

2024 전북대학교 공과대학  
산학 실전 캡스톤 프로젝트 본선 발표

< **작물 품질 상태 판별:  
이미지 프로세싱 및 인공지능 기술** >

TEAM: Guardians of Farm & sems games

강경빈, 김세원, 박은송, 유수현

지도교수: 유기형

## CONTENTS

### 1. 서론

- 1-1. 연구 배경
- 1-2. 연구 목적

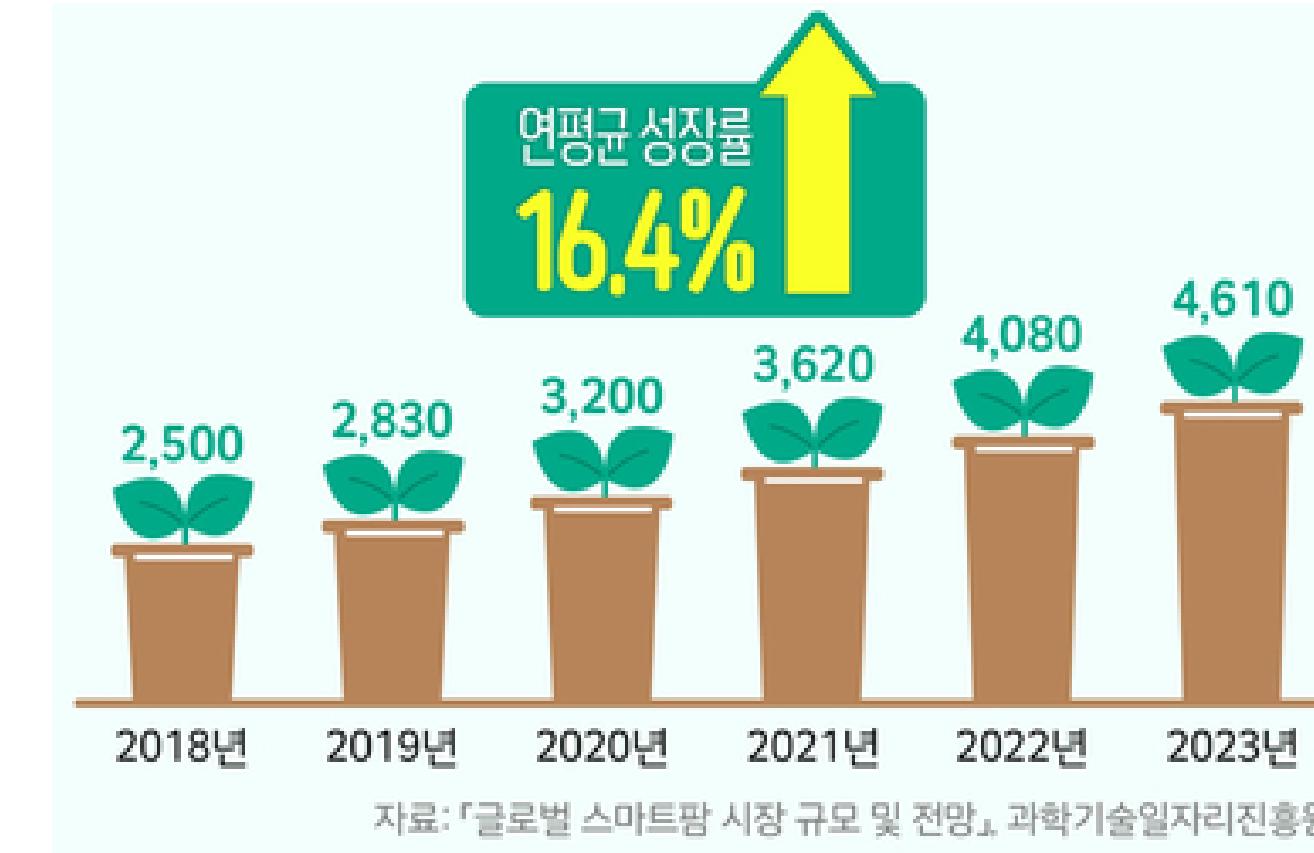
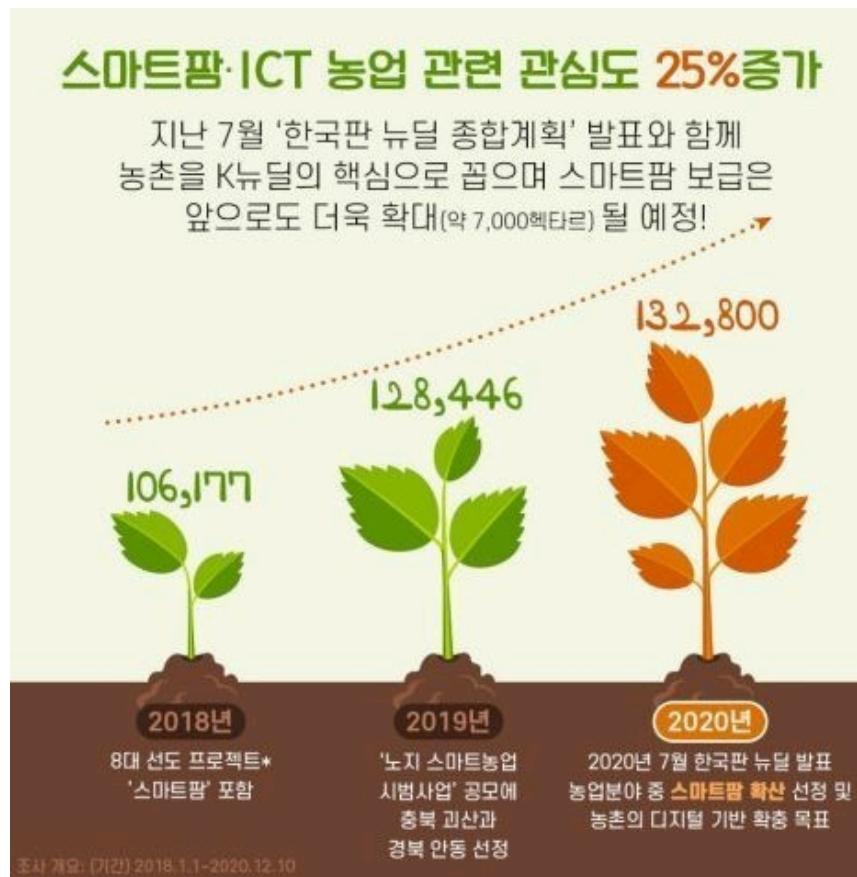
### 2. 본론

- 2-1. 사용 데이터 및 모델
- 2-2. 코드 로직 설계

### 3. 결론

- 3-1. 결과 분석
- 3-2. 정확도 측정
- 3-3. 개선 방향 및 결론

## 1-1. 연구 배경



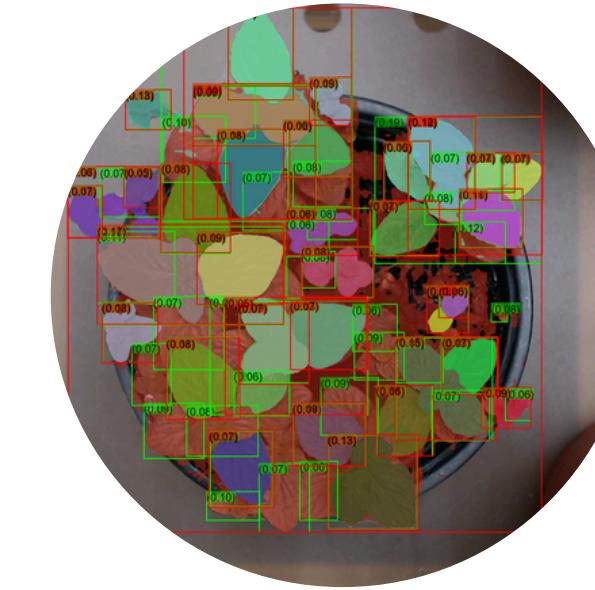
### 식물 질병 탐지의 중요성

- 최근 스마트팜 시장이 확대되며, 작물의 성장 과정에 질병 인식 및 해결의 중요성 증가
- 식물 질병은 작물 수확량, 품질, 수익성에 심각한 영향을 미칠 수 있음.
- 수동으로 식물 질병을 탐지하는 것은 시간이 많이 소요됨.

**작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술**

## 1-2. 연구 목적

작물 질병 탐지는 신속한 대처가 중요하지만  
기존 모델은 고용량으로 느린 속도와 실제 적용이 어려움



스마트팜 활용 앱  
데이터 적용

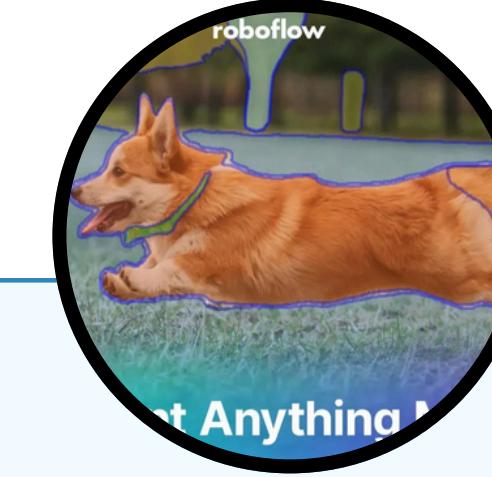
일정 구도 데이터 수집

기존의 모델 활용한  
신속한 질병 탐지

작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

## 2-2. 코드 로직 설계

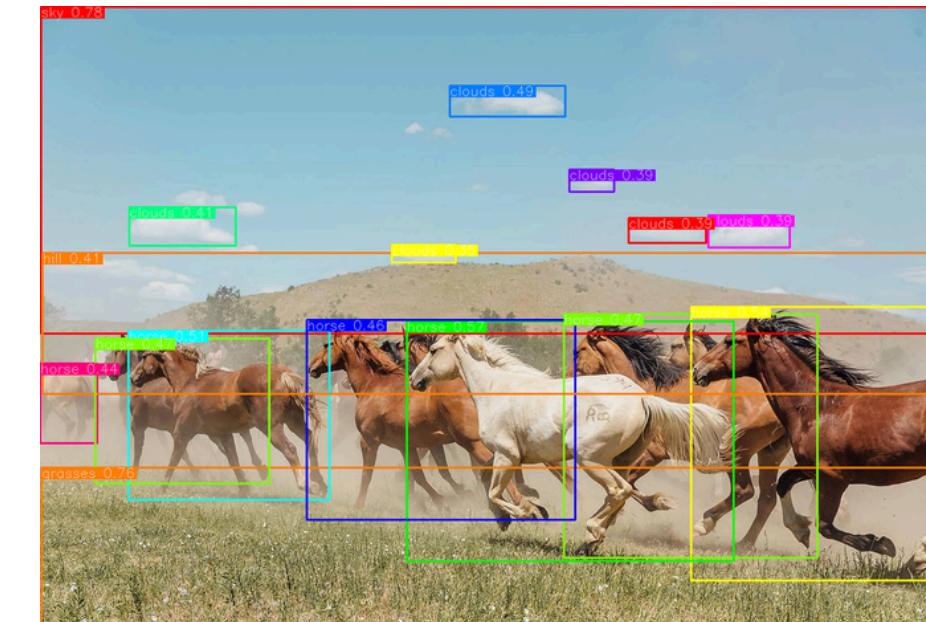
### 코드 개발 초기



#### GroundingDINO, SAM (Segment Anything)

#### 비지도 모델 활용 객체 인식 프로그램 개발

- 프롬프트 입력 텍스트를 활용
- 비지도 학습으로 객체 인식
- 배경과 분리 -> PNG 형식으로 인식
- 정상/비정상 잎과 총 개수를 식별



작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

## 2-2. 코드 로직 설계

```
def filter_abnormal_leaves(image_rgb, boxes, pred_phrases, threshold_factor=2):
    # 모든 박스의 특징값을 수집하여 평균 및 표준편차 계산
    brightness_values = []
    hue_values = []
    saturation_values = []

    # 각 박스의 특징값을 수집
    for box in boxes:
        x0, y0, x1, y1 = map(int, box)
        cropped_region = image_rgb[y0:y1, x0:x1]

        avg_brightness = np.mean(cropped_region)
        brightness_values.append(avg_brightness)

        cropped_hsv = cv2.cvtColor(cropped_region, cv2.COLOR_RGB2HSV)
        avg_hue = np.mean(cropped_hsv[:, :, 0])
        hue_values.append(avg_hue)

        avg_saturation = np.mean(cropped_hsv[:, :, 1])
        saturation_values.append(avg_saturation)

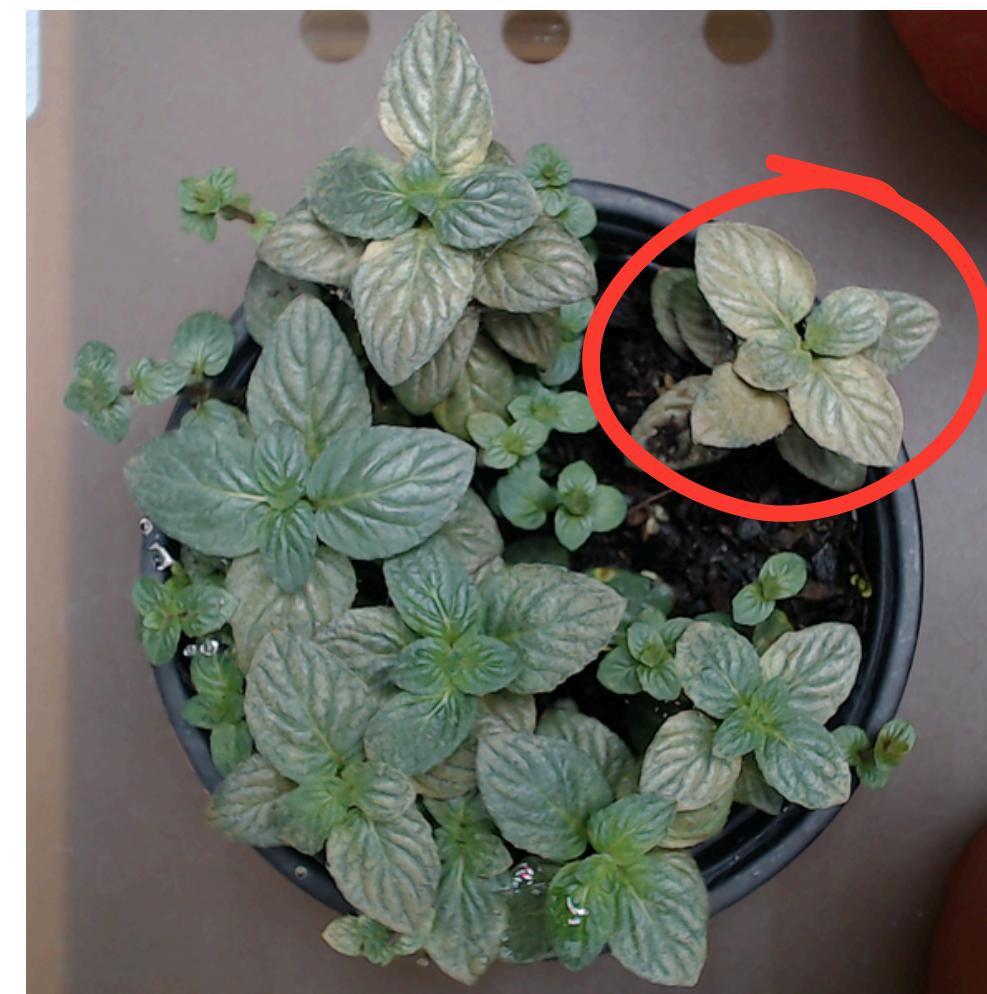
    # 각 박스가 평균에서 얼마나 벗어나는지 계산하고 임계값 이상이면 병든 잎으로 분류
    for box, phrase, avg_brightness, avg_hue, avg_saturation in zip(
        boxes, pred_phrases, brightness_values, hue_values, saturation_values
    ):
        if (
            abs(avg_brightness - brightness_mean) > threshold_factor * brightness_std or
            abs(avg_hue - hue_mean) > threshold_factor * hue_std or
            abs(avg_saturation - saturation_mean) > threshold_factor * saturation_std
        ):
            diseased_boxes.append(box)
            diseased_phrases.append("Too much Water: " + phrase)
        else:
            normal_boxes.append(box)
            normal_phrases.append("Chocolate Mint: " + phrase)

    return normal_boxes, normal_phrases, diseased_boxes, diseased_phrases
```

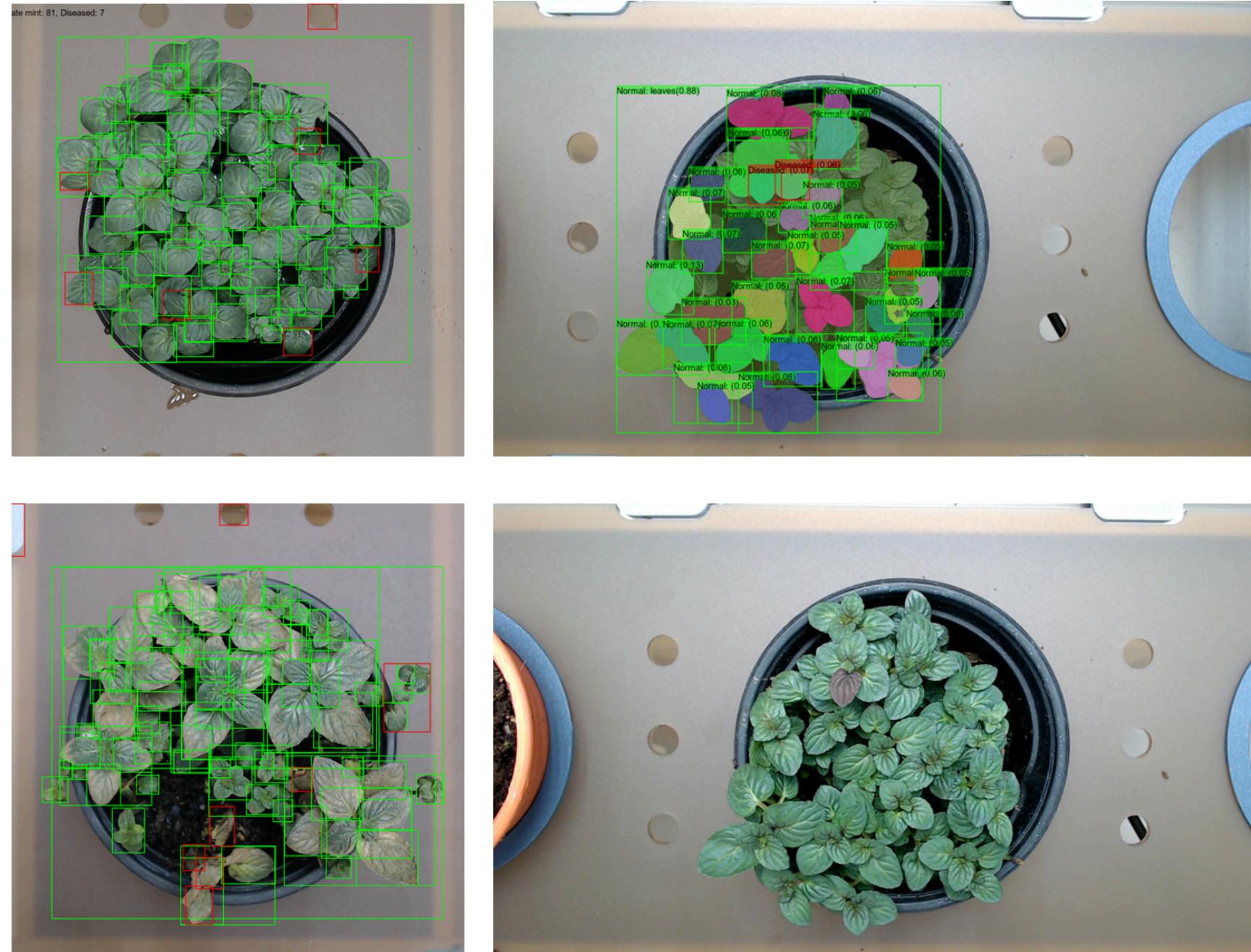
박스 영역을 밝기와 색상 기준으로 분류해 비정상 잎을 식별

기본 로직:

- 잎 객체로 인식된 박스에서 RGB 평균값 계산
- 비정상 잎은 평균보다 밝기/색상 편차가 큼



## 2-2. 코드 로직 설계

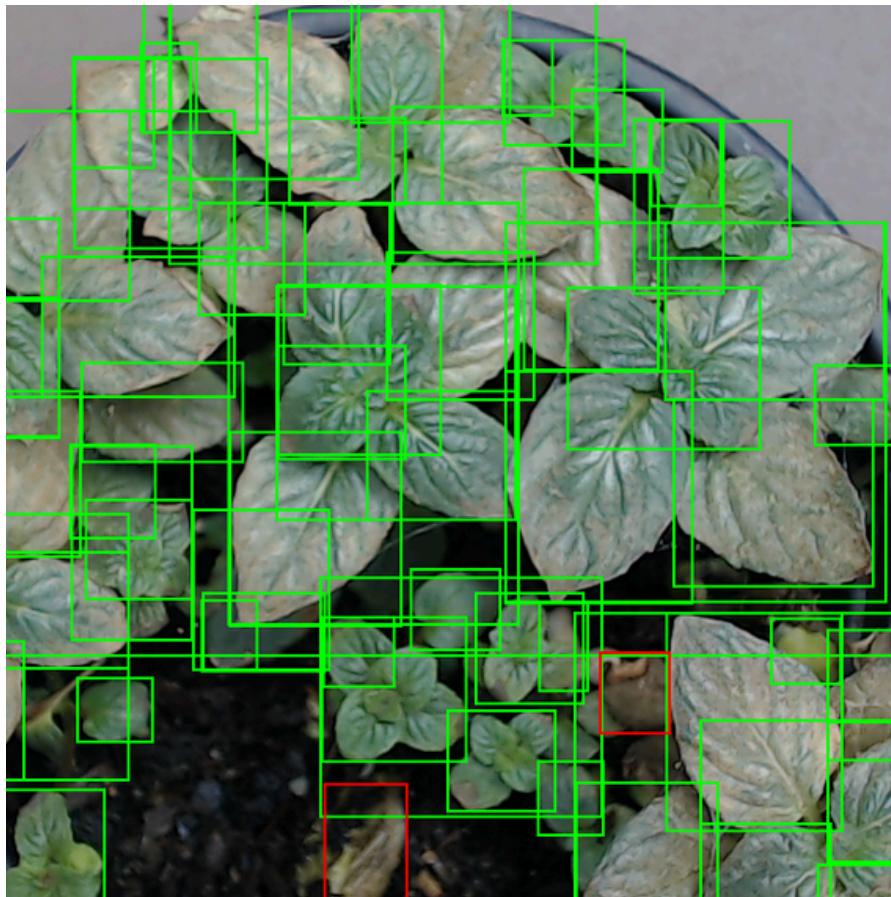


• 실제 활용 사진

작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

## 코드 로직 설계: 토기 코드의 문제점과 코드 변경

잎들의 평균 RGB값을 구해 평균에서 벗어나는 잎을 비정상 잎으로 설정



- 장점: 데이터셋 구축 지도학습 없이 색상 편차만으로 분류 가능
- 다양한 촬영 환경 변동에서도 비정상 잎 구별 가능

문제점 : **비정상 잎이 정상 잎보다 많을 때**  
정상 잎이 비정상 잎으로 분류되는 오류 발생

<- 황변된 비정상 잎들임에도 불구하고  
비정상 잎이 정상 잎보다 많기 때문에 정상 잎으로 분류됨

작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

## 2-2. 코드 로직 설계

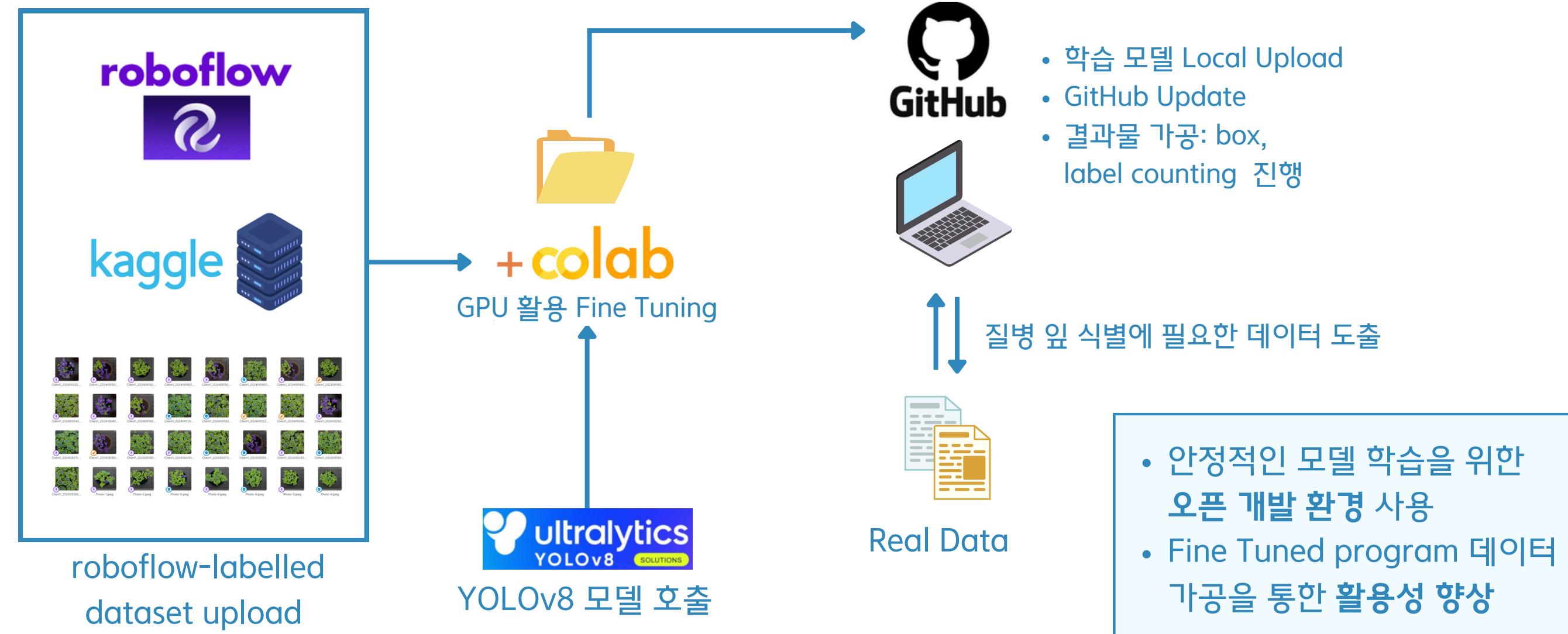
# 활용 모델 및 코드 로직 변경

	이전 모델	최종 모델
활용 모델	비지도 모델 (groundingDINO, SAM)	지도 모델 (YOLOv8)
분류 로직	<ul style="list-style-type: none"><li>비정상 잎은 정상 잎과 밝기/색상 편차 많이 발생하는 점에서 착안</li><li>잎 객체 박스에서 RGB 평균값 계산</li></ul>	<ul style="list-style-type: none"><li>데이터셋을 확장해 정확도 향상</li><li>RoboFlow로 라벨링 진행</li><li>데이터셋 증강</li><li>경량 모델 학습</li><li>다양한 비정상 잎 종류 활용</li></ul>
데이터셋	기업체 (sems games) 제공 초코민트 허브 데이터	기존 데이터 + Kaggle Open Source + 실생활 수집 데이터

작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

## 2-1. 사용 데이터 및 모델

# 程式 구성도



작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

## 2-1. 사용 데이터 및 모델

# 마그넷 구성 - 데이터셋



기존 보유 데이터

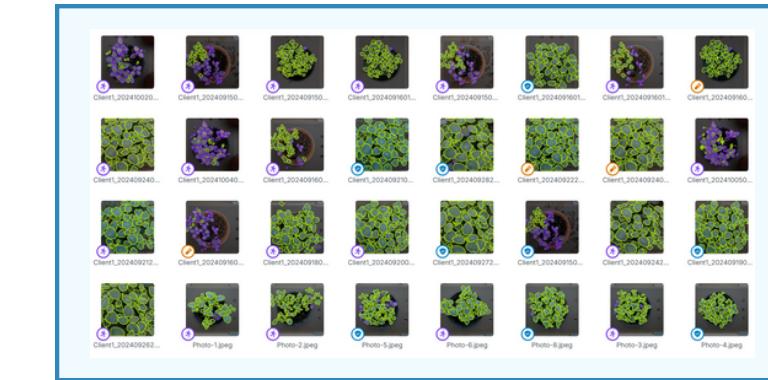
- 단순 JPEG 이미지 500장
- 정상/병충해 구별 X
- chocomint 종에만 한정



객체 라벨링(정상/비정상)  
Augmentation 활용 데이터 증강



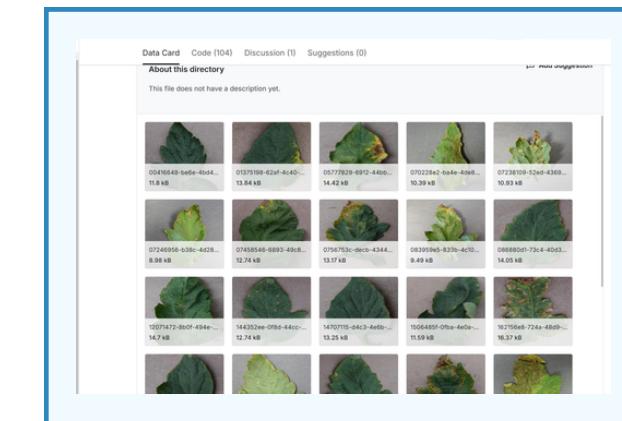
기존 데이터셋 비교  
약 60배 이상의 데이터 증량  
-> 안정적인 학습 데이터 확보



• 기업체 제공 데이터셋



• 실생활 수집 데이터

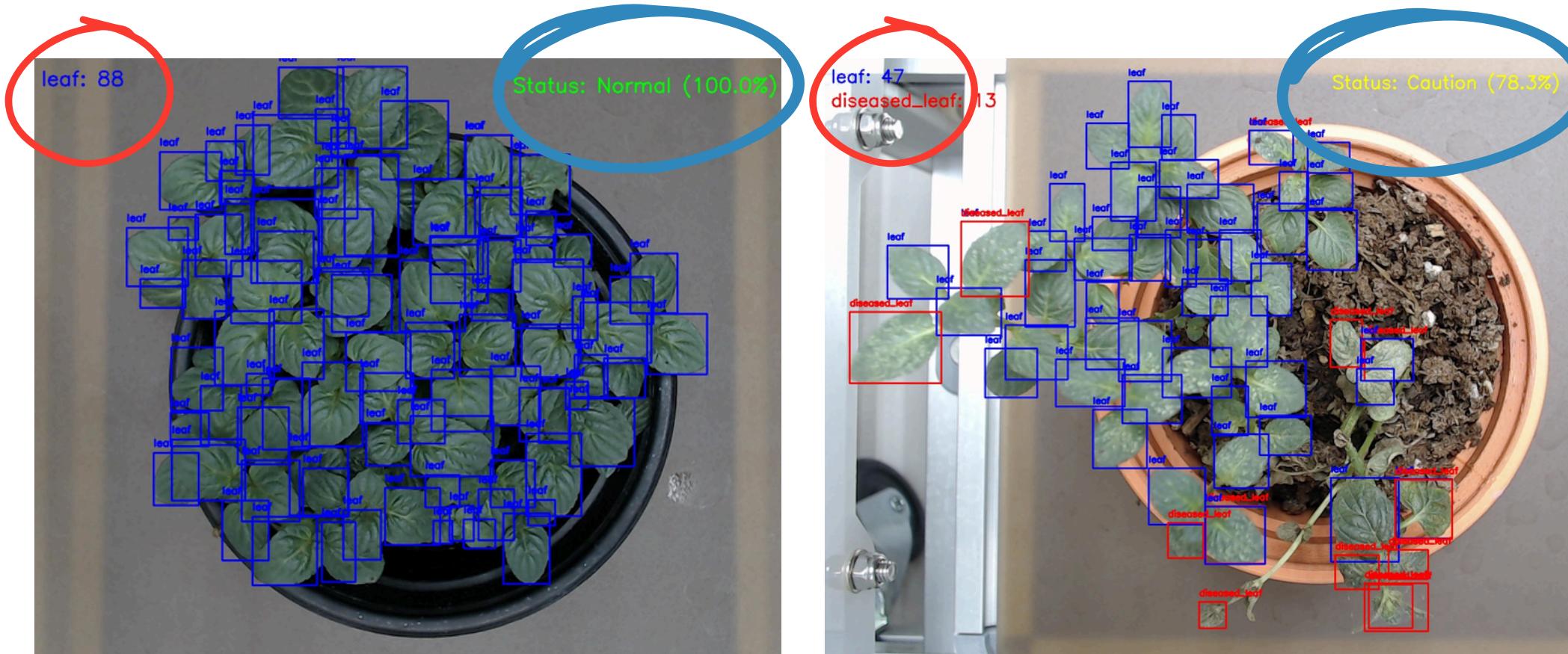


• Kaggle 오픈소스 활용한  
다양한 작물 데이터셋

작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

## 2-2. 코드 로직 설계

# 모델 학습 후 이미지 후처리



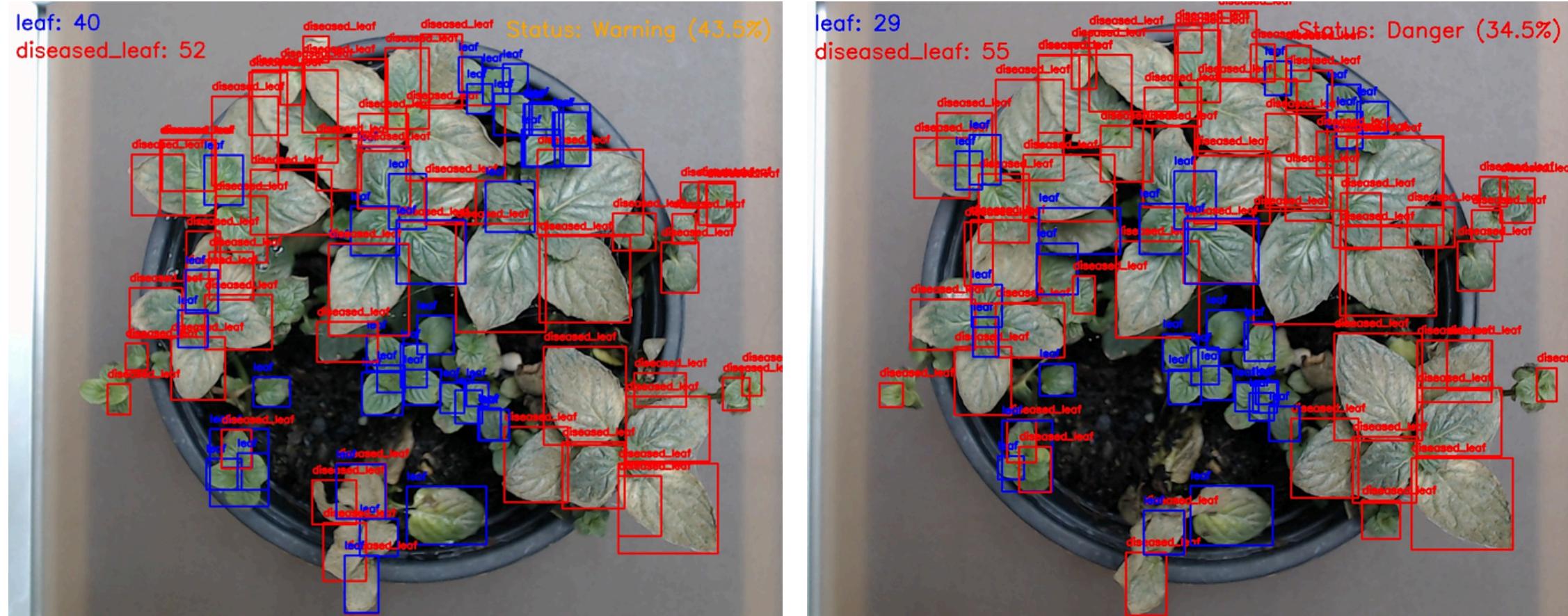
병충해 정도 및 상태 파악을 위한  
라벨 개수 counting

식별 박스 색 구별:  
정상 잎은 파랑/비정상 잎은 빨강

- 손상 상태 식별을 위한 비율 계산
- 80% 이상: Normal (초록색)
  - 60~80%: Caution (노란색)
  - 40~60%: Warning (주황색)
  - 20~40% Danger (빨간색)
  - 0~20%: Critical (보라색)

작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

## 모델 학습 후 이미지 후처리



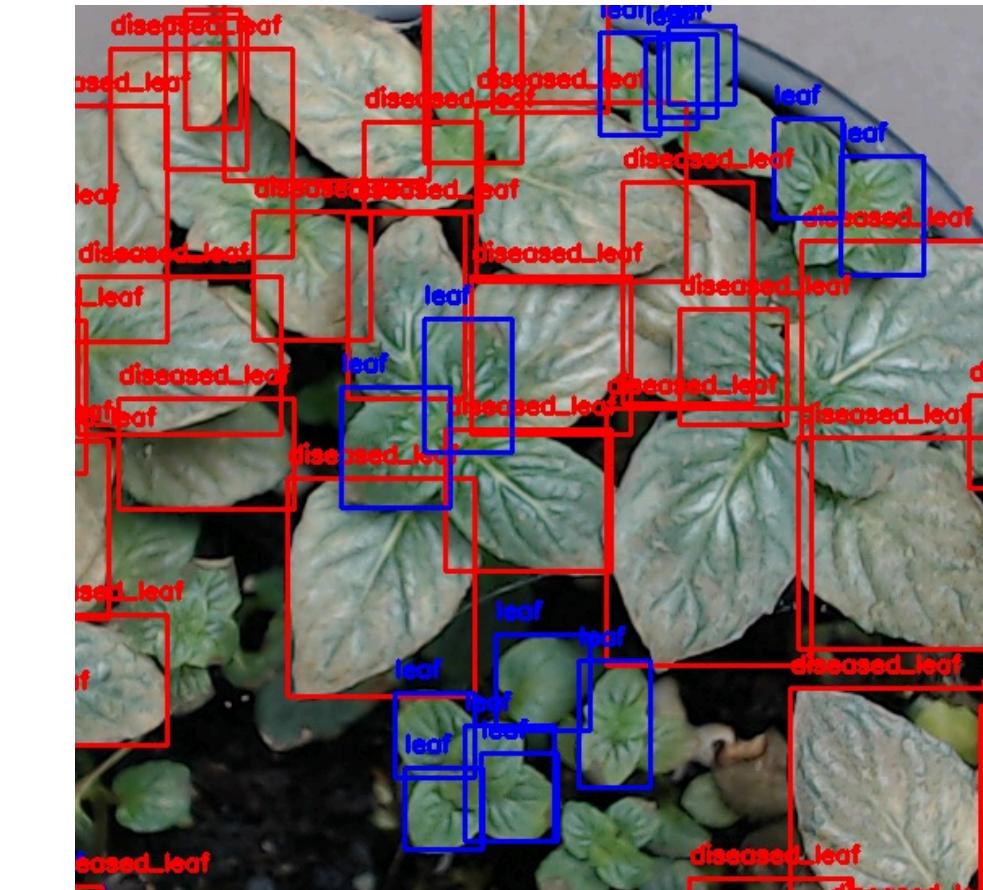
- 학습 데이터 기반 병충해 잎 식별로 피해 정도 식별 가능

## 발전된 코드 적용 예진

변경 전



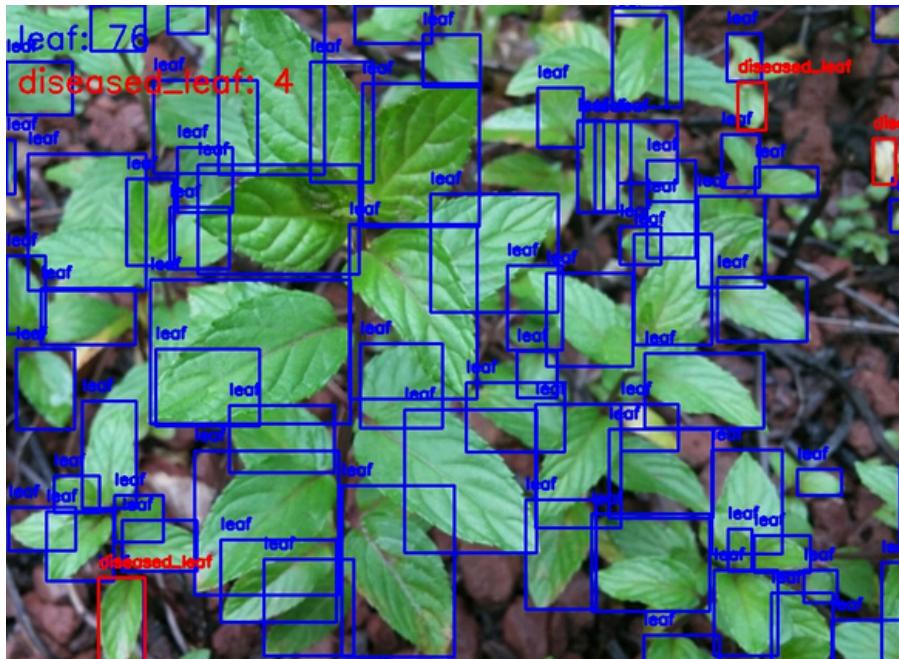
변경 후



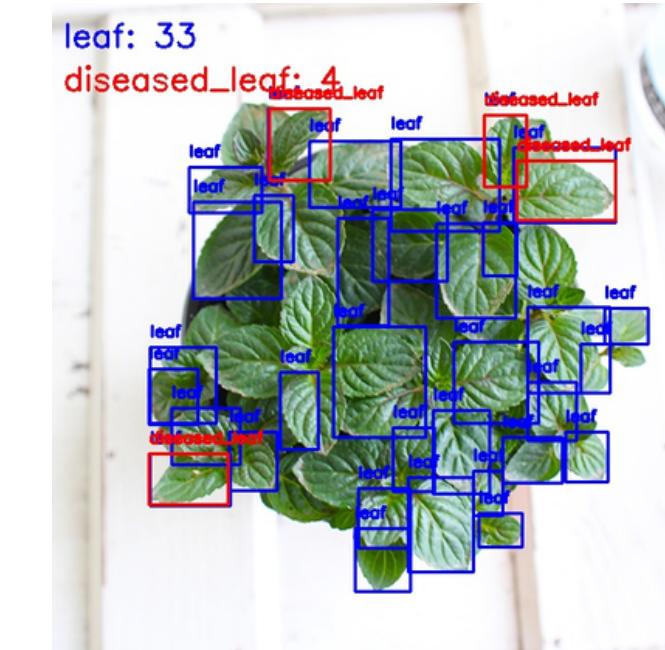
이전 모델에서는 오류가 발생하던 이미지에서 성공적으로 정상/비정상 검출되는 것을 확인

작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

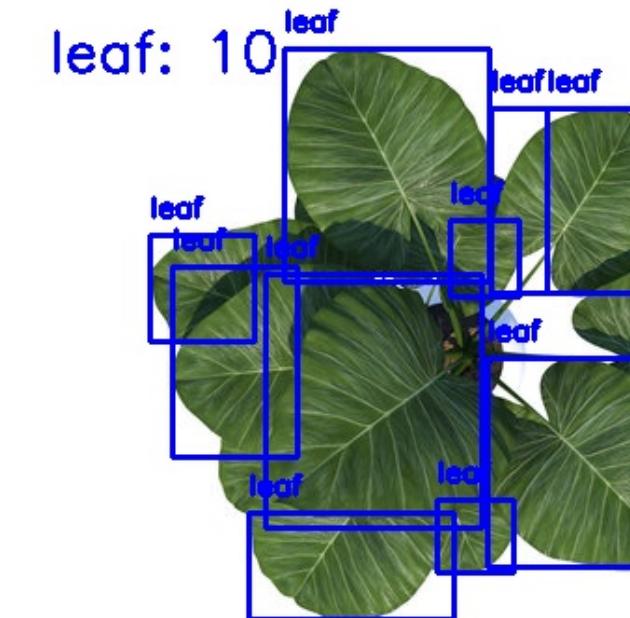
# 데이터셋 확대 기반 식별 식물 종류 다양화



페퍼민트



수국



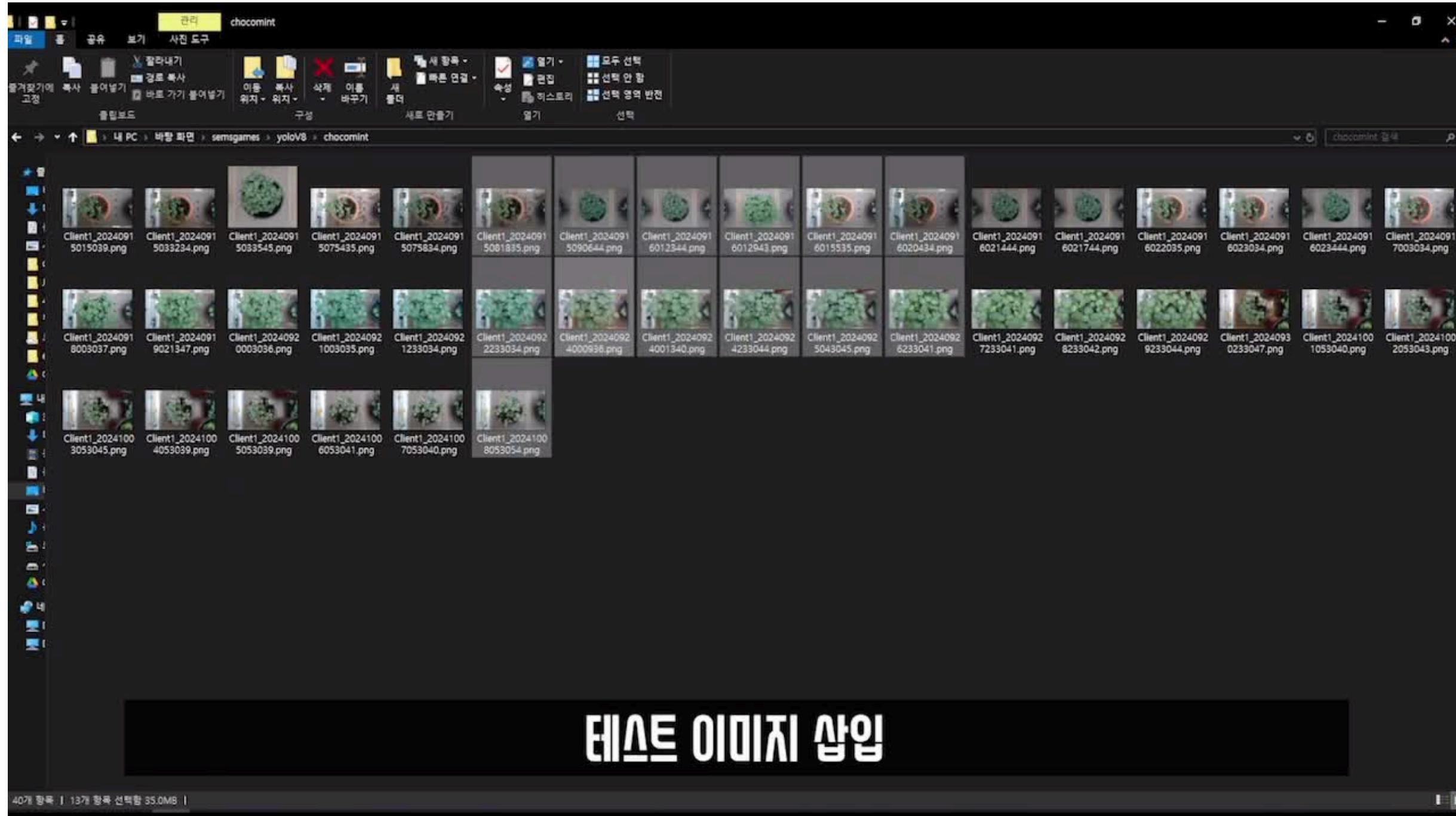
알로카시아  
브리즈베넨시스

기존 ‘초코민트’ 허브를 넘어  
다양한 식물에 높은 정확도로 객체 인식

작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

### 3-1. 결과 분석

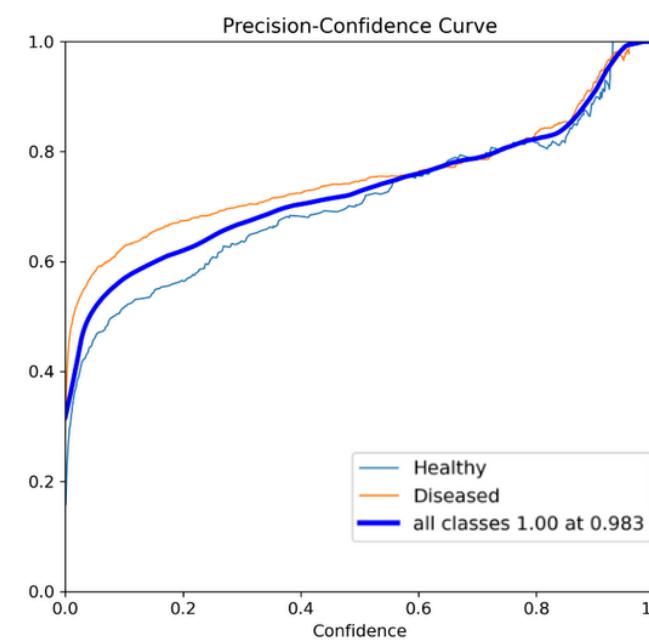
# 시연 영상



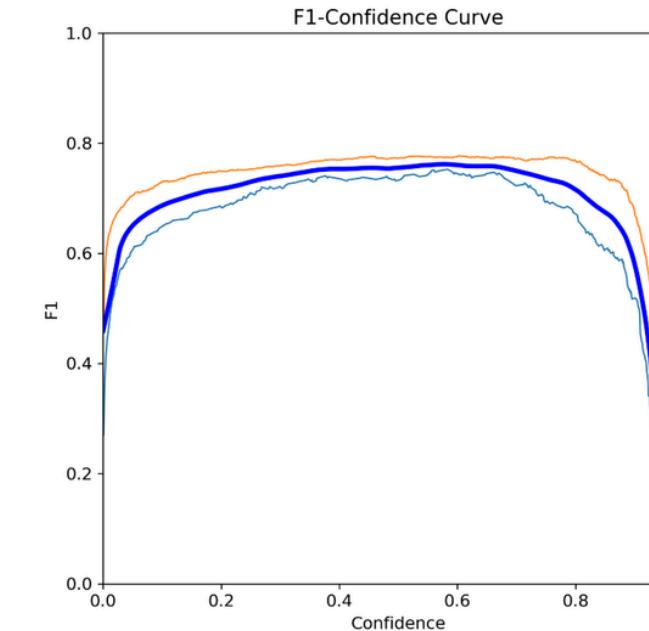
작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

### 3-2. 정확도 측정

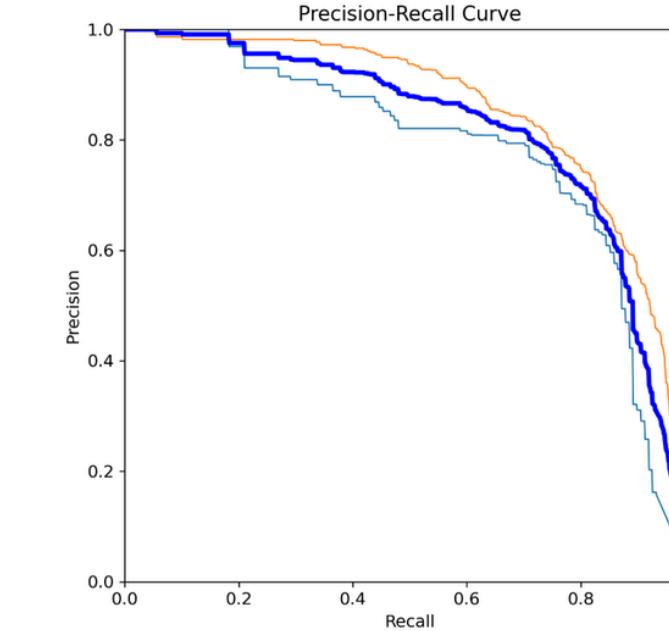
Precision-Confidence 커브



F1-Confidence 커브



Precision-Recall 커브



Diseased 클래스 AP: 0.846, mAP@0.5: 0.810

최고 F1 점수: 0.76

안정적인 속도로 학습이 진행되어 모델의 효용성 검증/ 높은 정확도 검증

깍물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술

### 3-3. 개선 방향 및 결론

#### 1. 주요 성과 요약

- YOLOv8을 활용한 경량 모델 개발: 정상/비정상 잎 탐지 시스템 구현
- Precision, Recall, F1 Score에서 우수한 성능 달성:
  - 평균 mAP@0.5: 0.810 (Diseased: 0.846, Healthy: 0.774)
- 비정상 잎의 탐지 정확도를 대폭 향상시켜 스마트팜에서의 활용 가능성 제시

#### 2. 기술적 기여

- 기존 수작업 기반 병진 관리의 한계 보완, 자동화 기술 제안
- RoboFlow를 활용한 데이터셋 라벨링 및 증강, YOLOv8의 효과적인 활용



**작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술**

### 3-3. 개선 방향 및 결론

#### 3. 기대 효과 및 전망

앞의 개수 기반 성장 정도/  
병충해 정도를 파악  
-> 시기 별 대처 시스템  
구축 가능

IoT 기반 스마트팜 시스템  
통합을 통해  
실제 상용화 가능성 확대

병든 잎 탐지의 정확성과  
효율성 증가  
-> 실생활 일반인도 사용  
가능한 판별 프로그램  
확장 가능성

**작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술**

## 팀원 소개



**박은공(팀장)**

: 학습 모델 개선

**김제원(팀원)**

: 데이터 전처리

**강경빈(팀원)**

: 라벨링 및 코드 안정성 평가

**유수현(팀원)**

: 학습 모델 구축

**작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술**

감사합니다 :)

Q & A

작물 품질 상태 판별 : 이미지 프로세싱 및 인공지능 기술