Autonomous Cloud Robotic System for Smart Agriculture

Tharindu Dharmasena
Dept. Of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
tharindu.d@sliit.lk

Rajitha de Silva
Dept. Of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
rajitha.d@sliit.lk

Nimsiri Abhayasingha
Dept. Of Electrical and Electronic Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
nimsiri.a@sliit.lk

Pradeep Abeygunawardhana
Dept. Of Computer Systems Engineering
Sri Lanka Institute of Information Technology
Malabe, Sri Lanka
pradeep.a@sliit.lk

***Abstract*—Agriculture sector occupies 25.9% of the world employment. The demand for food production is rapidly increasing with the increase of world population. Developing the existing agricultural infrastructure by incorporating modern technologies will help to match this increasing demand. This paper proposes a automated system to optimally control the climate and irrigation in a greenhouse by monitoring temperature, soil moisture, humidity and pH through a cloud connected mobile robot which can detect the unhealthy plants using image processing. A fuzzy controller will control the heating and cooling system, irrigation system and humidifiers installed in the greenhouse based on the sensor readings. The mobile robot navigates through a predefined map of the greenhouse and collect soil samples to perform measurements while onboard sensors will collect the ambient climate data. A camera mounted on the mobile robot will capture the plant and detect unhealthy crops based on the colour and the texture of the leaves.**

***Index Terms*—Cloud Robotics, Smart Agriculture, Internet of Things, Monitoring, Automation**

I. Introduction

Agriculture is an important and necessary process to sustain the human kind. The techniques are used for agriculture has been developed from ancient time. During industrial revolution in 18th century , there is an huge development in agricultural industry along with other industries.

Automation has not efficiently penetrated into greenhouse farming. It is clear that introducing automation into greenhouses and robots into the farming sector will be economical and it will reduce the work load of the farmer. It will also increase the productivity and reduce the amount of human intervention in farming [1].

Greenhouses provide a good and safe environment for the growth of plants, as it protects the plants form harsh weather conditions and most of the external pests. It allows the plants to grow under optimal conditions, which maximizes the growth potential of the plants. Since the water and heat can be retained within a greenhouse, the plants have all necessary conditions to grow all year round [2], [3]. A greenhouse allows the growers to produce plants in places where climate would otherwise be unfeasible for the growing of plants. The production of crop plants is independent of the geographic location and time of the year. A direct example for this situation would be that the climate in Colombo,Sri Lanka is not suitable for the growth of strawberries.

This paper presents an image processing based disease detection system implemented with a mobile robot. The robot has various other measurement capabilities to deliver a wide range of capabilities. This paper only focus on the image processing based disease detection component of the robotic system.

II. Related Work

Implementing image processing techniques to detect plant diseases was widely adopted by many researchers conducting research on image processing and machine learning algorithms [4]–[6]. Researchers in [7] has developed a cloud robotics architecture based on ROS which features separate subsystems to sensor manager, robot manager and actuation manager.

Multiple researches supports the idea of plant disease detection based on the pictures of the respective plant leaves. Surface texture of the leaves changes due to diseases and this texture is unique to each disase [8], [9]. Some researches indicate that the K-mean clustering algorithm proven to be efficient in disease detection from plant leaves [10].

There were several researches which developed robotic systems to data acquisition and controlling of farming within greenhouses [11]–[13]. Some of them uses ground robots and some of them uses aerial robots. It has to be noted that the mobile robots could be used with greenhouse automation infrastructure instead of aerial robots. There are systems which have been developed to control greenhouse micro-climate depending on the ambient climate sensor information [14].

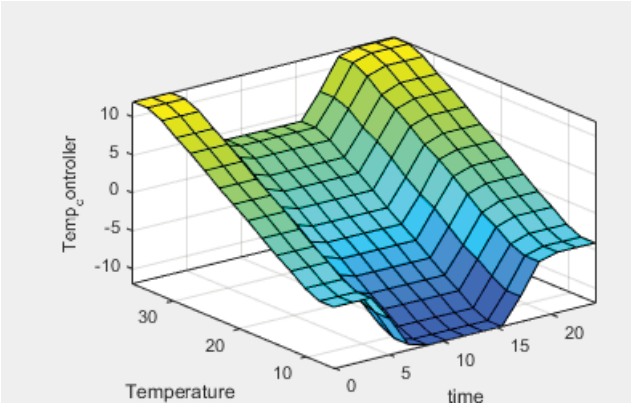

Fig. 1: Response surface for input output relation from fuzzy controlled temperature mechanism

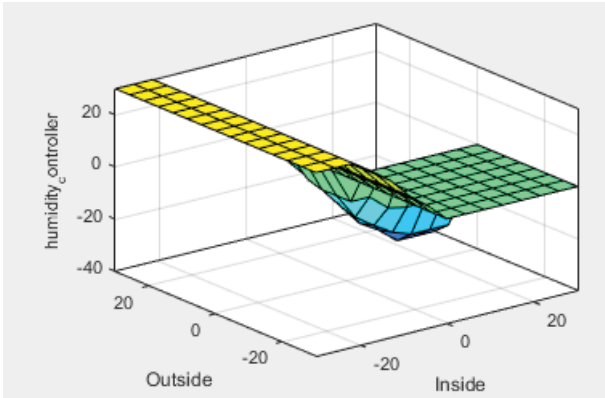

Fig. 2: Response surface for input output relation from fuzzy controlled humidity mechanism

III. Methodology

The cloud robotic platform is centered around its main robotic agent which houses multiple sensors: Humidity and temperature, soil moisture, soil pH and a camera module. The robotic agent is operated by a RaspberryPi with ROS platform for its navigation, image capturing and sensor data processing. The sensors are connected to a NodeMCU board to collect data and transfer to the online database. The robotic agent will directly communicate to the control system of the greenhouse to control lighting and water supply for optimum growth of plants. The control system of the greenhouse is not discussed in detail within this research assuming the greenhouses already have electronically controlled actuators for watering and lighting. The proposed system is designed to be easily interfaced with the existing greenhouse control infrastructure. Robotic agent transfers the captured images to be processed to a local server which implements the disease identification subsystem of the proposed cloud platform. The robot will obtain a soil sample from the respective planter and run a pH test on the obtained sample when a diseased plant is identified. The system will then inform the farmer about the diseased crop and it will ensure the plant is supplied with necessary watering and lighting to recover from the infection.

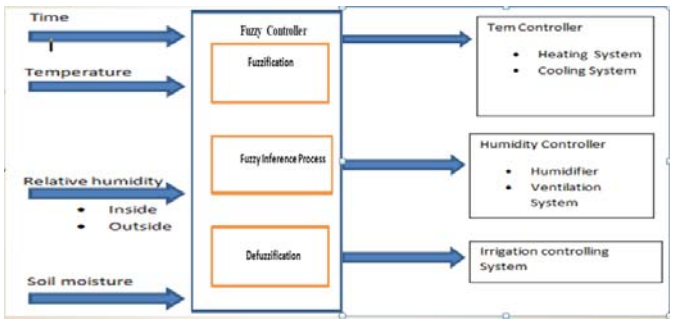

Fig. 3: Fuzzy Controlling System

A. Cloud Platform

Database for the cloud storage was created using MYSQL. This database was created at the backend of the website. This cloud storage was used to store all the climate data of each and every greenhouse (when this system is implemented in many greenhouses) [15]. Climate data includes humidity, temperature, soil moisture and pH. A table in the online database is allocated for each and every user (each and every greenhouse) for their own greenhouse climate data. Moreover a graphical representation was created using JavaScript, for each and every user table in the online cloud storage. To directly insert sensor data to the cloud storage, a connection is made between the sensors and the online cloud storage using the NodeMCU by connecting it directly through Wi-Fi. At the database side a PHP code was used to make the connection with the database through an URL. Using DHT 22 (Digital) sensor temperature and humidity data were collected. DHT 22 was interfaced with the Node MCU to sense surrounding temperature and relative humidity. Soil moisture data was obtained using Soil Humidity Hygrometer Moisture Detection Sensor YL-69. These sensor data were uploaded to the cloud automatically by coding the Node MCU board to connect to the Wi-Fi and generating the URL of the online database [16]–[18].

B. Robotic Communication Network

In order to maintain optimum values in instances where the climate values differ from them, a communication was created between robotic agents and respective actuators. Thus, in order to create a robotic network, each robotic agent consisted of a Node MCU which was used create the communication platform, between the robotic agents that collects data and other respective robotic agents which act as actuators. Using MQTT (Message Queuing Telemetry Transport) the communication between Node MCUs was established. MQTT is an ISO standard publish-subscribe-based messaging protocol. It works on top of the TCP/IP protocol. Online broker acts as the intermediate, and the messages will be published (by robotic agents) for a particular topic and the robotic agents and actuators subscribed to the relevant topic will receive the messages, thereby communication between robotic networks is established.

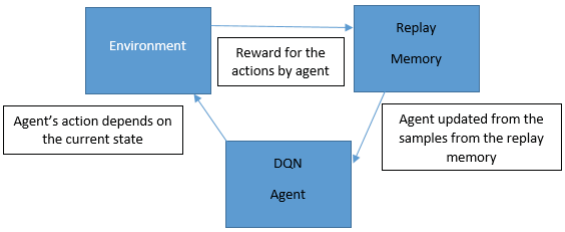

Fig. 4: Cycle of an episode in the learning process

Using Node Red an interface was designed to force control greenhouse robots and view greenhouse data in a clearer manner. Using MQTT sensors and actuators were connected to the Node Red interface. Node Red nodes are subscribed to the appropriate topics in which the data is published.

C. Mechanism to control the Actuators

Conventional greenhouses uses human expertise to maintain the crops and its growth. It is essential to implement a sophisticated controlling mechanism which will provide a reliable, power conserving and autonomous control system for the greenhouse. Artificial Intelligence (AI) is one of the most common approaches to fulfill the current requirement of implementing an efficient control system. Fuzzy Inference system was implemented to fulfill the decision making process as it is the most suitable application of logic to control actuators of the green house. The fuzzy controller developed within the project could be interfaced with control system of the greenhouse to control watering and lighting.

D. Method of Navigation

Machine learning was used to avoid static and moving obstacles and also to navigate at bends in the greenhouse path. The type of machine learning used was Reinforcement learning. It is an algorithm where the computer learns, based on the compensation for actions performed in an environment, with Deep Q- Learning (DQN) [19].

During the robot navigation, robot should move to the given goal by using the current state, information from the environment, goal position, and set of predefined actions. Predefined actions and their respective velocity parameters for ROS velocity topic are listed in Tabele I. Navigation agent was constructed using the deep Q network known as DQN.

$$Q_{t+1} = Q_t + \alpha d_{tgoal} + \gamma \cos\theta_t + \beta d_{lid} \quad (1)$$

$$\delta Q_t = Q_t - Q_{t-1} \quad (2)$$

Q_{t+1} is the Next state of the robot and the Q_t is the current state of the robot. α and β are Discount rate and learning rate of the algorithm.R_t d_{lid} and $d_{tgoat}$are the Rewards, reading lidar sensor, and distance to goal at present respectively.

The robotic agent, decides to take an action from the predefined available actions (forward, turn left, turn right, backwards), and executes it in the environment. The result of that action, makes the agent closer or not to its goal (to avoid static obstacles and take sharp turns). If the robot is getting closer to achieve the goal then it gets a good reward. If it knocks at obstacles it gets a bad reward. In any case, the agent perceives the current state of itself and the environment (where it is located now), and then feeds reward, previous state, new state and action taken to the learning algorithm (to learn the results based on its actions). Then the process repeats again for the number of steps the robot is allowed to experiment. When the number of steps is done, the final reward is obtained and the robot starts again from the initial position, now with an improved algorithm (with the previous state and reward information the robot has learned). The whole process is performed repeatedly for a given number of episodes (usually high). Where an episode is just one cycle (previous state, the action and then the new state) in the training process [20].

TABLE I: Actions used in the training algorithm

Type of Action	Speed value assigned
Forward	0.5
Backward	0.5
Turn Left	0.3 angular speed
Turn Right	0.3 angular speed

TABLE II: Important Parameters

Parameters	Used value	Use of the parameter
Epsilon	0.9	The probability of choosing a random action
Epsilon_discount	0.999	Represents how much future events lose their value according to how far away.
Number of Episodes	500	Number of times the training is done
Number of steps	500	Maximum number of steps used per episode

The three main algorithms running are environment, training and learning algorithms.Environment algorithm –used for the interaction of the robot with the environment performing the actions (forward, turn left, turn right, backwards). When launching the entire training program, the code was declared as a node of ROS, therefore all the actions were performed using ROS packages. Environment algorithm provides information such as rewards and the new state for the training algorithm, which was used in the learning process.

Training algorithm –Looped to run the code for the number of episodes (a predefined value) and each episode runs for a predefined number of steps or until the robot hits an obstacle in the environment. It interacts with the environment algorithm and learning algorithm for each and every episode to perform necessary actions in the considered environment with respect to the previous rewards, actions and states.

Learning algorithm –The whole process where the robot was made to learn is based on the obtained results at each and every previous episode, by using Q-learning. Based on the current state the learning algorithm chose an action to be performed by the robot in the environment. More importantly

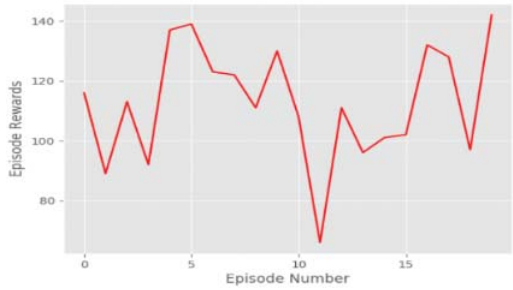

Fig. 5: Training results obtained for 20 episodes of training using Gazebo predetermined simulation in ROS development studio

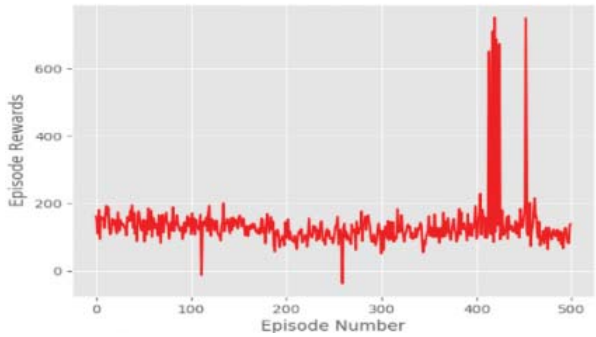

Fig. 6: Training results obtained for 500 episodes in Gazebo at ROS development

based on state, action, reward and next state the robot was made to learn from the previous results.

E. Image processing for disease identification

The proposed cloud robotic platform has the capability of disease detection using image processing techniques. A camera mounted to the mobile robot will capture the images of plant leaves and send them to a server computer for further processing as indicated in 7. The plants within the operating greenhouses are found to be healthy, since the existing greenhouses are maintained by the owners with necessary care. To this end, this system was tested with images from the eggplant leave images obtained from outdoor farms from home gardens. The images obtained to test the disease identification subsystem were manually captured by the researchers because the robotic agent was designed to operate in a greenhouse environment.

Bacterial wilt, southern blight and several other diseases which affects eggplants could be identified with the visual abnormalities of its leaves [21]. The type of visual abnormality is dependent on the disease [22]. After acquisition of the images from the leaves of the plants, the prepossessing task was conducted to remove the noise of the images. This is mainly done by using median filter. Then the segmentation process was conducted to extract the area of the leaf that has been affected by the disease. Generally the area of the healthy part of the leaf consists of green colored pixels. Therefore the R G B components of green colored pixels should be masked and removed (figure 8). After the preprocessing stage, it is required to define the color space and convert the RGB image

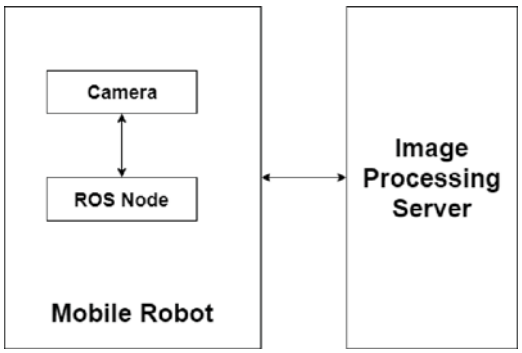

Fig. 7: Image Processing Subsystem Architecture

Fig. 8: Image Pre-Processing Stages

to HSV (Hue, Saturation, Value) color space. This is done because the RGB is a device dependent color space. Upon analyzing the histograms of each of the components of the HSV image, a common threshold was found, and by using that threshold the green pixels were masked and removed. Then the disease prediction is conducted using thresholding (figure 9) method and K-mean clustering method (figure 10).

After the segmentation process it is necessary to extract the features of each cluster for further processing. Color co-occurrence method was used for texture analysis. The texture features such as contrast, energy, homogeneity and correlation were extracted and computed, to classify the image for disease identification.

IV. RESUTLS AND DISCUSSION

A. Fuzzy Interference System

The designing of the fuzzy interference system for controlling actuators was carried out using the MATLAB simulation tool as shown in figure 11. The software developed for the proposed work was tested under different input conditions and the results obtained were good in terms of accuracy. After analyzing the Simulink results, it can be concluded that the design of the fuzzy system behaves according to the defined rules and conditions. Furthermore, the designed fuzzy interference system was used to fulfil the controlling requirements of the actuators that are used in the greenhouse. As a future development, this fuzzy interference system can be converted to a neuro-fuzzy system in order to improve the productivity of the control system of the green house.

The output results that were obtained from MATLAB Simulink and Node MCU are listed in Table III.

B. Plant disease identification through Image Processing

As the first step the implemented algorithm was trained and tested, by using a data set of four classes(four targets values which has three diseases and healthy leaves).They are

Fig. 9: Thresholding Method

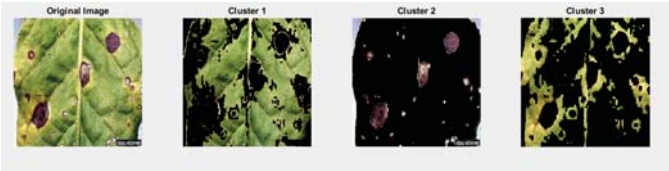

Fig. 10: K-Mean Clustering Method

Anthracnose, Bacterial Blight, Frog eye disease and healthy leaves. All together the total data set consists of 150 images. In here since it has been used four machine learning classification algorithms, the results obtained from each classification algorithm are shown in figure 12 and figure 13.

After observing the classification results of each algorithm as shown in figures 12 and 13 it is clearly identified, although training accuracy has some highest values the testing accuracy is poor. Also the cross-validation accuracy has some poor results. This can be occurred due to over fitting. Since Decision Trees, k-Nearest Neighbors and Support Vector Machines are low bias and high variance algorithms the corresponding models have high probability of being over fitting. Generally a machine learning model to be obtained high prediction accuracy, the model should be low bias and low variance (There is a trade of between bias and variance). Therefore if a model has high variance (over fitting case) one of method to improve the accuracy is increase the number of samples. Therefore the sample size increases to 200 and the following results were obtained.

According to the above results after increasing the sample size from 150 to 200 the testing accuracy and the cross validation accuracy results have been significantly improved. By observing the training and testing accuracy results it can be clearly noticed the over fitting problem have been avoided.(Because the training accuracy has been decreased and testing and cross validation accuracy increased. Therefore there isn't a big difference between training and testing accuracy).

After observing the above data it can be clearly identified that the testing accuracy of the each classification algorithm has significant improvement when using k-mean clustering segmentation method. But the main drawback of the usage of k-mean clustering algorithm is this process cannot be fully automated. Because it has to be chosen the disease affected cluster by manually. Therefore inhere the tradeoff between accuracy and automation should be considered according to user requirement.

When considering all above accuracy testing results for each algorithm, the decision tree algorithm has highest accuracy. These results are corresponding to the data set that has been collected. But if there is any changes of the data set (eg:

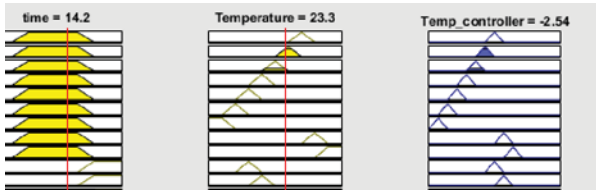

Fig. 11: Temperature results in MATLAB

TABLE III: Temperature Results

Time	Temperature	MATLAB Simulink output results	Results from NodeMCU
14.2	21	-4	-4
14.2	23.3	-2.54	-2.47
14.2	32.9	3.88	3.93
5.61	19.2	-1.22	-1.27
2.53	21.1	4.13	4.06
21.5	26	7.3	7.34

changes the disease types, add more targets classes) can be caused the accuracy result for each classification algorithm. Therefore it is better to choose the most accurate classification algorithm out of above four algorithms, according to the data set which was created after considering user requirements(eg: which diseases, how many diseases.).

To improve the accuracy of the above results the sample size should be increased further and also it is better to create its own data base for each plants (Inhere the database has been created which consists of same diseases with different plants which tends to decrease the accuracy).And also by changing the number of features can be improved the accuracy results (choose most significant features for model training).

V. Conclusion

A fully autonomous system to maintain greenhouses, with a cloud storage for climate data, mobile robot network as data collectors and actuators controlling the climate conditions was implemented. Further, robots in this robotic network consists of automated navigation along with machine learning (reinforcement learning) in order to avoid obstacles. When the number of episodes was increased to 500, the training

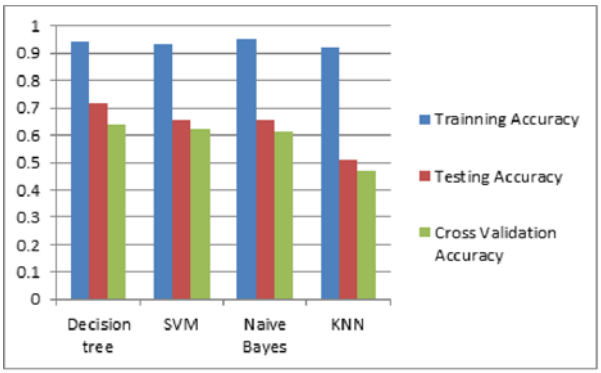

Fig. 12: Accuracy results for thresholding method (200 samples)

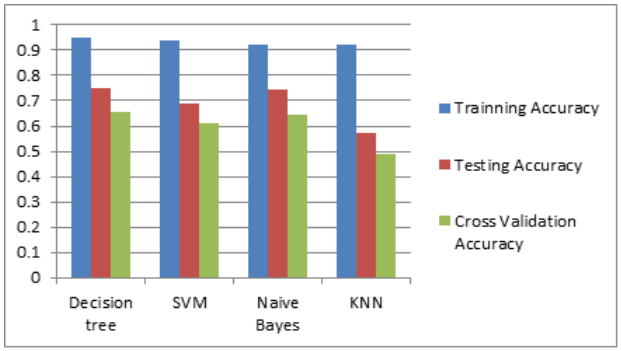

Fig. 13: Accuracy results for k-mean clustering method (200 samples)

curve of the robot showed an improvement in comparison to the 20 episodes. This proves that the robot learns to avoid obstacles and sharp turns. Fuzzy logic was used to obtain the most appropriate decision for the actuators to control the climate conditions in the greenhouse. More importantly, plant disease detection algorithm was created using image processing in order to identify the exact disease of the plant, starting with K-mean clustering to separate the disease and then machine learning classification algorithm was used to identify the disease.

References

[1] Abu Zafar Abbasi, Noman Islam, Zubair Ahmed Shaikh, et al. A review of wireless sensors and networks' applications in agriculture. *Computer Standards & Interfaces*, 36(2):263–270, 2014.

[2] Aya Bseiso, Brett Abele, Shae Ferguson, Peter Lusch, and Khanjan Mehta. A decision support tool for greenhouse farmers in low-resource settings. In *2015 IEEE Global Humanitarian Technology Conference (GHTC)*, pages 292–297. IEEE, 2015.

[3] Kelsey A Czyzyk, Shayne T Bement, William F Dawson, and Khanjan Mehta. Quantifying water savings with greenhouse farming. In *IEEE Global Humanitarian Technology Conference (GHTC 2014)*, pages 325–332. IEEE, 2014.

[4] Nameirakpam Dhanachandra, Khumanthem Manglem, and Yambem Jina Chanu. Image segmentation using k-means clustering algorithm and subtractive clustering algorithm. *Procedia Computer Science*, 54:764–771, 2015.

[5] Mrunalini R Badnakhe and Prashant R Deshmukh. An application of k-means clustering and artificial intelligence in pattern recognition for crop diseases. In *International Conference on Advancements in Information Technology*, 2011.

[6] Keri Woods. Genetic algorithms: Colour image segmentation literature review. 2007.

[7] S. A. Miratabzadeh, N. Gallardo, N. Gamez, K. Haradi, A. R. Puthussery, P. Rad, and M. Jamshidi. Cloud robotics: A software architecture: For heterogeneous large-scale autonomous robots. In *2016 World Automation Congress (WAC)*, pages 1–6, July 2016.

[8] J. P. Shah, H. B. Prajapati, and V. K. Dabhi. A survey on detection and classification of rice plant diseases. In *2016 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC)*, pages 1–8, March 2016.

[9] H. Sabrol and K. Satish. Tomato plant disease classification in digital images using classification tree. In *2016 International Conference on Communication and Signal Processing (ICCSP)*, pages 1242–1246, April 2016.

[10] S. D. Khirade and A. B. Patil. Plant disease detection using image processing. In *2015 International Conference on Computing Communication Control and Automation*, pages 768–771, Feb 2015.

[11] S. JÃąnos and I. Matijevics. Implementation of potential field method for mobile robot navigation in greenhouse environment with wsn support. In *IEEE 8th International Symposium on Intelligent Systems and Informatics*, pages 319–323, Sep. 2010.

[12] R. N. Abutalipov, Y. V. Bolgov, and H. M. Senov. Flowering plants pollination robotic system for greenhouses by means of nano copter (drone aircraft). In *2016 IEEE Conference on Quality Management, Transport and Information Security, Information Technologies (IT MQ IS)*, pages 7–9, Oct 2016.

[13] H. DurmuÅ§, E. O. GÃijneÅ§, and M. KÄśrcÄś. Data acquisition from greenhouses by using autonomous mobile robot. In *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, pages 1–5, July 2016.

[14] R. B. Salikhov and A. A.Zainitdinova. System of monitoring and remote control of microclimate in greenhouses. In *2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE)*, pages 265–267, Oct 2018.

[15] Supun Athukorala, Irunika Weeraratne, Dumindu Jayathilaka, Asitha Bandaranayake, and Roshan Ragel. Affordable real-time environment monitoring system for greenhouses. In *2016 Manufacturing & Industrial Engineering Symposium (MIES)*, pages 1–5. IEEE, 2016.

[16] Mustafa Alper Akkaş and Radosveta Sokullu. An iot-based greenhouse monitoring system with micaz motes. *Procedia computer science*, 113:603–608, 2017.

[17] Aji Hanggoro, Mahesa Adhitya Putra, Rizki Reynaldo, and Riri Fitri Sari. Green house monitoring and controlling using android mobile application. In *2013 International Conference on QiR*, pages 79–85. IEEE, 2013.

[18] Goran Martinović and Janos Simon. Greenhouse microclimatic environment controlled by a mobile measuring station. *Njas-wageningen journal of life sciences*, 70:61–70, 2014.

[19] H. Sasaki, T. Horiuchi, and S. Kato. A study on vision-based mobile robot learning by deep q-network. In *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 799–804, Sep. 2017.

[20] Iker Zamora, Nestor Gonzalez Lopez, Victor Mayoral Vilches, and Alejandro Hernandez Cordero. Extending the openai gym for robotics: a toolkit for reinforcement learning using ros and gazebo. *arXiv preprint arXiv:1608.05742*, 2016.

[21] Supannee Cheewawiriyakul, Brad Gabor, John Kao, and Raquel Salati. Pepper-eggplant pest and disease guide, 2006.

[22] Sai Arivazhagan, R Newlin Shebiah, S Ananthi, and S Vishnu Varthini. Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features. *Agricultural Engineering International: CIGR Journal*, 15(1):211–217, 2013.