# Application of Vision Sensors for Auto-guidance and Yield Monitoring of an Unmanned Orchard Robot

Seung-Yeon Yu[1], Semin Ji[1], Yong-Hyun Kim[1, 2], Sung-Joo Kim[1], Hak-Jin Kim[1, 2*]

[1] Department of Biosystems & Biomaterials Science and Engineering, Seoul National University, Seoul, 08826, Korea
[2] Global Smart Farm Convergence Major, College of Agriculture and Life Sciences, Seoul National University, Seoul 08826, Korea

* Corresponding Author : T. 02-880-4604 F. 02-873-2049 Email: kimhj69@snu.ac.kr

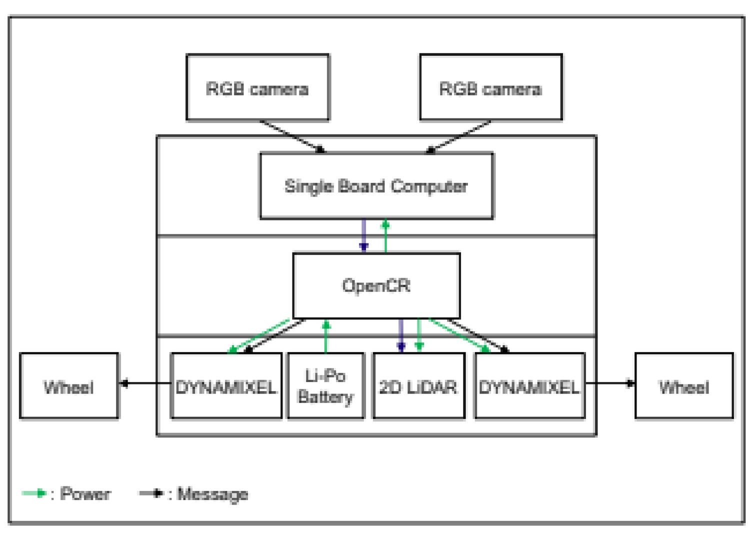
## Introduction

- Unmanned robots in the orchard can help the orchard management by monitoring fruit growth status information in real time while driving by itself.
- The use of vision sensors is high for the purpose of solving the problem of poor GNSS reception due to the canopy of orchards, avoiding scattered obstacles, and detecting the growth status of fruits.
- In this study, a ROS-based robot platform that can perform autonomous driving and fruit growth status monitoring functions using vision sensors was constructed.

## Materials and Methods

### ❖ Materials

- The Turtlebot3 Burger (Robotis) was equipped with Jetson Nano (NVIDIA) as a single board computer, OpenCR 1.0 (Robotis) that involve IMU as a controller, RPLiDAR A2M8 (Slamtec) as a 2D LiDAR sensor and two webcams c270 (Logitech) as cameras as shown in Figure 1.
- An experimental environment which was imitated the orchard environment was constructed with a fruit tree model and boxes as shown in Figure 2.

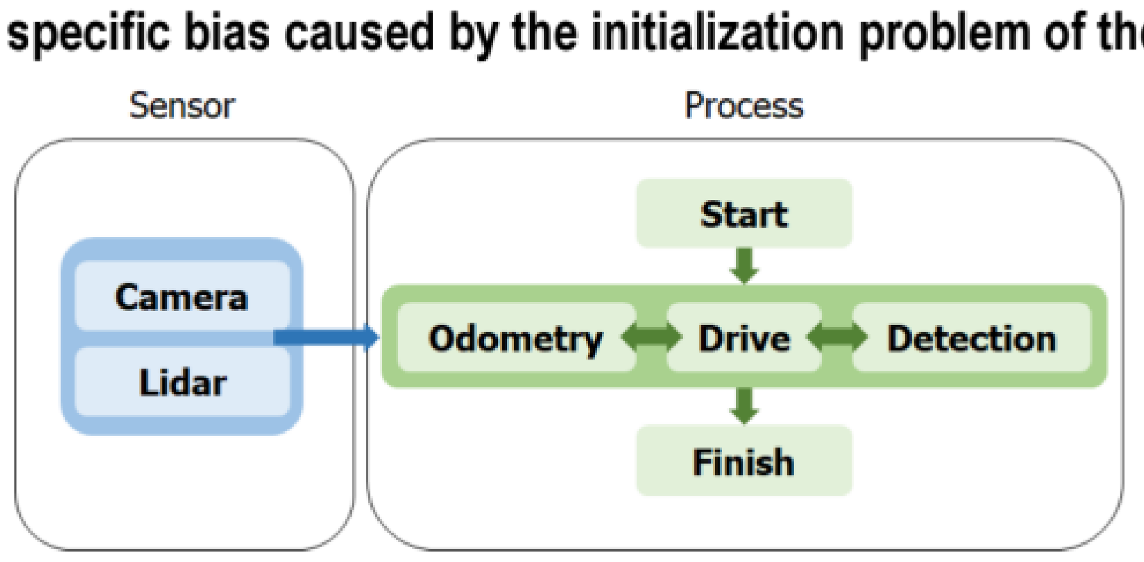



< Figure 1. Hardware physical connection>

< Figure 2. Experimental environment >

### ❖ Methods

- In order to recognize fruit growth status information in various environments and to reduce the load on jetson nano, a learning model was created using the yolov5n model.
- After learning 793 images for one fruit tree model, 346 images for several fruit tree models were additionally transfer-learned and tuned to the camera and illumination conditions.
- The classes were set to three: Tree, Fruit, and SickFruit, and batch=4, decay=0.0005, and learning_rate=0.0001 were used as parameter values for model learning.
- In order to distinguish each fruit tree without a feature point on the image, an area where one fruit tree can be recognized was set, and the odometry value in each area was obtained to calculate the tree number of the left image and the right image as shown in Figure 3.
- When calculating the odometry of the robot, the accuracy was improved by correcting a specific bias caused by the initialization problem of the IMU sensor.

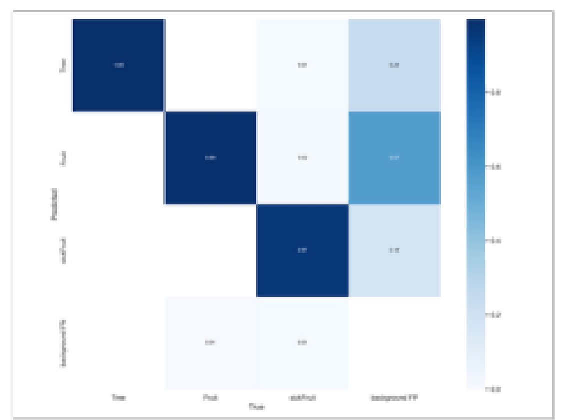



< Figure 3. Algorithm Overview >

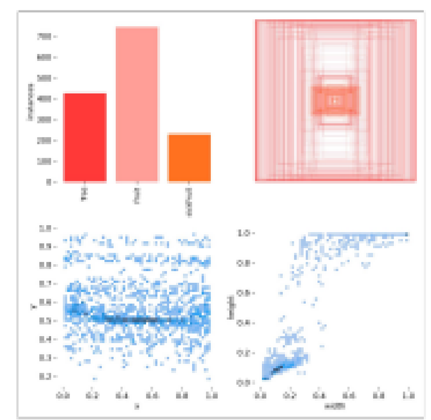
## Results and Discussion

### ❖ Performance of Detection

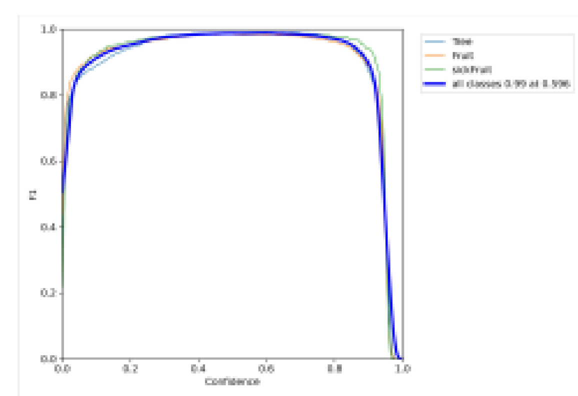
- Through the confusion matrix for tree, fruit, and sick fruit, which are detection targets in Figure 4, it was found that tree was detected with 100%, fruit was 99%, and sick fruit was 97% accuracy.
- Figure 5 is a group of graphs showing box information of labels. The size of boxes size are variously configured, and the distribution of x and y is also evenly distributed.
- The graph for the F1 score is shown in Figure 6.
- The values of precision, recall, and mAP at the last epoch are shown in Table 1.



< Figure 4. Confusion Matrix >



< Figure 5. Information of labels >



< Figure 6. F1 score curve >

Table 1. Performance evaluation of the model

| Precision (%) | Recall (%) | mAP_0.5 (%) | mAP_0.5:0.95 (%) |
|---|---|---|---|
| 99.13 | 98.59 | 99.45 | 88.20 |

### ❖ Performance of Algorithm

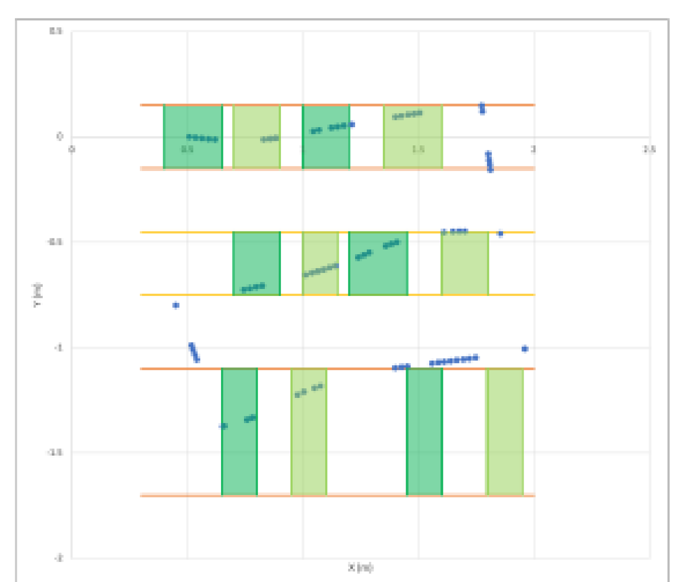
- Table 2 evaluates the mapping results according to each fruit tree number. There are a total of 16 fruit trees, and there are 4 rows: 1 to 4, 5 to 8, 9 to 12, and 13 to 16. Results from No. 1 to No. 8, detected at the beginning of the robot's movement, show high precision and recall, but results from No. 9 to No. 16, detected at the end of the robot's movement, show low recall because it often miss fruit detection.

Table 2. Performance evaluation of the Algorithm using the model

| Range | Index | Fruit | sickFruit | Total |
|---|---|---|---|---|
| Whole | Precision (%) | 97.06 | 81.25 | 89.39 |
| | Recall (%) | 51.56 | 86.67 | 62.77 |
| 1 ~ 8 | Precision (%) | 95.83 | 84.62 | 90.00 |
| | Recall (%) | 95.83 | 91.67 | 93.75 |
| 9 ~ 16 | Precision (%) | 100.0 | 66.67 | 87.50 |
| | Recall (%) | 25.00 | 66.67 | 30.43 |

### ❖ Discussion

- The robot's odometry information was used to determine the fruit tree number as shown in Figure 7, but a slight error occurred every time it was driven, and the error was accumulated while driving, so it often miss fruit detection at the end of driving.
- It is expected that accuracy can be further improved if a large amount of odometry data is collected and clustered while the robot is repeatedly driven and the standard for judging the number of fruit trees is set.



< Figure 7. Position Mapping>

## Conclusions

- Applicating vision sensors, a ROS-based robot platform capable of monitoring fruit growth while autonomously driving in real time was implemented.

## References

[1] Yan, Bin, et al. "A real-time apple targets detection method for picking robot based on improved YOLOv5." Remote Sensing13.9 (2021): 1619.

SEOUL NATIONAL UNIVERSITY
BIOSYSTEMS CONTROL & PRECISION AGRICULTURE LABORATORY

Keit Korea Evaluation Institute of Industrial Technology

※ This work was supported by the Technology Innovation Program (the project 20010493 for development of open integrated control system for level 3 autonomous tractors) funded by the Korea Evaluation Institute of Industrial Technology (KEIT, Korea)