

개인 농업인을 위한 토마토 생육관리용
온디바이스 AI 모델 개발

Developing an On-device AI Model for Tomato
Cultivation Management for Small-Scale
Farmers

이용우
Department of Computer Science
상도로 369 숭실대학교
레지던스홀 929호
6683121@naver.com

Coauthor
Affiliation
Address
email

Coauthor
Affiliation
Address
email

Coauthor
Affiliation
Address
email

Coauthor
Affiliation
Address
email
(if needed)

개요(Abstract)

본 연구는 개인 농업인이 네트워크 연결에 제약 없이 현장에서 실시간으로 토마토의 생육 상태를 진단하고 맞춤형 조언을 얻을 수 있는 온디바이스 AI 시스템 개발을 목표로 한다. 기존의 농업 AI 시스템이 중앙 서버에 의존하여 통신 제약 및 비용 문제를 야기하는 한계를 극복하고자, 모든 연산이 모바일 기기 내에서 완결되는 하이브리드 AI 파이프라인을 제안한다. 제안하는 시스템은 세 단계로 구성된다. 첫째, 경량 객체 탐지 모델인 YOLOv11n를 사용하여 이미지 내 11개의 주요 토마토 관련 객체(잎, 과실, 꽃 등)를 인식한다. 둘째, 특정 객체(예: 생장 부위)가 탐지되면, 해당 영역의 이미지를 MobileNetV3-Small 기반의 경량 회귀 모델에 입력하여 식물의 키(cm)와 같은 정량적 생육 지표를 예측한다. 셋째, 이 시각적 분석 결과를 구조화된 텍스트로 변환하고, 사전에 정의된 토마토 재배 지식과 함께 Google의 Gemma-2-2B-IT 언어 모델에 입력하여 현재 상태에 대한 진단과 구체적인 관리 방안을 자연어로 생성한다. 실험 결과, YOLOv11n 모델은 주요 클래스에 대해 높은 탐지 성능을 보였으며(전체 mAP50-95 0.421, 주요 11개 클래스 평균 mAP50-95 약 0.601), 회귀 모델은 식물 키를 평균 약 6.66cm의 오차로 예측했다. 본 연구를 통해, 전문화된 비전 모델과 대규모 언어 모델을 결합한 온디바이스 시스템이 개인 농업인의 의사결정을 효과적으로 지원할 수 있는 실용적인 해결책이 될 수 있음을 확인하였다.

1 서론(Introduction)

최근 농업 분야는 농가 인구의 감소 및 고령화, 기후 변화로 인한 불확실성 증가 등 여러 도전에 직면해 있다. 이러한 문제에 대응하기 위해 데이터를 기반으로 생산성과 효율성을 높이는 스마트팜 기술이 주목받고 있으며, 인공지능(AI) 기술은 그 핵심적인 역할을 담당하고 있다. 그러나 현재 상용화된 많은 농업용 AI 서비스들은 복잡한 데이터 처리를 위해 중앙 서버와의 통신을 필요로 한다. 이는 인터넷 연결이 불안정한 농업 현장에서의 사용을 제약하고, 서버 구축 및 유지보수에 따른 비용 부담을 농가에 지우는 한계점을 가진다.

본 연구는 이러한 한계를 극복하기 위해, 외부 통신 없이 모바일 기기 자체에서 모든 연산이 이루어지는 **온디바이스(on-device) AI 시스템**을 제안한다. 본 연구는 특정 작물인 토마토를 대상으로 한 연구이다. 토마토는 2024년 농촌진흥청이 발표한 자료에 따르면 국내 5400ha면적에서 33만톤 규모로 생산되고 있으며, 전량이 시설에서 재배되고 있다는 것이 특징이다. 토마토는 다 과채류에 비교하여 단가는 약간 낮지만, 연중 가격이 안정되어 농가에서 선호하는 작물이다. 토마토는 통제된 환경에서 재배되기 때문에 학습 데이터를 얻기 수월하며, 변인이 잘 통제되어 학습 데이터와 실제 데이터가 유사하여 AI를 통한 예측 및 솔루션이 잘 적용될 확률이 높다. 개인 소규모 농장주나 귀농한 청년 농업인들이 전문가의 도움 없이도 현장에서 실시간으로 작물의 생육 상태를 쉽고 정확하게 진단하고, 이에 대한 맞춤형 조언을 얻을 수 있도록 돕는 것을 목표로 한다. 이를 위해 본 시스템은 세 가지 AI 모델을 결합한 하이브리드 파이프라인을 구축한다. 첫째, 경량 객체 탐지 모델인 YOLOv11n를 사용하여 이미지 내에서 잎, 과실, 꽃 등 토마토의 주요 상태를 나타내는 객체들을 신속하게 탐지한다. 둘째, MobileNetV3-Small 기반의 회귀 모델을 통해 탐지된 특정 객체의 이미지를 분석하여 식물의 키(cm)와 같은 정량적 생육 지표를 예측한다. 마지막으로, 이렇게 분석된 시각적, 정량적 정보를 구조화된 텍스트로 변환하고, 방대한 재배 지식을 담은 프롬프트와 함께 대규모 언어 모델(LLM)인 Gemma에 전달하여 현재 상태에 대한 종합적인 진단과 실용적인 관리 방안을 자연어로 생성한다. 이 연구를 통해, 전문화된 경량 비전 모델과 LLM의 추론 능력을 결합한 온디바이스 시스템의 구현 가능성과 실용성을 검증하고자 한다.

2 데이터셋 (Dataset)

본 연구는 AI Hub에 공개된 "**지능형 스마트팜 통합 데이터(토마토)**" 데이터셋을 기반으로 진행되었다. 이 데이터셋은 실제 스마트팜 환경에서 수집된 고품질 데이터로, 토마토의 생육 단계 및 환경에 대한 종합적인 정보를 포함하고 있다

데이터 출처: AI Hub

(<https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=534>)

2.1 데이터 구성

데이터셋은 크게 원천 데이터(이미지)와 라벨링 데이터(JSON)로 구성된다.

- **원천 데이터:** 토마토의 생육 단계, 부위, 병해충 등에 따라 카테고리별로 분류된 PNG 이미지 파일이다.
- **라벨링 데이터:** 각 이미지에 대한 메타데이터와 어노테이션 정보를 담고 있는 JSON 파일이다. 여기에는 이미지 촬영 정보(농장 ID, 주차, 날짜 등)와 생육 지표(growth_indicators), 그리고 이미지 내 객체에 대한 바운딩 박스(Bounding Box) 및 폴리곤(Polygon) 좌표와 클래스 라벨(shapes) 정보가 포함되어 있다

2.2 데이터 전처리

1. **클래스 선정:** 전체 16개의 라벨 클래스 중, tom_stem_dimeter_bb 와 같이 최종학습 결과에서 mAP가 현저히 낮아 유의미한 학습이 어렵다고 판단된 5개 클래스를 제외하고, 주요 11개 클래스를 최종 학습 대상으로 선정했다.

2. **포맷 변환:** 원본 라벨인 JSON 파일을 YOLO 학습에 적합한 .txt 포맷으로 변환했다. 이 과정에서 각 클래스에 정수 ID를 부여하고, 바운딩 박스 및 폴리곤 좌표를 이미지 크기에 맞춰 0과 1 사이의 값으로 정규화(Normalization)했다. 특히, 좌표값이 이미지 경계를 벗어나는 오류(out of bounds coordinates)를 방지하기 위해 경계값을 조정하는 로직을 추가했다.

3. **데이터셋 분할:** 전체 데이터셋은 학습(train), 검증(validation), 테스트(test) 세트로 분할하여 사용했다.

92 3 제안하는 시스템 (Proposed System)

93 본 연구에서는 토마토 이미지를 분석하여 생육 상태에 대한 진단 및 조언을 제공하는
94 3단계 파이프라인 시스템을 제안한다. 각 단계는 전문화된 AI 모델을 활용하여 정확성과
95 효율성을 높이고, 최종적으로 모바일 기기에서의 실행을 목표로 한다.

96

97 3.1 1단계: YOLOv11s를 이용한 객체 탐지

98 1. **모델:** 객체 탐지에는 Ultralytics의 YOLOv11n 모델을 사용했다. YOLOv11s 모델의
99 성능이 더 우월했으나, 엣지 디바이스에서의 사용을 전제로 하는 본 프로젝트에서 경량
100 모델을 보다 우선시했다. (주요 지표 성능차이 5% 미만)

101 2. **학습:** 사전 학습된(pre-trained) yolo11s.pt 가중치를 기반으로, 준비된 토마토 데이터셋에
102 대해 전이 학습(transfer learning)을 수행했다. 주요 학습 파라미터는 다음과 같다.

103 • **Epochs:** 100

104 • **Batch Size:** 32

105 • **Image Size:** 640x640

106 • **Optimizer:** AdamW (lr=0.001)

107 3. **역할:** 이 모델은 입력된 이미지에서 11개의 주요 클래스(잎, 과실, 꽃, 생장 부위
108 등)를 인식하고, 각 객체의 위치를 바운딩 박스로 제공하는 역할을 한다.

109

110 3.2 2단계: MobileNetV3-Small을 이용한 생육 지표 예측

111 1. **모델:** 특정 객체의 물리적 크기를 예측하기 위해, MobileNetV3-Small을 특징
112 추출기(feature extractor)로 사용하는 회귀 모델(GrowthRegressor)을 설계했다. 이는 모바일
113 환경에서의 효율성을 고려한 선택이다.

114 2. **학습:**

115 • JSON 파일에 기록된 실제 측정값(예: plantHeight)을 정답(ground truth)으로
116 사용했다.

117 • JSON의 바운딩 박스 정보(tom_growth_bb 등)를 이용해 원본 이미지에서 해당
118 객체 영역만 잘라낸 이미지 패치를 생성했다.

119 • 이 (이미지 패치, 실제 측정값) 쌍을 사용하여 모델을 학습시켰으며, 주요
120 파라미터는 다음과 같다.

121 ○ **Epochs:** 50

122 ○ **Batch Size:** 32

123 ○ **Optimizer:** Adam (lr=0.0005)

124 ○ **Loss Function:** MSELoss

125 3. **역할:** YOLO가 탐지한 특정 객체(예: tom_growth_bb)의 이미지 패치를 입력받아, 해당
126 객체의 실제 크기(cm)를 예측한다.

127

128 3.3 3단계: Gemma를 이용한 종합 진단 및 조언 생성

129 1. **모델:** 최종적으로 사용자에게 제공될 조언을 생성하기 위해 Google의 경량 언어
130 모델인 google/gemma-2-2b-it를 사용했다.

131 2. **프롬프트 엔지니어링:**

132 • **시스템 프롬프트:** 모델이 "토마토 재배 전문가 AI 어시스턴트" 역할을
133 수행하도록 지시하고, 라벨의 의미, 최적 환경 조건, 월별 관리 팁 등 방대한
134 양의 전문 지식을 미리 제공했다.

135 • 사용자 프롬프트: 1, 2단계에서 분석된 결과(탐지된 객체 목록, 예측된 식물 높이
136 등)와 이미지 촬영 날짜를 구조화된 텍스트로 만들어 함께 입력했다.

137 3. 역할: 이미지의 시각적 분석 결과와 사전 제공된 전문 지식을 종합적으로 고려하여,
138 현재 토마토의 상태를 진단하고 사용자를 위한 맞춤형 관리 방안을 자연어로 생성한다.

139

140 4 실험 결과 및 분석 (Experiments & Results)

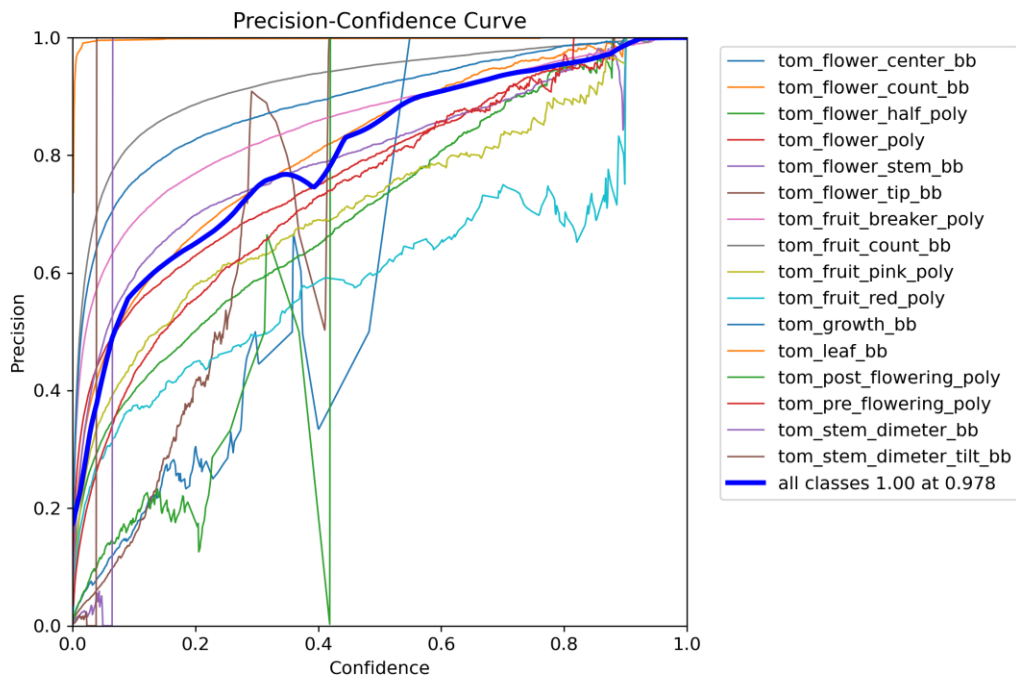
141 모델의 학습 과정과 성능을 더 다각적으로 분석하기 위해 주요 성능 지표 변화 및 혼동
142 행렬을 시각화 하였다.

143

144 4.1 YOLO 객체 탐지 모델 성능

145

146 1) YOLO v11-n Precision 그래프



147

148

149

150

151

152

153

154

155

156

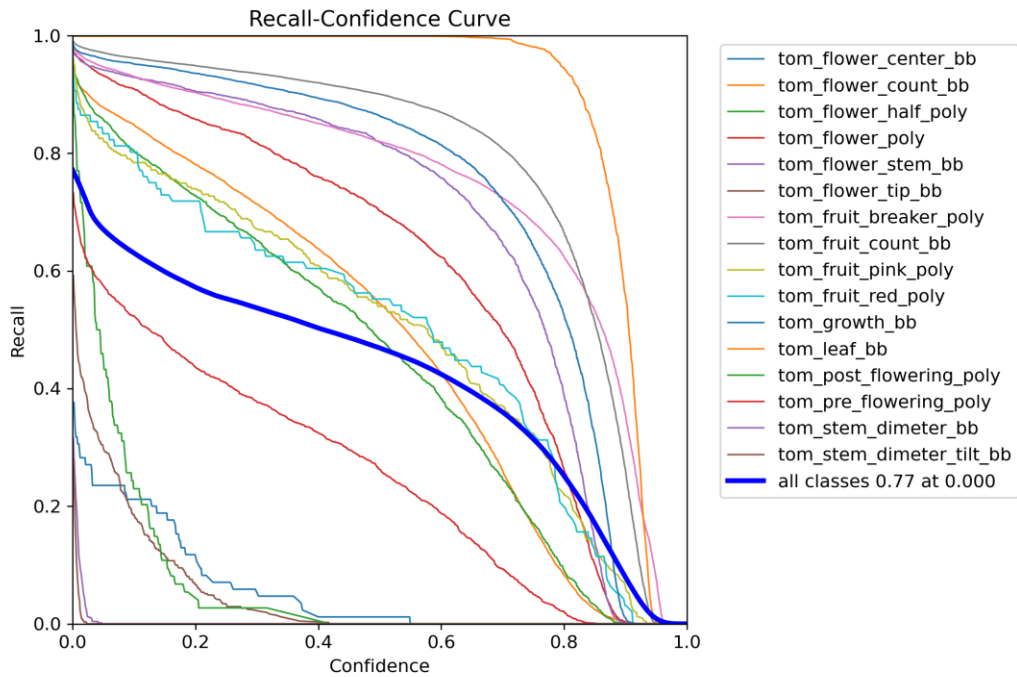
157

158

159

160

2) YOLO v11-n Recall 그래프



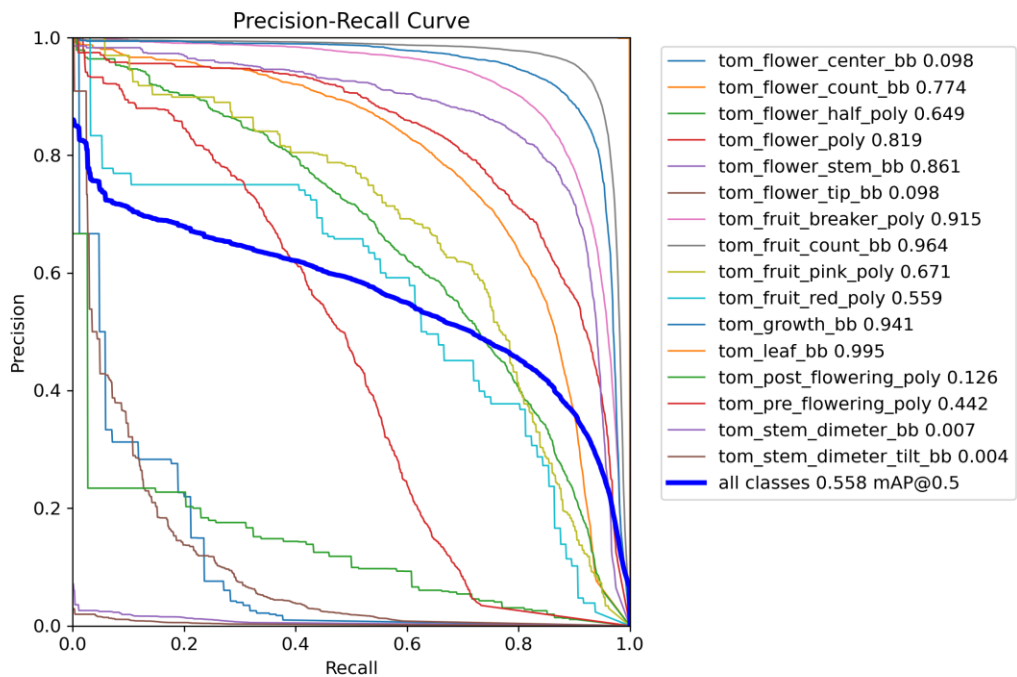
161

162 이 두 그래프는 에폭(epoch)이 진행됨에 따라 정밀도(Precision),
 163 재현율(Recall)이 어떻게 변화하는 지 보여준다.

164 두 그래프 모두 학습이 진행되면서 꾸준히 **confidence**가 상승하는
 165 상향 형태를 보이는 것을 알 수 있다. 이는 모델이 학습을 통해
 166 점점 더 정확해지고, 놓치는 객체도 줄어들고 있음을 의미한다.

167

3) YOLO v11-n PR 그래프



168

169

170 이 그래프는 모델의 신뢰도(confidence) 임계값을 변경함에 따라
171 정밀도와 재현율이 어떻게 변하는지를 보여준다.

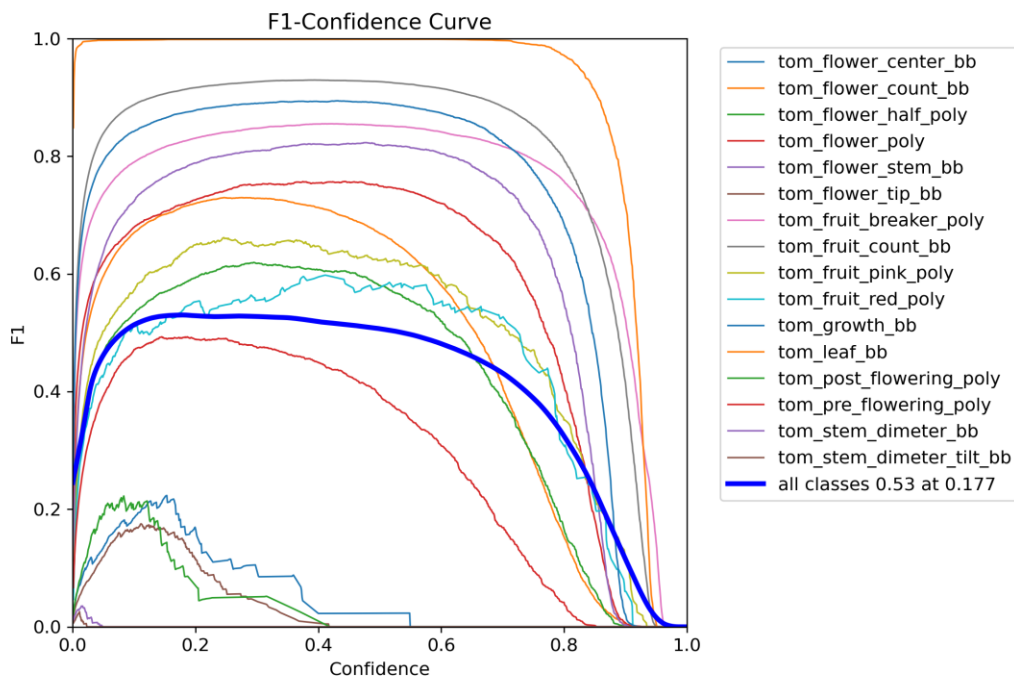
172 이상적인 모델은 그래프가 오른쪽 상단(정밀도=1, 재현율=1)에
173 가깝게 그려진다. 현재 그래프를 보면, 대부분의 클래스(특히
174 tom_leaf_bb, tom_fruit_count_bb 등)가 오른쪽 상단에 근접한 좋은
175 형태를 보이고 있다. 이는 정밀도와 재현율의 균형이 잘 잡혔다는
176 의미로 해석할 수 있다.

177

178

179 4) YOLO v11-n F-Confidence 그래프

180



181

182

183 F1 점수(F1-score)는 정밀도(Precision)와 재현율(Recall)의 조화
184 평균으로, 모델의 전반적인 정확도를 나타내는 중요한 지표다.
185 [그림 1]에서 볼 수 있듯이, 학습 예폭이 진행됨에 따라 대부분의
186 주요 클래스에 대한 F1 점수가 꾸준히 상승하여 최종적으로
187 0.8~0.9 이상의 높은 값에 수렴하는 것을 확인하였다. 이는 모델이
188 과적합(overfitting) 없이 안정적으로 학습되었음을 보여준다.

189

190

191

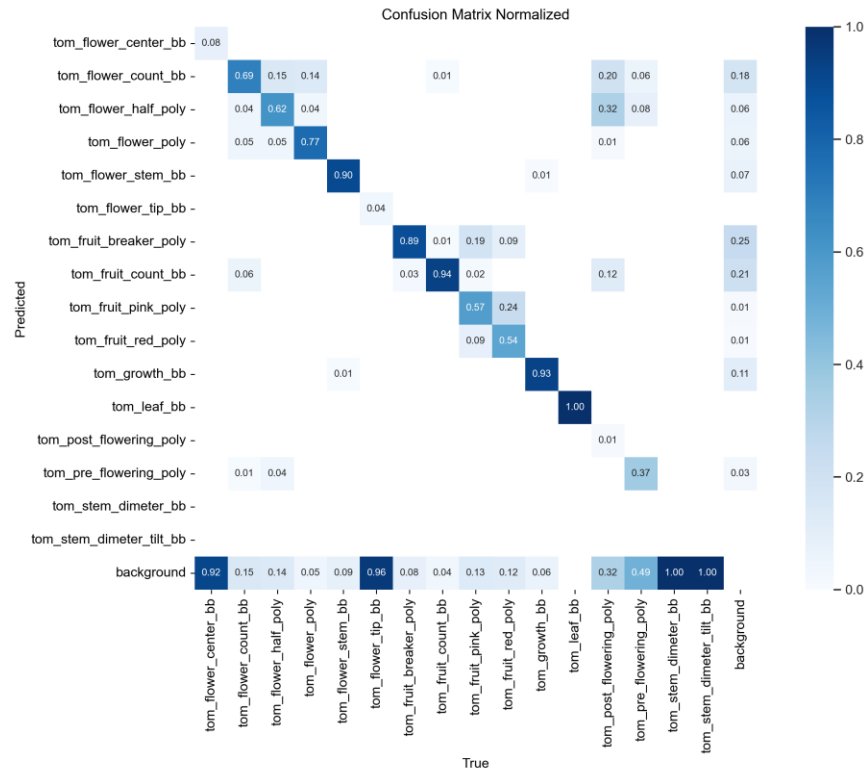
192

193

194

195

5) Confusion matrix normalized



클래스별 혼동 행렬 분석 : 상단의 그림은 정규화된 혼동 행렬(Normalized Confusion Matrix)로, 모델이 각 클래스를 얼마나 정확하게 분류하고 다른 클래스와 혼동하는지를 보여준다.

대각선상의 값이 1에 가까울수록 해당 클래스를 정확하게 예측했다는 의미이다. tom_leaf_bb (ID 11), tom_fruit_count_bb (ID 7) 등 시각적 특징이 명확한 클래스들은 0.9 이상의 높은 값으로 거의 혼동이 없었음을 알 수 있다. 반면, tom_post_flowering_poly (ID 12)와 같이 데이터 수가 적고 특징이 모호한 클래스는 배경(background)으로 잘못 예측하는 비율이 상대적으로 높게 나타났다. 이러한 분석은 mAP 수치만으로는 알 수 없는 클래스 간의 관계 및 모델의 구체적인 강점과 약점을 파악하는 데 도움을 준다.

6) YOLO v11-n Class별 지표

Class	Images	Instances	Box(P)	Box(R)	mAP50	mAP50-95
all	29274	77812	0.645	0.577	0.558	0.421
tom flower center bb (만개 꽃잎 중앙)	85	85	0.253	0.118	0.0977	0.0271
tom_flower_count_bb (개화군)	3308	8916	0.659	0.787	0.774	0.475
tom_flower_half_poly (개화)	1362	2128	0.493	0.733	0.649	0.424

tom_flower_poly (만개)	1514	2314	0.622	0.861	0.819	0.546
tom_flower_stem_bb (화방 높이)	2219	2835	0.692	0.908	0.861	0.668
tom_flower_tip_bb (만개 꽃잎 끝)	86	410	0.375	0.0756	0.0983	0.0293
tom_fruit_breaker_poly (과실색-초록)	5463	14391	0.764	0.905	0.915	0.796
tom_fruit_count_bb (착과군)	12886	33649	0.883	0.95	0.964	0.77
tom_fruit_pink_poly (과실색-핑크)	255	315	0.571	0.743	0.671	0.601
tom_fruit_red_poly (완숙)	81	96	0.442	0.719	0.559	0.472
tom_growth_bb (생장길이)	4074	8365	0.822	0.937	0.941	0.724
tom_leaf_bb (옆장옆폭)	1059	1059	0.998	0.999	0.995	0.879
tom_post_flowering_poly (과실 없이 시든 꽃)	65	74	0.2	0.0508	0.126	0.0677
tom_pre_flowering_poly (미개화)	853	1784	0.548	0.444	0.442	0.26
tom_stem_dimeter_bb (줄기 두께)	753	753	1	0	0.00734	0.00229
tom_stem_dimeter_tilt_bb (기울어진 줄기 두께)	638	638	1	0	0.00379	0.000983
형광색 부분을 제외한 통계	mAP50	mAP50-95				
평균	0.7809	0.601				

214 평가 지표:

215 **P(Precision, 정밀도) Box(P):** 모델이 탐지했다고 감지한 박스 중 실제로 정답인 것의 비율

216 **R(Recall, 재현율) Box(R):** 실제 이미지에 존재하는 모든 정답 객체 중에서, 모델이 얼마나
217 검출해냈는 지의 비율.

218 **mAP50 (mean Average Precision at IoU=0.5):** 모델의 전반적인 정확도를 나타내는 지표.
219 IoU(Intersection over Union)라는 기준을 0.5로 설정했을 때의 평균 AP(Average Precision).

220 **mAP50-95 (mean Average Precision at IoU=0.5:0.95):** 모델의 전반적인 정확도를 나타내는
221 지표. IoU(Intersection over Union)라는 기준을 0.5부터 0.95까지 0.05씩 높여가며 각
222 기준으로 mAP를 계산한 후, 그 평균을 낸 값.

223 **IoU(Intersection over Union):** 모델이 예측한 바운딩 박스와 실제 바운딩 박스가 n%
224 겹쳤을 때 올바르게 탐지했다고 판단하는 기준. IoU 0.5 = 50% 겹침, IoU 0.95 = 95% 겹침.

225 mAP가 낮게 측정된 5개의 class(tom_flower_center_bb,
226 tom_flower_tip_bb, tom_post_flowering_poly, tom_stem_diameter_bb, tom_stem_diameter_tilt_bb)를
227 drop하였으며,

228 유효하다고 판정한 11개의 class의 평균은 mAP50 : 0.7809, mAP50-95 : 0.601로 YOLO v11
229 nano 모델의 COCO 데이터셋의 학습값인 mAP50 : 0.542, mAP50-95 : 0.409보다 우수하므로
230 실제 task에 유의미하게 활용할 수 있는 수치이다.

231 (COCO 데이터셋 map값 출처 : [YOLO11 Small vs YOLO11 Nano](#))

232

233 **4.2 생육 지표 예측 모델 성능**

234 plantHeight를 예측하는 GrowthRegressor 모델을 50 에폭 동안 학습한 결과, 가장 우수했던
235 검증 손실(Best val Loss)은 **44.32**를 기록했다.

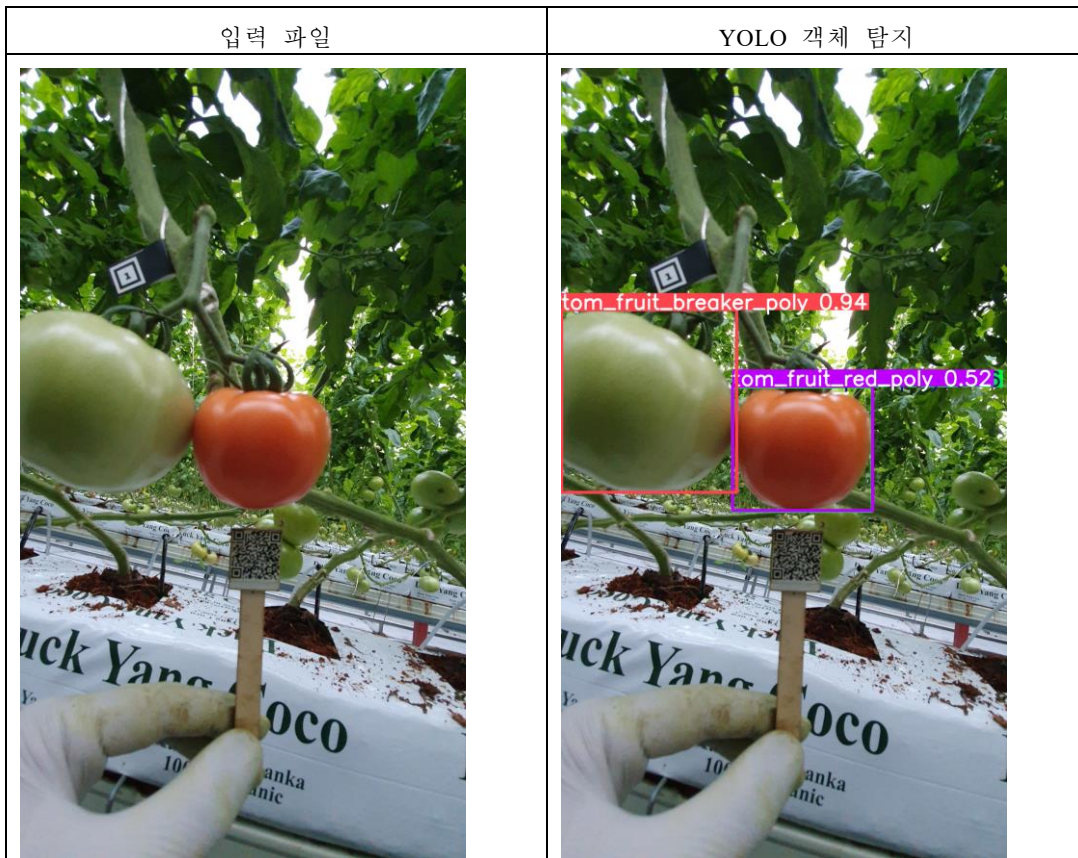
236 이는 평균 제곱근 오차(RMSE) 기준으로 약 **6.66 cm**의 평균 예측 오차를 의미한다.
237 토마토는 2m 이상 자랄 수 있기 때문에 이는 전반적인 성장 추세를 파악하는 데
238 유의미한 수준의 정확도이다.

239

240 **4. Gemma 조연 생성 예시**

241 입력 (분석 결과): 촬영 시점: 2021.11.10, 탐지된 객체: tom_fruit_breaker_poly,
242 tom_fruit_red_poly, tom_fruit_pink_poly

243



244

245

246

```

--- AI 기반 관리 조언 ---
안녕하세요! 토마토 재배 전문 AI 어시스턴트입니다. 😊 사진에서 분석 결과, 토마토는 착색 시작 단계인 `tom_fruit_breaker_poly`에서
`tom_fruit_pink_poly`로 넘어가는 핑크색으로 변화하고 있습니다.

**현재 상태:**

* **생장 단계:** `tom_fruit_breaker_poly`에서 `tom_fruit_pink_poly`로 넘어가는 핑크색으로 변화하고 있습니다.
* **결실 및 수확기:** 열매가 열리고 익기 시작하며, 성장, 개화, 결실, 수확이 동시에 일어나는 복합적인 관리기입니다.
* **꽃봉오리:** 꽃봉오리(아직 피지 않음)

**추천 행동 계획:**

1. **수확 준비:** 현재 익어가는 핑크색 열매는 곧 완전히 익어 붉은색으로 변화할 것입니다.
2. **결실 제거:** 과일 품질과 식물 전체의 성장을 위해 한 화방 당 4~5개의 과일만 남기고 나머지는 꽃일 때 미리 제거해주는 것이 좋습니다.
   결실 제거는 오전에 손으로 하는 것이 좋습니다.
3. **온도 및 햇빛:** 토마토는 낮 25~27℃, 밤 18~20℃의 온도를 선호합니다. 햇빛이 부족하면 꽃이 떨어질 수 있습니다.
4. **수확:** 토마토는 익은 상태에서 수확하는 것이 좋습니다.

**주의 사항:**

* **야간 온도:** 야간 온도가 13~24℃ 범위일 때 수확이 가장 잘 됩니다.
* **병충해:** 잿빛곰팡이병, 잎곰팡이병, 역병 (주로 저온다습 환경 유익), 온실가루이, 파밤나방, 진딧물.

**추가 조언:**

* 토마토의 성장과 품질을 위해 햇빛이 충분히 들어오도록 돕는 것이 중요합니다.
* 결실 제거는 오전에 손으로 하는 것이 좋습니다.
* 토마토의 품질을 위해 익은 상태에서 수확하는 것이 좋습니다.

**마무리:**

토마토 재배는 즐거운 일이지만, 꾸준한 관찰과 관리가 필요합니다.

**궁금한 점이 있으면 언제든지 질문해주세요!** 😊
<end_of_turn>

--- 결과가 'Final_Versio_final/data/result_for_test\Test_img_2_result.txt' 파일에 저장되었습니다. ---

```

248 --- 결과가 'Final_Versio_final/data/result_for_test\Test_img_2_result.txt' 파일에 저장되었습니다. ---

249 Gemma-2-2B-it 모델은 프롬프트에 사전 입력된 지식과 이미지에서 검출된 지식을 통해
250 농업 지식을 제공한다.

251

252 5 결론 및 고찰 (Conclusion & Discussion)

253 본 연구는 개인 농업인이 네트워크 제약 없이 현장에서 토마토의 생육 상태를 진단하고
254 맞춤형 조언을 얻을 수 있는 온디바이스 AI 시스템 개발을 목표로 하였다. 이를 위해
255 YOLO, MobileNet, 그리고 Gemma LLM을 결합한 3단계 하이브리드 파이프라인을
256 성공적으로 구축하며 가시적인 성과를 달성하였다.

257

258 5.1 연구 결론

259 본 연구의 핵심 성과는 하나의 거대한 멀티모달 모델에 모든 분석을 의존하는 대신, 각
260 작업에 특화된 경량 엣지 모델들을 분산적으로 활용하는 접근 방식의 유효성을 입증한
261 것이다.

262 첫째, 객체 탐지 단계에서는 모바일 환경을 고려하여 YOLOv11n 모델을 최종 선택하였다.
263 더 큰 YOLOv11s 모델이 전체 mAP50-95 기준 0.46으로, YOLOv11n 모델(0.421)보다 소폭
264 우수한 성능을 보였으나, 엣지 디바이스에서의 경량성과 속도를 우선 고려하여
265 YOLOv11n을 채택하였다. 이 모델은 tom_leaf_bb(mAP50-95: 0.879),
266 tom_fruit_breaker_poly(0.796) 등 주요 11개 클래스에 대해 충분히 유의미한 탐지 성능을
267 보여주었다.

268 둘째, 식물 키 예측을 위한 회귀 모델 선정 과정에서는 MobileNetV3-Small,
269 MobileNetV4-small, MobileNetV4-hybrid-medium 세 가지 모델에 대한 비교 실험을
270 수행하였다. 놀랍게도 가장 최신 아키텍처인 MobileNetV4 계열 모델들(최저 val Loss:
271 45.67)보다 **MobileNetV3-Small 모델이 가장 낮은 검증 손실(Best val Loss: 44.32)**을
272 기록하며 가장 우수한 성능을 보였다. 이는 최신 아키텍처가 모든 특정 작업에서 항상
273 최고의 성능을 보장하는 것은 아니며, 데이터셋의 특성과 문제 정의에 따라 더 단순한
274 모델이 더 효과적일 수 있다는 중요한 시사점을 제공한다.

셋째, 최종 조언 생성 단계에서는 Gemma 2B-IT(google/gemma-2-2b-it) 모델을 활용하였다. 초기에는 Gemma 3n과 같은 최신 멀티모달 모델을 고려했으나, 자료 부족 및 커널 충돌과 같은 환경 설정의 어려움으로 인해 안정적으로 사용 가능하며, 양자화를 통한 경량화가 용이하고, 정교한 프롬프트 엔지니어링이 가능한 Gemma 2B-IT를 최종 선택하였다. 시스템 프롬프트를 통해 토마토 재배에 대한 방대한 전문 지식을 미리 주입하고, YOLO와 회귀 모델의 분석 결과를 동적으로 결합하여 사용자에게 전달함으로써, 상황에 맞는 구체적이고 실용적인 조언을 생성할 수 있었다.

이러한 결과들을 통해, 본 연구에서 제안한 특화된 경량 비전 모델들과 LLM을 결합한 하이브리드 파이프라인이 범용 온디바이스 멀티모달 모델보다 특정 도메인(토마토 생육)에 대해 더 정밀하고 신뢰도 높은 분석을 제공할 수 있는 유의미한 접근 방식임을 확인하였다.

286

287 5.2 고찰 및 향후 과제 (Discussion & Future Work)

288 본 프로젝트를 진행하며 몇 가지 한계점과 향후 개선 방향을 도출할 수 있었다.

289 첫째, LLM 선택 및 프롬프트 엔지니어링의 한계가 있었다. 시간 제약으로 인해 Gemma 2B-IT 외에 더 최적화된 소형 언어 모델(sLLM)을 충분히 탐색하지 못했다. 또한, 현재는 YOLO가 탐지한 객체 목록과 예측된 키 정보를 일괄적으로 텍스트화하여 LLM에 전달하고 있으나, 탐지된 객체의 종류(예: 꽃, 열매, 잎)에 따라 동적으로 다른 종류의 지식을 필터링하여 프롬프트를 구성하는 고도화된 엔지니어링 기법을 적용한다면 더 정확한 맞춤형 조언 생성이 가능할 것이다.

295 둘째, 생육 지표 예측 모델의 확장성이다. 현재는 식물의 키(plantHeight) 예측 모델만 구현하였으나, 데이터셋에는엽장/엽폭, 화방 크기, 줄기 두께 등 다양한 생육 지표가 포함되어 있다. 이러한 지표들을 예측하는 추가적인 회귀 모델들을 학습하고 파이프라인에 통합한다면, 훨씬 더 다각적이고 정밀한 생육 상태 진단이 가능해질 것이다.

300 마지막으로, 실제 온디바이스 환경에서의 최적화 및 검증이 필요하다. 현재 개발된 모델들을 TensorFlow Lite로 변환하고 양자화를 적용하여, 실제 모바일 기기에서의 추론 속도, 메모리 사용량, 배터리 소모량 등을 측정하고 검증하는 과정이 후속 연구로 이루어져야 한다. 이를 통해 진정한 의미의 "온디바이스 AI 시스템"으로서의 실용성을 확보할 수 있을 것이다.

305

306 6 도전 목표 평가 (Challenge Goals Evaluation)

307 본 연구는 개인 농업인이 네트워크 제약 없이 현장에서 토마토의 생육 상태를 진단하고 맞춤형 조언을 얻을 수 있는 온디바이스 AI 시스템 개발을 목표로 하였다.

309

310 6.1 목표 평가

311 본 연구는 초기 제안서에서 설정한 "실생활 문제 해결을 위한 온디바이스 AI 시스템 개발"이라는 도전 목표를 성공적으로 달성하였다고 평가한다. 초기 계획대로 YOLO, 경량 회귀 모델, LLM을 결합한 하이브리드 파이프라인을 구축하여 네트워크 없이 작동하는 시스템의 프로토타입을 완성했다.

315 다만, 구현 과정에서 초기 계획이 일부 수정 및 구체화되었다. 초기에는 MobileNetV4나 최신 Gemma 3n 모델 사용을 고려했으나, 실제 실험 결과 MobileNetV3-Small이 더 안정적이고 우수한 성능을 보였고, Gemma 3n 모델을 윈도우 환경에서 사용하는 것이 어렵다는 문제가 발견되어 Gemma-2-2B-IT가 환경 안정성면에서 더 적합하다고 판단하여 최종 모델을 변경하였다. 이는 이론적인 최신 기술을 맹목적으로 따르기 보다, 실제 데이터와 문제에 가장 적합한 모델을 실험적으로 검증하는 과정이 더 중요함을 보여준다

321

322

323

324 7 작업 계획 평가 (Work Plan Evaluation)

325 본 연구가 part1에서의 계획을 잘 수행했는지 평가한다.

326

327 7.1 계획 평가

328 초기 제안서에서 각 단계별 예상 소요 시간을 계획했으나, 실제 프로젝트 진행
329 과정에서는 많은 차이가 있었다. "데이터셋 확보, 분석 및 전처리" 단계는
330 예상(10시간)보다 훨씬 더 많은 시간이 소요되었다. 특히, 라벨 데이터의 좌표 오류(out of
331 bounds coordinates)나 중복 문제를 해결하고, 각 모델에 맞는 데이터셋 구조를 구축하는 데
332 예상치 못한 디버깅 시간이 필요했다.

333 반면, "소형 LLM 연구 및 통합" 단계는 Hugging Face transformers 라이브러리의 높은
334 편의성 덕분에 예상(15시간)보다 빠르게 프로토타입을 완성할 수 있었다. 전반적으로,
335 실제 모델 학습 시간보다는 데이터의 품질을 확보하고 전처리하는 과정에 더 많은
336 노력이 필요하다는 것을 알 수 있었다.

337

338 8 테스트 (Testing)

339 본 프로젝트의 결과 신뢰성을 확보하기 위해 다음과 같은 테스트를 수행했다.

340

341 8.1 단위 기능 테스트

342 각 Python 스크립트의 주요 함수(예: JSON-to-YOLO 변환, 데이터 로딩, 모델 추론 등)에
343 대해, 소수의 샘플 데이터 파일(예: 5~10개의 이미지 및 JSON)을 입력으로 사용하여
344 의도한 대로 출력물이 생성되는지 확인했다.

345

346 8.2 파이프라인 통합 테스트

347 최종적으로 test_pipeline.py 스크립트를 작성하여 전체 파이프라인(YOLO 탐지 → 회귀
348 모델 예측 → LLM 조인 생성)이 끊김 없이 작동하는지 검증했다. 이 과정에서 각 모델
349 간의 데이터 전달 형식이 올바른 지 확인했다.

350

351 8.3 시각적 검증

352 테스트 과정에서 YOLO가 탐지한 바운딩 박스를 실제 이미지에 그려서 저장하는 기능을
353 추가하여, 모델이 엉뚱한 곳을 탐지하지는 않는지 시각적으로 직접 확인하며 디버깅을
354 진행했다. 이를 통해 mAP 수치만으로는 알 수 없는 질적인 성능을 검증할 수 있었다.

355

356 9 협업 (Collaboration)

357 본 프로젝트는 기본적으로 단독으로 진행되었으나, 개발 과정에서 다음과 같은 외부
358 리소스의 도움을 받았다.

359

360 9.1 Hugging Face

361 Gemma 2B-IT 모델 및 transformers 라이브러리를 사용했으며, 모델 카드와 공식 문서를
362 참고하여 모델을 로드하고 파이프라인을 구축했다.

363

364 9.2 Ultralytics YOLO

365 YOLOv11 모델의 학습 및 추론을 위해 Ultralytics의 공식 문서와 GitHub 저장소를
366 참고하여 하이퍼파라미터 설정 및 API 사용법을 익혔다.

367

368 9.3 PyTorch 및 timm 라이브러리

369 MobileNetV3/V4 기반의 회귀 모델을 구현하고 학습시키기 위해 PyTorch 공식 문서와
370 timm 라이브러리의 문서를 참조했다

371

372 9.4 YOLO v11 기반 농산물 객체 검출 오토레이블링 딥러닝 모델 373 개발 (<https://doi.org/10.33778/kcsa.2024.24.5.097>)

374 유사한 주제의 선행연구로, 해당 논문은 사과(열매)에 대하여 생리장해, 흠집, 기형,
375 병충해 4가지의 class를 YOLO v11n, YOLO v11s로 mAP50, mAP50-95를 측정하여 농산물
376 객체를 검출, 분류하였다. 해당 논문의 P-curve, R-curve, mAP 지수를 참고하여 본 연구의
377 유효성을 검증하는데 사용하였다.

378

379 10 참고 문헌 (References)

380 [1] kais99.org/jkais/journal/Vol16No12/p65/9s.html

381 [2]<[37302020B1E8BBEFB1D92DBDAB8B6C6AE20B3F3BEF7C0BB20C0A7C7D120594F4C4F20](https://doi.org/10.33778/kcsa.2024.24.5.097)
382 [B1E2B9DD20C0DBB9B020C7D8C3E620C5BDC1F620B8F0B9D9C0CF2E687770](https://doi.org/10.33778/kcsa.2024.24.5.097)>

383 [3][https://www.nongsaro.go.kr/portal/ps/psz/psza/contentSub.ps?menuId=PS03172&sSeCode=335001](https://www.nongsaro.go.kr/portal/ps/psz/psza/contentSub.ps?menuId=PS03172&sSeCode=335001&cntntsNo=101611&totalSearchYn=Y)
384 [&cntntsNo=101611&totalSearchYn=Y](https://www.nongsaro.go.kr/portal/ps/psz/psza/contentSub.ps?menuId=PS03172&sSeCode=335001&cntntsNo=101611&totalSearchYn=Y)

385 [4] [google/gemma-2-2b-it · Hugging Face](https://huggingface.co/google/gemma-2-2b-it)

386 [5] <https://roboflow.com/compare-model-sizes/yolo11s-vs-yolo11n>

387 [6] <https://aihub.or.kr/aihubdata/data/view.do?currMenu=115&topMenu=100&dataSetSn=534>

388 [7] [mobilenet_v3_small — Torchvision main documentation](https://pytorch.org/docs/stable/mobilenet_v3.html)

389 [8] <https://docs.ultralytics.com/ko/models/yolo11/>

390 [9] <https://koreascience.kr/article/JAKO202412757605231.pdf>

391 [10][https://kosis.kr/statHtml/statHtml.do?sso=ok&returnurl=https%3A%2F%2Fkosis.kr%3A443](https://kosis.kr/statHtml/statHtml.do?sso=ok&returnurl=https%3A%2F%2Fkosis.kr%3A443%2FstatHtml%2FstatHtml.do%3FtblId%3DDT_1ET0027%26orgId%3D101%26)
392 [%2FstatHtml%2FstatHtml.do%3FtblId%3DDT_1ET0027%26orgId%3D101%26](https://kosis.kr/statHtml/statHtml.do?sso=ok&returnurl=https%3A%2F%2Fkosis.kr%3A443%2FstatHtml%2FstatHtml.do%3FtblId%3DDT_1ET0027%26orgId%3D101%26)