

CNN Tutorial

Object Detection

목차

1.What is Detection ?

2.프로젝트 설명

3.데이터 설명

4.사용한 모델 설명

5.결과

6.Augmentation

7.Plan

1. What is Detection?

Classification이 한장의 이미지를 보고 어떤 라벨에 속하는지 찾는것이라면,

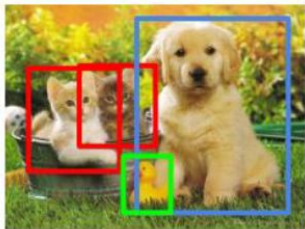
Detection은 이미지 내에서 라벨에 속하는 물체들을 모두 찾고,
그 물체들의 위치를 바운딩 박스를 통해 나타내주는 것이라고 할 수 있다.

Classification



CAT

Object Detection



CAT, DOG, DUCK

Object Detection

= Multi-labeled Classification + Bounding Box Regression

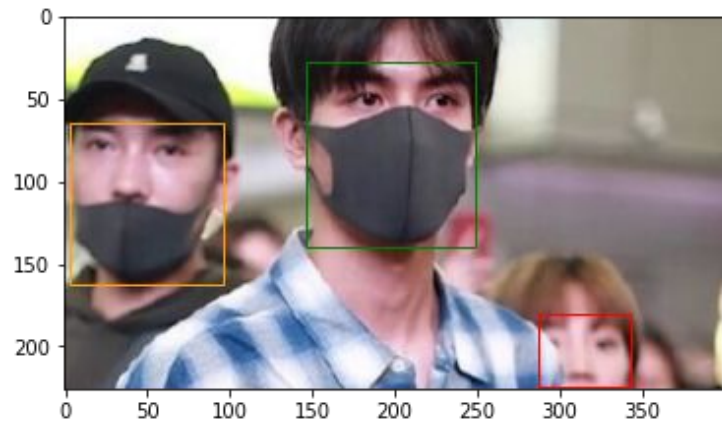
일반적으로 **object detection**은 자율 주행 자동차, CCTV, 스포츠 경기, 무인 점포 등에서 사용된다.

2. 프로젝트

Mask Detection

이미지에서 마스크를 착용한 사람을 Detection

- `with_mask`, `mask_wared_incorrect`, `without_mask`



3. 데이터

사용한 데이터셋: Face Mask Detection(출처: Kaggle)

classes: with_mask, mask_wared_incorrect, without_mask



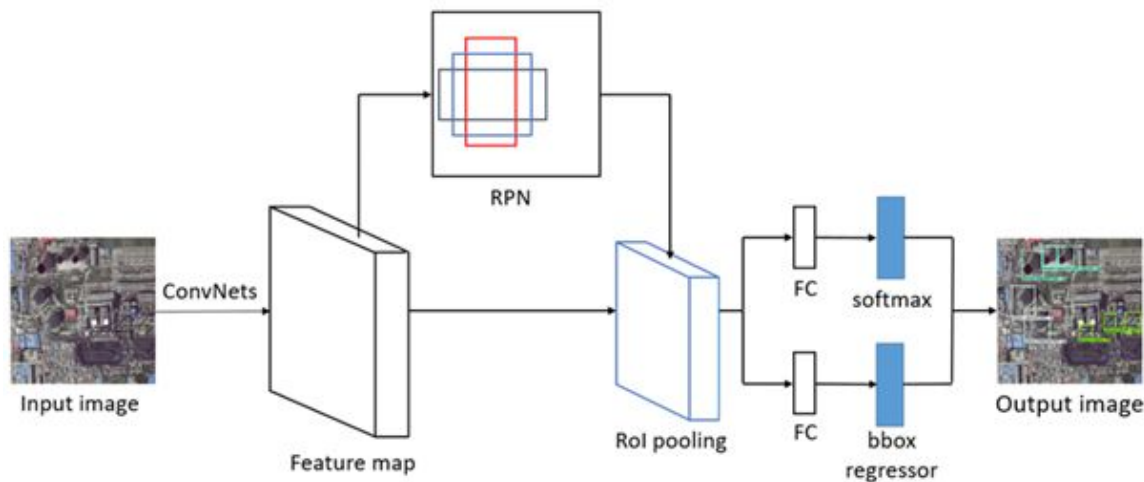
images/maksssskssss0.png

```
...  
<object>  
  <name>with mask</name>  
  <pose>Unspecified</pose>  
  <truncated>0</truncated>  
  <occluded>0</occluded>  
  <difficult>0</difficult>  
  <bndbox>  
    <xmin>185</xmin>  
    <ymin>100</ymin>  
    <xmax>226</xmax>  
    <ymax>144</ymax>  
  </bndbox>  
</object>  
...
```

annotations/maksssskssss0.xml
(일부분)

4. 사용한 모델

Faster R-CNN



RPN(Region Proposal Network)

- GPU를 사용하여 이미지의 후보 영역을 찾는다.

Anchor-box

- 미리 정해놓은 여러개의 비율과 크기의 **bounding box**를 사용

NMS(Non-Maximum Suppression)

- 동일한 **object**에 여러개의 **bounding box**가 있을때, 가장 **score** 높은 **box**만 남기고 나머지는 제거하는 것

4. 사용한 모델

Pretrained model

```
model = torchvision.models.detection.fasterrcnn_resnet50_fpn(pretrained=True)
in_features = model.roi_heads.box_predictor.cls_score.in_features
model.roi_heads.box_predictor = FastRCNNPredictor(in_features, 4)
```

optimizer: SGD

lr=0.005, momentum=0.9

batch size: 4

epochs: 100

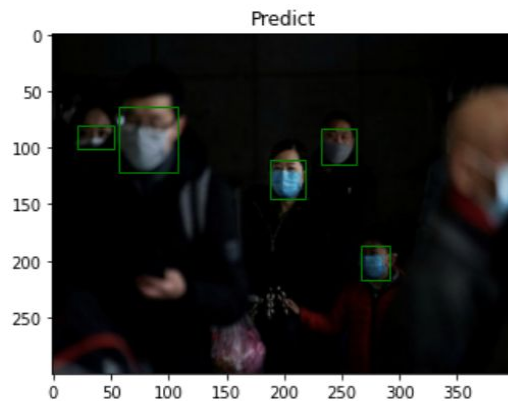
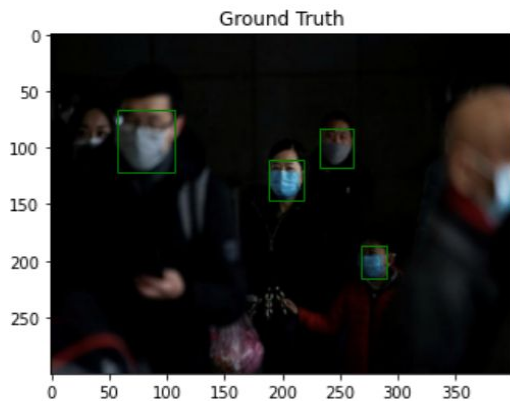
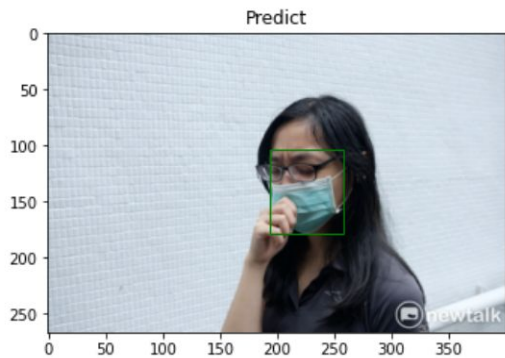
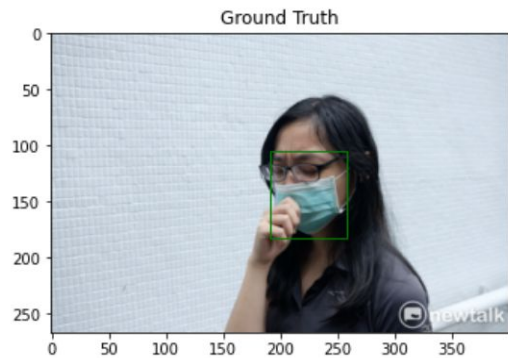
결과

epoch: 100/100, Loss: 11.68

mAP : 0.74

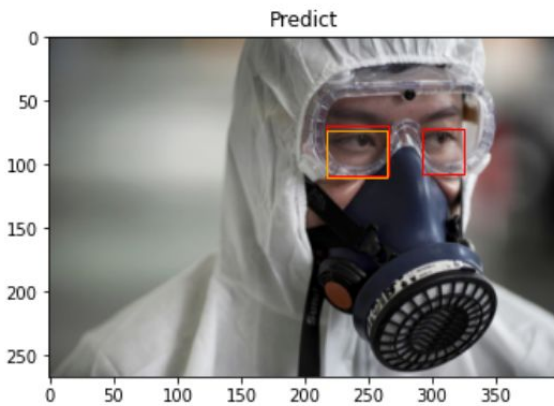
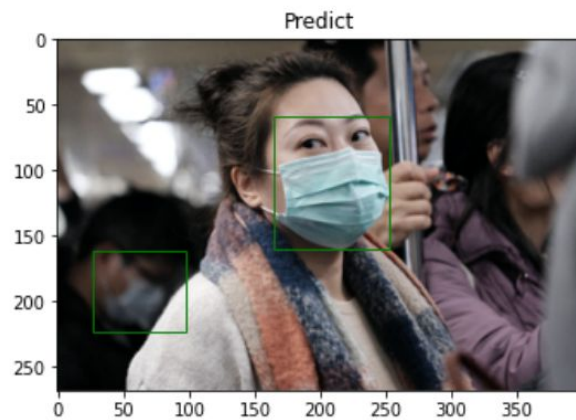
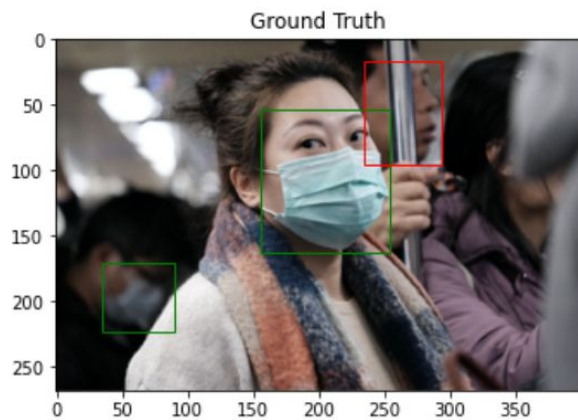
5. 결과

Best Predict

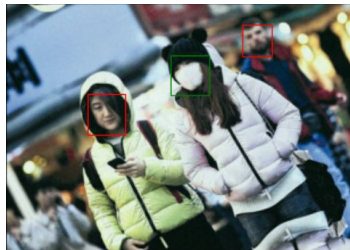


5. 결과

Worst Predict



6. Augmentation



albumentations 모듈

- 이미지 Augmentation을 지원해주는 라이브러리
- 이미지와 bounding box를 같이 변환 할 수 있다.

```
Augmented_transform = A.Compose(  
    [A.HorizontalFlip(p=1),  
      A.Rotate(p=1),  
      A.OneOf([  
          A.ChannelShuffle(p=1),  
          A.RGBShift(p=1)  
      ], p=1),  
      A.OneOf([  
          A.MotionBlur(p=1),  
          A.GaussNoise(p=1)  
      ], p=1),  
      A.pytorch.transforms.ToTensor()],  
    bbox_params=A.BboxParams(format='pascal_voc', label_fields=['labels']),  
)
```

7. Plan

- 미완성된 **Augmentation** 진행!!
- 상대적으로 속도가 더 빠른 **One-stage object detection** 추가 공부
- study 방향성 조정