A Smart IoT-Based Irrigation System with Automated Plant Recognition using Deep Learning

Jessica Kwok
Student
Claremont High School
1601 N.Indian Blvd,
Claremont, CA 91711
(626) 425-8118
jessica.kwok19@gmail.com

Yu Sun
Assistant Professor
Department of Computer Science
California State Polytechnic University, Pomona
Pomona, CA 91768
(909) 869-3449
yusun@cpp.edu

ABSTRACT
Machine Learning allows systems to learn and improve automatically from experiences without hand-coding. Thus, in recent years, many technology companies have been developing such application if Artificial Intelligence, from face recognition by Facebook, to the AlphaGo program by Google. The irrigation systems in the market nowadays mostly allow users to set them to a certain amount of water and at specific time intervals. However, there are usually more than one type of plants in a garden, and each species requires different amount of water. In order to resolve this issue, in this paper, we have developed an irrigation system, with the use of deep learning, that is able to adjust the amounts of water foe each type pf plant through plants recognition. There are two main parts of the solution, the software and the hardware. The prior is connected with cameras to undergo plant recognition, and utilizes database to find the suitable amount of water; the latter controls the amount of water that is able to flow out.

CCS Concepts
• Information systems → Mobile information processing systems • Computing methodologies → Classification and regression trees.

Keywords
Machine Learning; Irrigation System; Image Classification; Soil Moisture Content.

1. INTRODUCTION
Irrigation systems have become almost essential in the current society. From private gardens in backyards to large-scale farms, people no longer have the time and effort to water their plants manually every day; most choose to rely on irrigation systems and professional gardeners to take care of their plants.

In the current market, there is an array of irrigation systems that have different advantages and disadvantages depending on the size of the area, natural conditions, the type of crops, cost, and labor[1]. The most commonly used ones include the drip system and the overhead sprinklers[2]. Despite the difference in the designs and structures, most systems contain valves that are connected to a controller, as shown in Figure 1. The controller allows user to program in the time periods that the valves should be open. When the valves are open, then the water can flow through. Because of these irrigation systems, watering becomes extremely convenient.

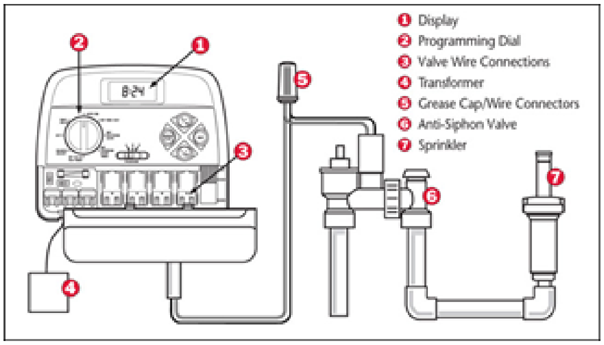

Figure 1. Irrigation System Diagram (from [3]).

However, the automatic watering system has its problems. One of the most significant problems is the improper irrigation scheduling. Proper irrigation means allows the soil moisture to be at a level that will lead to the maximum productivity of the plant. Soil moisture is defined as the amount of water between soil particles. Improper irrigation may lead to two different consequences: overwatering, and underwatering. Overwatering not only waste money and water, it might cause the crops to die as they are unable to absorb oxygen in the soil. On the other hand, underwatering means that the crops are not absorbing enough nutrients from the soil[4]. Since different plants require various levels of soil moisture, for people inexperienced in gardening, deciding how much water each type of plants need seems not so intuitionistic. In addition, weather and other factors would easily affect the moisture content in the soil. It becomes problematic to users when they have to constantly update the programs in the controllers depending on these unavoidable changes.

The current approach to this problem is to divide the yard into different functional zones depending on the plants. The controllers are designed to usually have about three pre-set programs. Thus, users can then select which program to use in different zones. This approach does increase the flexibility of the irrigation system, but it is often not enough when a garden has more than three zones. In addition, it is unlikely for the approach to address the weather aspect of the problem. In regions that are always in a drought, like

ICCMS 2018, January 8–10, 2018, Sydney, Australia

ACM ISBN 978-1-4503-6339-6/18/01…$15.00
https://doi.org/10.1145/3177457.3177506

California, it is essential to design an irrigation system that can optimize the usage of water.

In this paper, we proposed a resolution to both increase the flexibility and convenience of irrigation systems. The incorporation of both machine learning, database, and moisture sensors addresses the mentioned problems in the current irrigation systems.

The remaining paper is organized as the following: In Section 2, the challenges are explained and elaborated; In Section 3, the solution will be addressed; In Section 4, existing work will me mentioned; In Section 5, we will summarize the solution and the limitations that follow.

2. DEVELOPMENT CHALLENGES

2.1 Challenge 1: Varying soil moisture content

The first challenge that will be addressed is the varying soil moisture content for different types of plants. Each type of plant usually requires different amount of water to optimize its growth, thus they each need different amount of irrigation time and frequency. It is extremely inconvenient for the owner to find the suitable irrigation time for each of the plants in the garden, and add them into the program. For example, cotton would need a total of around 1,000mm of water in its total growing period, while sunflower only needs around 800mm of water[5]. If the garden waters the plants uniformaly and from the same amount of time, one or the other would not grow properly.

Crop	Crop water need (mm/total growing period)
Beans	300 - 500
Citrus	900 - 1200
Cotton	700 - 1300
Groundnut	500 - 700
Maize	500 - 800
Sorghum/millet	450 - 650
Soybean	450 - 700
Sunflower	600 - 1000

Figure 2. Approximate values of seasonal crop water needs (adapted from [5]).

2.2 Challenge 2: Inconsistent weather affects soil moisture

Throughout the four seasons, the humidity and precipitation fluctuate drastically for many cities. For example, in San Diego, California, the humidity went from 0% to around 90% through the year of 2016[6]. The humidity level greatly affects the soil moisture percentage. When the humidity is high, it often shortens the necessary irrigation time.

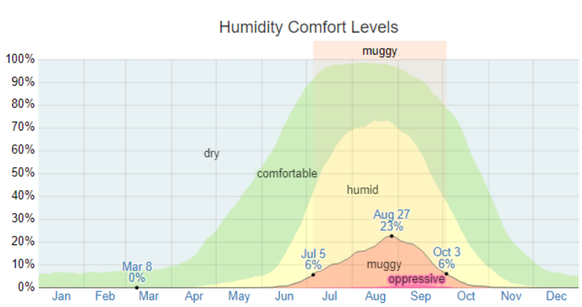

Figure 3. Humidity Comfort Levels in San Diego, 2016 (adapted from [6]).

In addition to that, the difference in temperature can also affect the soil content significantly.

The weather each year is unpredictable, so the owner cannot pre-program an irrigation schedule suitable for each season before hand. Many would keep the program the same, and this would cause potential overwatering or underwatering. Second, it might be a waste of water and money. It would be time-consuming for the owner to adjust the program every season in the year.

3. SOLUTION

To address these issues, we have designed and implemented an IoT system that both provides flexibility and convenience to users. The overall process of the solution is illustrated in Figure 4.

3.1 Plant Recognition

In order to get the most accurate configuration for the plant watering, an automated mobile application tool has been developed to automatically detect the type of the plants, which is based on deep learning. The implementation is based on the following tools and environments:

- Docker
- TensorFlow
- ImageNet database
- Android Studio

3.1.1 Docker Configuration

Docker is a software container platform that allows to collaborate with co-workers without running it on the machine. Developers can run the package software in an isolate form, increasing efficiency and the trouble of a full operating system[7]. Because the data that needs to be used in the developed software is rather big, it is necessary to utilize such platform.

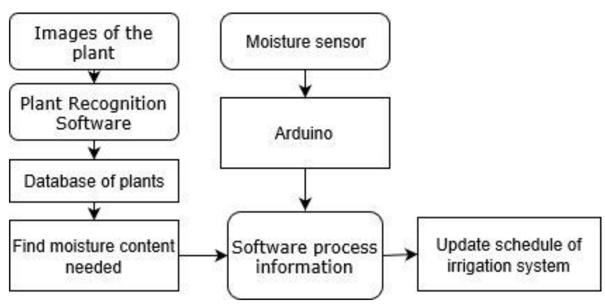

Figure 4. Flow Chart of General Process.

3.1.2 TensorFlow Configuration

TensorFlow is an open-source software library for Machine Intelligence developed by the Google Brain Team to conduct machine learning and deep neural networks research[8]. TensorFlow provides the deep learning aspect of this project. Deep learning can be defined as a type of machine learning that uses a non-linear approach to process data. This means that it can consider multiple factors while finding a pattern [9]. This is significant because the results would often be more accurate. In this paper, deep learning will be used to classify plants through image recognition after downloading TensorFlow. It can be downloaded through cloning the git repository.

3.1.3 Image Dataset Training

For this prototype, transferable learning is used, which means that we are starting with a pre-existing model. The model that we worked on have been trained on ImageNet Large Visual Recognition Challenge dataset, a model that can classify up to 1,000 classes [10]. This is because deep learning from scratch might take a very long period of time.

To make the software recognize the plant, we need to upload a huge number of images of each type of plants into the TensorFlow environment. In this prototype, we chose approximately 10 species for experimental purpose, including orchid, roses, lily, cherry blossom etc. First, 10 files were made, each named with a plant. Then, 20-30 images of that species were downloaded and saved into the respective file. Then, the following code was entered to train the images:

```
python retrain.py\
 --bottleneck_dir=bottlenecks \
 --how_many_training_steps=500 \
 --model_dir=inception \
 --summaries_dir=training_summaries/basic \
 --output_graph=retrained_graph.pb \
 --output_labels=retrained_labels.txt \
 --image_dir=flower_photos
```

3.1.4 Image Classification and Prediction

The process of predicting an image becomes extremely convenient with the use of TensorFlow. Once the user uploads an image into the environment and saved it as <your file name>, the software can predict the how many percent probability that the image would fit into each category with the following code.

For example, after uploading an image of a cherry blossom into the environment and running the code, the following result is returned:

```
root@90199e481a62:/tf_files# python label_image.py flc
ossoms_in_Vancouver_3_crop.jpg
cherry (score = 0.89120)
roses (score = 0.10147)
dandelion (score = 0.00257)
tulips (score = 0.00243)
sunflowers (score = 0.00109)
white lily (score = 0.00094)
daisy (score = 0.00031)
root@90199e481a62:/tf_files#
```

This shows that the software thinks the image has an 89% probability of being a cherry blossom, which is fairly accurate.

3.1.5 Mobile Integration

Figure 5. Screenshot of Plat Recognition on the Mobile Device.

The finally part of developing the software is to incorporate it into an app so that it can be downloaded on mobile devices. In this project, Android Studio, and Android-specific development environment is used. An android project is already built to link the camera function of a device with the plant recognition proportion of the software [11]. After optimizing the TensorFlow model, the trained model will be optimized for the Android platform to use followed by being injecting to the into the pre-built android project. We decide to pre-load all the trained models to the mobile app instead of relying on a backend prediction server, because the users can have the best flexibility to use the system without being affected by the Internet connection status. In addition, running the image classification on the mobile device is generally faster than going through the backend for remote prediction.

In the project, after a plant is identified, its most ideal soil moisture content will be searched in the database. Figure 5 shows a screenshot of the mobile app recognizing the plant type.

3.2 The IoT Irrigation Controller

The actual control of the watering system is implemented as an IoT system. The system has both moisture and temperature sensors to monitor the physical environment in real-time. Meanwhile, the system also connects to the power of the irrigation system and triggers the switch based on the condition of the soil environment and the specific configuration for the plants. The construction of the IoT system is based on the following platforms and tools:

- Arduino UNO REV3
- Arduino IDE
- Temperature/Moisture Sensor

3.2.1 Arduino UNO REV3

Arduino is a microcontroller board with inputs, output, USB connection, power jack, header, and reset button [12]. In this system, it is used to receive inputs from the moisture sensor, then transmit them to a computer through Bluetooth.

3.2.2 Arduino IDE

The Arduino IDE is a development environment specifically for the Arduino board that uses the language C++. It allows users to code, upload the program to the board, and receive data from the board easily. We used it to program the Arduino so that it sends data to the computer every three seconds. It can also directly convert the numbers received from the senor to understandable moisture level, through simple lines of code.

3.2.3 Temperature/Moisture Sensor

The temperature/moisture sensor that we used in this prototype is a Phantom YoYo Arduino compatible High Sensitivity Temperature/Moisture Sensor. It is highly compatible with the Arduino interface and it can measure the moisture just by inserting it into the soil. The temperature/moisture sensor is connected to the Arduino Uno as the following:

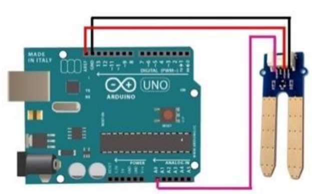

Figure 6. Temperature/Moisture Sensor Connected to Arduino.

Before understanding how it measures the moisture, we need to understand the fact that since water is polar, it makes electricity to be conducted more easily. Thus, when the sensor is inserted into the soil, with the two probes, it tries to pass current through the soil. Then, the moisture sensor reads the amount of resistance to get the moisture level [13]. After it reads the moisture level, the sensor then sends the data to the Arduino as inputs. Finally, through Bluetooth, the computer receives these results.

3.3 Attacking the Challenges

After finding the most ideal soil moisture with the app, combining with the result of the current soil moisture level with the Arduino, the software will be able to calculate how long to water the plants, thus manipulate the program in the irrigation system controller to adjust the amount of time valve should be opened.

The plant recognition function of the app resolves the first challenge – the varying soil moisture content in different plants. With the use of machine learning, a snap of picture can allow the database to search for the most ideal soil moisture for the respective plant. There are two main benefits of the mentioned function: first, it allows users inexperienced with a specific plant to still grow it successfully without additional research; second, it saves the trouble for users to set various programs manually for each zone in their garden, as the controller can automatically be set with the information given from the plant recognition app.

The soil moisture sensor proportion of this prototype resolves the second challenge – inconsistency in weather that affects soil moisture. Since the moisture sensor is constantly sending data to the machine, we can adjust the period for irrigation depending of this information.

For example, if the most ideal soil moisture level for rose is 50%, and it is currently sunny, and the sensor shows that the moisture level is 40%, then then the valve will continue to be opened for a certain time. Once the percentage became ideal, which is approximately 50% for a certain time, the app will tell the controller to close the valve.

The advantages of this design are the following. First, watering will stop so that no excess water is wasted. Second, no matter if the weather has a higher or lower humidity, the soil moisture sensor makes the irrigation system more flexible.

4. RELATED WORK

Similar work also applied machine learning to irrigation system. But instead of identifying the plant's type, the study used machine learning of climate data, which means that it learns the pattern through maximum and minimum temperature every day to predict the crop water demand [14]. This work is significant as it also incorporated deep learning. However, even though this study does address the second challenge of the problem that weather could change the soil moisture, it did not incorporate the varying crop water demand for different types of plants.

Besides related work about the irrigation system, there are also studies that incorporated deep learning into the well-being of the plants. Crop diseases have been extremely detrimental, especially for countries that lack infrastructures or experienced professionals. To solve this issue, a study is designed to detect plant diseases, like Apple Scalp, through its images. This is done by using similar process in this paper. It also utilized deep learning to recognize images with visible disease. This research has very positive result, its trained model has an accuracy of 99.35% [15].

5. CONCLUSION AND FUTURE WORK

To summarize, this project had aimed at making irrigation systems more adjustable and simple for users. To save the trouble for inexperienced users in gardening to figure out information about each plant, this projected used deep learning to identify plants, and database to find its most ideal soil moisture level. Furthermore, to allow the irrigation systems to adjust accordingly to the weather, this project used soil moisture meters and Arduino to constantly update the irrigation system.

For future work, we plan to test the modified system on a larger scale. Instead of just having about 10-20 classes, we aim to include a bigger variety of classes. In addition, we plan to improve the accuracy of the model by training a bigger number of images for each class. Furthermore, from studying related works, we plan to also integrate more machine learning into project. This can be done by predicting the soil moisture levels based on data from the weather forecast. Maximum and minimum temperature, humidity, and precipitation should all be considered.

6. REFERENCES

[1] "7. Choosing an irrigation method." FAO Corporate Document Repository, Natural Resources Management and Environment Department. DOI= http://www.fao.org/docrep/s8684e/s8684e 8.htm

[2] Types of Irrigation Systems. GardenSoft, 2017. DOI= www.ladwp.cafriendlylandscaping.com/Garden-Resources/IrrigationType.php.

[3] "How To Design An Irrigation System At Home." DOI= www.backlinkservices.us/how-to-design-an-irrigation-system-at-home/.

[4] Law, Inc. WP. "The Most Common Problems with Farm Irrigation Systems." 7 Nov. 2013. DOI= www.wplawinc.com/agricultural-irrigation-blog/the-most-common-problems-with-farm-irrigation-systems.

[5] "2. Water and Soil Requirement." FAO Corporate Document Repository, Natural Resources Management and Environment Department. DOI= www.fao.org/docrep/u3160e/u3160e04.htm.

[6] WeatherSpark. *Average Weather in San Diego, California, United States, Year Round*, Cedar Lake Ventures. DOI= weatherspark.com/y/1816/Average-Weather-in-San-Diego-California-United-States-Year-Round.

[7] *What Is Docker*. Docker Inc, 1 Sept. 2017. DOI= www.docker.com/what-docker.

[8] TensorFlow, and Google. "TensorFlow For Poets." Codelabs. DOI= codelabs.developers.google.com/codelabs/tensorflow-for-poets/#0.

[9] Momoh, Osi. "Deep Learning." Investopedia, 30 Oct. 2016. DOI= http://www.investopedia.com/terms/d/deep-learning.asp

[10] Olga Russakovsky*, Jia Deng*, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg and Li Fei-Fei. (* = equal contribution) ImageNet Large Scale Visual Recognition Challenge. IJCV, 2015.

[11] Yu, Sun. "Tensorflow Mobile Example." GitHub, 25 Aug. 2017. DOI= https://github.com/sunyu912/tensorflow-mobile-example

[12] "Arduino Uno Rev3." Arduino, 2017. DOI= https://store.arduino.cc/usa/arduino-uno-rev3

[13] Demay, Vincent. "Measure Soil Moisture with Arduino – Gardening." Homautomation, 20 June 2014. DOI=

http://www.homautomation.org/2014/06/20/measure-soil-moisture-with-arduino-gardening/

[14] Torres, Alfonso F., et al. “Forecasting Daily Potential Evapotranspiration Using Machine Learning and Limited Climatic Data.” *Agricultural Water Management*, vol. 98, no. 4, 2011, pp. 553–562., DOI= 10.1016/j.agwat.2010.10.012.

[15] Mohanty, Sharada P., et al. “Using Deep Learning for Image-Based Plant Disease Detection.” *Frontiers in Plant Science*, vol. 7, 2016, DOI= 10.3389/fpls.2016.01419.