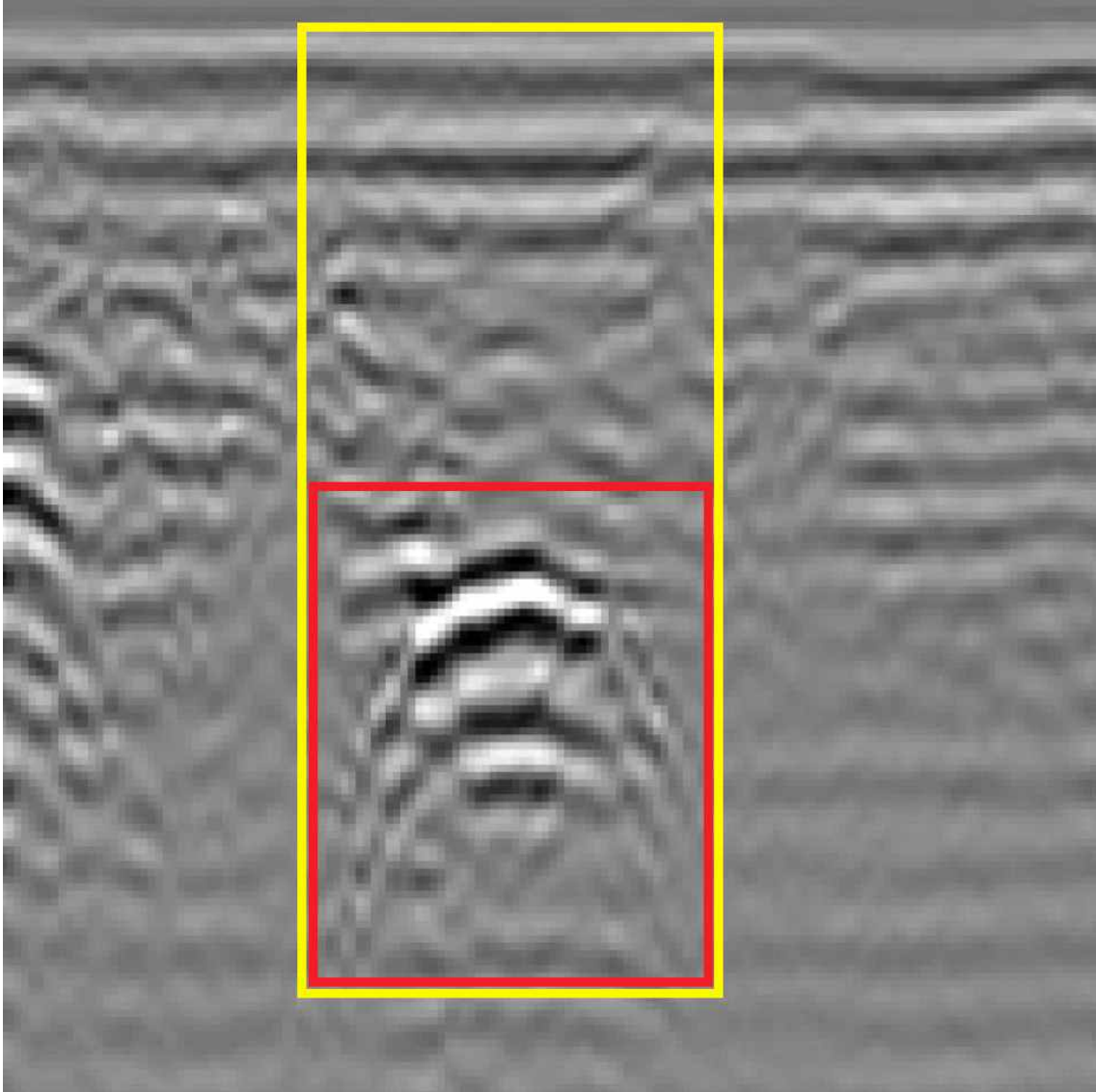


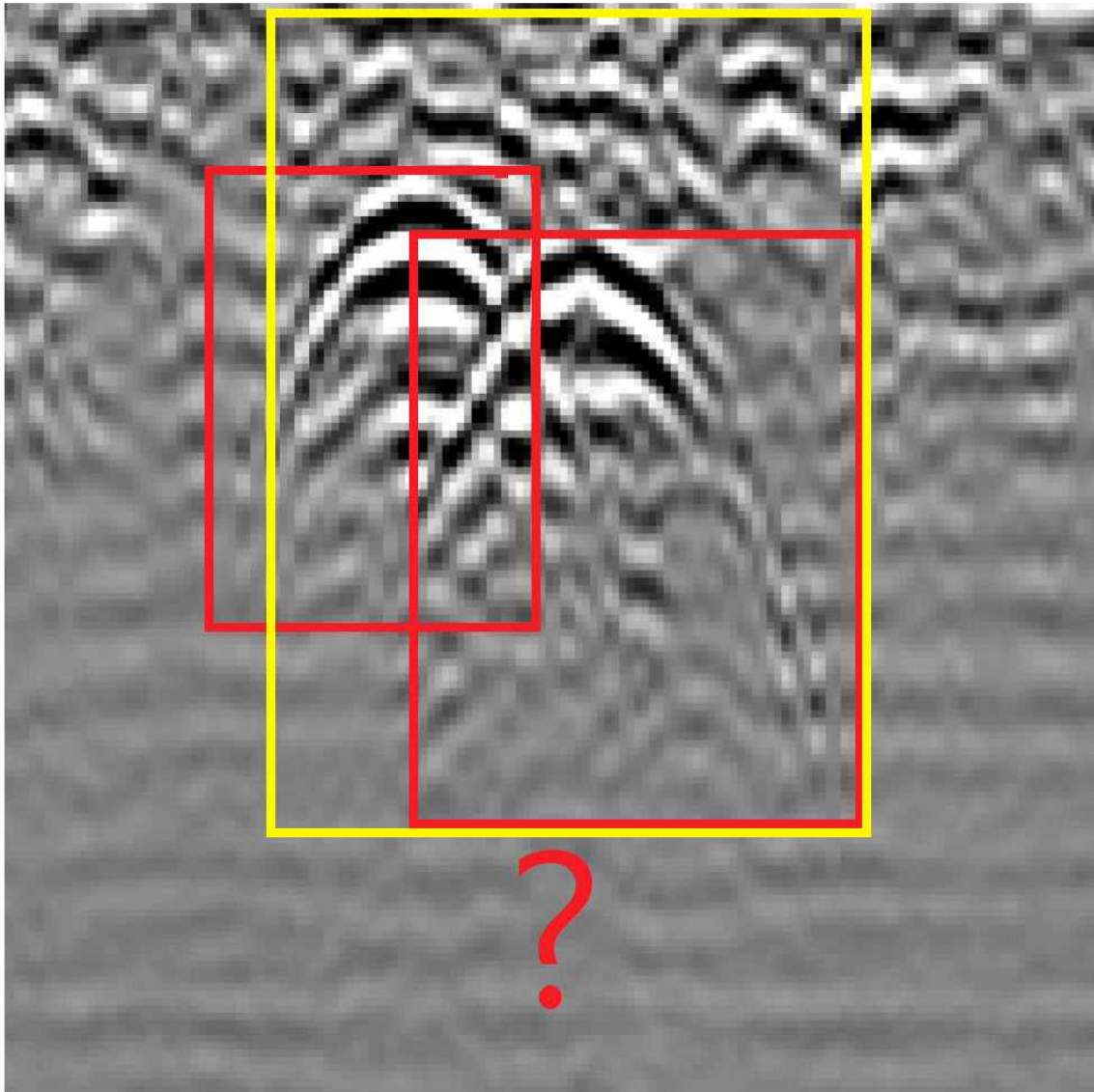
도로에서 공동(Cavity)의 탐지

유에이치에스(주)

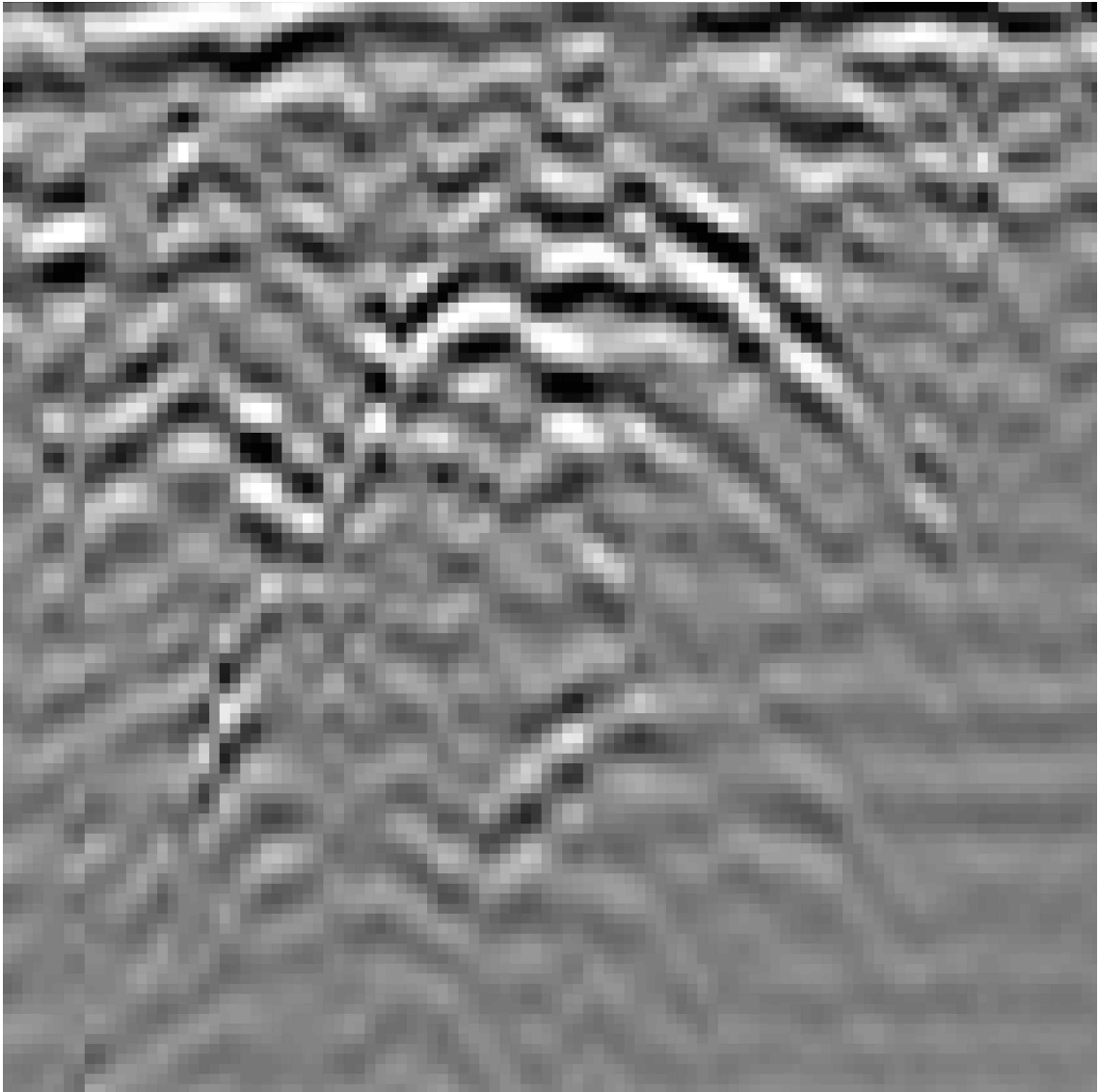
관로 공동 이미지 샘플1



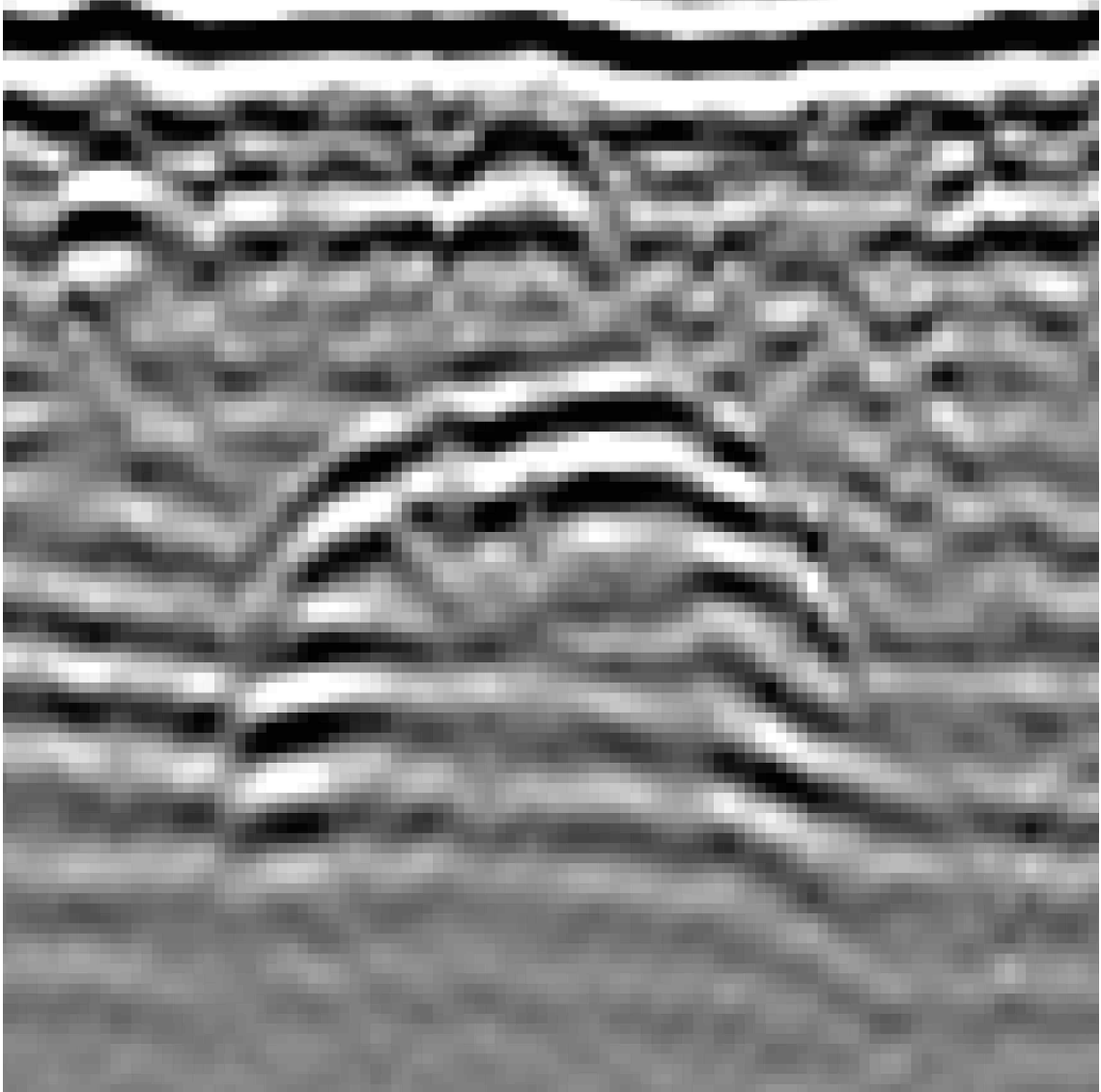
관로 공동 이미지 샘플2



공동 공동 이미지 샘플1



공동 공동 이미지 샘플2



1. 문제의 발견

- 수작업에 의한 공동 탐색이 갖는 직접적인 문제
 - 느린 작업속도(작업효율)로 인한 수익성 문제
 - 작업의 정확도와 균일성(품질), 미발견 공동의 가능성
 - 작업자의 눈/허리 건강
 - 위 3가지 문제에 따른 인력관리의 어려움
 - 공동 관리 업계에서의 상대적 기술 경쟁력

2. 문제의 해결 및 활용

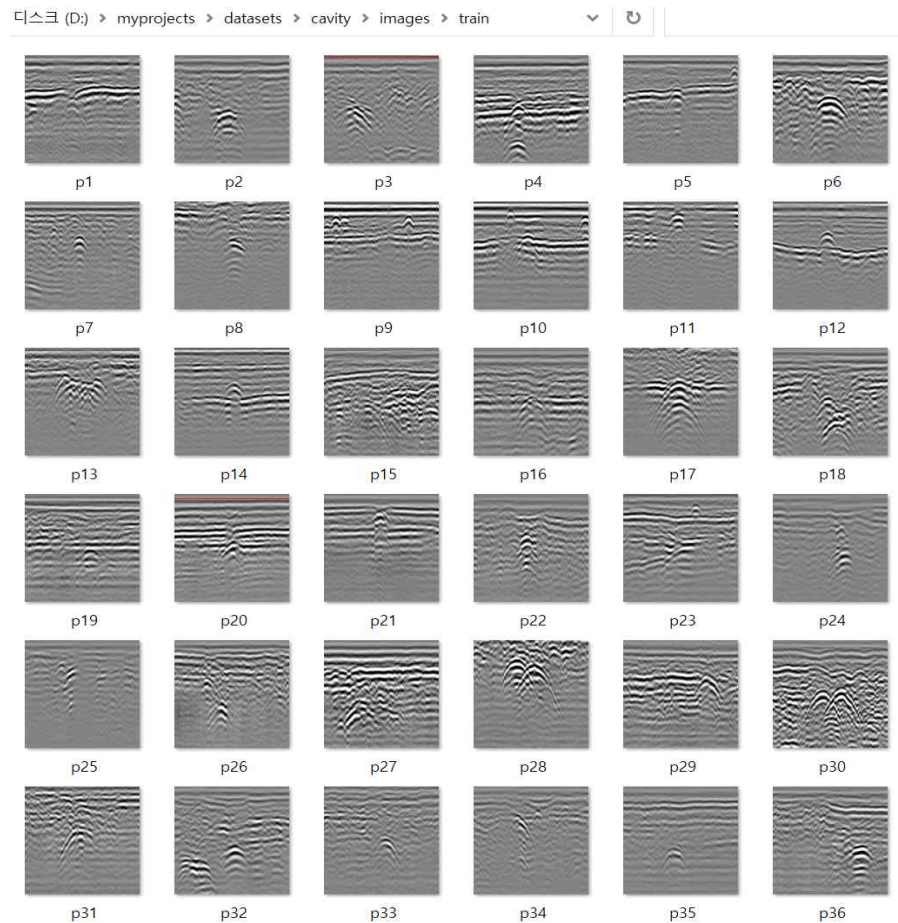
- AI기술을 이용한 자동화로 전술한 문제들을 해결
 - 작업속도의 획기적 개선으로 수익성 증대
 - 작업의 정확도 향상 (기계가 해결 못하는 대상만 사람이 작업)
 - 직원 관리 및 직원 만족도 향상
 - AI기술력을 가짐으로 공동 발견 업계에서 경쟁력 및 기업 가치 상승
 - 해외로 기계 및 용역의 수출

3. 문제 해결을 위한 기술적 문제의 발견

- 충분한 학습용/테스트 데이터의 준비
- 학습 데이터의 정확성이 관건
- 가능한 공동의 종류 및 그에 따른 레이더 이미지의 차이 확인
- 공동의 종류에 따른 학습/출력 방법의 확인(공동과 관로의 구분 여부)
- 종단면 외에 평면 이미지의 활용 여부
- 최적의 학습 환경/변수의 발견 및 설정
- Image Classification과 Object Detection 비교 및 앙상블 기법 사용
- 자동 촬영 및 자동 검사를 위한 장치의 개발

4. YOLO5를 이용한 문제 해결

4.1 데이터(종단면 이미지)의 준비



디스크 (D:) > myprojects > datasets > cavity > labels > train

이름	수정된 날짜	유형	크기
p1	2021-07-19 오후 9:19	텍스트 문서	1KB
p2	2021-07-19 오후 9:20	텍스트 문서	1KB
p3	2021-07-19 오후 9:21	텍스트 문서	1KB
p4	2021-07-19 오후 9:21	텍스트 문서	1KB
p5	2021-07-19 오후 9:21	텍스트 문서	1KB
p6	2021-07-19 오후 9:22	텍스트 문서	1KB
p7	2021-07-19 오후 9:22	텍스트 문서	1KB
p8	2021-07-19 오후 9:22	텍스트 문서	1KB
p9	2021-07-19 오후 9:22	텍스트 문서	1KB
p10	2021-07-19 오후 9:23	텍스트 문서	1KB
p11	2021-07-19 오후 9:23	텍스트 문서	1KB
p12	2021-07-19 오후 9:23	텍스트 문서	1KB
p13	2021-07-19 오후 9:23	텍스트 문서	1KB
p16	2021-07-19 오후 9:24	텍스트 문서	1KB
p17	2021-07-19 오후 9:24	텍스트 문서	1KB
p18	2021-07-19 오후 9:24	텍스트 문서	1KB
p19	2021-07-19 오후 9:24	텍스트 문서	1KB
p20	2021-07-19 오후 9:25	텍스트 문서	1KB
p21	2021-07-19 오후 9:25	텍스트 문서	1KB
p22	2021-07-19 오후 9:25	텍스트 문서	1KB
p23	2021-07-19 오후 9:25	텍스트 문서	1KB
p24	2021-07-19 오후 9:25	텍스트 문서	1KB
p25	2021-07-19 오후 9:26	텍스트 문서	1KB
p26	2021-07-19 오후 9:26	텍스트 문서	1KB
p27	2021-07-19 오후 9:26	텍스트 문서	1KB
p28	2021-07-19 오후 9:26	텍스트 문서	1KB

4.2 학습

```
! cavity.yaml X
D: > myprojects > yolov51 > data > ! cavity.yaml
1
2
3   # Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [pa
4   path: ../datasets/cavity # dataset root dir
5   train: images/train # train images (relative to 'path') 128 images
6   val: images/valid # val images (relative to 'path') 128 images
7   test: images/test # test images (optional)
8
9   # Classes
10  nc: 2 # number of classes
11  names: [ 'pipe','cavity' ] # class names
12
13  |
14  # Download script/URL (optional)
15  #download: https://github.com/ultralytics/yolov5/releases/download/v1.0/coco128.zip
```

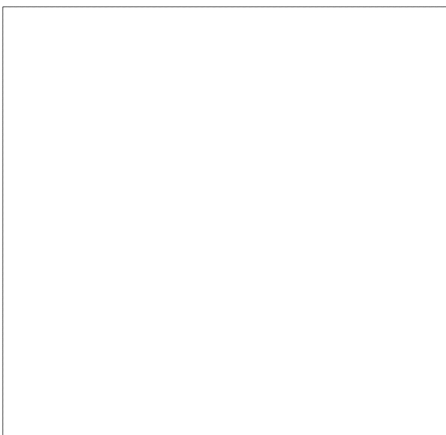
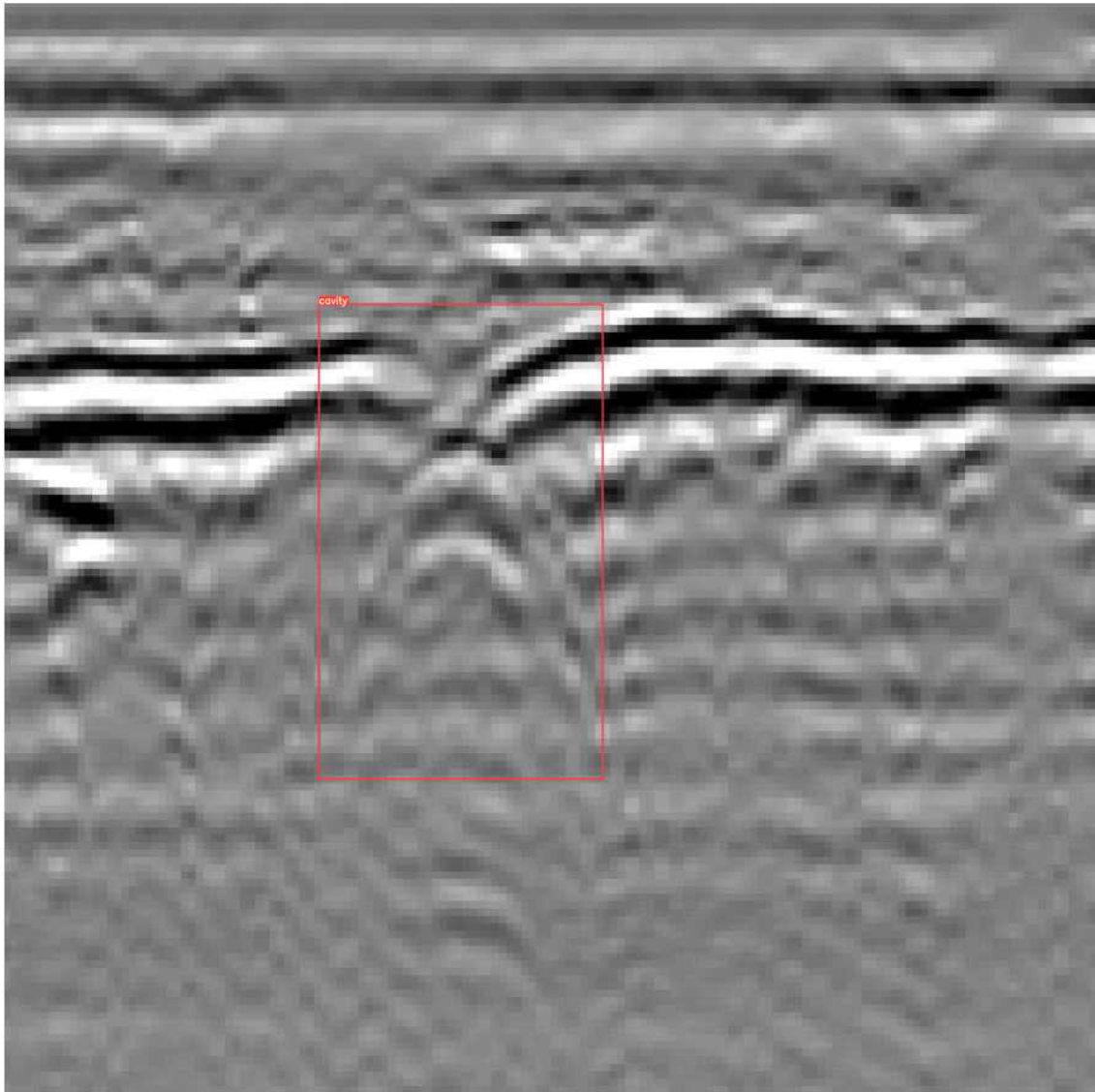
```
python train.py --img 640 --batch 8 --epochs 1000 --data
data/cavity.yaml --cfg models/yolov5l.yaml --weights yolov5l.pt
```

4.3 실행

```
python detect.py --source ../datasets/cavity/images/train --image
1024 --weights runs/train/exp??/weights/best.pt --conf 0.5
```

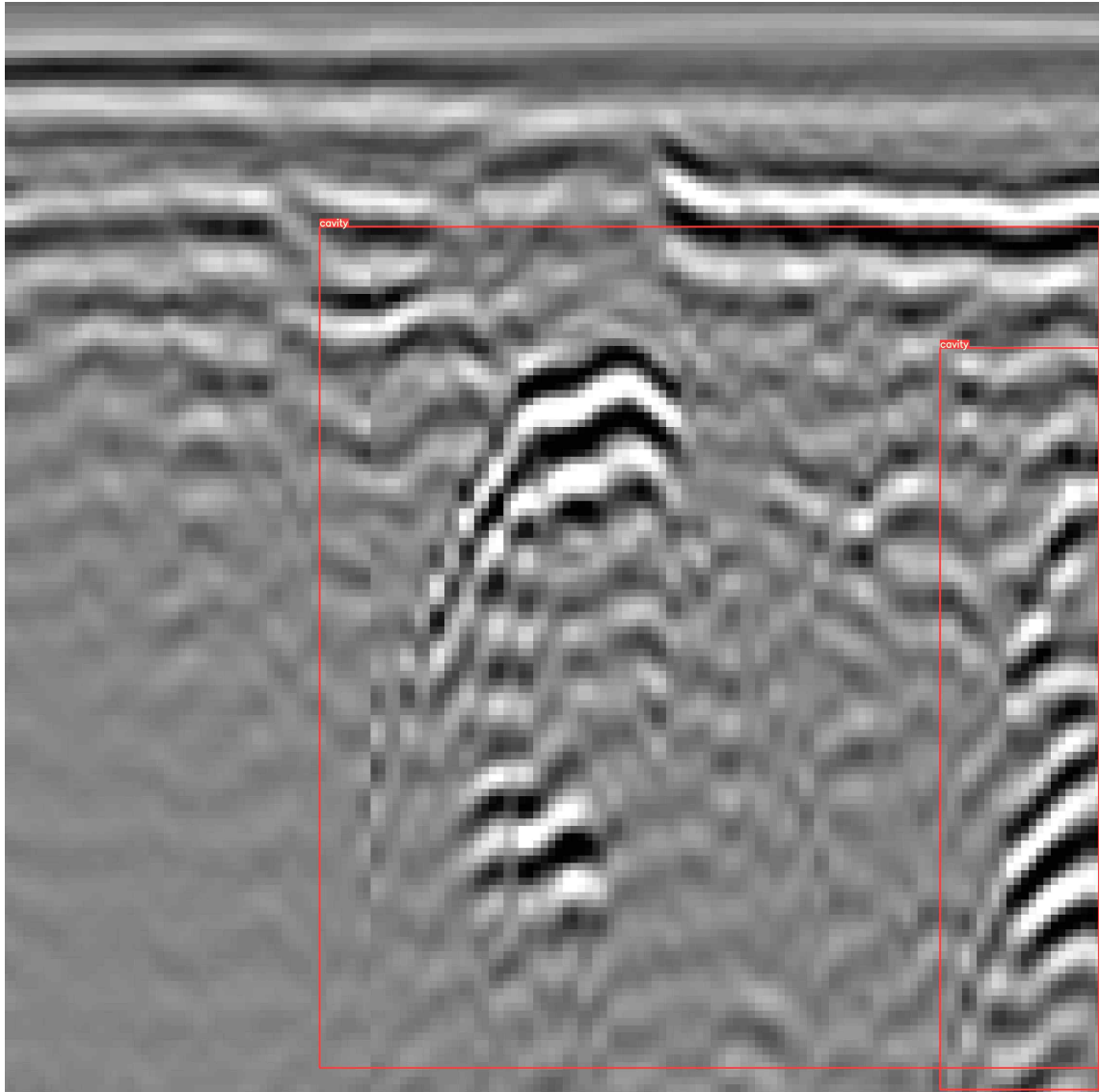
(see next page)

학습 데이터에 대한 인식 결과
<종단면 이미지>

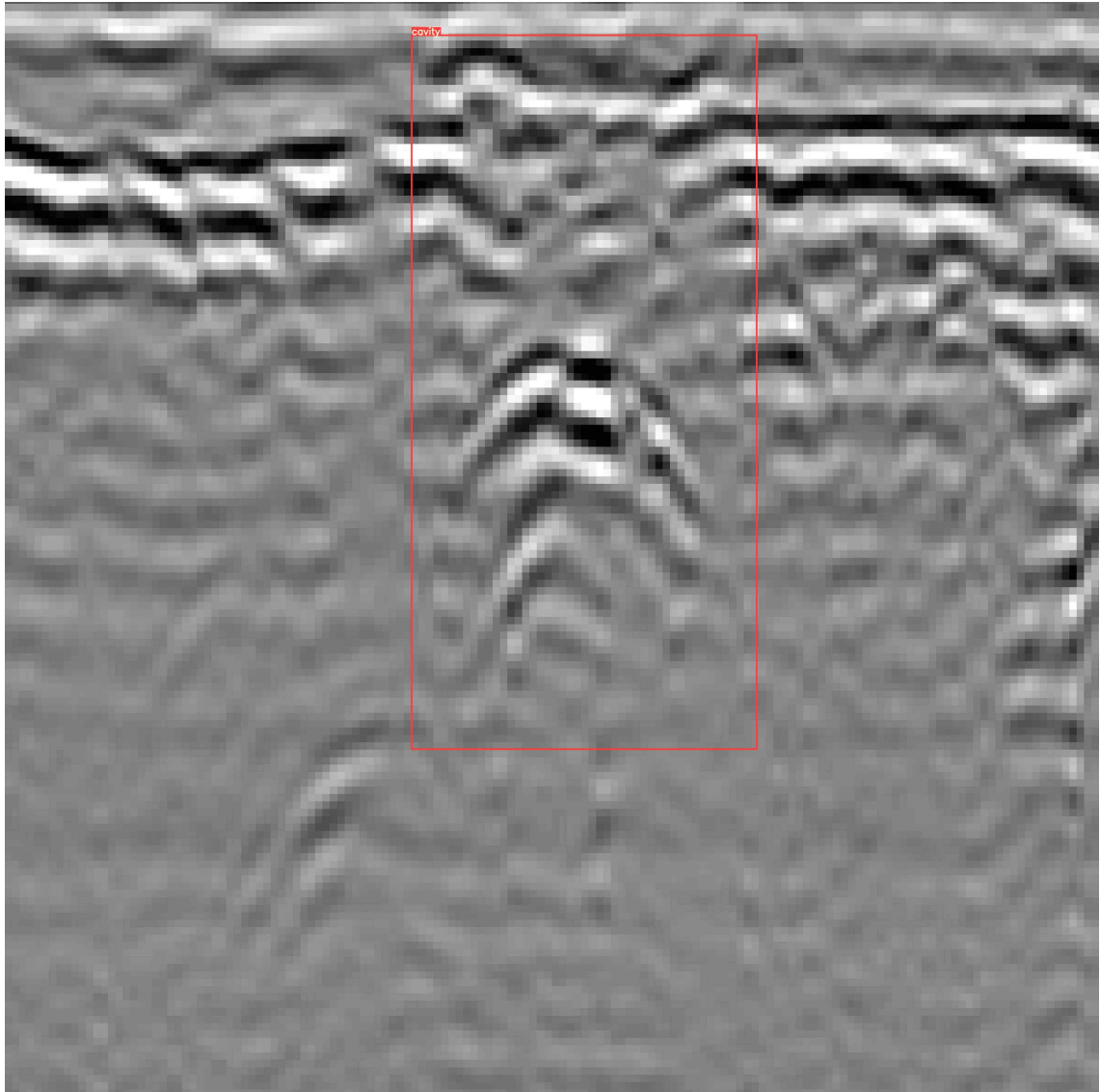


<평면 이미지의 예>

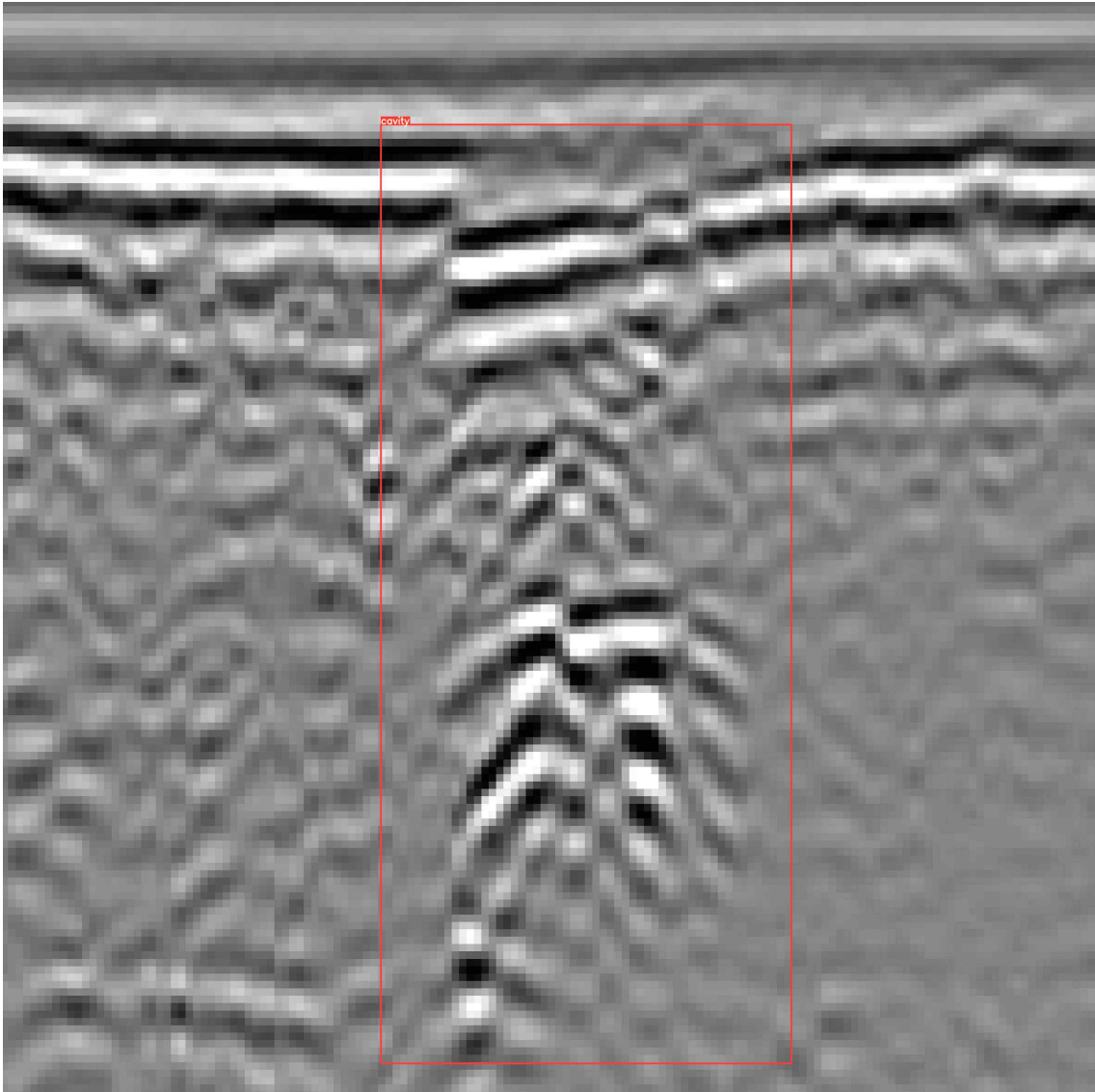
테스트 데이터에 대한 인식 결과 1



테스트 데이터에 대한 인식 결과 2



테스트 데이터에 대한 인식 결과 3



[[결론]]

1. YOLOv5 만으로도 충분히 공동의 발견이 가능할 것으로 판단
2. 제일 중요한 요소는 충분한 데이터 확보와 정확한 라벨링
3. 실제 사용을 위해서는 앙상블 기법으로 신뢰도를 높일 필요가 있음