Autonomous Leaf Grasping: A Hybrid CV-ML Approach

Project Overview:

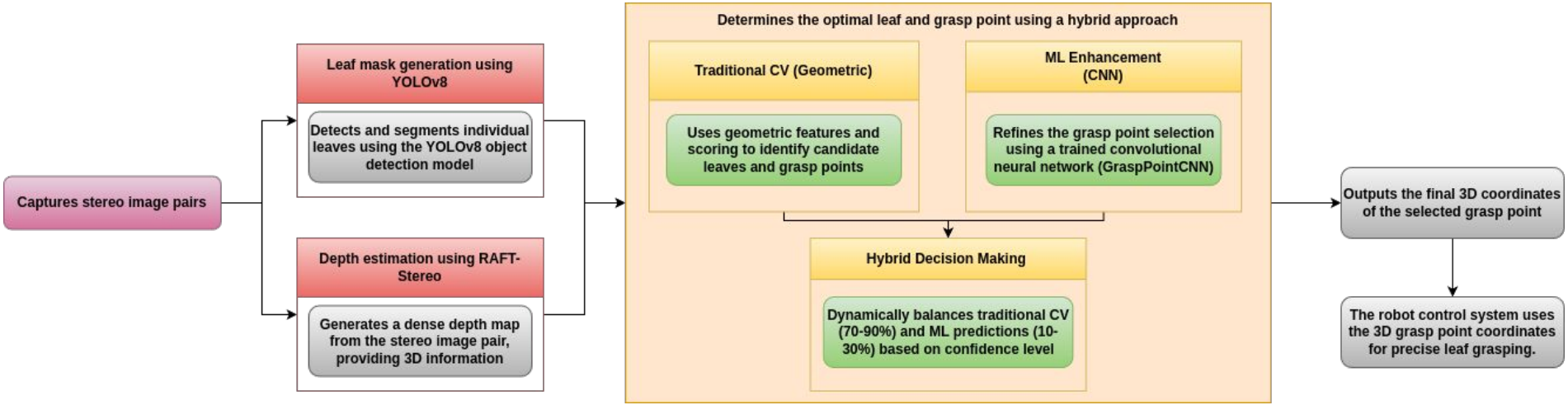
A real-time vision system for autonomous robotic leaf manipulation combining geometric computer vision techniques with deep learning. This hybrid system integrates YOLOv8 for leaf segmentation, RAFT-Stereo for depth estimation, and a custom CNN (GraspPointCNN) for grasp point optimization. The architecture features self-supervised learning that eliminates manual annotation, and a confidence-weighted decision framework that dynamically balances traditional CV algorithms with CNN predictions to achieve superior grasping performance.

GitHub Repository: LeafGrasp-Vision-ML

Key Technologies and Skills Used:

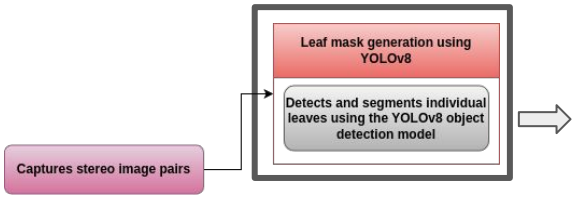
Languages: Python, C++
Frameworks: PyTorch, CUDA, OpenCV, Scikit-learn, Numpy, Pandas, Matplotlib, ROS2, MLflow
Computer Vision: Instance Segmentation, Depth Estimation, Point Cloud Processing, SDF, 3D Perception
Deep Learning: CNN Architecture Design, Self-Supervised Learning, Model Training & Optimization, Attention Mechanisms
Cloud Computing: AWS EC2

Pipeline:

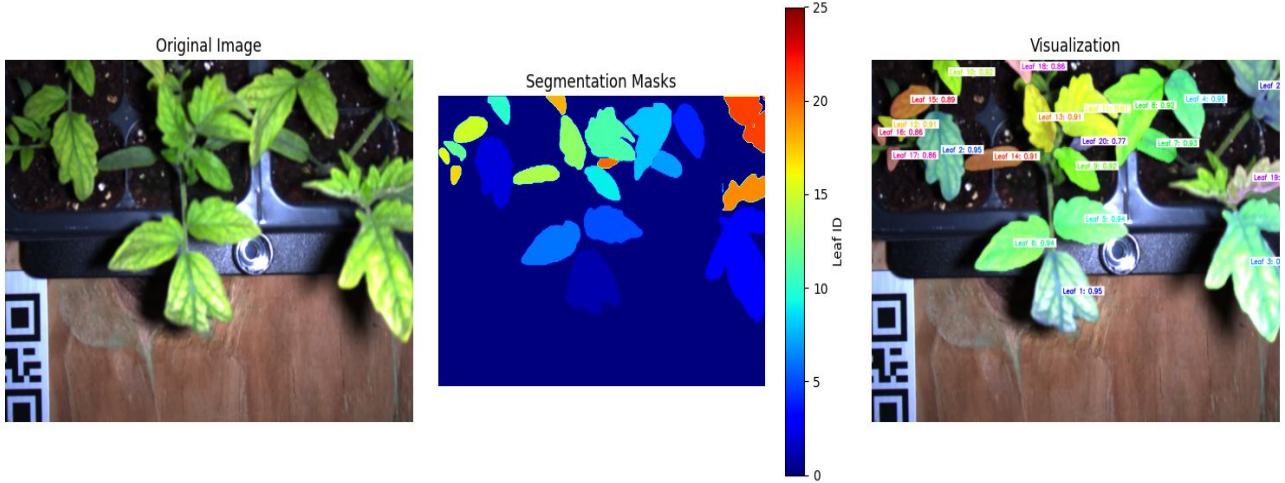


High-Level System view

YOLOv8 Leaf Instance Segmentation



High-Level System view

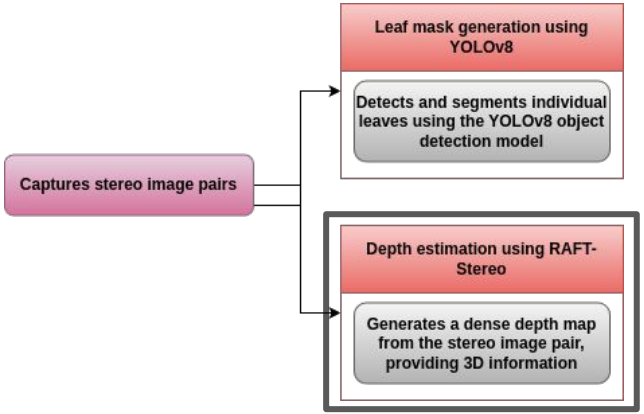


Output from YOLOv8 pipeline: Left: Original RGB image showing tomato plant; Center: Generated segmentation masks; Right: visualization with leaf IDs and confidence scores

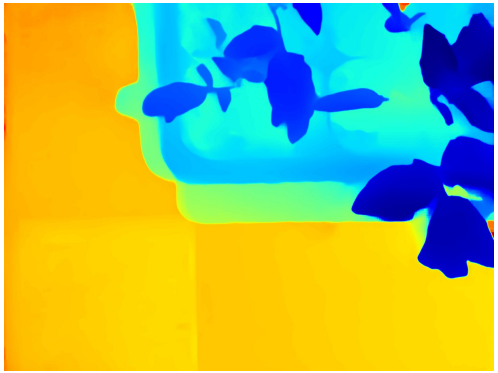
Key Details:

- Custom dataset: ~900 images of soybean and tomato plants
- 68% mAP@[0.5:0.95] for leaf mask generation

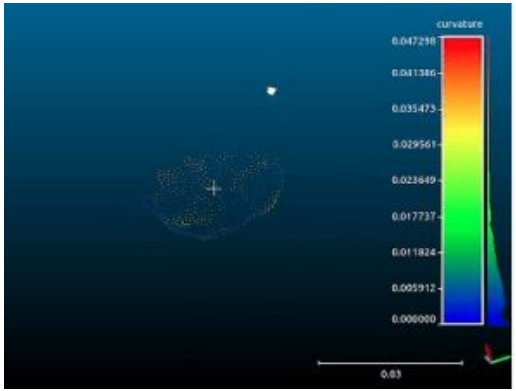
High-Precision Depth Estimation with RAFT-Stereo



High-Level System view



Depth Map



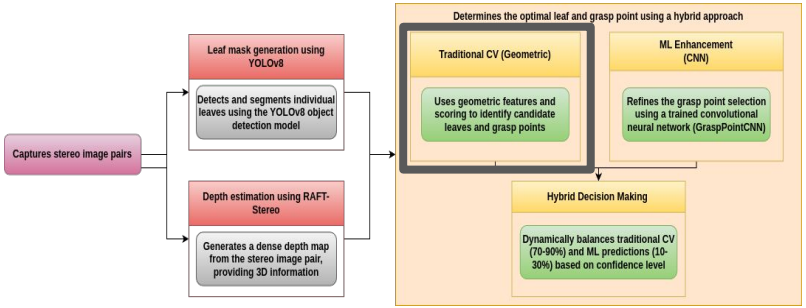
Geometric capture of leaf surface

Key Technical Details:

- Recurrent GRU refinement (the disparity map)
- 4D correlation volume computation
- Sub-pixel disparity accuracy (<0.5px) on 1080p stereo pairs
- Processing speed: ~150ms total pipeline latency on RTX 3080

Reconstructed 3D point cloud enabling precise spatial understanding

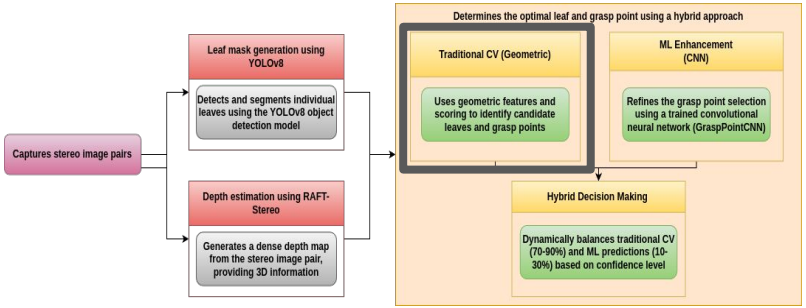
Traditional CV Pipeline: Geometric Grasp Point Selection



High-Level System view

Original camera view with leaf midrib

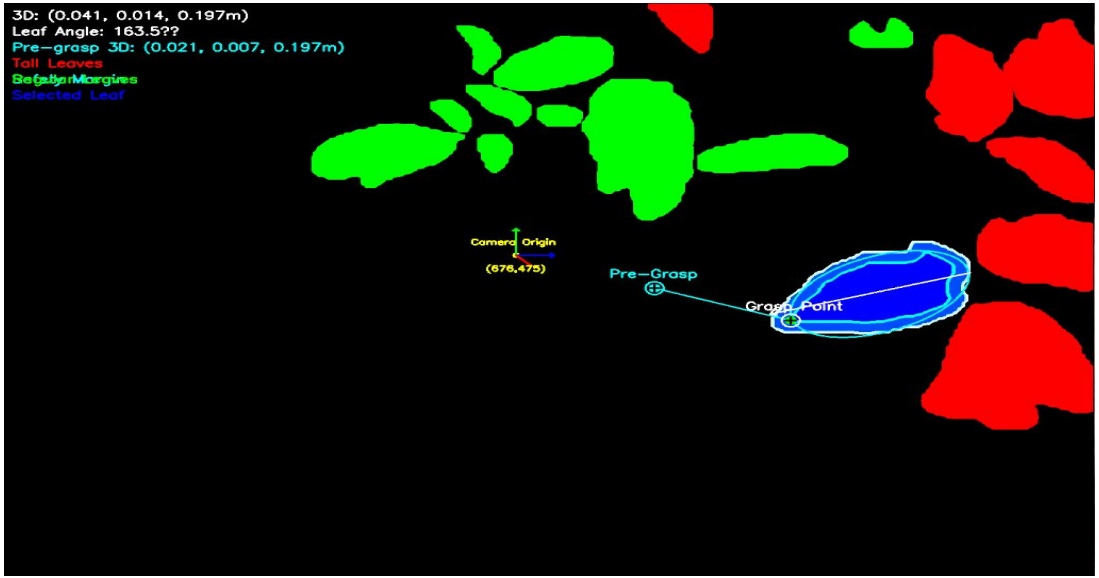
Traditional CV Pipeline: Geometric Grasp Point Selection



High-Level System view

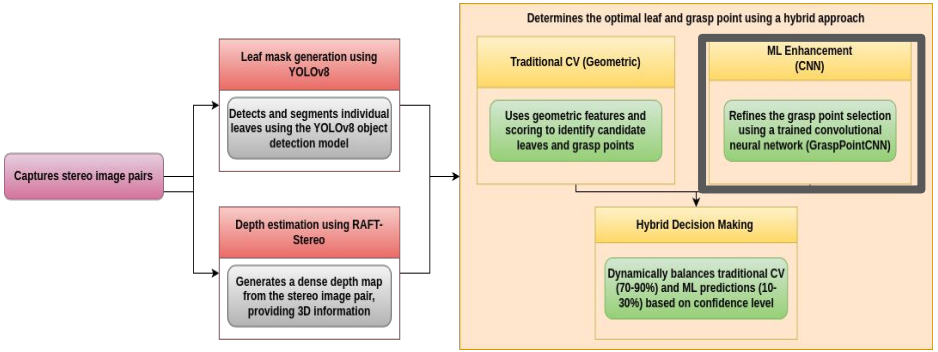
Optimal Leaf Selection (Pareto Optimization):
- Clutter Score (35%): Leaves that are isolated
- Distance Score (35%): Leaves that are closer to camera
- Visibility Score (30%): Leaves that are fully visible

Grasp Point Selection:
- Flatness Score (25%): Flatter regions of the leaf surface
- Approach Vector Score (40%): Approach vector (from the camera to the point) is closer to the camera's Z-axis
- Accessibility (15%): Distance of the grasp point from the camera origin and position relative to the center of the image
- Edge Awareness (20%): Grasp points within leaf boundary



Segmented leaves with optimal leaf & grasp point visualization

GraspPointCNN: ML-Based Grasp Refinement



High Level System view

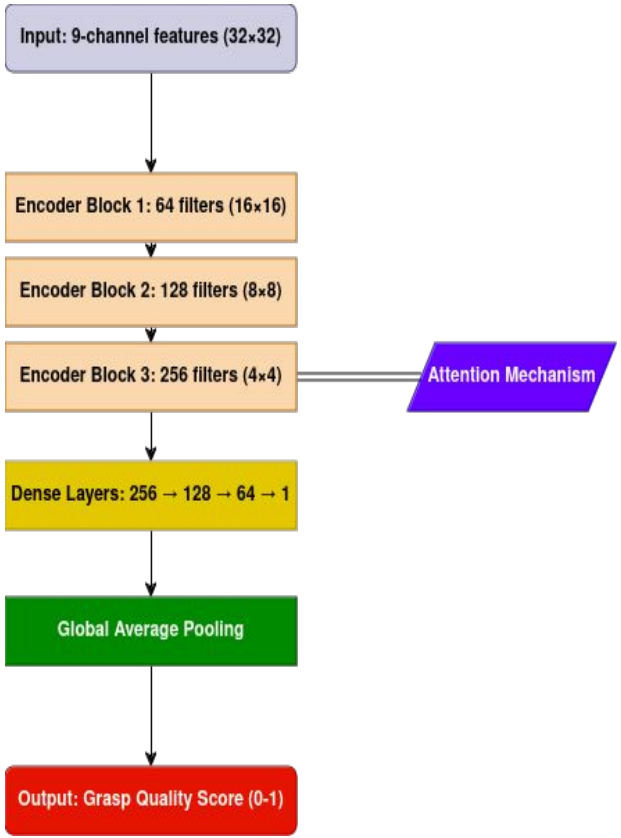
Self-Supervised Learning Pipeline:

- Input: 9-channel features (32×32)

- Depth map (1)
- Binary mask (1)
- Score maps (7): SDF, approach vector, flatness, isolation, distance, accessibility, stem penalty

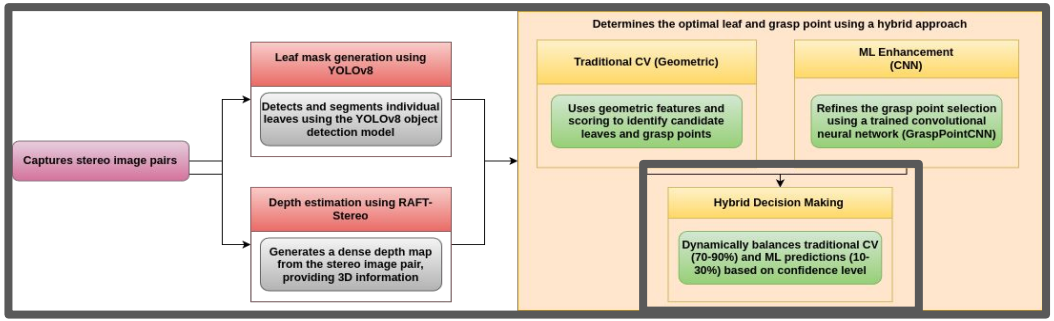
- Training Configuration:

- BCEWithLogitsLoss (pos_weight=2.0)
- Adam (lr=0.0005, weight_decay=0.01)
- ReduceLROnPlateau scheduler
- Early stopping (patience=15)
- Data augmentation: 90°,180°,270° rotations

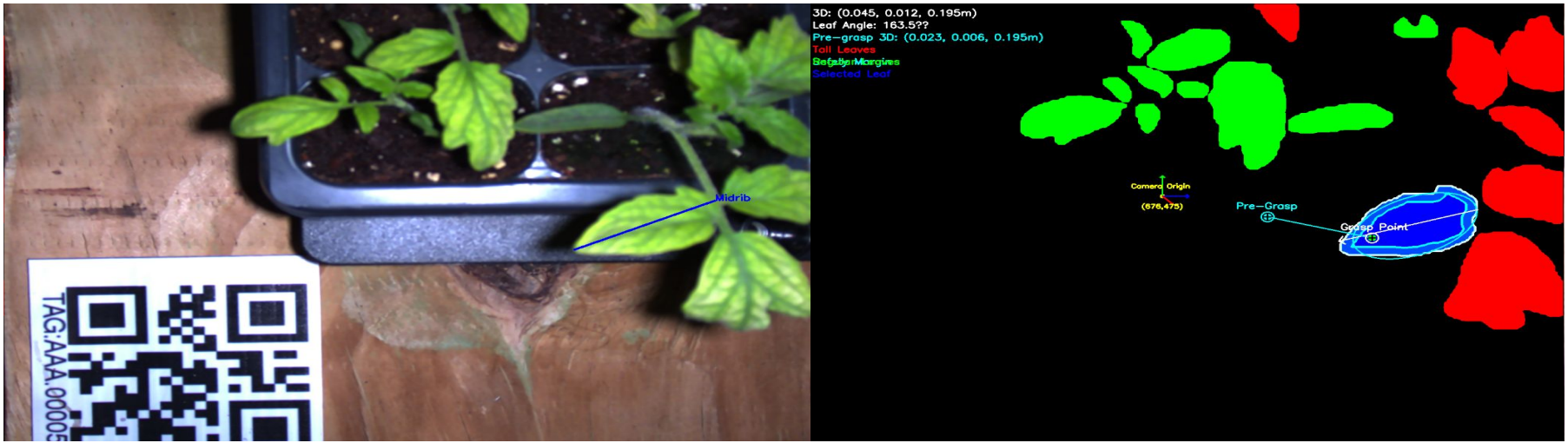


GraspPointCNN Architecture

Hybrid Decision Integration: Combining CV & ML

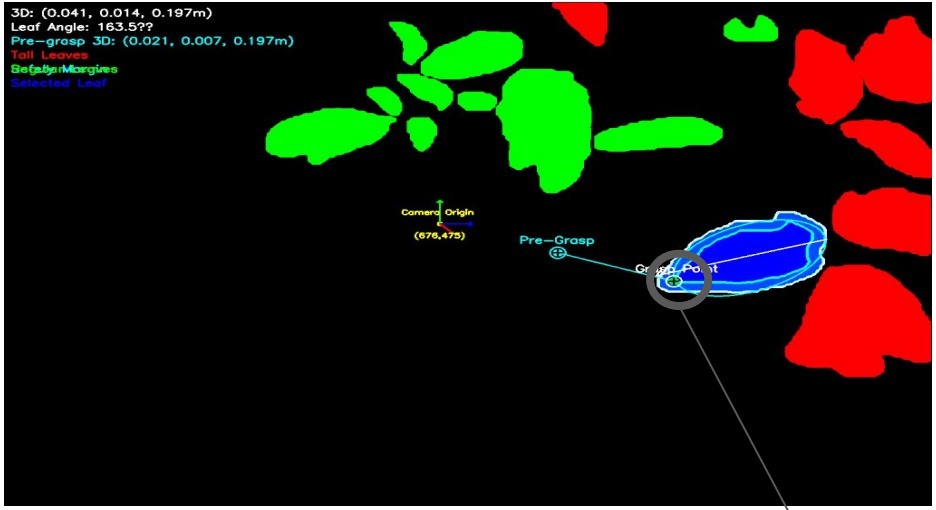


High-Level System view

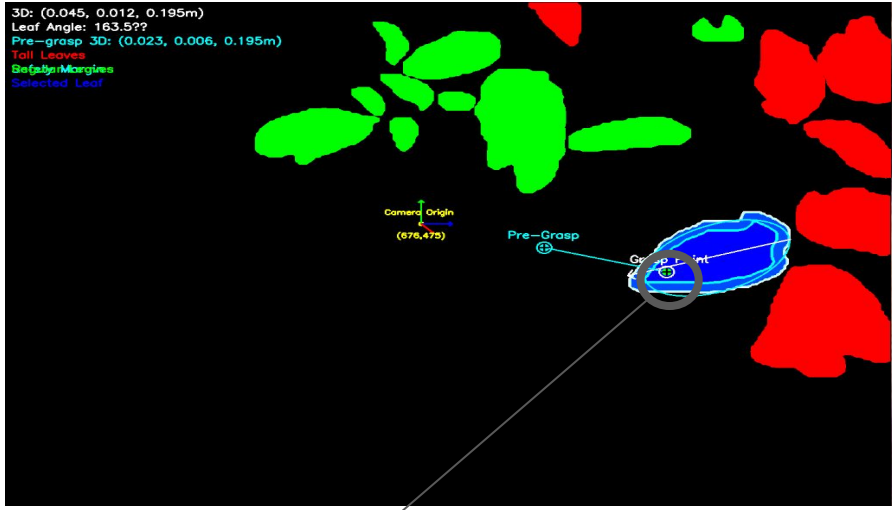


Hybrid CV-ML grasp point selection: Left - Original camera view with leaf midrib ; Right - Segmented leaves with grasp point visualization

Traditional CV vs Hybrid Decision System



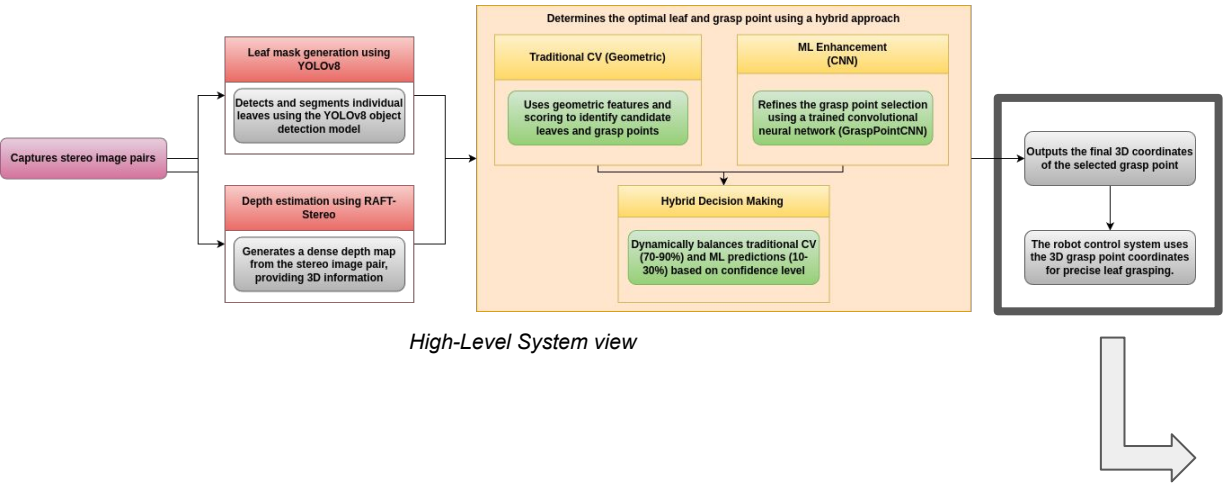
Traditional CV grasp point selection



Hybrid CV-ML grasp point selection

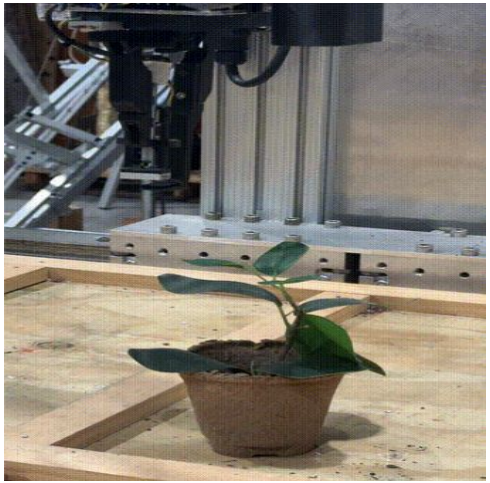
Grasping at the leaf tip often fails as the REX robot struggles to secure it, leading to missed grasps or leaf displacement. The hybrid grasp point selection method outperforms traditional CV, achieving a 4.66% improvement over 150 test cases.

REX Robot Integration for Leaf Grasping

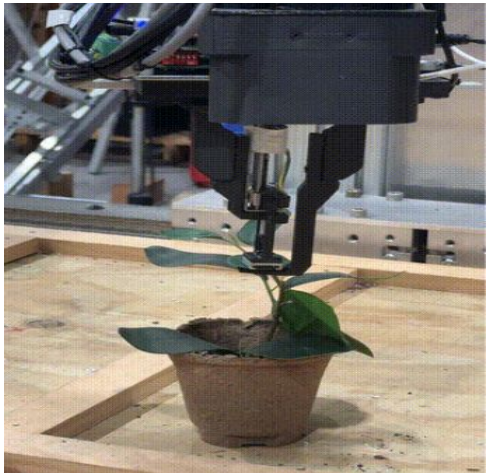


High-Level System view

(1)



(2)



(1) Robot capturing image (2) Robot grasping the leaf