

객체탐지를 활용한 반도체 불량 자동 분류 모델 구현

DATA FLOW

김나영
김예슬
손아
최정인
한나영

분석 환경 및 활용 라이브러리

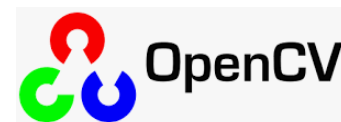
분석언어



개발환경



라이브러리 및
프레임워크



CONTENTS

1. 프로젝트 개요

**2. 프로젝트
팀 구성 및 역할**

**3. 프로젝트
수행 절차 및 방법**

**4. 프로젝트
수행 결과**

**5. 자체
평가 의견**

개요

목적

객체 탐지를 활용하여 반도체 결함 정보/개수 및
수율을 보여주는 웹 구현

기대효과

- 품질 검수를 통한 반도체 수율 증가
- EDS 과정에서의 인건비 절약



**반도체 제조 과정의
효율성과 생산성 향상**

팀 구성 및 역할

김나영

모델링

데이터 분석 및 자료조사

이미지 전처리 및 어노테이션(로보플로우)

김예슬

데이터 분석 및 자료조사

이미지 전처리 및 어노테이션(로보플로우)

손아

웹 서비스 구현

데이터 분석 및 자료조사

최정인

모델링

이미지 전처리 및 어노테이션(로보플로우)

PPT 제작 및 발표

한나영

웹 서비스 구현

이미지 전처리 및 증식(OpenCV)

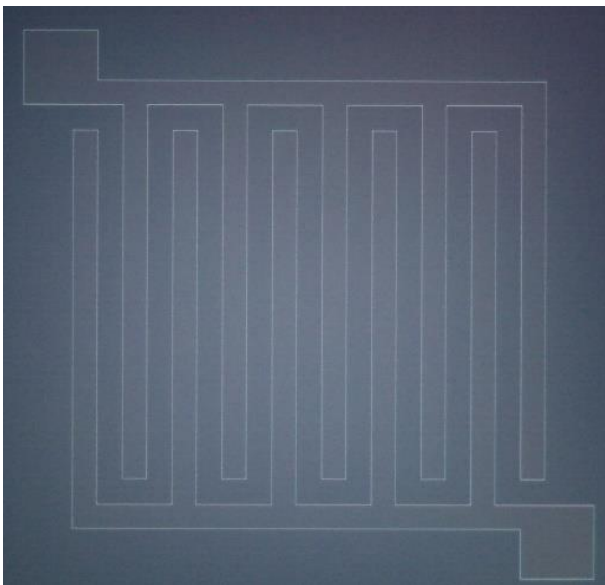
데이터 분석 및 자료조사

반도체 8대 공정 소개

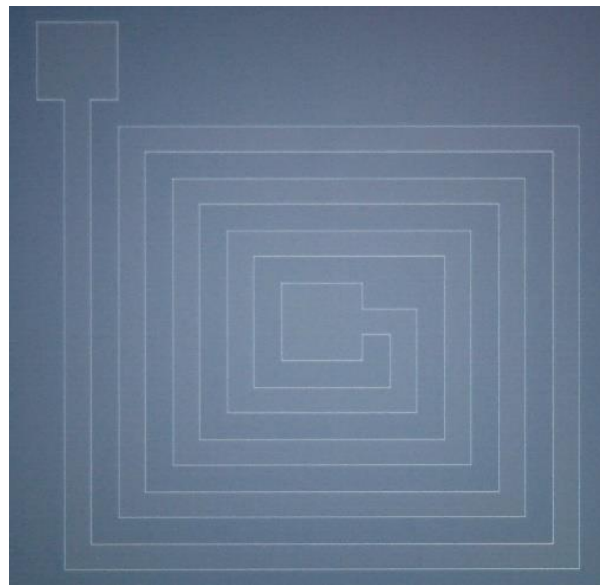


데이터 탐색

이미지 정보	반도체 사진	
파일 형식	Jpg	
패턴 종류	2 가지	
이미지 데이터 총 개수	Pattern1 : 136장 Pattern2 : 201장	Total : 337장





Pattern 1





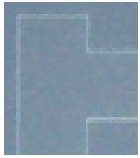

Pattern 2

결함 유형 정의

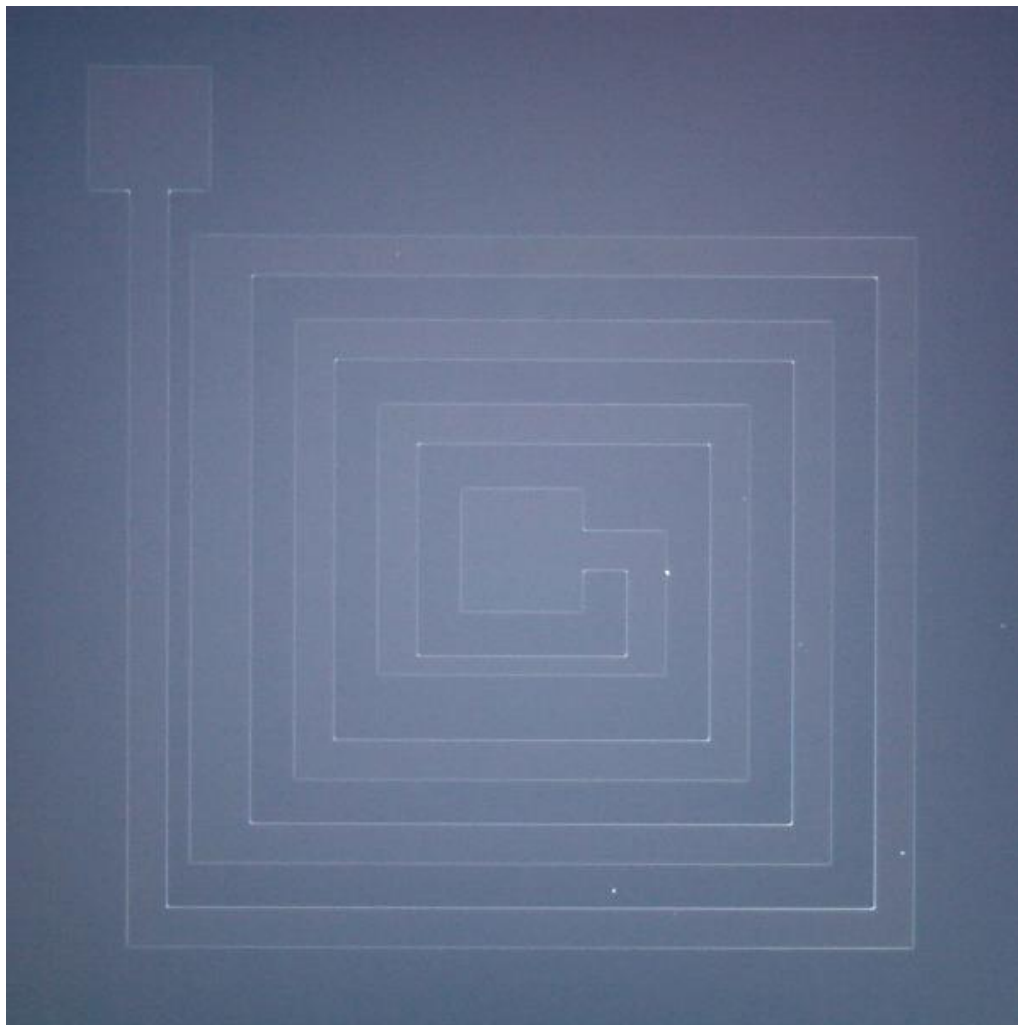
■ 이물 결함 : 2 가지

	
dot	long
작은 형태	긴 형태

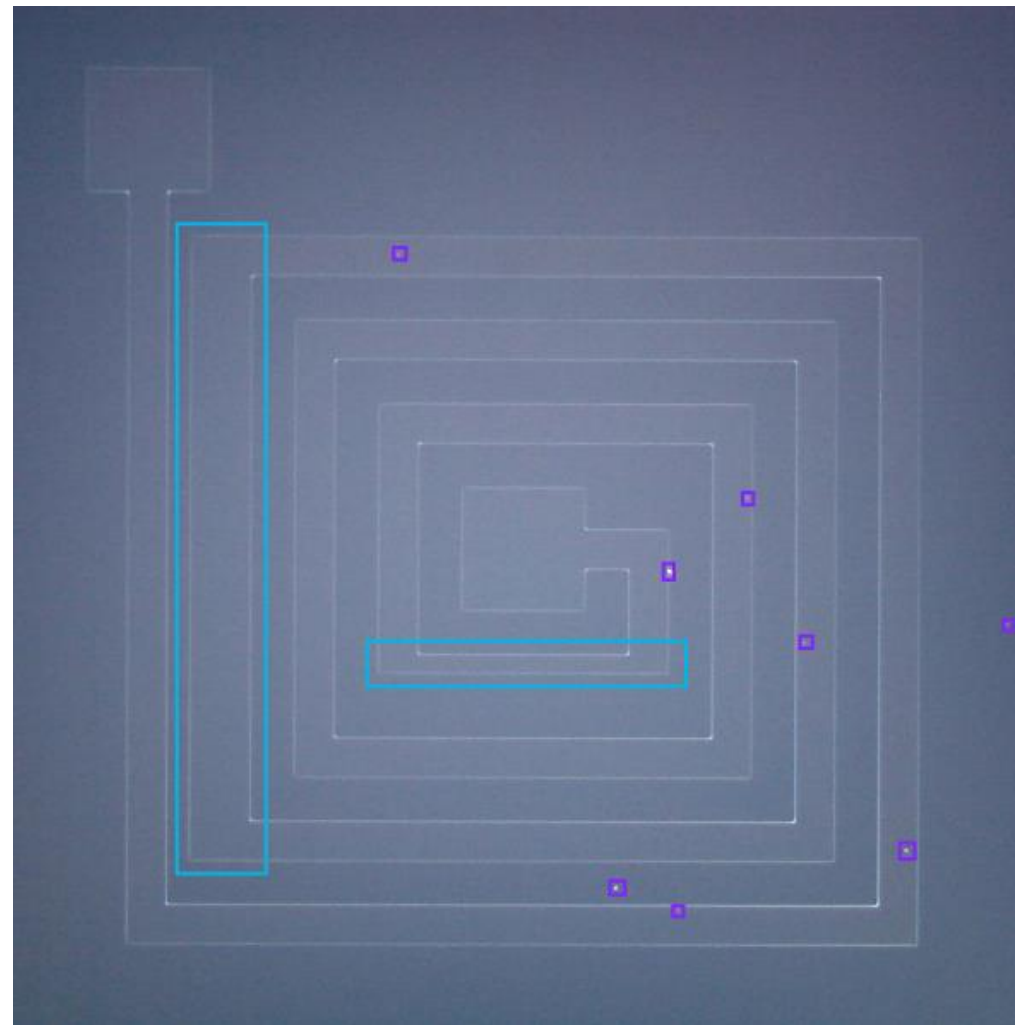
■ 회로 결함 : 8 가지

							
circle	round	cut	slope	empty	square	block	thickness
원	흰 회로	끊긴 회로	잘린 회로	빈 회로	들어간 회로	돌출된 회로	두께가 다른 회로

ROBOFLOW 활용



■ 원본



■ 바운딩 박스 적용 후

객체 탐지 모델

예시 모델

- R-CNN
- Fast R-CNN
- Faster R-CNN

신속성↓
정확성↑

예시 모델

- YOLO
- SSD
- FPN

신속성↑
정확성↓

1 단계

2 단계

다단계 탐지
2 stage detector

단일 단계 탐지
1 stage detector

모델별 성능 비교

초기

	mAP50
YOLOv8	0.351

전처리



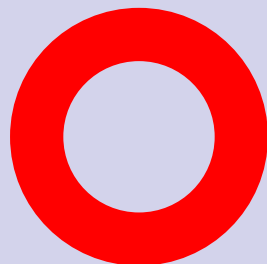
중간

	mAP50
YOLOv8	0.713
DETECTRON2	0.848

학습 소요시간
약 20분 내외

약 7시간

- 데이터 증강
- 라벨링 조정



현재

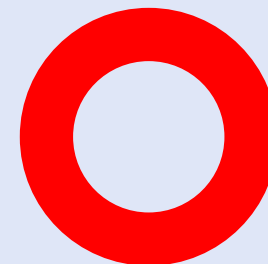
	mAP50
YOLOv8	0.961
YOLOv7	0.976
YOLOv5	0.963

학습 소요시간
약 20분 내외

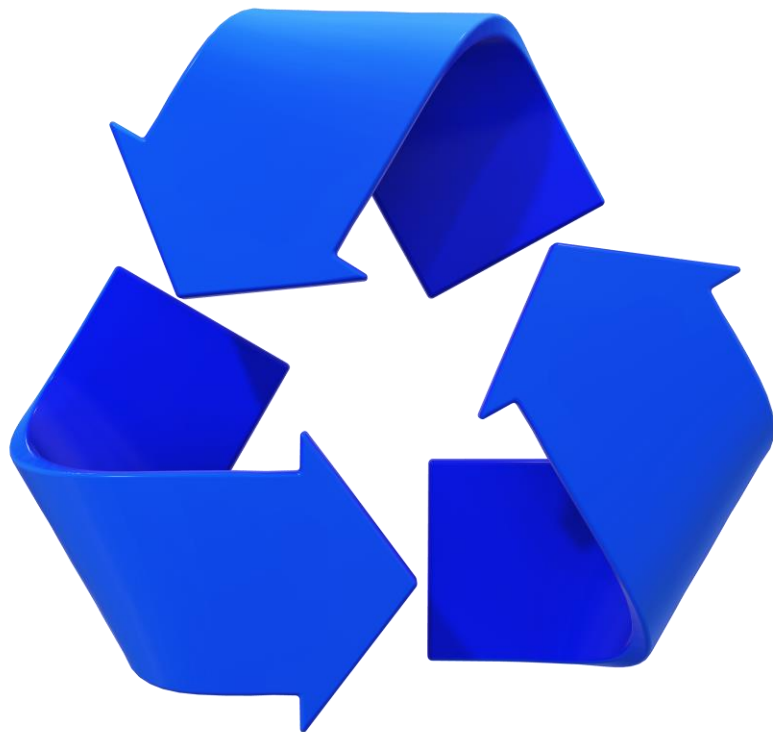
약 50분 내외

약 20분 내외

- 데이터 증강
- 라벨링 조정
- 하이퍼파라미터 조정



전처리 - 데이터 리셋

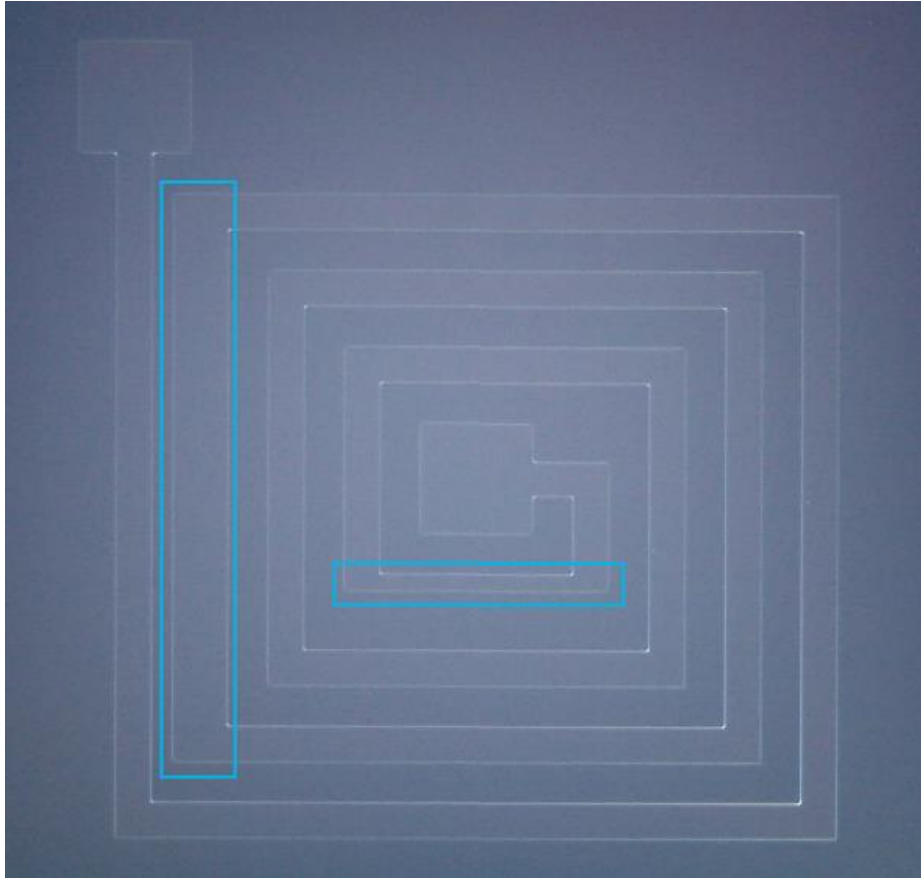


YOLOv8 기준
mAP50 : 0.713

1. 유형별로 바운딩 박스 재조정
2. 클래스 재정의

YOLOv8 기준
mAP50 : 0.775

전처리 - 데이터 증강 실패 사례



TEST셋 이미지를 크롭화

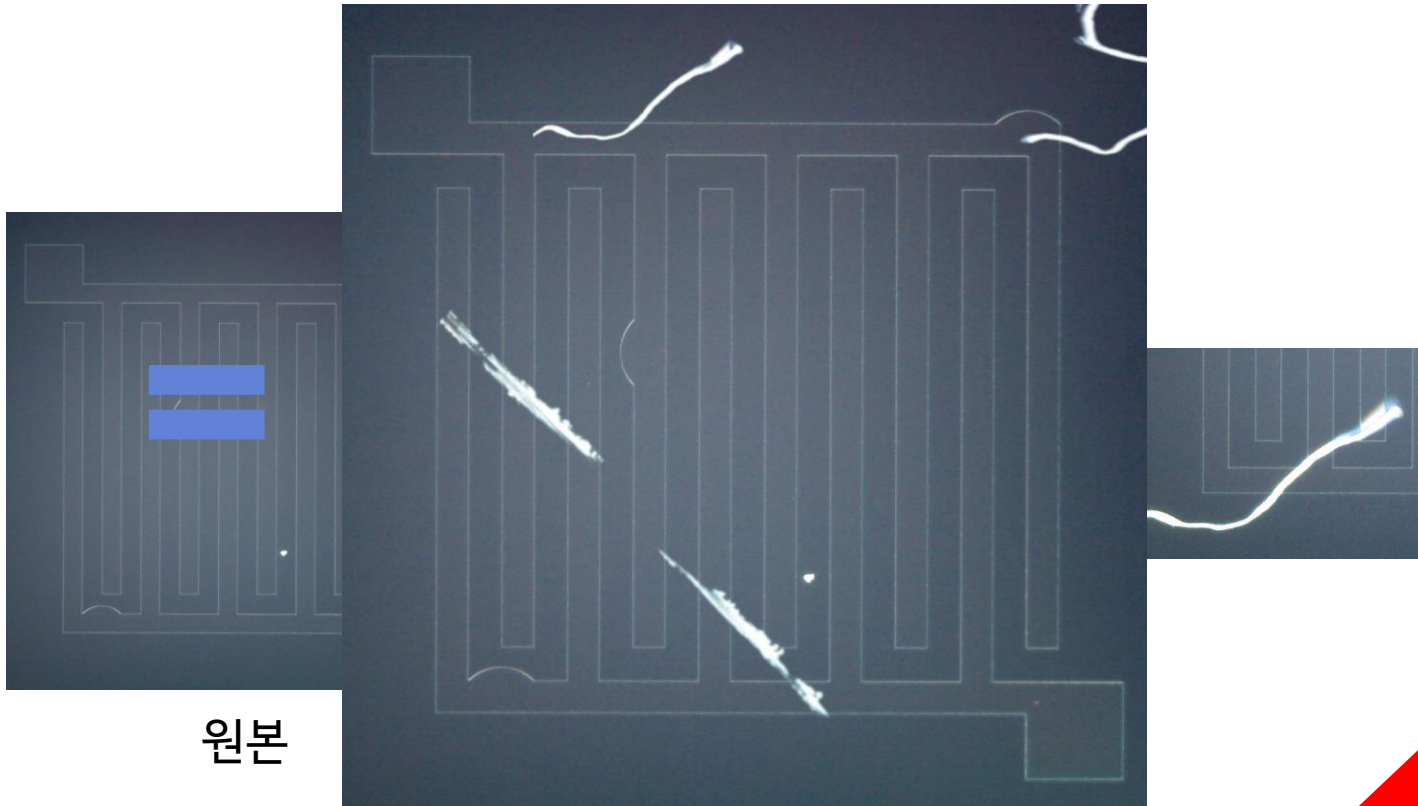
YOLOv8 기준
mAP50 :
약 0.7 → 0.4



바운딩 박스 및 결함 자체가 분리되어
모델이 이를 인식하기 어렵다 판단

전처리 - 데이터 증강

1

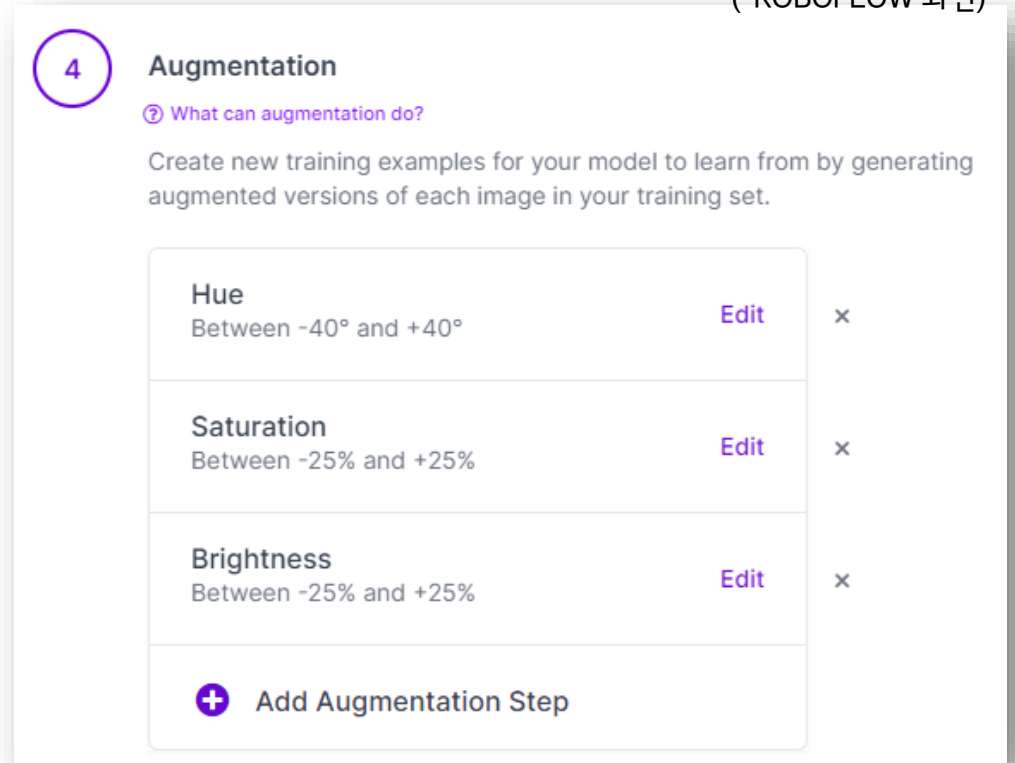


원본

합성된 이미지

2

(*ROBOFLOW 화면)



YOLOv8 기준
mAP50 :
0.775 → 0.884

이물 결함(long)
AP 대략 0.6

색조, 채도, 밝기 옵션으로
추가 증강
(ROBOFLOW 활용)

전처리 - BOUNDING BOX 재조정



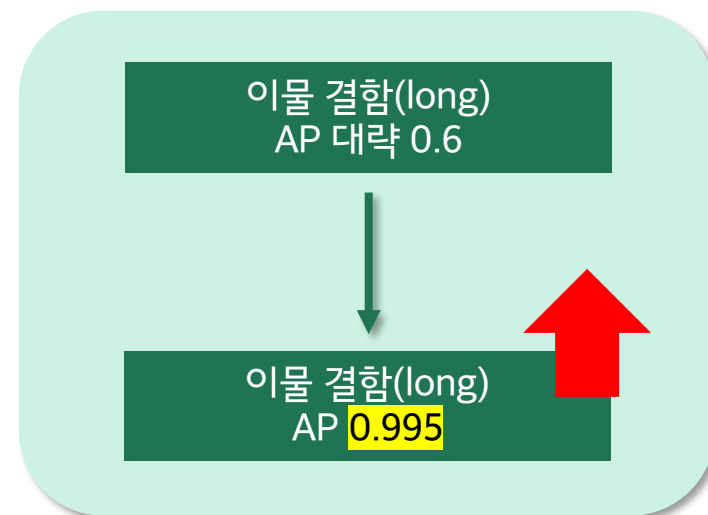
멀티 박스 형식



테두리 형식



박스 형식

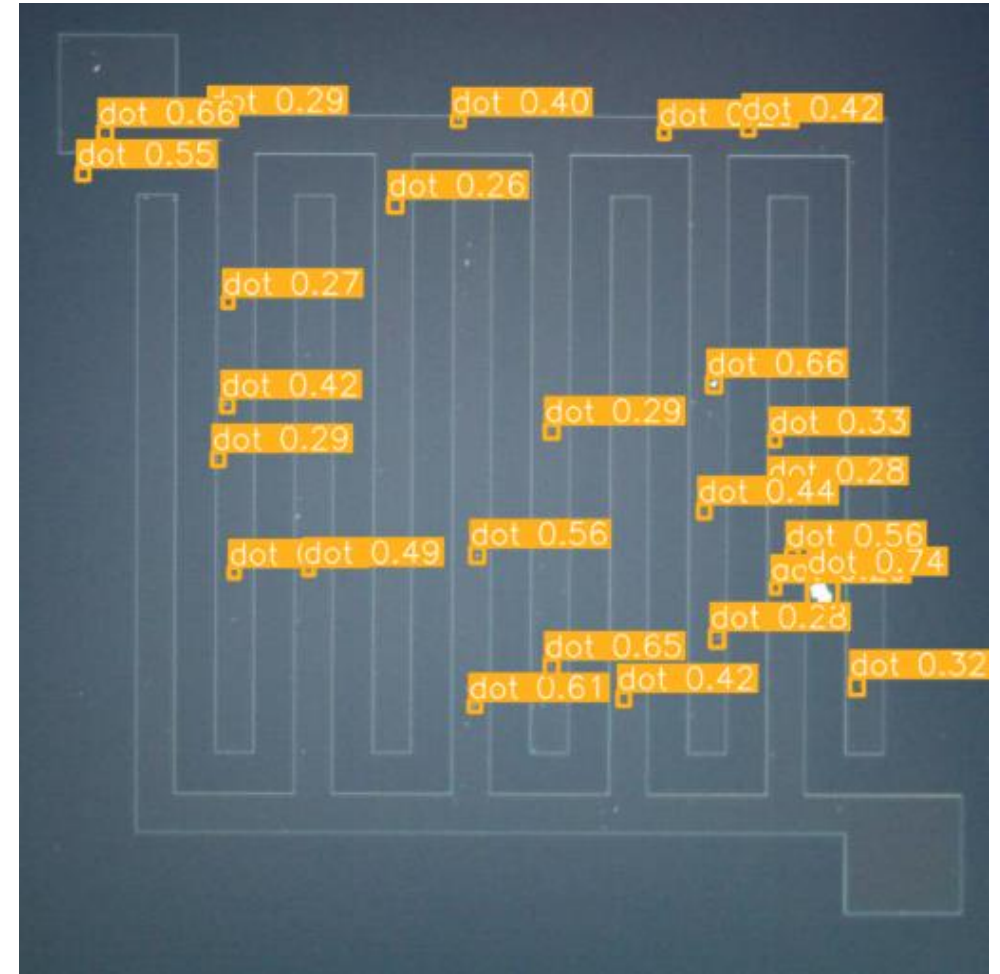


YOLOv8 기준
mAP50 :
0.884 → 0.959

전처리 - 하이퍼파라미터 조정 1



예측 박스가 크게 잡히는 문제점



하이퍼파라미터 Weight decay(가중치 감소) 수치 조정으로 해결

YOLOv8 : 0.001 → 0.00015

증식 옵션별 성능 비교 - YOLOv8 기준

	mAP50
VALIDATION	0.951
TEST	0.945

회전+뒤집기

	mAP50
VALIDATION	0.908
TEST	0.94

색변환

	mAP50
VALIDATION	0.905
TEST	0.938

노이즈

	mAP50
VALIDATION	0.947
TEST	0.941

합성

	mAP50
VALIDATION	0.956
TEST	0.948

회전+뒤집기+색변환+합성

최종 모델



데이터셋 구성 비율

Training Set 85%

777 images

Validation Set 4%

41 images

Testing Set 11%

99 images

모델 선정 기준

- 객체 탐지 성능
- 걸리는 시간
- 일반화 능력

하이퍼파라미터 정보

- batch: 64
- weight_decay: 0.00015
- patience: 30
- momentum: 0.9
- epochs: 100
- lr0(초기학습률): 0.00015

클래스별 구성 비율

dot	991
round	158
circle	156
long	128
slope	63
thickness	38
cut	36
square	36
block	20
empty	11

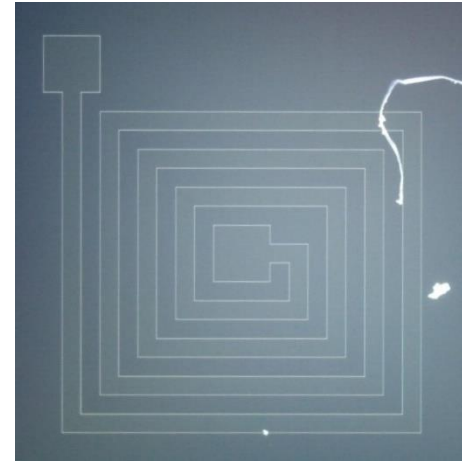
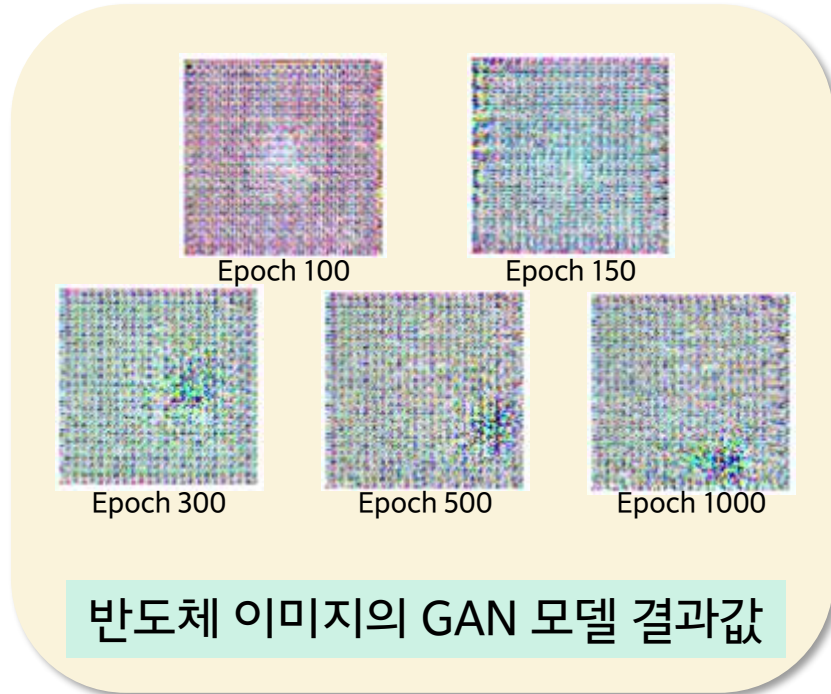
클래스별 mAP 수치

Class	mAP50	mAP50-95
block	0.995	0.696
circle	0.995	0.782
cut	0.995	0.697
dot	0.632	0.246
Empty	0.995	0.838
long	0.891	0.493
round	0.988	0.699
slope	0.995	0.703
square	0.995	0.735
Thickness	0.995	0.886

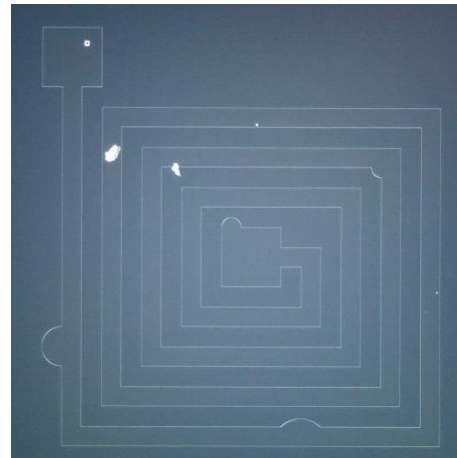
mAP별 수치

mAP	0.677
mAP 50	0.948
mAP 75	0.792
mAP50-95	0.677

데이터 증강 모델 - GAN



“새로운 이미지”



CycleGAN

StyleGAN

PGGAN

ProGAN

...

GAN 계열 모델들

1. 원본 해상도
2. 소량의 데이터

웹 구현 (FLASK)



자체 평가 의견

김나영

흔하게 구할 수 있는 데이터가 아닌 반도체라는 특수한 데이터셋을 접해볼 수 있어 좋았습니다. 데이터의 수가 적어 난항을 겪었지만, 오히려 그로 인해 (목표 성능을 끌어올리기 위해) 다양한 시도들을 했던 것이 학습에 많은 도움이 되었습니다. 아쉬운 점은 데이터 증식이나 라벨링에 집중하느라 모델에 관한 공부가 조금 부족했다는 것이었습니다.

김예슬

쉽게 접할 수 없었던 반도체 분석을 함께 고민하고 배워나갈 수 있어서 의미있는 경험이었습니다. 딥러닝 부분에서 처음 접해보는 정보들이 많아 어려움을 겪었지만, 팀원들과 함께 협력하며 조금씩 습득해나갈 수 있었습니다. 이번 기회를 통해 배운 내용을 내 것으로 만들고, 더 배워나가야겠다는 생각이 들었습니다.

손아

반도체 불량 검출 프로그램을 만들면서 전반적으로 웹 만드는 작업을 도맡아 했습니다. 웹을 만들며 적용하는 모델을 모르겠다 느껴 공부를 하게 되었습니다. 코딩을 하며 웹을 자세히 공부한 적이 없었는데 이번 프로젝트를 진행하면서 웹에 흥미를 느끼게 되었습니다. 딥러닝과 머신러닝 웹을 할 수 있게 되어 저의 역량을 한층 더 업그레이드 시킨 시간이었습니다.

최정인

이번 프로젝트를 통해 책으로만 공부하는 것과 달리 충분히 내가 연구하고 여러 시행착오를 겪으며 오류를 해결할 수 있는 경험을 얻을 수 있었습니다. 하나의 데이터를 가지고 성능 평가를 하기에 앞서 전처리 과정에서 생각 보다 많은 시간을 소요하게 되었습니다. 이로 인해 YOLO 외 다른 모델에 대한 학습 시간을 할애하기 어려웠습니다. 차후 또 다른 모델을 연구해보고자 하는 학구열이 고무되었습니다.

한나영

이번 프로젝트를 통해 불량품 탐지가 필요한 반도체 이미지를 입력 시 자동으로 불량품 예측해주는 프로그램을 만들어 보면서 지금까지 배웠던 딥러닝을 활용할 수 있어서 좋았습니다. 특히, 반도체에 대해 알게 된 점과 roboflow, openCV와 웹과 객체탐지 모델들에 대해 자세히 알게 되어 좋은 경험이고 성장의 기회가 되었다고 생각합니다.



Q & A



THANK YOU