トマトの検出と収量予測

2024/8/09

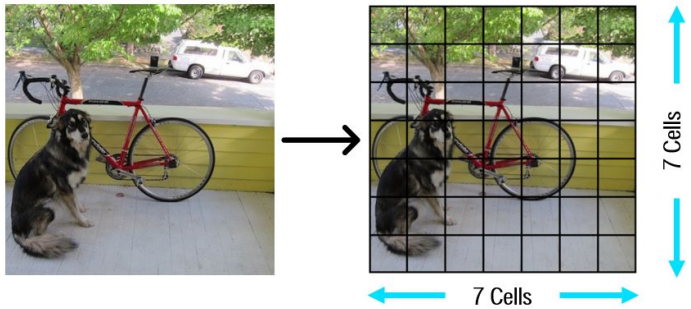
Identifying tomatoes and determining ripeness

Identifying tomatoes and determining ripeness

- **First Subgoal** - create a base model to detect the tomato fruits
- **Second Subgoal** - Image pre-processing and fine-tuning of hyperparameters
- **Third Subgoal** - Count the tomatoes from video material
- …
- **Final Goal** - Determine ripeness

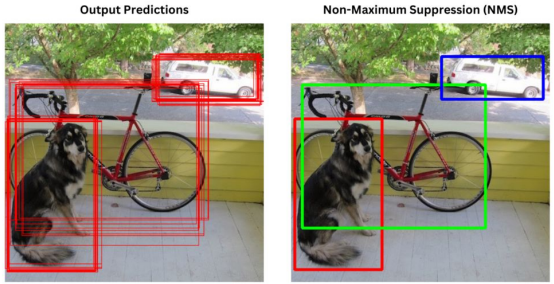
YOLO - You only look once

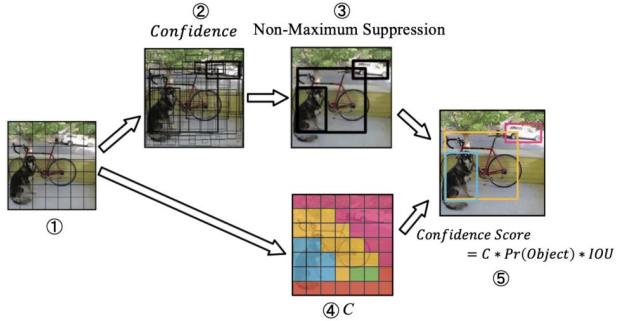
- **Single Pass Detection** - YOLO detects objects in one pass, unlike traditional methods that use region proposals followed by classification.
- speeding up the process and enabling **real-time object detection**.
- **Grid Division** - The input image is divided into an S×S grid, with each cell responsible for detecting objects whose centers fall within the cell.



YOLO - You only look once

- **Bounding Box Prediction** - Each grid cell predicts bounding boxes, including coordinates (center x, center y, width w, height h), and a confidence score for each box.
- **Class Probability** - Each grid cell also predicts the probability of the object class within that cell.
- **Combined Output** - Final predictions combine class probabilities and bounding box confidence scores, with Non-Maximum Suppression (NMS) removing duplicates.





Segmentation

- **Pixel-Level Classification** - Segmentation involves classifying each pixel in an image into predefined categories, providing detailed object boundaries.
- **instance segmentation -** allowing it to not only detect objects but also accurately segment them from the background
- **Use in my Application** - object recognition of the tomates for more precise localization than bounding boxes.
- **Output Format** - The result is a mask over the image, where each pixel is labeled with its corresponding class, highlighting the specific regions of interest.

YOLOv8-seg

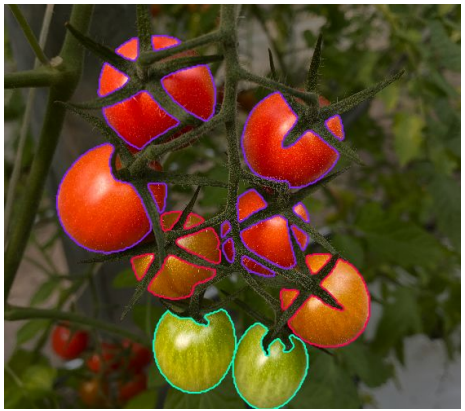
- Supports **multiple tasks**, including object detection and instance segmentation making it a flexible choice for diverse computer vision applications.
- **Combined Detection and Segmentation Heads** - Separate heads for bounding boxes and segmentation masks to ensure precise instance segmentation.
- **Improved Training Techniques** - Utilizes advanced data augmentation, regularization, and loss functions are used to optimize both detection and segmentation tasks.

YOLOv8

Model	size (pixels)	mAP^{box} 50-95	mAP^{mask} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n-seg	640	36.7	30.5	96.1	1.21	3.4	12.6
YOLOv8s-seg	640	44.6	36.8	155.7	1.47	11.8	42.6
YOLOv8m-seg	640	49.9	40.8	317.0	2.18	27.3	110.2
YOLOv8l-seg	640	52.3	42.6	572.4	2.79	46.0	220.5
YOLOv8x-seg	640	53.4	43.4	712.1	4.02	71.8	344.1

Dataset - annotated Images

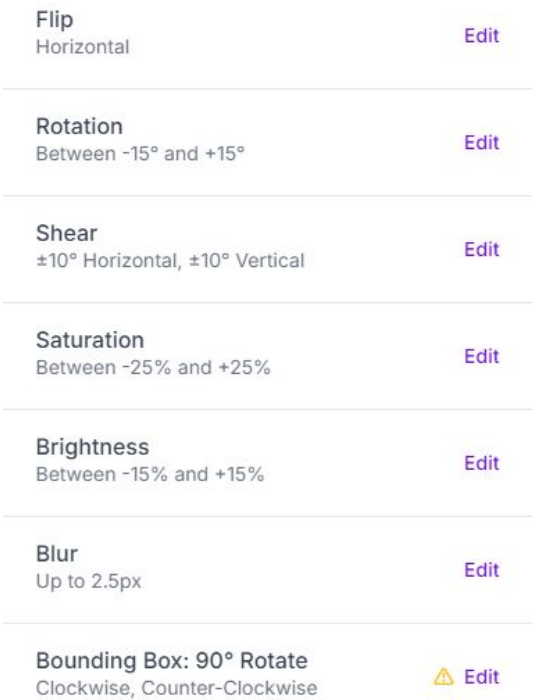
- Expanded from 80 images to 240 images through augmentation techniques.
- Tomatoes are classified into four degrees of ripeness.
- Only tomatoes in the main area are labeled to ensure accurate detection.
- Small dataset used for faster training.



Augmentation and Regularization

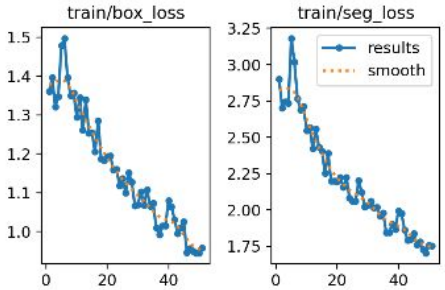
- **Dropout**: "drops out" neurons during training to prevent overfitting -> redundancy
- **Weight Decay**: Punishment for large weights in the model to prevent overfitting
- **Batch Normalization**: Normalizes layer inputs during training, speeding up convergence and stabilizing learning.

- **Data Augmentation**: Enhances training by creating varied versions of images



Training and Choice of Hyperparameters

- **Image Size (imgsz=640)** - Input images are resized to 640x640 pixels for consistent model processing
- **Batch Size (batch=8)** - Model trained using batches of 8 images at a time
- **Adam Optimizer** - YoloV8 adapts the learning rate during training with Adam, starting at **lr0=0.0001**
- **Training Configuration** - Trained over 100 epochs with early
- stopping after 15 epochs of no improvement (patience of 15)

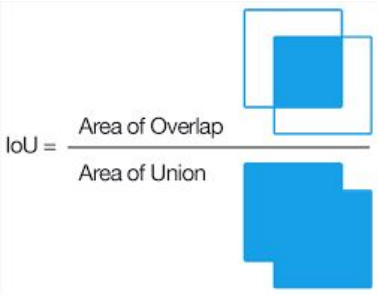


metrics for evaluating the model

- **mAP (mean Average Precision):** Measures the average precision across all classes
- **IoU (Intersection over Union):** measuring accuracy of the bounding boxes and segmentation.
- **Precision:** proportion of true positive predictions out of all positive predictions made
- **Recall:** Reflects the proportion of true positives identified out of all actual positives, showing how well the model detects objects.

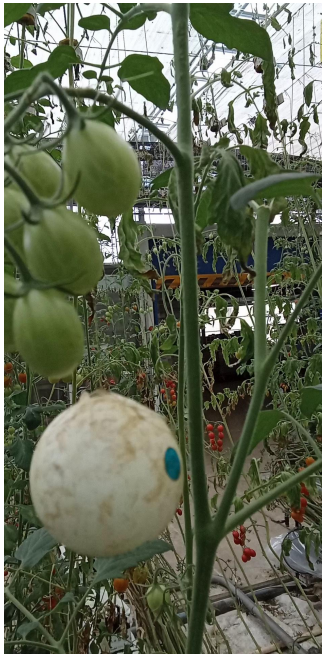
inference step

- **Inference** - The model performs object detection on each frame of the video
- **IoU (Intersection over Union) Threshold:** Determines the minimum overlap required between the predicted and ground truth bounding boxes
- **Confidence Score Threshold:** Sets the minimum confidence level for the model to consider a prediction as valid



YOLOv8-seg

raw image:



pretrained:

trained:

Count the tomatoes from video material

- **tracking algorithm -** Implemented SORT (Simple Online and Realtime Tracking) to count detected objects.
- **Initialization:** Start tracking when objects are detected.
- **Prediction:** Use Kalman filters to predict the next position of each object.
- **Update:** Associate new detections with existing tracks using the Hungarian algorithm.
- **Counting:** Maintain a count of objects based on their tracked IDs.

Tracking Algorithm - SORT

- **Max Age** - Maximum number of frames a track can be inactive before it is considered lost.
- **Min Hits** - Minimum number of consecutive frames a detection must be present to initiate a new track.
- **IoU** - Used to determine if a detection is considered a match

Final Goal

- **Final Goal** - Utilize regression analysis to predict the remaining time until tomato ripeness
- **Addressing Inaccuracies** - additional features to improve prediction accuracy
- **Training data structure** - color, size and other relevant attributes
- **Output** will be the estimated remaining time until the ripeness