Detection and segmentation of plant images for phenotyping

Department of Electrical and Computer Engineering

Team Number – 114

Presented By,

Ramachandran Sekanipuram Srikanthan (rsekani)

Mayur Sanap (msanap)

Chinmayee Dande (cpdande)

Guide By,

Prof. Edgar Lobaton

Outline

- Goal and Motivation
- Data
  - Data collection
  - Data pre-processing
  - Data Augmentation
- Baseline model
- Model Architecture
- YOLO Version 8 Architecture
- Model Evaluation parameter
- Implementation baseline YOLO8 nano model and results
- Implementation baseline YOLO8 large model and results
- YOLO8 based Instance Segmentation
- Dataset Preparation.
- Fine Tuning the model
- Results
- Model Training
- Summary
- Challenges

Part 1 - Detection

Goal and Motivation

- Detecting plants from images captured in the lab experiments .
- Isolating each and every plant samples which are being detected from the images.

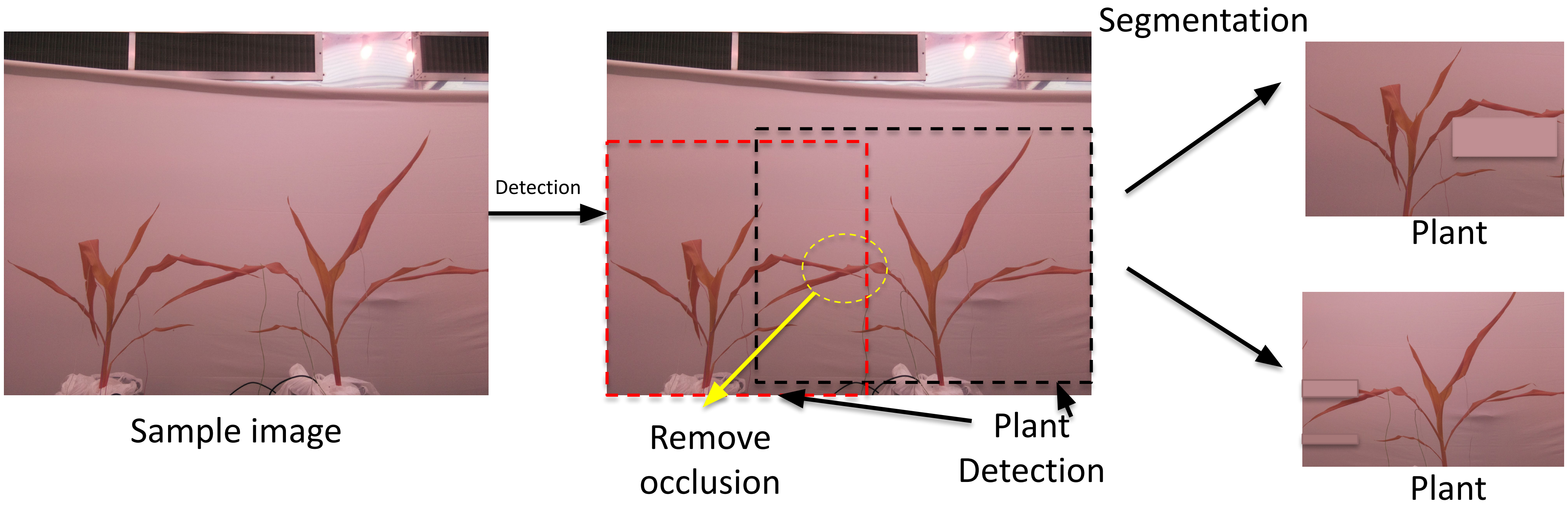


Fig.1 Overall project implementation

Data Collection

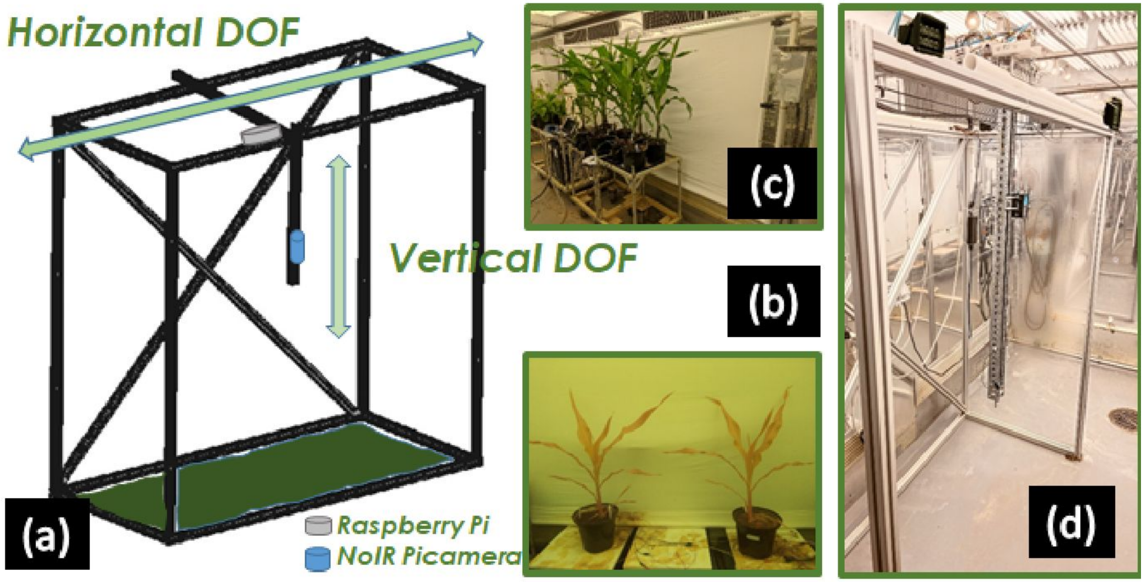


Fig.2 Data collection

(a) Schematic of the PDC showing DOF of the camera.
(b) A sample of how the maize is placed in front of PDC.
(c) Sample image.
(d) Actual picture of setup.

Source of Data samples :

1. Data samples collected from project "Feasibility Study of Water Stress Detection in Plants using a High-Throughput Low-Cost System" in ARoS Lab.
2. The images capture by Raspberry Pi Camera V2.1.
3. Data image samples use in this project – 1011.

Ref. – R. Silva, N. Starliper etc., "Feasibility Study of Water Stress Detection in Plants using a High-Throughput Low-Cost System", 2020 IEEE SENSORS.

Data Pre-processing

- The dataset created using the previous approach is pre-processed using the RoboFlow Environment.
- Around 1011 lab images samples are taken as dataset performing the plant detection.
- These images are manually annotated as per the YOLO 8 format.
  - <class> <x> <y> <width> <height>
- The images are auto oriented and resized to 640 x 640 pixels.
- This pre-processed images is split into train, validation and test set with a ratio being 87% - 9% - 4%.

Set Name	No. Images
Training Set	703
Validation Set	208
Testing Set	100

Table.1 Number of images per set

Ref. - https://roboflow.com/

Data Augmentation

- To avoid overfit and to achieve higher accuracy data augmentation of the train dataset is being performed.
- The 703 images in the train set are passed through the various augmentation listed below. The train dataset contains 2109 images after augmentation.
- Augmentation -

a) Outputs per training example: 3
b) Flip: Horizontal
c) Crop: 0% Minimum Zoom, 20% Maximum Zoom
d) Rotation: Between -15° and +15°
e) Saturation: Between -25% and +25%
f) Brightness: Between -25% and +25%
g) Bounding Box: Rotation: Between -15° and +15°
h) Bounding Box: Brightness: Between -25% and +25%

Ref. -
https://roboflow.com/

Baseline model

- An object detection model is used to detect the plants from the images.
- The You Only Look Once (YOLO) model is being used to detect the location of plants in the image.
- The lab images are passed as inputs to the model and corresponding bounding boxes of the plants are passed as annotation files.
- The model is being trained and optimized using Square Error Loss function for predicting the bounding box.

Ref. - https://pjreddie.com/darknet/yolo/

Model Architecture

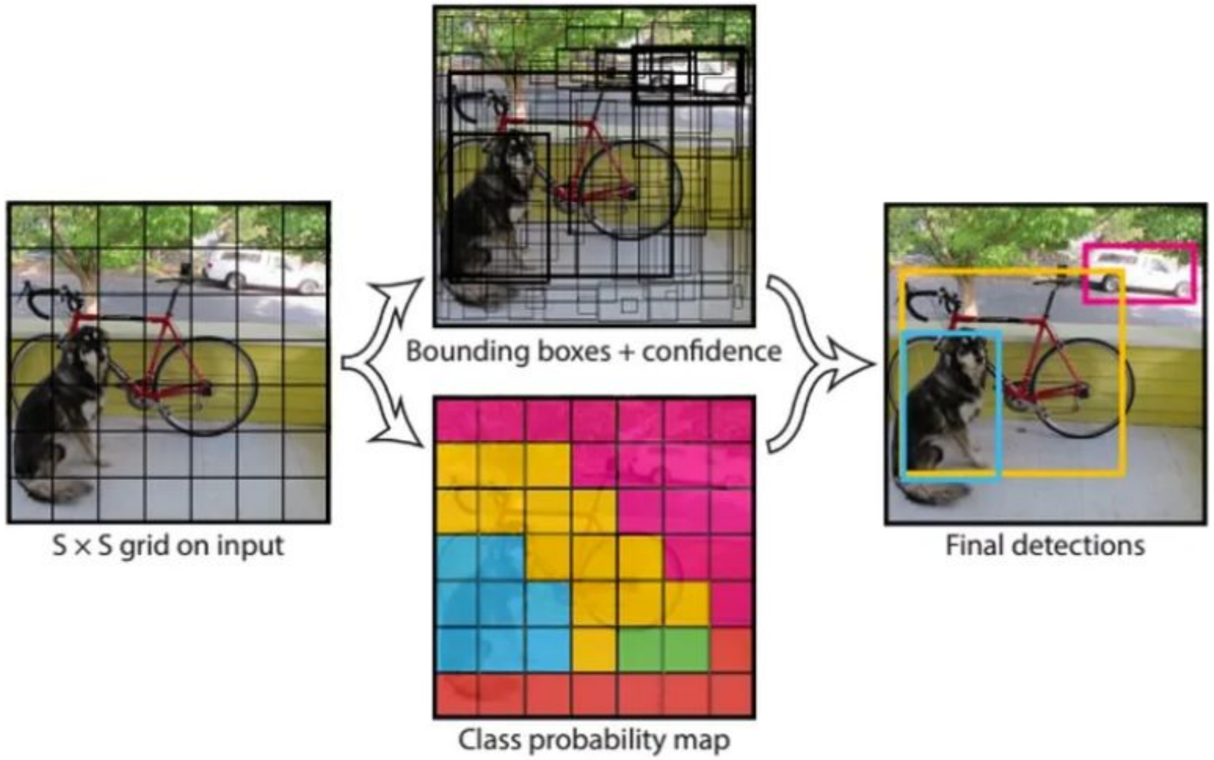


Fig.3 YOLO Model Architecture

- YOLO (You Only Look Once) uses a single pass of a neural network to perform both classification and prediction of bounding boxes for detected objects
- The input is split into an S × S grid with B bounding boxes associated with each class and a single class multinomial probability of size C, i.e. the number of classes
- There are differential weights for confidence predictions from boxes that contain object and boxes that don't contain object during training.

Ref. - https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc923000

YOLO Version 8 Architecture

- YOLO version 8 produces outcomes with higher mean average precision. The issue of prolonged training, which was a concern from version 5 to version 7, is addressed
- The trade-off between training time and precision is achieved more in version 8
- YOLOv8 provides significant developer-specific features in the form of CLI and Python package as opposed to previous versions where tasks were split across many Python files .
- Anchor Free Detection - YOLOv8 is an anchor-free model. This means it predicts directly the center of an object instead of the offset from a known anchor box.

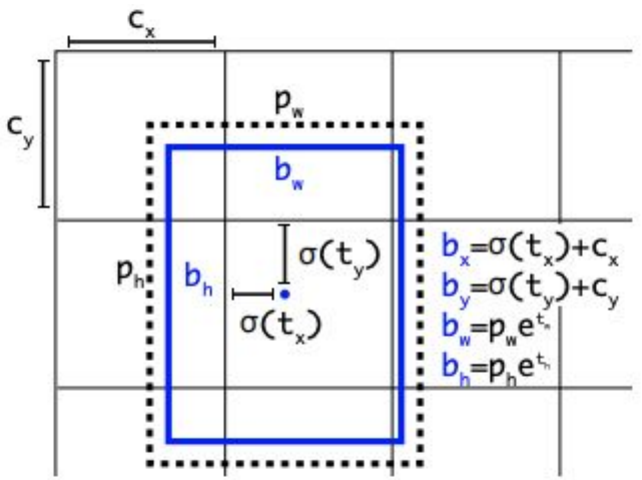


Fig.4 Visualization of an anchor box in YOLO

Ref. - https://blog.roboflow.com/whats-new-in-yolov8/

Model Evaluation

- mAP50 – It is stands for mean average precision at 50% overlap. It's a common evaluation metric used in object detection to measure the accuracy of the model in detecting objects within an image.
- A higher mAP50 indicates better accuracy in detecting and localizing objects. The mAP50 score can range from 0 to 1, where 1 indicates perfect detection and localization of objects.
- The mAP50 score is sensitive to the choice of the IoU threshold. Typically, mAP is calculated at multiple IoU thresholds, such as 0.5, 0.75, and 0.95, and reported as mAP@[IoU threshold]. The choice of the IoU threshold depends on the application and the level of precision required.
- The mAP50 score is affected by various factors, such as the size and diversity of the training dataset, the architecture and hyperparameters of the model, and the quality of the annotations.
- Improving the mAP50 score requires a combination of different approaches, such as increasing the size and diversity of the training dataset, fine-tuning the model with transfer learning, optimizing the architecture and hyperparameters of the model, and using a multi-scale approach

Ref. -
https://medium.com/@mukilankrishnakumar2002/model-building-and-model-evaluation-72fc974eeebc

Baseline YOLO 8 nano model

- The YOLO version 8 nano model is used as a baseline model for plant detection.
- It has around 3.2 Million parameters.
- The yolo8n model was trained for 200 epochs.
- The model got saturated in 180 epochs and early stopping operation was performed. The training box loss of 0.2381 and classification loss of 0.1976.

```
  Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
181/200      2.34G     0.2381     0.1976     0.8564         61        640: 100%|██████████| 132/132 [01:08<00:00,  1.94it/s]
             Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 7/7 [00:06<00:00,  1.08it/s]
               all        208        679      0.567      0.421      0.426      0.168
```

Fig.5 Results YOLO 8 nano model

Results – baseline model

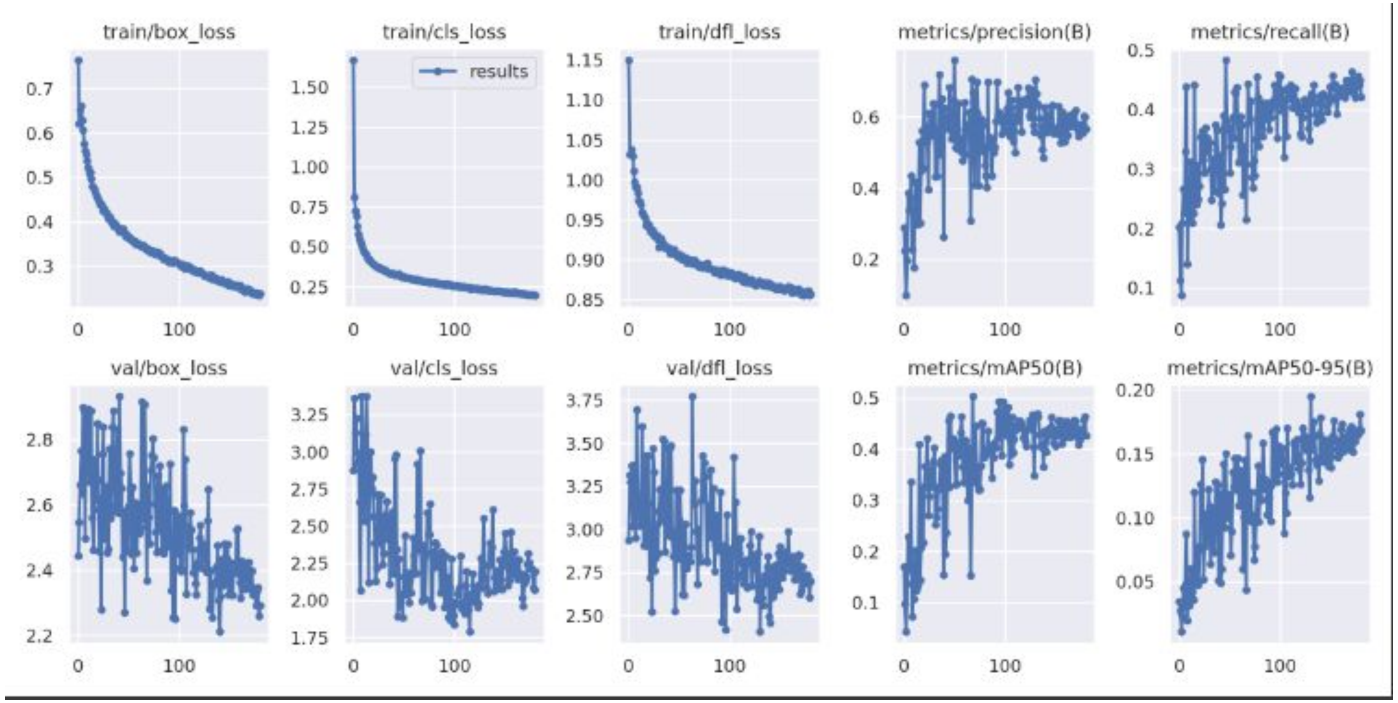


Fig.6 Training graphs YOLO 8 nano model

```
val: Scanning /content/Plant-finder-2/valid/labels.cache... 208 images, 0 backgrounds, 0 corrupt: 100%|██████████| 208/208 [00:00<?, ?it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 13/13 [00:07<00:00,  1.81it/s]
                   all        208        679      0.704      0.421      0.466      0.195
```

Fig.7 Results YOLO 8 nano model

Testing – Baseline model

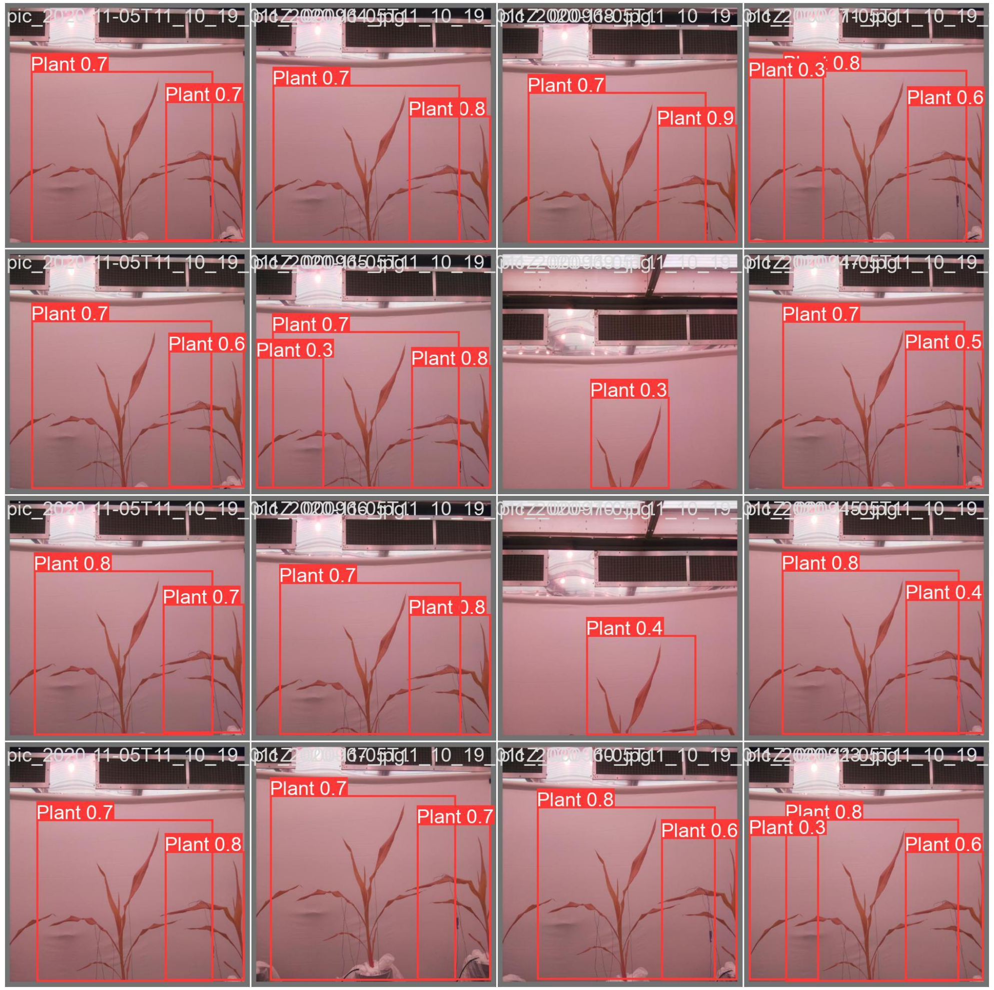


Fig.8 Detection results YOLO 8 nano model

YOLO V8 large

- To further reduce the loss and to speed up the training, the YOLO v8 large model is used.
- The model contains 43.7 Million parameters.
- The model is trained for 200 epochs.
- The model got saturated in 168 epochs and early stopping operation was performed. The training box loss of 0.1949 and classification loss of 0.1528.

```
  Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
168/200      10.7G     0.1949     0.1528     0.8522         70        640: 100%|██████████| 132/132 [01:39<00:00,  1.32it/s]
             Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 7/7 [00:09<00:00,  1.32s/it]
               all        208        679      0.628      0.401      0.419      0.136
```

Fig.8 Results YOLO 8 large model

Result – YOLO V8 large

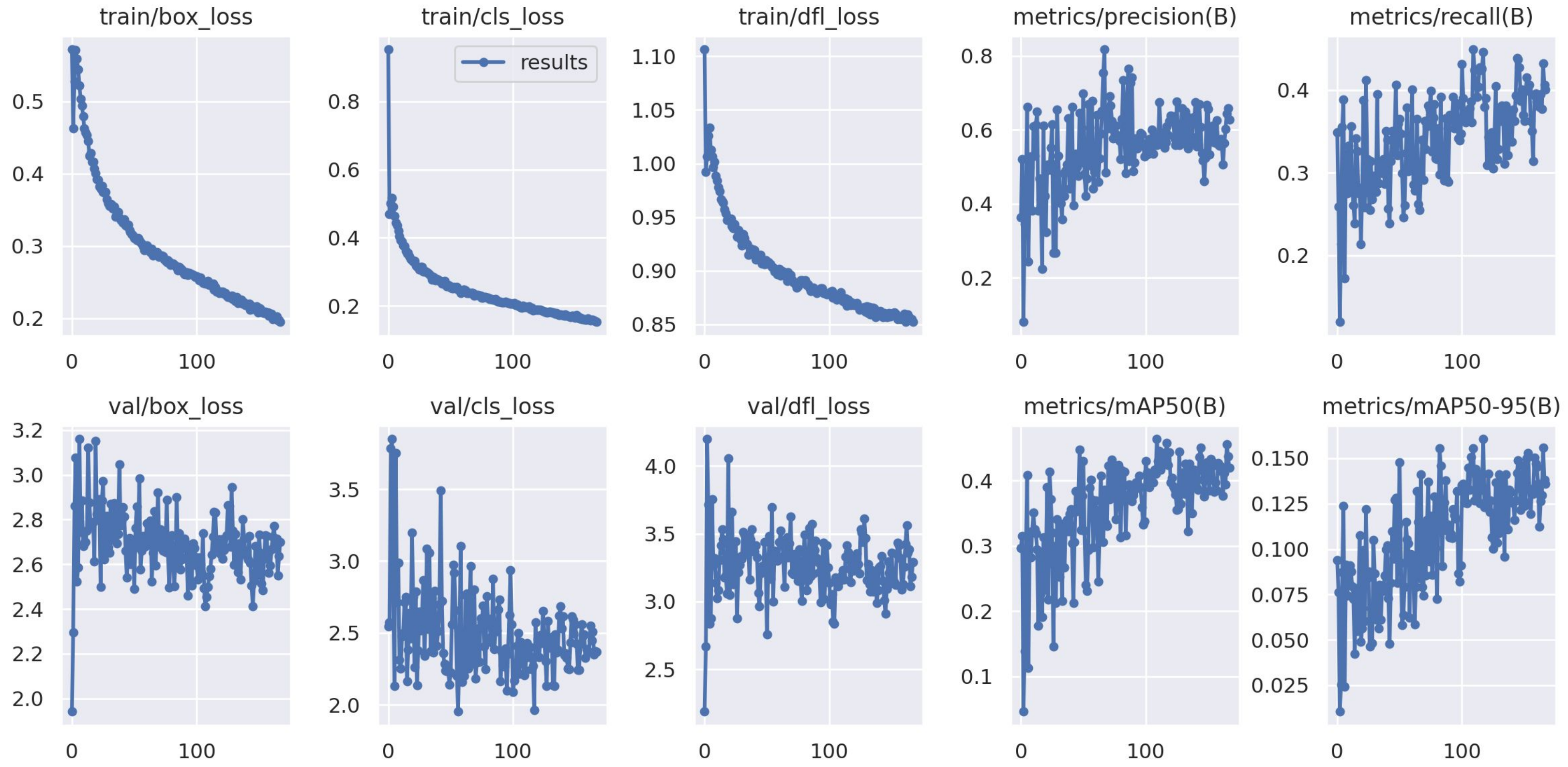


Fig.9 Training graphs YOLO 8 large model

```
val: Scanning /content/Plant-finder-2/valid/labels.cache... 208 images, 0 backgrounds, 0 corrupt: 100%|██████████| 208/208 [00:00<?, ?it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100%|██████████| 13/13 [00:13<00:00,  1.04s/it]
                   all        208        679      0.611      0.446      0.458      0.161
```

Fig.10 Results YOLO 8 large model

Testing – YOLO v8 large

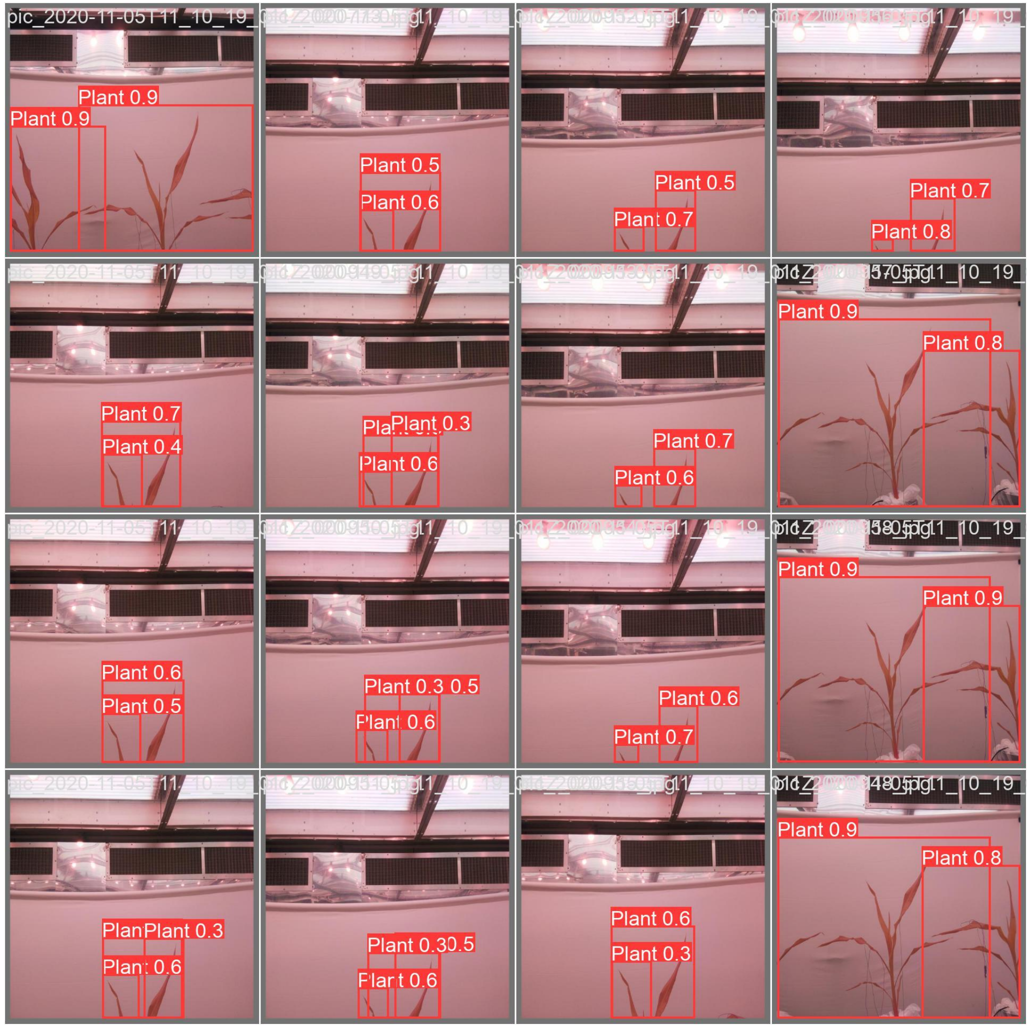


Fig.11 Detection results YOLO 8 large model

Part 2 – Segmentation and classification

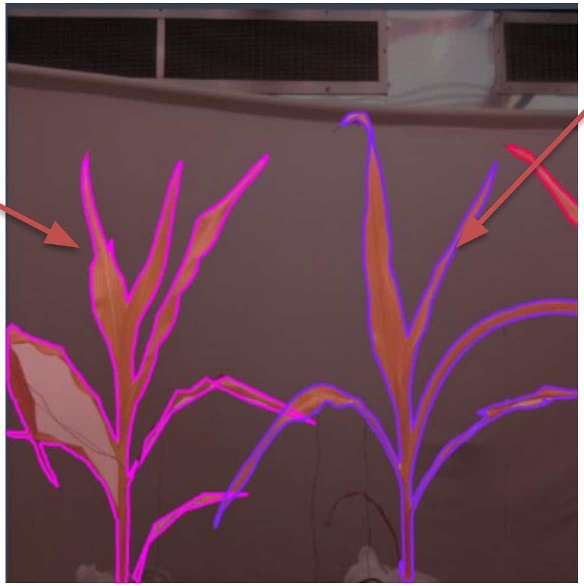
YOLO8 based Instance Segmentation

- Instance segmentation goes a step further than object detection and involves identifying individual objects in an image and segmenting them from the rest of the image.
- The output of an instance segmentation model is a set of masks or contours that outline each object in the image, along with class labels and confidence scores for each object.

Dataset Preparation

- Roboflow tool is used to prepare the dataset.
- The plants are outlined by using the smart polygon feature and labelled .

Plant 1



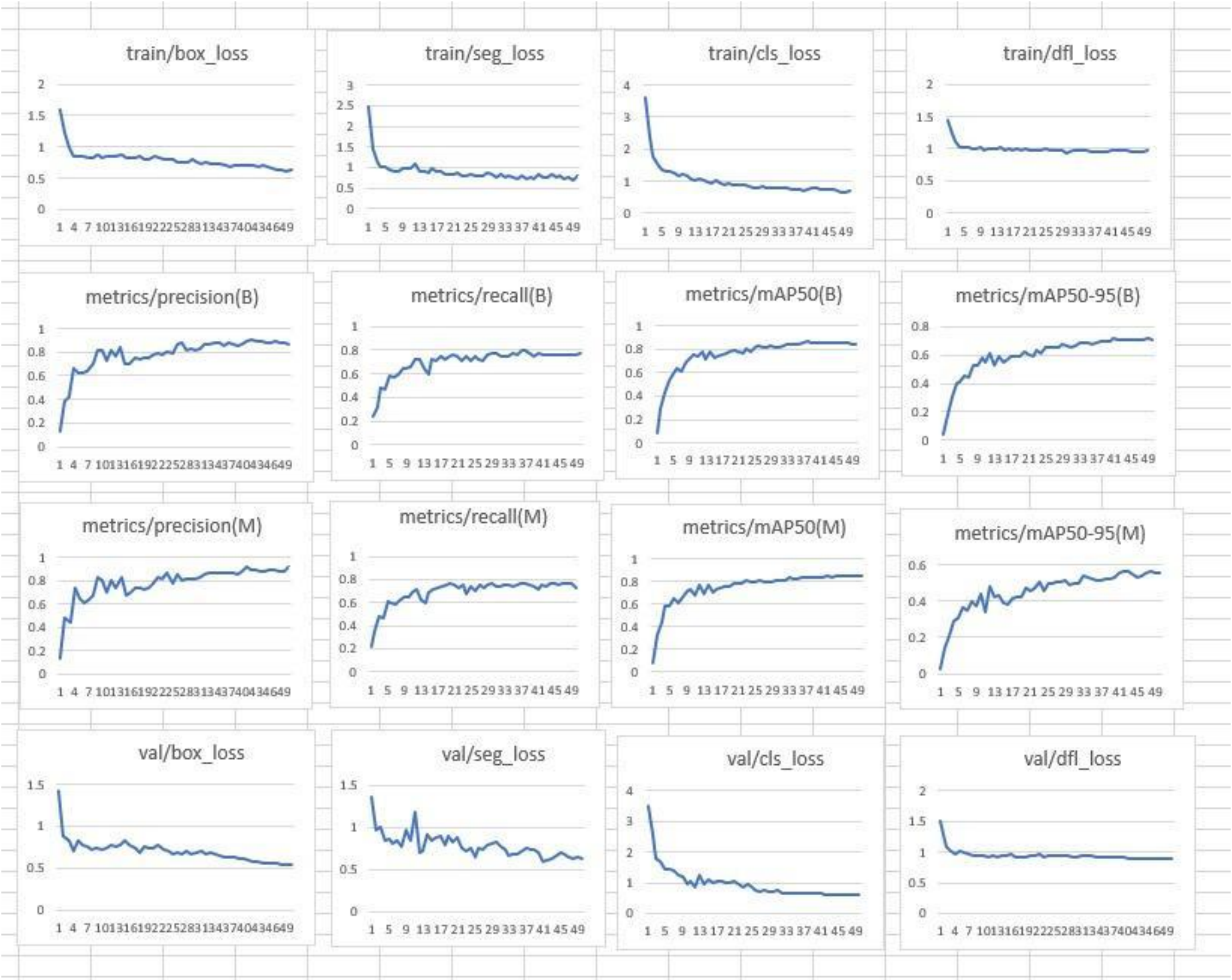
Plant 2

A total of 500 images have been annotated for image segmentation. The images have been split between train, validation and test set as 380, 80 and 40 respectively.

Baseline approach - Fine Tuning the Model

- The YOLO8 small model is loaded with pretrained weights and the weights first 10 layers frozen for training.
- The model is trained for 50 epochs.
- A box loss of 0.8 and segmentation loss of 0.7 is being obtained.

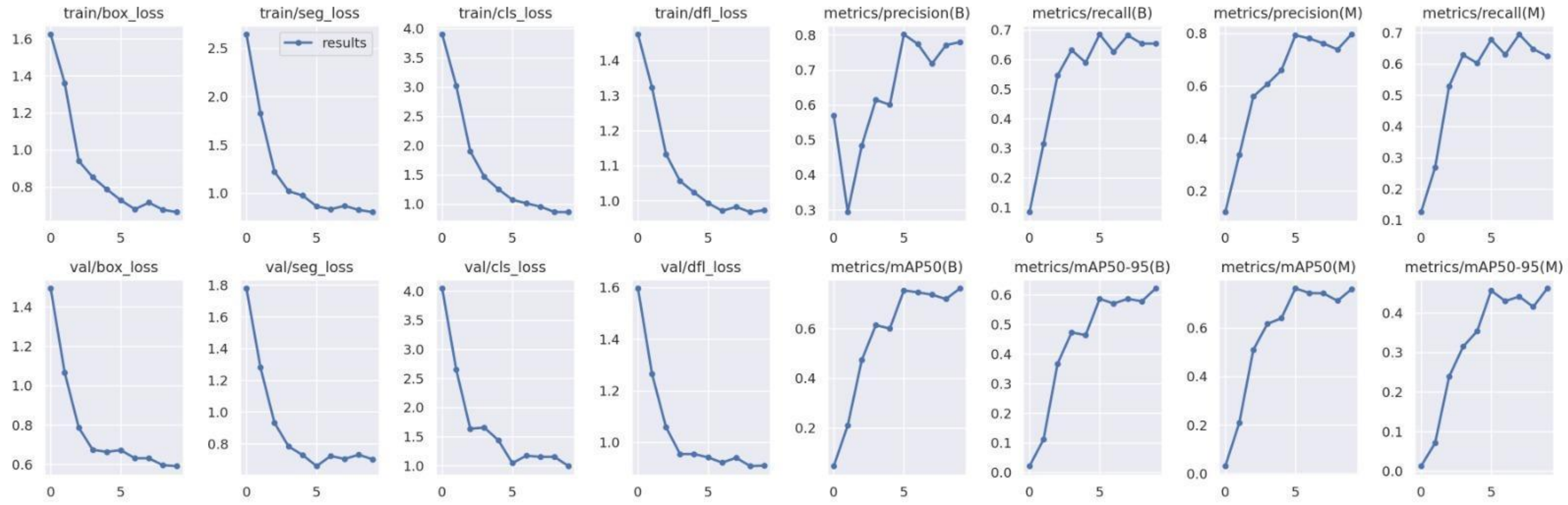
Results


train/box_loss
train/seg_loss
train/cls_loss
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
metrics/mAP50(B)
metrics/mAP50-95(B)
metrics/precision(M)
metrics/recall(M)
metrics/mAP50(M)
metrics/mAP50-95(M)
val/box_loss
val/seg_loss
val/cls_loss
val/dfl_loss

Model Training

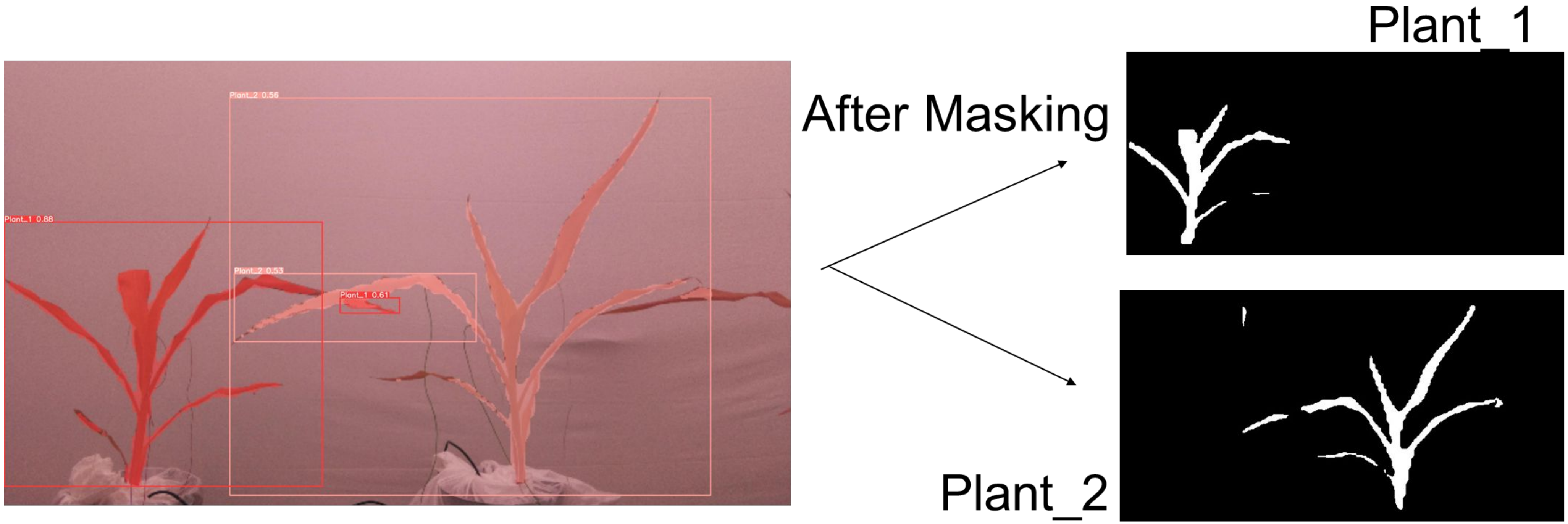
- Yolo8 small model is used for segmentation and classification.
- The model is fully trained.
- The Mean square loss function is used as the cost for bounding box and dice loss function is used as a cost for segmentation.
- A box loss of 0.568 and segmentation loss of 0.432 is being achieved which is better than fine tuning approach.
- The model is trained for 50 epochs and a mAP50 of 0.861 is achieved in train set and 0.827 is achieved in validation set.

Results


train/box_loss
train/seg_loss
results
train/cls_loss
train/dfl_loss
metrics/precision(B)
metrics/recall(B)
metrics/precision(M)
metrics/recall(M)
val/box_loss
val/seg_loss
val/cls_loss
val/dfl_loss
metrics/mAP50(B)
metrics/mAP50-95(B)
metrics/mAP50(M)
metrics/mAP50-95(M)

Results – contd.

- The model first performs the detection of plants in the image and then performs segmentation of each plant.



Summary

- **Detection**
  - The YOLO v8 nano model takes more epochs to train and the final bounding box loss value saturates at 0.2381.
  - The YOLO v8 large model takes lesser number epochs to train and the final bounding box value saturates at 0.1949.
  - The Mean Average Precision (mAP) at 50% overlap scores of both the models are nearly the same of around 0.466.
- **Detection and Segmentation**
  - The YOLO v8 small model is being fine tuned by freezing the first 10 layers and the box loss values saturates at 0.8 in 50 epochs.
  - On training the whole model, the box loss value was being reduced to 0.568 and mAP50 of 0.861 is achieved.

Challenges

- Illumination: YOLOv8's performance can be negatively affected by changes in lighting conditions, especially in low-light environments
- Limited dataset: The performance of YOLOv8 heavily depends on the quality and size of the training dataset. Insufficient or biased training data can result in poor object detection performance.
- Accuracy vs. speed trade-off: YOLOv8 is optimized for speed, which can come at the cost of accuracy. If high accuracy is required, slower but more accurate models may be more appropriate.
- Data Annotation : The annotation for detection and segmentation takes a long time and the plants should be outlined meticulously to get better results.

Thank You!