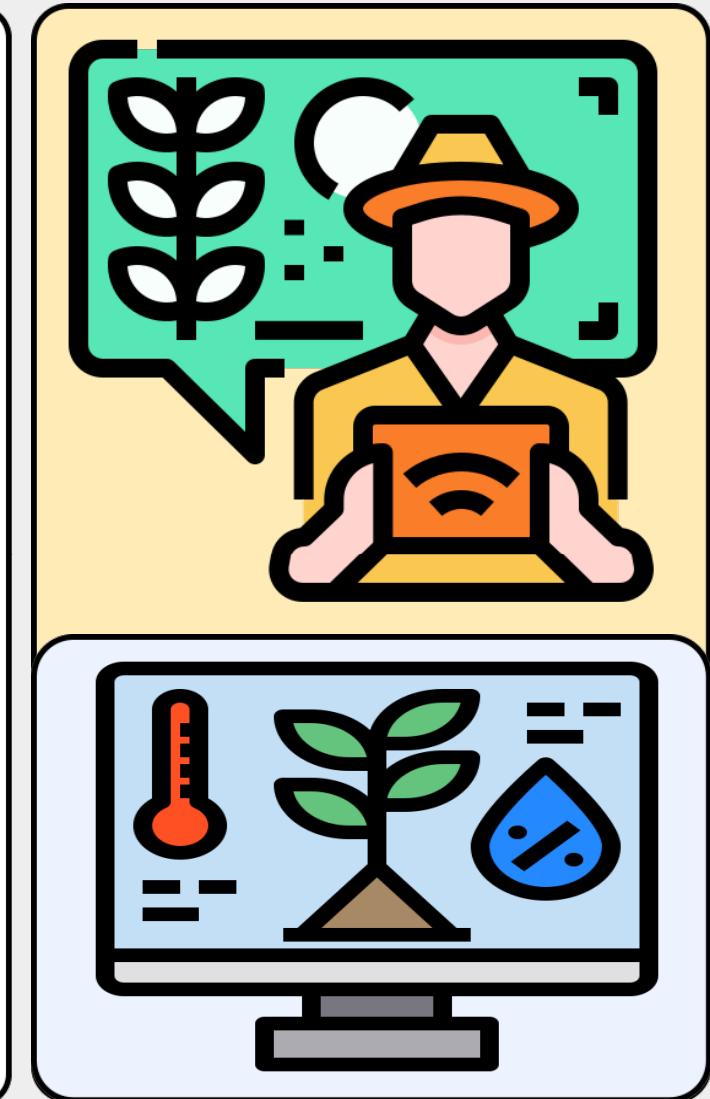
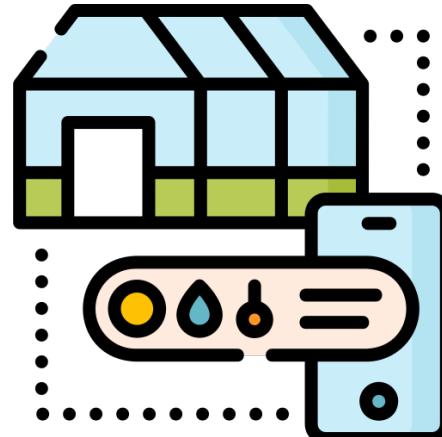


국내 스마트팜 도입을 위한 AI 기반 토마토 생체정보 모니터링 시스템 및 품질 예측 모델 개발

2팀) 토마토는 거꾸로 읽어도 토마토

<https://github.com/naleetwo/FinalProject.git>



CONTENTS

01

프로젝트 개요

02

역할 및 담당

03

데이터 베이스

04

품질 예측 모델

05

질병 진단 모델

06

생육단계 판단 모델

07

WEB

08

자체 평가 의견

01. 프로젝트 개요

배경 및 목적

세계 식량 위기 - 분쟁, 기후 재난, 경제 충격

러시아와 우크라이나의 주요 곡물 세계 시장 점유율 ※2018~2020년 기준, 단위: %



자료: 국제식량정책연구소(IFPRI)



2022 GLOBAL REPORT ON FOOD CRISES

JOINT ANALYSIS FOR BETTER DECISIONS



세계 가뭄 주요 현황

영국

- 5개월(3~7월) 연속 월간 강수량이 예년보다 낮은 수준
- 8월 12일, 8개 지역을 공식 가뭄 지역으로 선포
- 메릴라 바닥을 드러낸 템스강

독일

- 라인강 수위 급감으로 바지선 운송 제한

프랑스

- 7월 강우량 1959년 이후 최소치
- 메릴라 바닥을 드러낸 루아르강

이탈리아

- 7월, 북부지역 포강 주변 5개 주 비상사태 선포
- 메릴라 바닥을 드러낸 포강



출처 : google images, IPK(institute Pasteur Korea), 경향신문

01. 프로젝트 개요

배경 및 목적

세계 식량 위기 - 분쟁, 기후 재난, 경제 충격

러시아와 우크라이나의 주요 곡물 세계 시장 점유율 ※2018~2020년 기준, 단위: %



세계 가뭄 주요 현황

2022년 5월 04일

전 세계 식량 위기 보고서: 사상 최악의 식량 불안정



2022 GLOBAL REPORT ON FOOD CRISES

JOINT ANALYSIS FOR BETTER DECISIONS



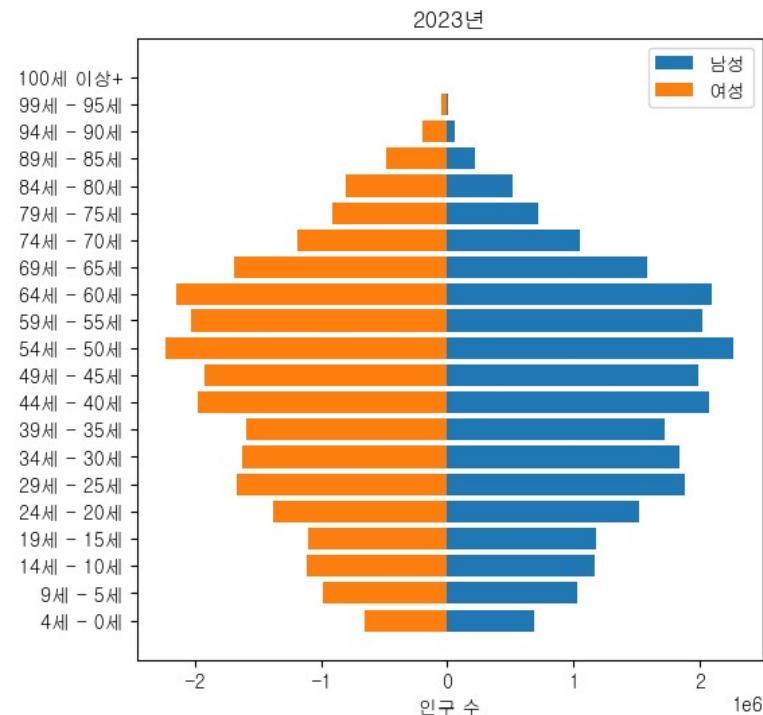
연합뉴스

출처 : google images, IPK(institute Pasteur Korea), 경향신문

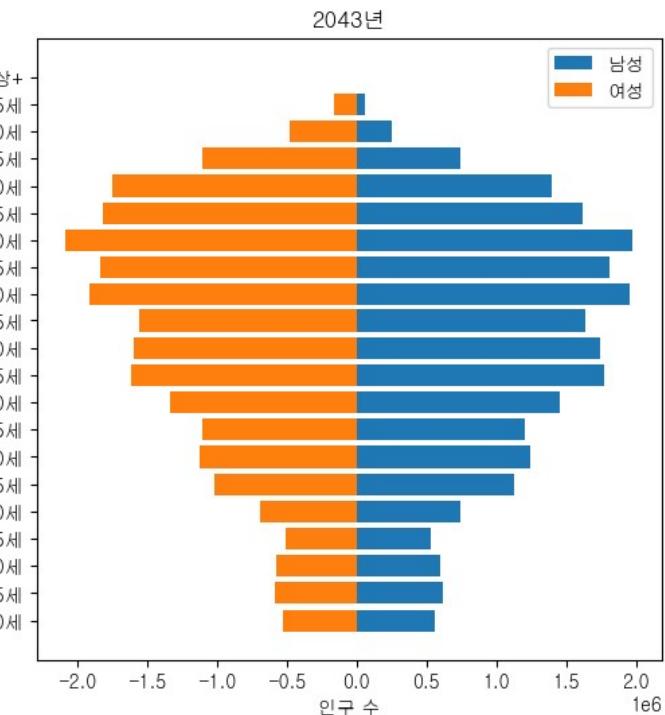
01. 프로젝트 개요

배경 및 목적

대한민국 현재 & 미래 평균연령 및 인구 - 인구 감소 및 고령화



총 인구 : 약 5,140만 명
평균연령 : 44.5 세



총 인구 : 약 4,650만 명
평균연령 : 53.8 세

출처 : 통계지리정보 서비스 - 통계청

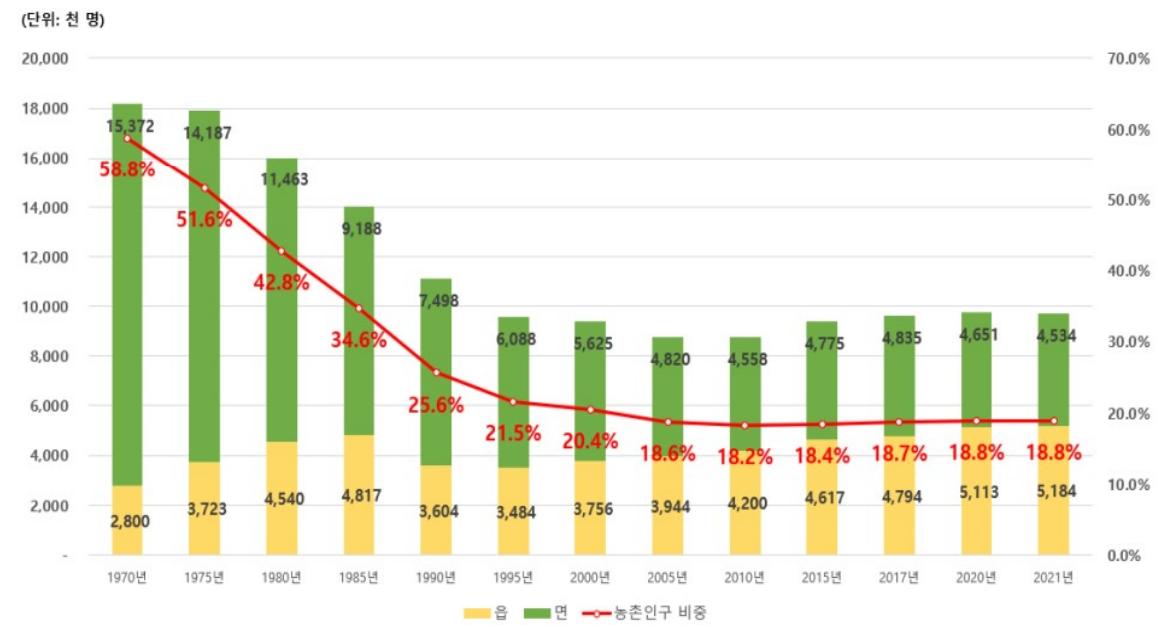
01. 프로젝트 개요

배경 및 목적

국내 농업 현황 및 전망 - 국내외 곡물자급률, 국내 농촌 인구 변화율



국내 농촌 인구 변화율



출처 : 중앙일보, 이미디어, 한국농촌경제연구원

01. 프로젝트 개요

배경 및 목적

국내 농업 현황 및 전망 - 국내외 곡물자급률, 국내 농촌 인구 변화율

국내외 곡물



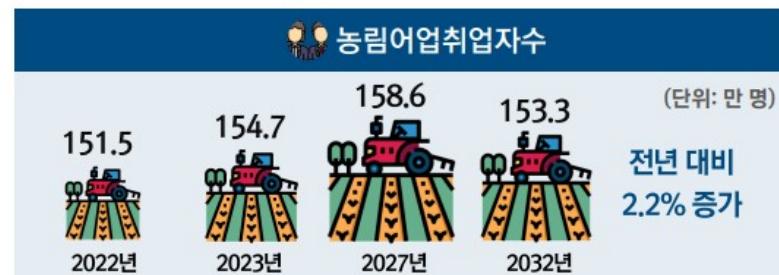
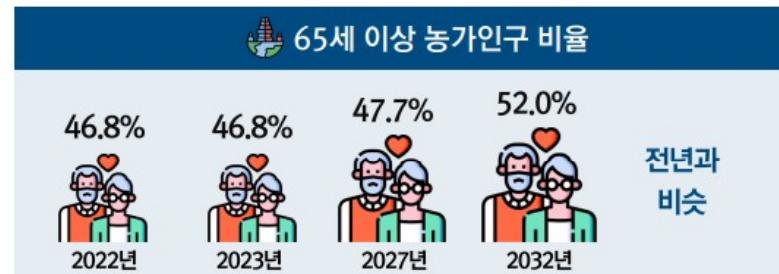
출처 : 중앙일보, 이미디어, 한국농촌경제연구원

01. 프로젝트 개요

배경 및 목적

국내 농업 현황 및 전망 - 농가 인구 전망

2023년 농가호수, 농가인구 하락 추세 전망



출처 : 한국농촌경제연구원 – 농업관측센터

01. 프로젝트 개요

배경 및 목적

스마트팜 - 스마트팜 개념

비닐하우스, 유리온실 등의 시설원예, 축사 등에 사물인터넷(IoT), 빅데이터, 인공지능, 로봇 등 정보통신기술(ICT)를 접목하여 작물과 가축의 생육환경을 원격, 자동으로 적정하게 유지, 관리할 수 있는 농장

스마트팜·ICT 농업 관련 관심도 25%증가

지난 7월 '한국판 뉴딜 종합계획' 발표와 함께
농촌을 K뉴딜의 핵심으로 꼽으며 스마트팜 보급은
앞으로도 더욱 확대(약 7,000헥타르)될 예정!

106,177 128,446 132,800

2018년
8대 선도 프로젝트*
'스마트팜' 포함

2019년
'노지 스마트농업
시범사업' 공모에
충북 괴산과
경북 안동 선정

2020년
2020년 7월 한국판 뉴딜 발표
농업분야 중 **스마트팜 확산** 선정 및
농촌의 디지털 기반 확충 목표

조사 개요: (기간) 2018.1.1~2020.12.10



출처 : 농림수산식품교육문화정보원, Priva.com

01. 프로젝트 개요

배경 및 목적

한국형 스마트팜 모델 - 1세대 모델 (원격 감시 및 제어)



농민이 영상을 통해 직접
“원격 수동제어”

- | **센서정보** | 기상정보: 온도, 습도, 풍향, 풍속
온실환경: 온도, 습도, CO₂
- | **장치제어** | 천창, 측창, 보온재, 유동팬, 환기팬

01. 프로젝트 개요

배경 및 목적

한국형 스마트팜 모델 - 1세대 모델 (원격 감시 및 제어)



“스마트 기술로 농작업 편리성 향상”
온실 환경관리에 매여 있었던 시간과 장소의 구속에서 해방



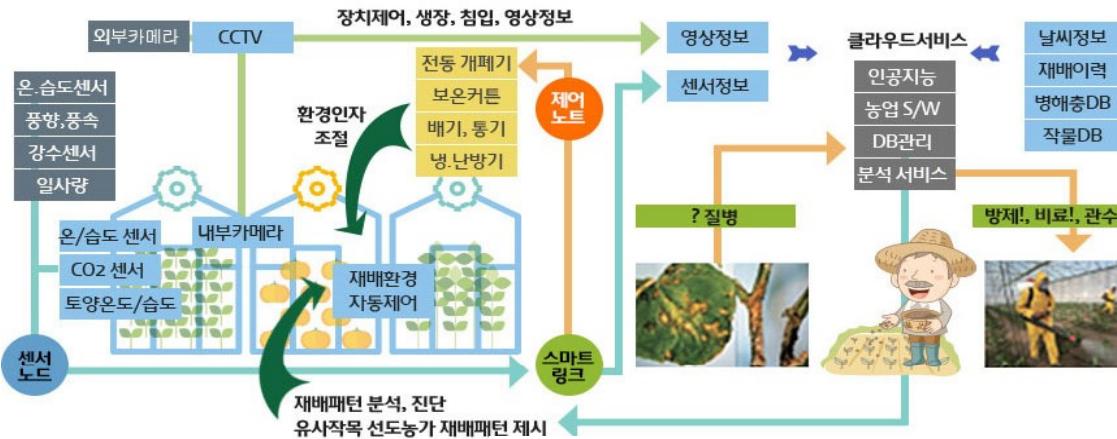
“원격 수동제어”

- 센서정보: 온도, 습도, 풍향, 풍속
온실환경: 온도, 습도, CO₂
- 장치제어: 천창, 측창, 보온재, 유동팬, 환기팬

01. 프로젝트 개요

배경 및 목적

한국형 스마트팜 모델 - 2세대 모델 (복합 환경제어 및 클라우드서비스)



작물의 지상부/지하부 생육환경을
“자동제어”

- | 센서정보 | 기상정보: 온도, 습도, 풍향, 풍속, 일사
온실환경: 온도, 습도, CO₂
- | 장치제어 | 천창, 측창, 보온재, 유동팬, 환기팬
관수, 양액공급

01. 프로젝트 개요

배경 및 목적

한국형 스마트팜 모델 - 2세대 모델 (복합 환경제어 및 클라우드서비스)



“스마트 기술로 생산성과 품질향상”
Big-Data 분석과 지능적 처방으로 농사기술의 상위 평준화



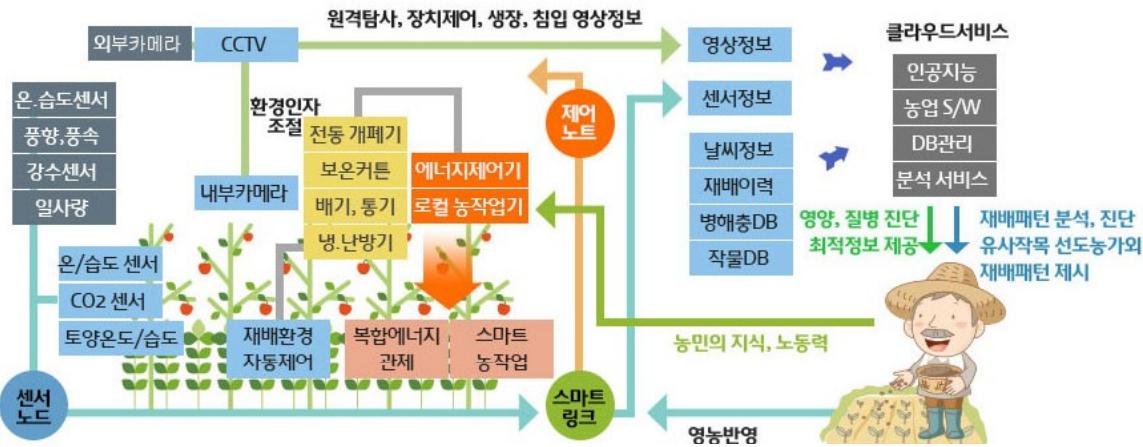
작물의 지상부/지하부 생육환경을
“자동제어”

- I 센서정보 I 기상정보: 온도, 습도, 풍향, 풍속, 일사
온실환경: 온도, 습도, CO₂
- I 장치제어 I 천창, 측창, 보온재, 유동팬, 환기팬
관수, 양액공급

01. 프로젝트 개요

배경 및 목적

한국형 스마트팜 모델 - 3세대 모델 (복합에너지관리 및 스마트 농작업)



스마트 온식 시스템의 “최적 에너지관리와 로봇 농작업”

센서정보
기상정보: 온도, 습도, 풍향, 풍속, 일사
온실환경: 온도, 습도, CO₂, 토양온도,
수분, EC, 작물진단센서,
에너지관제센서, 로봇향법센서

장치제어
천창, 측창, 보온재, 유동팬, 환기팬
관수, 양액공급, 로봇농작업기,
에너지관제시스템

01. 프로젝트 개요

배경 및 목적

한국형 스마트팜 모델 - 3세대 모델 (복합에너지관리 및 스마트 농작업)



“한국형 스마트 온실로 농산업 성장동력화”
국제 규격 적용과 부품 표준화로 글로벌 시장 진출



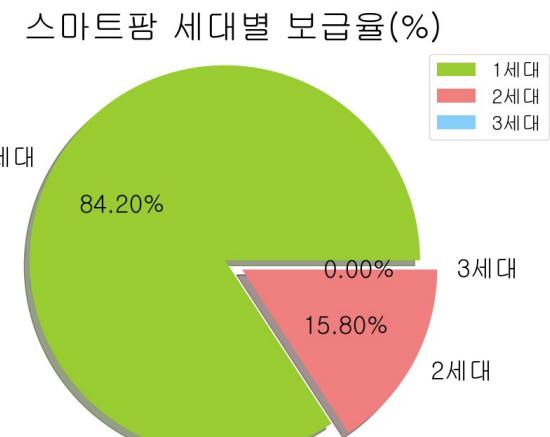
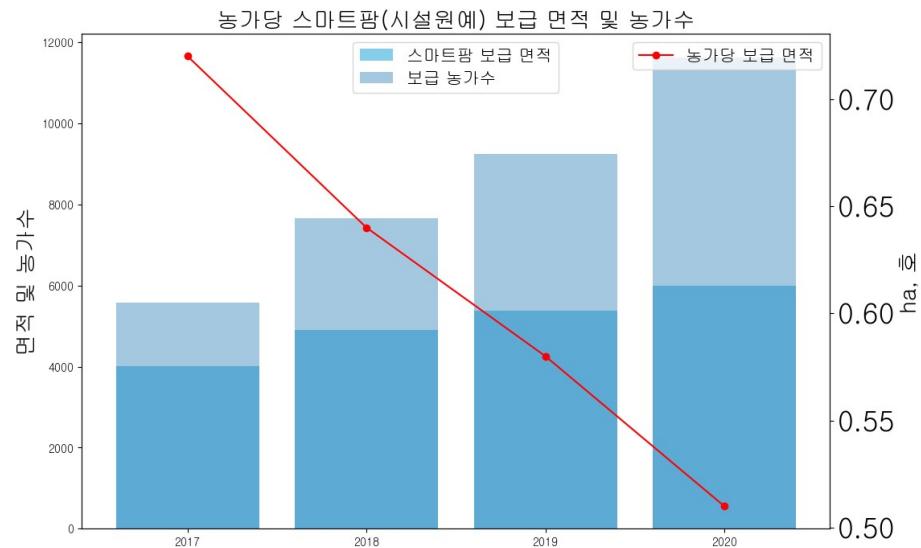
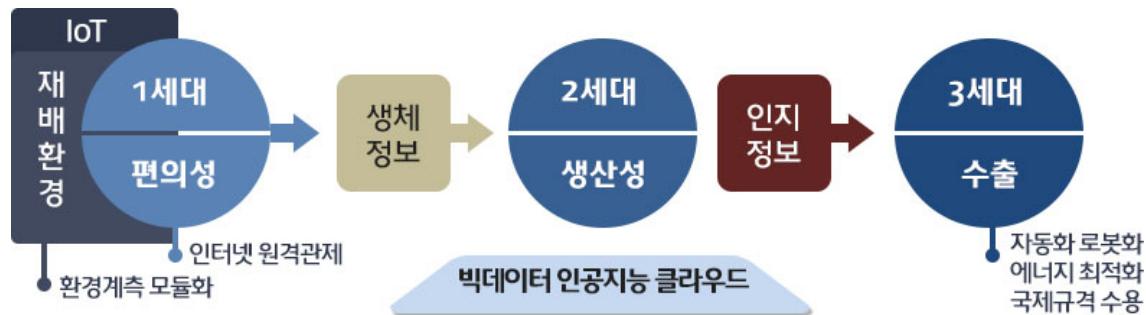
스마트 온식 시스템의
“최적 에너지관리와 로봇 농작업”

- | **센서정보** | 기상정보: 온도, 습도, 풍향, 풍속, 일사
온실환경: 온도, 습도, CO₂, 토양온도,
수분, EC, 작물진단센서,
에너지관제센서, 로봇향법센서
- | **장치제어** | 천창, 측창, 보온재, 유동팬, 환기팬
관수, 양액공급, 로봇농작업기,
에너지관제시스템

01. 프로젝트 개요

배경 및 목적

스마트팜 세대별 국내 모델 현황 - 국내 스마트팜 농가 보급율



출처 : 농촌진흥청, 국회예산정책처

01. 프로젝트 개요

주제 선정

토마토 스마트팜 AI 모델 구축

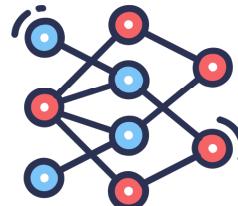
토마토 스마트팜 데이터를 이용한 품질 예측 모델, 질병 진단 및 생육단계 판단 모델



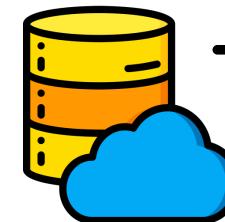
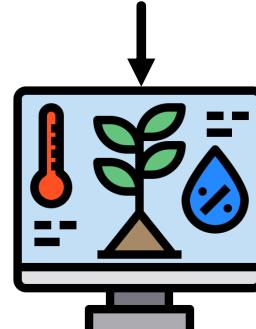
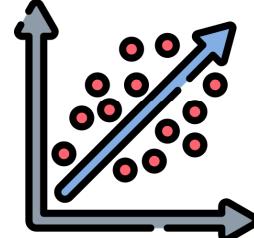
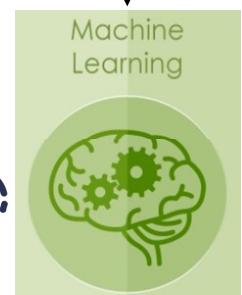
농촌진흥청



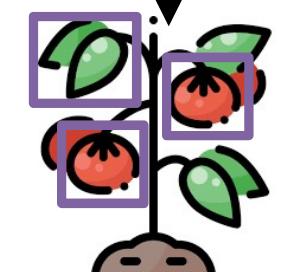
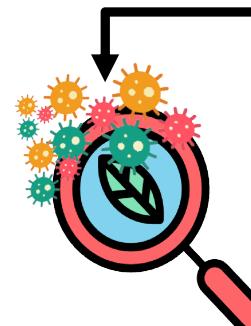
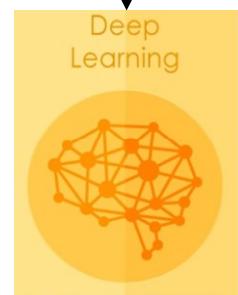
스마트팜 데이터 마트
SMARTFARM DATAMART



Machine
Learning



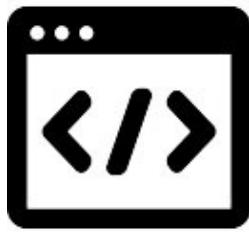
Deep
Learning



출처 : 농촌진흥청, 농림수산식품교육문화정보원, 한국지능정보사회진흥원

01. 프로젝트 개요

개발환경



Language



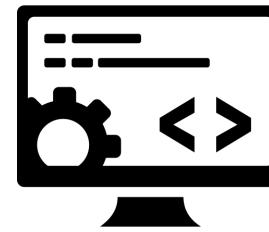
HTML



CSS



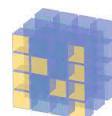
SQL



Tool



library



NumPy



web development,
one drop at a time



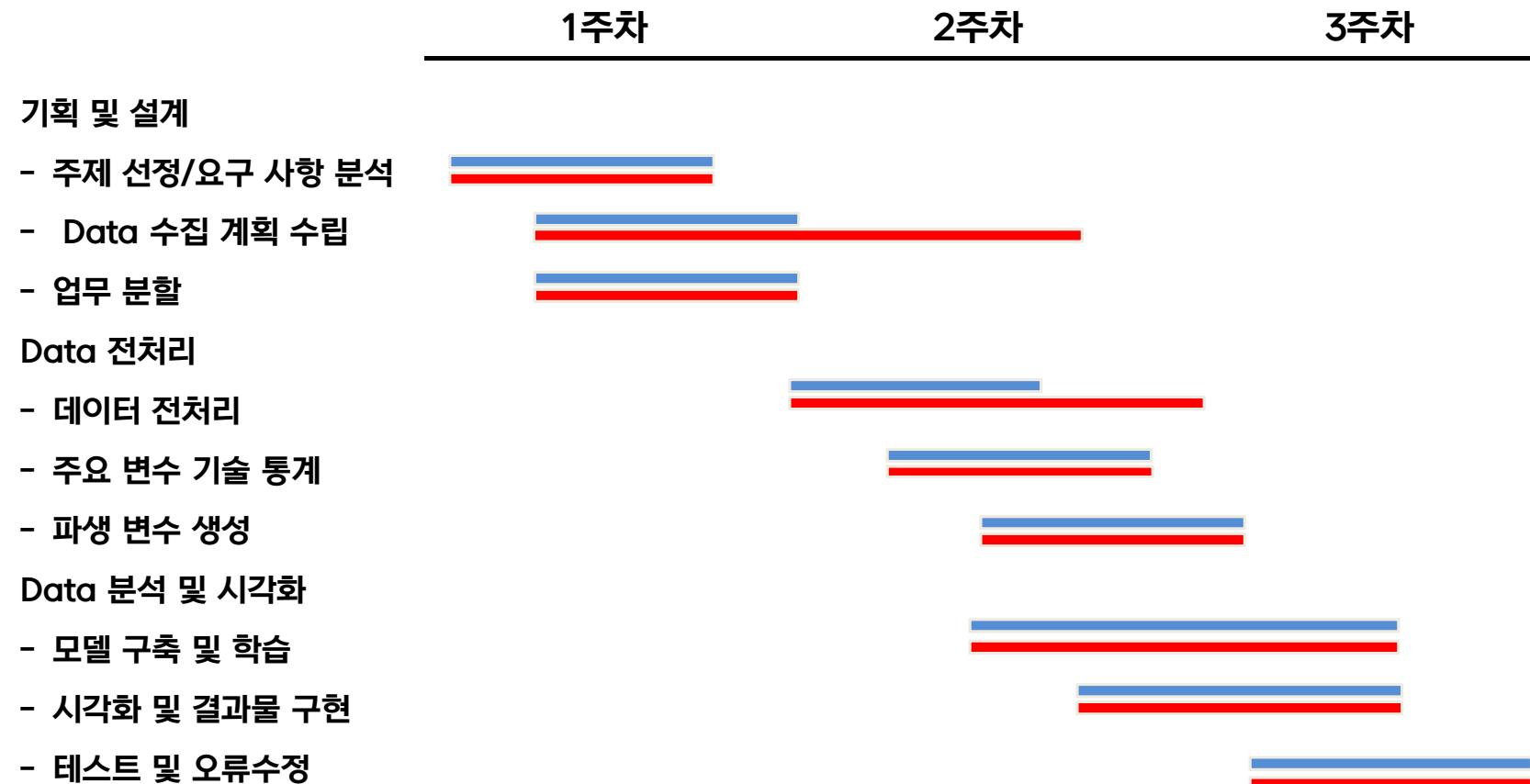
TensorFlow

출처 : Google images

01. 프로젝트 개요

계획
소요기간

프로젝트 진행 일정



02. 역할 및 담당

윤 진석

팀 장

프로젝트 관리 / 데이터 전처리
분석 및 생장예측 모델 구현



김 현경

팀 원

데이터 수집 / 전처리
분석 및 생육 상태 판단 모델 구축



나 석원

팀 원

데이터 수집 / 전처리 / 시각화
분석 및 생장예측 모델 구현
PPT / 발표



송 호준

팀 원

데이터 수집 / 전처리
분석 및 생장예측 모델 구현
프론트 엔드 web 구현



정 승빈

팀 원

데이터 수집 / 전처리
생육 상태 판단 모델 구축
PPT / 발표



황 준하

팀 원

데이터 수집 / 전처리
질병진단 모델 구현



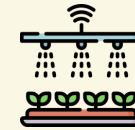
03. 데이터 베이스

Maria DB 구축

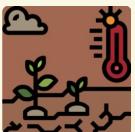
급액 데이터
3,446개



배액 데이터
3,446개



환경 데이터
3,446개



급액, 배액
제외한 데이터
3,446개



이미지 정보
34,506개



이미지가
가진 생체 정보
34,506개



04. 품질 예측 모델

데이터 확인

Raw 데이터 확인

데이터 구성

00_(환경제어기, 양액기)	2023-06-15 오후 1:16	파일 풀더
01_(함수율저울, 열화상센서, 작물환경센서)	2022-07-14 오후 5:15	파일 풀더

```
df_raw.columns  
Index(['날짜', '시간', '설정량1', '설정량2', '설정량3', '잔여량1', '잔여량2',  
'잔여량3', '누적량1', '누적량2', '누적량3', '시작', '종료',  
'설정EC(dS)', '현재EC(dS)', '설정PH(pH)', '현재PH(pH)', '일출시간',  
'일몰시간', '현재일사(₩)', '누적일사(J)', 'J/Day'],  
dtype='object')
```

00_(환경제어기, 양액기)

```
df_raw1.columns  
Index(['time', 'EC급액', 'EC배액', 'pH급액', 'pH배액', '저울급액', '저울배액', '온도토양', '온도급액',  
'온도배액', '내부이산화탄소', '내부광량', '내부PT100온도센서1번건구', '내부PT100온도센서2번습구',  
'내부PT100온도센서3번', '내부PT100센서를이용한계산습도'],  
dtype='object')
```

01_(함수율저울, 열화상센서, 작물환경센서)

독립변수 (feature) 선정 * 농촌진흥청 생육상태별 최적환경설정 데이터셋 참고

생육상태별 최적환경설정 데이터셋

생육증기(11~12월)의 적정 생육을 위한 환경설정

누적일사량 (J/m ² /day)	생산량 (kg/3.3m ²)	생장길이 (cm)	줄기굵기 (mm)	화방높이 (cm)	일일 평균온도 (°C)	주간 평균습도 (%)	잔존 CO ₂ (ppm)	급액횟수 (회)	급액 EC (dS/m)	급액 pH	1회 급액량 (cc/day)
----------------------------------	--------------------------------	--------------	--------------	--------------	-----------------	----------------	-----------------------------	-------------	-----------------	-------	--------------------

04. 품질 예측 모델

데이터 탐색

Raw 데이터 확인

독립변수(환경데이터) 결측치 확인(%)

날짜	0
EC급 액	0
EC배 액	0
pH급 액	0
pH배 액	0
내부이산화탄소	0
평균온도	0
J/Day	0

1번 농장

날짜	0.0
J/Day	100.0
평균온도	0.0
EC급 액	0.0
EC배 액	0.0
pH급 액	0.0
pH배 액	0.0
내부이산화탄소	0.0
온도토양	0.0
온도급액	100.0
온도배액	100.0

2번 농장

날짜	0.0
J/Day	70.9
평균온도	70.9
EC급 액	100.0
EC배 액	100.0
pH급 액	100.0
pH배 액	100.0
내부이산화탄소	70.9
온도토양	100.0
온도급액	100.0
온도배액	100.0

3번 농장

날짜	0.00
J/Day	0.46
평균온도	98.43
EC급 액	98.43
EC배 액	98.43
pH급 액	98.43
pH배 액	98.43
내부이산화탄소	98.43
온도토양	98.43
온도급액	98.43
온도배액	98.43

4번 농장

날짜	0.0
J/Day	0.0
평균온도	0.0
EC급 액	0.0
EC배 액	0.0
pH급 액	0.0
pH배 액	0.0
내부이산화탄소	0.0
온도토양	0.0
온도급액	0.0
온도배액	0.0

5번 농장

날짜	0.00
J/Day	15.33
평균온도	67.81
EC급 액	67.81
EC배 액	67.81
pH급 액	67.81
pH배 액	67.81
내부이산화탄소	67.81
온도토양	67.81
온도급액	67.81
온도배액	67.81

6번 농장

04. 품질 예측 모델

데이터 탐색

Raw 데이터 확인

독립변수(환경데이터) 결측치 확인(%)

날짜	0
EC급 액	0
EC배 액	0
pH급 액	0
pH배 액	0
내부이산화탄소	0
평균온도	0
J/Day	0

1번 농장

날짜	0.0
J/Day	100.0
평균온도	0.0
EC급 액	0.0
EC배 액	0.0
pH급 액	0.0
pH배 액	0.0
내부이산화탄소	0.0
온도토양	0.0
온도급액	100.0
온도배액	100.0

2번 농장

날짜	0.0
J/Day	70.91
평균온도	0.0
EC급 액	0.0
EC배 액	0.0
pH급 액	0.0
pH배 액	0.0
내부이산화탄소	0.0
온도토양	100.0
온도급액	100.0
온도배액	100.0

사용불가

3번 농장

날짜	0.0
J/Day	0.43
평균온도	0.0
EC급 액	0.0
EC배 액	0.0
pH급 액	0.0
pH배 액	0.0
내부이산화탄소	0.0
온도토양	0.0
온도급액	0.0
온도배액	0.0

사용불가

4번 농장

날짜	0.0
J/Day	0.0
평균온도	0.0
EC급 액	0.0
EC배 액	0.0
pH급 액	0.0
pH배 액	0.0
내부이산화탄소	0.0
온도토양	0.0
온도급액	0.0
온도배액	0.0

5번 농장

날짜	0.00
J/Day	15.33
평균온도	67.81
EC급 액	67.81
EC배 액	67.81
pH급 액	67.81
pH배 액	67.81
내부이산화탄소	67.81
온도토양	67.81
온도급액	67.81
온도배액	67.81

6번 농장

04. 품질 예측 모델

데이터 탐색

Raw 데이터 확인

종속변수(토마토 품질측정데이터) 확인 (21년 45주차 ~ 22년 6주차)

수집항목	과폭	과고	과중			경도	수분율
			g	Brix %	0-14		
개체번호	cm	cm	kgf	%			
1	7.4	5.8	205	4.1	4	2.65	71.25
2	7.9	6.2	210	4.7	4	2.7	72.5
3	7.7	6	205	4.7	4	2.7	71.25
4	7.8	5.7	206	4.7	4.5	2.72	75
...
97	8	6.4	243	5.1	4	2.961964	70
98	8.2	6.5	260	4.8	4	2.98326	76.5
99	7.5	6.6	233	5.4	4	3.004556	71.25
100	9	6.5	360	5.1	4	3.025851	71.25
평균값	8.22	6.388	245.85	4.489	4.025	2.713516	71.8965

종속변수 컬럼 선정

주차 별 Data 100개

X

14주차

↓

총 1400개

04. 품질 예측 모델

데이터 탐색

데이터 전처리

「독립변수 Dataframe」 ↔ 「종속변수 Dataframe」 프레임 합치기

독립변수		종속변수				
1	166	주차 과중 당도 산도				
2	168	1	45	205	4.1	4
3	167	2	45	210	4.7	4
4	96	3	45	205	4.7	4
41	14	4	45	206	4.7	4.5
42	172	5	45	223	4.3	4
43	203
44	200	96	45	156	4.8	3.5
45	466	97	45	243	5.1	4
46	523	98	45	260	4.8	4
47	377	99	45	233	5.4	4
48	203	100	45	360	5.1	4
49	335					
50	278					
51	118					
52	168					

공통된 주차

주차별 종속변수 Data 개수에 맞추어
독립변수 Data Random추출 및 concat

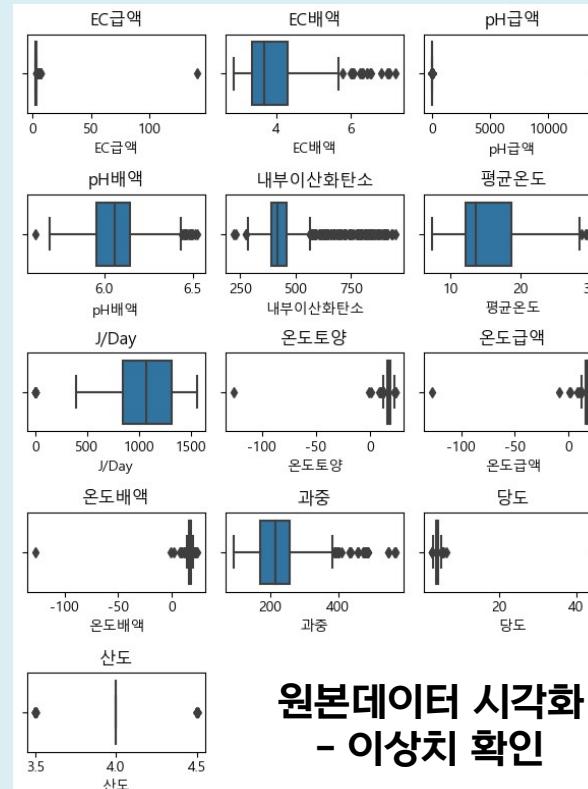
Name: 수자, dtype: int64

04. 품질 예측 모델

데이터 탐색

```
EC급액      28.34
EC배액      55.52
pH급액      13.71
pH배액      11.96
내부이산화탄소 51.67
평균온도    65.80
J/Day       78.26
온도토양    21.15
온도급액    83.19
온도배액    82.86
과중        0.00
당도        0.00
산도        0.00
dtype: float64
```

결측치 비율 (%)



원본데이터 시각화
- 이상치 확인

04. 품질 예측 모델

데이터 탐색

```
tom_origin["온도"].max()
```

44.5

Brix : 44.5 → 4.5
(코카콜라 Brix : 11)

```
32    -126.250000
53     -0.339782
Name: 온도토양, dtype: float64
-----
32    -127.267857
Name: 온도급액, dtype: float64
-----
32    -127.375
Name: 온도배액, dtype: float64
```

0도 이하 온도 → null

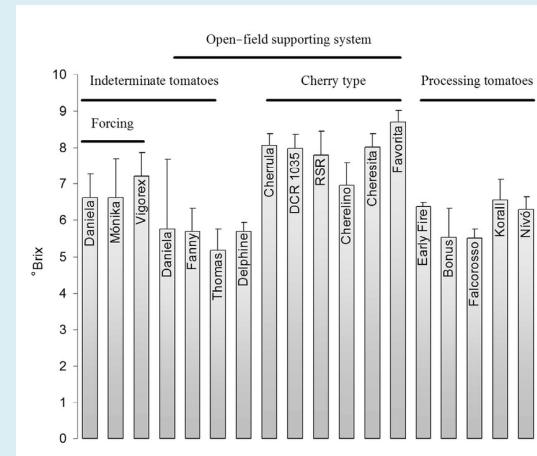
데이터 전처리 – 원본 데이터 이상치 수정

이상치 제한 기준 설정

환경데이터 Set

- J/Day : 600 ~ 1300
- EC 급액 : 2.5 ~ 3.5
- EC 배액 : 3.5 ~ 5.0
- pH 급액 : 5.0 ~ 6.5
- pH 배액 : 5.0 ~ 7.0
- 이산화탄소농도 : 400 ~ 600
- 평균온도, 온도토양/급액/배액 : 15 ~ 30

토마토 Brix : 4 ~ 12



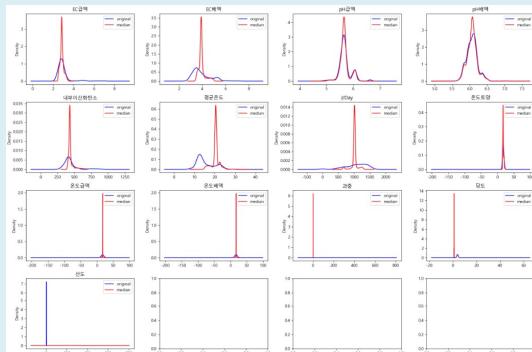
출처 : 농총진흥청 , (L. Helyes, Z. Pék, and A. Lugasi, 2008)

04. 품질 예측 모델

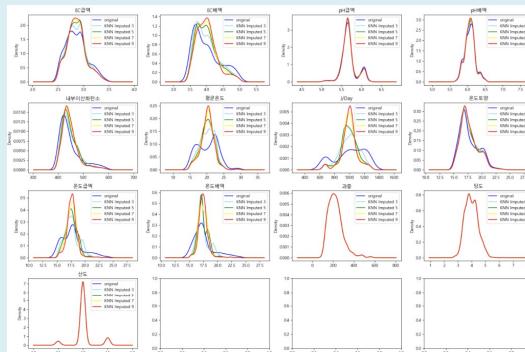
데이터 탐색

데이터 전처리 - 결측치 대체 방법

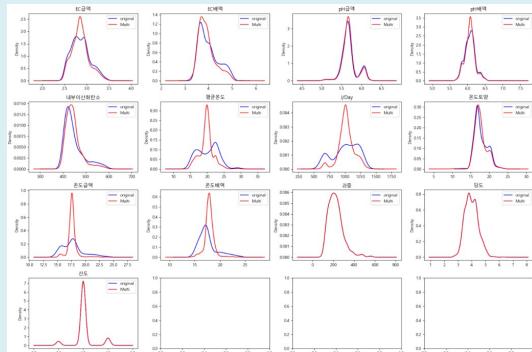
Simple imputer (평균값/중앙값)



KNN imputer



Multivariate feature imputation



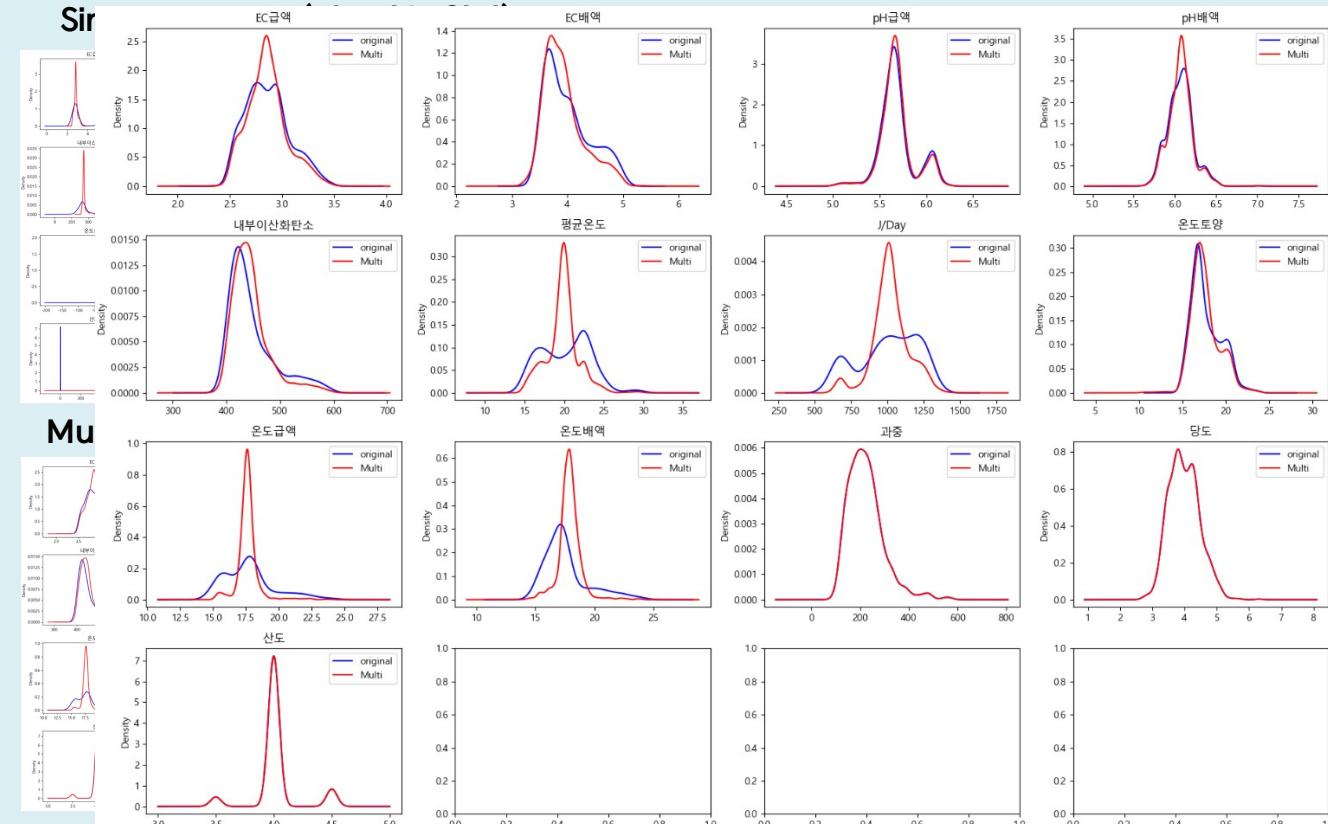
```
from sklearn.experimental import enable_iterative_imputer
from sklearn.impute import IterativeImputer
imputer = IterativeImputer(max_iter=10000, random_state = 0)
tom_mul = imputer.fit_transform(tom_origin)
tom_mul
```

max_iter = 10000, random_state = 0

04. 품질 예측 모델

데이터 탐색

데이터 전처리 - 결측치 대체 방법



ter
 $\hat{z} = \theta$)

04. 품질 예측 모델

데이터 탐색

데이터 전처리 - 파생변수 추가

	EC급액	EC배액	pH급액	pH배액	내부이산화탄소	평균온도	J/Day	온도토양	온도급액	온도배액	EC비율	pH비율	과증	당도	산도
0	2.957932	4.006233	6.076204	5.830677	447.373519	22.491605	1176.000000	16.894489	17.831317	17.304435	1.354403	0.959592	205.0	4.1	4.0
1	2.957932	4.006233	6.076204	5.830677	447.373519	22.491605	1107.000000	16.894489	17.831317	17.304435	1.354403	0.959592	210.0	4.7	4.0
2	2.864418	4.006233	5.684571	6.066748	447.373519	20.192301	1297.000000	17.751303	17.730408	17.545886	1.398620	1.067231	205.0	4.7	4.0
3	2.864418	4.006233	5.785397	5.952397	447.373519	16.703968	1004.820565	17.751303	17.730408	17.545886	1.398620	1.028866	206.0	4.7	4.5
4	2.957932	4.006233	6.076204	5.830677	447.373519	22.491605	1176.000000	16.894489	17.831317	17.304435	1.354403	0.959592	223.0	4.3	4.0

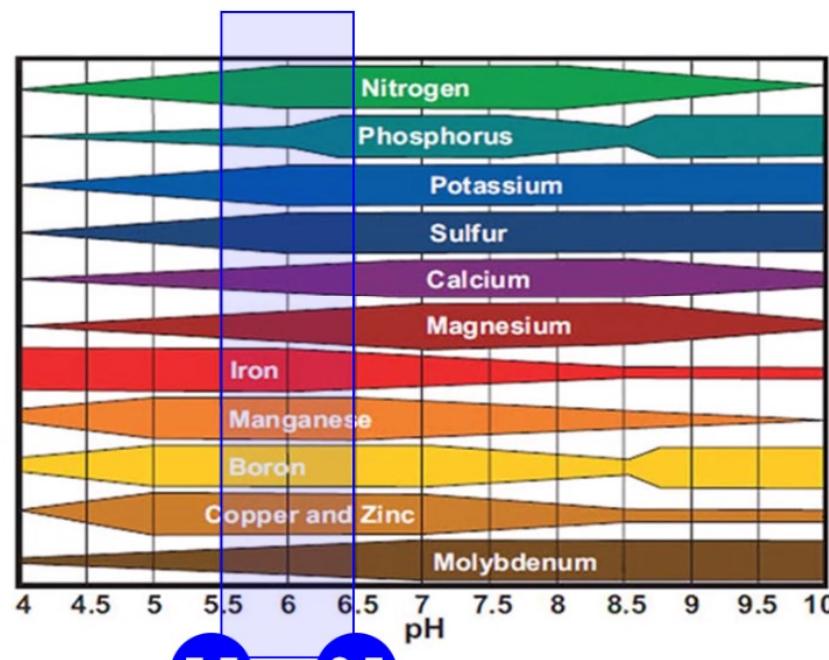
EC 급액·배액수치, pH 급액·배액수치의 차이는
배양액에 포함된 무기이온 흡수율에 영향을 줌
토마토 성장과 품질에 영향을 미침
→ 「EC 비율 / pH비율」 파생변수 추가

04. 품질 예측 모델

데이터 탐색

데이터 전처리 - 파생변수 추가

	EC급액	EC배액
0	2.957932	4.006233
1	2.957932	4.006233
2	2.864418	4.006233
3	2.864418	4.006233
4	2.957932	4.006233



물	과증	당도	산도
12	205.0	4.1	4.0
12	210.0	4.7	4.0
11	205.0	4.7	4.0
16	206.0	4.7	4.5
12	223.0	4.3	4.0

- 질소
- 인
- 인산
- 유황
- 칼슘
- 마그네슘
- 철
- 망간
- 붕소
- 구리&아연
- 몰리브덴

04. 품질 예측 모델

Model 생성

Model 구축 및 학습

예시) Ridge MODEL

```
<과중>  
파라미터 : {'alpha': 10.0}  
훈련 점수 : 0.14321478608992255  
테스트 점수: 0.21578937724533243  
rmse : 58.84050878677888
```

```
<당도>  
파라미터 : {'alpha': 100.0}  
훈련 점수 : 0.03675679984426068  
테스트 점수: 0.051946696799873515  
rmse : 0.4540635453679355
```

```
<산도>  
파라미터 : {'alpha': 1000.0}  
훈련 점수 : -0.0031222237595489367  
테스트 점수: -0.01333518345476925  
rmse : 0.1781367363168933
```

Mean

```
<과중>  
파라미터 : {'alpha': 10.0}  
훈련 점수 : 0.14281056570269735  
테스트 점수: 0.1922023337433103  
rmse : 59.71883900458697
```

```
<당도>  
파라미터 : {'alpha': 1.0}  
훈련 점수 : 0.03679337087243317  
테스트 점수: 0.04586109920356929  
rmse : 0.45551854153342003
```

```
<산도>  
파라미터 : {'alpha': 1000.0}  
훈련 점수 : -0.003520026925697484  
테스트 점수: -0.013407142906317837  
rmse : 0.17814306117094597
```

Median

```
<과중>  
파라미터 : {'alpha': 1.0}  
훈련 점수 : 0.319751643306118  
테스트 점수: 0.40371625873863826  
rmse : 51.30817209760441
```

```
<당도>  
파라미터 : {'alpha': 100.0}  
훈련 점수 : 0.030928521672514187  
테스트 점수: 0.048688378067466604  
rmse : 0.4548431506594999
```

```
<산도>  
파라미터 : {'alpha': 1000.0}  
훈련 점수 : -0.004093330112243487  
테스트 점수: -0.0144626980971152  
rmse : 0.17823581307949557
```

KNN
Imputation

```
<과중>  
파라미터 : {'alpha': 0.01}  
훈련 점수 : 0.7095418320371063  
테스트 점수: 0.7558331202543773  
rmse : 32.83246088836895
```

```
<당도>  
파라미터 : {'alpha': 0.01}  
훈련 점수 : 0.4965851275084784  
테스트 점수: 0.6427433079419861  
rmse : 0.27873431902325196
```

```
<산도>  
파라미터 : {'alpha': 0.1}  
훈련 점수 : 0.14915782752884943  
테스트 점수: 0.04695914072125462  
rmse : 0.1727558212721164
```

Multi
Imputation

- KNN, Linear, Lasso, Ridge, ElasticNet, AdaBoost, GradientBoost, RandomForest, SVR 등 다양한 회귀모델 테스트
- random.seed 조정을 통한 최적 모델 탐색

04. 품질 예측 모델

Model 생성

Model 구축 및 학습 - 과중·당도 예측 ML Model

Ridge regression Model

〈과중〉

파라미터 : {'alpha': 0.0001}
훈련 점수 : 0.9951702242843163
테스트 점수: 0.9958381324654477
rmse : 4.892308237100221

〈당도〉

파라미터 : {'alpha': 0.01}
훈련 점수 : 0.9675564762749337
테스트 점수: 0.953183980958138
rmse : 0.1031485764884748

과중 예측 Model 성능

당도 예측 Model 성능

04. 품질 예측 모델

Model 생성

Model 구축 및 학습 - 산도 분류 Model

DNN(Deep Neural Network) Model

```
4.0    1015
4.5    117
3.5    64
Name: 산도, dtype: int64
```

산도는 연속변수이나
현 Data에서는 3개의
label로 카테고리화

Model: "sequential_35"		
Layer (type)	Output Shape	Param #
flatten_35 (Flatten)	(None, 12)	0
dense_152 (Dense)	(None, 64)	832
dropout_117 (Dropout)	(None, 64)	0
dense_153 (Dense)	(None, 64)	4160
dropout_118 (Dropout)	(None, 64)	0
dense_154 (Dense)	(None, 128)	8320
dropout_119 (Dropout)	(None, 128)	0
dense_155 (Dense)	(None, 3)	387
=====		
Total params: 13,699		
Trainable params: 13,699		
Non-trainable params: 0		

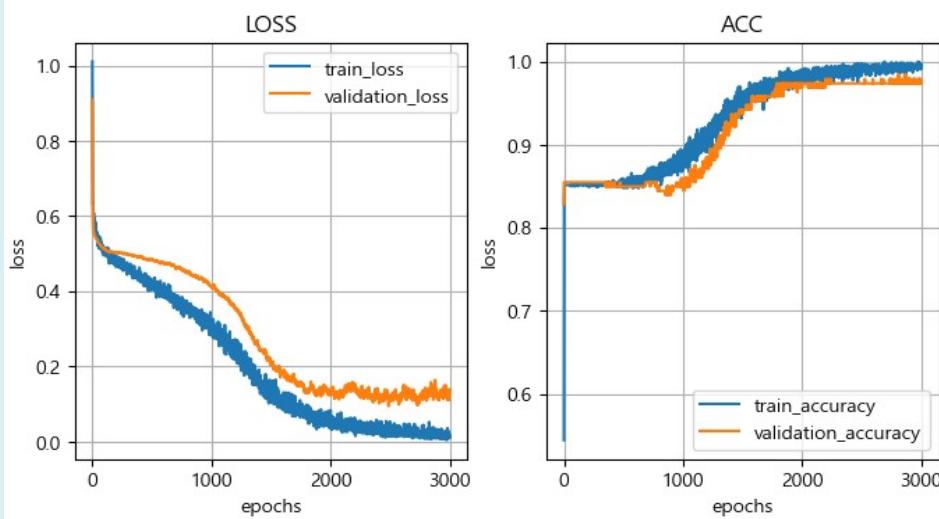
분류 DNN 모델 구축

04. 품질 예측 모델

Model 생성

Model 구축 및 학습 - 산도 분류 Model

DNN 모델 성능



▶ Train Score

loss: 0.0139 - accuracy: 0.9961

▶ Evaluate

loss: 0.1609 - accuracy: 0.9667

▶ Validation Score

val_loss: 0.1392 - val_accuracy: 0.9740

05. YOLO 모델

YOLO 모델 소개

Model 설명 - 질병 진단 및 생육단계 판단 모델

YOLO v8
Detect

- 질병 상태 판별 YOLO

잎 사진으로
질병 양성 여부를 판별하고
질병이라면 진행 단계를 출력

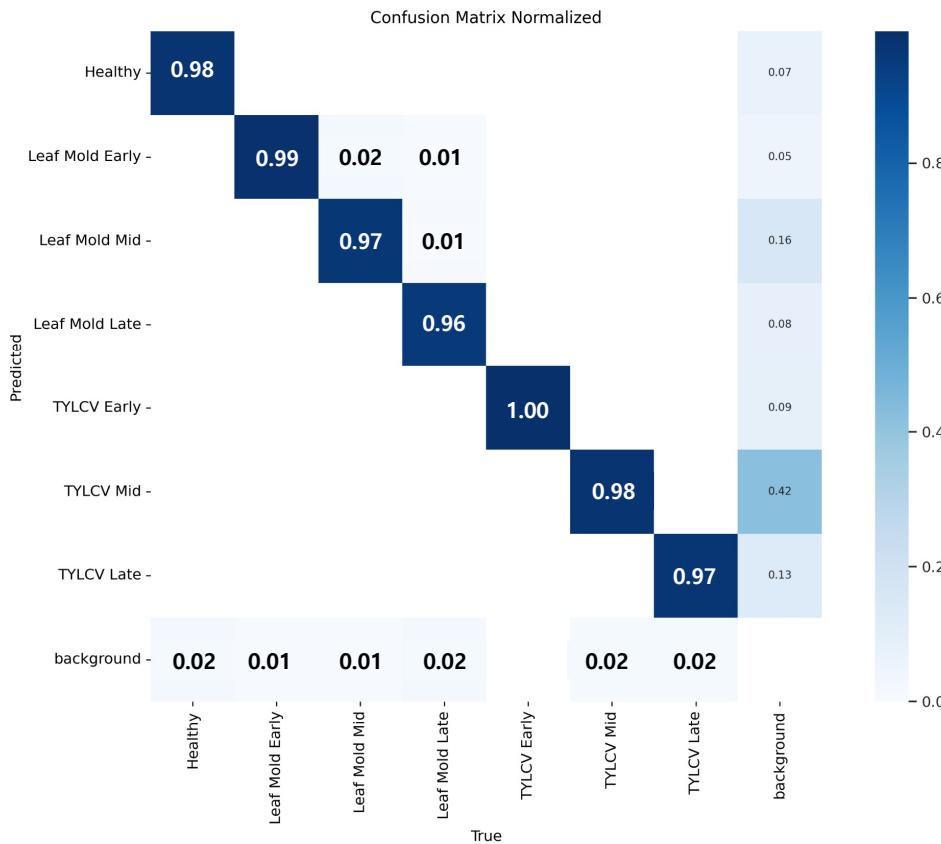
- 생육 단계 판별 YOLO

꽃과 열매 사진으로
생장 단계를 출력

05. YOLO 모델

질병 진단 모델

질병 진단 모델 성능



- 하이퍼 파라미터
 - model: yolov8n
 - epochs: 100
 - patience: 40
 - batch: 32
 - imgsz: 416

05. YOLO 모델

질병 진단 모델

질병 진단 모델 - 데이터셋 및 전처리

원본 데이터셋

- [라벨]11.토마토_0.정상.zip
- [라벨]11.토마토_1.질병.zip
- [라벨]11.토마토_9.증강.zip
- 11.토마토_0.정상.zip
- 11.토마토_1.질병.zip
- 11.토마토_9.증강_(1).zip
- 11.토마토_9.증강_(2).zip



라벨링 데이터

```
{'description': {'image': 'V006_77_1_18_11_03_13_1_3248b_20201005_57_a0000.jpg',  
'date': '2020/10/05',  
'worker': '',  
'height': 3024,  
'width': 4032,  
'task': 77,  
'type': 1,  
'region': 4,  
'original': 'V006_77_1_18_11_03_13_1_3248b_20201005_57.jpg'},  
'annotations': {'disease': 18},  
'crop': 11,  
'area': 3,  
'grow': 13,  
'risk': 1,  
'points': [{'xtl': 908, 'ytl': 712, 'xbr': 3104, 'ybr': 2108}],  
'augmented': {'brightness': 0.9370387671475437,  
'contrast': 0.8037306995298512,  
'saturation': 0.9335766146012883,  
'hue': 0.0921173427008942,  
'transpose': 0,  
'hflip': 0,  
'vflip': 1}}
```

```
train_aug["disease"].value_counts()
```

```
19    10550  
18     8460  
1      10  
15     10  
Name: disease, dtype: int64
```

19: 토마토 황화잎말이바이러스병
18: 토마토 잎곰팡이병

1: 가지 잎곰팡이병
15: 쥬키니호박 오이녹반 모자이크
바이러스

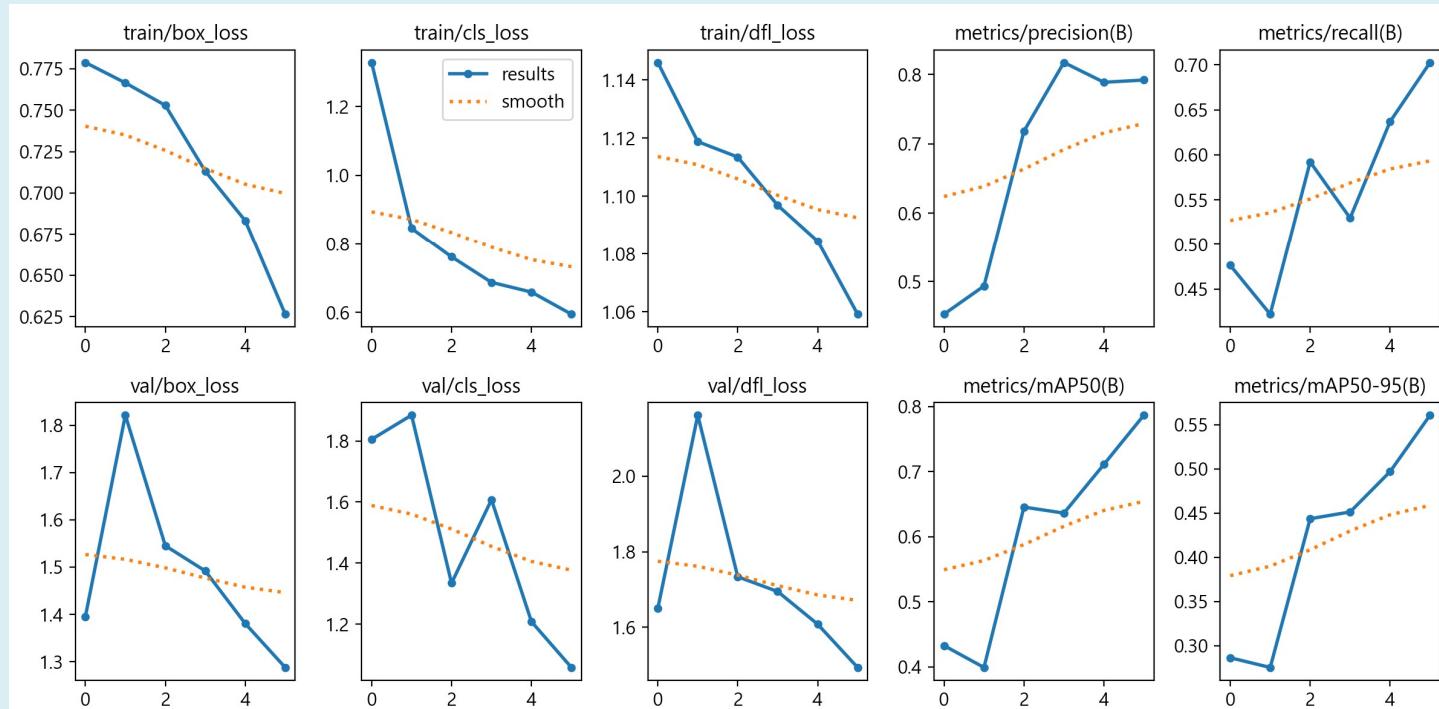
05. YOLO 모델

질병 진단 모델

질병 진단 모델 - 모델 구축

X: 잎, 열매, 꽃, 줄기 (79.2 GB) 모두 사용

Y: 정상, 질병

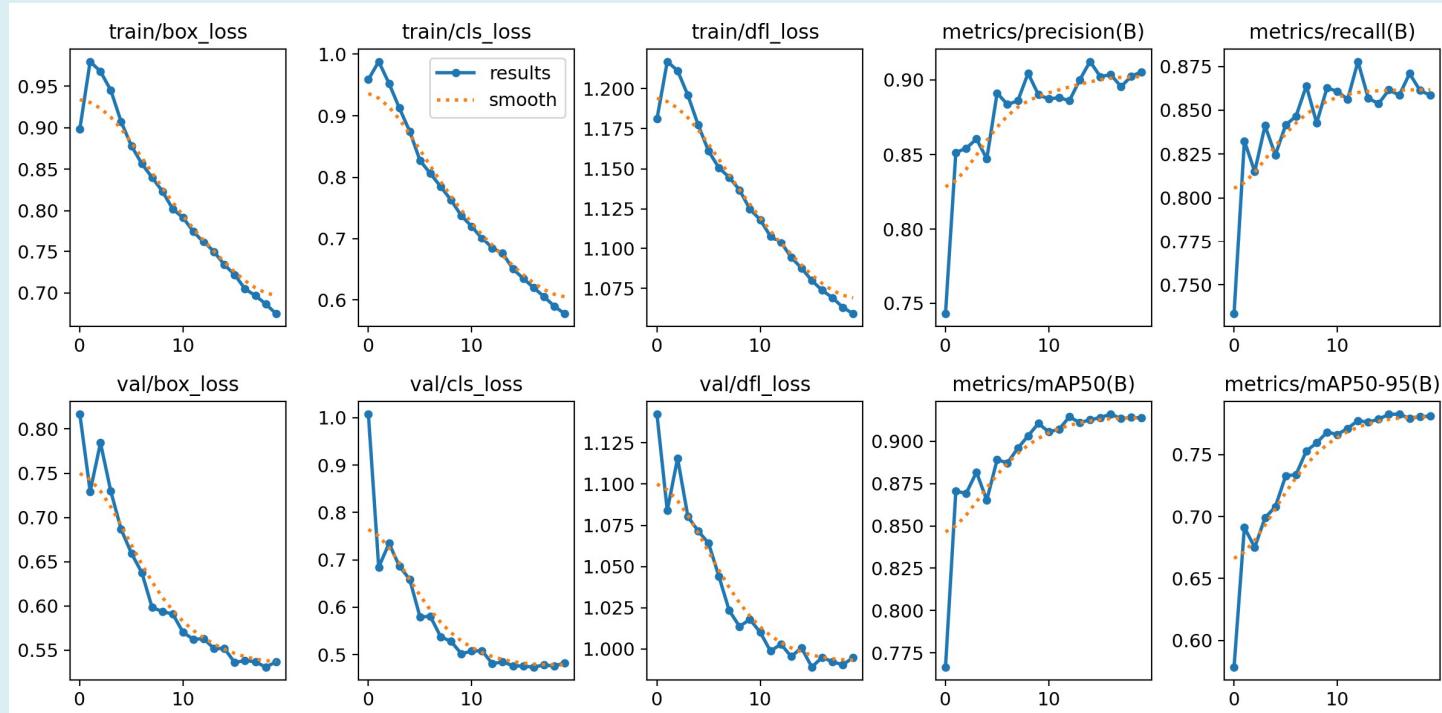


05. YOLO 모델

질병 진단 모델

질병 진단 모델 - 모델 구축

X: 열매, 꽃, 줄기 데이터 제거, $\frac{1}{4}$ 로 리사이즈, 최종 7.39GB, epoch=20
Y: 정상, TYLCV, Leaf Mold



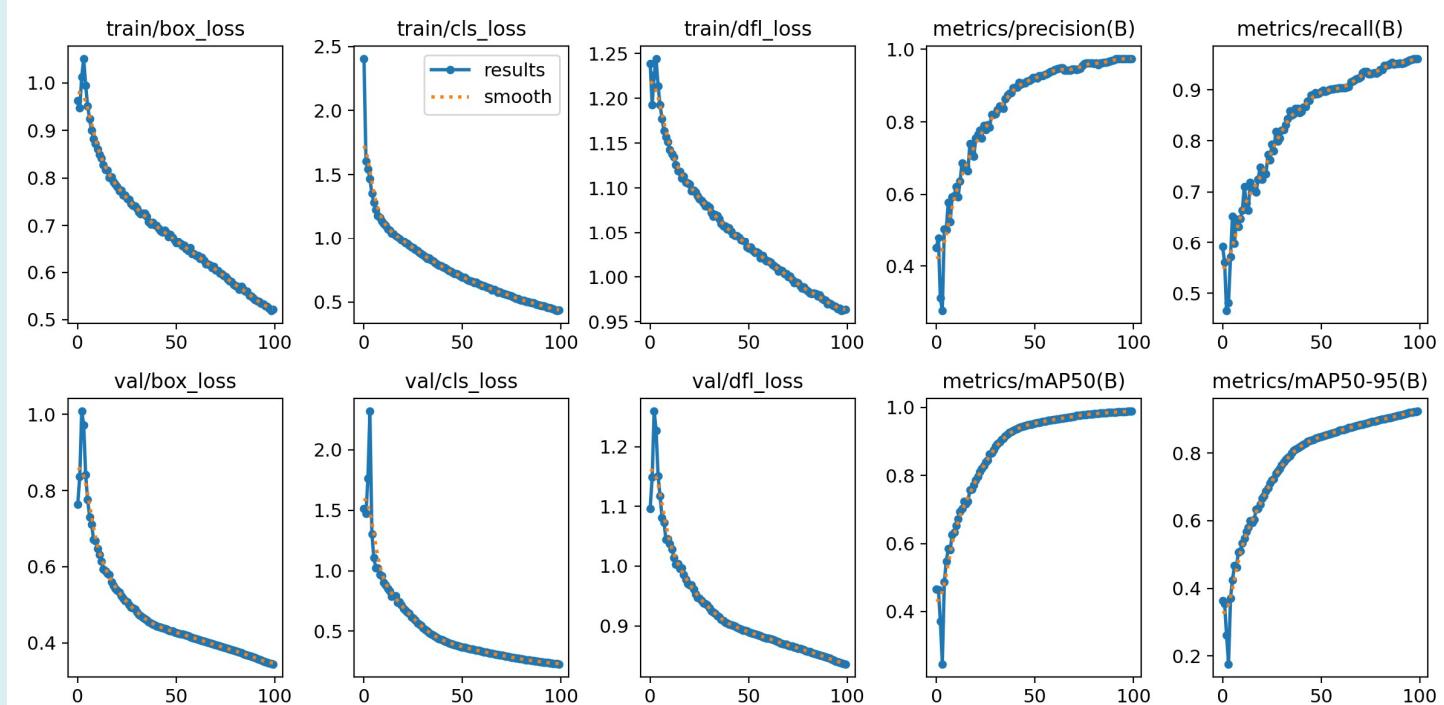
05. YOLO 모델

질병 진단 모델

질병 진단 모델 - 모델 구축

X: epoch=100

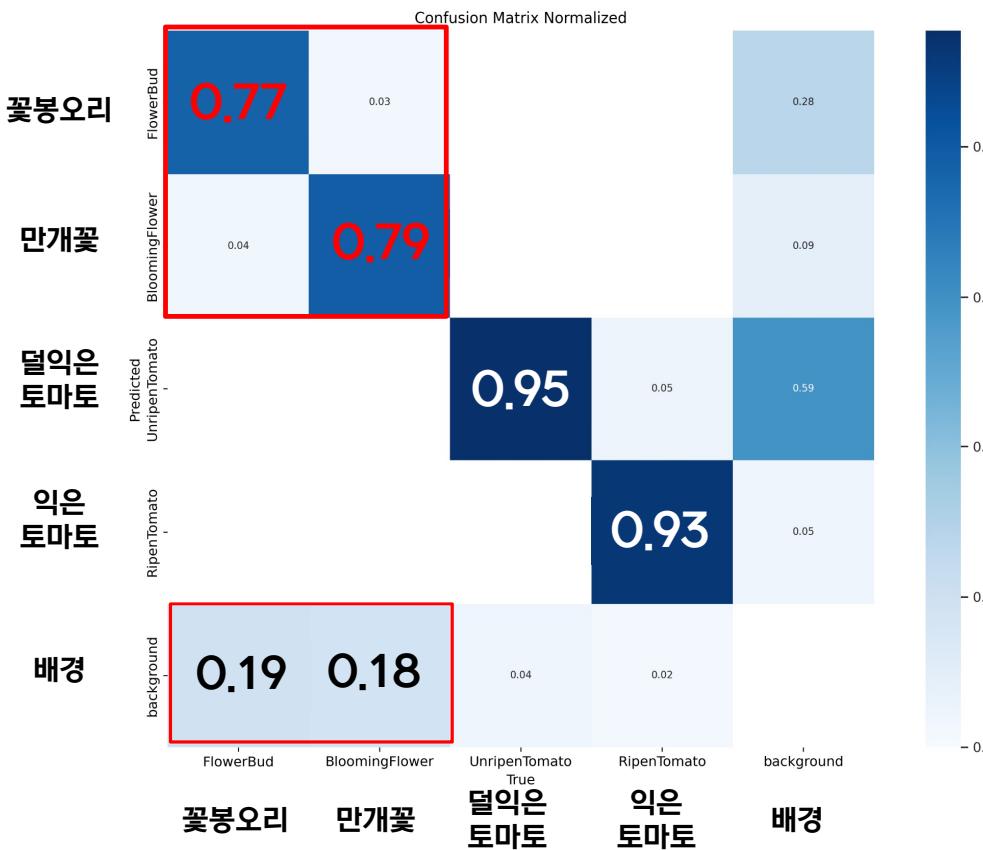
Y: 정상, TYLCV 초기 증기 말기, Leaf Mold 초기 증기 말기로 확장



06. YOLO 모델

생육 단계 판단 모델

생육 단계 판단 모델 - 모델 개요



- 하이퍼 파라미터
 - model: yolov8l
 - epochs: 31
 - patience: 5
 - batch: -1
 - imgsz: 640
 - device: 0
 - save_period: 5
 - exist_ok: True
 - cache: True
 - workers: 8

06. YOLO 모델

생육 단계 판단 모델

생육 단계 판단 모델 - 데이터셋 및 전처리

메인 데이터

TL_a1생장길이.zip
TL_a2화방노이.zip
TL_a3풀기두께.zip
TL_a4엽장엽폭.zip
TL_b개화군.zip
TL_c착과군.zip
TL_d1꽃_개화군.zip
TL_d2꽃_착과군.zip
TL_e열매.zip
TL_f만개꽃.zip
TS_a1생장길이.zip
TS_a2화방노이.zip
TS_a3풀기두께.zip
TS_a4엽장엽폭.zip
TS_b개화군.zip
TS_c착과군.zip
TS_d1꽃_개화군.zip
TS_d2꽃_착과군.zip
TS_e열매.zip
TS_f만개꽃.zip
TS_Timeseries.zip



서브 데이터

[라벨]11.토마토_0.정상.zip
[라벨]11.토마토_1.질병.zip
[라벨]잎마를병(토마토)_0.정상.zip
[라벨]잎마를병(토마토)_1.질병.zip
[라벨]잎마를병(토마토)_9.증강.zip
[원천]잎마를병(토마토)_0.정상.zip
[원천]잎마를병(토마토)_9.증강.zip
[원천]잎마를병(토마토)_질병.zip
11.토마토_0.정상.zip
11.토마토_1.질병.zip
Tomato-Mixed-4000x1660-(636 images)



1. 이미지 라벨 파일 전처리

json → txt

각 객체에 대한 폴리곤 좌표는 각각의 shape 값에 저장

```
10 # 도형들의 정보 출력
11 for i, shape in enumerate(data["shapes"]):
12     print(f"Shape {i + 1}:")
13     print(f"  Label: {shape['label']}")
14     print(f"  Points: {shape['points']}")
15     print(f"  Shape type: {shape['shape_type']}")\n16
```

Shape 1:
Label: tom_fruit_breaker_poly
Points: [[596.9090909090909, 828.9966329966329], [566.03367003367, 843.4747474747475], [547.5151515151515, 859.973063973064], [532.026936026936, 889.2659932659932], [528.666666666666666, 918.0], [536.0673400673401, 951.8922558922558], [552.0, 974.0], [561.333333333333], 4, 980.0], [581.0, 990.0], [596.6734006734006, 996.6734006734006], [627.6498316498316, 1003.4074074074074], [650.8821548821549, 1000.7138047138046], [681.0, 988.0], [697.0, 973.0], [708.0, 956.0], [722.0, 922.0], [715.0, 892.0], [720.0, 871.0], [643.0, 829.0], [619.0, 825.0]]
Shape type: polygon

Shape 2:
Label: tom_fruit_breaker_poly
Points: [[706.0, 738.0], [696.0, 738.0000000000009], [661.6666666666666, 750.6818181818181], [634.0, 774.6666666666666], [626.6666666666666, 786.0], [617.6666666666665, 811.7179487179487], [617.6666666666665, 824.5384615384615], [643.3333333333334, 828.2222222222222], [702.4444444444445, 871.333333333334], [715.7777777777778, 894.0], [721.5555555555555, 923.333333333334], [707.7777777777778, 95.9777777777778], [765.833333333333, 960.9090909090909], [799.92424242424, 949.5454545454545], [835.6153846153845, 921.9743589743589], [848.6666666666666, 889.555555555555555], [850.0, 858.4444444444445], [843.1060606060605, 831.3636363636364], [821.8939393939394, 80.5.6060606060606], [800.6818181818181, 788.9393939393939], [785.530303030303, 780.2272727272726], [774.0769230769231, 768.128205128205], [747.2463768115941, 749.4202898550724]]
Shape type: polygon

2. 이미지 라벨링 재처리

열매 폴더에서 익은 열매 분리
꽃 폴더에서 만개꽃 분리

3. 추가 이미지 라벨링

부족한 익은 토마토 라벨수 늘리기

06. YOLO 모델

생육 단계 판단 모델

생육 단계 판단 모델 - YOLO 스캔 중 corrupt 발생

```
import os
import re

log = r"""
val: WARNING D:\Python\project\AI\smart_tom\2.Validation\images\
V001_tom4_41_041_d1_07_20210116_10_07150225_66695123.png:
ignoring corrupt image/label: non-normalized or out of bounds coordinates
[   1.2503      1.0795      1.1435]
"""

# 이미지 파일 경로 추출하기
image_paths = re.findall(r'D:[^:]+\.png', log)

# 각 이미지 파일에 대해서
for image_path in image_paths:
    # 파일 경로에서 \를 /로 변환
    image_path = image_path.replace("\\", "/")

    # 비 출력 가능한 문자를 제거
    image_path = ''.join(ch for ch in image_path if ch.isprintable())

    # 이미지 파일이 실제로 존재하는 경우 삭제
    if os.path.exists(image_path):
        os.remove(image_path)
        print(f"Deleted image file: {image_path}")
    else:
        print(f"No such image file: {image_path}")

    # 레이블 파일 경로 생성
    label_path = image_path.replace("/images/", "/labels/").replace(".png",
".txt")
    # 레이블 파일이 실제로 존재하는 경우 삭제
    if os.path.exists(label_path):
        os.remove(label_path)
        print(f"Deleted label file: {label_path}")
    else:
        print(f"No such label file: {label_path}")
```



06. YOLO 모델

생육 단계 판단 모델

생육 단계 판단 모델 - 이미지 라벨링 재처리

꽃 분류 성능 문제 발생

```
Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.125 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 268 layers, 43609692 parameters, 0 gradients
    Class   Images Instances   Box(P)      R      mAP50      mAP50-95):
        all     11007    31009    0.666      0.685      0.665      0.544
        FlowerBud 11007    7989    0.839      0.475      0.63       0.401
        BloomingFlower 11007    1625    0.217      0.432      0.151      0.0925
        UnripeTomato 11007    21002    0.925      0.893      0.966      0.86
        RipeTomato   11007     393    0.682      0.941      0.914      0.822
Speed: 0.1ms preprocess, 9.1ms inference, 0.0ms loss, 0.9ms postprocess per image
```

06. YOLO 모델

생육 단계 판단 모델

생육 단계 판단 모델 - 꽃 폴더에서 만개꽃 분리

```
import cv2
import os
import matplotlib.pyplot as plt
import matplotlib.patches as patches

folder_path = ".\\data\\data\\val\\images"
labels_path = ".\\data\\data\\val\\labels"

def parse_yolo_coordinates(yolo_str):
    parts = yolo_str.split(" ")
    return float(parts[0]), float(parts[1]), float(parts[2]), float(parts[3]), float(parts[4])

def change_class_id(yolo_str, new_class_id):
    parts = yolo_str.split()
    parts[0] = str(new_class_id)
    return " ".join(parts)

image_files = [f for f in os.listdir(folder_path) if f.endswith(".png") and ("d1" in f or "d2" in f)]

resume_file = "V001_tomb_42_067_d1_04_20211022_07_86102933_93260846.png"

img_width, img_height = 1600, 1000
#img_width, img_height = 1000, 1000

if resume_file in image_files:
    idx = image_files.index(resume_file)
    image_files = image_files[idx:]

total_files = len(image_files)
farm_counts = [{"farm": i+1} for i in range(6)]
previous_farms_counts = [{"farm": i+1} for i in range(6)]

for i, img_name in enumerate(image_files):
    img_name, _ = os.path.splitext(img_name) # assuming 'tomb' always at this position

    img_path = os.path.join(folder_path, img_name)
    txt_path = os.path.join(labels_path, img_name.replace('.png', '.txt'))

    image = cv2.imread(img_path)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    with open(txt_path, 'r') as file:
        yolo_strs = file.readlines()

    skip_image = False
    for idx, yolo_str in enumerate(yolo_strs):
        if skip_image:
            break

        img, ax = plt.subplots(1, figsize=(15, 6))
        ax.imshow(image)

        class_id, x, y, w, h = parse_yolo_coordinates(yolo_str)
        x, y, w, h = x * img_width, y * img_height, w * img_width, h * img_height
        bbox = patches.Rectangle((x - w / 2, y - h / 2), w, h, linewidth=1, edgecolor="r", facecolor="none")
        ax.add_patch(bbox)

        ax.pause(0.01)
        plt.show()

        current_index = image_files.index(img_name)
        current_farm_count = current_index - previous_farms_counts[i]

        print(f"Processing image {i+1} of {total_files}, farm {farm_counts[current_farm_count]} of {farm_counts[farm_name]}")

        answer = input("Do you want to change class id of object {idx} in image {img_name}? (y/n/s): ").lower()
        if answer == "y":
            yolo_strs[idx] = change_class_id(yolo_strs[idx], "1") + "\n"
        elif answer == "s":
            skip_image = True
        else:
            print("Invalid input")

    plt.close()

if len(image_files) != len(farm_counts):
    previous_farms_counts += farm_counts[len(image_files):]

with open(txt_path, 'w') as file:
    file.writelines(yolo_strs)

print("Finished")
```



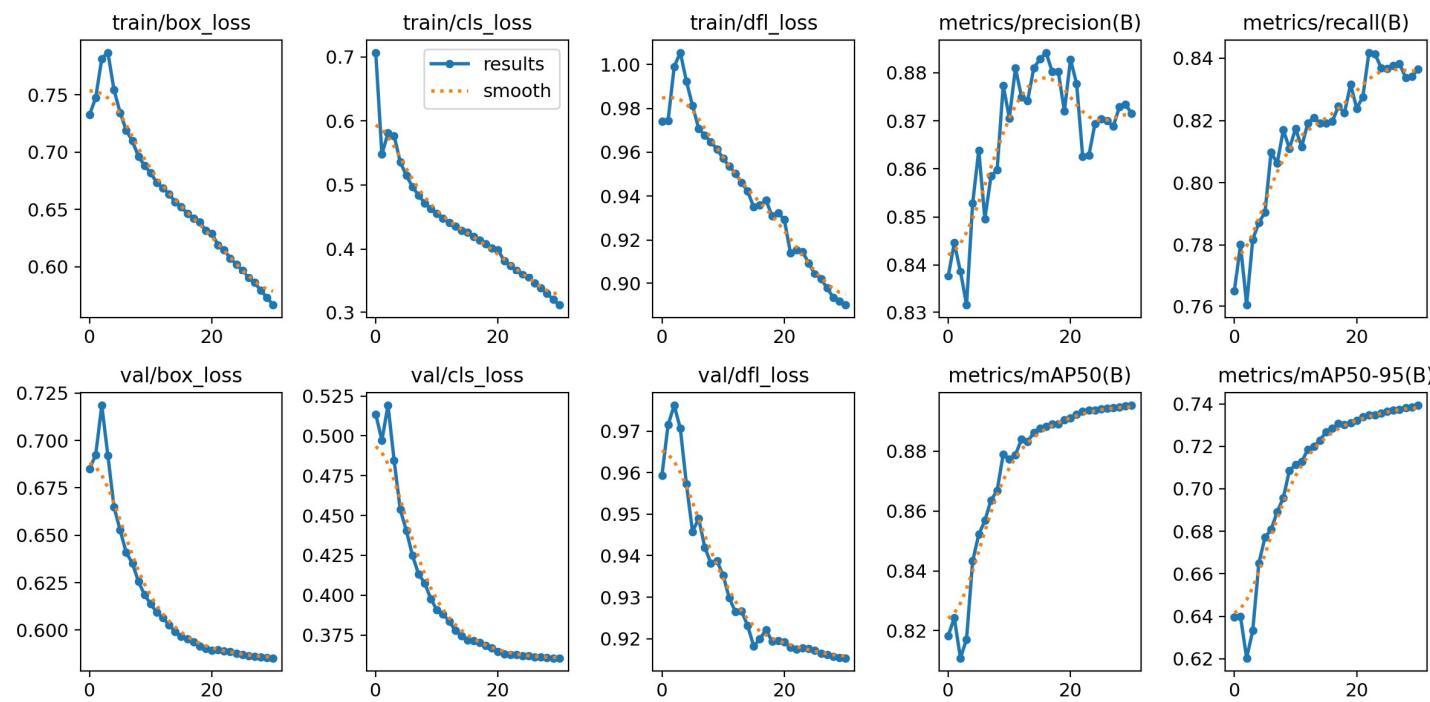
Processing image 1 of 90, farm tomb (1 of 90)
Do you want to change class id of object 0 in image V001_tomb_42_067_d1_04_20211022_07_86102933_93260846.png? (y/n/s): y

06. YOLO 모델

생육 단계 판단 모델

생육 단계 판단 모델 - 최종 모델 결과

yolov8l

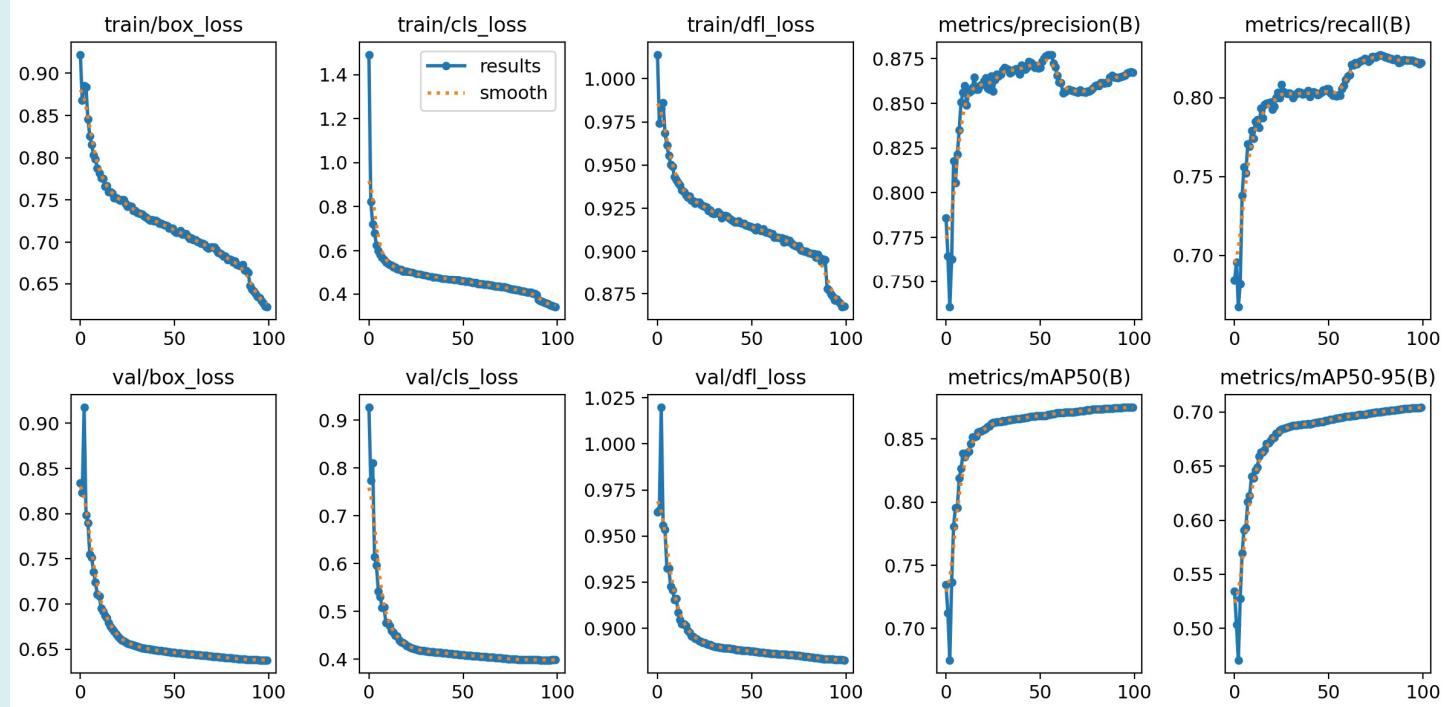


06. YOLO 모델

생육 단계 판단 모델

생육 단계 판단 모델 - 배포를 위한 경량화

yolov8l → yolov8n

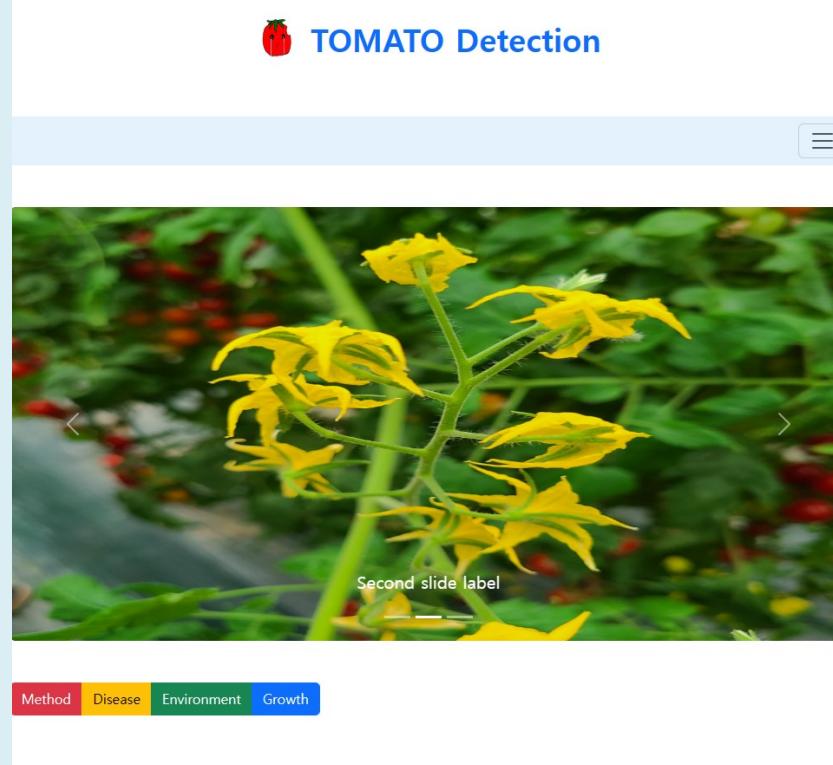


07.

WEB

웹사이트
메인화면

WEB



- 자바스크립트를 이용한 네비게이션 바
- carousel을 메인화면에 추가
- 바로가기 버튼 추가

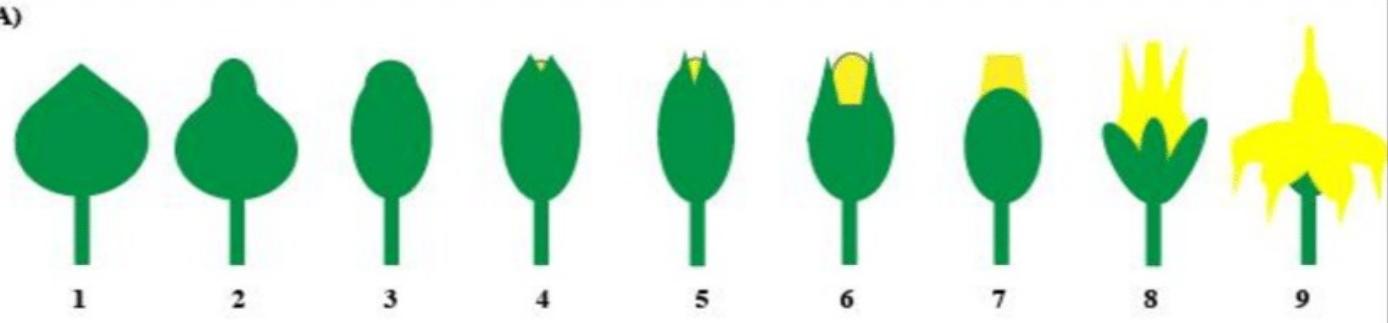
08. 자체 평가 의견

생육 모델
개선 사항

토마토 꽃과 열매의 생장 단계

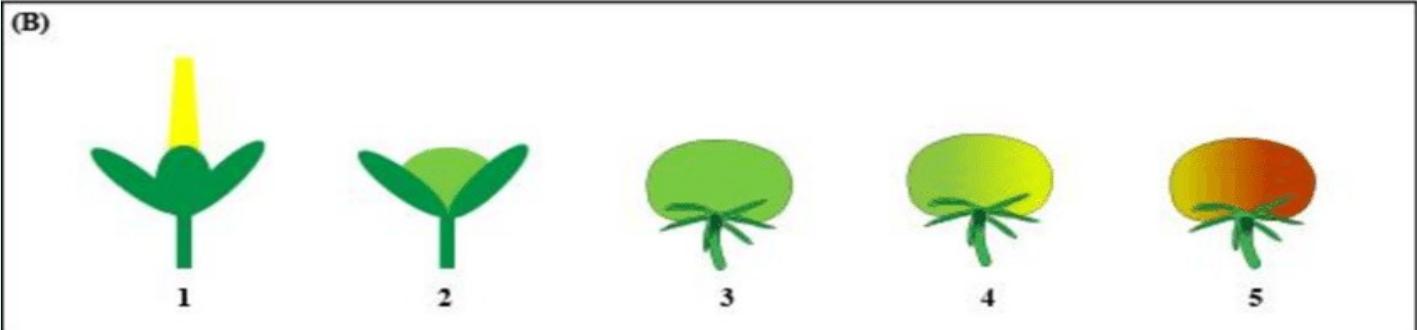
수분 전

(A)



수분 후

(B)



08. 자체 평가 의견

생육 모델
개선 사항

생육단계 판단 모델



Train(imshow)



원본 이미지 + 도형

08. 자체 평가 의견

생육 모델
개선 사항

생육단계 판단 모델



Train(imshow)



Predict

08. 자체 평가 의견

한계

디스크 사용량 제한

- 70 GB 가 넘는 대용량 파일 처리 어려움
- YOLO의 resume parameter 사용 불가능

활성화 가능 시간

- 최대 Epoch 수 제한

장비문제

컴퓨팅 단위 문제

- 일정량 연산 이후 사용 불가능

Azure, AWS 사용 시 비용 문제

08. 자체 평가 의견

아쉬운 점

데이터셋 문제

너무 많은 결측치

완성도

부실한 생육지표

체계적이지 못한 데이터셋

- 같은 데이터셋 내에서 농장별로 feature 등 데이터 구조 다름
- 모든 농장에 대한 품질 데이터 결여
- 제각기 다른 측정 시간 간격

파괴지표에 필수적인 id의 부재

08. 자체 평가 의견

아쉬운 점

데이터셋 문제



국회예산정책처
NATIONAL ASSEMBLY BUDGET OFFICE

나. 데이터 수집·활용 강화 필요

- 스마트농업을 구현하기 위해서는 생산과 환경, 유통, 소비 전반에 이르는 데이터 수집·활용이 필수적인 요소인데, 현재 데이터 수집·분석 비중이 높지 못하고 대부분 생산 관련 데이터가 중심이 되고 있어 향후 적극 개선노력 필요
 - 2020년도 기준으로 스마트농업 데이터 수집²²⁾을 실시하는 비중이 시설원예의 경우 65.2%, 축산의 경우 58.9% 노지작물·과수의 경우 22.4%에 불과²³⁾
 - 그리고, 수집된 데이터를 바탕으로 분석²⁴⁾을 하는 비중이 시설원예의 경우 57.9%, 축산의 경우 51.4%, 노지작물·과수의 경우 20.1%에 불과
 - 데이터 수집 및 분석 시에도 외부 등 전문기관 보다는 자가의 비중이 높게 나타났음

08.
Q&A

Thank you ~

