ROBOT VISION

TERM PROJECT FINAL REPORT

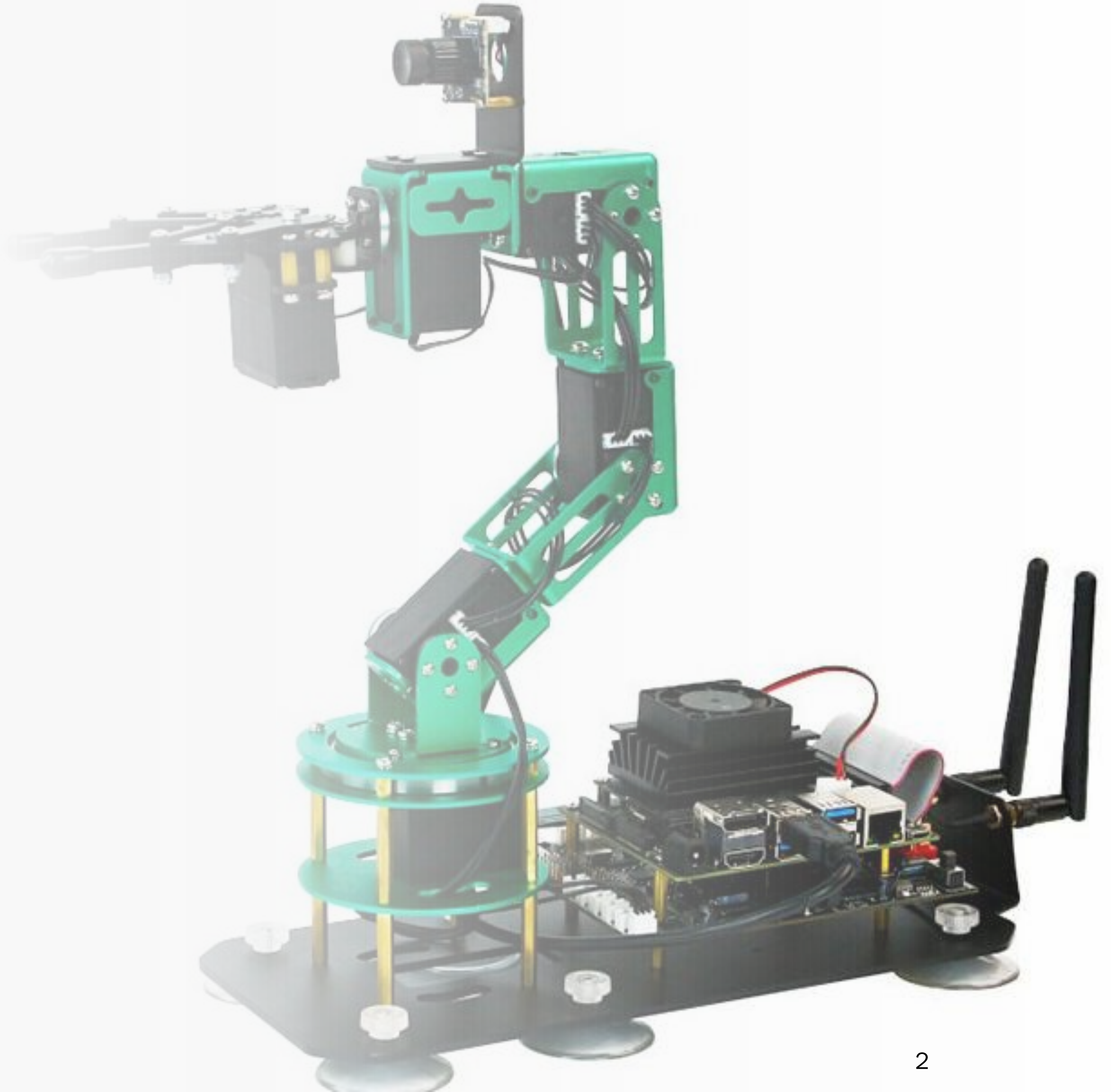
AUTOMATIC HARVESTING GRIPPER

Group Number : 7

Member: 李岳軒 R13522831
白峻維 R13522848
簡子恒 R13631001

Outline

- Introduction
- Objective
- Methods
- Results
- Conclusion
- References

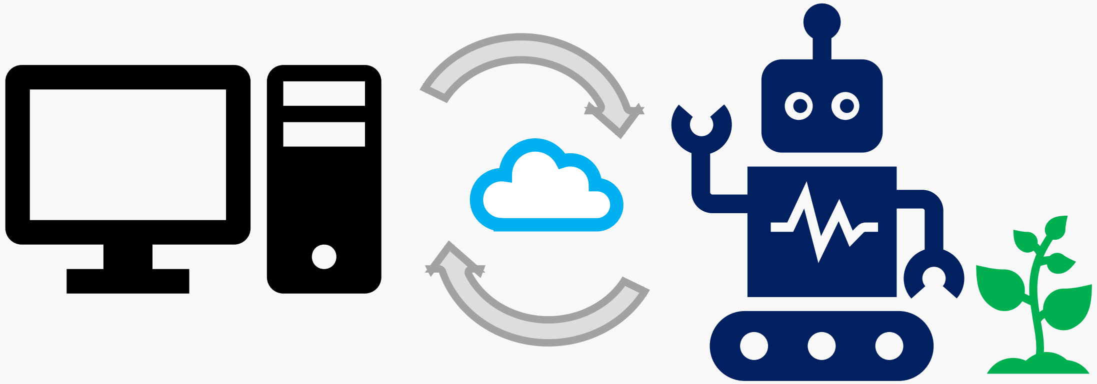


Introduction

Due to a significant decline in the agricultural labor force :

- The yield of agricultural products has become insufficient.
- The agricultural labor force is largely composed of older individuals.

→ Apply **automation** to improve production efficiency!!



Introduction

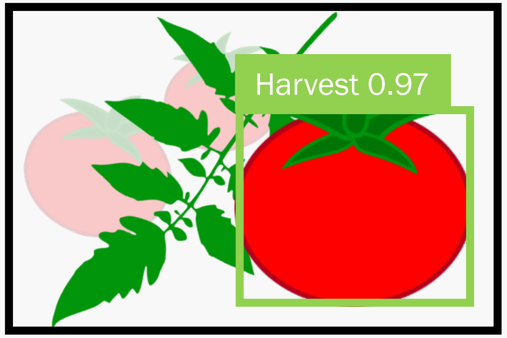
Challenge of Smart Agriculture :

- Complex growth environment - irregular layout / obstacles / light variation
- High cost of customized robot for farmer

Objectives

Tomato Ripeness and Defected Classification

- Implement machine learning models
- Determine the ripeness stage of tomatoes



Precisely Localization

- Adjust the robot arm according to the position of tomato
- Using stereo vision for size estimation

Integration with Robotic Gripper System

Methods - Tomato Gripper Algorithm

Launch Detection Model

⇩

Object Tracking ⇨ Return the **angle matrix** when the tomato is at the center of image

$$Angle^{1}_{Original} = [90, 158, 0, 5, 90, 90]$$

$$Angle^{2}_{Original} = [\theta_1, 105, 30, 10, 90, 90]$$

$$Angle^{1}_{Position} = [\theta_1, 158, 0, \theta_4^1, 90, 90]$$

$$Angle^{2}_{Position} = [\theta_1, 105, 30, \theta_4^2, 90, 90]$$

⇩

Positioning in 2D System ⇨ Use **triangulation** to locate the tomato in the world coordinate system

⇩

Positioning & Size Estimation ⇨ Use **pinhole camera model** to estimate the size of tomato

⇩

Grip the Tomato

Methods - System Environment

YAHBOM DOFBOT AI Vision Robotic Arm

- Ubuntu18.04 (JetPack 4.4)

- Python 3.6.9

- Pytorch 1.10.0 / torchvision 0.11.0

Restrict the version of the deep learning model used.

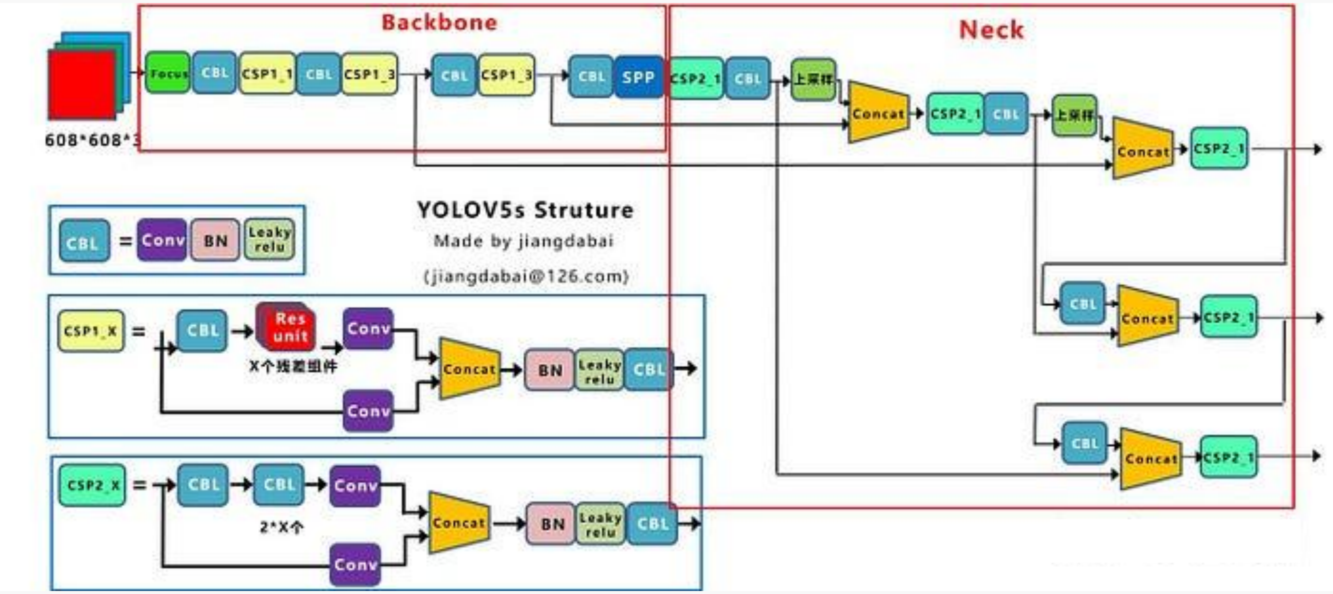
Methods - Tomato Detection Model

YOLO v5

<u>Version 4(2020): SiLu + AutoAnchor</u>

Version 5(2021): CSPDarknet + SPPF --- Pytorch ≥ 1.7

Version 6(2022): TensorRT --- Python ≥ 3.7



Tomato Dataset：

Label：Green / Turning / Harvest

Amount：1073

Ratio：8 : 2

Methods - Tomato Detection Model

YOLO v5

<u>Version 4(2020): SiLu + AutoAnchor</u>

Version 5(2021): CSPDarknet + SPPF --- Pytorch ≥ 1.7

Version 6(2022): TensorRT --- Python ≥ 3.7

Model	mAP50	Sped CPU	Params	FLOPs	FPS
YOLOv5n	**45.7**	**45**	**1.9**	**4.5**	**8.654**
YOLOv5s	**56.8**	**98**	**7.2**	**16.5**	**4.735**
YOLOv5m	64.1	224	21.2	49.0	-
YOLOv5l	67.3	430	46.5	109.1	-
YOLOv5x	68.9	766	86.7	205.7	-

(Performance on Jetson Nano)

Tomato Dataset：

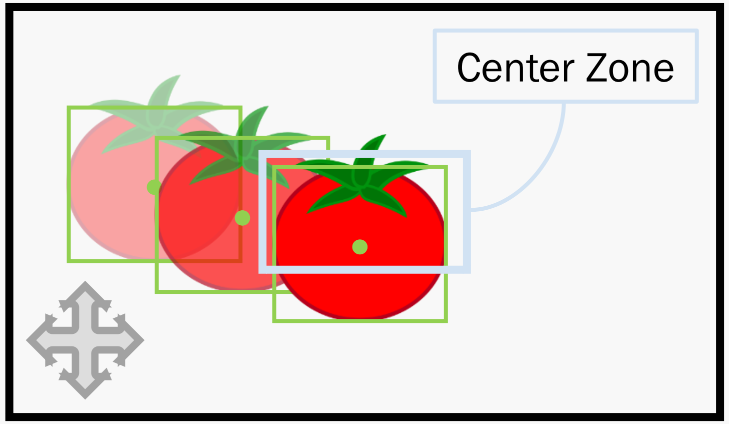
Label：Green / Turning / Harvest

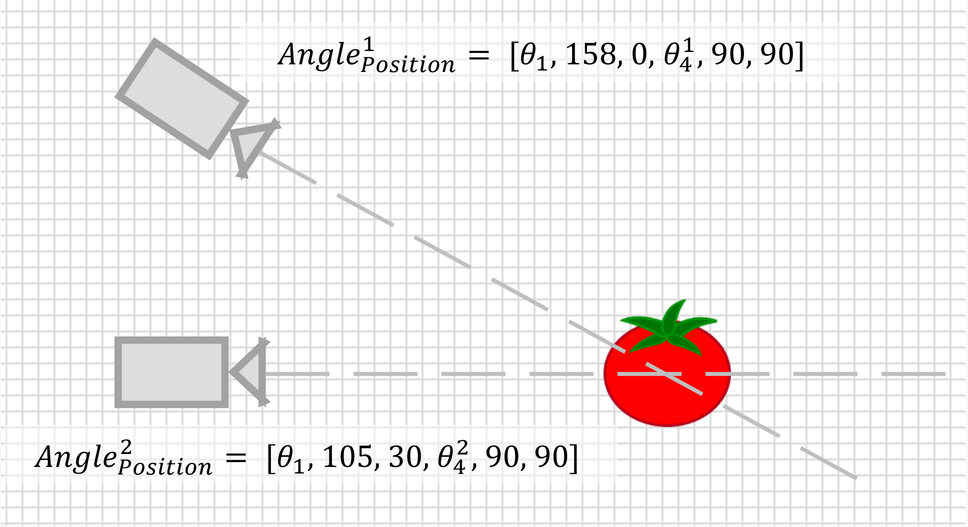
Amount：1073

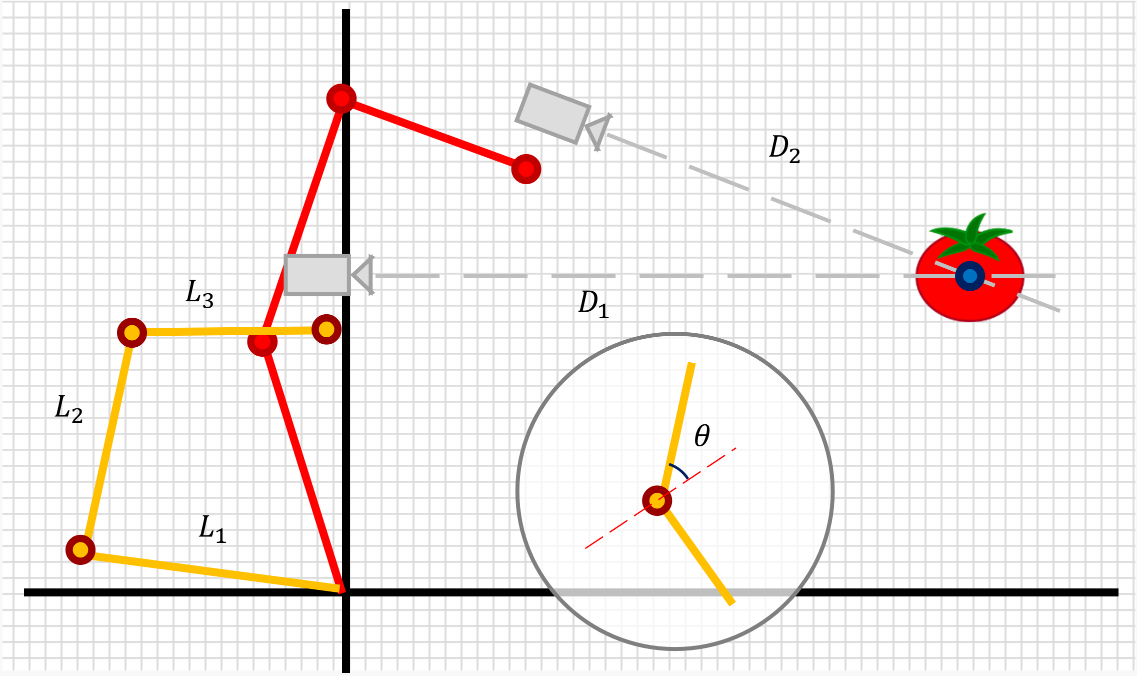
Ratio：8 : 2

Methods - Tomato Tracking & Positioning

Triangulation :

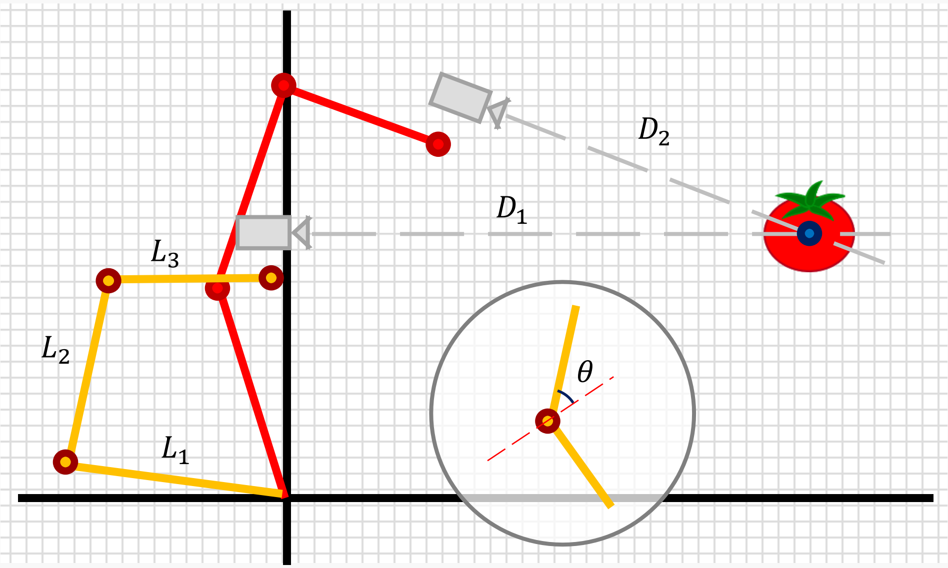






Methods - Tomato Tracking & Positioning

Forward Kinematics :

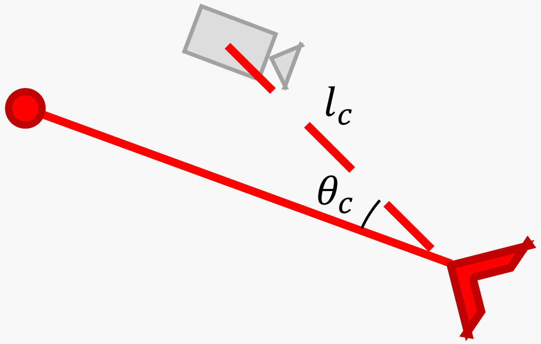


$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ 0 & L_1 \end{bmatrix} \cdot \begin{bmatrix} \cos \theta_1 \\ \sin \theta_1 \end{bmatrix}$$

$$\begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \begin{bmatrix} L_2 & 0 \\ 0 & L_2 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_1 + (\theta_2 - 90°)) \\ \sin(\theta_1 + (\theta_2 - 90°)) \end{bmatrix}$$

$$\begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \begin{bmatrix} L_3 & 0 \\ 0 & L_3 \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_1 + (\theta_2 - 90°) + (\theta_3 - 90°)) \\ \sin(\theta_1 + (\theta_2 - 90°) + (\theta_3 - 90°)) \end{bmatrix}$$

Hand-eye Calibration :



$$\begin{bmatrix} x_c \\ y_c \end{bmatrix} = \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} - \begin{bmatrix} L_c & 0 \\ 0 & L_c \end{bmatrix} \cdot \begin{bmatrix} \cos(\theta_1 + (\theta_2 - 90°) + (\theta_3 - 90°) - \theta_c) \\ \sin(\theta_1 + (\theta_2 - 90°) + (\theta_3 - 90°) - \theta_c) \end{bmatrix}$$

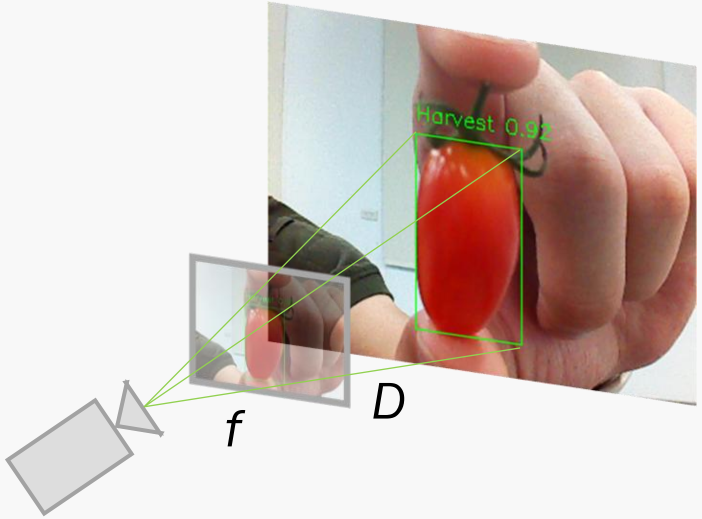
Tomato Positioning

$$[x_1\ y_1\ \theta_1\ x_2\ y_2\ \theta_2]$$

$$\tan \theta_1 (x_{Tomato} - x_1) + y_1 = \tan \theta_2 (x_{Tomato} - x_2) + y_2$$

$$x_{Tomato} = \frac{\boldsymbol{y_2} - \boldsymbol{y_1} + \tan \theta_1 x_1 - x_2 \tan \theta_2 x_2}{\tan \theta_1 - \tan \theta_2}$$

Methods - Tomato Size Estimation



(張正友標定法)

Utilize **Zhang's calibration method** provided by OpenCV to estimate the camera's **intrinsic parameters** and **distortion coefficients**.

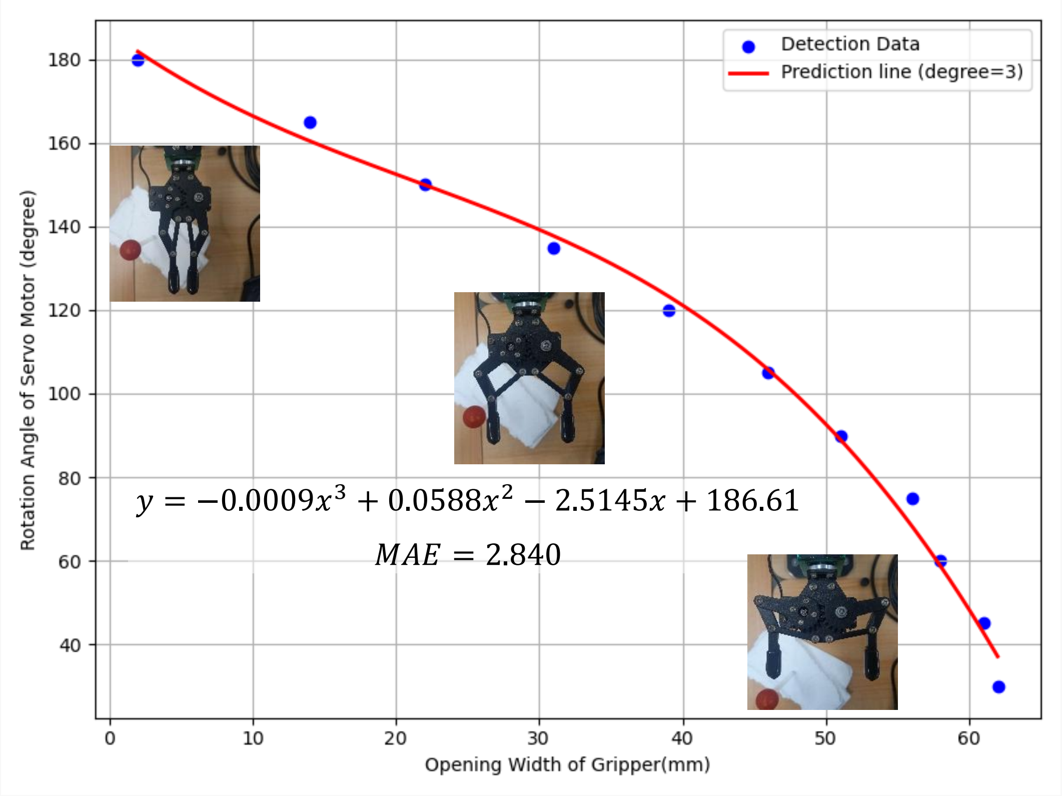
A. Capturing multiple images of a known-size checkerboard pattern.

B. Detecting 2D image coordinates of the corner points.

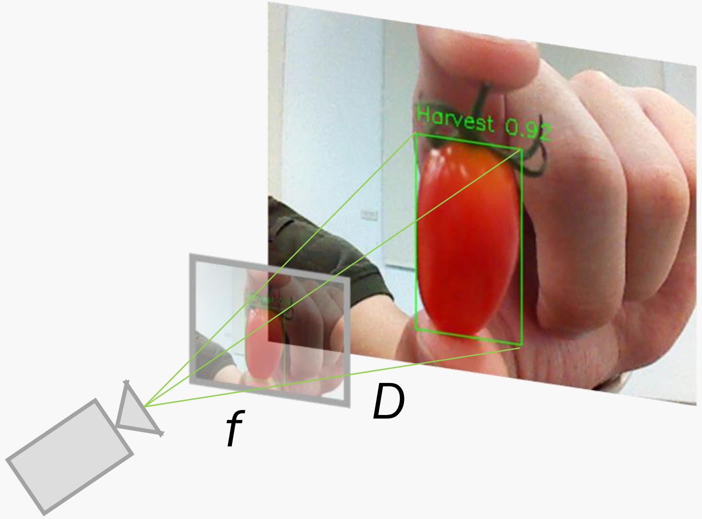
C. Matching to the corresponding 3D world coordinates.

D. Using nonlinear optimization to compute inintrinsics parameters (K) and distortion parameters (D).

The Relationship Between Gripper Opening Width And Servo Motor Rotation Angle



Methods - Tomato Size Estimation

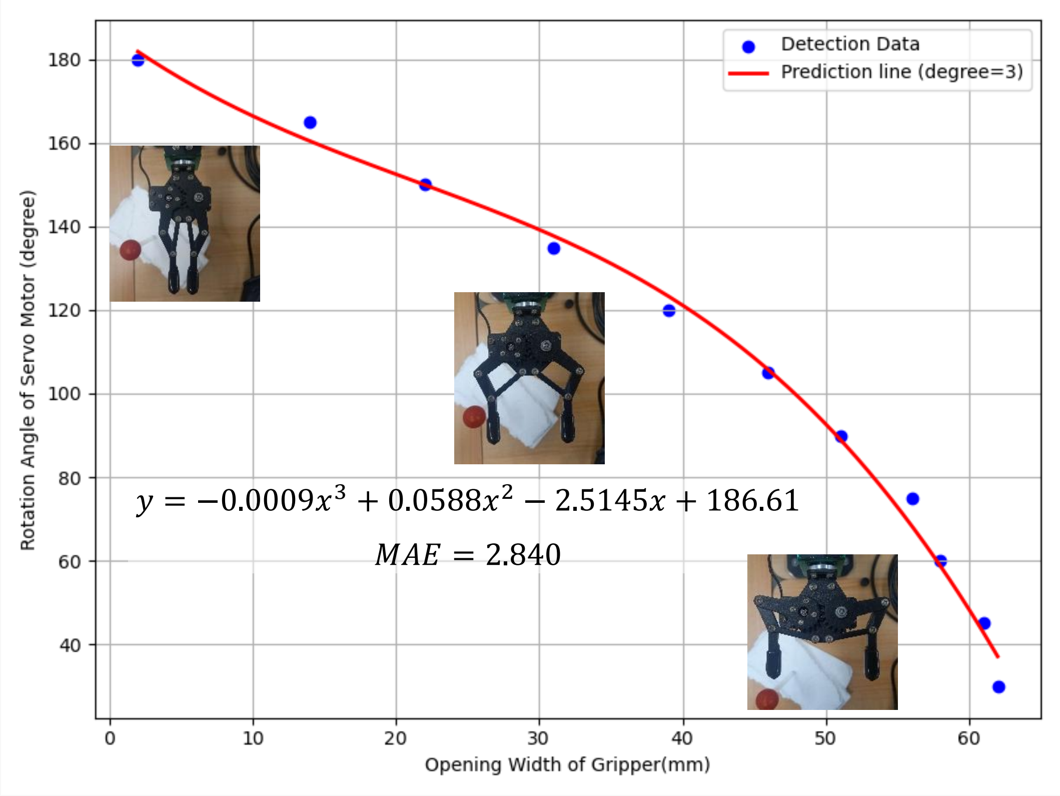


$$\begin{array}{cc} & \color{red}{f_x} \\ \mathbf{K} = & \begin{bmatrix} \boxed{976} & 0 & 282 \\ 0 & 978 & 235 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

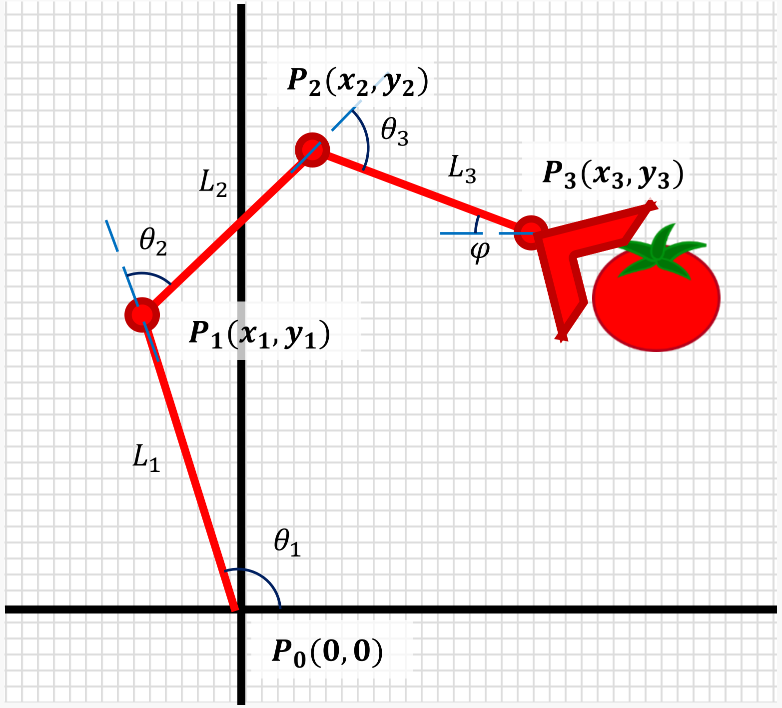
$$\mathbf{D} = \begin{bmatrix} -0.589 & 3.93 & 1.04 \times 10^{-3} & 3.72 \times 10^{-3} & -24.5 \end{bmatrix}$$

$$w_{real} = \frac{w_{pixel}}{f_x} \times \text{Distance}$$

The Relationship Between Gripper Opening Width And Servo Motor Rotation Angle



Methods - Inverse Kinematics



Pre-defined the desired grasping angle φ

- The angle **φ** is influenced by the picking task (fruit occlusion / gripper working range).
- To increase the robustness, **φ** is controlled within the range of 10° to 20°.

Inverse Kinematics

(1) The desired grasping angle φ is set as a reasonable angle in advance.

(2) Obtain equations ① and ② from coordinate point transform and end effector.

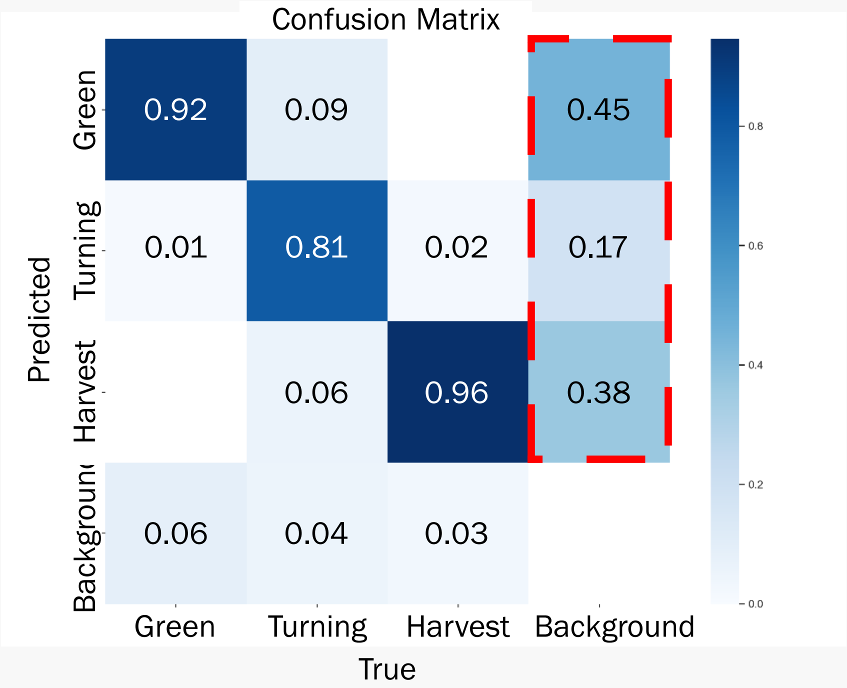
(3) Based on the geometric information, we can compute θ_3 by the equation ③.

$$\begin{cases} L2 \times \cos(\theta_1 - \theta_2) + L1 \times \cos(\theta_1) = x_2 = tomato_x - L3 \times \cos\varphi & \text{------ ①} \\ L2 \times \sin(\theta_1 - \theta_2) + L1 \times \sin(\theta_1) = y_2 = tomato_y + L3 \times \sin\varphi & \text{------ ②} \end{cases}$$

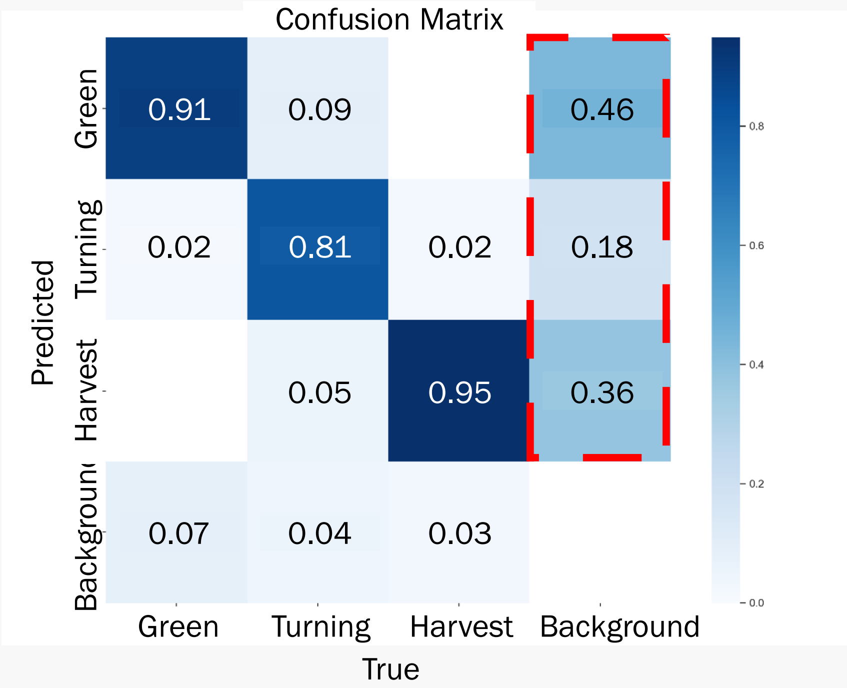
$$\theta_1 - \theta_2 - \theta_3 = -\varphi \quad \text{------ ③}$$

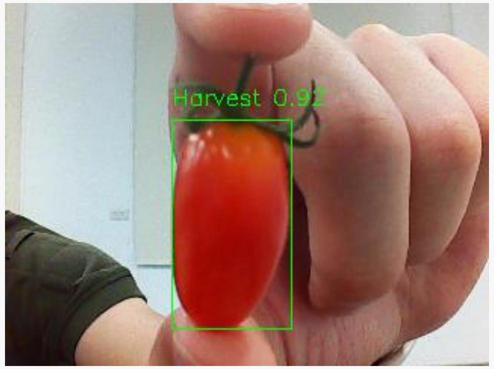
Results - Tomato Detection Model

YOLOv5n



YOLOv5s





$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

	YOLOv5n	YOLOv5s
Precision	0.868	0.873
Recall	0.833	0.877

Results - Positioning & Size Estimation

Positioning

True		Prediction		Error (Euclidean Distance)
X	Y	X	Y	
22.8	10.5	21.86	10.91	1.03
19.0	10.6	15.87	11.43	3.24
19.9	11.5	20.22	11.54	0.32
22.7	6.8	21.10	7.50	1.73
19.5	13.2	21.54	13.86	2.14
19.4	12.8	18.40	12.57	1.03
28.6	6.2	26.30	7.34	2.57
29.3	6.8	29.10	6.32	0.52
24.7	6.5	25.45	6.41	0.76
25.0	6.5	26.98	6.10	2.02

Average : 1.53

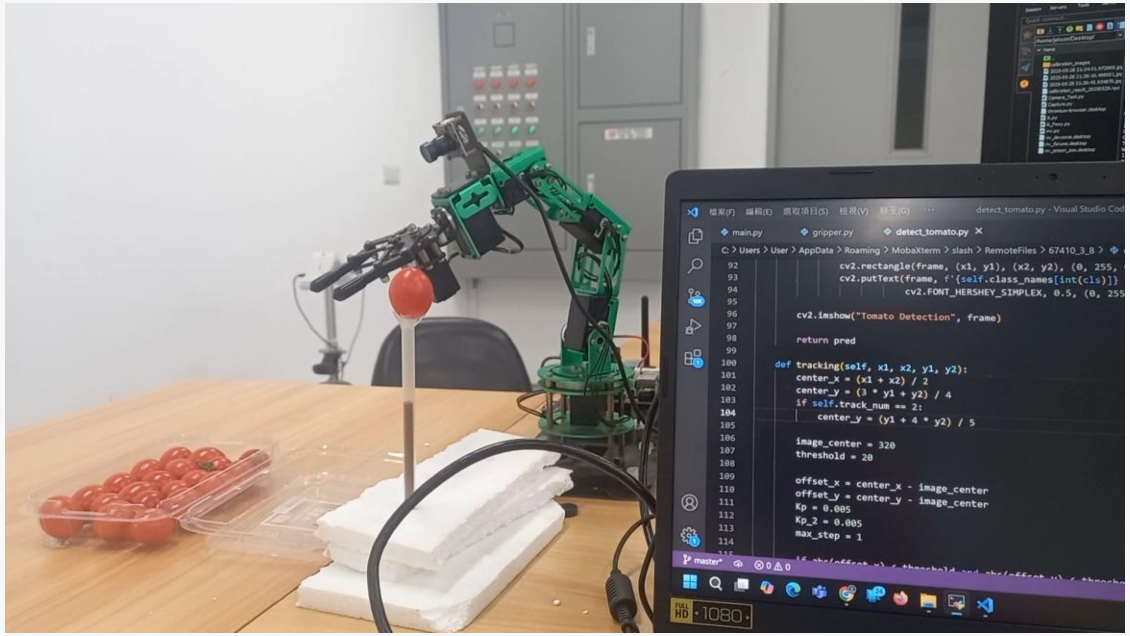
(Unit: cm)

Size Estimation

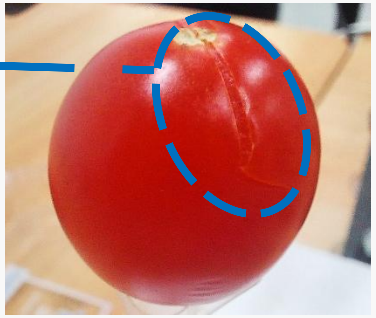
True	Prediction	Error
2.52	2.73	0.21
	2.82	0.30
	2.61	0.09
	2.41	0.11
	2.44	0.08
2.33	2.38	0.05
	2.14	0.19
	2.54	0.21
	2.34	0.01
	2.28	0.05

Average : 0.13

Results - Tracking & Gripping



Fruit cracking caused by mechanical impact



System Limitations

- The small difference in observation angles (15°~17°) leads to positioning error.
- Capturing images from a distance that is too close may lead to lose focus.

Conclusion

Development of an Automated Tomato Harvesting Gripper

- Tomato ripeness recognition model: Precision <u>0.86</u>, Recall <u>0.88</u>
- Stereo vision positioning accuracy: <u>1.53</u> cm
- Tomato size estimation accuracy: <u>0.13</u> cm

Future Works

- Reduce size estimation errors to lower the chance of cracks during the process
- Identifying cracks and related defects
- Shorten overall recognition time

References

- Yahboom DOFBOT AI Vision Robotic Arm with ROS for Jetson NANO 4GB B01 and Raspberry Pi 4B 8GB/4GB
- Strawberry Harvesting Robot BERRY
- 飆機器人-DOFBOT NVIDIA AI ROS 機械手臂
- Ge, Y., Xiong, Y., & From, P. J. (2019). "Instance segmentation and localization of strawberries in farm conditions for automatic fruit harvesting," IFAC-PapersOnLine, 52(30), 294-299.
- 農業部農業藥物試驗所-作物病蟲害診斷系統(診斷編號1070620005)
- Yi-Hsuan Huang, Ta-Te Lin, "Development of a High-Throughput Phenotyping system for greenhouse tomato fruits based on deep learning", Department of Bio-Industrial Mechatronics Engineering , National Taiwan University, 2019.
- GitHub-ZJU-lishuang / yolov5-v4
- GitHub-ultralytics / yolov5
- YOLO v5 — Explained and Demystified(Object Detection)
- Camera Calibration 相機校正
- OpenCV - Camera Calibration and 3D Reconstruction

The End

李岳軒	白峻維	簡子恒
研究動機發想 相機校正參數提取 逆向運動學算法與程式 結果分析與討論	文獻探討 研究動機發想 逆向運動學算法 結果分析與討論	夾爪演算法設計與系統建立 番茄辨識模型訓練 前向運動學程式 番茄尺寸預估程式 逆向運動學算法與程式 夾爪開度線性關係式建立 結果分析與討論 簡報架構製作