

# Smart Plant Disease Detector for (Tomato)

Team member: Mohanad Yassin

Project Tier: Tier 1



## **What problem are we solving?**

Farmers and small holders lose yield when tomato diseases go undetected. Manual inspection is time consuming, error, and often delayed.

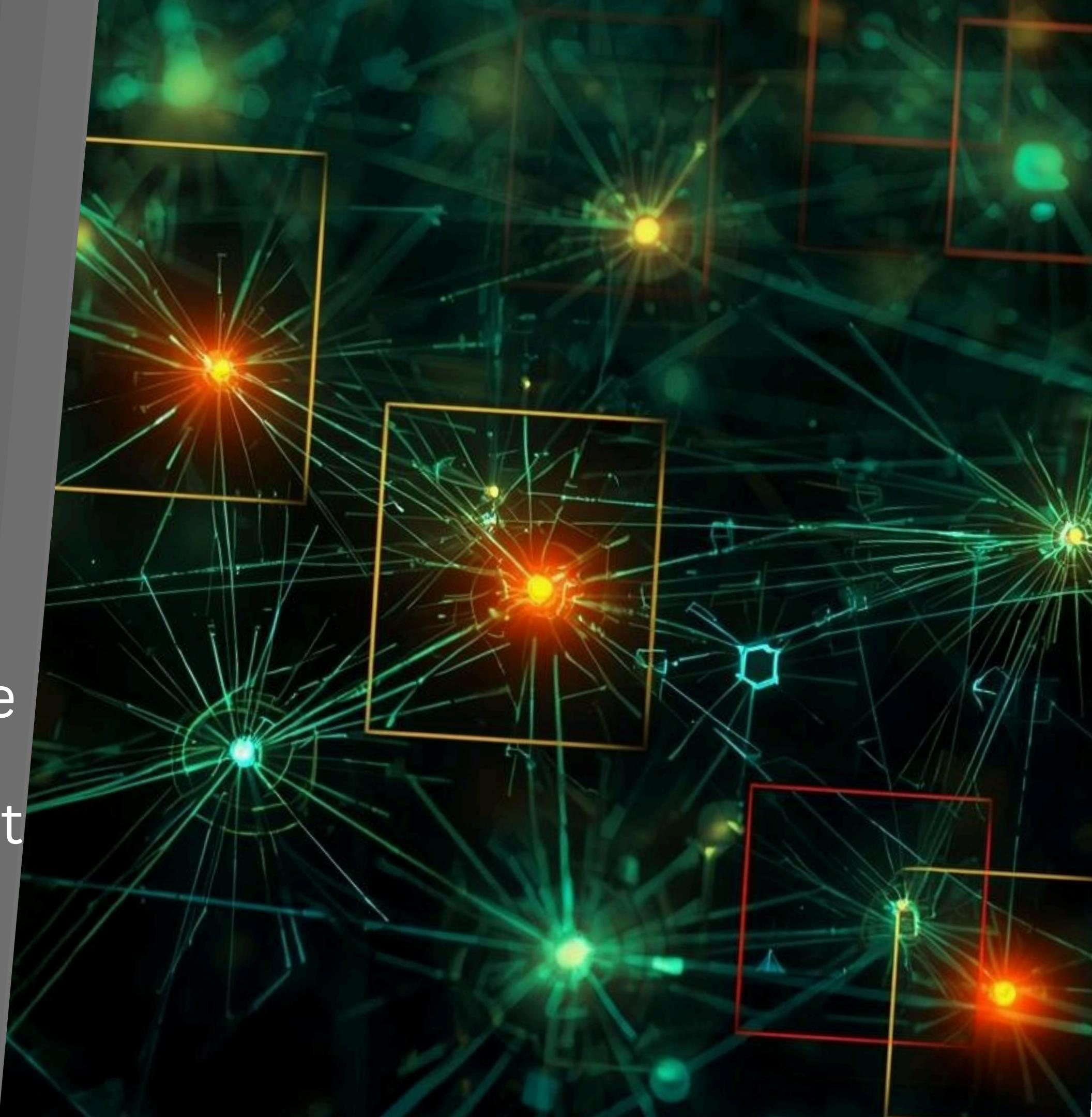
**Who cares?** Small to medium farmers, extension agents, agricultural cooperatives, agricultural startups.

**Why It's important?** Early detection reduces crop loss and improves yields and income, and supports faster interventions.



**One-sentence system description:** A YOLO-based detector that analyzes photos of tomato leaves to locate and classify common diseases and provide short treatment suggestions.

**How it works (overview):**  
Camera/photo → YOLO model  
detects leaf regions and disease class → UI shows disease name,  
confidence, and short treatment tip.



**Computer vision technique:**  
Object detection (single-stage  
YOLO) + light post processing for  
suggestions.

Model: YOLOv8 (fine tune on plant  
disease detection).

Ultralytics Docs Framework:  
PyTorch / Ultralytics YOLO package  
(for training & inference).

**Why YOLO?** Fast, accurate in real  
time, easy to deploy on CPU/GPU  
and mobile via conversion  
(ONNX/TFLite/CoreML). Good  
trade-off for in field apps.

AT DI-inince



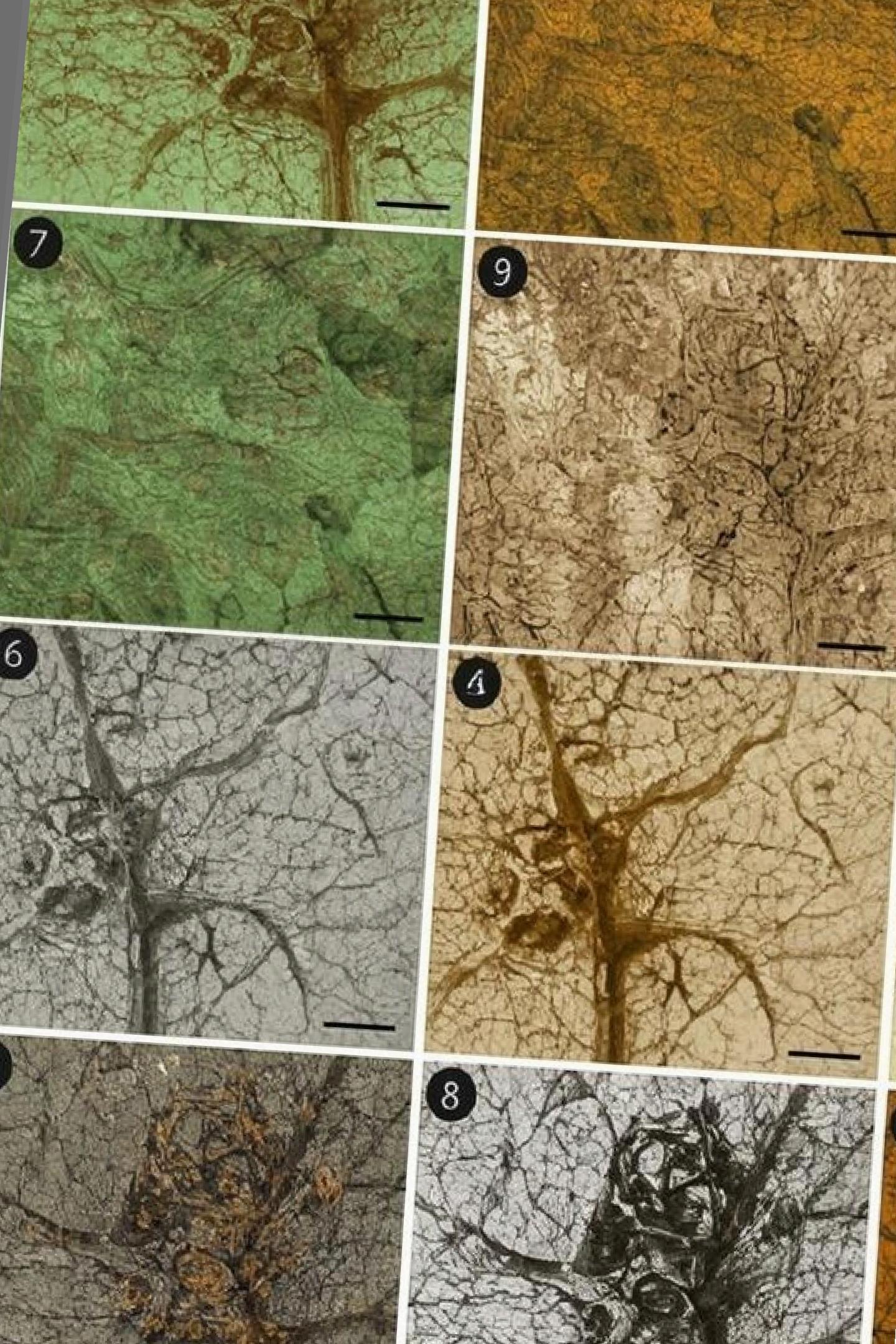
## Sources:

**Public:** PlantVillage tomato classes (Kaggle / TensorFlow Datasets / Roboflow variants). 10–30k images available in total across splits. Kaggle+TensorFlow

**Size (approx):** 10k–20k labeled images , augmented to increase variability. (PlantVillage contains many tomato leaf images separated by disease classes.)

**Labels:** disease class ( Early blight, Late blight, Healthy) and bounding boxes (for detection). Use Roboflow/KF tools to convert classification datasets to detection labels where needed.

**Preparation:** clean/normalize, train/val/test split, bounding-box labeling.



**Simple flow:** [Phone/Camera Image] → [Preprocess: resize, normalize] → [YOLOv8 inference (detect + classify)] → [Post-

**process:** NMS, map class → suggestion] → [UI / Report: disease name, confidence, treatment tip]

**Components labeled:** images (dataset), model inference (YOLOv8), visualization ( notebook demo).

