Innovation Hub

5G Agriculture Project

AI Section Documentation

Artificial Intelligence Team

Abdelrahman Shalabi

Mohammad Al-Jadallah

Ala’a Abdelqader

Layan Al-Hyasat

1) Working process

- Dataset selection

  After searching for a good dataset with big images included and with poor resources for the dataset, we collected a dataset includes more than 3000 images to train and validate the model on it. The dataset contains the images and their annotation files (which are the coordinates of the locust in the image).

- Model selection

  According to the team researches, we decided to choose the YOLO v8 models to train the model, according to its powerful performance on image processing and object detection. We fine-tuned the model on our dataset to detect the Locust object. The technique was to train and test all the yolov8 models (yolov8_s, yolov8_m, yolov8_l, yolov8_x) and choose the best model that perform well on our dataset. Each models from the mentioned before have different architecture such as the network size, so the goal is to choose the best network that is compatible with the dataset size and the features wanted to be detect in the image.

- Training history

  The models are trained with 200 epochs, but some models stopped before the end of the 200 epochs (according to the Early Stopping technique, which stops the training when there is no improvement on the model performance.

- Models comparing

  After training all models on our custom dataset and testing them on real life examples we decided to choose the yolov8_m model which performed well according to the dataset size and features in the images,
  Which be reasonable option according to the model infrastructure (size of the neural network and its training parameters)

- Yolov8 working technique

  The yolov8 architecture saves the model after in two files, the best model which have the best performance across the training process, and the last epoch trained which have the last performance of the model.

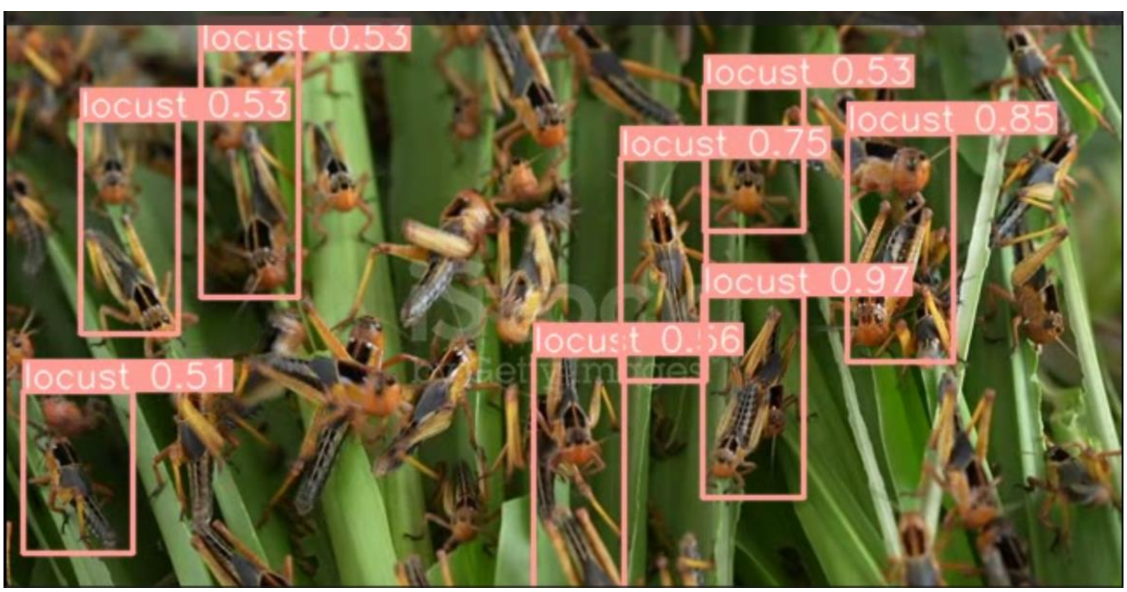
2) Real Time Model Performance

- Video testing

  In the following section the performance on a video to see how the model detects the locust (Please download the video to get the full resolution):

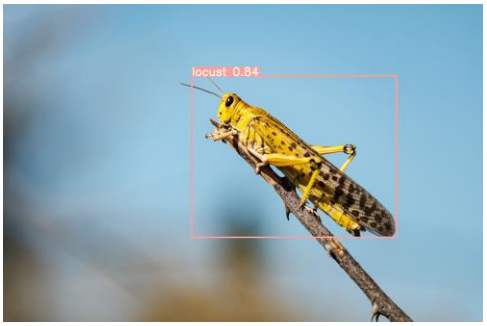
[Video Detection]

Image from the video:



- Image testing

To make sure the model detects correctly, we tested images for non-locust objects to see if there is misclassification or not, and fortunately the model classified correctly on the most images, here is some examples:



3) Model Evaluation

- Losses

  After training the model, it's time to plot and explain the measures which includes the various losses measures and the accuracy measures including the precision and recalls.

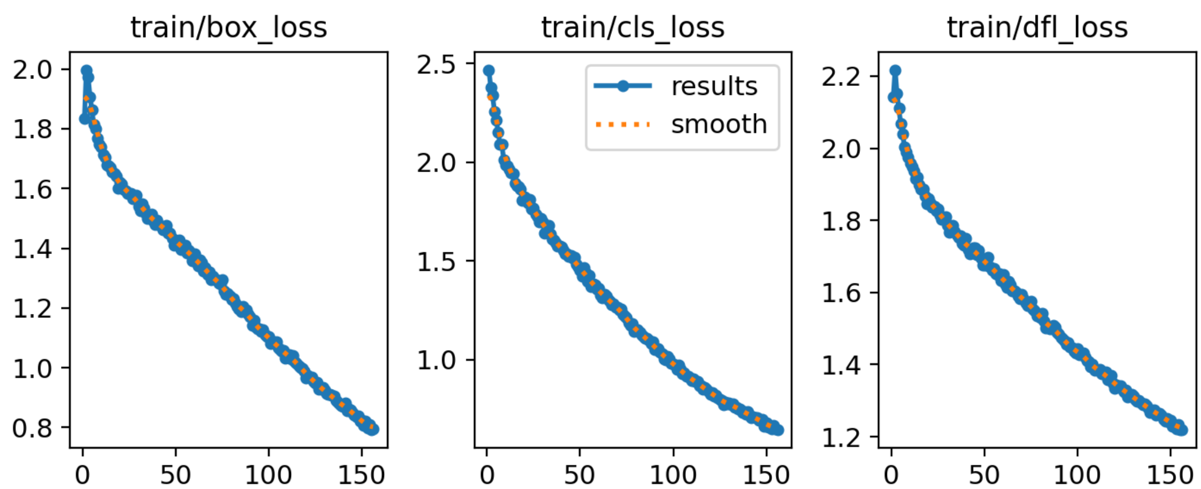


Figure (1)

- **Box loss** is a metric used to measure how well the predicted bounding boxes around objects in an image align with the ground truth, which are the actual bounding boxes. A lower box loss indicates that the model is better at predicting bounding boxes that match the ground truth. In this graph it is generally decreasing over the course of training epochs. This suggests that the model is improving its ability to predict bounding boxes around objects in the images.

- The **train/cls_loss** visualization aims to illustrate how the loss associated with object classification evolves during the training process of an object detection model. This loss reflects the disparity between the predicted class probabilities and the ground truth class labels for objects in the training dataset.
  In the graph, the train/cls_loss line appears to be generally decreasing over the course of training epochs. This suggests that the model is improving its ability to classify between the different classes in the training data.

- The graph demonstrates the model's learning process. With each training epoch, the model gets better at predicting the size and location (bounding boxes) of objects within images. This is achieved by the model minimizing the **DFL** (Distribution focal loss), a measure of how far off its predictions are from the actual bounding boxes.

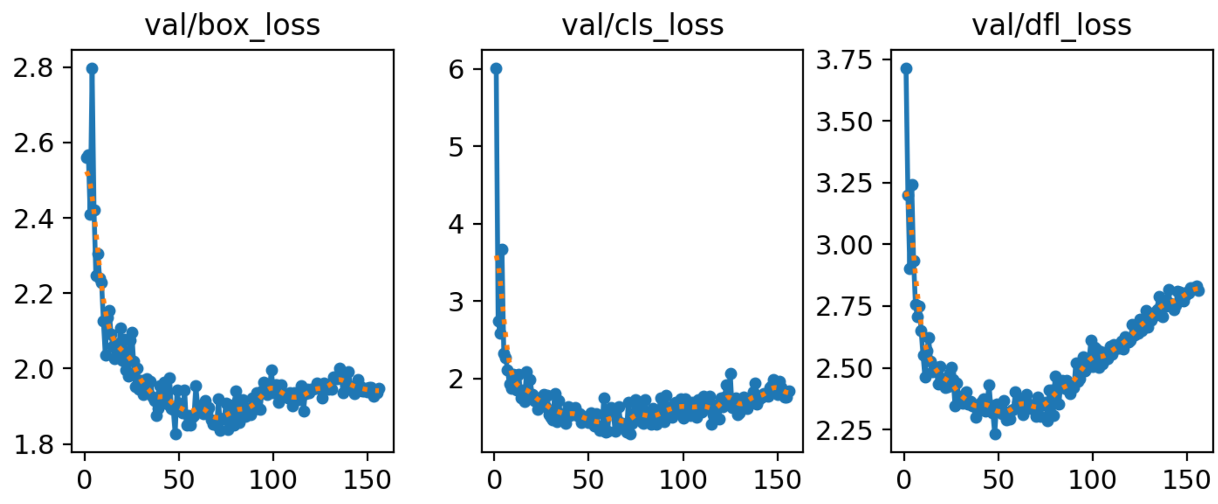


Figure (2)

- The **Val box_loss** refers to the validation box loss, which is a metric used to evaluate the performance of an object detection algorithm during the validation phase. It assesses how accurately the algorithm can locate the center of an object and how well the predicted bounding box covers the object in the validation dataset. Effectively, the validation box loss indicates the algorithm's ability to identify objects within a given region of interest. A low validation box loss suggests that the algorithm is proficient at accurately localizing objects and generating bounding boxes that encompass them well. During the validation phase, the goal is for the validation box loss to decrease over time, indicating that the algorithm is improving in its ability to accurately locate and delineate objects within images. However, it's important to note that there may be a point where the validation box loss starts to saturate, meaning it reaches a plateau where further improvement is minimal. This behavior is considered normal and is expected during the training and validation of object detection algorithms.

- The **Val cls_loss** refers to the validation classification loss, which is another metric used to evaluate the performance of an object detection algorithm during the validation phase. Unlike the validation box loss, which focuses on the accuracy of bounding box predictions, the validation classification loss assesses how well the algorithm can correctly classify objects within the bounding boxes. Effectively, the validation classification loss measures the algorithm's ability to predict the correct class or category of objects present within the detected regions of interest. This loss helps determine the algorithm's proficiency in distinguishing between different object classes, such as identifying whether an object is a car, a person, or a tree, among other possibilities. It's important for both the validation box loss and the validation classification loss to decrease during the training and validation process, as this signifies that the object detection algorithm is improving its overall performance in accurately localizing and classifying objects within images.

- The graph demonstrates the model's learning journey in predicting object bounding boxes accurately. It represents the model's performance on unseen validation data. A decreasing or stable **Val dfl_loss** indicates the model is generalizing its learning from training data to new data.

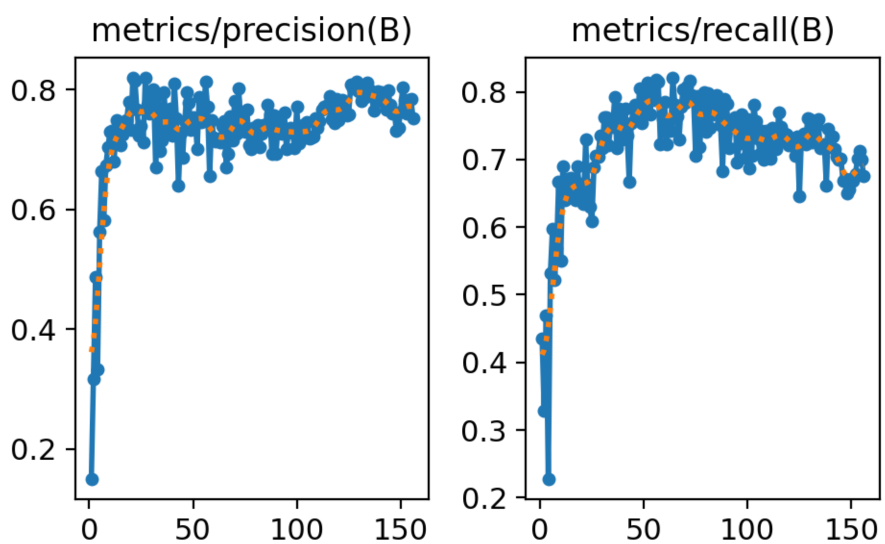


Figure (3)

- The **metrics/precision** visualization aims to illustrate the progression of precision within the detection model throughout the training phase. Precision is a critical metric in object detection, representing the proportion of correctly identified positive instances (true positives) among all instances classified as positive. In the context of information retrieval, precision measures the fraction of relevant items among those retrieved. In the graph, higher values of precision correspond to larger numbers of epochs. This indicates that as the model undergoes more training, precision increases. This improvement signifies the model's ability to better distinguish true positives from false positives. The overarching objective is to ensure the model achieves accurate and reliable detections while minimizing false positives.

- The **metrics/recall** visualization illustrates the evolution of recall within the detection model throughout training. Recall, also referred to as sensitivity or true positive rate, quantifies the proportion of actual positive instances (true positives) correctly identified by the model. It serves as a vital metric in machine learning, reflecting the fraction of relevant instances retrieved. In the initial training stages depicted in the graph, recall experiences rapid growth as the model learns to detect more positive instances. This swift ascent can be attributed to the model's early acquisition of general patterns and features associated with the objects it's trained to detect. Consequently, the recall metric escalates swiftly, signifying the model's adeptness in capturing a greater proportion of true positive instances.

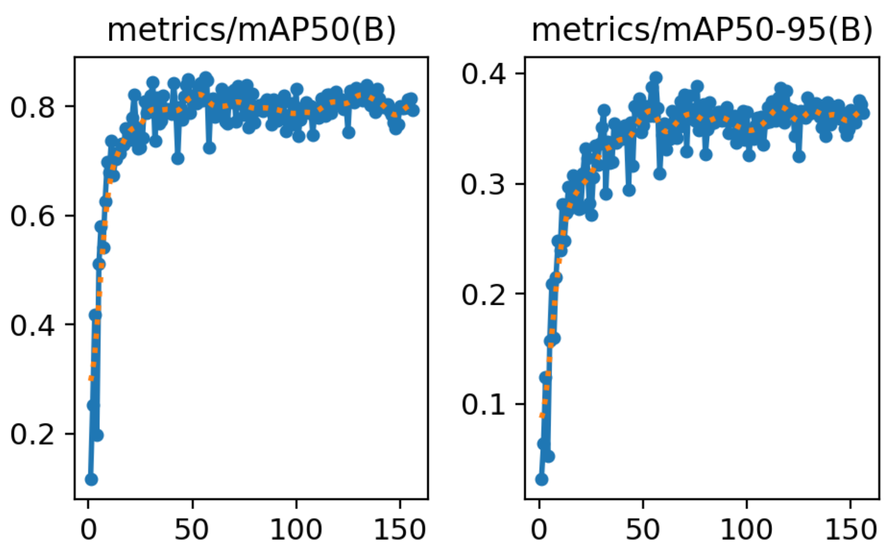


Figure (4)

- The **metrics/mAP500** graph illustrates the Mean Average Precision (mAP) at a Box Average Precision (B) of 500, a common metric used in object detection to evaluate model performance on a dataset. mAP considers both precision and recall, where precision measures the relevance of retrieved items in information retrieval, and recall measures the fraction of relevant instances retrieved in machine learning. In this context, higher mAP values indicate better performance in the object detection task. For each detected class, Average Precision (AP) calculates the area under the precision-recall curve, reflecting how the model's precision changes with varying recall thresholds.

- The label '50-95' in the **metrics/mAP50-95** refers to the Intersection over Union (IoU) threshold utilized for evaluating object detection. IoU measures the degree of overlap between the bounding box predicted by the model and the ground truth bounding box. This visualization represents the Mean Average Precision (mAP) computed across a spectrum of IoU thresholds, typically ranging from 0.5 to 0.95, in increments of 0.05. For each detected class, Average Precision (AP) calculates the area under the precision-recall curve, indicating how the model's precision varies with changes in the recall threshold. An increasing trend in this graph suggests that the model is improving in its ability to accurately localize objects within images across various levels of overlap between predicted and ground truth bounding boxes. This signifies that the model's predictions increasingly match the actual positions and shapes of objects in the images.

4) Summary

- **Model Development:** Dataset collection involved gathering over 3000 images for training YOLO v8 models, fine-tuning them to detect locusts based on performance criteria such as dataset compatibility and feature detection.
- **Real-time Performance:** The models were tested on both videos and images to assess their accuracy in locust detection, with satisfactory results in identifying non-locust objects as well.
- **Model Evaluation:** Various loss measures and accuracy metrics were analyzed, including box loss, classification loss, and distribution focal loss, demonstrating the model's learning process and proficiency in object detection over training epochs.
- **Performance Metrics:** Metrics like precision, recall, Mean Average Precision (mAP), and Intersection over Union (IoU) were employed to evaluate the model's performance, indicating improvements in accurately localizing and classifying objects within images.