

불량 반도체 탐지 프로그램 개발

YOLO v5를 이용한 객체 검출 알고리즘의 활용



목차

1 프로젝트 소개

2 데이터 분석 및 모델 개발

3 Web Application 구현

4 프로젝트 정리 및 소감

Part 1 프로젝트 소개



프로젝트 소개

1

기획배경 및 목적

2

데이터셋 소개

3

개발 환경

기획배경 및 목적

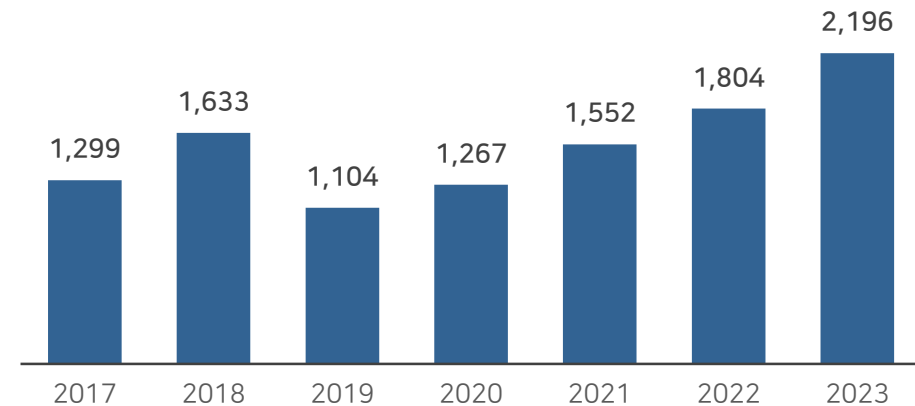
- 5G 시장 확대, 비대면 경제 확산 지속
→ 전반적인 반도체 산업 수요 증가
- 기업체의 수요 대비 부족한 전문인력



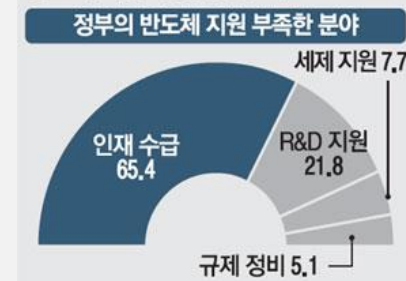
반도체산업의 성장방향에맞춘
객관적인 공정 자동화 및
전문 인력 보완 요소의 필요성 증대

글로벌 메모리 반도체 시장 전망

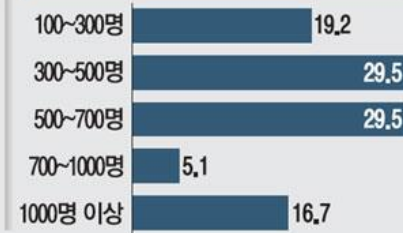
(단위:억달러)



반도체 기업 설문 (단위=%)



수도권대 반도체 전공 적정 정원



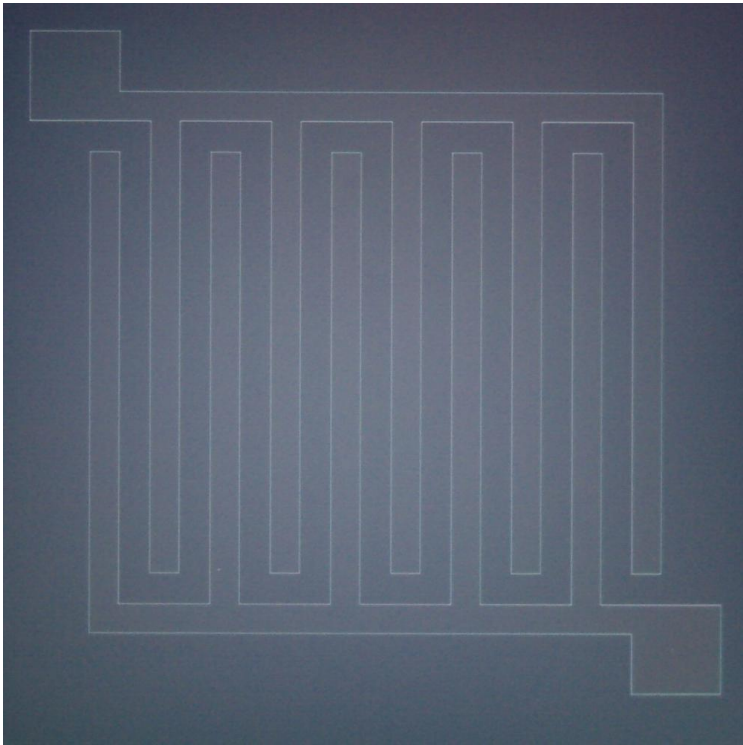
*설문 대상은 국내 반도체 생산 기업 78개사, 조사 기간은 12월 13~20일, 자료=매일경제·한국반도체산업협회

자료 출처: IC인사이트, 매일경제

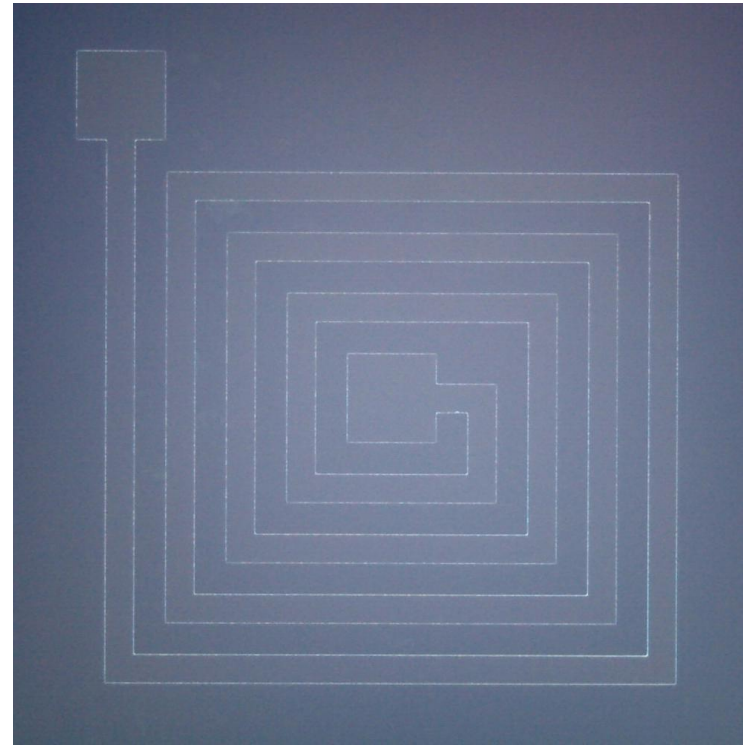
기획 배경 및 목적

기존 EDS 공정의 한계	딥러닝 알고리즘 도입 시 기대 효과
검출자의 숙련도나 컨디션에 크게 영향을 받음	기존 결함 학습을 통한 객관적인 결함 탐지
반도체 공정의 미세화에 따른 기계 고장으로 인한 신규 결함 발생	실시간으로 발생하는 새로운 결함에도 즉각적인 대처 가능
<p>⇒ 검출자의 영향을 받지 않으며 일정 수준 이상의 검출 능력을 갖춘 자동화 시스템으로 반도체 수율 상승, 인건비 절감 등의 기업 경쟁력 제고</p>	

데이터셋 소개



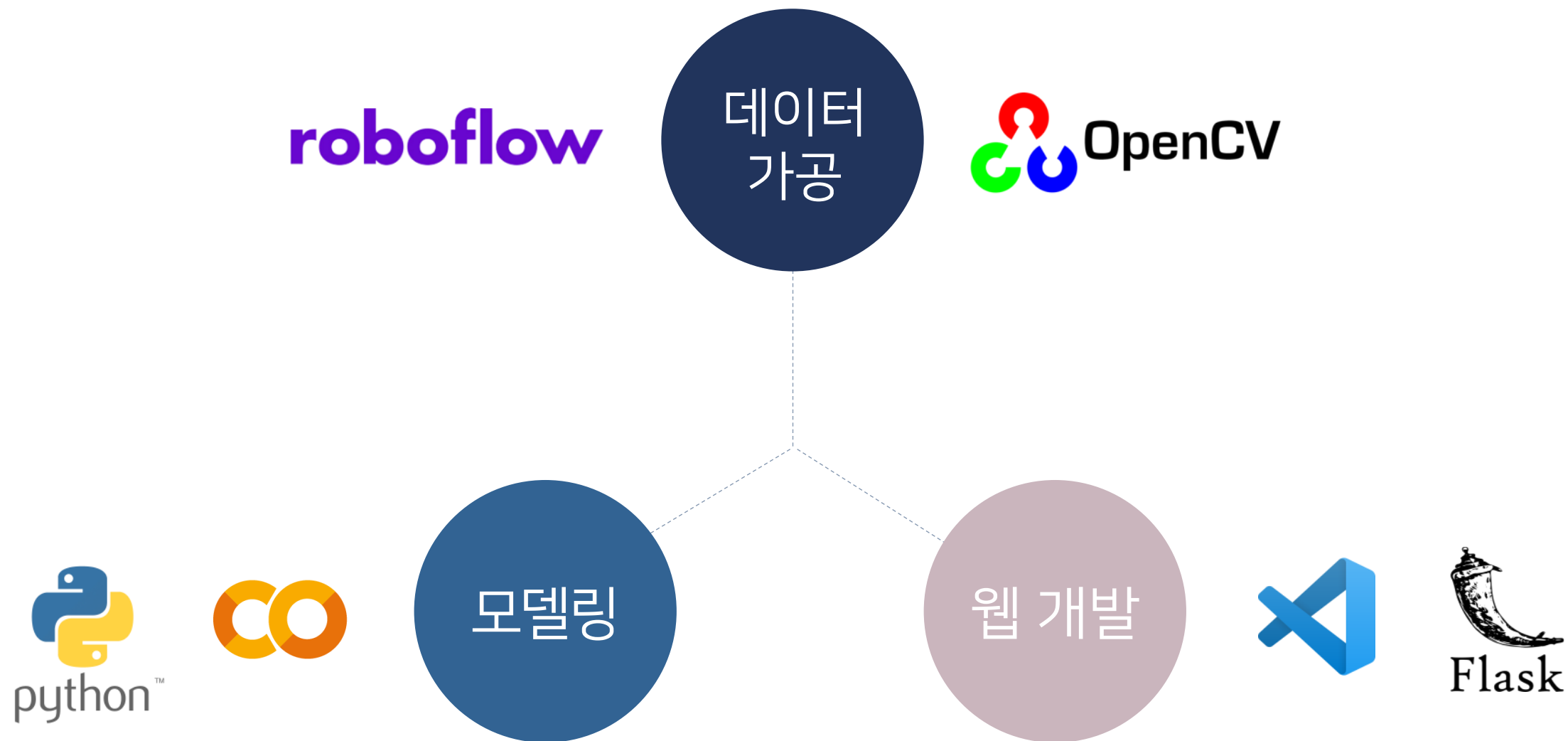
패턴 1



패턴 2

- 데이터 형식: JPG 이미지(1000*1000px)
- 데이터 개수: 280 장
- 동일한 환경에서 촬영된 반도체 웨이퍼 사진

개발 환경



Part 2 데이터 분석 및 모델 개발



데이터 분석 및 모델 개발

1

프로세스 요약

2

데이터 가공 (어노테이션/데이터 증식/라벨링)

3

모델링

결함의 유형 파악 후 바운딩박스처리

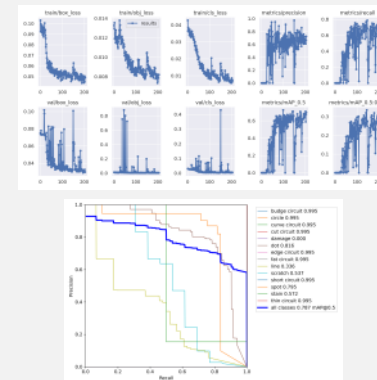
Step 2

YOLO Series
Faster R-CNN
RetinaNet
⋮

데이터셋에 적합한 객체 검출 알고리즘 찾기

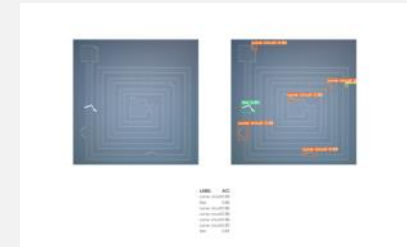
>>

Step 3

지표해석및
모델파라미터최적화

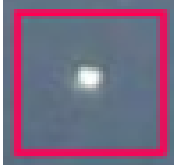

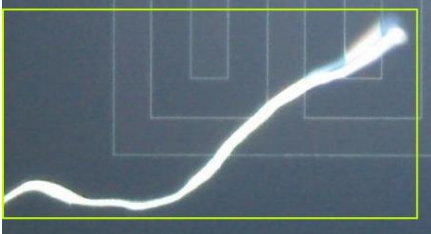

>>

Step 4

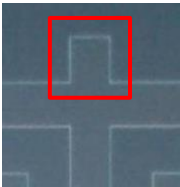


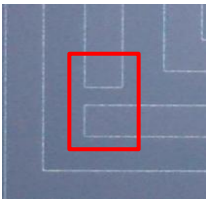
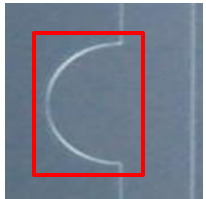





기술확장을위한 Web Application 구현

데이터 가공 — 어노테이션

구분	어노테이션 명	결함 예시
작은 점	dot	
큰 점	spot	
곡선형 이물질	line	
긁힌 자국	scratch	

데이터 가공 — 어노테이션

구분	어노테이션 명	결함 예시	구분	어노테이션 명	결함 예시
튀어나온 결함	bulge circuit		길게 잘린 회로	cut circuit	
구멍 결함	circle		짧게 잘린 회로	short circuit	
곡선형 결함	curve circuit		굵은 회로	fat circuit	
사선으로 잘린 결함	edge circuit		가는 회로	thin circuit	

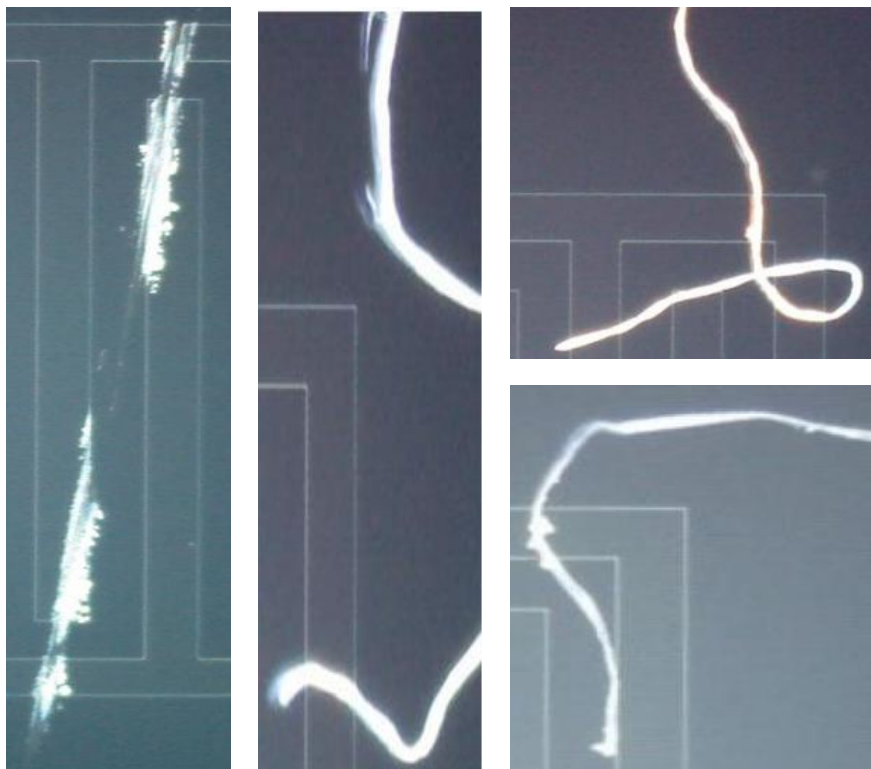
한정된 객체의 수, 비정형 결함의 존재

“어떻게 하면 학습 효율을 높일 수 있을까?”

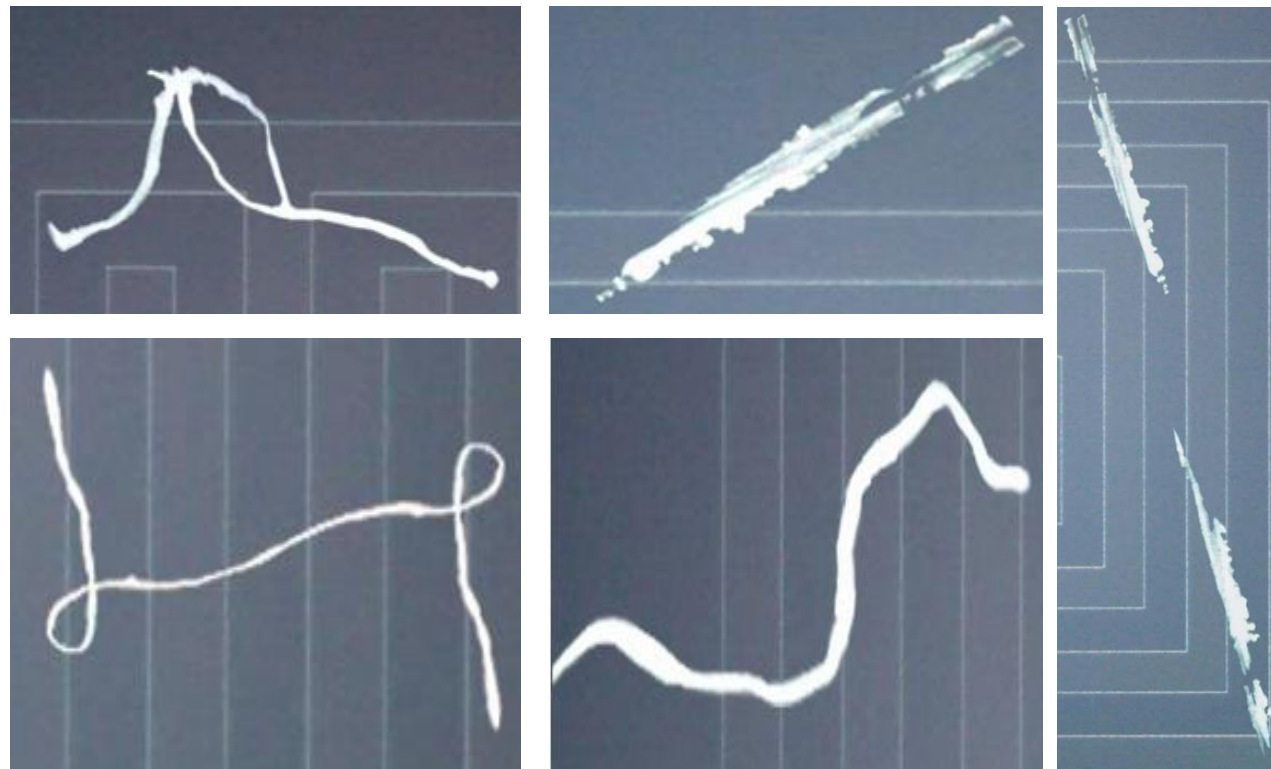
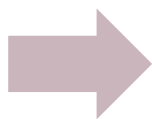
데이터 가공 — 데이터 증식

특정 결함의 절대량 부족

전체 이미지 280장 중 line 11개, scratch 10개



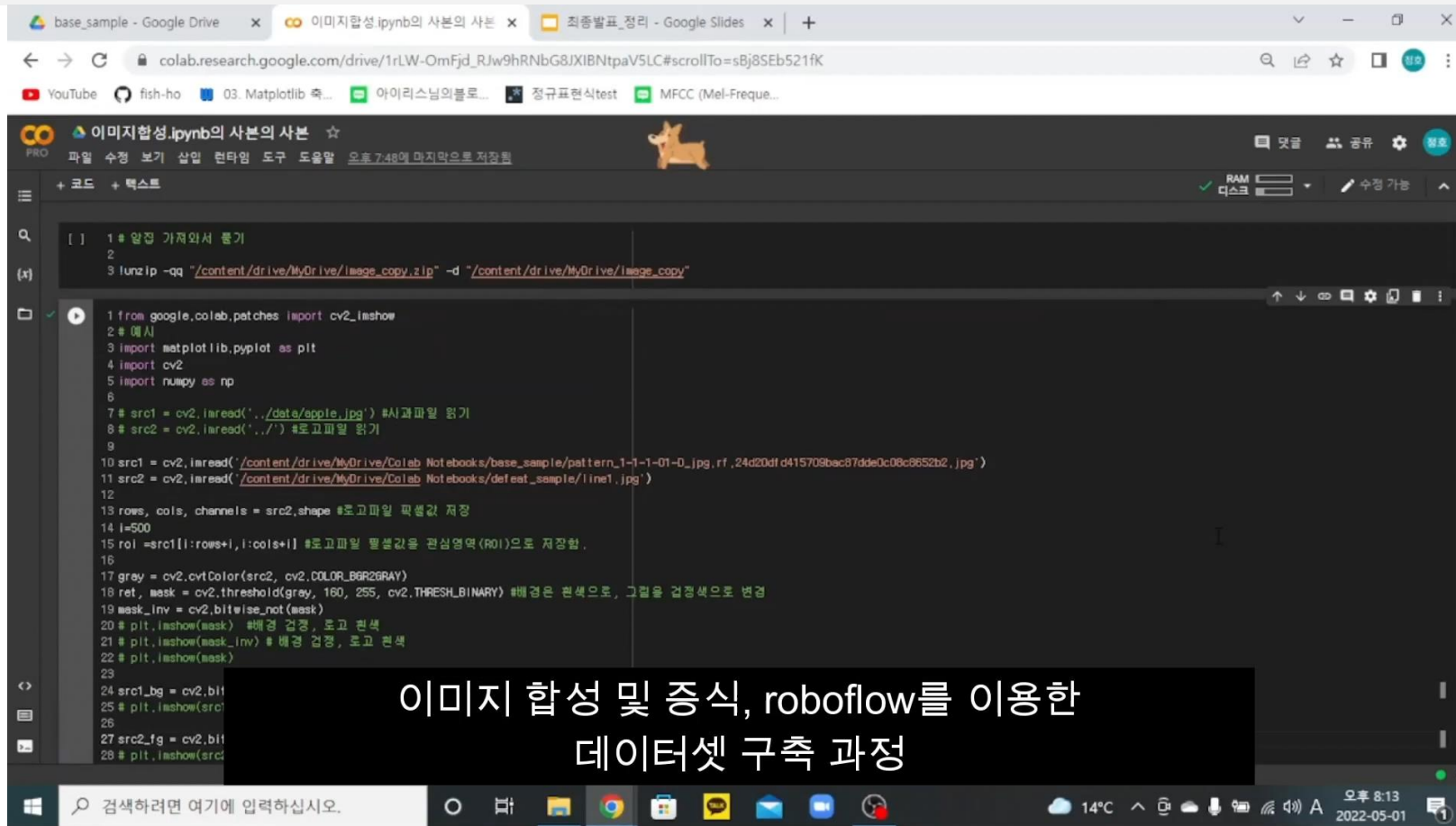
원본 데이터



합성을 통해 새롭게 생성한 데이터

데이터 가공 — 데이터 증식

OpenCV 와 roboflow 를 이용한 데이터 증식 방법



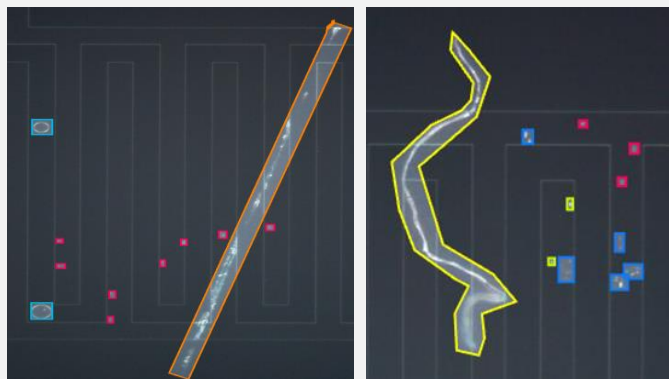
```

1 # 알집 가져와서 풀기
2
3 !unzip -qq "/content/drive/MyDrive/image_copy.zip" -d "/content/drive/MyDrive/image_copy"

4
5
6
7 from google.colab.patches import cv2_imshow
8 # 예시
9 import matplotlib.pyplot as plt
10 import cv2
11 import numpy as np
12
13 # src1 = cv2.imread('../data/apple.jpg') # 사과파일 읽기
14 # src2 = cv2.imread('../') # 로고파일 읽기
15
16 src1 = cv2.imread('/content/drive/MyDrive/Colab Notebooks/base_sample/pattern_1-1-01-0.jpg', cv2.IMREAD_COLOR)
17 src2 = cv2.imread('/content/drive/MyDrive/Colab Notebooks/defeat_sample/line1.jpg')
18
19 rows, cols, channels = src2.shape # 로고파일 픽셀값 저장
20 l=500
21 roi = src1[l:l+rows, l:l+cols] # 로고파일 픽셀값을 관심영역(ROI)으로 저장함.
22
23 gray = cv2.cvtColor(src2, cv2.COLOR_BGR2GRAY)
24 ret, mask = cv2.threshold(gray, 180, 255, cv2.THRESH_BINARY) # 배경은 흰색으로, 그릴을 검정색으로 변경
25 mask_inv = cv2.bitwise_not(mask)
26 # plt.imshow(mask) # 배경 검정, 로고 흰색
27 # plt.imshow(mask_inv) # 배경 검정, 로고 흰색
28 # plt.imshow(mask)
29
30 src1_bg = cv2.bitwise_and(src1, src1, mask=mask_inv)
31 # plt.imshow(src1_bg)
32
33 src2_fg = cv2.bitwise_and(src2, src2, mask=mask)
34 # plt.imshow(src2_fg)
  
```

이미지 합성 및 증식, roboflow를 이용한
데이터셋 구축 과정

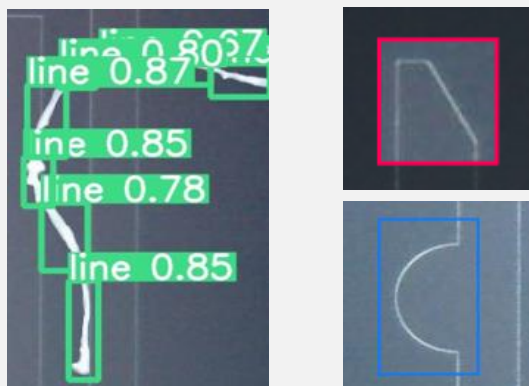
데이터 가공 — 라벨링



1차 라벨링

- 단순 이물 결함 어노테이션 세분화
- 비정형 결함은 폴리곤 처리

➔ 지나친 세분화로 검출 결과 혼재
결함의 특징 학습 못함

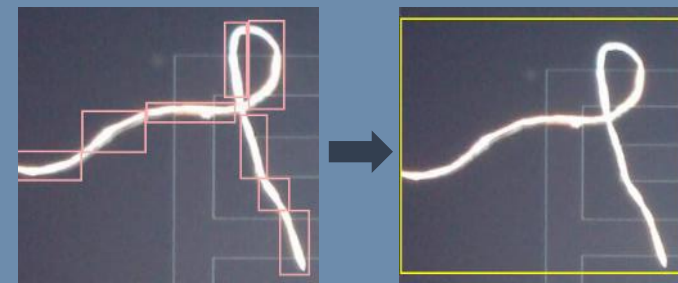


2차 라벨링

- 단순 이물 결함 4종으로 통합
- 회로 결함 8종 추가
- 긴 결함은 나눠서 바운딩 박스 처리

➔ 객체 자체 검출 능력 ↑
바운딩 박스 개수 차이로 인한 mAP 하락

<최종 데이터셋>



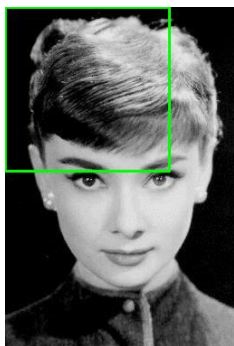
3차 라벨링

- 비정형 결함은 하나로 묶어서 라벨링
- dot와 spot은 영역을 좁게 라벨링
- 부족한 객체는 합성으로 증식

➔ 비슷한 객체 간 구분 능력 ↑
mAP@0.5 **0.96** 달성

- VJ Det.
- HOG Det....

Traditional Detection



One-Stage Detector

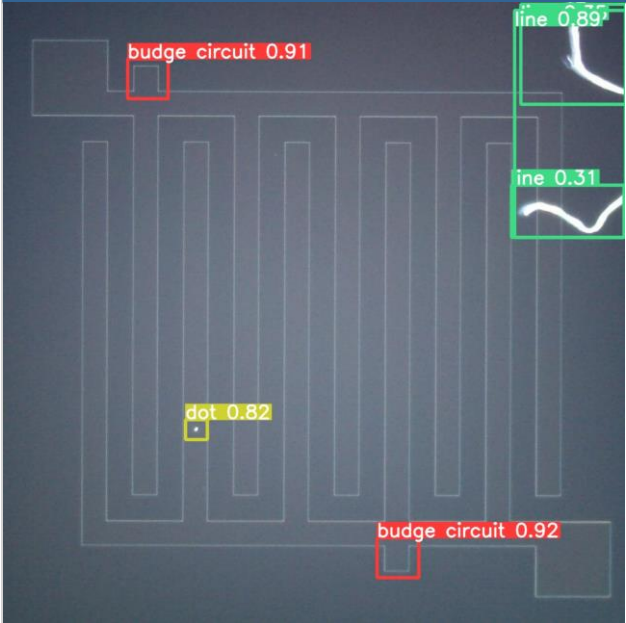
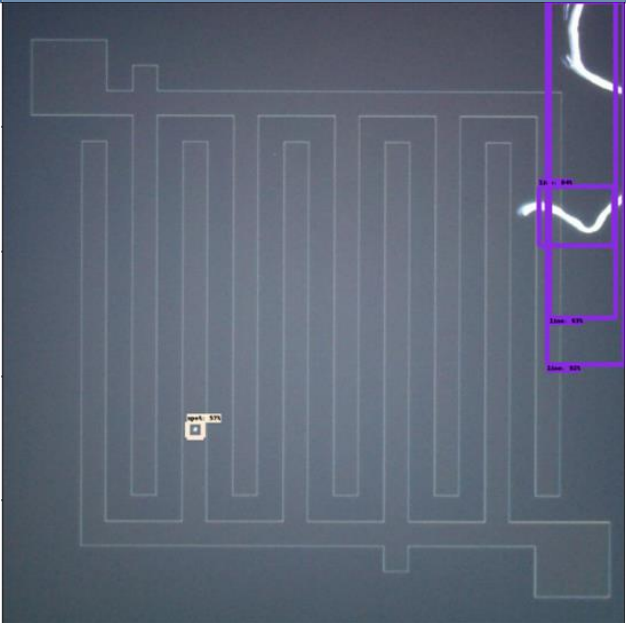
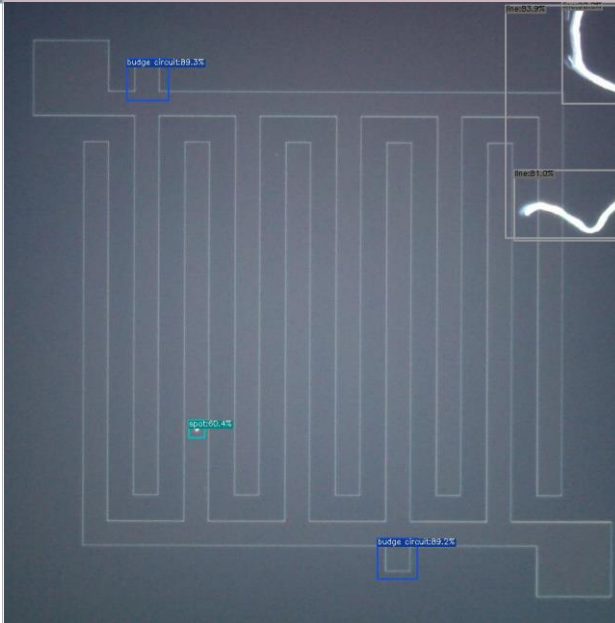
- SSD (2016)
- RetinaNet (2017)
- YOLO v5 (2020) ★
- YOLO X (2021) ★

Deep Learning Based Detection

Two-Stage Detector

- R-CNN (2014)
- Fast R-CNN (2015)
- Faster R-CNN (2016) ★
- Mask R-CNN (2017)

모델링

★ YOLO v5	Faster R-CNN	YOLO X
		
1시간 이내	5시간	1시간 이내
mAP@0.5 0.967	mAP@0.5 0.325	mAP@0.5 0.624
흐리거나 미세한 객체에 대한 검출율 ↓	긴 학습시간 보유한 데이터에 대한 오검출율 ↑ 모델의 편의성 ↓	신생 모델 → 추가적 연구 필요 성능 지표 별 시각화 기능 미흡

모델링

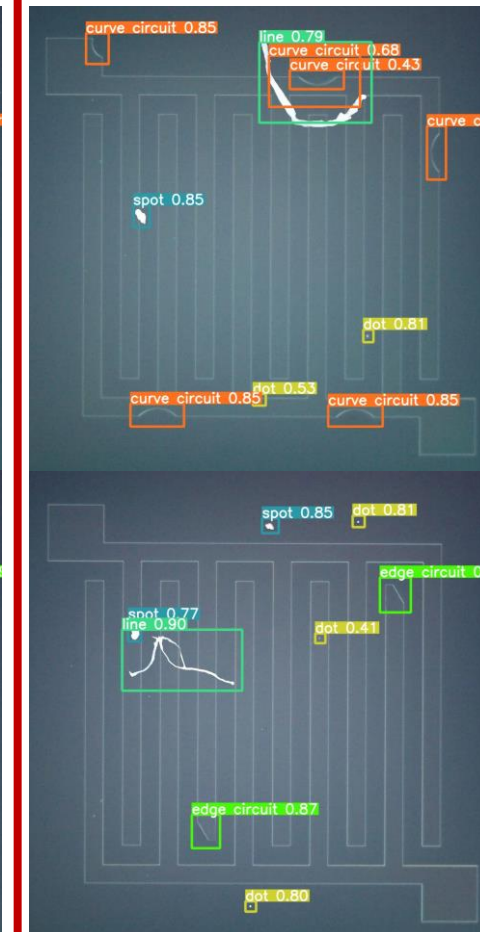
YOLOv5 Small	mAP@0.5(비증식)	0.964	→
	mAP@0.5(증식)	0.962	
	소요 시간	1시간 이내	
YOLOv5 Medium	mAP@0.5(비증식)	0.96	→
	mAP@0.5(증식)	0.97	
	소요 시간	약 1시간 30분	
YOLOv5 Large	mAP@0.5(비증식)	0.964	→
	mAP@0.5(증식)	0.972	
	소요 시간	약 2시간	



학습 이미지 640px
mAP@0.5 0.972



학습 이미지 1,024px
mAP@0.5 0.964



※ 비증식 Train Data 200장 / 증식 Train Data 600장

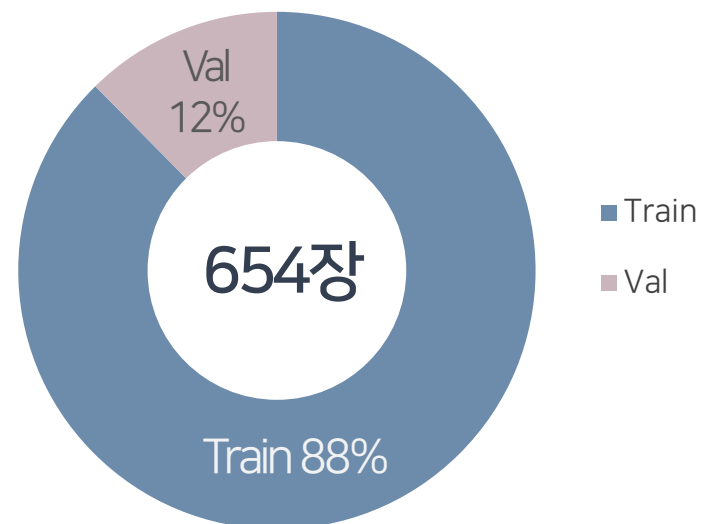
최종 모델 파라미터

- YOLO v5 L
- Train Image Size: **1,024 px**
- Batch Size : **10**
- Epochs: **200**

—	budge circuit 0.995
—	circle 0.995
—	curve circuit 0.995
—	cut circuit 0.995
—	dot 0.820
—	edge circuit 0.995
—	fat circuit 0.995
—	line 0.896
—	scratch 0.933
—	short circuit 0.995
—	spot 0.956
—	thin circuit 0.995
—	all classes 0.964 mAP@0.5

최종 데이터셋

- 원본 데이터 280장
- 합성 데이터 40장
- Train : Validation = 70 : 30
- Train Data 202장약 3배 증식 → 573장



실제 환경 구현 —양품 90%, 결함품 10%

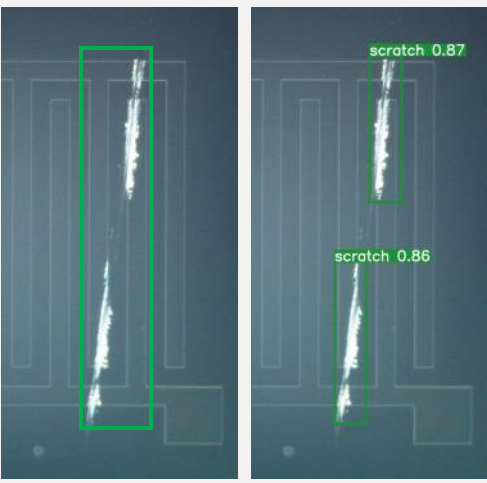


<예측>

<실제>

- budge circuit 0.995
- circle 0.906
- curve circuit 0.995
- cut circuit 0.995
- dot 0.026
- edge circuit 0.995
- fat circuit 0.995
- line 0.948
- scratch 0.000
- short circuit 0.995
- spot 0.641
- thin circuit 0.995
- all classes 0.790 mAP@0.5

<실제리벨링과예측 결과>



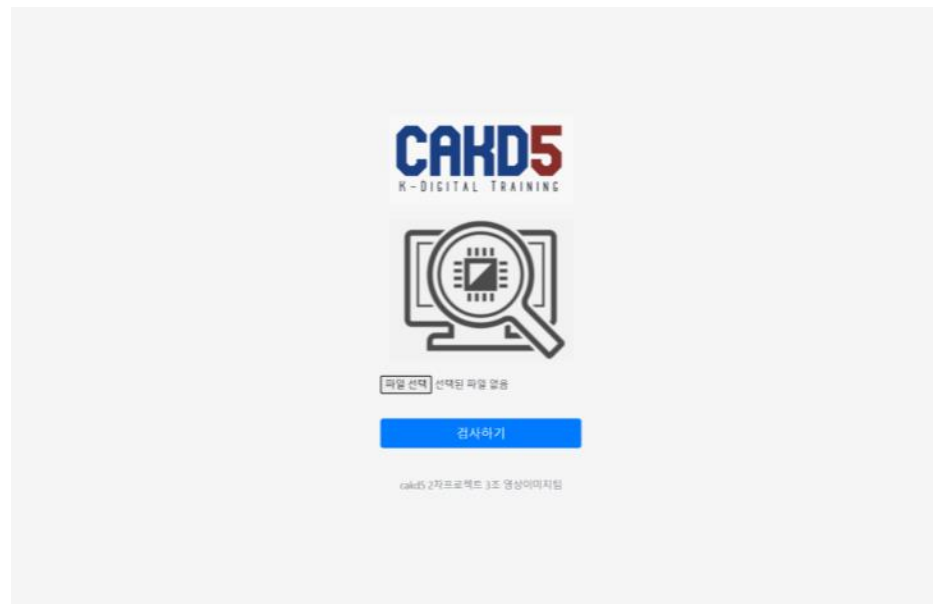
IoU 임계 수준을 넘지 못한 일부 결함 mAP가 하락했으나
결함품 이미지내 12가지 객체 탐지 모두 성공

Part 3 Web Application 구현



Web 구현

메인 화면



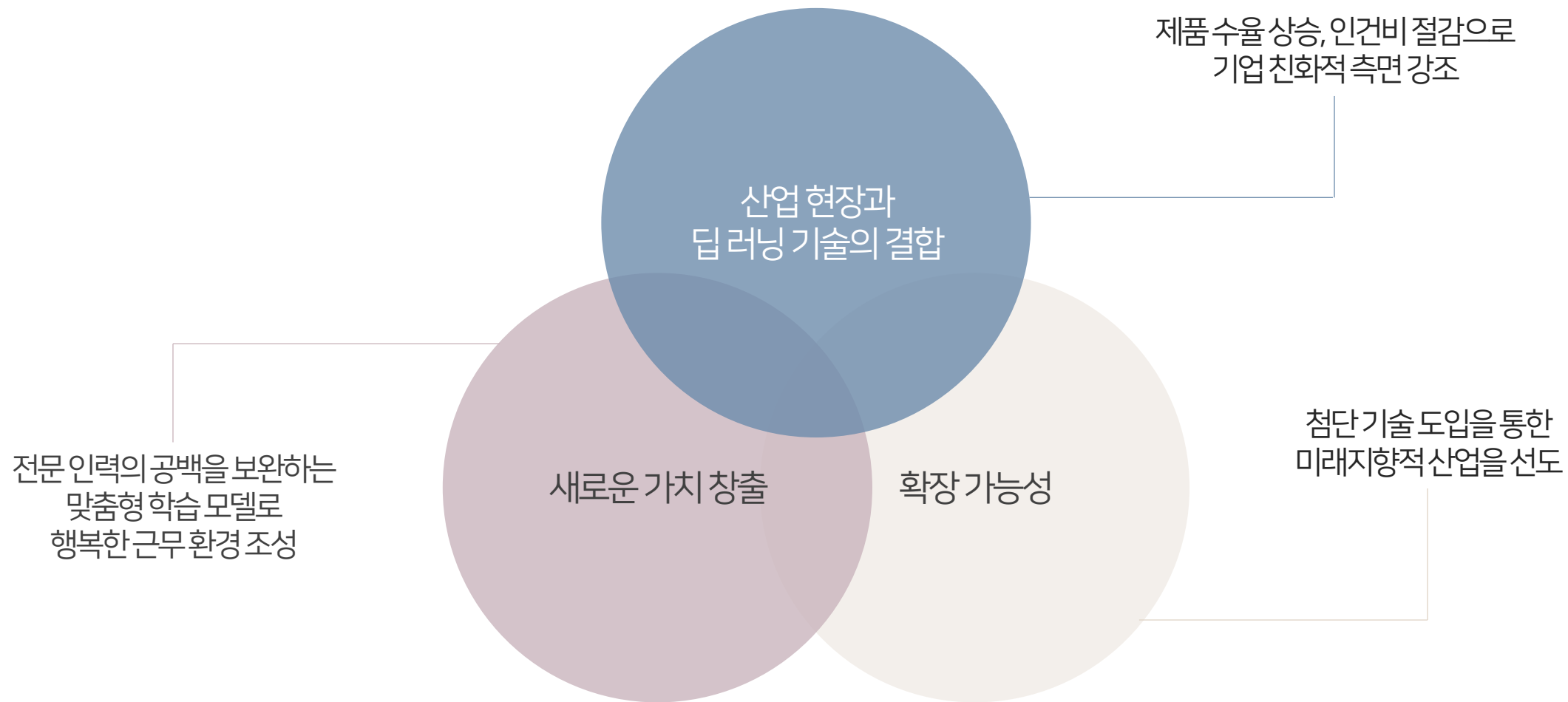
실제사용 화면



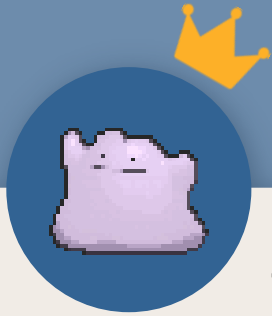
Part 4 프로젝트 정리 및 소감



프로젝트 정리



#2차_프로젝트 #영상이미지 #YOLO #참여_소감



용지영 | 기획총괄

이미지데이터를 사용해본 적은
처음이라 많이 생소한 작업이었지만
그만큼 이번 프로젝트에서
많이 배울 수 있어서 뿌듯합니다.



김재경 | Web구현

다양한 객체 탐지 모델과
웹 페이지 구성의 전체적인 짜임새를
배울 수 있었고, 팀 플레이의 중요성을
다시 한번 깨달았습니다.



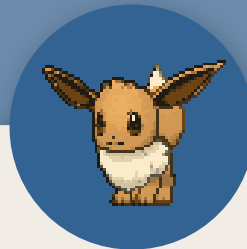
어정호 | 모델최적화

관심 있던 object detection과
이미지 데이터에 대해 공부할 수 있어
유익한 시간이었고, 컴퓨터 비전 분야에
대해 더 흥미가 생기게 되었습니다.



이재우 | 모델분석

이미지에서 객체를 탐지하기 위해
설계된 딥러닝 모델의 동작 방식과
우리의 목적에 부합한 모델을 채택하는
과정을 배울 수 있어 좋았습니다.



정현우 | 데이터정제

객체 탐지의 전체적인 흐름을
알 수 있는 좋은 기회였고,
특히 ROBOFLOW의 빠르고 편리함이
인상 깊었습니다.



최지원 | 모델링 및 발표

딥러닝 기술이 기존 산업과 만나
어떻게 시너지 효과를 일으키는지
그 가능성을 직접 확인할 수 있었던
뜻깊은 경험이었습니다.

감사합니다

