

## 차량 흡집 전/후 심각도 판정 및 박스 그리기

---

### 전체 흐름

1. 탁송 건 선택
  2. 전/후 사진 업로드[ 업로드 할 때 사진에 '역할' 붙이기] - stage: "before" 또는 "after"- view: "front", "rear", "left", "right", "left\_side", "right\_side" 같은 각도/컷  
기사가 코드를 몰라도 앱 UI로 선택하게 만들면 됨.  
버튼: "전면 / 후면 / 좌측 / 우측"  
토글: "탁송 전 / 탁송 후"  
서버는 db에 "탁송 차량" 기준으로 업로드 목록을 저장  
-> drivers\_2026.xlsx에 있는 A1열 '번호' 기준으로 폴더 생성
  4. before/after가 "한 쌍"이 되면 자동으로 모델 실행
  5. 박스 이미지, 심각도 판정 점수 나오기
  6. 이 점수가 엑셀에 저장이 됨
- 

1. 심각도 판정 (파일 이름: severity.py)

#### -역할

탁송 전/후 이미지 쌍을 입력받아

차량 흡집의 변화 정도를 0~4 단계 심각도 점수로 판정

결과를 JSON + 시각화 이미지(히트맵/오버레이) 형태로 반환

- 이미지 저장 경로(서버 로컬)

예시 : uploads/{job\_id}/before\_{view}.jpg

uploads/{job\_id}/after\_{view}.jpg

#### -실행 조건

같은 job\_id + view에 대해

before 이미지와 after 이미지가 모두 존재할 때 실행

#### - 출력:

심각도 점수: 0 ~ 4

view별 결과 JSON

시각화 이미지 생성

results/{job\_id}/severity.json

results/{job\_id}/front\_heatmap.png

results/{job\_id}/front\_overlay.png

json 예시

```
{  
  "job_id": "26_2"  
  "view": "front",  
  "severity": 3,  
  "label": "심각",  
  "confidence": 0.87  
}
```

2. 전/후 이미지 차이를 기반으로 실제 새로 생긴 흠집만 박스로 추출하는 규칙 기반 탐지기(파일 이름: damage\_box)

- 역할

YOLO 등 객체 탐지 결과 + 전/후 이미지 차이 정보를 활용  
기존 흠집은 제거하고, 새로 발생한 흠집만 박스로 필터링  
severity 결과와 함께 웹 시각화용 박스 정보 생성

-처리 로직 (개념)

after 이미지에서 YOLO로 흠집 후보 탐지  
before/after 차이(픽셀 변화, heatmap) 계산  
변화가 없는 영역의 박스 제거  
변화가 있는 영역만 "신규 흠집"으로 판정  
severity 점수와 결합

3. watcher.ipynb

기사가 사진을 업로드 하면 자동실행이 되게 하는 코드