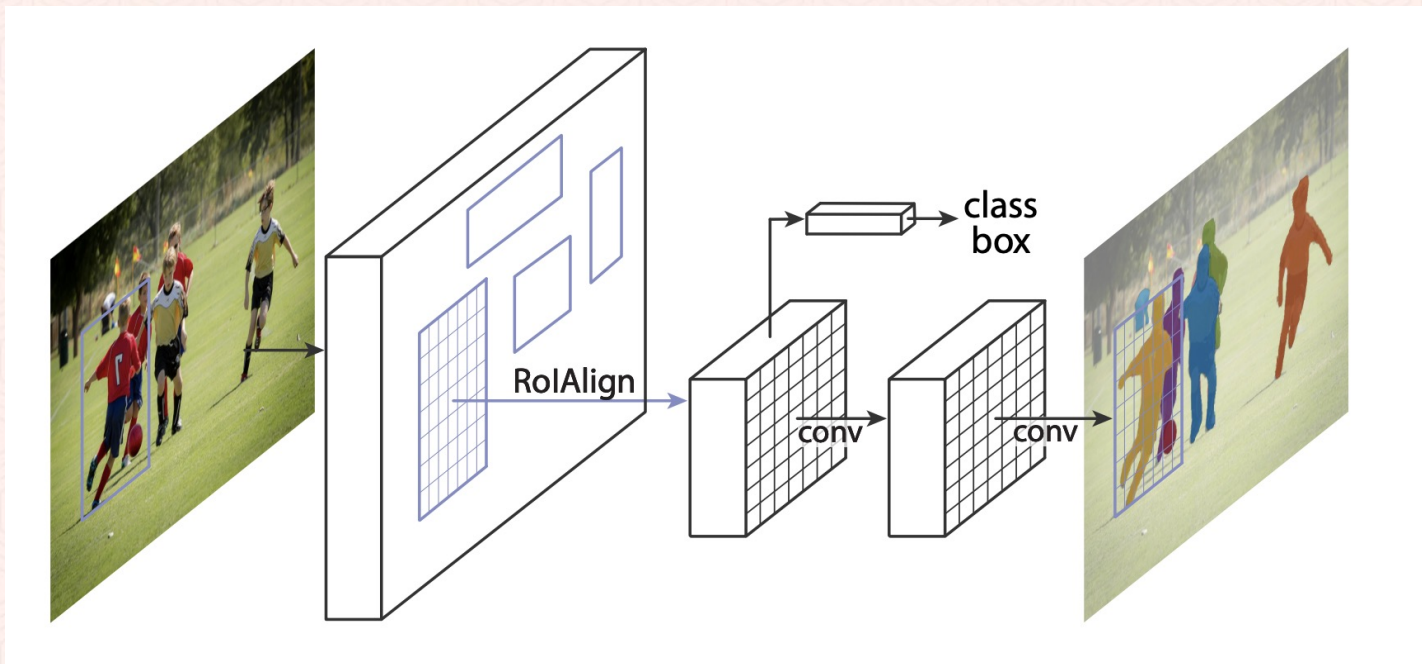
Fast R-CNN

- ❖ Mask R-CNN은 Fast R-CNN 구조를 Instance Segmentation을 위해 수정한 모델입니다.



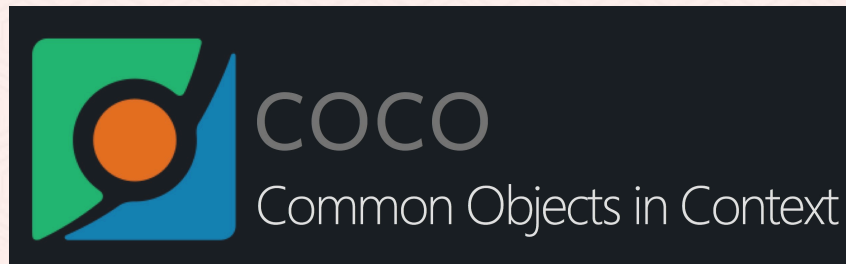
Mask R-CNN

- ❖ Mask R-CNN 구조를 이용하면 이미지에 대한 Instance Segmentation을 수행할 수 있습니다.

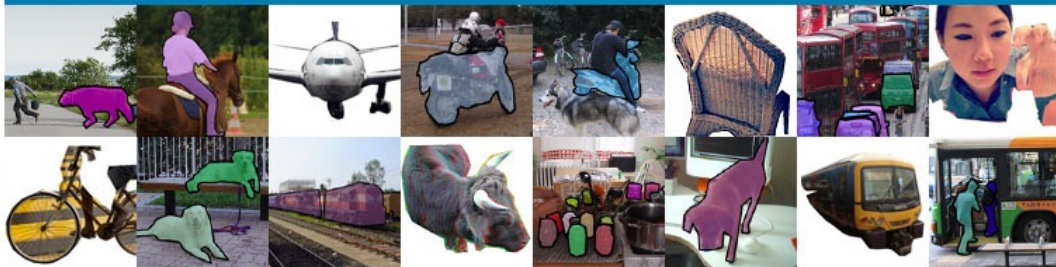


◆ Open Datasets – MS-COCO

- ❖ MS COCO Dataset은 MicroSoft사에서 구축한 Object Detection 용 데이터셋입니다.
- ❖ <https://cocodataset.org/#home>

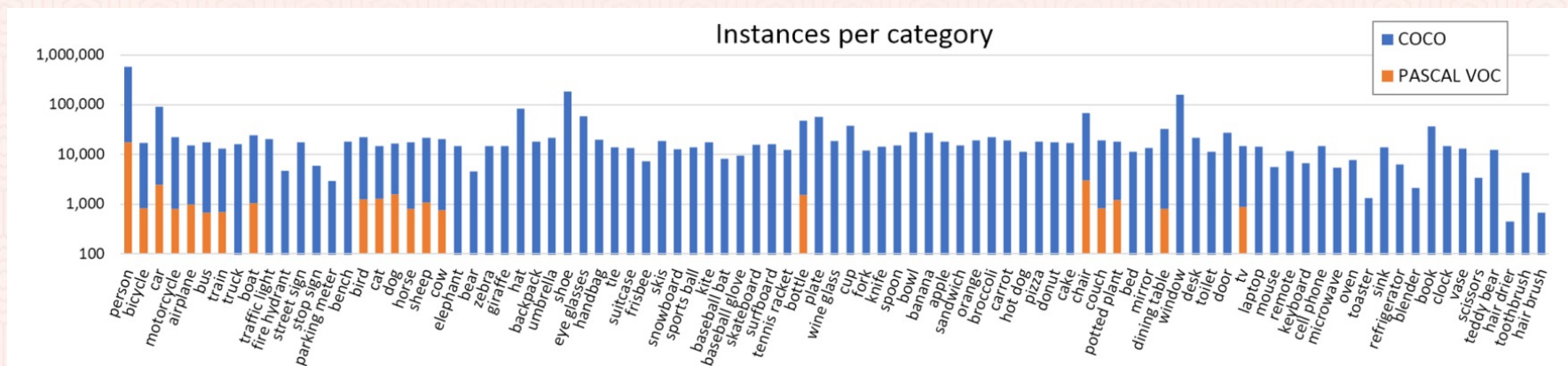


Dataset examples



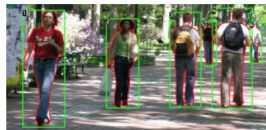
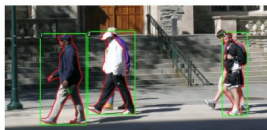
◆ Open Datasets – MS-COCO

- ❖ MS COCO 데이터셋은 80,000장의 training image, 40,000장의 validation image, 20,000장의 test image로 구성되어 있고 80개의 label로 구성되어 있습니다.
- ❖ <https://gist.github.com/AruniRC/7b3dadd004da04c80198557db5da4bda>



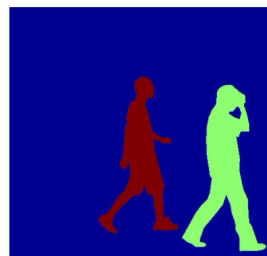
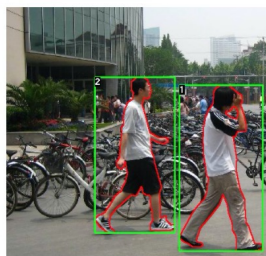
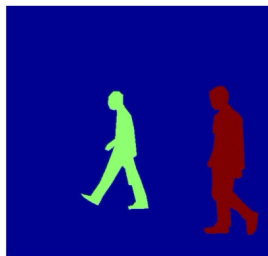
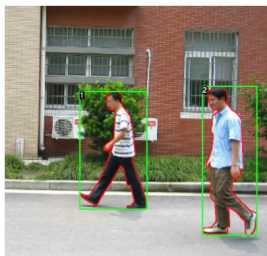
◆ Penn-Fudan Database for Pedestrian Detection and Segmentation 데이터셋

- ❖ Penn-Fudan Database for Pedestrian Detection and Segmentation 데이터셋은 보행자에 대한 Semantic Image Segmentation 레이블을 포함하고 있는 데이터셋입니다.



PennPed00015

PennPed00071

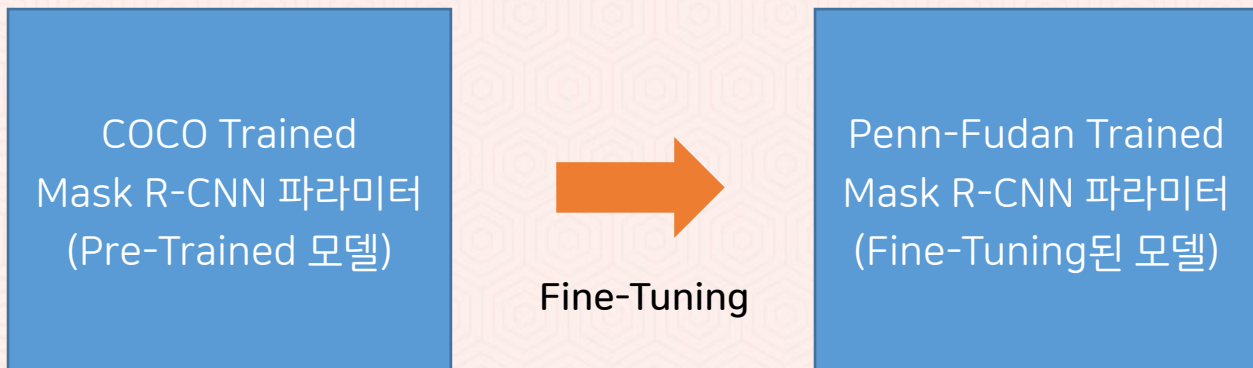


FudanPed00001;

FudanPed00009

Mask R-CNN Fine-Tuning

- ❖ Mask R-CNN의 파라미터를 COCO 데이터셋에서 Penn-Fudan 데이터셋에 맞게 파라미터를 Fine-Tuning합니다.



Metric – Precision, Recall, F1

- ❖ **Precision** : 정밀도(Precision)는 검색된 결과들 중 관련 있는 것으로 분류된 결과물의 비율
- ❖ **Recall** : 재현율(Recall)은 관련 있는 것으로 분류된 항목들 중 실제 검색된 항목들의 비율
- ❖ **F1** : Precision과 Recall의 조화 평균. Precision과 Recall을 한번에 비교할 수 있습니다.

TP = True positive

TN = True negative

FP = False positive

FN = False negative

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

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

▶ 파이토치(PyTorch)를 이용해서 Mask R-CNN Fine-Tuning하기

- ❖ 파이토치(PyTorch)를 이용해서 Mask R-CNN Fine-Tuning하기
- ❖ 10강_pytorch를_이용한_mask_rcnn_finetuning.ipynb
- ❖ https://colab.research.google.com/drive/1SUJ530Pj_UdcYzUsEMoxpSGGmxMAxUYt?usp=sharing