

이미지 인식을 통한 딸기 주요 병해 파악과 대처 방안 알림 서비스 개발



농벤저스

마경수,오주완,김민성,
조경윤,박종석,이재호

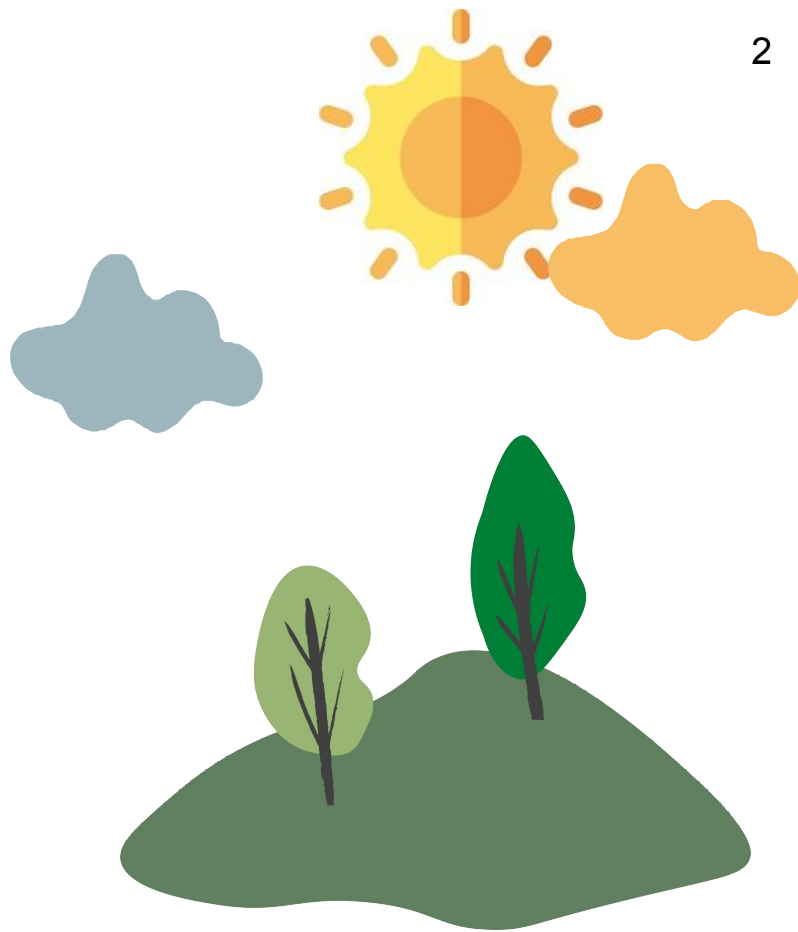
목차

1.기획배경

2.시나리오 및 활용기술

3.데이터셋 제작과정 및 웹 구현

4.기대가치 및 향후 활용방안



식물병에 관한 사건

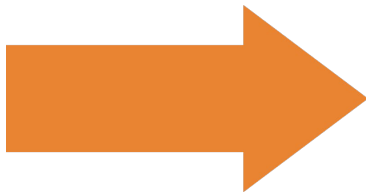
- 벼 깨씨무늬병

인도에서 나타난 벼의 깨씨무늬병에 의해 200만명 이상이 사망

식물병의 발생원인과
방제법의 중요성

- 아일랜드 감자 역병

영국과 아일랜드의 대흉년으로 100만명 이상이 사망하고, 150만명 이상이 신대륙으로 이주(1845~1860)





Part 1.

기획 배경



기획 배경

스마트 팜 모델의 한계점

주요기능 : 원격 수동 제어

병해탐지
기능 X

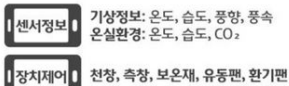
주요기능 : 자동 제어, 클라우드 서비스

주요기능 : 로봇을 통한 농작업,
에너지 관리

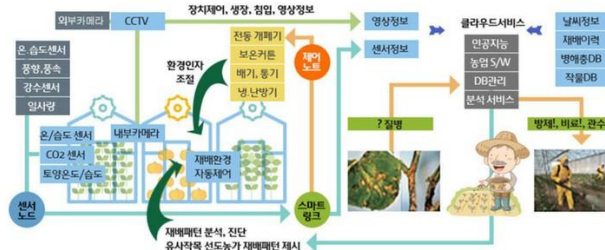
1세대 모델('16) 원격 감시 + 원격 제어



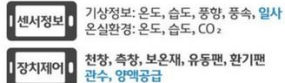
농민이 영상을 통해 직접
“원격 수동제어”



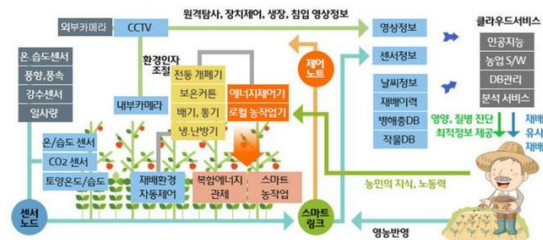
2세대 모델('18) 지상부 복합환경제어 + 클라우드서비스



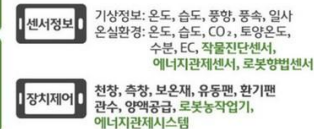
작물의 지상부/지아부 생육환경을
“자동제어”



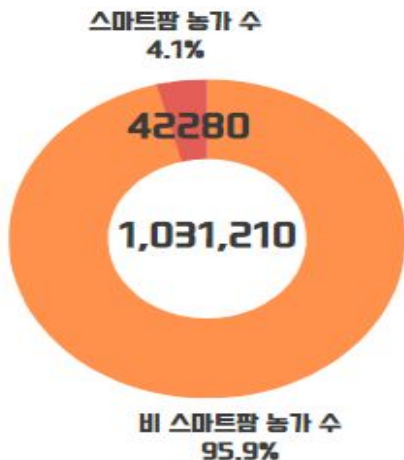
3세대 모델('20) 복합에너지관리 + 스마트 농작업



스마트 온실 시스템의
“최적 에너지관리와 로봇 농작업”



대상 분류



스마트 팜 비중

1. 스마트팜 도입을 한 농가

대상 : 20-30대 청년층 or 정부의 지원을 받아 귀농을 한 사람

서비스 : 재배환경에 설치된 **CCTV 이미지**를 활용하여 실시간으로 병해 탐지 기능 추가, 매일 아침, 저녁에 발견된 병해들의 위치와 수 알림 서비스

2. 스마트팜 도입을 하지 않은 농가

대상 : 40-60대 중장년층 or 도시나 집에서 취미로 가꾸는 농가

서비스 : **스마트폰 어플**을 통하여 간편하게 이미지를 입력하면 해당 병해의 종류와 방제법 알림 서비스

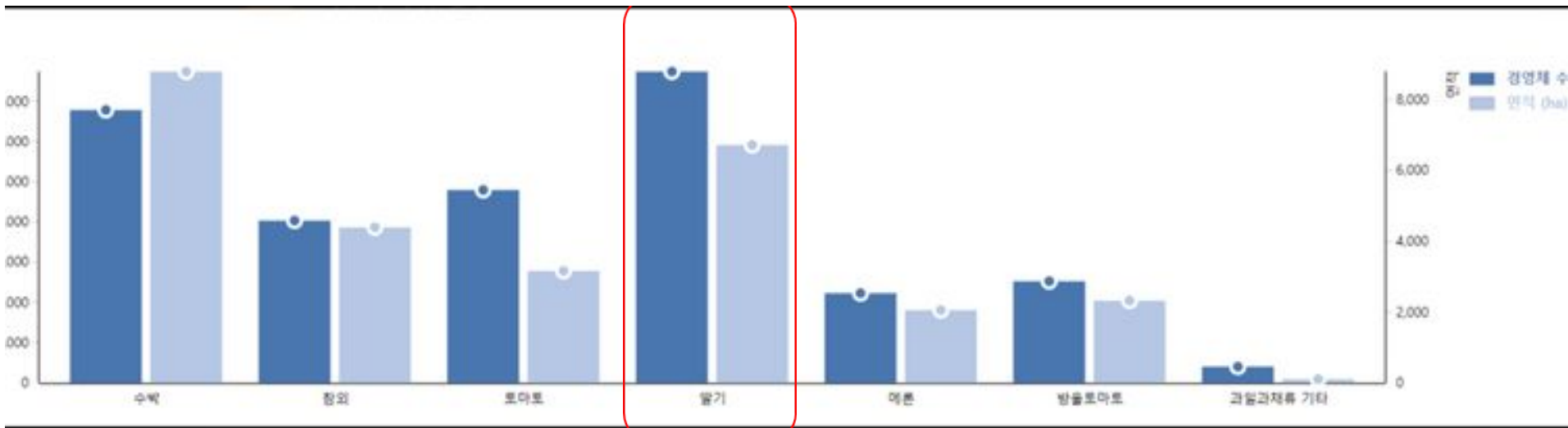
WHAT?

- 딸기



선정 근거 : 현재 시설재배 되고 있는 과수 중 **가장 많은 경영체**가 운영하고 있는 작물 (딸기 - 수박 - 토마토 순)

시설재배를 선택한 이유 : **환경변수를 차단**할 수 있는 최소한의 조건을 갖춘 곳이라고 판단



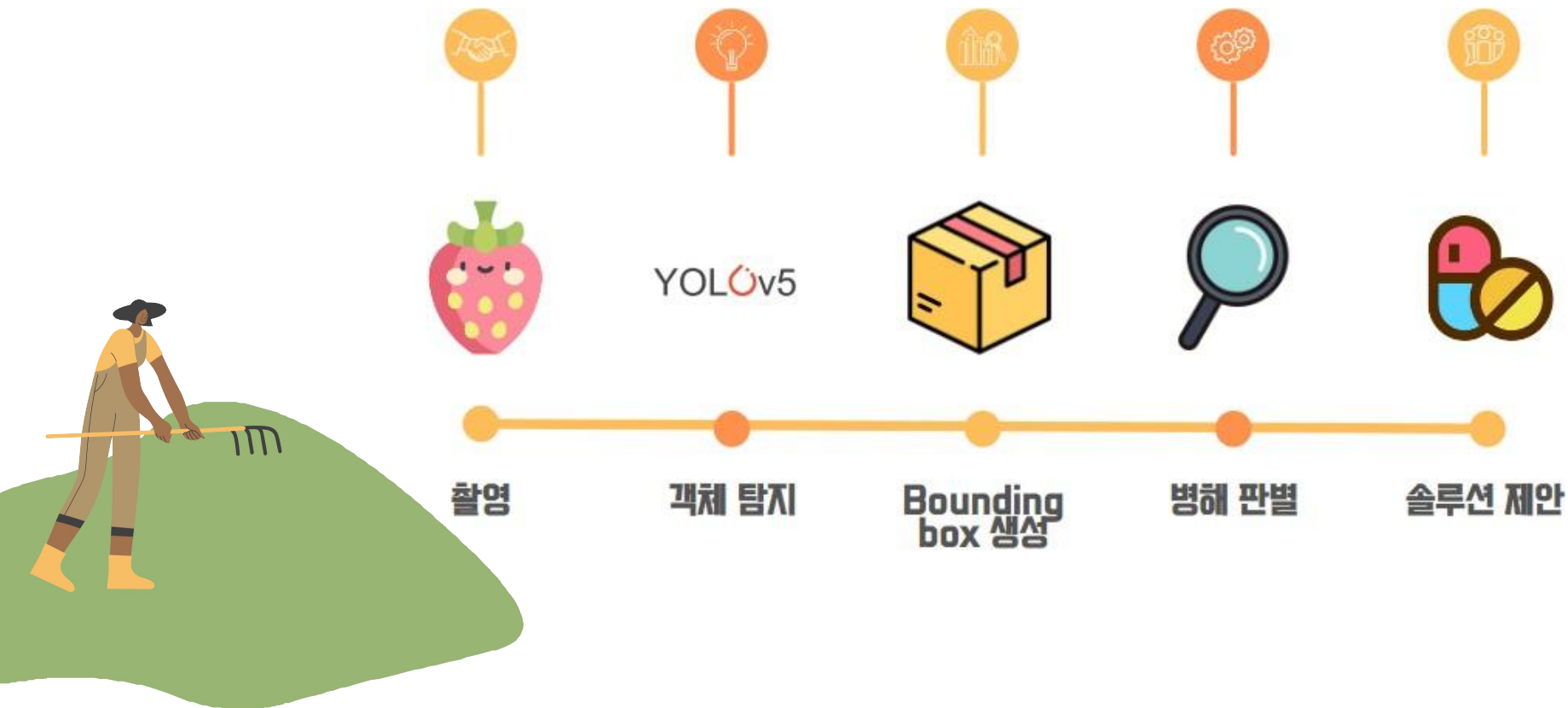


part2.

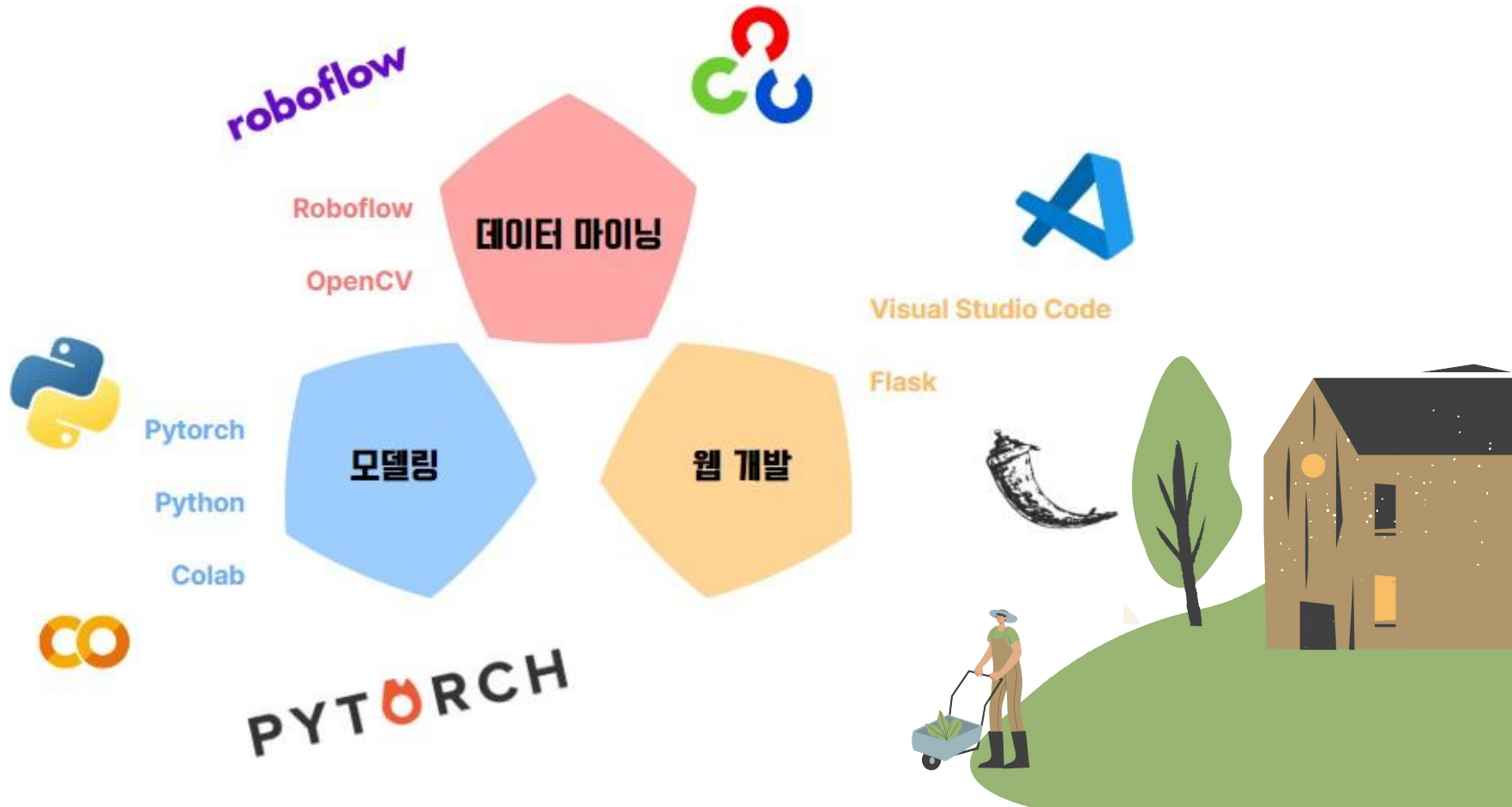
시나리오 및 활용기술



시나리오



활용 기술





part3.

데이터셋 제작과정 및 웹 구현

데이터셋 제작과정 및 웹 구현

프로세스

STEP1

데이터 수집 및 전처리

딸기 질병 이미지
데이터 수집



객체인식을 위한 이미지
라벨링 처리

STEP2

모델 학습

전처리된 데이터로
학습한 모델 생성



데이터에 적합한
알고리즘
선택



모델의 파라미터 최적화

STEP3

라벨링

객체 인식률을 높이기 위
한 라벨링 개선 작업 및
이미지 증식

STEP4

웹 구현

커스텀 모델을 활용한
웹 구현

데이터 수집/전처리

주석

Class

leaf spot 뱀눈무늬병

powdery mildew leaf 흰가루병 잎

gray mold 잿빛 곰팡이병

angular leafspot 세균모무늬병

blossom blight 꽃 곰팡이병

powdery mildew fruit 흰가루병

과일 anthracnose fruit rot 탄저병

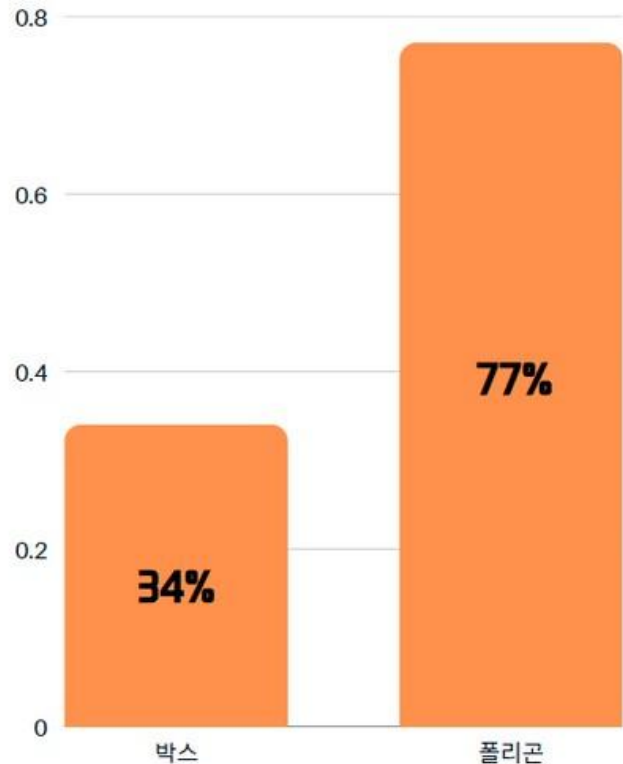
• 박스 방식



• 폴리곤 방식



결과

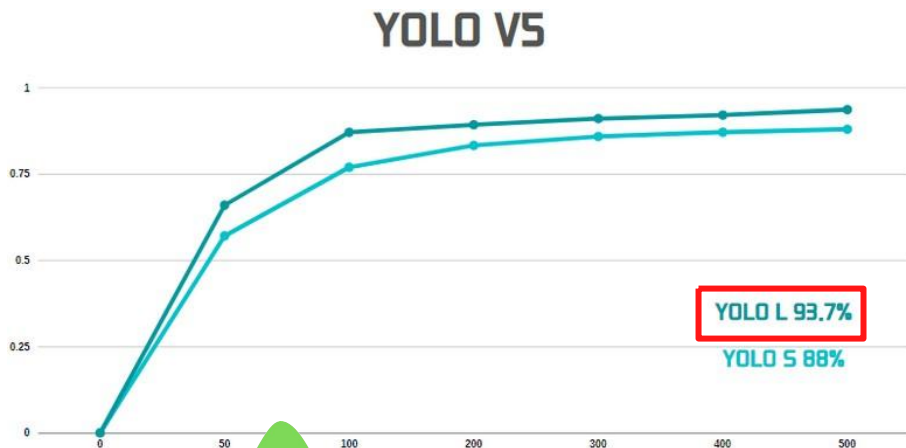


YOLO Vs S

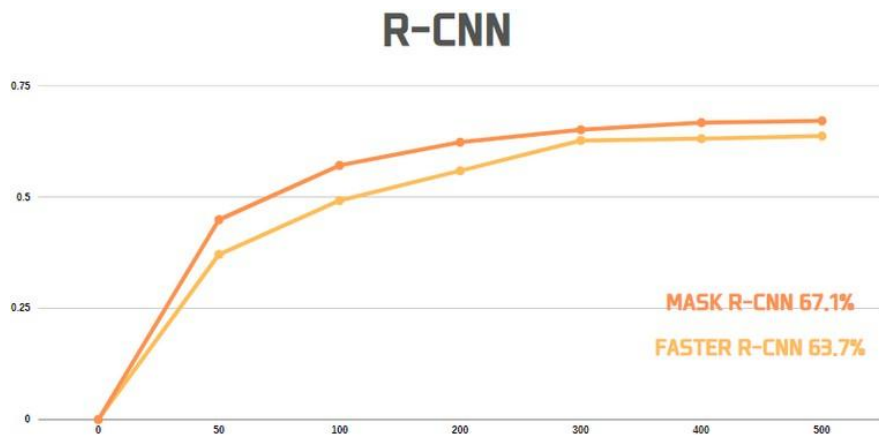
모델 학습 비교

14

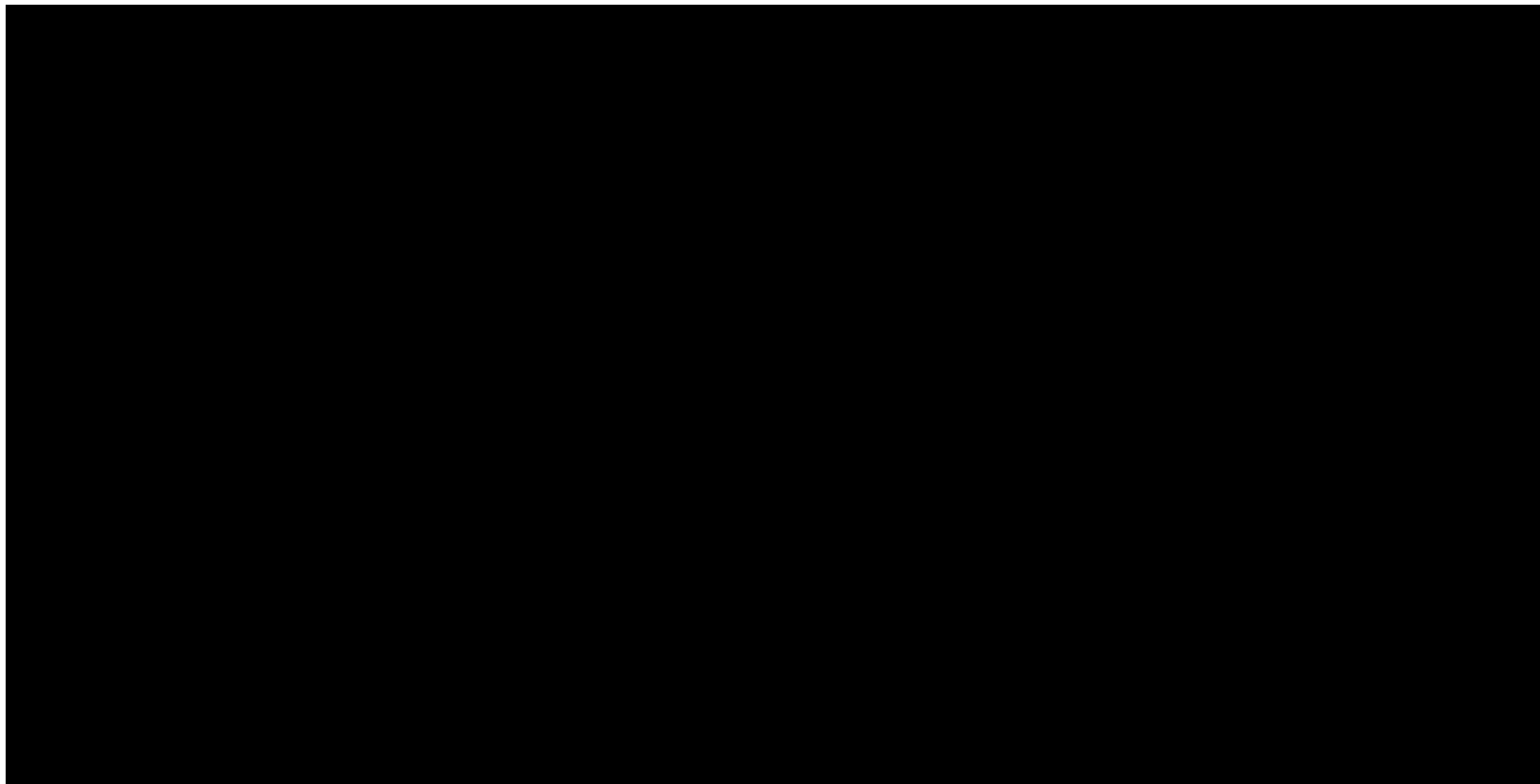
One-stage detector



Two-stage detector



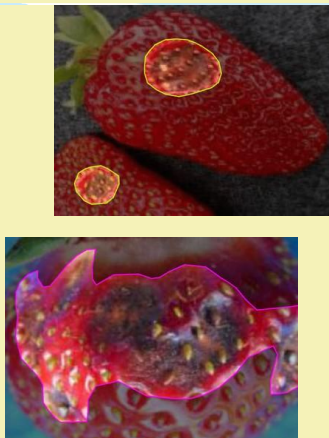
모델 구성 과정



모델성능 개선 과정

77%

1. 다른 class에 비해 과실의 인식률이 낮은 것을 확인
2. 과실 데이터가 적어 데이터의 불균형이 인식률을 낮춘다고 판단
3. 과실질병 이미지를 증식



85%

1. 증식결과 과실질병 인식률이 73%로 여전히 저조
2. 특히 인식률이 낮았던 gray mold클래스가 발병부위나 생김새가 다양하여 인식률이 낮다고 판단



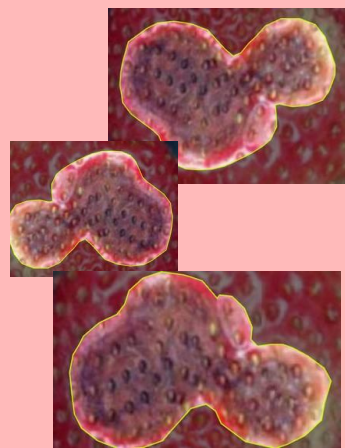
88%

1. 인식률을 올리기 위해 라벨링 패턴을 추가
2. 팔기의 줄기부분에도 라벨링을 추가하여 인식률을 개선
(- 건물객체인식 논문 참조)



93.7%

1. 전체 데이터의 수가 적어 인식률이 낮을 수도 있다는 멘토링을 통해 전체 데이터를 증식하여 4000개 이미지 데이터 수 증가
2. 파라미터 최적화를 통하여 최종적으로 인식률 0.937까지 상승



데이터 증식

증식 전 이미지 데이터 수 : 5120개

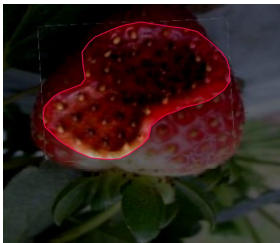
증식 후 이미지 데이터 수 : 9600개

1. 상하 반전

원본



상하 반전

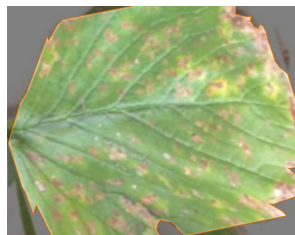


2. 흐림 처리

원본



흐림처리



3. 좌우 반전

원본



좌우 반전

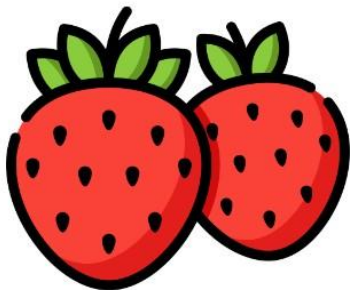


웹 구현

메인 화면(index)

실제 결과 화면(result)

농벤저스



파일 선택 leaf_spot.jpg

Match!



질병 없음
정상입니다.

잘 가주세요



뱀눈무늬병
미정상입니다

원인/대처법 보러 가기



웹 구현 코드

```
if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Flask app exposing yolov5 models")
    parser.add_argument("--port", default=5000, type=int, help="port number")
    args = parser.parse_args()

# local model
model = torch.hub.load(
    'ultralytics/yolov5', 'custom', 'C:\cakd5\strawberry_3rd\strawberry\models.pt', autoshape=True
) # force reload = recache latest code
model.eval()

flask_options = dict(
    host='0.0.0.0',
    debug=True,
    port=args.port,
    threaded=True,
)

app.run(**flask_options)
```

webapp.py

torch.hub.load() API를 사용해 yolo v5와 사전 학습된 모델을 호출

```
@app.route("/", methods=['GET', 'POST'])
def detect():
    if request.method == "POST":
        DeleteAllFiles('C:/cakd5/strawberry_3rd/strawberry/static/result') # 탐지 파일 저장소 비우기

    # 다중파일 업로드
    if "file" not in request.files:
        return redirect(request.url)
    files = request.files.getlist("file")
    results=[]

    if not files:
        return
    # result.html에서 출력할 요소들을 리스트화
    filenames=[] # 파일 이미지
    pf=[] # 검사/재검사
    nm=[] # 질병 명칭
    lk=[] # 방제법 제공 사이트 링크
    st=[] # 링크 버튼에 들어가는 간단한 메시지

    for file in files:
        filename = file.filename.rsplit("/")[-1] #파일경로에서 파일명만 추출
        print("processing :", filename)

        img_bytes = file.read()
        img = Image.open(io.BytesIO(img_bytes))

        result = model(img, size=1024)
        results.append(result)
        result.render() # results.imgs에 바운딩박스과 라벨 처리
        data = result.pandas().xyxy[0][['name']].values.tolist() # results.imgs의 name값만 가져오기
        print("data:",data)
```

webapp.py

app.route

def detect(탐지 결과 및 솔루션 제공을 위한 함수)

1. 모델 사용의 원활함을 위해 매 탐지 진행마다 이미지 저장내역 초기화
2. 이미지를 입력받게 되면 결과창에 출력해야 하는 탐지 결과 이미지, 질병의 유무, 질병의 명칭, 방제법 사이트 링크를 append 시킬 리스트 생성
3. 객체탐지를 진행하고 그 결과중 bounding box의 name만 별도로 선언

웹 구현 코드

```

for img in result.imgs:
    img_base64 = Image.fromarray(img)
    img_base64.save(f'static/result/{filename}', format="JPEG") # 탐지한 이미지 저장
    print('saving detectfile')

if len(data) == 0:
    # data 리스트의 값이 0일경우 정상
    pf.append("정상입니다.")
    nm.append("이상 없음")
    st.append("위로가기를 눌러주세요")
    lk.append("/") # index.html로 돌아가기

if len(data) > 0:
    pf.append("비정상입니다.") # data 리스트의 값이 0이 아닐경우 class name 확인
    if data == [{"Angular Leafspot"}]: # class name값을 확인하고 번역
        nm.append("세균무늬병")
        lk.append("https://ncpms.rda.go.kr/npms/SicknsInfoDt1R.np?sSearchWord=%E
        st.append(" 원인/대처법 보러 가기 ")
    if data == [{"Anthracnose Fruit Rot"}]:
        nm.append("탄저병")
        lk.append("https://ncpms.rda.go.kr/npms/SicknsInfoDt1R.np?sSearchWord=%E
        st.append(" 원인/대처법 보러 가기 ")
    if data == [{"Blossom Blight"}]:
        nm.append("꽃 곰팡이병")
        lk.append("http://www.nongsaro.go.kr/portal/ps/psb/psbb/farmUseTechDt1.
        st.append(" 원인/대처법 보러 가기 ")
    if data == [{"Gray Mold"}]:
        nm.append("갯빛 곰팡이병")
        lk.append("https://ncpms.rda.go.kr/npms/SicknsInfoDt1R.np?sSearchWord=%E
        st.append(" 원인/대처법 보러 가기 ")
    if data == [{"Leaf Spot"}]:
        nm.append("뱀눈무늬병")
        lk.append("https://ncpms.rda.go.kr/npms/SicknsInfoDt1R.np?sSearchWord=%E
        st.append(" 원인/대처법 보러 가기 ")
    if data == [{"Powdery Mildew Fruit"}]:
        nm.append("흰 가루병 (과실)")
        lk.append("https://ncpms.rda.go.kr/npms/SicknsInfoDt1R.np?sSearchWord=%E
        st.append(" 원인/대처법 보러 가기 ")
    if data == [{"Powdery Mildew Leaf"}]:
        nm.append("흰 가루병 (잎)")
        lk.append("https://ncpms.rda.go.kr/npms/SicknsInfoDt1R.np?sSearchWord=%E
        st.append(" 원인/대처법 보러 가기 ")

root = "static/result/" # 탐지 파일 저장경로 설정
filenames.append(root + filename) # 탐지 파일 경로 및 파일 명

```

webapp.py

app.route

def detect(탐지 결과 및 솔루션 제공을 위한 함수)

1. 탐지가 완료된 이미지를 지정해둔 경로에 저장
2. 탐지된 결과에 **bounding box**가 없을경우엔 정상이미지임을, 있을 경우엔 해당 병명과 방제법을 제공할 사이트 링크를 만들어둔 리스트들에 저장



```

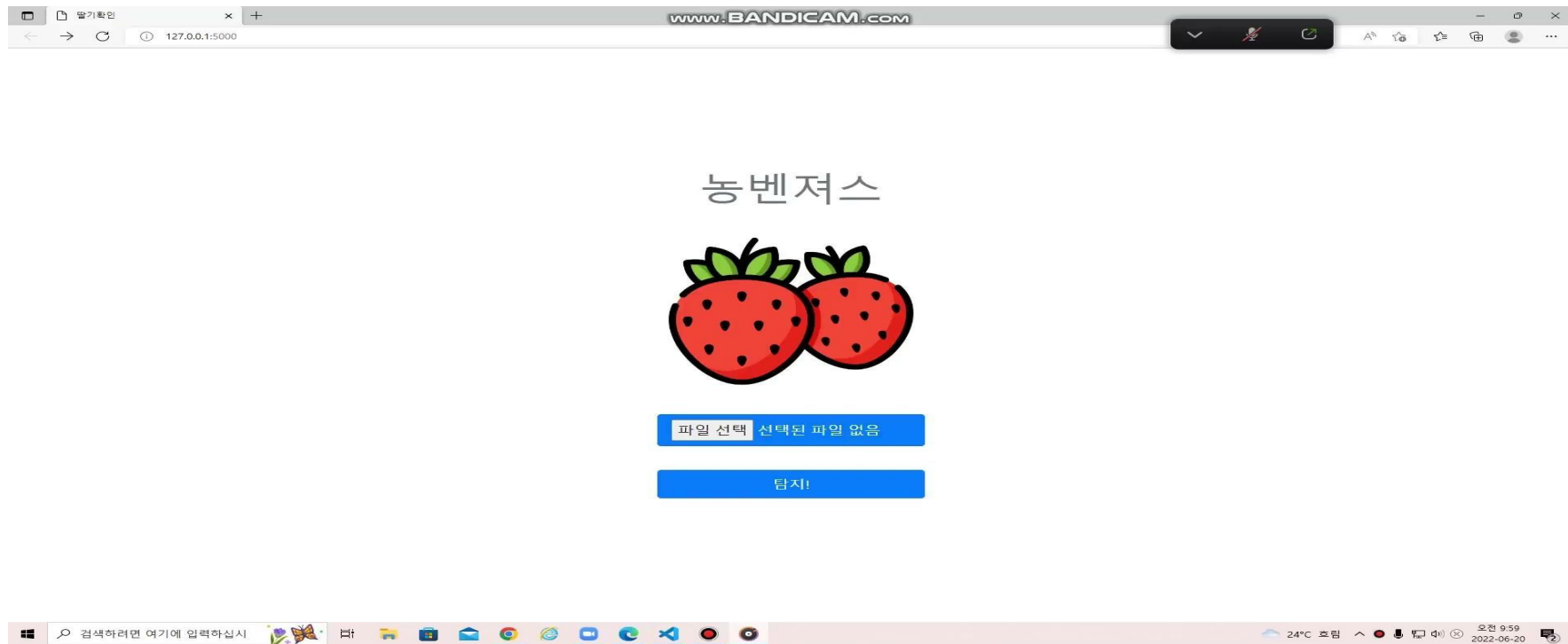
<body>
  {for i in range(0, len(files))}
  <div class="card mx-30" style="width: 20rem; height: 30rem">
    <img class="card-img-top" src={{files[i]}} alt="Card image cap" style="width: 13rem; height: 15rem">
    <div class="card-body">
      <h5 class="card-title">{{nm[i]}}</h5>
      <p class="card-text">{{pf[i]}}</p>
      <a href={{lk[i]}} class="btn btn-primary" target='_blank'>{{st[i]}}</a>
    </div>
  </div>
  {endfor %}
</body>

```

result.html

1. for문과 부트스트랩의 카드 디자인을 활용하여 결과창에 출력할 요소들을 배치

웹 구현 시현영상





part 4.

기대 가치 및
향후 활용 방안

기대가치



시간 단축



검사 의뢰 시간의
절약을 통하여 빠르게
병해에 대처



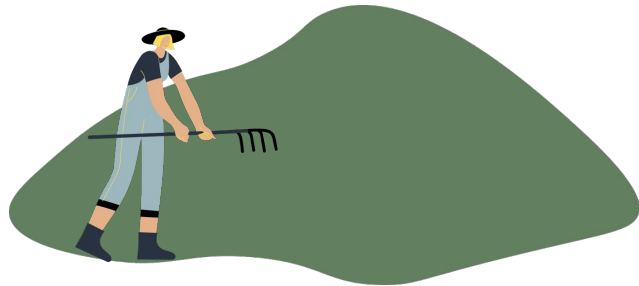
수확량 증가



대규모로 발생하는
병들의 빠른 대처를
통하여 수확량 보호 및
증가



- 적용 가능한 작물의 범위를 확대
 - 딸기 다음으로 비중이 높은 토마토, 수박을 시작으로 다양한 시설, 노지 작물들로 범위를 넓혀 추가할 예정
- 농업 교육 인프라 구축
 - 식물병에 관하여 궁금증을 가지고 있거나 도움이 필요한 사람들에게 교육적인 목적으로 사용
- 병해 데이터를 활용하는 유관기관의 협약
 - 유관기관과 협력하여 새로운 질병에 대한 실시간 업데이트 및 솔루션 제공



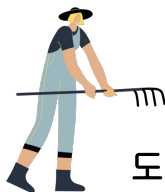
• 기존 스마트팜 모델과의 결합

- 시설 농가의 CCTV를 활용하여 실시간으로 병해 탐지, 방제를 자동으로 해주는 기능이 추가된 스마트팜 모델 개발

스마트팜 시장 규모

세계적인 스마트팜 시장의 규모가 증가하는데 비해 우리나라의 성장세는 더딤

>> 방제 기능을 추가한 모델을 개발한다면 스마트팜 가속화 예상



도시농업 시장 규모

도시농업에 참여자의 증가 및 텃밭면적의 증가

>> 도시농업에 디지털화를 적용함으로써 진입장벽을 낮춰 디지털 인식 전환 및 도입 가속화



팀원소개

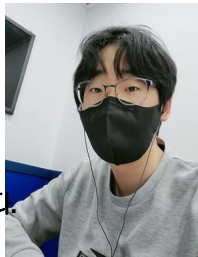


이름: 마경수

역할: 데이터 모델링, 기획 총괄

github: <https://github.com/EDPS-7532>

소감: 프로젝트를 진행하면서 데이터 작업에 대한 중요도를 인식하였고, 의미있는 성과를 도출해 낼 수 있어서 좋았습니다.

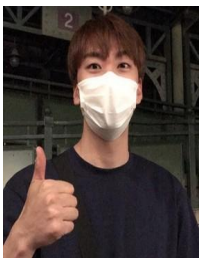


이름: 오주완

역할: 도메인 조사, 기획, ppt, 데이터전처리

github: <https://github.com/joowaun93>

소감: 전공분야였던 농업과 관련해서 프로젝트를 해보고싶었는데 유의미한 결과를 낼 수 있었던 것 같아서 좋은 경험을 한 것 같습니다.



이름: 김민성

역할: 데이터 모델링 및 데이터 마이닝

github: <https://github.com/nycticebus0915>

소감: 객체인식을 위한 라벨링을 개선, 파라미터 최적화를 통해 데이터 분석의 어려움과 중요도를 직접 느낄 수 있어 좋은 경험이 되었습니다.



이름: 조경운

역할: 데이터 전처리, 웹 구현(프론트 엔드)

github: <https://github.com/kkyxxn>

소감: 데이터 분석에 앞서 분석에 필요한 데이터 요건을 정의하고 데이터를 확보하는데 좋은 경험을 할 수 있었습니다.



이름: 박종석

역할: 데이터 전처리, 웹 구현(백 엔드)

github: <https://github.com/blazestar95>

소감: 프로젝트를 진행 도중 여러 부분에서의 문제점 도출 및 해결 과정을 통해 코딩과정의 이해가 더욱 깊어졌다 느껴 좋은 경험을 얻었습니다.



이름: 이재호

역할: 데이터전처리 및 수집, 자료조사, 기획보조

github: <https://github.com/dlwogh799>

소감: 맡은 역할을 수행하면서 자료나 데이터 수집이 얼마나 중요한 것인지 새삼 다시 느낄 수 있어서 좋은 경험이 된 것 같습니다.

Q&A

감사합니다