

Pre-trained된 Mask-RCNN을 활용한

Watermark에 따른 이미지 내 객체 탐지 성능 비교 보고서

2021116864 김세희

1. 서론

Mask-RCNN을 선택하여 워터마크가 있는 이미지와 원본 이미지에서 객체 탐지 성능을 비교하였습니다.

Mask-RCNN을 선택한 주요 이유는 다음과 같습니다:

- 1) Segmentation 기능: Mask-RCNN은 객체 탐지뿐만 아니라 객체의 픽셀 단위 segmentation을 제공합니다. 이는 단순한 바운딩 박스 객체 탐지보다 더 정밀한 객체 탐지를 가능하게 합니다.
- 2) 강력한 성능: COCO 데이터셋에서 학습된 Mask-RCNN은 높은 성능을 보이며, 다양한 크기와 형태의 객체를 효과적으로 탐지합니다.
- 3) 실험 목적: 워터마크가 객체 탐지 성능에 미치는 영향을 분석하는 데 있어, 픽셀 단위의 정확한 세그멘테이션을 제공하는 Mask-RCNN이 더욱 적합하다고 판단하였습니다.

2. 실험 방법

2.1 데이터셋 준비

- 1) 이미지 선택: 동일한 원본 이미지와 watermark가 추가된 이미지를 입력하였습니다.
- 2) 전처리: 이미지는 OpenCV를 사용하여 BGR 형식으로 로드된 후, RGB 형식으로 변환했습니다. 변환된 이미지는 torchvision.transforms.functional.to_tensor를 사용하여 PyTorch 텐서로 변환하였습니다. 이 과정에서 이미지의 픽셀 값은 0과 1 사이로 정규화되었습니다.

2.2 모델 선택

- 1) Pre-trained된 Mask-RCNN 모델을 선택하였고, COCO 데이터셋으로 사전 학습되어 다양한 객체를 탐지하였습니다.
- 2) PyTorch 및 torchvision 라이브러리를 사용하여 모델을 불러왔습니다.

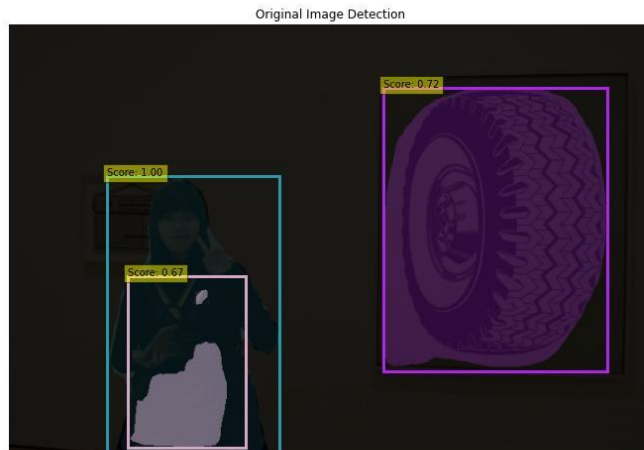
2.3 실험 과정

- 1) 이미지 로드: 원본 이미지와 watermark가 있는 이미지를 각각 load
- 2) 객체 탐지: 두 이미지에 대해 Mask-RCNN 모델을 사용하여 객체를 탐지
- 3) 결과 비교: 탐지된 객체의 신뢰 점수(confidence score)를 비교

3. 실험 결과

3.1 신뢰 점수(confidence score) 비교

1) 원본 이미지



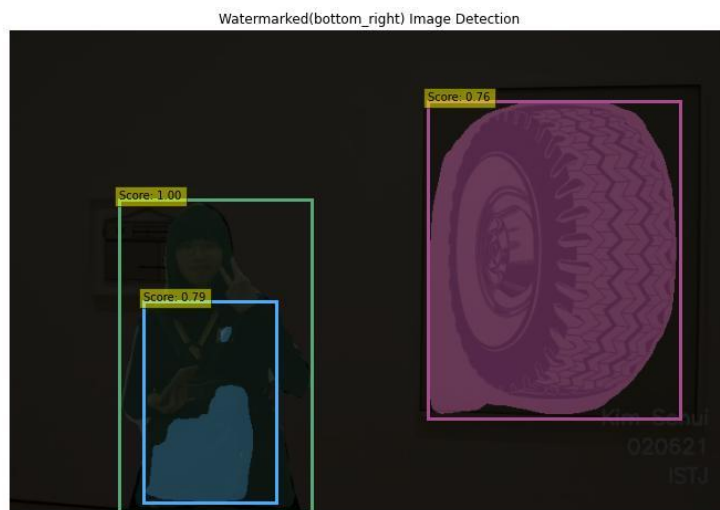
- i. 가방: 0.67
- ii. 사람: 1.00
- iii. (액자 속 그림) 바퀴: 0.72

2) 워터마크 이미지 (bottom-left)



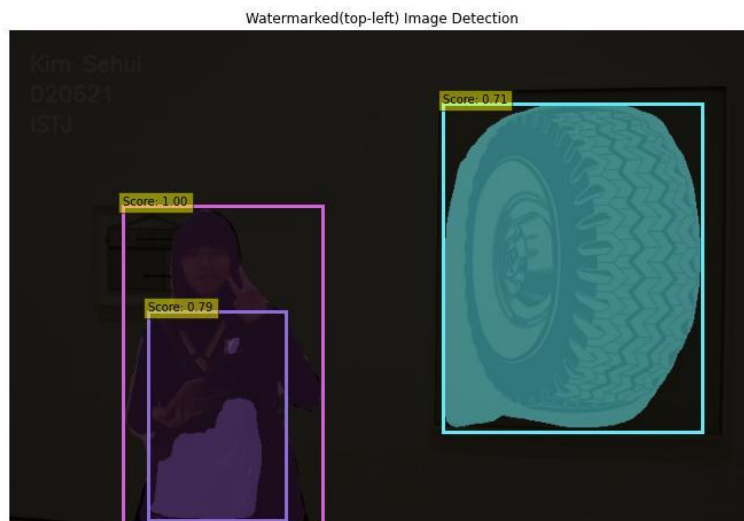
- i. 가방: 0.80
- ii. 사람: 1.00
- iii. (액자 속 그림) 바퀴: 0.71

3) 워터마크 이미지 (bottom-right)



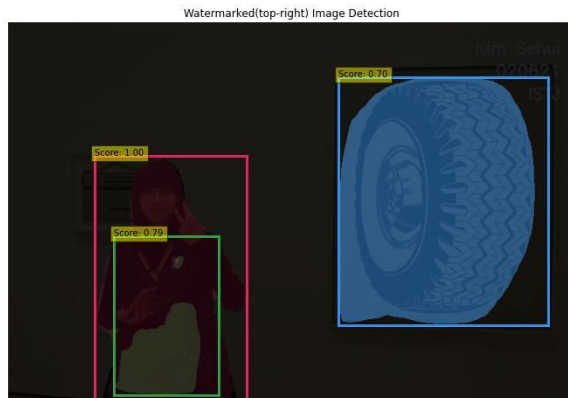
- i. 가방: 0.79
- ii. 사람: 1.00
- iii. (액자 속 그림) 바퀴: 0.76

4) 워터마크 이미지 (top-left)



- i. 가방: 0.79
- ii. 사람: 1.00
- iii. (액자 속 그림) 바퀴: 0.71

5) 워터마크 이미지 (top-right)



- i. 가방: 0.79
- ii. 사람: 1.00
- iii. (액자 속 그림) 바퀴: 0.70

4. 분석

1) 탐지 성능

원본 이미지와 비교할 때, watermark가 있는 이미지의 가방에 대한 신뢰 점수가 더 높게 나타났습니다. 이는 워터마크가 객체 탐지를 방해하지 않고, 오히려 탐지를 도와주는 역할을 하였다고 보여집니다.

2) 탐지 영향

watermark를 다양한 위치(top-left / top-right / bottom-left / bottom-right)에 추가하여도 객체와 겹치는 부분이 거의 없어서 객체 탐지를 방해하지 않았고, 이미지 내 다른 중요한 영역을 강조하여 탐지 성능이 향상된 것으로 보입니다. (색조 대비를 뚜렷하도록 watermark의 색상을 변경하여도 동일한 결과를 보임)

5. 결론

워터마크가 객체 탐지 성능에 미치는 영향은 위치와 형태에 따라 다를 수 있습니다. 이번 실험에서는 워터마크가 오히려 탐지 성능을 향상시키는 결과를 보였습니다. 다만, 사용된 이미지가 단순하였기 때문에 더 복잡한(객체가 다양한) 이미지를 사용한다면 결과가 다를 수 있습니다.

6. 참고 문헌

PyTorch Documentation

`torchvision.models.detection maskrcnn_resnet50_fpn`