An Automatic System for Crop Monitoring and Culture Based on IoRT

Tran Dang Khoa Phan, Van Thanh Vu*

Abstract—The development of automated farming systems in precision agriculture has been attracting increasing research interest. In this paper, we present an automatic system for crop culture and monitoring based on the Internet of Robotic Things (IoRT). The system includes: an agricultural farming robot with built-in object detection block based on deep learning to locate and classify plant and weed in the image; a device that monitors the parameters of the soil and air environment. The information from the robot and the monitoring device is combined to help the robot determine the right way to take care of the plants. Technical solutions for each component of the system are fully proposed, implemented and evaluated. The experimental results show that the robot is capable of detecting objects with high accuracy; monitoring equipment operates stably; suitable robot design, allowing precise movement of the actuator to any position in the field of view.

Index Terms—IoRT, Deep Learning, Object Detection, YOLOv3, Precision Farming.

1. Introduction

One of the goals of sustainable agricultural development is to increase the yield of crops but at the same time reduce dependence on pesticides and herbicides. Precision agriculture seeks techniques: to monitor the parameters of the soil and air environment of the cultivated area; and to monitor the necessary indicators to assess the growth of the crop. From there, the necessary care measures are applied precisely to each plant object to reduce the amount of chemicals used. Along with the development of agricultural robots, automation in agricultural farming is getting closer to reality.

In this paper, we study the automatic crop monitoring and culture system. The core part of this system is the object detection block, which locates objects of a certain class in the image. To date, most of these systems have relied on image analysis to detect plants and weeds. Bawden et al. proposed a weeding robot, in which classical image processing techniques such as color-based image segmentation, Local Binary Pattern (LBP) are combined with machine learning (ML) model to detect plants and weeds in images [1]. Different types of image features and classifiers have been combined to improve the effectiveness of object detection [2]–[4]. However, the classical ML approach does not achieve high accuracy because the complexity of handcraft features and ML models is not enough to allow the model to generalize the data.

Recently, the rapid development of deep learning (DL) has greatly improved many problems such as image classification, object detection, etc. Convolutional neural networks have been applied to solve problems in the field of precision agriculture. Encoder-decoder model has been proposed to segment images containing weeds [5]. The RGB color image and the near-infrared image were combined and fed into the model. In [6], [7], the authors applied U-Net to segment an image into plants and weeds. MobileNets [8] was used to increase the inference time to 10 FPS on Raspberry Pi.

In this paper, we present an automatic crop monitoring and care system based on the Internet of Robotic Things (IoRT). The term IoRT refers to intelligent robotic devices that can monitor events; aggregate sensor data from various sources; use local and distributed artificial intelligence (AI) to determine the best measure; and then act to manipulate physical objects [9]. The goal of the system is to build a robot with built-in blocks that detects plants and weeds in images obtained from the camera mounted on the robot. Based on the information about the class and coordinates of the objects, the robot performs care operations corresponding to each type of object. In addition, the system is also integrated with equipment to monitor the parameters of the soil and air environment to help the system to make decision on care measure. The main contribution of this study is to propose a complete system for monitoring and caring crops based on IoRT. Technical solutions for each block are proposed, implemented and evaluated completely.

The rest of the paper is organized as follows. In Section 2, we present the proposed system and the detailed design of blocks. The experimental results and evaluation of the proposed system are described in Section 3.

Tran Dang Khoa Phan and Van Thanh Vu are with Faculty of Electronics and Telecommunication Engineering, The University of Danang - University of Science and Technology, Vietnam (E-mail: ptdkhoa@dut.udn.vn, vvthanh@dut.udn.vn).

**Corresponding author: Van Thanh Vu (E-mail: vvthanh@dut.udn.vn)*
Manuscript received October 11, 2022; revised November 19, 2022; accepted December 18, 2022.
Digital Object Identifier 10.31130/jst-ud.2022.504

2. Proposed system

2.1. The structure

The proposed system consists of four main blocks: the sensor station, the crop detection, the farming robot, and the remote monitoring and control (Fig. 1).

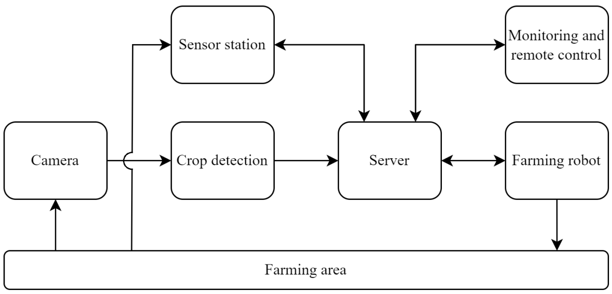


Fig. 1: The structure of the proposed system.

The sensor station collects the parameters of the plant's growth environment including temperature, fertilizer content, humidity, ... These data is sent to the database periodically and used for making decision on which farming operation should be applied.

The crop detection block uses a camera to capture images of the working space. A model based on deep learning is developed to detect and locate objects (crops and weeds) in images. The model is deployed on an embedded computer. The output of the model is a data series including coordinates of the object and the class of detected objects. That data string is sent to a storage server through Internet.

The information of objects' coordinates and classes is received by the farming robot via the storage server. This block controls a robot frame to move an actuator to perform farming operations at positions of objects. Precise care measure for each class improves crop yields, while reducing energy consumption and water usage.

The remote monitoring and control block uses a cloud computing service to store the data of the sensor station block. Besides, it allows users to monitor the parameters of the farming system and control the system remotely via an Android application and a website.

2.2. Hardware and software design

In this section, we present the hardware and software design for the system proposed in Section 2.1.

2.2.1. Sensor station

The structure of the sensor station is presented in Fig. 2. The Wi-Fi microcontroller ESP8266 is used to read the data from the sensors and periodically transmit it to the database via Wi-Fi. The power for all sensors is supplied from solar energy and battery storage.

The algorithm diagram for the sensor station block is shown in Fig. 3. First, the interfaces (UART, I2C), timer and connections (Wi-Fi, Firebase) are initialized. Every minute data is collected from sensors and is updated to a database. According to research [10], compared with other cloud computing services such as cloud MQTT, FRD, Firebase has outstanding advantages, such as data transmission rate, number of connections, data storage method, service cost. Thus, we choose Firebase Realtime Database service to store and synchronize data with the system.

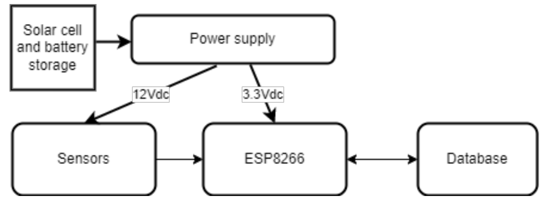


Fig. 2: The structure of the sensor station.

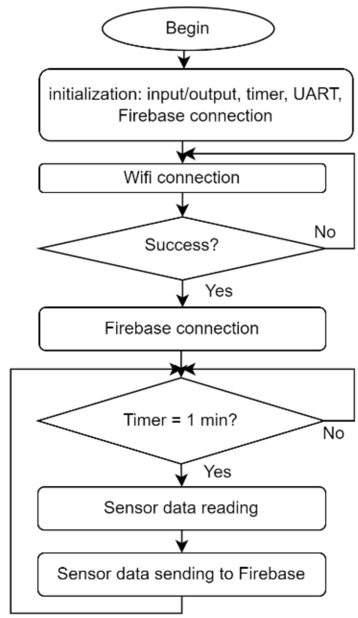


Fig. 3: The algorithm diagram for the sensor station.

2.2.2. Crop detection

The crop detection block has the function of locating and classifying plants and weeds in images. The input to the block is a color image; and the output is the coordinates of the bounding box surrounding the object and its classification. Because we plan to implement this block on an embedded computer (particularly, a Raspberry Pi 4), we choose the model based on the compromise between the accuracy and the processing speed. From the comparative results of the study [11], we decide to choose the YOLOv3 model. The overall structure of YOLOv3 is shown in Fig. 4.

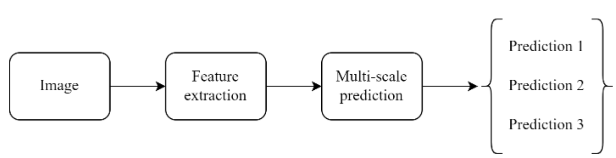


Fig. 4: The structure of YOLOv3.

YOLOv3 is based on a CNN architecture. The input image is first passed through a feature extraction block.

The Darknet-53 architecture is used to extract feature maps. Then, the features are fed into a multi-resolution prediction block. YOLOv3 uses 3 prediction stages with different resolutions. The output of each prediction stage is the position of the bounding boxes around the objects and their classes. This technique makes the model capable of detecting objects appearing in images at many different resolutions. Skip connections are used to integrate low-level features into prediction stages to increase prediction efficiency. Post-processing techniques such as K-means, Non-Maximum Suppression are used to select the best prediction results.

After training and testing, the model is deployed on Raspberry Pi 4, which is an embedded computer. The algorithm diagram of the crop detection block is shown in Fig. 5. First, the camera captures images of the working space. A captured image is divided into 4 equal parts. Each part of the image is processed sequentially in order to reduce the computing complexity. When a command is received from Firebase, images are in turn passed to the YOLOv3 model to perform object detection. The object's class and coordinates are updated to Firebase.

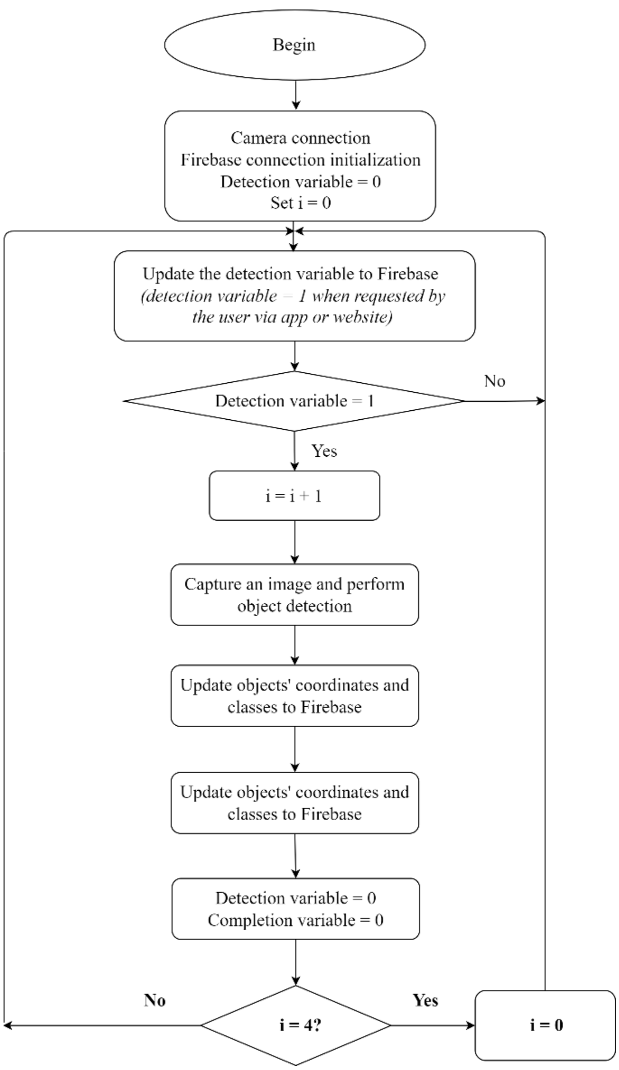


Fig. 5: The algorithm diagram of the crop detection block.

2.2.3. Farming robot

The design of the farming robot consists of the robot frame and the control circuit. The design of the robot frame is shown in Fig. 6. The basic specifications are as follows: dimensions - $0.7m \times 1.6m \times 0.81m$; distance from the ground - $0.23m$. The robot requires 3 stepper motors for moving along 3 axes. Since the Y but with the long Y axis, it needs to be stable while movi axis is long, we use 2 stepper motors in parallel to make the motion along this Y axis stable and accurate. The robot also has 3 position switches for each axis to determine the limit of each axis, avoiding the situation that the motor still rotates when it goes to the end of its path. The sprinkler actuator is capable of moving along the X and Z axes.

Fig. 6: The robot frame.

According to the requirement of the load, we choose the stepper motor 57HS56 with the following basic parameters: voltage - $4.42V$; maximum load current - $3A$, 2 phase. Accordingly, we select the control circuit TB6600 with the following parameters: voltage - $9 \div 42V$; maximum load current - $4A$; optically isolated and high-speed input; built-in over-current and over-voltage protection; micro-stepping modes - $1/2, 1/4, 1/8$ and $1/16$. The microcontroller ATMega2560 is chosen to control the stepper motors thanks to the number of pins for digital and analog communication and stable operation. Regarding data collection from the server, the ESP8266 is used as the Wi-Fi microcontroller.

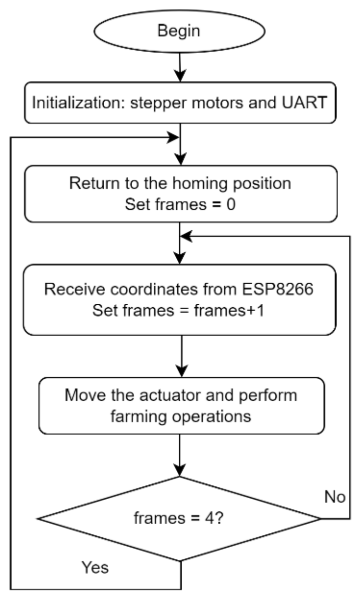


Fig. 7: The algorithm diagram for robot control.

The algorithm diagram for robot control is shown in Fig. 7. The ATMega2560 microcontroller initializes the communication with the ESP8266 (UART) and sets the initial values for the stepper motor control circuit, position switches, and sensors. Then, the actuator is moved to the home position, waiting for the command from the ESP8266. Depending on the command received, the microcontroller executes the required modes from the ESP8266. Upon completion, the actuator is returned to the home position, waiting for commands from the ESP8266.

When receiving a command for farming operation, the robot updates the image coordinates of plants and weeds. The system calculates the actual coordinates in the working space. The actuator is moved to each position according to these coordinates. At each location of crops, the robot measures soil moisture and water crops if the humidity is below the threshold. For weeds, the robot sprays herbicide.

The algorithm diagram for receiving and sending data to Firebase is shown in Fig. 8. The ESP8266 connects to Wi-Fi and initializes the setting for Firebase connection. Then, the ESP8266 reads the variables of operating modes in turn. At the same time, the ESP8266 sends a command to the ATMega2560 to perform the requested mode.

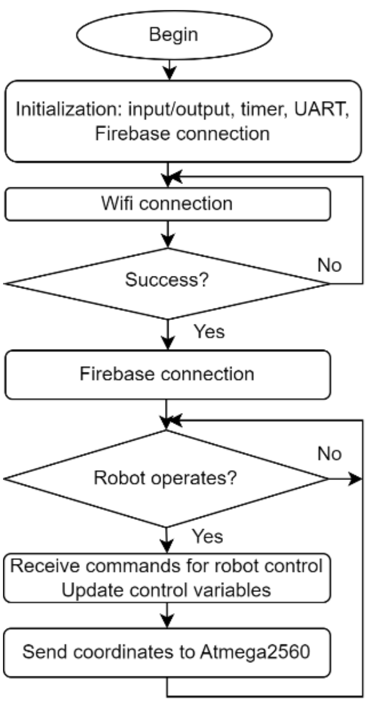


Fig. 8: The algorithm diagram for receiving and sending data to Firebase.

2.2.4. Remote monitoring and control

In order to remotely monitor and control the system, we design an user interface via a website and an Android application. The website and the mobile application have a similar design, including 3 pages (tabs) with the functions described as the diagram below (Fig. 9).

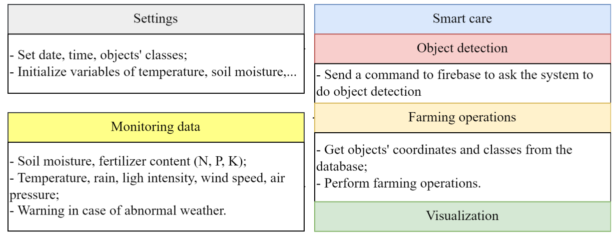
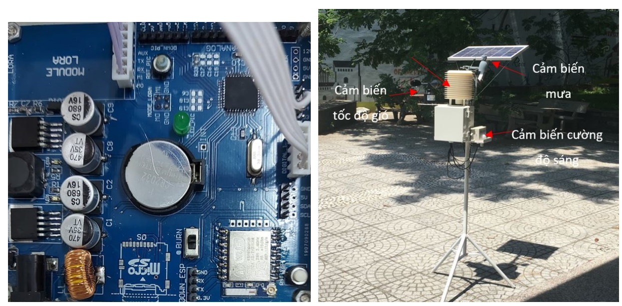


Fig. 9: The structure of the website and the Android application.

3. Experimental results

3.1. Sensor station

The main circuit board of the sensor station block is designed in double-sided printed circuit with a compact size of $6cm \times 5cm$ (Fig. 10a). The block is equipped with sensors according to the industry standards ISO $10012-1$ and ISO $10012-2$. In addition, a solar panel with a capacity of $35W$ and a battery station of $12.6V/7.8Ah$ is used to maintain operating power for this block. All devices are integrated on a single rack to ensure compactness and ease of installation. The cubic volume of the entire sensor station block is about $24cm \times 18cm \times 12cm$. The construction results of sensor station block are shown in Fig. 10b.



(a) (b)

Fig. 10: The sensor station: (a) The main circuit board; (b) Installation on a rack.

To evaluate the performance of the sensor station block, we have monitored the data acquisition and transmission process of this block within 10 days in the condition of stable Wi-Fi condition and outdoor environment. Fig. 11 shows the data updated from Firebase and displayed on the website. Monitoring results show that the data is continuously updated and there are no interruptions or data errors during the transmission.

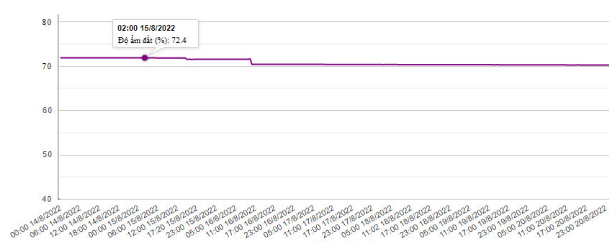


Fig. 11: The data of soil moisture updated from Firebase.

3.2. Farming robot

The construction results of the farming robot are shown in Fig. 12. Fig. 12a is the robot made of industrial aluminum frame, firmly coupled, ensuring stability for the movement of the actuator. The sprinkler actuator is shown in Fig. 12b. This actuator, which is controlled by stepper motors, is connected to a water pump motor and a soil moisture sensor.

(a) (b)

Fig. 12: The farming robot: (a) The robot frame; (b) The sprinkler actuator.

We evaluate the farming robot block by the accuracy of the actuator control. Positions with known coordinates are evenly distributed on the working plane. Then, the control parameters are calculated to move the actuator to these positions. For each position, the actuator is moved to the position and returned to the home position. This process was performed 50 times for each position to evaluate errors. For each experiment, the Euclidean error between the ground-truth position and the actual position is recorded. Statistical results show that the error fluctuates in the range of $4 \div 8mm$ and does not change significantly with distance (within the working plane). The position switches allow the system to eliminate the accumulated error.

3.3. Crop detection

The crop detection block is evaluated through training and testing on our database. We collected a database, including images of lettuces and weeds from vegetable farms in Lien Chieu district (Da Nang). Crops were recorded at different growth stages. The total number of images is $5,038$. We labeled images manually. Each object in the image is assigned a bounding box and a class. Illustration of the labeling process is shown in Fig. 13.

According to the opinions obtained from the farmers, lettuces at different growth stages have different care methods. Therefore, we divided lettuces into 2 classes: large lettuces and small lettuces. Meanwhile, different types of weeds are gathered into a single layer. The total number of labels is $51,375$. The number of labels for the large lettuces, small lettuces, and weeds are $21,833$,$16,366$ and $13,176$, respectively. The split ratio between the training set and the test set is $80 \div 20$.

To evaluate the model, we use the following criteria: AP (Average Precision), mAP (Mean Average Precision), and IoU (Intersection over Union) [11]. The model is implemented by Python. The training parameters are

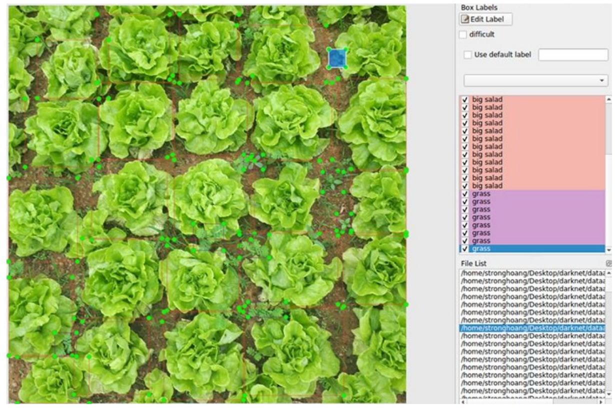


Fig. 13: Illustration of the labeling process.

set as follows: batch size – 64; number of loops – $30,000$; initialization coefficient – 0.001.

The change of the loss function and mAP with the number of iterations for the training set is shown in Fig. 14. It can be seen that the loss function converges after about $20,000$ iterations. However, we trained the model for $30,000$ iterations to ensure the model completely converges. In the results, the mAP value converged to the value of about 79%.

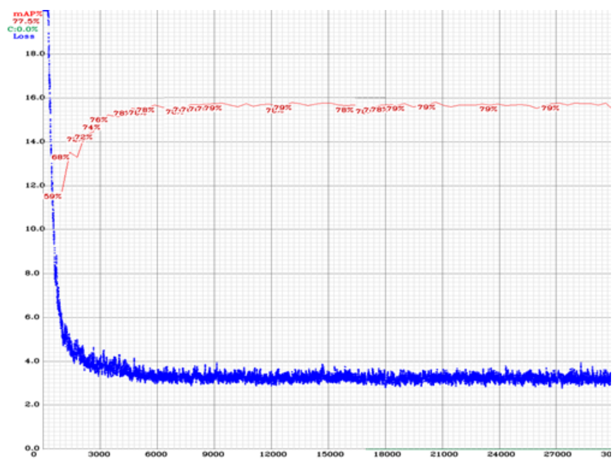


Fig. 14: The dependence of the loss function and mAP on the number of iterations.

The AP results of each class for the test set are shown in Table 1. It can be seen that the APs for large lettuces and small lettuces are high, while the AP for weeds is much lower. The possible causes are: (i) the number of samples of the lettuce classes is majority, so the model tends to detect this object to minimize the loss function; (ii) compared with lettuces, weeds are diverse in shape and color, so it would be more difficult to detect weeds.

TABLE 1: The AP results for the test set.

Classes	Large lettuce	Small lettuce	Weeds
AP	91.16%	89.31%	61.47%

For the test set, the model achieved the mAP result of 80.49%. Knowing that, the mAP results of the YOLOv3 model in the original study [11] ranged from 51.5% to 57.9%. Accordingly, the obtained results are positive. In addition, the average IoU result reached 0.64. The illustrative results for an image are shown in Fig. 15a. The information of objects' coordinates and classes are updated to Firebase (Fig. 15b).

(a) (b)

Fig. 15: (a) Illustrative results for object detection; (b) Coordinates and classes of detected objects updated to Firebase.

Since the data is unbalanced between classes, we plan to improve the training and testing results of the model by the following ways: (i) increase the number of the minority class (weeds) to achieve the balance between classes; (ii) adjust the loss function to assign greater weight to the minority class samples.

3.4. Remote monitoring and control

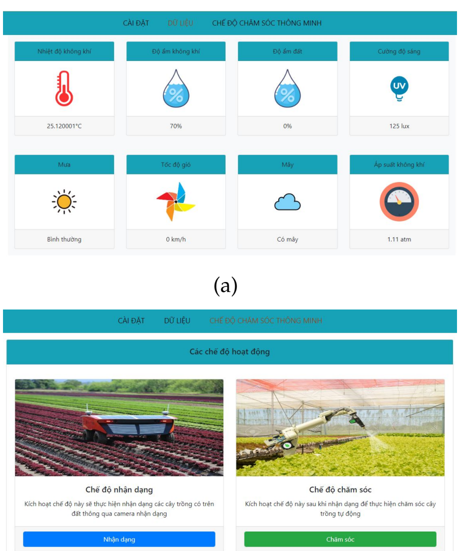
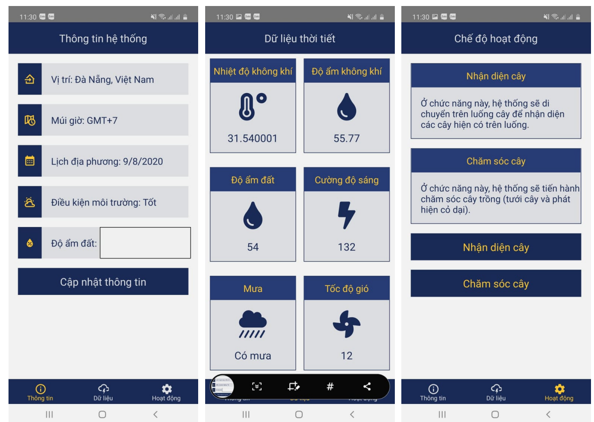


Fig. 16: Website interface: (a) Smart care mode; (b) Data monitoring mode.

The interface of the website is shown in Fig. 16. The website allows users to choose the following modes: smart care, data monitoring, and setting. For the smart care mode, the user can select the crop detection function, which is usually done first with the new crop tray. After that, the system will periodically work to identify the objects on the planting tray. The information of objects' location and classes is updated to Firebase. The robot relies on the data of object detection from Firebase to perform farming operations. For the data monitoring mode, the user can monitor the parameters collected from the sensor station. In the case that the measured values exceed the given thresholds, the effects will appear for warning.

The interface of the mobile application is similar to that of the website (Fig. 17). The system information page provides location, time and environmental conditions updated from Firebase. It also allows users to set parameters such as humidity thresholds.



(a) (b) (c)

Fig. 17: Application interface: (a) System information; (b) Data monitoring mode; (c) Smart care mode.

4. Conclusion

In this paper, we have presented an automatic crop care and monitoring system based on IoRT platform. The system includes: a robot frame with integrated object detection block based on the YOLOv3 model; and a soil and air environment monitoring device. Hardware and software design for each component of the system was fully proposed, implemented and evaluated. The experimental results showed that the system is capable of correct operation and can be deployed in practice. In the future, we will develop localization and navigation blocks for the robot.

Acknowledgement

This research is funded by Funds for Science and Technology Development of the University of Danang under project number B2020-DN02-79.

References

[1] O. Bawden, J. Kulk, R. Russell, C. McCool, A. English, F. Dayoub, C. Lehnert, and T. Perez, "Robot for weed species plant-specific management," *Journal of Field Robotics*, vol. 34, no. 6, pp. 1179–1199, 2017.

[2] J. Adams, Y. Qiu, Y. Xu, and J. C. Schnable, "Plant segmentation by supervised machine learning methods," *The Plant Phenome Journal*, vol. 3, no. 1, p. e20001, 2020.

[3] S. Haug, A. Michaels, P. Biber, and J. Ostermann, "Plant classification system for crop/weed discrimination without segmentation," in *IEEE winter conference on applications of computer vision*. IEEE, 2014, pp. 1142–1149.

[4] P. Lottes, M. Hoeferlin, S. Sander, M. Müter, P. Schulze, and L. C. Stachniss, "An effective classification system for separating sugar beets and weeds for precision farming applications," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 5157–5163.

[5] I. Sa, Z. Chen, M. Popović, R. Khanna, F. Liebisch, J. Nieto, and R. Siegwart, "weednet: Dense semantic weed classification using multispectral images and mav for smart farming," *IEEE robotics and automation letters*, vol. 3, no. 1, pp. 588–595, 2017.

[6] Ł. Chechliński, B. Siemiątkowska, and M. Majewski, "A system for weeds and crops identification—reaching over 10 fps on raspberry pi with the usage of mobilenets, densenet and custom modifications," *Sensors*, vol. 19, no. 17, p. 3787, 2019.

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[9] O. Vermesan, R. Bahr, M. Ottella, M. Serrano, T. Karlsen, T. Wahlstrøm, H. E. Sand, M. Ashwathnarayan, and M. T. Gamba, "Internet of robotic things intelligent connectivity and platforms," *Frontiers in Robotics and AI*, vol. 7, p. 104, 2020.

[10] T. Van Vu, N. N. Huong, and P. T. D. Khoa, "A survey of communication protocols and cloud services for iot systems," *Journal of Science and Technology: Issue on Information and Communications Technology*, vol. 17, no. 12.2, pp. 7–12, 2019.

[11] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

Tran Dang Khoa Phan received the B.E., M.S.E., and Ph.D. degrees from Tula State University (Russia) in 2008, 2010 and 2015, respectively. He is currently working at The University of Danang - University of Science and Technology (Vietnam). His current research interests include digital image processing, computer vision, and machine learning.

Van Thanh Vu received the B.E. and M.S.E degrees from The University of Danang - University of Science and Technology (Vietnam) in 2008 and 2020, respectively. He is currently working at The University of Danang - University of Science and Technology (Vietnam). His research topics includes IoTs, embedded systems, robotic.