|  |  |  |  |
| --- | --- | --- | --- |
| **실험날짜** | **2021.12.08~12.10** | **작성자** | **남준원** |
| **실험제목** | **YOLOv5 헬멧 착용 모델링 연구** | | |
| **실험개요** | **YOLOv5(pre-trained)모델을 사용하여 헬멧 착용여부 확인** | | |
| **Keyword** | **YOLOv5, Colab, Helmet Detection** | | |
| **-What is Yolo5-**  **YOLO : You Only Look Once, 객체 감지 딥러닝 모델**   * **이미지에서 객체를 식별하고 주변에 경계 상자를 그리는 작업**   **YOLOv5 : Yolo의 5번째(최신) 모델**    **-Why using Yolo5-**   1. **이미지 전체를 한번만 보는 것이다.**  * YOLO 이전의 R-CNN은 이미지를 여러 장으로 분할하고, CNN모델을 이용해 이미지를 분석했다. 그렇기 때문에 이미지 한장에서 Object Detection을 해도 실제로는 여러 장의 이미지를 분석하는 것과 같았다. 하지만 YOLO는 이러한 과정 없이 이미지를 한 번만 보는 강력한 특징을 가지고 있다. 이미지 전체를 한 번에 바라보는 방식을 이용하므로 class에 대한 맥락적 이해도가 다른 모델에 비해 높아 낮은 False-Positive를 보인다.  1. **통합된 모델을 사용하는 것이다.**  * 기존 Object Detection 모델은 다양한 전처리 모델과 인공 신경망을 결합해서 사용했다. 하지만 YOLO는 통합된 모델을 사용해 간단하다.  1. **실시간으로 객체를 탐지할 수 있는 것이다.**  * YOLO가 유명해진 이유는 높은 성능은 아니더라도 준수한 성능으로 실시간으로 Object Detection이 가능했기 때문이다. 기존의 Faster R-CNN보다 6배 빠른 성능을 보여준다.   **-Yolo5 사용 전략-**   1. **Pretraining data(coco dataset)에 추가적으로 학습**  * Coco dataset : 33만장 이미지, 150만 객체, 80개 카테고리로 구성 * 헬멧 쓴 이미지와 안 쓴 이미지를 추가하여 학습   **-Yolov5 사용법-**   1. Git clone을 통해 YOLOv5 github주소를 받기 2. Requirements.txt로 환경설정 3. Roboflow에서 만든 train/test/valid 파일들을 추가 4. Data.yaml을 통해 라벨들의 종류를 확인 5. Yolov5(s, m, l, x).yaml 중 원하는 모델을 선택하여 커스터마이징 6. Yolov5에 있는 train.py를 이용하여 학습  * Img:416, batch:16, epochs:100  1. mAP 0.5를 0.95이상 될 수 있도록 2. 어떠한 이미지를 학습했는지 확인 3. 학습시킨 가중치(weight)와 YOLOv5의 detect.py를 이용하여 학습이 잘 되었는지 test데이터로 확인  * Img:416, conf:변동  1. 학습이 잘 되었는지 확인 후 가중치를 구글드라이브에 저장 2. 따로 준비해둔 이미지로 한번 더 확인   **-학습 과정-**  **- 과정 1**   1. **사용법 3 수정 (데이터 셋)**  * 구글에서 헬멧착용 사진과 미착용 사진 각각 50개씩 다운로드 * CVAT을 사용하여 annotation (설명서 따로 있음) * Roboflow에서 train/test/valid(0.7,0.2,0.1)로 나누고 augmentation하여 300장으로 증강  1. **사용법 5 수정 (모델 선정)**  * 가장 가벼운 모델인 Yolov5s 사용  1. **사용법 7 확인 (mAP 확인)**      1. **사용법 9 수정 (Detecting 파라미터 설정)**  * Img:416, conf:0.4  1. **사용법 10 확인 (Detected 이미지 확인)**      1. **결론**  * mAP는 0.95이상이 나왔지만 confidence가 0.4로 낮음에도 불구하고 전혀 인식을 못하고 있음   **- 과정 2**   1. **사용법 3 수정 (데이터 셋)**  * 증강할 때 사진들이 겹쳐서 과대적합이라고 판단하여 Roboflow에서 수정  1. **사용법 5 수정 (모델 선정)**  * YOLOv5s가 간단한 모형이라고 판단하여 Yolov5m으로 수정      1. **사용법 7 확인 (mAP 확인)**      1. **사용법 9 수정 (Detecting 파라미터 설정)**  * img:416, conf:0.45  1. **사용법 10 확인 (Detected 이미지 확인)**      1. **결론**  * 과정 1보다는 인식을 잘하지만 헬멧 인식 기능이 여전히 떨어짐   **- 과정 3**   1. **사용법 3 수정 (데이터 셋)**  * 과정 1, 2에서 300장의 데이터가 부족하다고 판단하여 Kaggle에서 5000장 데이터를 가져와서 Roboflow 진행(train/test/valid – 0.7/0.2/0.1) * <https://www.kaggle.com/andrewmvd/hard-hat-detection>  1. **사용법 5 수정 (모델 선정)**  * 데이터 양이 많아 YOLOv5m 모델로 돌리기에 시간이 많이 걸린다고 판단하여 YOLOv5s 모델로 사용 * 실행결과 예측률이 떨어져서 시간이 걸리지만 Yolov5m 모델로 다시 변경  1. **사용법 7 확인 (mAP 확인)**      1. **사용법 9 수정** **(Detecting 파라미터 설정)**  * img:416, conf:0.5 * 예측률이 높아서 confidence를 0.5까지 올림  1. **사용법 10 확인** **(Detected 이미지 확인)**      1. **결론**  * 예측률이 가장 높음 * 이 모델을 사용하기로 결정   **-향후 개선방안**-   * 다양한 환경(어두운 밤, 흐린 날씨, 비오는 날 등)에서도 잘 인식될 수 있도록 데이터 추가 * 빠르게 예측을 할 수 있도록 모델을 가볍게 만들기 | | | |