불량품 검별 Model

Model powered by SUPERB AI

C반 2조
김교희, 박상욱, 이건도

Table of contents

Part 1

회사 소개 및 프로젝트 목표

Part 1

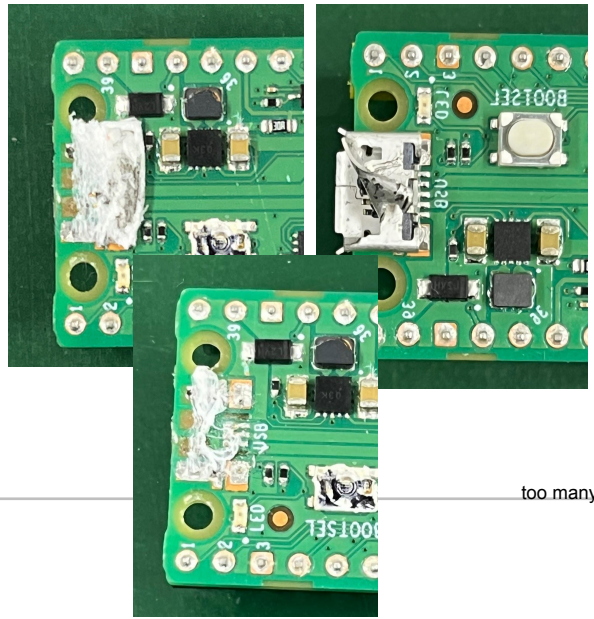
목표

1. 고객의 생산 시스템에 맞는 불량품 감지 모델 생성
2. 모니터링 작업 인력 최적화
3. 시스템 상용화

Part 1

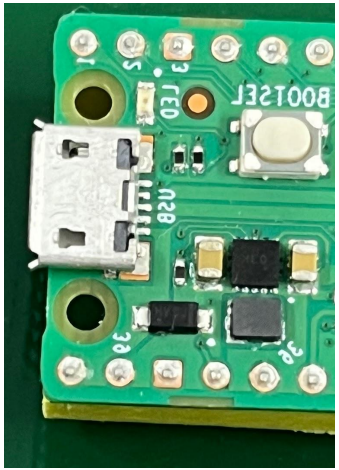
모델 생성 방향

불량품



too many case..

양품




QUESTION
성능 좋은 모델 이란?

Part 1

좋은 Model의 조건?

FN

:양품을 불량품으로 측정

- 고가의 부품 or 희소한 자원
- 생산 속도가 중요한 경우

FP

:불량품을 양품으로 측정

- 불량품이 고객에게 미치는 영향이 크거나 치명적인 경우
- 브랜드 이미지와 신뢰도가 중요한 경우
- 안전 문제

TP

: 올바른 양품 확인

TF

:올바른 불량품 감지

Rasberry Pi 제품 특성

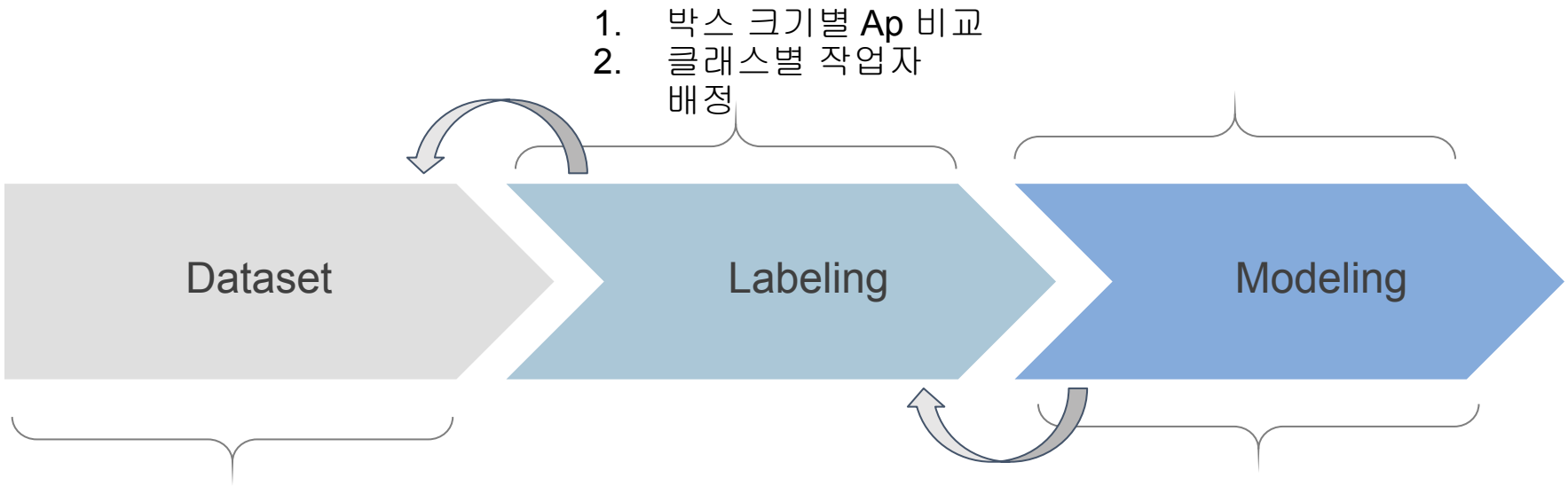
양품 기준

기준	내용
class	hole 제외 5개
Confidence	• 각 Class별로 Confidence threshold 다르게 설정 • hole- 0.5 이상, 다른 class- 0.7 이상

threshold
• 높은 Threshold → (양품)탐지 민감도 증가 → Precission 증가, FN 증가 위험 • 낮은 Threshold → (양품)탐지 신뢰도 증가 → Recall 증가, FP 증가 위험.

Part 3

모델 생성 과정



Part 3

dataset

+16%

mAP(%) @0.5:0.95	mAP(%) @0.5
53.0	77.4

Name	AP(%) @0.5:0.95	AP(%) @0.5
CHIPSET	62.3	79.0
Hole	⚠ 33.3	69.6
RASPBERRY PICO	85.0	96.0
USB	⚠ 50.0	71.7
bootsel_narrowly	58.6	88.3
oscillator	⚠ 28.7	59.6

데이터 수:
285개

mAP(%) @0.5:0.95	mAP(%) @0.5
62.3	89.1

Name	AP(%) @0.5:0.95	AP(%) @0.5
CHIPSET	72.2	90.5
Hole	⚠ 38.9	81.9
RASPBERRY PICO	81.7	94.1
USB	65.9	90.0
bootsel_narrowly	60.6	88.0
oscillator	54.3	90.2

350개(+75개, ↑22%)

Part 3

dataset

저화질

AP(%) @0.5:0.95	AP(%) @0.5	Precision @0.5	Recall @0.5
▲ 28.0	72.1	0.98	0.64

+41%

+12%

+ 중화질

AP(%) @0.5:0.95	AP(%) @0.5	Precision @0.5	Recall @0.5
▲ 39.7	81.3	0.85	0.77

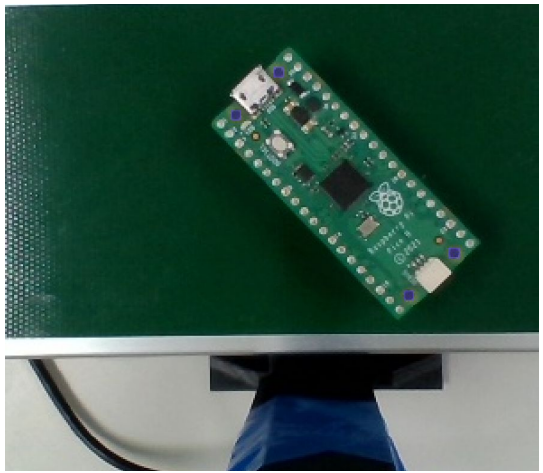
+ 고화질

▲ 29.5	64.6	0.78	0.61

Part 3

dataset

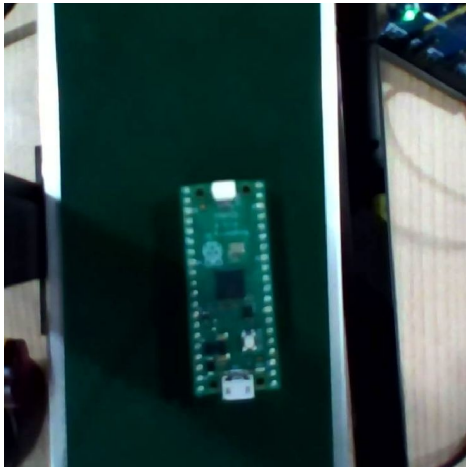
- Type A. 저화질 (10~30kb)



- 실제 촬영 환경
- 픽셀수 적음
- 선명해 보임

5 : 1 비율
(총 350장)

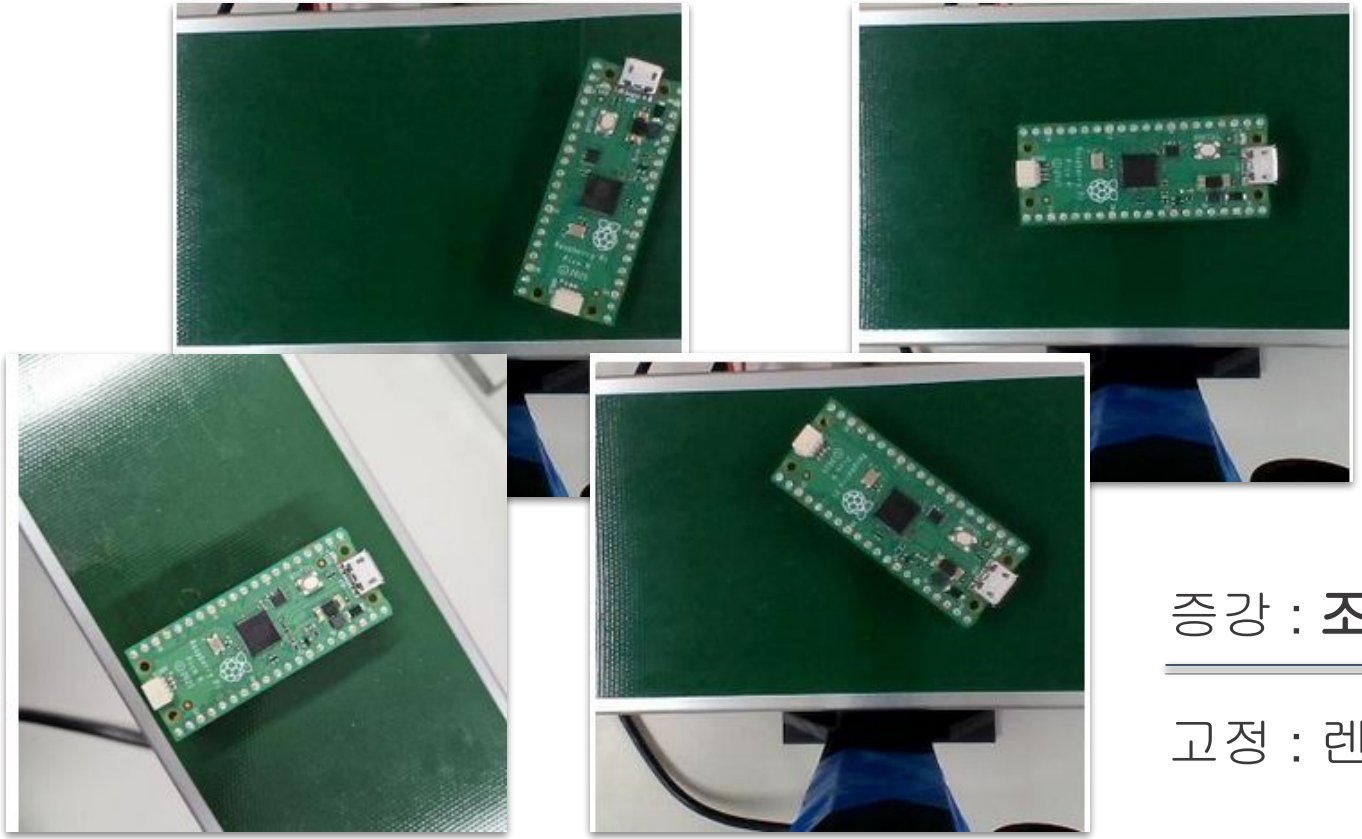
- Type B. 중화질 (40~80kb)



- 픽셀수 상대적 많음
- 촬영시 흔들림 있음

Part 3

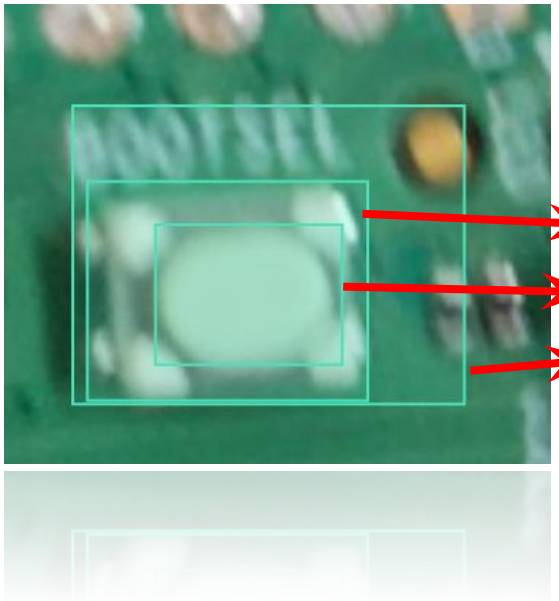
dataset



증강 : **조명, 방향**

고정 : 렌즈~object거리

박스 구역에 따른 AP 차이

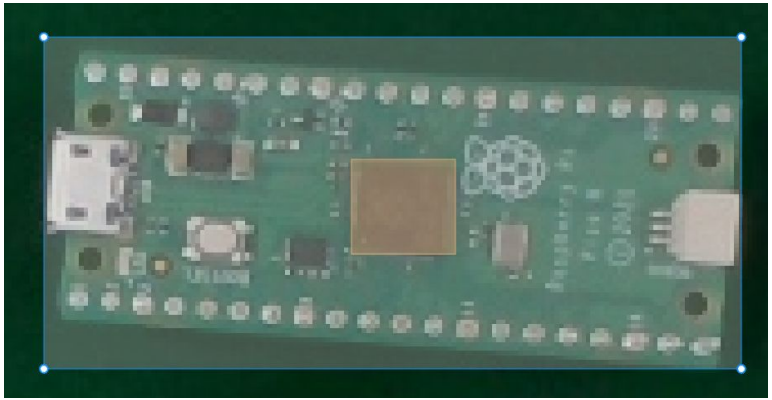


Class name	Model performance AP(%)	Precision	Recall
bootsel_narrowly	33.6	0.89	0.57
bootsel_right	0.0	0.00	0.00
bootsel_wide	8.4	0.32	0.33

Part 3

dataset **labeling**

- Rasberry pico



- pico의 네 꼭짓점에 접하도록 bounding
- pico 의 모든 object가 포함 될 수 있도록 bounding

- Rasberry pico edgecase

- truncation

Part 3

dataset **labeling**

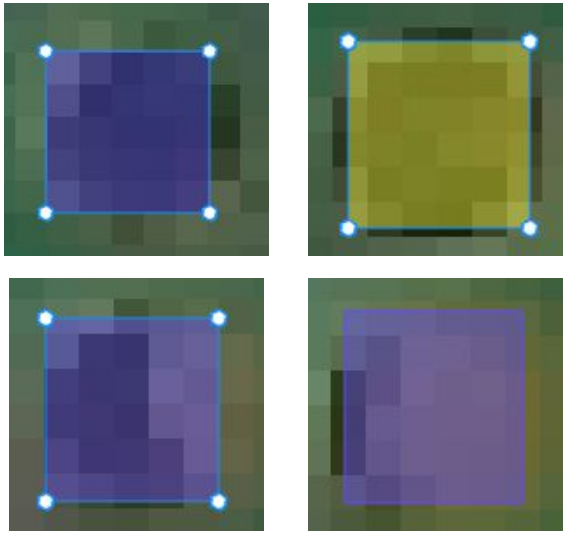
- oscillator

- 납땜자국 포함
- 다각도로 틀어진 data는 네 꼭짓점에 접하게 bounding

- oscillator_Edgecase

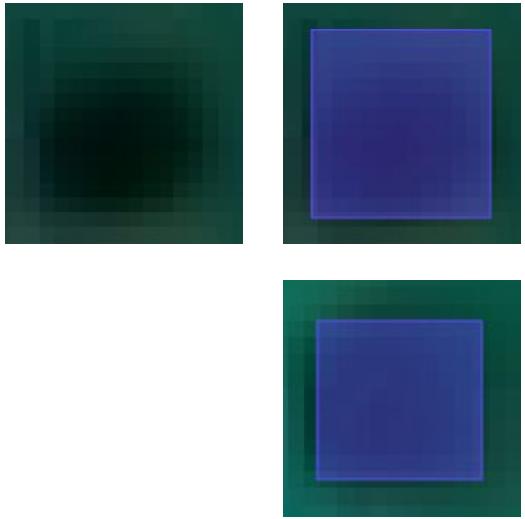
- 금속의 빛번짐, 흐릿해 경계가 모호한 경우 데이터는 전체 데이터의 약 15%만 수집

- hole_저화질



- 저화질 약 5*5 pixel ~ 6*6pixel
- 갈색이 많아도 픽셀 기준으로 동일

- hole_중화질

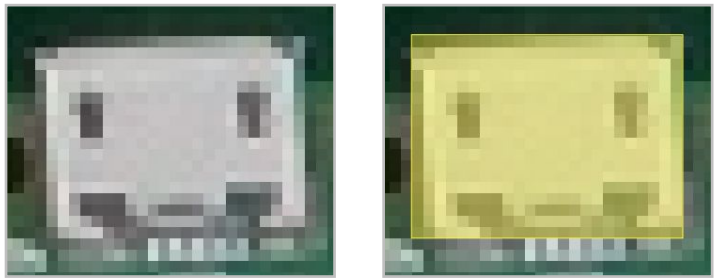


- 최외각에서 1,2칸 작게 설정

Part 3

dataset labeling

- USB_저화질



- 위에 핀 모양 제외
- 위쪽은 꺾은선 있는 부분까지

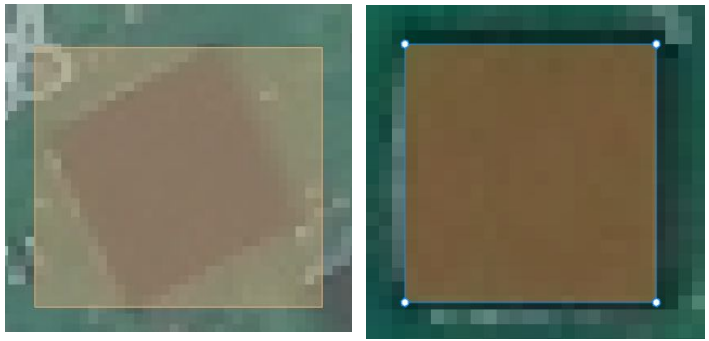
- USB_중화질

- 위에 핀 모양 제외
- 다리 꺾여 있는 다리까지만 박스

Part 3

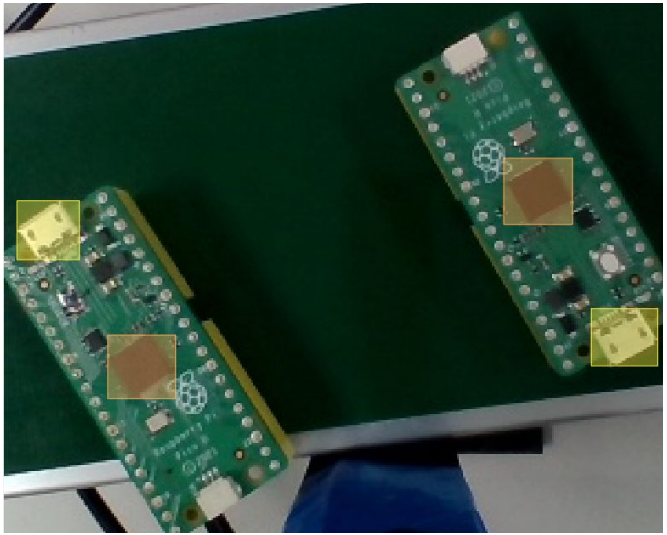
dataset **labeling**

- chipset



- 납땜 부분 제외 검은 사각형 소자만 포함하도록 bounding
- 사각형 꼭짓점에 box가 접하도록 함

- chipset_Edgecase

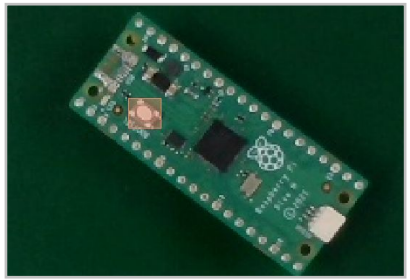


- 한 화면 내 chipset이 2개 나오면 2개 다 라벨링
- 빛번짐 , 구분이 불가능 할 때 라벨링 X

Part 3

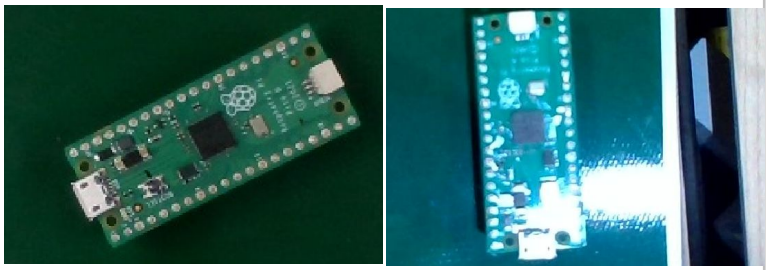
dataset labeling

- bootsel



- 납땜 자국 포함

- EdgeCase



- 부서진 bootsel은 라벨링 X
- 빛번짐이 너무 심하여 알아볼 수 없을 때는 라벨링 X

YOLOv6-N

총 데이터수 350
train : 280 / validation : 70

Part 3

최종 Model 분석

mAP(%) @0.5:0.95	62.3	mAP(%) @0.5	89.1	Low AP	Low precision	Low recall	Class name

Name	AP(%) @0.5:0.95	AP(%) @0.5	Precision @0.5	Recall @0.5
CHIPSET	72.2	90.5	0.97	0.87
Hole	⚠ 38.9	81.9	0.92	0.73
RASPBERRY PICO	81.7	94.1	0.99	0.95
USB	65.9	90.0	0.89	0.86
bootsel_narrowly	60.6	88.0	0.93	0.87
oscillator	54.3	90.2	0.90	0.91

Part 3

리스크 관리 계획

Auto Labeling 으로 데이터 증가

Name	AP(%) @0.5:0.95	AP(%) @0.5	Precision @0.5	Recall @0.5
CHIPSET	72.2	90.5	0.97	0.87
USB	65.9	90.0	0.89	0.86

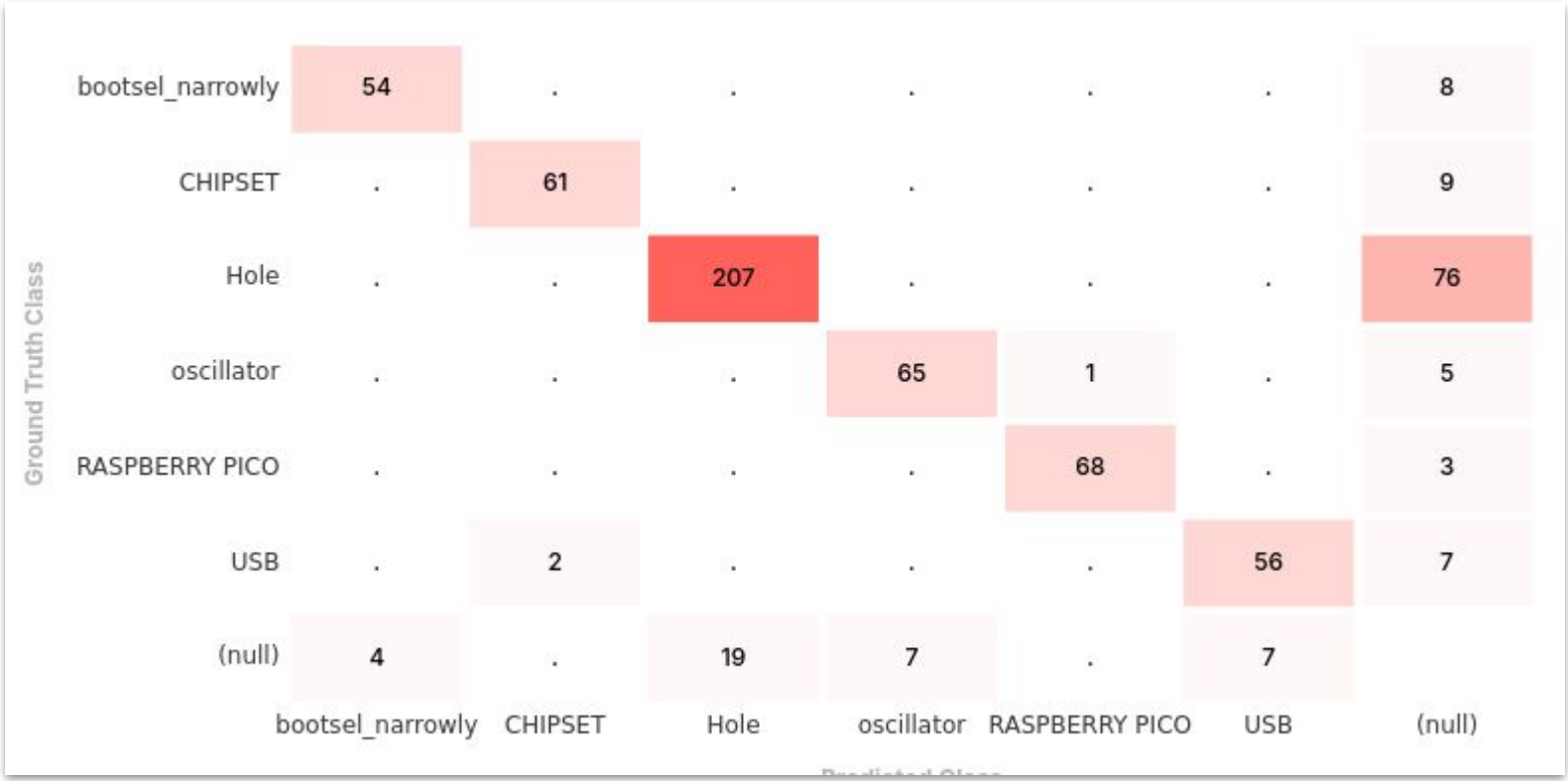
Name	AP(%) @0.5:0.95	AP(%) @0.5	Precision @0.5	Recall @0.5
CHIPSET	74.2	92.6	1.00	0.91
USB	81.0	95.2	1.00	0.91

Part 4

최종 Model 분석

Part 4

최종 Model 분석

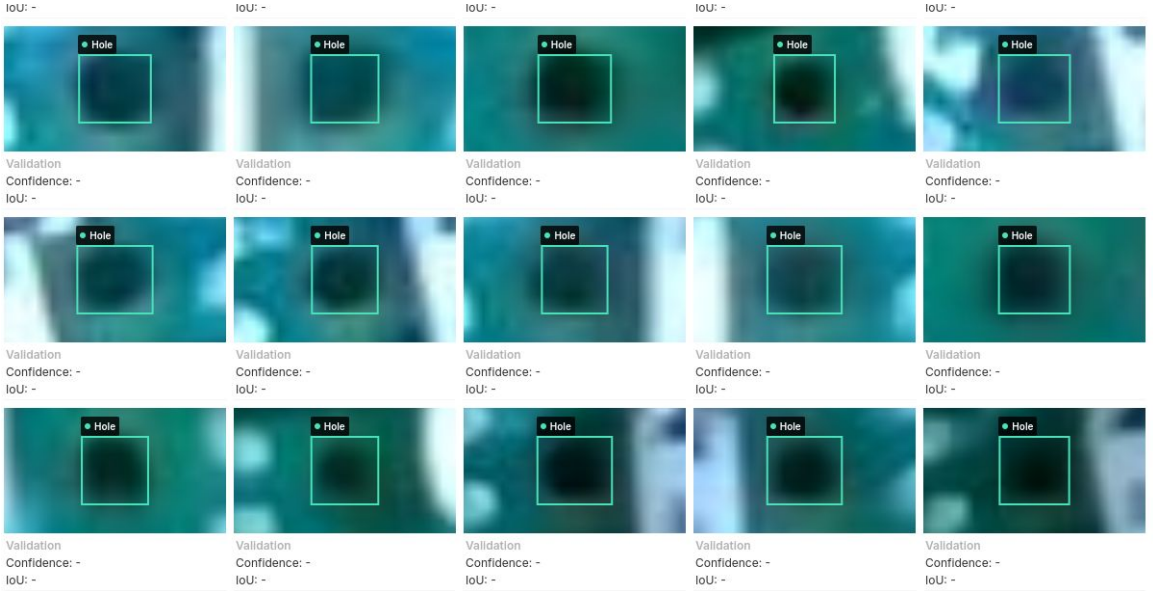


Part 4

최종 Model 분석

hole

모델 받침대

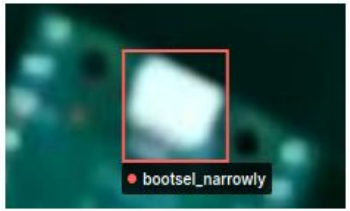


중화질 비율 조절

Part 4

최종 Model 분석

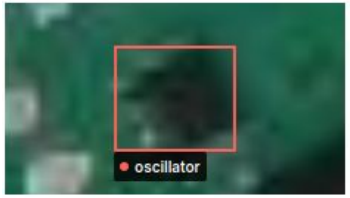
전처리



Validation
Confidence: 0.75
IoU: -

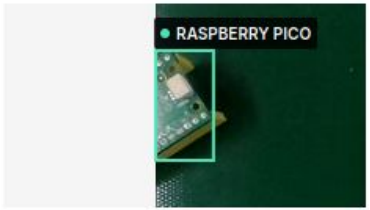
Validation
Confidence: 0.49
IoU: -

학습량

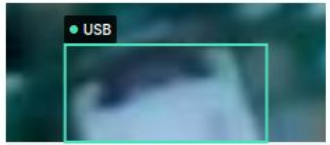


Validation
Confidence: 0.53
IoU: -

엣지 케이스



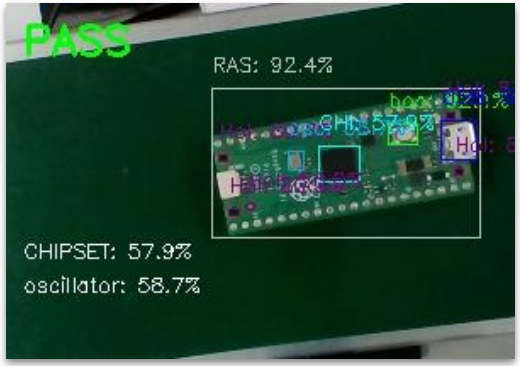
Validation
Confidence: -
IoU: -



Validation
Confidence: -
IoU: -

Part 4

최종 Model 분석




```
=== Detection Results ===
          Class Score (%)
RASPBERRY PICO      92.4%
       bootsel      92.1%
       CHIPSET      57.9%
    oscillator      58.7%
           USB      67.4%
          Hole      86.7%
          Hole      83.3%
          Hole      80.7%
          Hole      80.3%
          Hole      56.0%
Image saved to ./capt/annotated_image.jpg
```

양품 판정기준

```
if len(missing_classes) == 0 and
hole_count >= 3:
    return "PASS"
elif len(missing_classes) >= 2:
    return "FAIL"
else:
    return "Inspection"
```

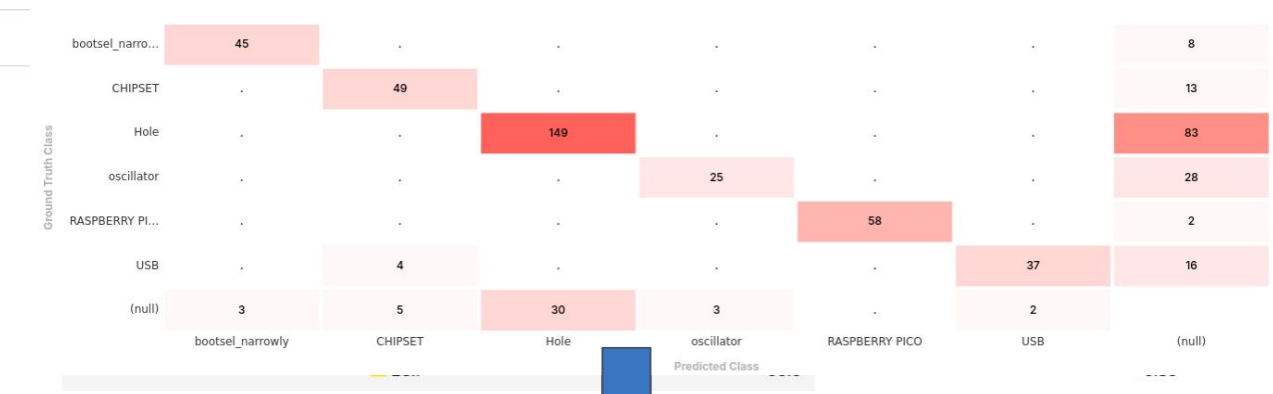
score 0.5 이상만 인정

```
for obj in results['objects']:
        if obj is None or 'class' not in obj
or 'box' not in obj:
            continue
        if obj['score'] <= 0.5:
            continue  # Score가 0.5 이하인 경우
건너뜀
        box = obj['box']
        class_name =
obj['class'].replace("bootsel_narrowly",
"bootsel")  # 클래스 이름 수정
        detected_classes.add(class_name)  #
탐지된 클래스 저장
        score = obj['score']
```

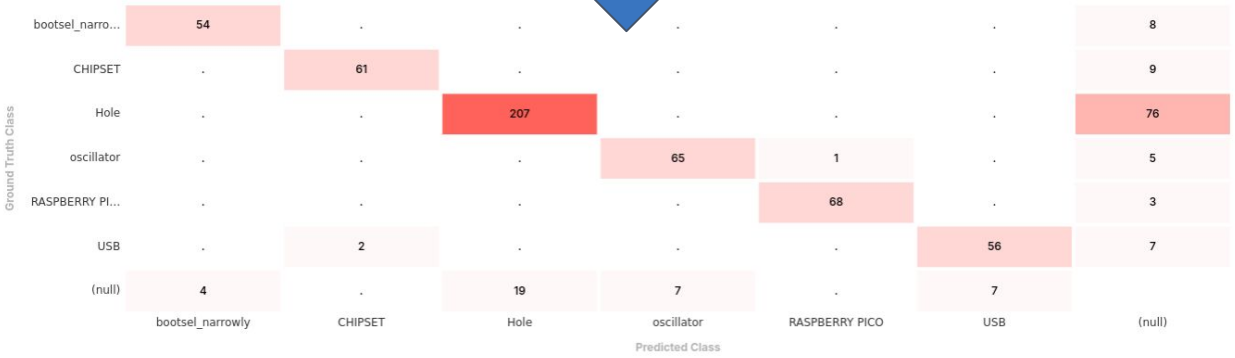
Part 4
기대효과

Part 4
성과-정확도

Name	Recall @0.5
CHIPSET	0.79
Hole	0.64
RASPBERRY PICO	0.96
USB	0.64
bootsel_narrowly	0.84
oscillator	⚠ 0.47



Name	Recall @0.5
CHIPSET	0.87
Hole	0.73
RASPBERRY PICO	0.95
USB	0.86
bootsel_narrowly	0.87
oscillator	0.91



Part 4

기대효과 - 프로젝트 단기적 효과와 장기적 효과

장기 효과

- 신뢰도 및 브랜드 이미지 향상:
  -> 지속적인 품질 관리와 안정적인 제품 공급
- 인건비 절감
  -> 운용 및 개발 비용 증가

단기 효과

- 정확한 불량품 검출: 생산 효율↑, 비용 손실↓
- 작업 흐름 최적화

Part 4

리스크 관리 계획

- edge case의 데이터 부족으로 오탐지 가능성↑
  - 학습량 증가 후 다시 라벨링

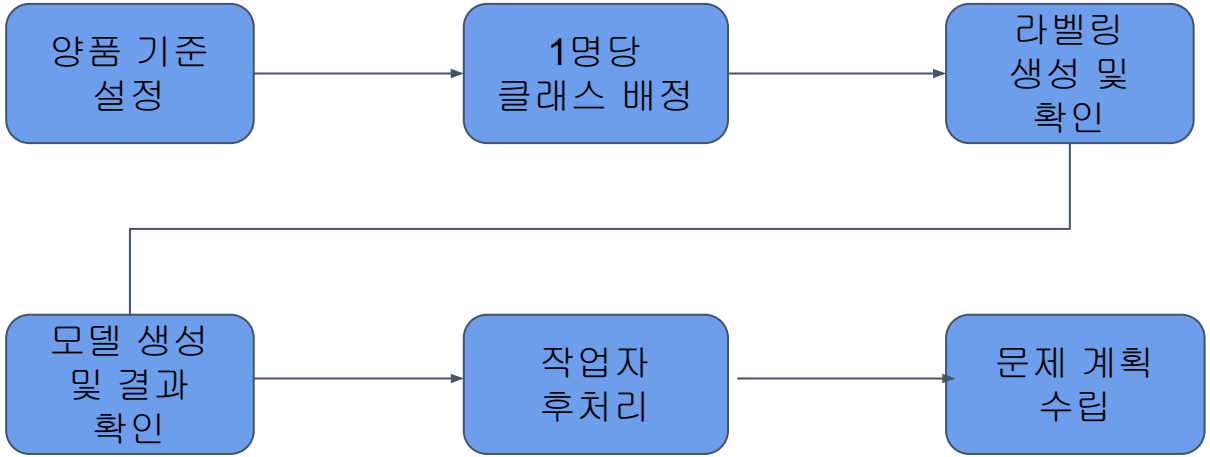
- 안정화 전까지 작업자가 후처리 하는 작업이 필요

Part 5

결론

Part 5

요약 및 결론



Q&A