PLANTAE: An IoT-Based Predictive Platform for Precision Agriculture

Mohamed Hossam, Mohamed Kamal, Mostafa Moawad, Mohamed Maher, Mohamed Salah, Youssef Abady, Amr Hesham, Ahmed Khattab
Electronics and Electrical Communications Engineering Department, Cairo University, Giza, 12613, Egypt

***Abstract*— This paper presents an Internet of Things (IoT) predictive platform for precision agriculture. The proposed platform aims to improve the productivity of crops through auto-controlling the plantation environment at low cost. Furthermore, the platform uses machine learning to predict plant diseases by implementing deep learning algorithms that extract hidden knowledge from the leaves' images to produce a model to achieve the highest possible accuracy of diseases classification. The platform consists of three layers. The first layer collects the needed information and applies the required actions. The second layer provides connectivity to the Internet. The last layer stores data, analyzes it, and makes it accessible to authorized users.**

Keywords—Internet of Things (IoT); precision agriculture; machine learning; neural networks

I. Introduction

Relying on traditional agriculture techniques developed centuriess ago, reduces the crop yield and quality. This is due to the incorrect irrigation mechanisms, inaccurate application of fertilizers and pesticides, wrong prediction of weather conditions, lack of moisture in the fields, etc. Precision agriculture emerged in the 1980's as the best practice for agriculture and was approved in 1997 by the US congress. Precision agriculture applies the right amount of inputs (water, pesticides, fertilizers, etc.) at the right place and time [1].

Precision agriculture received significant interest from both industry and academia. For example, [2–6] have developed wireless sensor network and Internet of Things (IoT) platforms to acquire the required environmental data, store it, and make simple decisions. Recently, machine learning has been used in precision agriculture to analyze the collected data and make predictive decisions such as in [7]. However, such recent works focus only on machine learning without regard to implementation or system integration issues.

In this paper, we present PLANTAE – Greek word for plant – as the first IoT platform for precision agriculture with built-in deep learning to predict plant diseases in addition to the autonomous control of agricultural attributes. PLANTAE follows a layered IoT architecture composed of three layers: (1) A physical layer composed of sensor and actuator nodes and image capturing devices; (2) A network layer which is a gateway to provide connectivity to the Internet; (3) An application layer that is responsible for data storage, user interfacing, and data analysis using deep learning.

The rest of the paper is organized as follows. We present PLANTAE and detail its components in Section II. We discuss the used deep learning approaches in Section III. Section IV presents the evaluation results and Section V concludes the paper.

II. PLANTAE Predictive IoT platform

A. Overall System Architecture

The PLANTAE IoT predictive system is composed of 3 main layers [8] which are shown in Fig. 1.

1) Physical Layer: The first layer consists of the embedded nodes which are deployed in the agriculture field. The nodes can be either equipped with sensors to collect the needed information or equipped with actuators to implement the actions taken based on the collected information. The nodes should be capable of communicating with the following layer to send the collected information and receive the actions.

2) Network Layer: This layer connects the system to the Internet to enable the communication between the physical layer and the application layer. Mainly, it consists of the gateway which collects data from the nodes, runs primitive analysis and simple processing on the data and forwards it to the next layer for detailed processing. The gateway also receives the responses from the application layer and forwards them to the targeted actuator nodes.

3) Application Layer: The final layer of the platform is responsible for analyzing the collected information, storing it for history-keeping, and providing an interface through which the system owner configures nodes, apply predefined actions, or access the data or upload plant photos from any Internet-enabled device. Therefore, the application layer is composed of 3 main components which are a database, a web server and a deep learning model to predict the planet diseases.

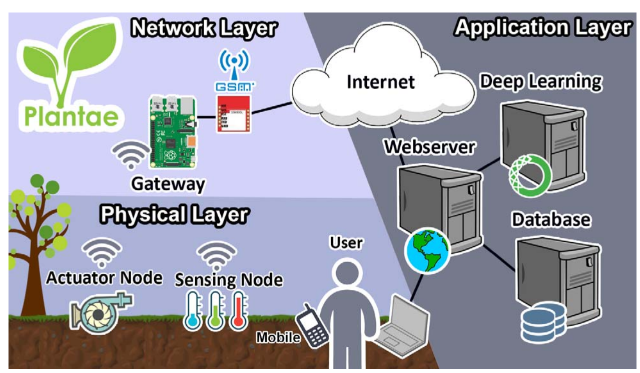


Fig. 1. PLANTAE overall system architecture.

B. Physical Layer

This layer contains 2 types of nodes which are physically deployed in the agriculture field in addition to image capturing devices that can be held with the platform user.

1) Sensor Nodes: These nodes are responsible for monitoring and recording the values of the needed environmental attributes of the field and sending them periodically to the gateway [5]. In order to be energy-efficient, the nodes enter a sleep mode after the readings are sent. A sensor node consists of a microcontroller, a set of sensors, and a wireless module.

For our platform, we use the ATmega16 microcontroller as its speed, power consumption and memory fit our requirements. We interface the microcontroller to a soil moisture sensor (SEN0114), a pH sensor (PHE-45P) and a temperature and humidity (DHT11) sensor. The soil moisture sensor measures the water content in the soil which can be used to determine the amount of water needed in the irrigation process to increase the field yield and improve the quality of the crops. The soil moisture sensor produces an analog output. Hence, an interface circuit is used to convert such analog signal to digital format using an analog to digital converter (ADC) followed by signal conditioning. The temperature and humidity values are to be measured as they contribute to problems such as foliar, root diseases, slow drying, and plant stress [1]. Single-bus data format is used for communication and synchronization between the microcontroller and the DHT11 sensor that produces a digital output [6].

Communication between the nodes and the gateway can be done wired or wirelessly. Wireless communication is more suitable than wired communication in farms due to its easy installation as wires are not practical nor safe in such environments. They may be damaged or exposed and the farm may be flooded which will electrify the field. Out of the available wireless technologies, Bluetooth, Ultra-wideband (UWB) and Infrared (IR) are not good candidates due to their short ranges which is not suitable to be used in farms. On the other hand, Zigbee and Wi-Fi have higher communication ranges. If power consumption is not a main focus (as nodes in precision agriculture are typically power using solar cells) Wi-Fi is suitable in communication between the nodes and the gateway. We use the ESP8266 Wi-Fi module which is a self-contained system-on-chip with integrated TCP/IP protocol stack that can give the microcontroller access to any Wi-Fi device [9]. The communication range of such a module is 150 feet which is good for our application.

2) Actuator Nodes: The role of the actuator node is to control certain physical quantities based on the sensor nodes' readings or based on the system owner's decision applied using Internet-enabled devices. More specifically, our actuator node controls the amount of irrigation water that reaches the soil and its pH based on the moisture level and the pH level.

Similar to a sensor node, an actuator node is composed of an Atmega16 microcontroller, a set of pumps (a water pump, a pH up pump and a pH down pump), and an ESP8266 Wi-Fi module. The pH up and down pumps contain liquids which are commercial products or regular household substances that are added to a hydroponic nutrient solution in order to instantly increase or decrease the solution's pH levels. We also equip the actuator node with a fan to control the temperature.

3) Image Capturing Devices: The proposed platform also uses images to detect plant diseases by applying machine learning techniques. One way to obtain such images is to deploy cameras across the monitored agriculture area and connect such a surveillance system to the Internet. However, such a system has a high cost and lacks flexibility. Instead, we propose to allow the platform users to use their mobile phones or cameras to take pictures of the leaves whenever needed and directly upload such photos to the application layer for further processing. This increases the flexibility and reachability of the image capturing module of the platform.

C. Network Layer

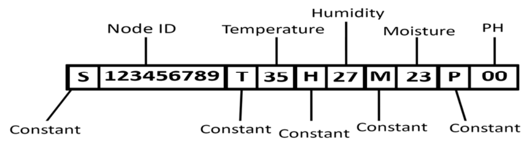
The network layer is responsible for connecting the physical layer sensor and actuator nodes to the components of the application layer. More specifically, it collects the information gathered by the sensor nodes, forms packets that contain such information and saves it to the application layer database via the web Application Programming Interface (API), and forwards the requests sent from the web API to the actuator nodes. It is also responsible for taking decisions in emergency situations and sends warnings to system owner via the messaging system developed on the web API.

The network layer is mainly implemented as a gateway. The gateway consists of two components: a powerful microcontroller to be able to perform the aforementioned preprocessing and wireless connectivity modules that connects the gateway to the physical layer nodes and to the Internet.

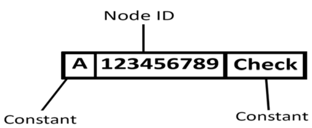
PLANTAE uses Raspberry Pi 3 as the microcontroller in the gateway which has a built-in Wi-Fi module which is the protocol used in communications between the sensor and actuator nodes and the gateway [10]. We added a GSM/GPRS module to the Raspberry Pi to connect the gateway to the cellular network, and hence, to the Internet.

The Raspberry Pi 3 runs the Raspbian OS. Hostapd software is the used to change the Wi-Fi module mode to work as an access point with certain SSID (which is known to the physical layer nodes) and with WPA2 encryption. When a node comes to live for the first time, it keeps searching for that SSID until it finds it and connect to it automatically. The gateway listens on port "1911" for any incoming packets. This port is known to the nodes such that they can send data and requests to it. This is implemented by multithreaded socket server coded in Python 3. The python script can analyze the packets sent by the nodes to extract information and then connects to the GSM module via the UART connection between the Raspberry Pi and the used GSM800L module with a baud rate equal to 9600.

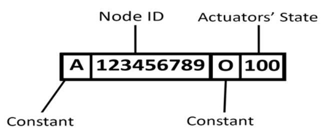
The Raspberry pi sends commands to the GSM module to connect to the cellular network and to start the GPRS. In case of sensor nodes, it submits the data to the web API via GET method such that the web API can store it directly to the database. The sent packet contains the "Node ID, Temperature sensor reading, Humidity sensor reading, Moisture sensor reading and PH sensor reading" as shown in Fig.2(a). In case of actuator nodes, it submits the request with the node ID as shown in Fig.2(b) to the web API which responses with the actuator's state which can be either ON or OFF (as shown in Fig.2(c)), then the gateway sends the response to the actuator node such that it can take action. The gateway supports a large number of nodes to be connected to it, that number can be modified by changing the subnet mask of the network provided by the DHCP server "DNSmasq".



a) Sensor node readings packet



b) Actuator check request packet



c) Actuator state packet

Fig. 2. Gateway messages exchanged with the senssor and actuator nodes.

D. Application Layer

The application layer is composed of a database for information storage, a webserver for human interfacing, and a machine learning model for data analysis using machine learning. The database used in the system is the relational MySQL database hosted online [11]. MySQL is used as it is supported by almost every programming language including PHP which is used in developing the web interface. The main database consists of 6 tables:

- "*Actuators*" which consists of all the current actuator nodes' states in the whole system.
- "*Sensors*" which consists of all the readings of every sensor node in the system, its ID and the timestamp when that reading was submitted to the system.
- "*Messages*" which consists of the warning messages sent to the users when the actuators' state changed by the system automatically and any other messages sent other users on the website.
- "*Pics*" which consists of file names of the photos of the leaves of the crops taken by the user which are uploaded via the website interface to be analyzed by the deep learning script to check for infection.
- "*Nodes*" which consists of a list of all the nodes (actuator and sensor) connected to the system and who owns each node.
- "*Users*" which consists of information about the system's users such as their emails, passwords, names, the crops they are currently farming and the geolocation of their farms.

The web server provides PALANAE with the IoT system's API, a human-to-machine interface, process automation, and a method to upload photos of leaves to be analyzed with deep learning. PHP is used in programming the website, while the user interface design is based on Dashio, a Bootstrap template, also Javascript and Jquery are used in development. The user interface allows the user to sign up for an account, login to it (after authentication and authorization), add nodes, visualize the data sent from the nodes in form of graphs, control the actuators remotely and send messages to other users. The farm geographical location is added by the user using Google Maps Javascript API, such that the system provides the user with important weather information via OpenWeatherMap API.

III. DEEP LEARNING FOR PLANT DISEASES CLASSIFICATION

Deep Learning is applied at the application layer of the proposed platform to detect diseases that infect the monitored crops. More specifically, we aim to detect the diseases from the shape of leaves of the plants from the photos taken by the users and uploaded to the webserver. Diseases cause several negative effects on the leaves which result in changing the shape of the leaf. Images can be taken by RGB sensors such as mobile phone cameras or standard RGB cameras. Image processing techniques (such as data augmentation) may be used before applying deep learning models to detect the disease from the leaves' shapes to achieve the maximum possible accuracy for classification.

The main classifiers used in the PLANTAE platform are made of convolutional neural networks, which are deep artificial neural networks used primarily to classify images. The network consists of different types of layers and each layer is made up of neurons. The main goal for each layer is to extract the main features from images.

Out of the existing neural network architectures, we chose GoogleNet, AlexNet and VGG-19 – due to their outstanding performance in image classification – to be implemented in the PLANTAE application layer to train the model from scratch for this work. AlexNet is one of the first deep networks developed to push ImageNet classification accuracy by a significant stride in comparison to traditional methodologies [12]. VGG-19 achieves competitive classification accuracy compared to more complicated networks despite using a simple linear chain of layers [13]. GoogleNet (originally referred to as Inception) was developed to set the new state of the art for classification and detection in the ImageNet competition. It makes several improvements in terms of the computational expense of running the model. This allowed for increasing the depth and width of the network while keeping the computational budget constant [14].

IV. EXPERIMENTAL EVALUATION

In this section, we present sample results of the experiments performed to demonstrate the performance of the performance of the PLANTAE platform. We first show how the platform is capable of autonomously controlling the agriculture field conditions based on the sensed attributes. Then, we evaluate the accuracy of the implemented deep learning algorithms used for detecting the plant diseases.

A. *Autonomous Farm Control*

Figures 3 and 4 depict the results of two experiments used to show the PLANTAE autonomous farm control capabilities. In Fig. 3, the water pump is signaled to operate when the soil moisture drops below 30%. The fan is turned on when the temperature exceeds 37° C in Fig. 4. Note that, in both experiments the ON duration of the actuator is computed based on real-life data to ensure resolving the sensed abnormal situation. This alleviates the need to keep the sensor and actuator nodes unnecessarily ON since the control of such environmental conditions is predictable, and there is no need to continuously sensing them during the control process.

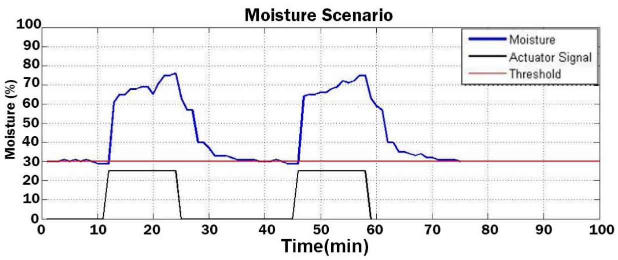


Fig. 3. The water pump opens once the soil moisutre drops below 30%.

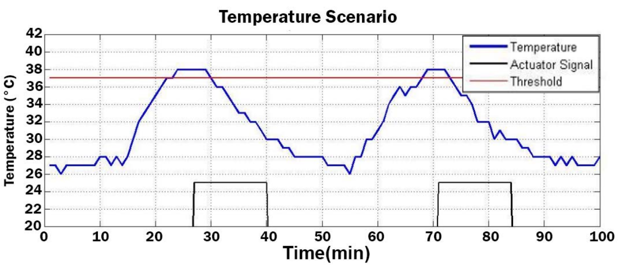


Fig. 4. The fan is turned on when the temperature exceeds 37° C.

B. *Plant Disease Detection*

Next, we evaluate the performance of the implemented deep learning algorithms used by PLANTAE for plant disease detection. We use PlantVillage dataset for training [15]. The dataset consists of 34 classes and contains more than 55,000 labelled images for 9 different crops. We divided the images to 80% for training and 20% for testing. In order to enrich the dataset and to introduce slight distortion to the images, data augmentation (through horizontal reflections, rotation and flipping) is used to increase the chance for the network to learn more features and reduce the problem of overfitting.

All training and testing operations are performed using Nvidia GTX 950m GPU, which have 640 CUDA core and 4 GB of VRAM. The software used was CUDA 9.1 and CudNN 7 for TensorFlow 1.5 on Ubuntu Linux distribution. Table I depicts the training time and the accuracy of the different used algorithms tested using the tomato dataset which consists of 15,000 training samples which became 90,000 after data augmentation and around 3500 test samples of 11 classes: 9 diseases, one healthy and one backgrounds class. GoogleNet achieves the highest accuracy which comes with the largest training time, whereas AlexNet achieves the least accuracy with the smallest training time. VGG-19 is a compromise.

TABLE I. PLANT DISEASE DECTION TIME AND ACCUARCY

Model	Training Time (hours)	Accuracy
GoogleNet [14]	5	98 %
VGG-19 [13]	4.12	97.3 %
AlexNet [12]	1.02	95.73 %

V. CONCLUSIONS

We have presented the PLANTAE IoT platform for precision agriculture. The platform is composed of 3 layers: physical layer, network layer, and application layer. It controls the agriculture process autonomously according to the sensed farm environment. All the data is accessible to any authorized user connected to the Internet. Using deep learning, PLANTAE detects plant diseases with high accuracy.

ACKNOWLEDGMENT

This publication was made possible by funding from the Egyptian Science and Technology Development Fund (STDF), project # DDP-5992. The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] United Nations International Strategy for Disaster Reduction (UNISDR), “Developing Early Warning Systems: A checklist”, *Third International Conference on Early Warning*, 2008.

[2] K. A. Patil, N. R. Kale “A Model for Smart Agriculture Using IoT”, in *Proc. of IEEE Int’l Conf. on Global Trends in Signal Processing, Information Computing and Communication*, 2016

[3] A. Cosma, C. Preda, M. Luculescu, L. Cristea, S. Zamfira “Data Acquisition System Used in Precision Agriculture for Vegetation Status Monitoring - Software Subsystem” in *Proc. of International Conference on Optimization of Electrical and Electronic Equipment (OPTIM)*, 2017.

[4] A. Khattab, A. Abdelgawad, K. Yelamarthi, “Design and Implementation of a Cloud-based IoT Scheme for Precision Agriculture," in *Proc. of IEEE Inte’l Conf. on Microelectronics*, 2016.

[5] K. Krishna, O. Silver, W. Fahad Malende, K.Anuradha, “Internet of Things Application for Implementation of Smart Agriculture System”, in *Proc. of Int’l Conf. on IoT in Social, Mobile, Analytics and Cloud*, 2017.

[6] K. O. Flores, l. M. Butaslac, J. Enric M. Gonzales, S. Matthew G. Dumlao, R. Reyes “Precision Agriculture Monitoring System using Wireless Sensor Network and Raspberry Pi Local Server”, in *Proc. of IEEE Region 10 Conference (TENCON)*, 2016.

[7] R. Gandhi, S. Nimbalkar, N. Yelamanchili and S. Ponkshe, "Plant disease detection using CNNs and GANs as an augmentative approach," in *Proc. of Int’l Conf. on Innovative Research and Development*, 2018.

[8] C. Zhong, Z. Zhu, R. Huang, “Study on the IoT Architecture and Gateway Technology”, *in Proc. of Int’l Symp. on Distributed Computing and Applications for Business Engineering and Science*, 2015.

[9] *“ESP-WROOM-32,* https://www.espressif.com/. Accessed: Nov., 2018.

[10] *“Raspberry Pi,” https://www.raspberrypi.org/*, Accessed: Nov., 2018. .

[11] S. Rautmare, D. M. Bhalerao, “MySQL and NoSQL database comparison for IoT application”, in *Proc. of IEEE Int’l Conf. on Advances in Computer Applications (ICACA)*, 2016.

[12] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "ImageNet Classification with Deep Convolutional Neural Network", Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[13] K. Simonyan and A. Zisserman. "Very deep convolutional networks for large-scale image recognition", *arXiv:1409.1556*, 2014.

[14] C. Szegedy et al., ‘‘Going deeper with convolutions,’’ *arXiv:1409.4842*, 2014.

[15] *"PlantVillage dataset", https://github.com/salathegroup/plantvillage_deeplearning_paper_dataset, Accessed*: Nov., 2018.