



# Harvester

An automated harvesting system for ground agricultural crops.



# Motivation



AI

ML

Image  
Processing

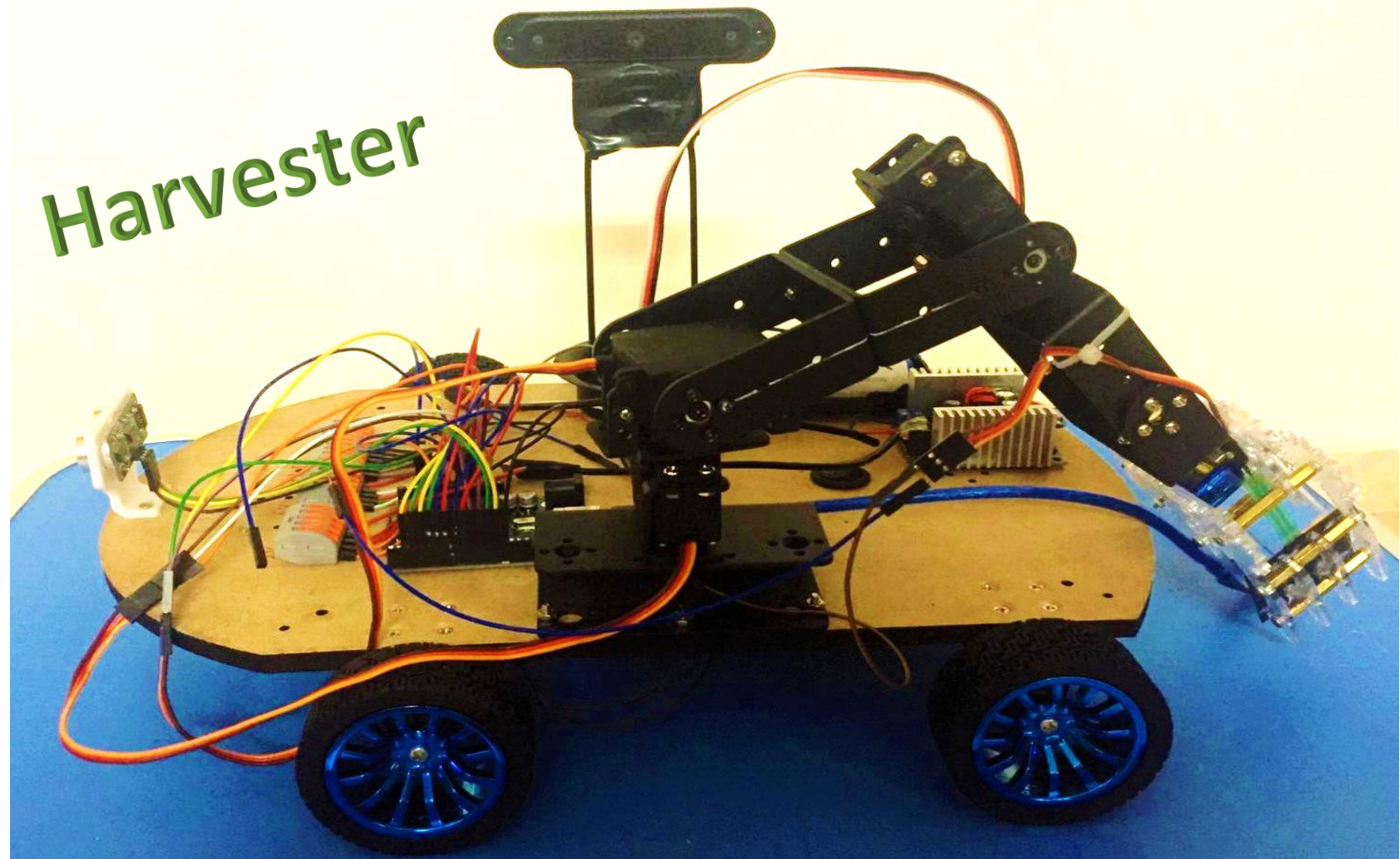


# Objectives

We aim to invent a robot called "Harvester" where its job is to harvest the plants from the agriculture ground such as tomatoes, cucumbers and peppers.



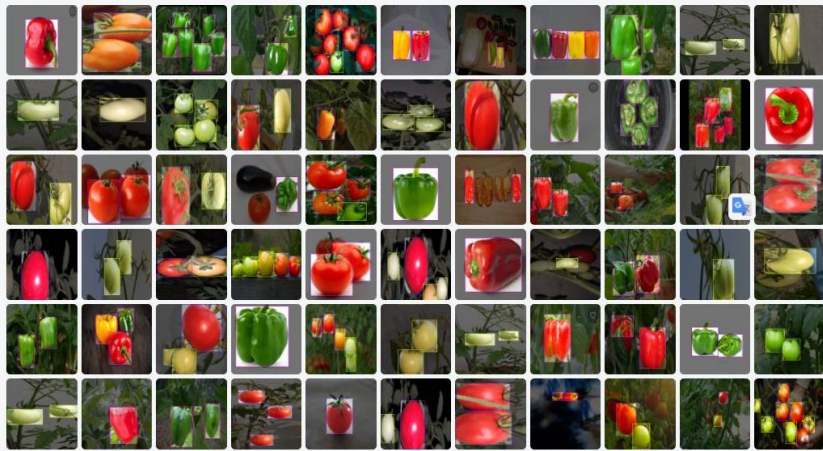
Harvester



# Dataset

1049 Total Images

Training Set 749 Validation Set 197 Testing Set 103



# Training

```
CO Harvester.ipynb ☆
PRO File Edit View Insert Runtime Tools Help Last edited on November 28

+ Code + Text

v3 (mse loss, Normalizer: (iou: 0.75, obj: 1.00, cls: 1.00) Region 23 Avg (IOU: 0.881891), count: 3, class_loss = 0.019418, iou_loss = 0.0
total_bbox = 2373223, rewritten_bbox = 0.117688 %

(next mAP calculation at 12000 iterations)
Last accuracy mAP@0.5 = 90.88 %, best = 92.39 %
12000: 0.691380, 0.553276 avg loss, 0.000010 rate, 0.948720 seconds, 768000 images, 0.019498 hours left
Resizing to initial size: 416 x 416 try to allocate additional workspace_size = 52.43 MB
CUDA allocate done!

calculation mAP (mean average precision)...
Detection layer: 16 - type = 28
Detection layer: 23 - type = 28
196
detections_count = 1016, unique_truth_count = 493
class_id = 0, name = green-tomatoes, ap = 87.25% (TP = 134, FP = 42)
class_id = 1, name = pepper, ap = 91.79% (TP = 145, FP = 14)
class_id = 2, name = red-tomatoes, ap = 92.94% (TP = 156, FP = 36)

for conf_thresh = 0.25, precision = 0.83, recall = 0.88, F1-score = 0.85
for conf_thresh = 0.25, TP = 435, FP = 92, FN = 58, average IoU = 69.92 %

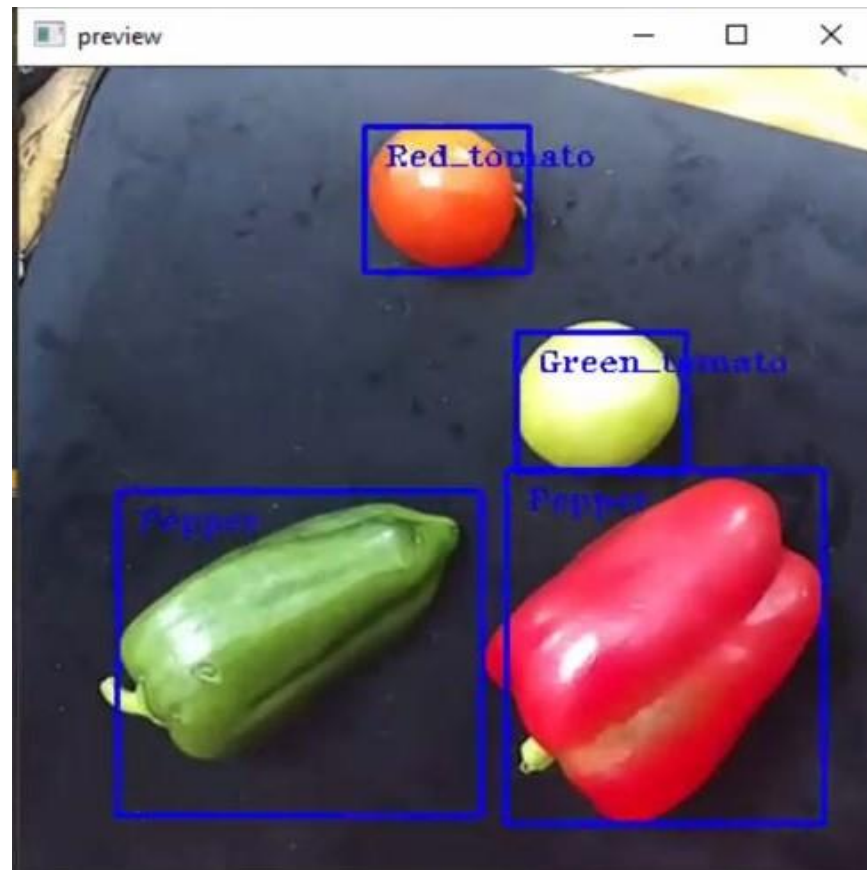
IoU threshold = 50 %, used Area-Under-Curve for each unique Recall
mean average precision (mAP@0.50) = 0.906623, or 90.66 %
Total Detection Time: 0 Seconds

Set -points flag:
'-points 101' for MS COCO
'-points 11' for PascalVOC 2007 (uncomment 'difficult' in voc.data)
'-points 0' (AUC) for ImageNet, PascalVOC 2010-2012, your custom dataset

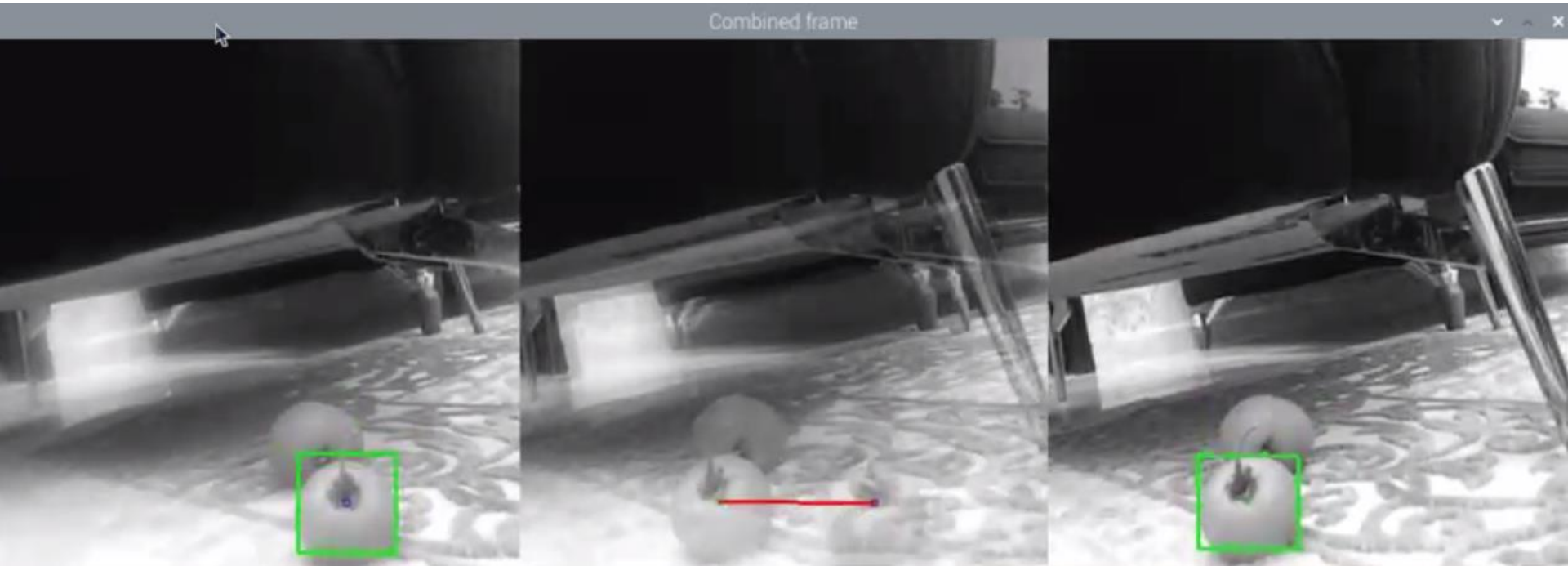
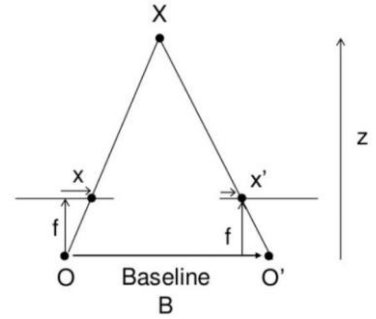
mean average precision (mAP@0.5) = 0.906623
Saving weights to backup//custom-yolov3-tiny-detector_last.weights
Saving weights to backup//custom-yolov3-tiny-detector_final.weights
If you want to train from the beginning, then use flag in the end of training command: -clear
```



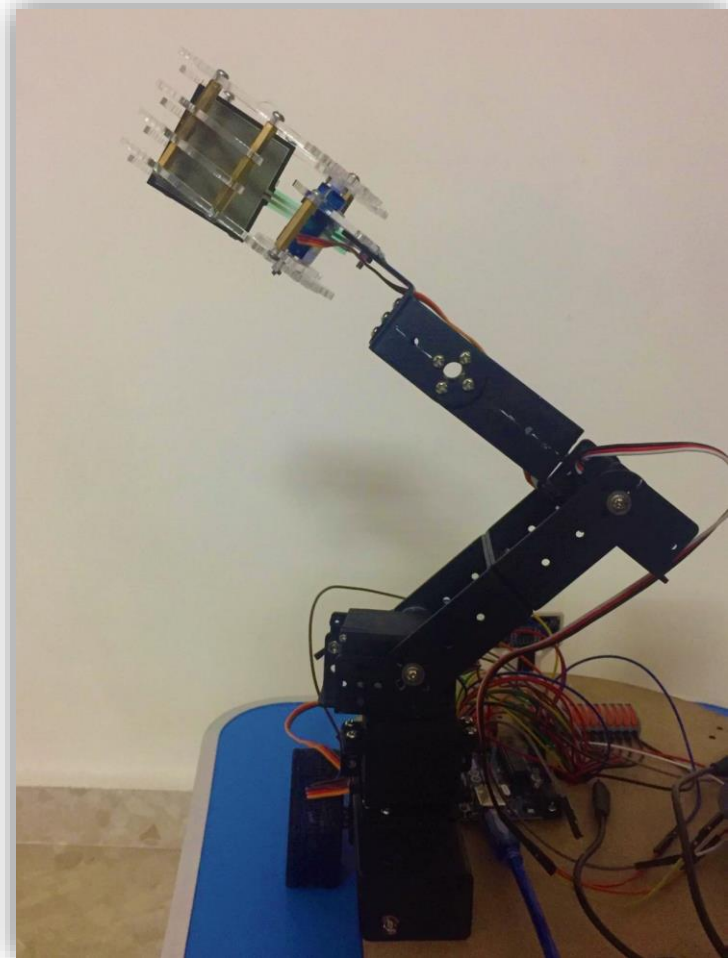
# Deploy



# Depth

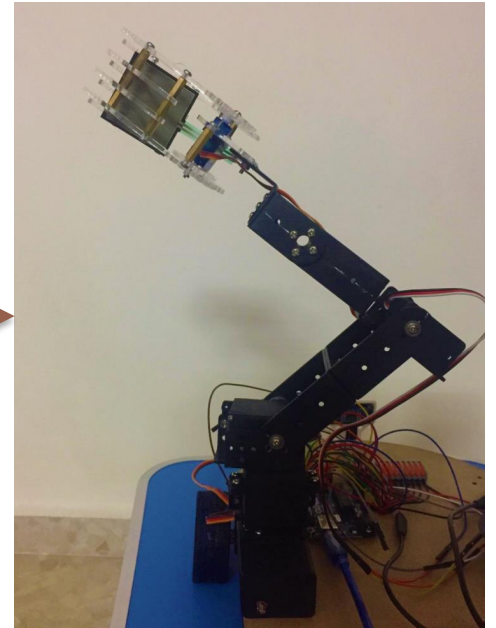


# Control the arm

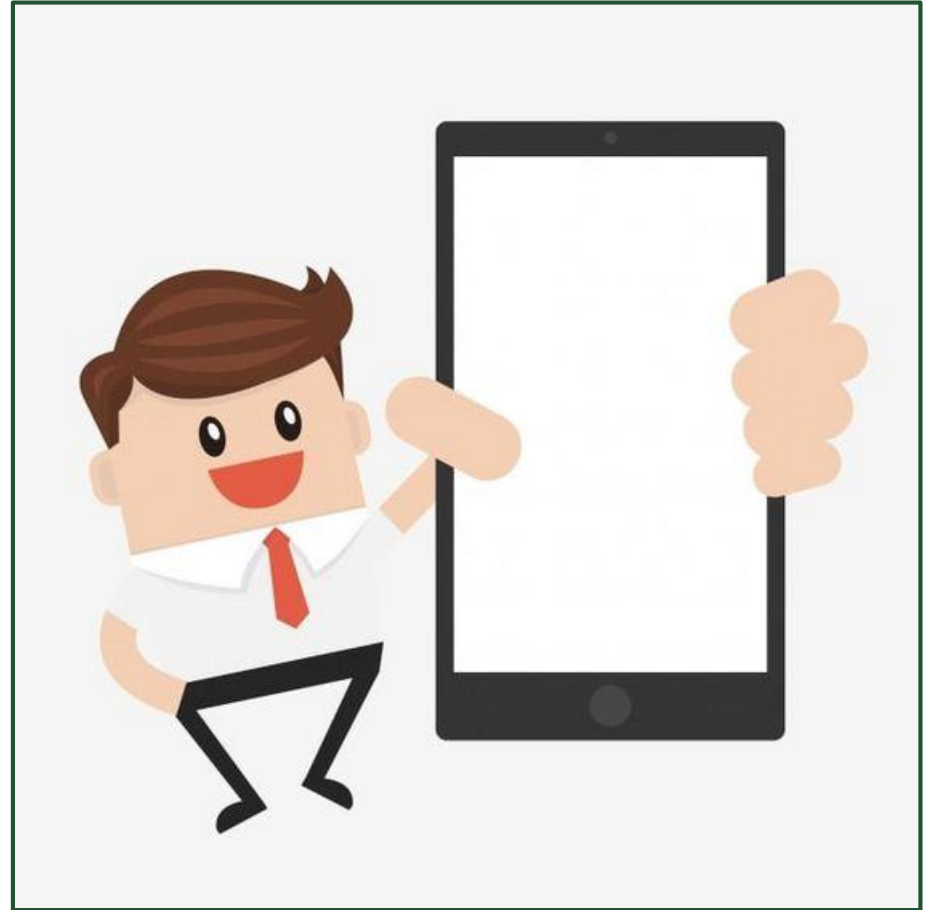




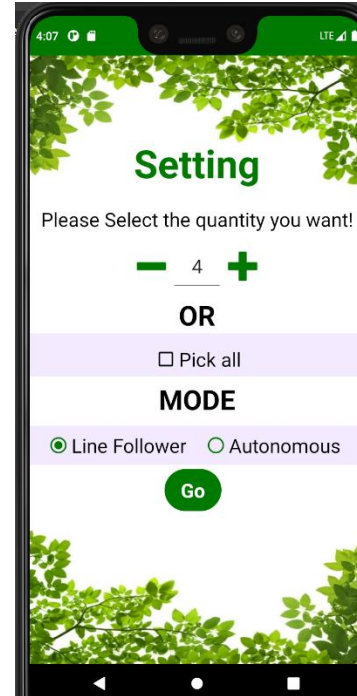
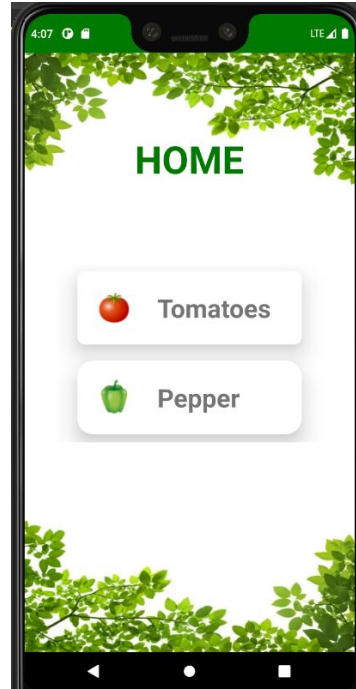
Where the results of localization and positioning from OAK-D camera measured in centimeters and we needed to turn them to angles moved by the 3 motors in the arm.

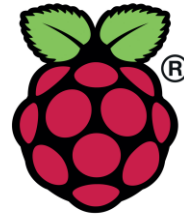
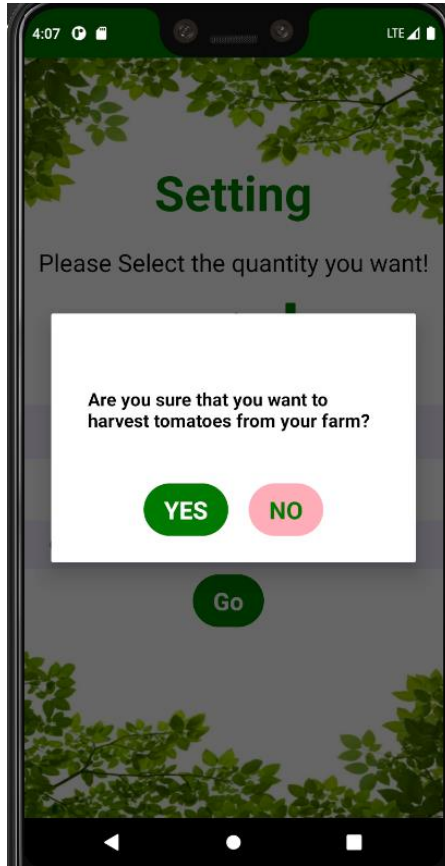


Using WIFI, "Harvester" gets information from the Harvester mobile application to specify what the farmer needs regarding **movement mode, plant type, and quantity.**



# Harvester Mobile Application





**Raspberry Pi**

**"Harvester" walks through  
the nursery searching for  
the demanded plant.**



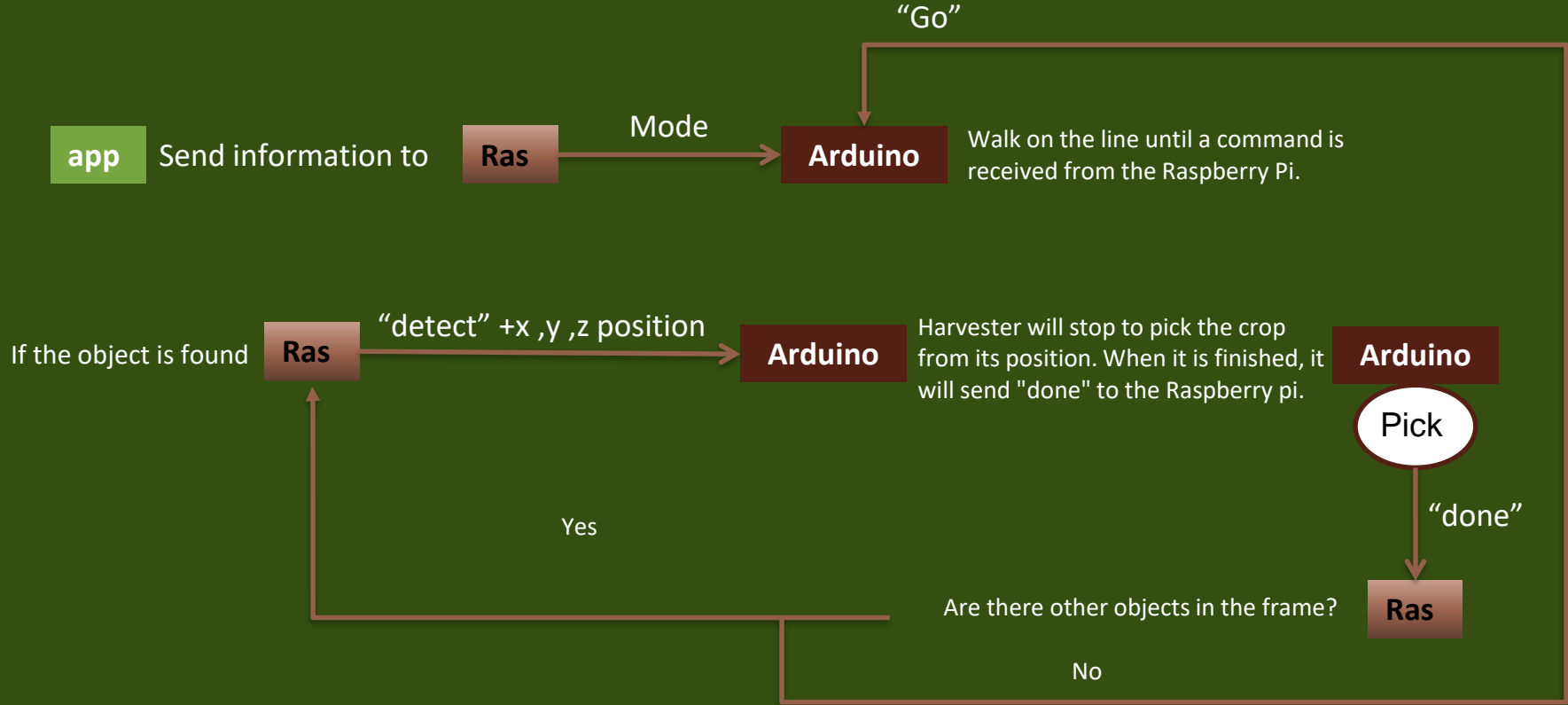


# **2 MODES**

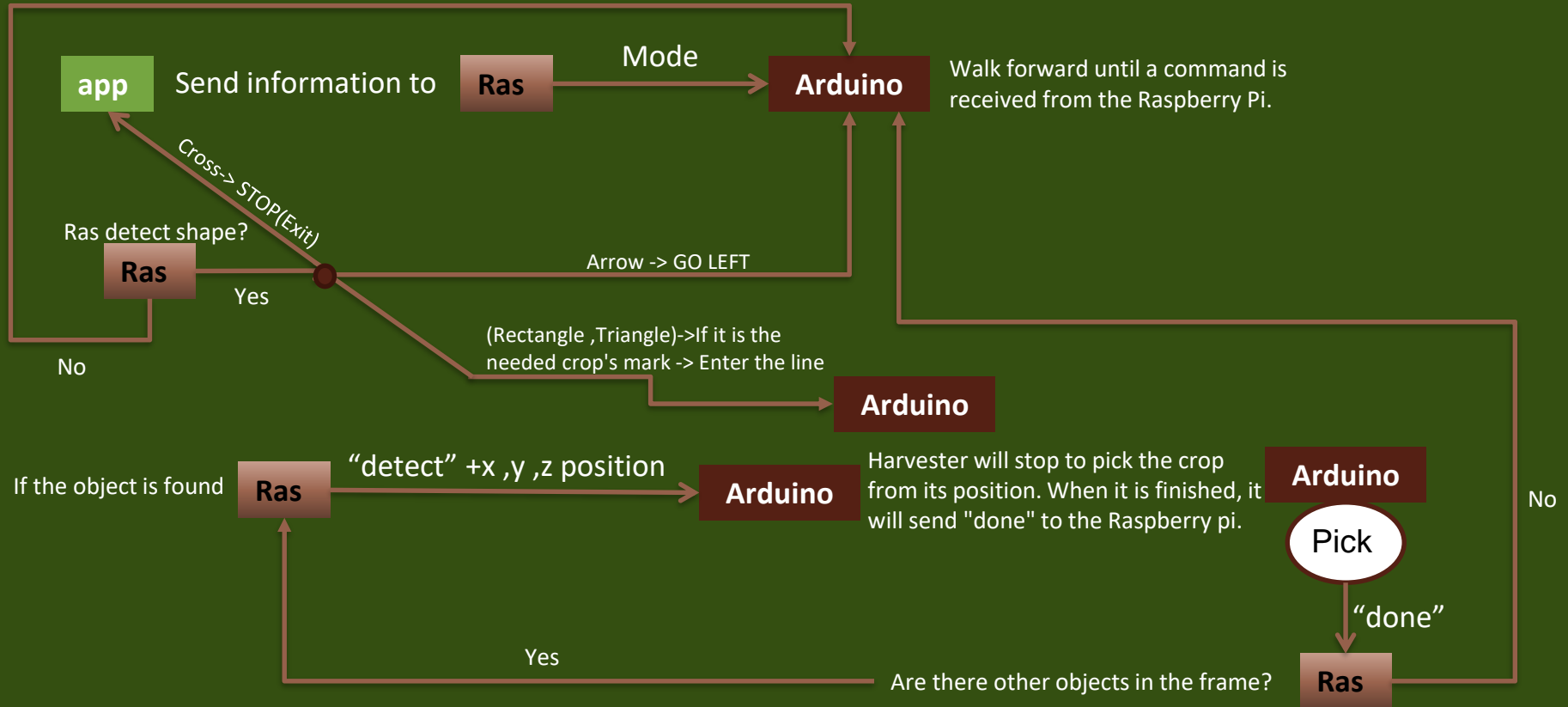
**Line Follower**

**Autonomous**

# Line Follower



# Autonomous



The background is a solid light green color. It features several white, organic, cloud-like shapes. One large white shape on the left contains the text "Thank You". There are also smaller white circles and several green leaf illustrations scattered around the white shapes. The leaves are simple line drawings with green fill and black outlines.

**Thank You**