

인공지능 기반 콩 성장분석 방법 연구⁺

(A Study on the Artificial Intelligence-Based Soybean Growth Analysis Method)

전 문 석¹⁾, 김 영 태²⁾, 정 유 석³⁾, 배 효 준³⁾, 이 채 원⁴⁾, 김 송 림^{5)*}, 최 인 찬^{6)*}
(Moon-Seok Jeon, Yeongtae Kim, Yuseok Jeong, Hyojun Bae, Chaewon Lee, Song Lim Kim, and Inchan Choi)

요 약 콩은 세계 5대 식량작물 중 하나로 식물성 단백질의 주요 공급원이다. 작물 특성상 기후변화에 따라 곡물 생산량에 큰 영향을 받기 때문에 국립농업과학원에서는 콩 품종별 성장 분석을 통해 작물표현형 연구를 진행하고 있다. 콩 품종별 성장 분석을 위한 성장 과정 사진 촬영은 자동화된 시스템으로 이루어지지만 성장 상태를 확인, 기록, 분석하는 과정은 수작업으로 진행되고 있다. 본 논문에서는 이러한 과정을 자동화 할 수 있도록 콩 작물의 영상 데이터에서 콩잎 객체를 검출하는 YOLOv5s 모델과 검출된 콩잎의 전개 여부를 판단하는 합성곱 신경망(Convolution Neural Network; CNN) 모델을 설계, 학습하였다. 두 모델을 결합하고 검출된 콩잎의 좌표데이터로 층을 구분하는 알고리즘을 구현하여 콩 작물의 시계열 데이터를 입력하여 성장을 분석하는 프로그램을 개발하였고, 그 결과 콩 작물의 제2~3복엽까지 성장 시기를 판단할 수 있었다.

핵심주제어: 인공지능, 객체 검출, 컴퓨터 비전, 합성곱 신경망, 표현형 분석

Abstract Soybeans are one of the world's top five staple crops and a major source of plant-based protein. Due to their susceptibility to climate change, which can significantly impact grain production, the National Agricultural Science Institute is conducting research on crop phenotypes through growth analysis of various soybean varieties. While the process of capturing growth progression photos of soybeans is automated, the verification, recording, and analysis of growth stages are currently done manually. In this paper, we designed and trained a YOLOv5s model to detect soybean leaf objects from image data of soybean plants and a Convolution Neural Network (CNN) model to judgement the unfolding status of the detected soybean leaves. We combined these two models and implemented an algorithm that distinguishes layers based on the coordinates of detected soybean leaves. As a result, we developed a program that takes time-series data of soybeans as input and performs growth analysis. The program can accurately determine the growth stages of soybeans up to the second or third compound leaves.

Keywords: Artificial intelligence, Object detect, Computer vision, Phenotype analysis

* Corresponding Author: greenksl5405@korea.kr, inchchoi@korea.kr
+ 이 논문은 농촌진흥청 연구사업(과제번호: PJ01486501)의 지원에 의해 이루어진 것임.

Manuscript received September 22, 2023 / revised October 20, 2023 / accepted October 24, 2023

1) 국립농업과학원 농업공학부, 제1공동저자
2) 국립농업과학원 농업생명자원부, 제1공동저자
3) 국립농업과학원 농업공학부, 제2저자
4) 국립식량과학원 중부작물부, 제2저자
5) 국립농업과학원 농업생명자원부, 공동교신저자
6) 국립농업과학원 농업공학부, 공동교신저자

1. 서론

콩(*Glycine max* (L.) Merr.)은 쌀과 더불어 세계적으로 중요한 식량작물 중 하나로, 식물성 단백질과 지방의 공급원으로써 큰 역할을 하고 있다 (Cho, 2006; Cho et al., 2021). 이에 세계적으로 콩에 대한 여러 연구가 진행되고 있다. 관련 연구로 콩밭에서 촬영한 영상 데이터에서 딥러닝 모델별 꽃, 꼬투리 검출 성능 비교 연구 (Pratama et al., 2020), 드론으로 촬영한 브라질 지역의 콩밭 다중스펙트럼(Multispectral) 데이터에서 딥러닝, 머신러닝을 통해 생장 시기, 작물의 크기, 곡물 수확량을 예측하고 결과를 비교하는 연구 (Teodoro et al., 2021) 등이 있다. 국립농업과학원에서는 다양한 콩 품종의 특성을 파악하고자 생장분석을 통해 작물의 형태적 특징을 수치화, 객관화 하여 분석하는 작물표현체 연구를 진행중이다(Jeong et al., 2020). 콩은 떡잎, 초엽, 제1복엽, 제2복엽에서 제5복엽까지 잎이 전개되며, 이러한 잎의 전개시기는 작물 생장의 중요한 지표이다. 콩의 초엽에서 제2복엽까지의 전개시기를 판단하면 다음 복엽이 생성되는 시기를 예측할 수 있으며 이를 통해 품종별 생장 차이를 발견할 수 있고 품종별 생장에 적합한 기후, 특성들을 파악할 수 있다. 이러한 생장 분석 연구과정은 콩 작물의 사진을 주기적으로 촬영하여 날짜, 시간별로 저장하는 단계와 사진을 확인하고 생장 상태를 기록, 분석하는 단계로 나뉜다. 콩의 영상 데이터는 대량검정시스템 컨베이어와 촬영 장비를 결합한 자동화된 시스템으로 수행되지만 작물 사진에서 생장 데이터 추출 및 기록은 수작업으로 진행되고 있어 인력이 소모되고 일정 시간에만 수행할 수 있는 단점이 존재한다. 따라서 생장 데이터 추출 및 기록, 분석 단계를 자동화한다면 생장 분석 과정을 수행하는 여러 연구 기관들의 인력 소모를 줄이고 시간에 구애받지 않고 작물의 생장 분석 과정을 수행할 수 있기 때문에 인공지능을 기반으로 한 생장 분석 방법에 대해 연구를 진행하였다.

먼저 생장 상태를 판단하기 위해 콩잎 객체 검출 작업을 수행해야 한다. 과거에는 주로 컴퓨터 비전 기반 지역 특징 매칭 방법을 통해 영

상 데이터에서 객체의 클래스를 분류하거나 객체의 위치를 찾는 객체 인식 작업을 수행했다. 이는 개발자가 특징점들을 선택하고, 지역 스케일 불변(local scale-invariant) 특징점 검출 방식을 사용해 특징벡터를 추출하는 방식으로 기계학습 초기 방법 중 하나인 Support Vector Machine (SVM)을 사용해 인식률을 높였다 (Lowe, 1999; Bay et al., 2006; Ko and Sim, 2017; Lee and Kim, 2020). 이러한 고전적인 기계학습과 다르게 딥러닝 알고리즘은 특징점을 선택하지 않아도 학습을 진행하면서 스스로 특징 추출, 분류 과정을 동시에 최적화한다. 또한 영상 데이터에서 공간적인 정보를 유지하면서 인공 신경망의 학습 효율과 정확도를 높이는 Convolutional Neural Network (CNN)의 등장 이후 AlexNet, GoogLeNet, YOLO 등 많은 딥러닝 알고리즘이 생겨났다. CNN 기반 딥러닝 모델들은 고전적인 기계학습 방식보다 영상 데이터에서 객체를 인식하는 성능이 뛰어났다 (Szegedy et al., 2015; Redmon et al., 2016; Krizhevsky et al., 2017). 콩 작물의 경우 잎의 개수가 많고 크기가 다양하게 나타난다. 따라서 본 연구에서는 콩잎 검출을 위해 데이터 처리 속도가 빠르고 한번에 여러 객체를 검출할 수 있는 YOLO 알고리즘을 사용하였다. 여러 객체를 검출할 수 있는 YOLOv5 알고리즘을 사용하여 콩잎 객체를 검출하고 검출된 콩잎의 전개 여부를 판단하기 위해 영상 데이터에서 패턴과 특징을 탐지하고 추출하는데 특화된 구조를 가진 CNN 모델을 설계 및 학습하였다. 영상 데이터에서 객체를 검출하는 YOLO 알고리즘을 통해 콩잎을 검출하면서 전개 여부를 파악하려면 객체의 종류(Class)가 ‘전개된 콩잎, 전개되지 않은 콩잎’으로 늘어나며 영상 데이터를 전처리 하는 시간이 증대된다. 또한 종류가 늘어난 만큼 학습 데이터를 추가해야 한다. 본 연구는 생장 분석 자동화를 위한 방법에 대한 초기 연구로 추후 각 콩잎에 대한 표현체 분석 작업도 수행할 수 있도록 콩잎만 검출하는 모델을 학습하여 연구 확장성을 염두에 두었다. 그리고 검출된 콩잎의 좌표 데이터를 활용해 생장 상태를 판단하는 생장 분석 프로그램을 개발하였다.

2. 시스템 구성

2.1 대량검정시스템

콩은 대량검정시스템의 컨베이어 위에 위치해 있으며 컨베이어가 촬영 공간으로 이동하여 영상 데이터를 획득한다. 촬영은 0°, 120°, 240°로 오전 1회, 오후 1회 수행하여 각 작물당 총 6장의 영상 데이터를 얻는다. Fig. 1은 대량검정시스템 획득한 영상 데이터의 관심영역(Region of Interest; ROI) 부분으로 작물의 성장상태, 일련번호를 저장한다. 저장된 영상 데이터의 파일명은 'ROI_B_06_352018-09-11 14_55_07VIS_SV0'으로 작물의 일련번호(B_06_35), 촬영 날짜(2018-09-11), 촬영 시간(14:55:07), 영상장비의 정보(VIS_SV0)를 저장한다. 획득한 영상 데이터의 원본 해상도는 4,384px*6,576px 이며 본 논문에서 사용한 ROI 영상 데이터의 해상도는 2,401px*2,951px 이다.



Fig. 1 Image data acquired by High-throughput imaging system

2.2 콩잎 검출용 인공지능 모델

본 연구에서는 콩 작물의 영상 데이터에서 콩잎 객체를 검출하기 위한 모델로 YOLOv5s를 선정했다. YOLO는 입력된 영상 데이터에서 객

체의 위치를 찾는 문제(Localization)와 객체가 무엇인지 분류(Classification)하는 문제를 한번에 해결하는 1-Stage 방식의 알고리즘이다. 격자 칸으로(Grid cell)로 나누고 그 안에 객체의 중심이 들어가면 해당 격자 칸이 물체를 감지한다. 각 격자 칸은 B 개의 경계 상자(Bounding box)와 각 경계 상자에 대한 신뢰도(Confidence score)를 예측한다. 또한 각 격자 칸은 객체를 포함하고 있을 때, 클래스별 확률도 계산한다. 이후 각 경계 상자에 대한 신뢰점수와 클래스별 확률을 곱하여 영상 데이터에서 객체를 검출하는 과정을 수행한다 (Redmon et al., 2016).

Fig. 1과 같은 콩 작물 영상 데이터에서 각각의 콩잎 객체를 검출하기 위해 Fig. 2와 같이 데이터 전처리(Labeling) 작업을 수행하여 YOLOv5s 모델을 학습하였다. YOLOv5s의 가장 큰 변화는 C언어로 구현된 Backbone을 Pytorch로 구현한 것으로 CSP-Darknet을 사용하며 구조는 Fig. 3과 같다. Backbone은 입력된 영상 데이터를 합성곱(Conv; Convolutional, C3) 연산을 통해 특징 맵(feature map)으로 변환해주는 역할을 수행한다. Neck은 특징 맵을 업샘플링(Upsampling)하여 크기를 키운 후 다른 특징맵과 연결(Concat; concatenate)하여 Backbone에서 추출된 특징점들을 적절하게 조화시키는 역할을 한다. Head는 위치 추정(Localization) 및 객체 분류(classification)를 수행하는 단계이다. Neck에서 여러 특징 맵을 두었기 때문에 하나의 영상 데이터에서 여러개의 콩잎 객체를 효과

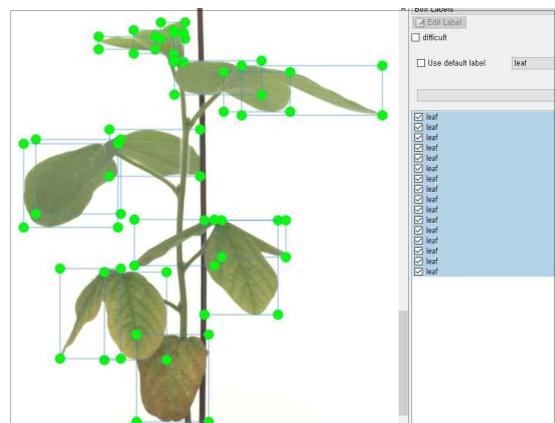


Fig. 2 Example of labeled data

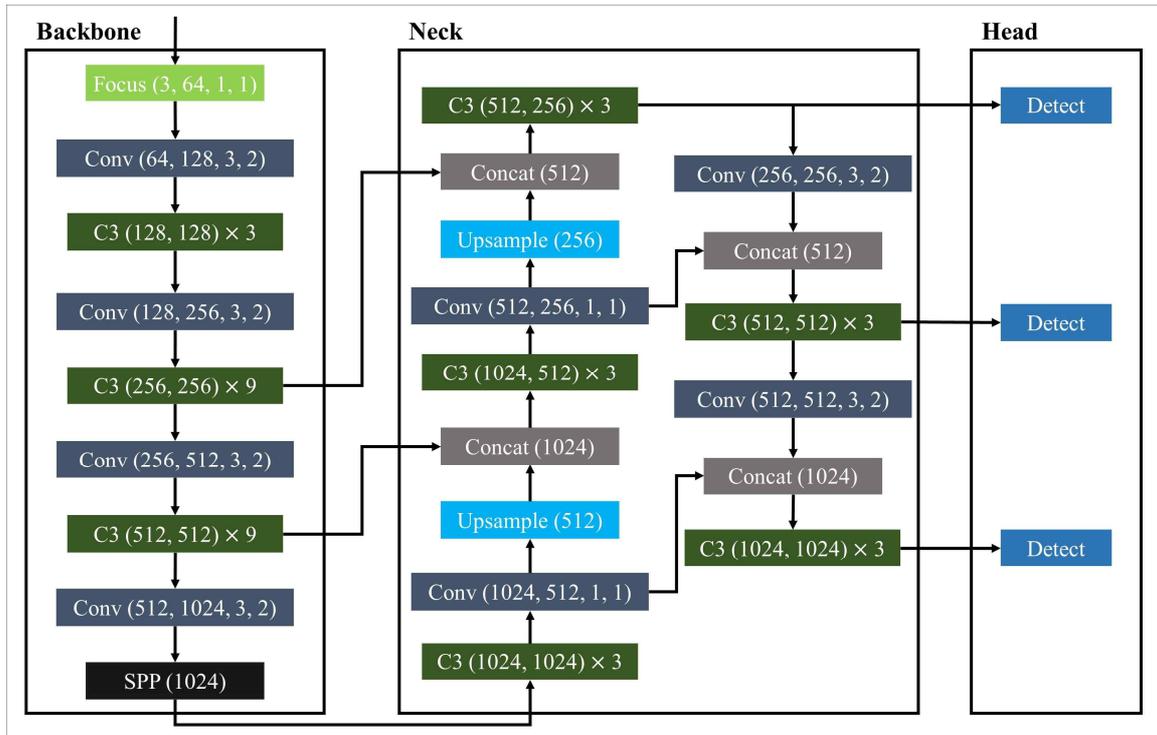


Fig. 3 Structure of YOLOv5s model consisting of the Backbone, Neck, and Head; multiple objects can be detected in one image data.

적으로 검출할 수 있다.

2.3 콩잎 전개 판단용 CNN 모델

새로운 콩잎이 생성되었을 때 잎의 끝단은 위를 향하고 있으며 전개되면서 점점 바닥쪽으로 기울게된다. 콩 성장분석 과정에서 제 n 복엽 전개기(V_n)는 콩잎의 기울기가 수평 이하로 떨어졌을 때 완전 전개되었다고 판단한다.

CNN은 영상 데이터에서 패턴과 특징을 탐지하고 추출하는데 특화된 구조를 가지고 있는 딥러닝 모델의 한 종류이며 본 논문에서 YOLOv5s 모델로 검출한 콩잎의 영상 데이터로 콩잎의 전개 여부를 판단하기 위해 Table 1과 같이 CNN 모델을 설계했다 (Gu et al.; 2018). 본 연구를 통해 설계한 모델은 일반적인 CNN 구조(Convolution, Pooling, Activation function)를 갖는다. Convolution(Conv) 레이어는 커널을 통해 영상데이터에서 특징맵을 추출하는 역할을

수행하며 MaxPooling 레이어는 특징맵의 크기를 줄여 연산 비용을 줄이는 역할을 수행한다. 콩잎 영상 데이터는 $[-1, 3, 64, 64]$ 형태로 모델에 입력되며 Conv, Linear 연산 이후 Rectified Linear Unit (ReLU) 활성화 함수를 사용했다. 입력된 영상 데이터의 크기가 작기 때문에 특징점을 추출할 수 있는 가장 작은 크기의 Kernel(3)을 사용하였고, Convolution 연산 이후 영상 데이터의 크기를 유지하기 위해 Kernel이 이동하는 폭 Stride는 1, 테두리에 0값을 삽입하는 Padding 옵션을 1로 설정하였다 (LeCun et al., 1998). 또한 모델 학습중 과적합(Overfitting)을 방지하기 위해 두 번째 Conv, 첫 번째 Linear 연산 이후 각각 0.25, 0.5의 드롭아웃(Dropout) 기법을 적용했다. 높은 값을 가진 인덱스를 찾는 함수 argmax를 모델의 출력값에 적용하여 전개 여부를 0(False), 1(True) 값으로 판단한다.

Table 1 Structure of CNN Model

Layer	Output Shape
Conv (K:3, S:1, P:1)	[-1, 32, 64, 64]
MaxPooling (K:2, S:2)	[-1, 32, 32, 32]
Conv (K:3, S:1, P:1)	[-1, 64, 32, 32]
MaxPooling (K:2, S:2)	[-1, 64, 16, 16]
Linear	[-1, 256]
Linear	[-1, 2]

2.4 콩잎 층 구분 알고리즘

Fig. 4는 작물의 콩잎 층 구분 시 활용하는 현재 입력된 영상 데이터, 직전에 입력된 영상 데이터를 나타낸다. 즉 비교되는 영상 데이터는 같은날 오전 데이터 혹은 전날 오후 데이터다. 두 영상데이터에서 검출된 콩잎의 바운딩 박스의 겹침정도(IoU)가 80% 이상인 객체는 같은 콩잎으로 판단하며 콩잎의 층, 전개여부 데이터를 물려받는다. 콩 작물의 특성상 어느정도 자란 콩잎들은 바운딩박스의 상단부가 거의 일치하고 처음 자라기 시작하는 콩잎들은 바운딩박스의 하단부가 거의 일치한다. 하지만 같은 층에서도 길게 자라는 콩잎이 존재하기 때문에 본 연구에서는 콩 작물 영상 데이터에서 검출된 콩잎의 바운딩박스 상단, 하단, 중심 y좌표값을 기반으로 작물의 층을 구분하는 알고리즘을 파이썬(Python)으로 Code 1과 같이 구현했다.

2.5 성장 분석 프로그램

본 연구를 통해 학습한 콩잎 검출용 YOLOv5s모델, 콩잎 전개 판단용 CNN 모델, 콩잎 층 구분 알고리즘을 결합한 성장 분석 프로그램의 순서도는 Fig. 5와 같다. 프로그램에 콩 작물의 시계열 영상 데이터가 입력된 후 영상 데이터의 파일명에서 일련번호, 촬영 날짜, 촬영 시간의 데이터를 추출한 뒤 본 연구를 통해 학습한 YOLOv5s 모델로 콩잎을 검출한다. 해당 작물의 일련번호로 저장된 데이터가 있을 시 이전 시간대에 저장된 데이터를 불러온 후 콩잎의 층 구분 작업을 수행한다. 층 구분 작업

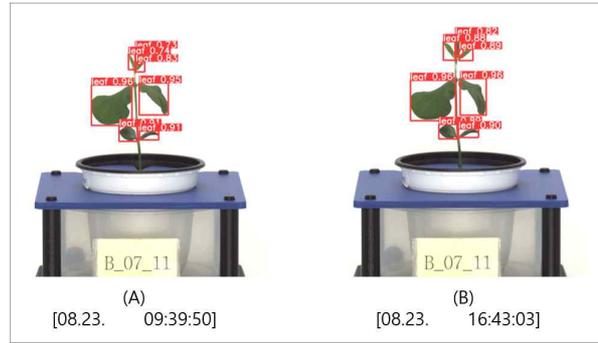


Fig. 4 Example of image data used for floor classification of soybean leaves. (A) Image data previously entered. (B) Image data currently entered.

이 수행된 후 각각의 콩잎은 바운딩박스 좌표값 (x_1, y_1, x_2, y_2)과 0~1 사이로 정규화된 값(w, y, x, h), 알고리즘으로 판단한 층(Floor)값, 콩잎 전개 여부(-1) 데이터를 저장한다. 그리고 각각의 본 연구를 통해 학습한 콩잎의 전개 판단 CNN모델로 콩잎 전개 여부 데이터를 갱신한다. 위 작업을 입력된 시계열 영상 데이터에 반복하며 마지막 영상 데이터까지 작업이 수행된 이후 더 이상 분석할 영상 데이터가 존재하지 않으면 데이터를 취합하여 날짜, 시간별 성장 분석 결과를 저장한다.

2.6 모델 성능 평가 방법

객체 검출 인공지능 모델은 예측 결과와 실제 값을 비교한 혼동행렬(Confusion matrix)을 이용하여 성능을 평가할 수 있다. 혼동행렬의 True Positive (TP)는 관심 범주를 정확하게 분류한 값을 나타내고 False Positive (FP)는 관심 범주가 아닌 것을 잘못 분류한 값을 나타낸다. 또한 True Negative (TN)는 관심 범주가 아닌 것을 정확하게 분류한 값을 나타내고 False Negative (FN)는 관심 범주를 잘못 분류한 값을 나타낸다. 본 논문에서는 혼동행렬의 값으로 정확도(Accuracy), 정밀도(Precision), 재현율(Recall) 총 3가지의 척도를 평가하였다. 각 척도는 식 1, 2, 3과 같이 계산할 수 있다.

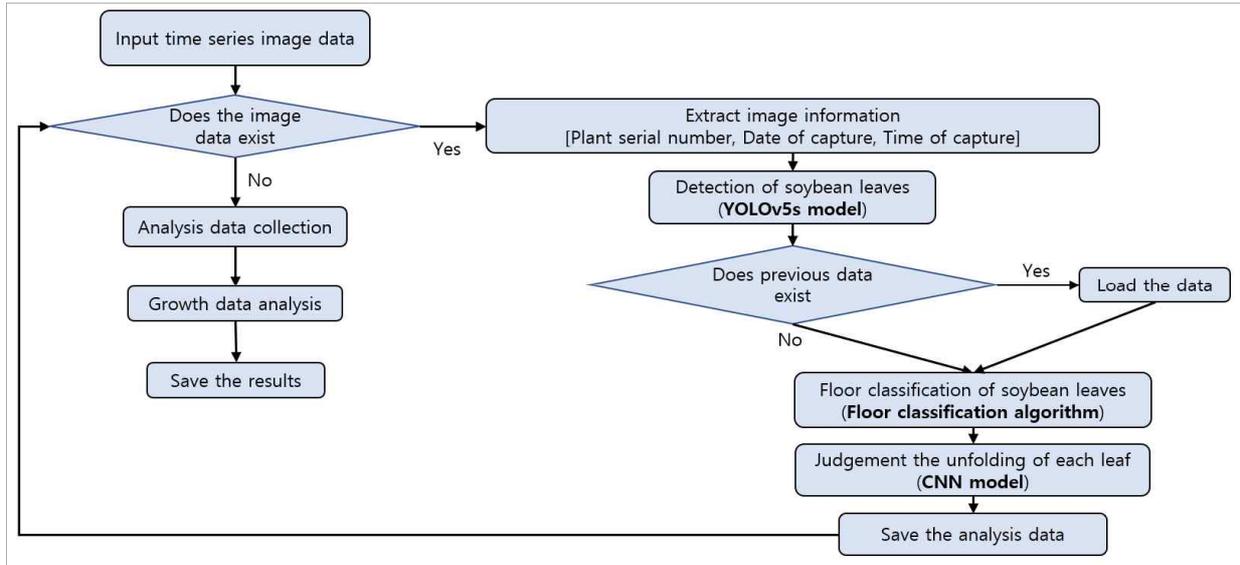


Fig. 5 Flowchart of the growth analysis program

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

Table 2 Hyper parameter used to train the model.

Option	Value
Epoch	300
Batch size	4
Resize	1280
Optimizer	Stochastic Gradient Descent
Learning rate	0.01
Momentum	0.937
Weight decay	0.0005

3. 결 과

3.1 콩잎 검출 모델 학습 및 성능평가 결과

모델 학습을 위해 작물 촬영 장비로 획득한 Hefeng 품종의 작물 영상 데이터는 좌표 데이터 생성 프로그램(LabelImg, MIT)을 사용하여 전처리 작업을 진행하여 270장의 데이터셋을 구성했다. 그 다음 220장의 데이터셋으로 YOLOv5 학습을 진행하였다. YOLOv5s 학습에 사용한 하이퍼 파라미터는 Table 2와 같다. 학습 결과 그래프는 Fig. 6과 같다. 학습 결과 그래프상 학습이 진행될수록 성능이 향상되는 것을 확인할 수 있다.

모델 학습을 마치고 나머지 50장의 데이터셋으로 학습된 모델의 혼동행렬을 이용하여 성능 평가를 진행하였다. 성능 평가 옵션으로 검출한 객체의 신뢰도(Confidence-threshold) 값을 0.7 이상으로 설정하였고 그 결과 정확도(Accuracy) 73.0%, 정밀도(Precision) 95.0%, 재현율(Recall) 75.9%의 성능을 보였다 (Table 3). 학습이 완료된 콩잎 검출 모델의 출력 결과는 Fig. 7과 같다. 연구를 통해 학습한 콩잎 검출 모델은 관심 범주가 콩잎 1가지이기 때문에 관심 범주가 아닌 것을 정확하게 분류한 값을 나타내는 TN 항목은 사용하지 않는다.

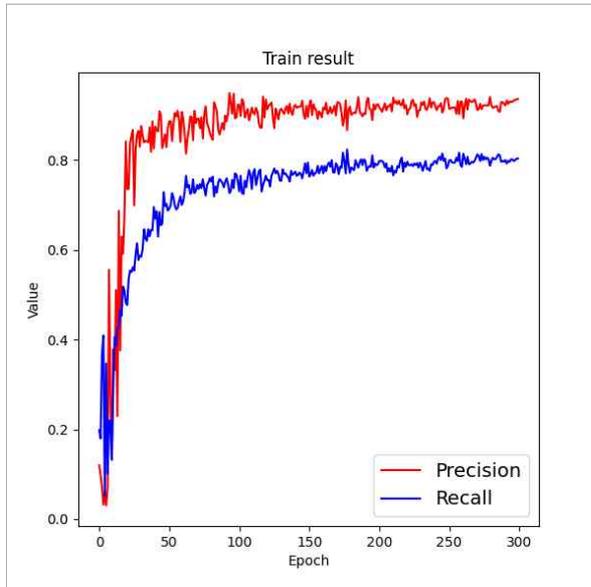


Fig. 6 Result of YOLOv5s model training.

Table 3 Validation results of the model

Type	Value
True Positive (TP)	414
False Positive (FP)	22
True Negative (TN)	0
False Negative (FN)	131
Accuracy	73.0%
Precision	95.0%
Recall	75.9%

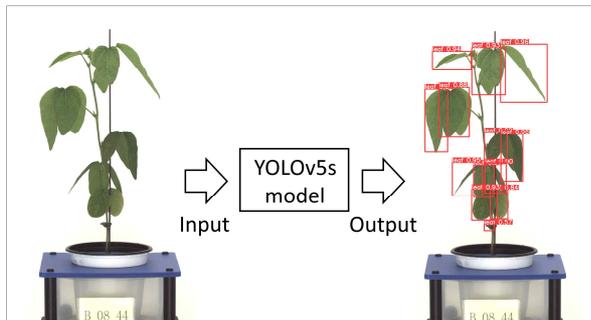


Fig. 7 Data input and output of the model

성능평가 결과 콩의 영상 데이터에서 배경과 콩잎 객체만 구분하는 정밀도(Precision)는 높게

측정되었다. 하지만 작은 콩잎들, 가려진 콩잎들이 검출되지 않아 재현율(Recall), 정확도(Accuracy) 성능이 다소 낮게 측정되었다. 생장 분석을 위해 검출된 콩잎의 데이터는 좌표값을 기준으로 위에서 아래 순서대로 정렬되어 텍스트 파일에 저장된다.

3.2 CNN 모델 학습 및 성능평가 결과

모델 학습을 위해 Dongdou, Hedou, Hefeng, 다윈, 대원, 선풍, 한남, 흑성 8품종의 콩 작물 영상 데이터에서 콩잎 검출 모델을 통해 Fig. 8 과 같은 64*64 크기의 콩잎 영상 데이터 4,520 장을 획득하였다. 이후 콩잎의 전개 여부를 판단하여 전처리 작업을 진행하고 4,102장의 학습용, 418장의 성능 평가용 데이터셋을 구성하였다.



Fig. 8 Example of leaf image data

CNN 모델은 학습 횟수(epoch) : 150, 학습 영상 분할 크기(batch size) : 16으로 설정하고 Adaptive moment estimation (Adam) 최적화 알고리즘(Optimizer)과 Pytorch에서 제공하는 CrossEntropyLoss 손실함수(Loss function)를 사용하여 학습을 진행하였으며 학습 결과는 Fig. 9와 같다.

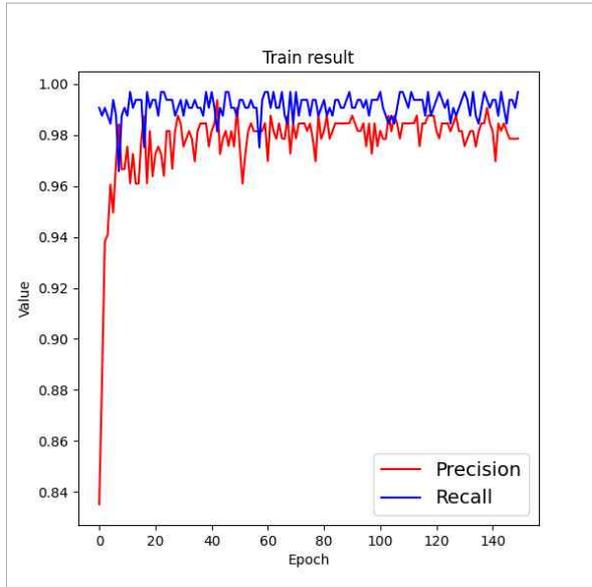


Fig. 9 Result of CNN model training.

모델 학습을 마치고 나머지 418장의 데이터셋으로 혼동행렬을 계산하여 성능 평가를 진행하였으며 그 결과 정확도(Accuracy) 98.3%, 정밀도(Precision) 98.4%, 재현율(Recall) 99.3%의 성능을 보였다 (Table 4). 성능 평가 데이터셋 일부의 콩잎 전개 여부 판단 결과는 Fig. 10과 같다.

Table 4 Validation results of the model

Type	Value
True Positive (TP)	320
False Positive (FP)	5
True Negative (TN)	91
False Negative (FN)	2
Accuracy	98.3%
Precision	98.4%
Recall	99.3%

3.3 실측, 프로그램 데이터 비교 결과

본 연구에서는 2018년 농촌진흥청 내 작물표현체연구동에서 8월 21일부터 9월 11일까지 촬영된 Hefeng 품종의 생장 과정을 담고있는 영상 데이터를 사용했다.

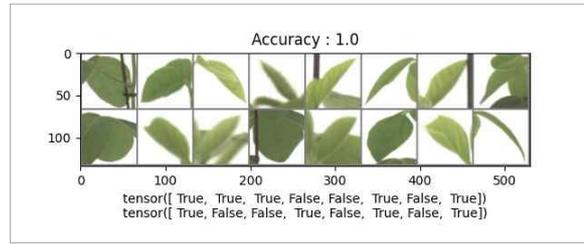


Fig. 10 Example of prediction results

Table 5.는 생장 분석 프로그램과 실측 데이터의 비교 결과를 나타낸다. 각 수치는 해당 날짜에 실측된 작물 생장 상태(N-복엽)을 나타내며, 0은 초엽을 의미한다. 실측 데이터와 다른 판단을 한 경우 *로 표기하였다. 비교 결과 제 1복엽이 전개되는 시기는 정확하게 판단하였고, 제 2~3복엽이 전개되는 시기는 약 60%의 정확도로 판단하였다.

4. 결론 및 향후 연구방향

본 연구는 국립농업과학원에서 반자동화로 수행되고 있는 콩 생장 분석 과정을 자동화 시키기 위해 콩의 시계열 영상 데이터를 YOLOv5s 모델로 콩잎 객체를 검출했다. 그리고 CNN모델을 통해 콩잎 전개 여부를 판단하여 생장 분석 과정을 수행하는 프로그램을 개발했다. Aich and Stavness(2017)와 Lu et al.(2021)는 작물 영상 데이터에서 딥러닝을 사용하여 식물 잎 수를 세는 방법을 제안했지만 이는 위에서 내려다보는(Top view) 영상 데이터에서 사용하는 방식이기 때문에 본 연구에 적용시킬 수 없었고 객체 검출 알고리즘인 YOLOv5s 모델을 학습시켜 시계열 작물 영상 데이터에서 콩잎 객체를 73.0%의 정확도(Accuracy), 95.0%의 정밀도(Precision), 75.9%의 재현율(Recall)로 검출할 수 있었다. 그리고 검출된 콩잎의 전개 여부를 판단하기 위해 CNN 모델을 설계하고 학습시킨 결과 98.3%의 정확도(Accuracy), 98.4%의 정밀도(Precision), 99.3%의 재현율(Recall)로 전개 여부를 판단할 수 있었다. 두 모델을 결합하여 생장 분석 프로그램을 개발하였고 실측 데이

Table 5 Comparison of growth state measurement data and program output data
 (* means that the actual data and the program output are different)

Sample	8.22	8.23	8.24	8.26	8.27	8.28	8.29	8.30	8.31	9.02	9.03
B_07_08	0	0	0	1	1	1	2	2	2	3	4
B_07_17	0	0	0	1	1	*2	2	2	*3	3	4
B_07_26	0	0	0	1	1	1	2	2	*3	3	*4
B_07_35	0	0	0	1	1	1	2	2	2	3	*3
B_08_08	0	0	0	1	1	1	*2	2	2	3	*4

터와 비교했을 때 제2~3복엽까지의 생장 시기를 판단할 수 있었다.

본 연구를 통해 구현한 층 구분 알고리즘의 경우 검출된 콩잎의 바운딩박스 좌표에 의존하기 때문에 겹쳐있거나 가려져있는 콩잎이 존재하는 경우 층 구분이 제대로 이루어지지 않는 단점이 존재한다. 따라서 향후에는 콩잎 검출을 개별의 콩잎이 아닌 같은 층의 콩잎을 한번에 검출하는 방식으로 변경한다면 생장 분석 프로그램의 정확도를 높이는 연구가 가능할 것으로 보인다. 또한 Saleem et al.(2019)의 식물체 잎에서 딥러닝을 사용하여 질병을 검출하는 방법과 Karlekar and Seal(2020)의 콩잎에서 질병을 검출하는 방식을 본 연구에 적용시킨다면 콩작물의 생장분석과 동시에 질병 연구 또한 가능할 것으로 보인다.

References

- Aich, S. and Stavness, I. (2017). Leaf counting with deep convolutional and deconvolutional networks. In Proceedings of the IEEE international conference on computer vision workshops (pp. 2080-2089).
- Bay, H., Tuytelaars, T. and Van Gool, L.(2006). Surf: Speeded up robust features. Lecture notes in computer science, 3951, 404-417. https://doi.org/10.1007/11744023_32.
- Cho, J. H. (2006). Effect of planting date and cultivation method on soybean growth in paddy field. Korean journal of organic agriculture, 14(2), 191-204.
- Cho, C., Kim, D. Y., Choi, M. S., Jin, M. and Seo, M. S. (2021). Efficient isolation and gene transfer of protoplast in korean soybean (*Glycine Max (L.) Merr.*) cultivars. Korean journal of breeding science, 53(3), 230-239.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T. (2018). Recent advances in convolutional neural networks. Pattern recognition, 77, 354-377.
- Jeong, Y. S., Lee, H. R., Baek, J. H., Kim, K. H., Chung, Y. S. and Lee, C. W. (2020). Deep Learning-based rice seed segmentation for phynotyping. Journal of the Korea Industrial Information Systems Research. 25(5), 23-29.
- Ko, K. E. and Sim, K. B. (2017). Trend of object recognition and detection technology using deep learning. Journal of Control Robotics and Systems, 23(3), 17-24.
- Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks. Communications of the ACM, 60(6), 84-90. <https://doi.org/10.1145/3065386>.
- Karlekar, A. and Seal, A. (2020). SoyNet: Soybean leaf diseases classification. Computers and Electronics in Agriculture, 172, 105342.
- Lowe, D. G. (1999). Object recognition from

local scale-invariant features. In Proceedings of the seventh IEEE international conference on computer vision, 2, 1150-1157. <https://doi.org/10.1109/ICCV.1999.790410>.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11), 2278-2324.

Lee, Y. H. and Kim, Y. (2020). Comparison of CNN and YOLO for Object Detection. *Journal of the semiconductor & display technology*, 19(1), 85-92.

Lu, S., Song, Z., Chen, W., Qian, T., Zhang, Y., Chen, M. and Li, G. (2021). Counting dense leaves under natural environments via an improved deep-learning-based object detection algorithm. *Agriculture*, 11(10), 1003.

Pratama, M. T., Kim, S., Ozawa, S., Ohkawa, T., Chona, Y., Tsuji, H. and Murakami, N. (2020, July). Deep learning-based object detection for crop monitoring in soybean fields. In 2020 International Joint Conference on Neural Networks (IJCNN) (pp. 1-7). IEEE.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 779-788. <https://doi.org/10.48550/arXiv.1506.02640>.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D. and Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1-9. <https://doi.org/10.48550/arXiv.1409.4842>.

Saleem, M. H., Potgieter, J. and Arif, K. M. (2019). Plant disease detection and classification by deep learning, *Plants*, 8(11), 468.

Teodoro, P. E., Teodoro, L. P. R., Baio, F. H. R., da Silva Junior, C. A., dos Santos, R.

G., Ramos, A. P. M., Pinheiro, M. M. F., Osco, L. P., Gonçalves, W. N., Carneiro, A. M., Junior, J. M., Pistori, H. and Shiratsuchi, L. S. (2021). Predicting days to maturity, plant height, and grain yield in soybean: A machine and deep learning approach using multispectral data. *Remote Sensing*, 13(22), 4632.



전 문 석 (Moon-Seok Jeon)

- 학생회원
- 군산대학교 컴퓨터정보공학과 공학사
- (현재) 군산대학교 컴퓨터정보공학과 학연협동 석사과정, 국립농업과학원 농업공학부

학연연구원

- 관심분야: 인공지능, 머신러닝, 컴퓨터 비전, 영상처리



김 영 태 (Yeongtae Kim)

- 경북대학교 농생물학과 농학사
- 경북대학교 농생물학과 농학석사
- 경북대학교 농생물학과 농학박사
- (현재) 농촌진흥청 국립농업과학원 전문연구원
- 관심분야: 표현체, 영상처리,

초분광



정 유 석 (Yuseok Jeong)

- 학생회원
- 군산대학교 컴퓨터정보공학과 공학사
- 군산대학교 컴퓨터정보공학과 공학석사
- (현재) 군산대학교 컴퓨터정보공학과 학연협동 박사과정, 국립농업과학원 농업공학부 학연연구원

농업공학부 학연연구원

- 관심분야: 딥러닝, 영상처리, 컴퓨터비전



배 효 준 (Hyojun Bae)

- 원광대학교 원예학과 농학사
- 전남대학교 원예학과 농학석사
- (현재) 원광대학교 원예학과 박사과정, 국립농업과학원 농업공학부 전문연구원
- 관심분야: 시설원예, 복합환

경제어, 표현체



이 채 원 (Chaewon Lee)

- 가톨릭대학교 생명공학과 공학사
- 전북대학교 농학과 농학석사
- 전북대학교 농학과 농학박사
- (현재) 농촌진흥청 국립식량과학원 농업연구사
- 관심분야: 콩 재배기술, 디지털 농업, 표현체

털 농업, 표현체



김 송 림 (Song Lim Kim)

- 정회원
- 한림대학교 생물학과 이학사
- 한림대학교 생물학과 이학석사
- 포항공대 생명과학과 이학박사
- (현재) 농촌진흥청 국립농업과학원 농업연구사



최 인 찬 (Inchan Choi)

- 정회원
- 전남대학교 물리·전기공학과 이학·공학사
- 중앙대학교 전자전기공학부 공학석사
- 중앙대학교 전자전기공학부

공학박사

- (현재) 농촌진흥청 국립농업과학원 농업연구관
- 관심분야: 지능시스템, 인공지능, 복합환경제어

Code 1. 검출된 콩잎 객체의 층 구분 알고리즘

```
# leaf_list : YOLOv5s로 검출한 콩잎 객체의 좌표값.

# all_top_list : 콩잎 객체별 상단 좌표값 차이를 저장.
# all_center_list : 콩잎 객체별 중심 좌표값 차이를 저장.
# all_bottom_list : 콩잎 객체별 하단 좌표값 차이를 저장.

# diff : 29 -> 이미지 크기의 1% 오차를 기준으로 사용.

# 콩잎 객체별 좌표 차이값을 계산하여 저장
for i in range(len(leaf_list)):
    if j in range(len(leaf_list)):
        #인덱스가 같을 경우 콩잎 객체가 동일하므로 0값 저장.
        if i == j:
            top_list.append(0)
            center_list.append(0)
            bottom_list.append(0)
        else:
            top_list.append(abs(leaf_list[i][top] - leaf_list[j][top]))
            center_list.append(abs(leaf_list[i][center] - leaf_list[j][center]))
            bottom_list.append(abs(leaf_list[i][bottom] - leaf_list[j][bottom]))

    all_top_list.append(top_list)
    all_center_list.append(center_list)
    all_bottom_list.append(bottom_list)

# predict_list : 콩잎 객체별 N번째 잎과 같은 층이라고 판단한 인덱스(N)를 저장.

for i in range(len(leaf_list)):
    min_value = -1
    min_index = -1
    #상단(Top) 좌표 비교
    for j in range(len(all_top_list[i])):
        # 같은 콩잎 객체는 건너뛴.
        if all_top_list[i][j] == 0:
            continue
```

```

else :
    # 모든 쿡잉 별 상단(Top) 좌표 차이를 확인하며 가장 차이가 적은 값과 인덱스를 저장.
    if min_value > all_top_list[i][j] or min_value == -1:
        min_value = all_top_list[i][j]
        min_index = j

# 상단(Top) 좌표 비교 후 가장 차이가 작은 값이 이미지 크기의 1% 보다 작다면 해당 인덱
# 스의 쿡잉과 같은 층에 있다고 판단.
# 그렇지 않으면 중심(Center) 좌표 비교 시작.
if min_value < diff:
    predict_list[i] = min_index
else:
    min_value = -1
    min_index = -1
    # 중심(Center) 좌표 비교
    for j in range(len(all_center_list[i])):
        # 같은 쿡잉 객체는 건너뛴.
        if all_center_list[i][j] == 0:
            continue
        else:
            # 모든 쿡잉 별 중심(Center) 좌표 차이를 확인하며 가장 차이가 적은 값과 인덱
            # 스를 저장.
            if min_value > all_center_list[i][j] or min_value == -1:
                min_value = all_center_list[i][j]
                min_index = j

# 중심(Center) 좌표 비교 후 가장 차이가 작은 값이 이미지 크기의 1% 보다 작다면 해
# 당 인덱스의 쿡잉과 같은 층에 있다고 판단.
# 그렇지 않으면 하단(Bottom) 좌표 비교 시작.
if min_value < diff:
    predict_list[i] = min_index
else:
    min_value = -1
    min_index = -1
    # 하단(Bottom) 좌표 비교
    for j in range(len(all_bottom_list[i])):
        # 같은 쿡잉 객체는 건너뛴.
        if all_bottom_list[i][j] == 0:
            continue

```

```

else:
    # 모든 콩잎 별 중심(Center) 좌표 차이를 확인하며 가장 차이가 적은 값과
    # 인덱스를 저장.
    if min_value > all_bottom_list[i][j] or min_value == -1:
        min_value = all_bottom_list[i][j]
        min_index = j
    # 하단(Bottom) 좌표 비교 후 가장 차이가 작은 값이 이미지 크기의 1% 보다 작다면
    # 해당 인덱스의 콩잎과 같은 층에 있다고 판단.
    # 그렇지 않으면 자신의 인덱스 저장 -> Y 축 좌표가 비슷한 콩잎 객체가 존재하지
    # 않음.
    if min_value < diff:
        predict_list[i] = min_index
    else:
        predict_list[i] = i

# predict_list 결과 예시
# predict_list = [1, 0, 3, 2, 5, 4]
# predict_list 정렬

for i in range(len(predict_list)):
    for j in range(len(predict_list)):
        if i is not j:
            if predict_list[j] == i:
                if predict_list[j] is not predict_list[i]:
                    predict_list[j] = predict_list[i]
# 정렬 결과 예시 -> [1, 1, 3, 3, 5, 5]

# 중복되는 숫자를 제거하여 층의 개수를 파악
n = len(list(set(predict_list))) - 1

# 검출된 콩잎의 층 정보를 정렬하여 출력
x = predict_list[0]
for i in range(len(predict_list)):
    if predict_list[i] > x:
        x = predict_list[i]
        n -= 1
    predict_list[i] = n
# 알고리즘 결과 예시 -> [2, 2, 1, 1, 0, 0]

```