Modeling Earth Systems and Environment
https://doi.org/10.1007/s40808-022-01452-0

ORIGINAL ARTICLE

Towards automating irrigation: a fuzzy logic-based water irrigation system using IoT and deep learning

Pranshu Patel[1] · Yugma Patel[1] · Utsav Patel[1] · Vrukshal Patel[1] · Nitya Patel[1] · Parita Oza[1] · Usha Patel[1]

Received: 1 May 2022 / Accepted: 17 June 2022

Abstract
Among the developing countries, India is one of the fastest-growing economies in the world today, and a considerable part of this economy is dependent on its agricultural sector. The country also boasts the world's second-largest population, implying that resource scarcity is constantly a concern. Fresh water is one of the most crucial and precious resources. We also have unpredictable monsoons and isotropic climate conditions to contend with. For many years, traditional techniques have been utilized to irrigate gardens and farms. This technique still needs automation in order to reduce the time and effort required for manual examination. We present a model for an automated watering system that attempts to reduce both human interaction and water usage. Our system offers a simple interface for cultivating in various settings, from home to industrial. This system will use data from soil moisture sensors and images of crops acquired by a camera and weather forecasts via the application programming interface (API). In addition, we used a deep learning model to classify captured images into a droop and healthy class. Finally, our fuzzy logic algorithm aggregates all these parameters and regulates the irrigation system's operation time.

Keywords Water irrigation · Sensors · IoT · Fuzzy logic · Deep learning

Introduction

India has always been an agricultural nation. Almost half of the employment in India comes from the agricultural sector (Agriculture employment in India 2021). Agriculture is the primary means of living for about 58% of India's population (Agriculture in India 2021). The share of agriculture in GDP was 19.9% in 2020–21 (Agriculture in India 2021). However, optimal water use during irrigation of crops needs to be addressed because the lack of fresh water is a rising concern in many parts of the world. Fig. 1 shows the decrease in per capita water supply over 100 years in India. Water availability per capita has decreased by 70% since 1950, and this trend is expected to continue over the next decade.

Before the advent of manual water irrigation methods, farmers primarily relied on rain for watering their crops. However, nowadays farmers cannot rely solely on rainwater due to global warming because climate change is unpredictable (global warming causing climate changes 2021). We have also observed abrupt drought conditions in many places in recent years. Moreover, only 2.5% of the total volume of water comprises freshwater, so the usage of freshwater needs to be appropriately managed (Fresh water in worldwide 2021). Irrigation is scheduled around the world based on farmers' visual assessment of crops, which wastes nearly half of the water used by traditional irrigation systems (Mulenga et al 2018). Approximately 70% of the entire freshwater reserves are used for agriculture, so the water wastage should be minimal to preserve the resources for future generations (Water Usage for agriculture irrigation 2021).

Sprinkle irrigation, drip irrigation, and furrow irrigation are examples of controlled irrigation methods that reduce water waste by 30–70% (national geographic data 2020). However, due to the open-loop layout, these techniques fail to maintain actual water content in the soil, resulting in lower crop quality and quantity when soil nutrients are depleted by under or over-irrigation (Morillo et al 2015).

✉ Parita Oza
parita.prajapati@nirmauni.ac.in

✉ Usha Patel
ushapatel@nirmauni.ac.in

[1] Department of Computer Science and Technology, Nirma University, Ahmedabad, Gujarat 382481, India

Published online: 09 July 2022

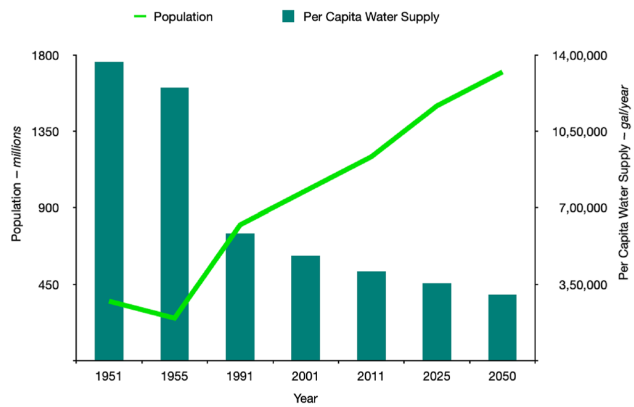


Fig. 1 Population and per capita water supply in India (KPMG 2010)

Due to India's high cost and lack of advanced methods, most farmers rely on flood irrigation, which releases water into the field for a set period. One disadvantage to this irrigation technique is the manual observation and decision-making of how much water to release. Farmers cannot measure the exact amount of soil moisture, and due to uncertain rainfall, it is challenging to determine irrigation duration manually.

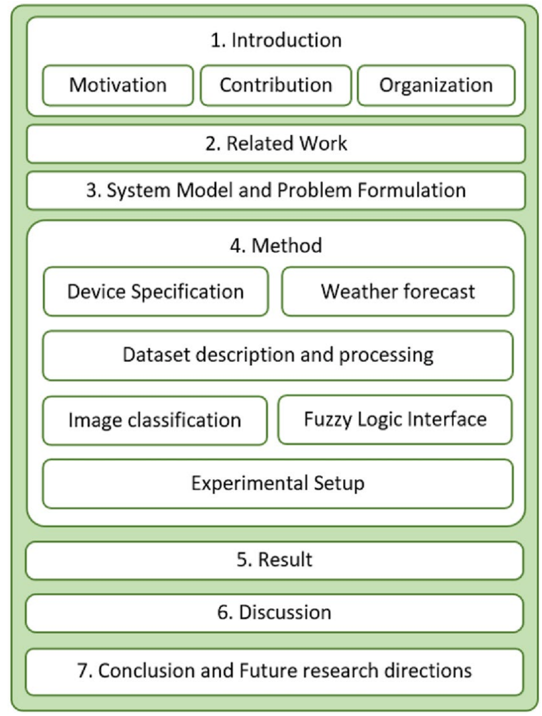


Fig. 2 Organization of paper

Motivations

The motivation for this article is mentioned below:

- Freshwater scarcity has been one of the world's most pressing issues. Thus, one of the most critical factors addressed by the authors is conserving water and minimizing water wastage while irrigating crops.
- Some of the essential nutrients plants require are present in the upper soil layers. However, these nutrients drain away due to water-logging and dehydration, which erodes the soil. Therefore, providing the right amount of water for the correct duration is crucial for improving crop yields.
- The manual operation of the water pump causes complications for farmers, leading to either over-irrigation or under-irrigation of the plants. In addition, farmers cannot determine the amount of moisture in the soil or the current weather conditions. Therefore, the authors proposed an automated system for irrigating crops that effectively addresses soil moisture, meteorological conditions, and crop health.

Research contributions

Watering crops at an adequate level is a difficult task due to multiple parameters taking part, making it hard to put them together and decide the duration for irrigation. Researchers worldwide have worked on building smart irrigation system that focuses only on soil moisture or weather conditions. However, no system exists that considers the image of the crop(to classify as a droop or healthy) along with the aforementioned parameters(soil moisture and weather predictions) that influence irrigation duration. In a nutshell, the following are the crisp objectives of the paper.

- We present a comprehensive study of the numerous smart irrigation systems currently in use and their drawbacks.
- A new fuzzy logic model, consisting of a soil moisture sensor, weather API, and image classification using a convoluted neural network (CNN) model, is proposed to decide the time for adequate watering of the field.
- The appropriate membership functions are chosen, and their boundary values are tuned to refine the fuzzy logic controller.
- Performance evaluation of the CNN model is conducted using various evaluation metrics, such as accuracy, F1-score, and curve of loss over the epochs.

Organization

The organization of the paper is shown in Fig. 2. The next section discusses the related work that has been done in the domain of smart irrigation. The following section discusses the system model and problem formulation for the autonomous water irrigation system. The next section describes the workflow of the proposed model. The following section discusses the results and evaluates the performance of the proposed model. The next section compares the proposed model with existing models presented by different papers. Finally, the last section discusses the future research directions followed by the conclusion of the paper.

Related work

The advent of IoT devices has led to significant advancements in many fields, and agriculture is no exception. Scientists have recently tried to incorporate machine learning and deep learning models with IoT devices to analyze the input data and predict the required output efficiently. We can design tools and technologies that are smart and can aid in reducing both environmental degradation and human efforts by properly using science and technology. A solar-based smart irrigation system is proposed in Uddin et al (2012). The authors have used solar electricity as the sole source of power. Sensors are installed in the paddy field to monitor the water level continuously and notify the user. The user can operate the motor based on the water level by sending an appropriate message from a remote location. In Pavithra and Srinath (2014), authors have come up with the idea of a mobile application for the automatic irrigation control system for efficient use of resources and crop planning. This application uses the general packet radio service (GPRS) feature of a mobile phone as a solution for the irrigation control system. The authors additionally employed the global system for mobile communication (GSM) to notify the user of the precise field condition. The user receives the information in the form of a short message service when they request it (SMS). Authors in Rajendranath and Hency (2015) built an automated irrigation system using a temperature sensor, humidity sensor, and soil moisture sensor. All these sensors are interfaced with the micro-controller, and the entire unit is placed under the plant's root zone. The authors have tested this irrigation system under different temperatures and humidity levels of different plants under normal and wet conditions. Đuzić and Đumić (2017) depicts the system that uses sensor technology with a micro-controller and other electronics that sense the moisture level of soil and irrigate the plant only if needed. However, these systems irrigate the field primarily based on soil moisture but do not consider real-time data and crop status. Authors of Kashyap et al (2021) proposed DLiSA (deep learning neural network-based IoT-enabled intelligent irrigation system for precision agriculture) that takes into account soil moisture, climate information, rainfall depth, and crop type to predict the required irrigation time. Authors of Zhang et al (2018) built a water-saving irrigation system using agricultural IoT (internet of things). This technology would be able to process weather data in real-time. In addition, the system could make irrigation selections based on the amount of water that has been dissipated. An automation system for sprinkler irrigation using a wireless sensor network is proposed in Nagarajan and Minu (2018). The proposed system uses ZigBee and GPRS technologies to transmit and store data. The system monitors soil conditions with the help of sensors, such as humidity sensors, pH sensors, and temperature sensors. The data sensed by the sensors is then sent to the controller for the process of monitoring. Barkunan et al (2019) proposed an irrigation system to water a plant as per the type of plant, as some crops or plants need a variable amount of water as they grow. The system begins by taking soil images with a smartphone, calculating wetness levels, and transmitting this information to the micro-controller via the GSM module. The controller then decides the irrigation duration and rate and sends the status of the field to the user's mobile phone. According to the authors, the system saves nearly 41.5% and 13% of water compared to traditional flood and drip irrigation methods, respectively. In Hamami and Nassereddine (2020), an automated irrigation system is proposed to save water and improve the performance of the irrigation system. The system uses soil and weather sensors to measure soil parameters and check the weather conditions.

System model and problem formulation

System model

There is a field of area A which is divided into i parts of equal area $\{A_1, A_2, \ldots, A_i\} \in A$. There are soil moisture sensors (S) such that $\{S_1, S_2, \ldots, S_i\} \in S$ for each area $A_j \in A$. There are cameras (C) such that $\{C_1, C_2, C_3, \ldots, C_i\} \in C$ and water pumps (W) such that $\{W_1, W_2, \ldots, W_i\} \in W$. In general, each area A_j is equipped with a soil moisture sensor S_j to measure the soil moisture, a camera C_j for capturing an image of the crop, and a water pump W_j which is used for irrigation ($0 < j \leq i$).

For sensors $S_j \in S$ in area A_j, $\{R_1, R_2, ..., R_n\} \in R$ are the readings taken within a time interval T which are averaged to a value λ_j. Images captured by camera C_j are indicated by $I_j \in I$. ϕ represents the condition of the crop(droop or healthy) classified by the image I_j where $\phi \in \{0, 1\}$. For $A_j \in A$, the precipitation and precipitation probability are indicated by ρ and θ, respectively. Since all of the regions are contained within a single field, the authors assume that the values of ρ and θ are the same. Table 1 represents the list of symbols used in the proposed model.

Table 1 List of symbols

Symbol	Name
A	Area
S	Set of soil moisture sensors
C	Set of cameras
W	Water pump
I	Image of plant
R	Readings from sensors
λ	Value of average soil moisture
ρ	Value of precipitation (in mm)
θ	Value of precipitation probability
ϕ	Crop status (0, 1)
Ω	Irrigation duration

Problem formulation

A farmer F growing crop in an area A has to manually decide, based on his experience and the wetness of the soil, to irrigate the area A. Most of the time, manual estimation of irrigation time results in over-irrigation or under-irrigation. Farmers cannot foresee rain in advance, so heavy rain could harm the crop if the field is already irrigated. Because many farmers in India have a limited quantity of water, efficient water management is also a significant concern.

As stated in Eq. 1, farmer F irrigates the area A_j with λ_j, ρ, θ, and ϕ_j as the average soil moisture of that area, precipitation, precipitation probability, and crop status (droop or healthy) of that area and the irrigation duration is T_x minutes.

$$F \xrightarrow{\text{irrigates}} A_j(\lambda_j, \rho, \theta, \phi_j) \text{ for } T_x \text{ minutes.} \tag{1}$$

$T_{\min}$ indicates the minimum amount of time that the plants should be watered in order for the plant to grow, and $T_{\max}$ indicates the maximum duration that the plants should be watered such that it does not cause damage to the crop or the soil. Waterlogging and dehydration are indicated by the letters α and β, respectively.

As stated in Eq. 2, water-logging (α) is caused by providing water for longer than the threshold $T_{\max}$, whereas dehydration (β) is caused by providing water for less than the minimum time ($T_{\min}$).

$$(T_x > T_{\max} \implies \alpha) \vee (T_x < T_{\min} \implies \beta). \tag{2}$$

The optimal irrigation time (Ω) must be estimated for better crop conditions. According to Eq. 3, the optimal irrigation duration, Ω, should prevent both α and β circumstances in area A. The right value of Ω must be chosen according to all the parameters ($\lambda_j, \rho, \theta, \phi_j$) for the water to be used as efficiently as possible, and water wastage must be minimum.

$$\Omega = \{T_y \mid (T_{\min} < T_y < T_{\max}) : (\neg\alpha \wedge \neg\beta) \text{ for } A_j(\lambda_j, \rho, \theta, \phi_j)\}. \tag{3}$$

The goal is to compute the value of Ω for which the motor W will run to irrigate the field.

Methods

The following paragraphs describe the flow of the proposed model. The section 'Device specification' describes the specifications of all the used devices along with their connection.

The proposed model comprises of multiple soil moisture sensors, a camera, and a motor. These sensors and cameras are connected to an IoT device (Raspberry Pi) P. First, the moisture content of the soil is detected by a capacitive soil sensor S_j. Then, the soil moisture value is fed into P via an I^2C bus. Moreover, precipitation (ρ) (in mm) in the past 24 hours and precipitation probability (θ) on the current day at that particular location are extracted from a weather API (2021) as in Algorithm 1. In addition, as illustrated in Algorithm 2, photos of the crops are captured, and a deep learning model (DenseNet201) is used to classify the plants as healthy or droop. Finally, the fuzzy system receives all the collected and processed data and outputs the irrigation duration Ω. The motor is activated for Ω seconds by the Raspberry Pi P. The Raspberry Pi powers off the motor after Ω seconds, and the system is programmed to sleep till the next day. Figure 3 comprehensively showcases the overall information flow of the proposed mode. The aggregation of soil moisture, precipitation probability, precipitation, crop status, and fuzzy logic for the computation of irrigation time and operation of the water pump is described by the Algorithm 3. Figure 4 depicts the schematic arrangement of the proposed model. The values of the parameters computed by Algorithm 3 are displayed on a Raspberry Pi touch display.

Device specification

Raspberry Pi model B

Figure 5a shows the Raspberry Pi model B, which is used for controlling all the sensors, processing the images, controlling the relay switch, and running the system. Raspberry Pi can be considered a mini-computer used for all the computation and processing. It consists of a Quad-core 64-bit 1.5 GHz processor with 8GB RAM. It is powered through a 5V DC via a USB-C connector (minimum 3A*). It consists of 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless Gigabit Ethernet for internet access. It also includes two USB 2.0 and two USB 3.0 connectors. The Raspberry Pi Touch Display uses a 2-lane Mobile Industry Processor Interface

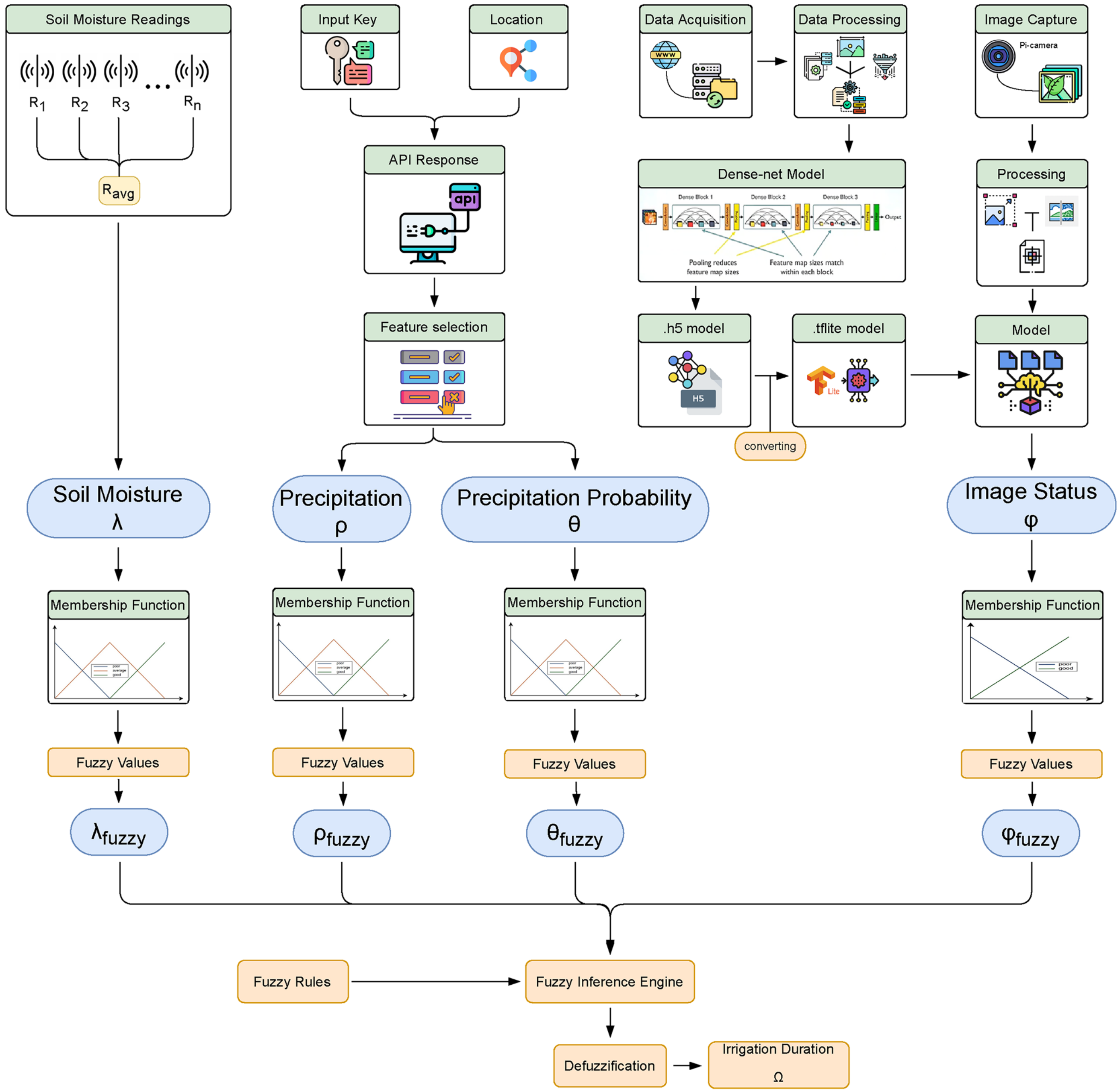


Fig. 3 Model flowchart

(MIPI), Display Serial Interface (DSI) display connector to display the interface, and a 2-lane MIPI Camera Serial Interface (CSI) camera port is available to link the camera to the Raspberry Pi. The Raspberry Pi has a total of 40 pins, including 26 GPIO (general purpose input output) pins, eight ground pins, four voltage (3.3 V and 5 V) connectors, and 2 EEPROM (Electronically Erasable Programmable Read-Only Memory) pins (Device specs for raspberry pi 2021).

Capacitive soil moisture sensor

The ATSAMD10 chip in the capacitive sensor is a built-in capacitive touch measurement device that provides a range of readings from 200 (very dry) to 2000 (extremely wet). The sensor has five pins: 3–5 V power, ground, I^2C SDA, and I^2C SCL. Figure 5e depicts the capacitive soil moisture sensor used for measuring the moisture content of the soil. Resistive soil moisture sensors are susceptible to corrosion, which causes measurement errors and only provides a binary output. Therefore, the capacitive moisture sensor is used to

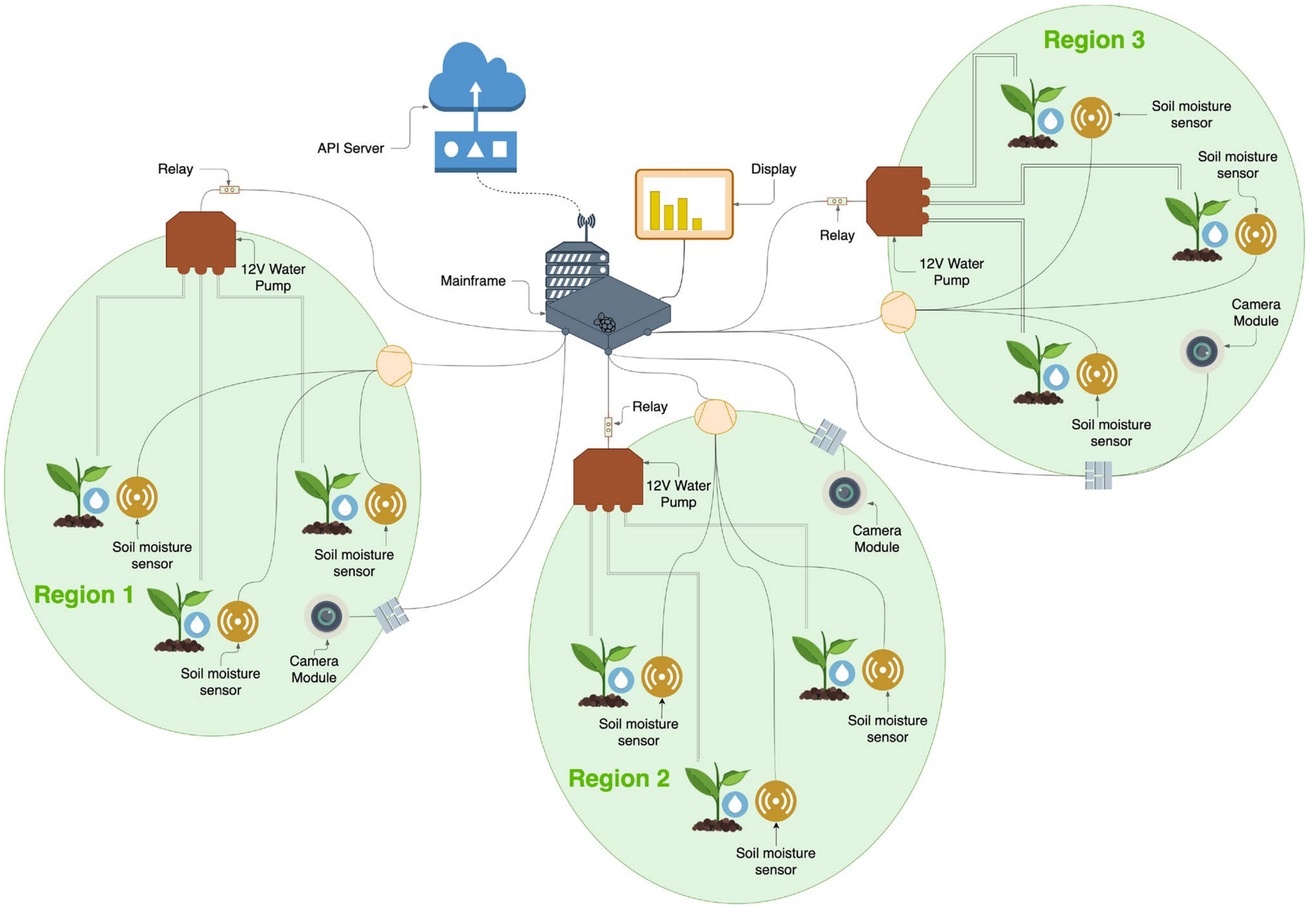


Fig. 4 Schematic arrangement of proposed model

measure the soil moisture as it is corrosion-resistant and also provides a precise value of soil moisture (Shawn 2021).

Camera module V2

The images of the plants are captured using a Raspberry Pi camera module version 2, as shown in Fig. 5f. A Sony IMX219 8-megapixel sensor is used in the camera module. The Raspberry Pi camera is connected to the Raspberry Pi via a ribbon wire.

Water pump and relay

A 200 psi, 2.5 A, 12 V DC water pump with a flow rate of 8 liters per minute is employed for water irrigation. The motor is powered by an adapter that the Raspberry Pi controls. A 12 V relay module controls the water pump. The relay switch comprises NO (normally open), NC (normally closed), and common terminal (COM). On the opposite side, it has a 5 V VCC, Ground, and one signal pin (relay switch 2021). The Raspberry Pi activates or deactivates the motor by triggering the relay. An AC connection uses a 12 V 6A adapter to power the DC motor. The DC motor and relay switch utilized are depicted in Fig. 5b, c), respectively.

Raspberry Pi touch display

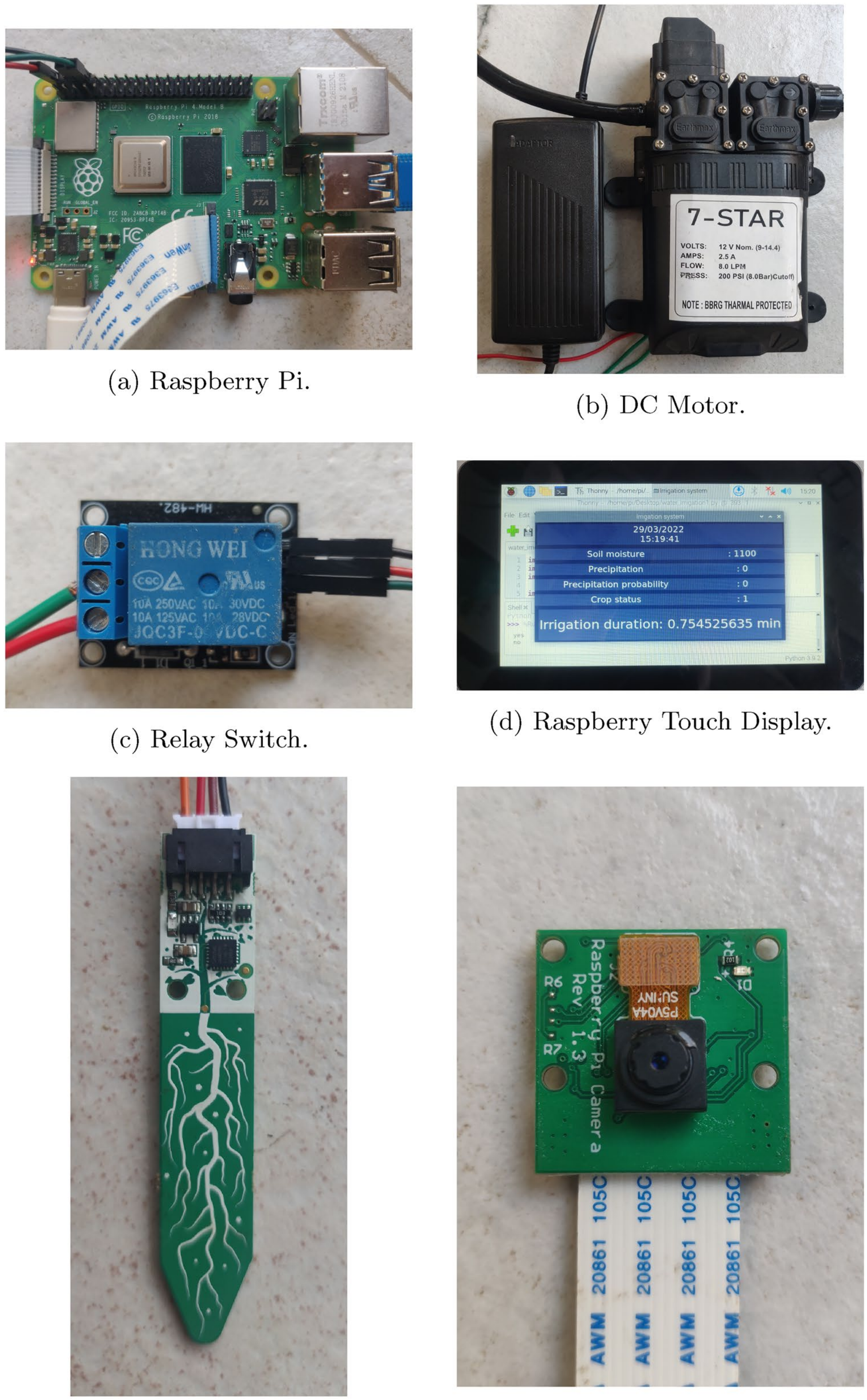
An interface with all of the parameters and irrigation duration would be displayed on the Raspberry Pi touch display. The display is 7 inches in size (diagonally). 800 (RGB) × 480 pixels is the display format. It has a DSI port that is used to connect to the Raspberry Pi (Touch Display 2021). The touch display shown in Fig. 5d is used to display the values of all the parameters as well as the irrigation duration.

Weather forecast

An application programming interface (API) is a service provided to an application on demand. An API key and the website's URL are required to send the request to the API. The API responds by returning a dictionary containing relevant data, from which the necessary data is retrieved.

When it comes to watering plants, one of the elements to consider is the amount of precipitation (rain). If precipitation

Fig. 5 Various hardware components used for building the water irrigation system



(a) Raspberry Pi.

(b) DC Motor.

(c) Relay Switch.

(d) Raspberry Touch Display.

(e) Moisture sensor.

(f) Raspberry Pi Camera Module.

on a given day is significant enough, the irrigation period must be low. Therefore, we considered two variables: the amount of precipitation that occurred in the previous 24 h and the probability of precipitation (likelihood of rain on a given day). RapidAPI is a weather API that returns several weather parameters, such as temperature, humidity, and many more, API used for weather parameters (2021) from which the aforementioned values are extracted and provided to the proposed module's Fuzzy logic controller. With area, A as the input parameter, Algorithm 1 provides the procedure for extracting precipitation and precipitation probability.

Algorithm 1 API

Input: A_x
Output: ρ, θ

```
procedure API(I)
    Get an API key to use the API service.
    Generate the request for information using the API key.
    Receive the response from server.
    Extract ρ and θ from the response.
    return ρ, θ
end procedure
```

Data set description and processing

The data set consists of droop and healthy plant images. A plant is termed droop when its leaf shrivels due to water deficiency. The authors developed one manually because a data set for droop plants was unavailable. It contains 110 droop images and 125 healthy images in .png format.

When it comes to deep learning, data pre-processing is crucial as it aids in the removal of anomalies and null data, resulting in more accurate results. It is essential to remove such data because failing to do so can result in poor model training. Furthermore, the model may not be able to give generalized results due to the small data set size. Therefore, data augmentation strategies are employed to avoid this problem. We employed data augmentation techniques such as random horizontal flipping and random rotation. A batch size of 16 images is taken while training a model. In addition, the array of images is normalized using the Min–Max approach into a range of −1 to 1.

Image classification

Apart from soil moisture and weather conditions, the plant's appearance also plays a crucial role in optimizing the irrigation duration. For example, plant leaves sometimes turn brown or yellow when deprived of water. In addition, leaves sometimes bend because of dryness since the cells of leaves cannot remain erect with less water (8 ways to tell when a houseplant needs water 2021). As a result, the condition of the plants is taken into account while selecting the watering time.

The camera captures an image of the plant, which is saved on the Raspberry Pi and classified into one of the categories. The Densenet201 deep learning model has only two possible outputs: "healthy" or "droop," which is used to calculate irrigation time.

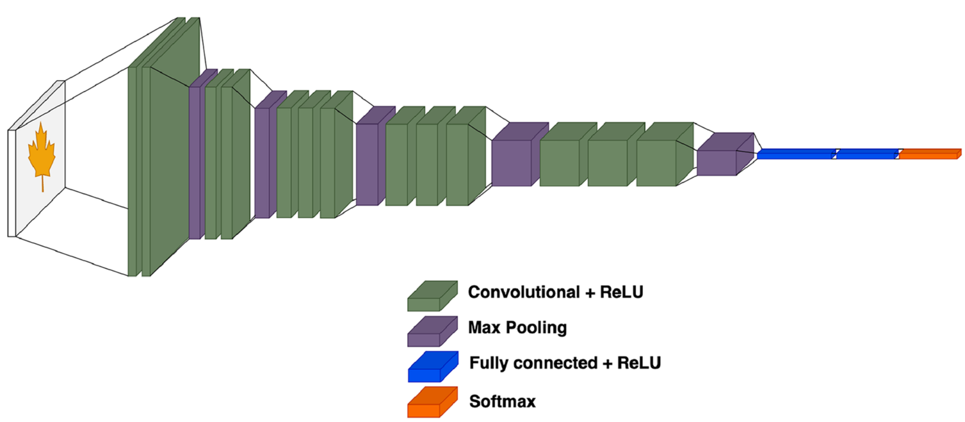


Fig. 6 Neural network

Algorithm 2 Image Classify

Input: I

Output: 0 *or* 1

```
procedure ImageClassify(I)
    Resize image to the required size.
    Load the TensorFlow Lite model and pass the image to it.
    The model returns the value between zero to one.
    The received value is passed through sigmoid function which returns the class
of the image.
    return 0 or 1
end procedure
```

The authors have used neural networks for image classification. Because neural networks can use many parameters and operate effectively even on small data sets, they are commonly used in machine learning and deep learning. As seen in Fig. 6, any model has several convolution layers, max-pooling, and activation functions that are utilized for training the model. Some layers must be eliminated as the learning advances to avoid the model over-fitting. In general, images are enormous and include too many features, and many of them are mostly redundant, so max-pooling is employed to avoid it. The neural network extracts the critical information using max-pooling, which minimizes the input size.

In recent times, deep learning has been quite effective in image classification. CNN is frequently used in image classification models among several deep learning approaches, such as ANN (artificial neural network) and RNN (recurring neural network). Various pretrained CNN models, such as MobileNet, EfficientNet, VGG16, and others, are available. Most of the CNN models are highly accurate in image classification and detection. The authors of Bondre and Sharma (2021) examined multiple CNN models in plant disease detection and concluded that deep learning is the best solution for enhancing disease detection and classification accuracy. The authors of Sirohi and Malik (2021) suggested hybrid models for disease detection in sunflower plants, concluding that the combination of Mobilenet and Vgg-16 utilizing ensemble learning outperformed other CNN models, such as AlexNet, InceptionV3, DenseNet-121, DenseNet201, Vgg-16, and MobileNet. Due to the lack of a large data set to train our model, building a CNN model from scratch would be under-fitting. Hence, the authors employed transfer learning to classify images. As the model does not have to learn all of the features from scratch, it helps to reduce data under-fitting.

We tested most of the aforementioned models because they were all demonstrated to produce an accurate classification. We chose DenseNet201 as it delivers the best accuracy, F1-score, and loss curve in our module. After training the Densenet201 model, we saved the model to *.h5* format file. We then converted the file to *.tflite* using Tensorflow Lite to deploy it on Raspberry Pi.

Fuzzy logic interface

An essential part of calculating the duration for which the motor should supply water to the plants is integrating all the factors (soil moisture content, meteorological conditions, and picture classification). The authors have used fuzzy logic to accomplish this, which produces a precise estimate of irrigation duration. The authors of Chen et al (2010) concluded that fuzzy logic is accurate in predicting irrigation duration. The final result (irrigation duration) is calculated using fuzzy logic, consisting of multiple fuzzy rules. Soil moisture, precipitation probability, precipitation, and image categorization are provided as inputs into the fuzzy logic. Fuzzification, fuzzy inference engine, and defuzzification are the three steps in fuzzy logic.

Fuzzification

This phase involves dividing or grouping the value into one of the fuzzy set's categories. Fuzzy sets are categories, such as high, average, low, and many more. A "fuzzified value" is a value that is created using a membership function. Membership functions are simple equations used to categorize the input value into categories from the fuzzy set when it overlaps with the function. Examples are the triangular

Table 2 Fuzzy rules

Sr. no.	Soil moisture	Precipitation probability	Precipitation	Crop status	Irrigation duration
Rule 1	Dry	Low	Low	Droop	High
Rule 2	Dry	Low	Low	Healthy	High
Rule 3	Dry	Low	Normal	Droop	Normal
Rule 4	Dry	Low	Normal	Healthy	Normal
Rule 5	Dry	Low	High	Droop	Low
Rule 6	Dry	Low	High	Healthy	Low
Rule 7	Dry	Normal	Low	Droop	Normal
Rule 8	Dry	Normal	Low	Healthy	Normal
Rule 9	Dry	Normal	Normal	Droop	Normal
Rule 10	Dry	Normal	Normal	Healthy	Low
Rule 11	Dry	Normal	High	Droop	Low
Rule 12	Dry	Normal	High	Healthy	Low
Rule 13	Dry	High	Low	Droop	Normal
Rule 14	Dry	High	Low	Healthy	Low
Rule 15	Dry	High	Normal	Droop	Normal
Rule 16	Dry	High	Normal	Healthy	Low
Rule 17	Dry	High	High	Droop	Low
Rule 18	Dry	High	High	Healthy	Low
Rule 19	Medium	Low	Low	Droop	High
Rule 20	Medium	Low	Low	Healthy	Normal
Rule 21	Medium	Low	Normal	Droop	Normal
Rule 22	Medium	Low	Normal	Healthy	Normal
Rule 23	Medium	Low	High	Droop	Low
Rule 24	Medium	Low	High	Healthy	Low
Rule 25	Medium	Normal	Low	Droop	High
Rule 26	Medium	Normal	Low	Healthy	Normal
Rule 27	Medium	Normal	Normal	Droop	Normal
Rule 28	Medium	Normal	Normal	Healthy	Low
Rule 29	Medium	Normal	High	Droop	Low
Rule 30	Medium	Normal	High	Healthy	Low
Rule 31	Medium	High	Low	Droop	Normal
Rule 32	Medium	High	Low	Healthy	Low
Rule 33	Medium	High	Normal	Droop	Normal
Rule 34	Medium	High	Normal	Healthy	Low
Rule 35	Medium	High	High	Droop	Low
Rule 36	Medium	High	High	Healthy	Low
Rule 37	Wet	Low	Low	Droop	Normal
Rule 38	Wet	Low	Low	Healthy	Normal
Rule 39	Wet	Low	Normal	Droop	Normal
Rule 40	Wet	Low	Normal	Healthy	Low
Rule 41	Wet	Low	High	Droop	Low
Rule 42	Wet	Low	High	Healthy	Low
Rule 43	Wet	Normal	Low	Droop	Normal
Rule 44	Wet	Normal	Low	Healthy	Low
Rule 45	Wet	Normal	Normal	Droop	Normal
Rule 46	Wet	Normal	Normal	Healthy	Low
Rule 47	Wet	Normal	High	Droop	Low
Rule 48	Wet	Normal	High	Healthy	Low
Rule 49	Wet	High	Low	Droop	Low
Rule 50	Wet	High	Low	Healthy	Low

Table 2 (continued)

Sr. no.	Soil moisture	Precipitation probability	Precipitation	Crop status	Irrigation duration
Rule 51	Wet	High	Normal	Droop	Low
Rule 52	Wet	High	Normal	Healthy	Low
Rule 53	Wet	High	High	Droop	Low
Rule 54	Wet	High	High	Healthy	Low

membership function, trapezoidal membership function, and other membership functions.

Fuzzy inference engine

The fuzzy inference engine uses the fuzzified values computed in the previous phase as inputs. The fuzzy inference engine accepts two more inputs: fuzzy sets and fuzzy rules. For the established categories, fuzzy rules are defined. A fuzzy rule should be defined for each permutation of the fuzzy sets. Because there are three categories for soil moisture, precipitation, and precipitation probability each, and two categories for image classification, the authors created $54(3 \times 3 \times 3 \times 2 = 54)$ fuzzy rules as shown in Table 2. As a result, 54 rules cover all possible category combinations. The irrigation duration is the output of the fuzzy inference engine. The output can have many values, and the third step is used to combine them into a single value.

Defuzzication

The multiple values generated in the previous steps must be combined to provide a single output value that can be defuzzified. Then, the lambda cut, centroid method, weighted average approach, and other defuzzification methods are utilized. Finally, a single output value or crisp output is achieved after utilizing one of the approaches.

The irrigation duration is generated as a crisp output once the fuzzy logic is applied, then utilized to trigger the relay switch. Relay switches manage the circuit by allowing it to open and close as needed. Finally, the switch is turned on for the calculated duration, triggering the DC motor to water the plants.

Algorithm 3 Irrigation Duration System

Input: $\{R_1, R_2, ..., R_n\}$, I, A_j
Output: Irrigation Duration (Ω)

procedure IrrigationDuration($\{R_1, R_2, ..., R_n\}, I, A_j$)
    $\lambda \leftarrow (R_1 + R_2 + \ldots + R_n) \, / \, n.$
    $\rho, \theta \leftarrow API(A_j)$ (Alg.1)
    $\phi \leftarrow$ `ImageClassify(I)` (Alg.2)
    According to the membership function(MF) fuzzified values of the variables is calculated.
    $\lambda_{fuzzy} = \mathtt{MF}(\lambda)$
    $\rho_{fuzzy} = \mathtt{MF}(\rho)$
    $\theta_{fuzzy} = \mathtt{MF}(\theta)$
    $\phi_{fuzzy} = \mathtt{MF}(\phi)$
    $\Omega = \mathtt{FuzzyController}(\lambda_{fuzzy}, \rho_{fuzzy}, \theta_{fuzzy}, \phi_{fuzzy})$
    Operate motor W for time Ω.
end procedure

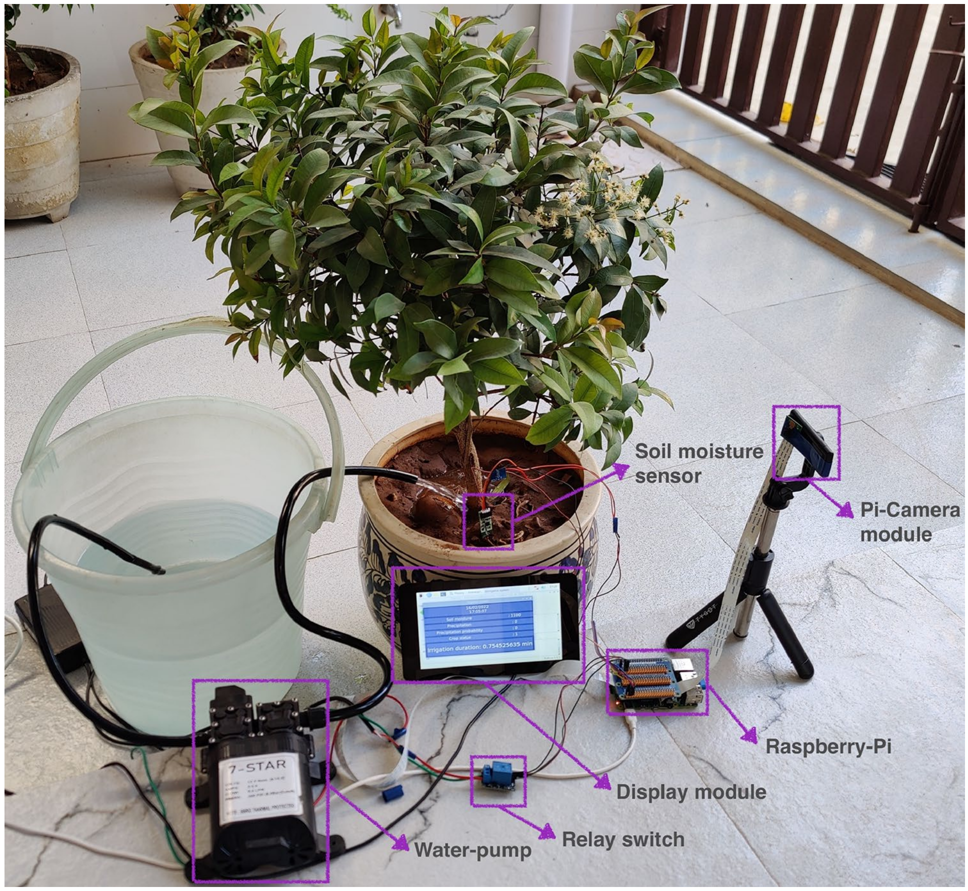


Fig. 7 Connection of all devices

Table 3 Hyperparameter tuning

Hyperparameter	Value
Batch size	16
Validation split	0.2
Epochs	30
Dropout	0.2
Loss function	Binary cross entropy
Optimiser	Adam
Learning rate	0.00001

Experimental setup

The components are wired together as indicated in Fig. 7. The Raspberry Pi can be thought of as a controller that controls all other devices. A USB-C adaptor is required to power it. The moisture sensor is attached to a 4 Pin JST PH 2mm Pitch Plug. The other ends of the JST plug are attached to the Raspberry Pi's V_in, ground, SCL, and SDA pins.

The camera module is linked to the Raspberry Pi's 2-lane MIPI CSI camera interface.

The relay switch's GND, VCC, and signal pins connect to the Raspberry Pi's ground, 5V, and GPIO pin 8. The motor is connected to the NC and COM pins on the relay's other side. The adaptor is linked in series with the motor for the power supply. When the signal pin of the relay is set to high, the motor turns on. The display is attached to the Raspberry Pi's 2-lane MIPI DSI display interface via a ribbon cable.

Computing facilities

We used CoLab's GPU runtime environment to reduce the time required for all computations. Google colab has a 2.20 GHz Intel(R) Xeon(R) processor, 66 GB of hard drive storage, and 13 GB of RAM.

Hyper-parameters of model

Hyperparameters are crucial in the training of any model. These parameters are in addition to those obtained from the data set. Therefore, to increase the model's accuracy, it is critical to get the correct values for these parameters. The values of hyperparameters we utilized to improve the model's learning capability are shown in Table 3.

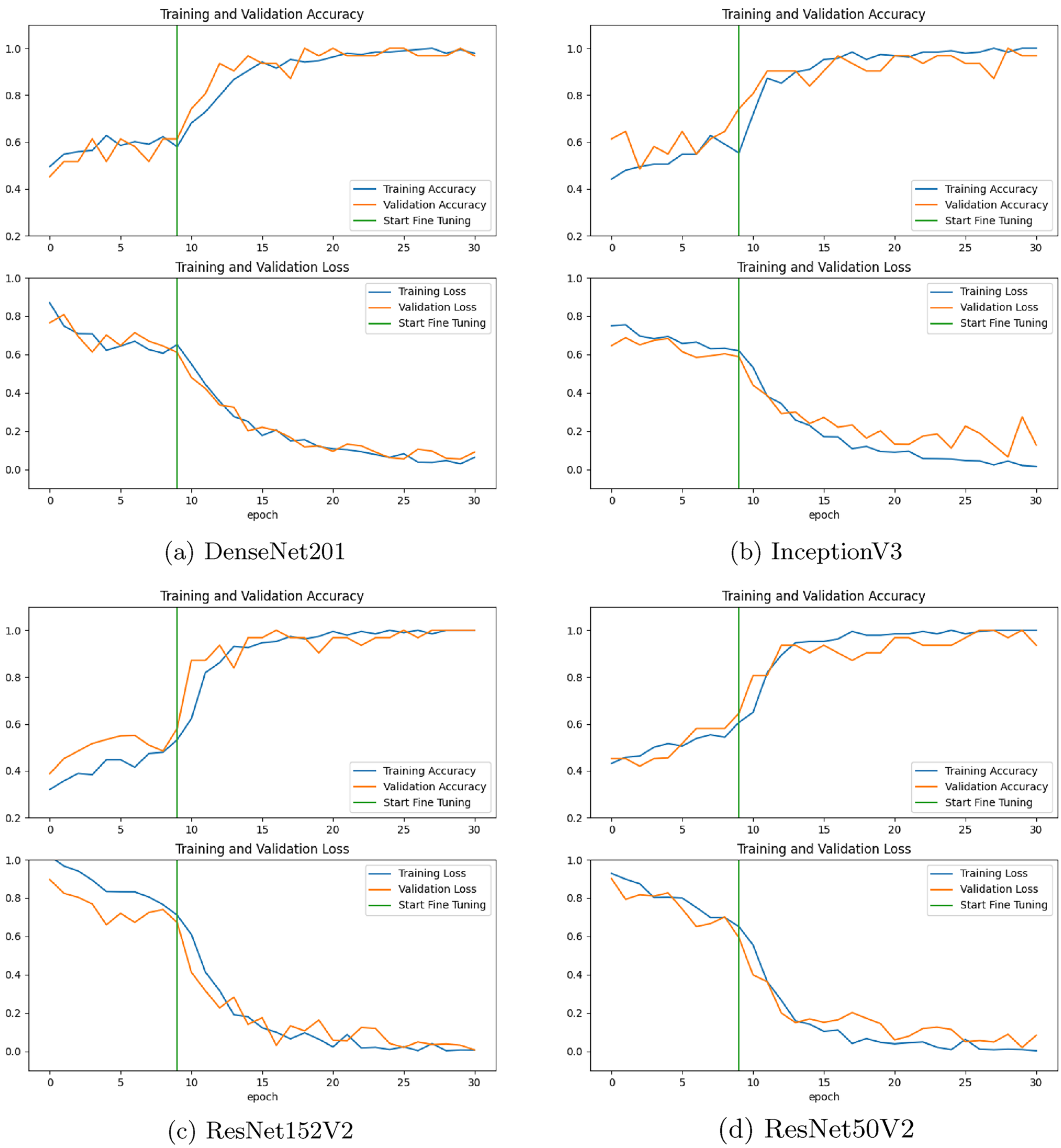


Fig. 8 Accuracy and validation loss of some eclectic models

Results

The main objective of our system is to optimize water usage along with automation which is achieved using fuzzy logic. The fuzzy logic in the proposed system requires four parameters: soil moisture, precipitation, precipitation probability, and crop status. The API directly feeds the values of precipitation and precipitation probability to fuzzy logic. The soil moisture sensor takes five readings within a time interval of one second. The average value of the readings is sent to fuzzy logic. The crop image is passed through a CNN model (Densenet201 in our case) whose output (droop or healthy) is passed to the fuzzy module as an input. Fuzzy

Table 4 Comparison of different models

Name of model	Accuracy	F1-score
DenseNet201	100.00	1.00
ResNet152V2	100.00	1.00
ResNet101V2	93.75	0.94
DenseNet169	93.75	0.94
ResNet50V2	93.75	0.92
NasNetLarge	93.75	0.94
MobileNet	87.50	0.86
InceptionV3	87.50	0.83
DenseNet121	87.50	0.83
Xception	87.50	0.83
NasNetMobile	81.25	0.77

Table 5 Fuzzy logic output for different conditions

Soil moisture λ	Precipitation probability ρ	Precipitation θ	Crop status ϕ	Irrigation duration Ω
550	20	20.0	1	2.87
1100	60	15.0	1	0.75
300	30	4.0	0	4.00
800	45	2.0	0	5.55
450	27	6.0	1	2.73
900	52	4.7	1	1.4

logic now calculates the irrigation duration according to the provided rules in Table 2.

We explored models such as Densenet201, ResNet152V2, and others for image classification. We observe the accuracy and loss curves of the CNN model and decide which model performs best for the given data set. A comparative analysis of pretrained CNN models in terms of overall accuracy and F1-score is presented in Table 4 which depicts that Densenet201, InceptionV3, Resnet152V2, and Resnet50V2 obtained better training and validation accuracy over 30 epochs than other models. As accuracy is not adequate to measure a model's performance in deep learning due to the risk of over-fitting, we employed the F1-score as an evaluation metric.

Table 7 Methodology comparison

References	Weather API	Fuzzy logic	Deep learning
Krishnan et al (2020)	–	✓	–
Geethamani et al (2021)	–	–	–
Proposed system	✓	✓	✓

However, evaluation metrics are insufficient in and of itself to determine a model's effectiveness; therefore, we looked at loss and accuracy curves to see which model would be ideal for our module. The loss and accuracy curves for the four models are shown in Fig. 8. There was a gradual decrease in loss and an increase in accuracy in each of the four graphs. Minimal spikes were found in the DenseNet201 model, as shown in Fig. 8, along with maximum F1-score, accuracy, and minimum loss. Therefore, we have incorporated it into our module for image classification.

DenseNet201 uses a condensed network to create models that are simple to train and very parametrically efficient. This adds variation to the next layer's input and improves performance. We chose DenseNet201 above the other three models for these reasons. The basic learning rate for the model was set at 0.0001 at the outset, and adaptive moment estimation (Adam) was employed for optimization since it is simple to implement and modifies the value of the learning rate in real-time as each epoch passes. The loss function is calculated using the binary cross-entropy approach, virtually perfect for binary classification issues.

Table 5 indicates some of the soil moisture values, precipitation, precipitation probability, and image classification and corresponding output for the irrigation.

Discussion

Some of the earlier research in this field relied entirely on soil moisture to determine how long to irrigate (Hadi et al 2020; Rivai et al 2019; Sayanthan et al 2018). However, they did not take into account current meteorological conditions

Table 6 Hardware comparison

Source	Sensor	pH sensor	DHT11	Pi camera module	Relay switch	Rain sensor
Krishnan et al (2020)	Resistive	–	✓	–	✓	✓
Geethamani et al (2021)	Resistive	✓	✓	–	✓	–
Proposed system	Capacitive	–	–	✓	✓	–

or forecasts for the future. After that, systems incorporating a soil moisture sensor as well as weather prediction were presented (Velmurugan 2020). Furthermore, temperature and humidity were also considered by several researchers when estimating irrigation duration (Nagarajan and Minu 2018).

In addition to considering the aforementioned characteristics, the paper's authors have also considered the crop images for computing the irrigation duration.

Table 6 shows the comparison of various hardware components used by authors of Geethamani et al (2021), Krishnan et al (2020) and the components used by proposed system. The authors in Geethamani et al (2021), Krishnan et al (2020) employed a resistive soil moisture sensor, which corroded with time, resulting in measurement mistakes. As a result, the system had to be re-calibrated more frequently, and the sensors had to be replaced. Our research uses a capacitive soil moisture sensor to address these issues.

Table 7 demonstrates a comparison of the authors' methodologies of Geethamani et al (2021), Krishnan et al (2020) and the methodology of the proposed system. The authors in Geethamani et al (2021) have taken a single image, and Krishnan et al (2020) have not captured any photographs of the plants for classification. In contrast, we have captured images of the crops every 24 h, giving the most up-to-date crop condition and optimal irrigation duration. Furthermore, the authors in Geethamani et al (2021) did not utilize any particular logic to combine all of the parameters generated. However, we used fuzzy logic to combine all the parameters and compute the optimum duration for watering the crops as given in Table 7. The authors of Krishnan et al (2020) use the rain sensor to detect rainfall. Rain sensors are switches that activate on rainfall and thus give binary information about whether it is raining or not. In contrast, when we use weather API, we get the precipitation of the current day (in mm) and the weather prediction for the next day.

Conclusions and future research directions

This research presents a fuzzy logic and IoT-based automated system for water irrigation. The system takes soil moisture, weather parameters, and crop quality as input. We considered precipitation and precipitation probability extracted from a weather API and crop quality (droop or healthy) to optimize the system. Various deep learning models, such as DenseNet201, InceptionV3, ResNet152, and ResNet50 were also used to categorize the crop as droop or healthy. Out of all the deep learning models, we discovered that DenseNet201 provided the most accurate classification (F1-score: 1). All these parameters are then passed into our fuzzy system controller, deciding the exact time to irrigate the crop. To check and validate the performance of our system, we ran several experiments by varying input parameters. The goal was to optimize the irrigation duration by varying input parameters, which is evident from the results. This automated system can assist in reducing water wastage and the need for manual monitoring.

This system can further be extended by incorporating a few other parameters, such as water need according to crop type and water availability. In addition, the concept may be improved by including a GSM module that allows farmers to receive notifications.

Data availability My manuscript has no associated data.

Declarations

Conflict of interest None declare.

References

8 ways to tell when a houseplant needs water (2021) 8 ways to tell when a houseplant needs water. https://smartgardenguide.com/ways-to-tell-when-a-houseplant-needs-water. Accessed 8 Jan 2022

Agriculture employment in India (2021) Agriculture employment in India. https://data.worldbank.org/indicator/SL.AGR.EMPL.ZS?locations=IN. Accessed 26 Jan 2022

Agriculture in India (2021) Agriculture in India. https://www.ibef.org/industry/agriculture-india.aspx. Accessed 26 Jan 2022

API used for weather parameters (2021) API used for weather parameters. Accessed 26 Jan 2022

Barkunan S, Bhanumathi V, Sethuram J (2019) Smart sensor for automatic drip irrigation system for paddy cultivation. Comput Electr Eng 73:180–193

Bondre S, Sharma AK (2021) Review on leaf diseases detection using deep learning. In: 2021 second international conference on electronics and sustainable communication systems (ICESC), pp 1455–1461. https://doi.org/10.1109/ICESC51422.2021.9532697

Chen Q, Li J, Gao Y (2010) Study on applying fuzzy inference to single factor prediction method for precipitation irrigation requirement forecast. In: 2010 2nd international workshop on intelligent systems and applications. IEEE, pp 1–5

Device specs for raspberry pi (2021) Device specs for raspberry pi. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/. Accessed 26 Jan 2022

Đuzić N, Đumić D (2017) Automatic plant watering system via soil moisture sensing by means of suitable electronics and its applications for anthropological and medical purposes. Coll Antropol 41(2):169–172

Fresh water in worldwide (2021) Fresh water in worldwide. https://en.wikipedia.org/wiki/Water_distribution_on_Earth#:~:text=The%20total%20volume%20of%20water,and%202.5%25%20being%20fresh%20water. Accessed 26 Jan 2022

Geethamani R, Jaganathan S, Murugappan S et al (2021) Implementation of smart agricultural monitoring system using raspberry pi. In: 2021 innovations in power and advanced computing technologies (i-PACT), pp 1–6. https://doi.org/10.1109/i-PACT52855.2021.9696484

Global warming causing climate changes (2021) Global warming making climate unpredictable. https://www.nationalacademies.org/

based-on-science/climate-change-global-warming-is-contributing-to-extreme-weather-events. Accessed 26 Jan 2022
Hadi MS, Nugraha PA, Wirawan IM, et al (2020) Iot based smart garden irrigation system. In: 2020 4th international conference on vocational education and training (ICOVET). IEEE, pp 361–365
Hamami L, Nassereddine B (2020) Application of wireless sensor networks in the field of irrigation: a review. Comput Electron Agric 179(105):782
Kashyap PK, Kumar S, Jaiswal A et al (2021) Towards precision agriculture: Iot-enabled intelligent irrigation systems using deep learning neural network. IEEE Sens J 21(16):17479–17491
KPMG (2010) Water sector in India: overview and focus areas for the future
Krishnan RS, Julie EG, Robinson YH et al (2020) Fuzzy logic based smart irrigation system using+ internet of things. J Clean Prod 252(119):902
Morillo JG, Martin M, Camacho E et al (2015) Toward precision irrigation for intensive strawberry cultivation. Agric Water Manag 151:43–51. https://doi.org/10.1016/j.agwat.2014.09.021
Mulenga R, Kalezhi J, Musonda SK et al (2018) Applying internet of things in monitoring and control of an irrigation system for sustainable agriculture for small-scale farmers in rural communities. In: 2018 IEEE PES/IAS PowerAfrica, pp 1–9. https://doi.org/10.1109/PowerAfrica.2018.8521025
Nagarajan G, Minu R (2018) Wireless soil monitoring sensor for sprinkler irrigation automation system. Wirel Pers Commun 98(2):1835–1851
National Geographic Data (2020) Freshwater crisis. https://www.nationalgeographic.com/environment/article/freshwater-crisis. Accessed 25 May 2022
Pavithra D, Srinath M (2014) Gsm based automatic irrigation control system for efficient use of resources and crop planning by using an android mobile. IOSR J Mech Civ Eng (IOSR-JMCE) 11(4):49–55
Rajendranath U, Hency VB (2015) Implementation of an automated irrigation system. In: Smart irrigation system. Presented at the IEEE sponsored 2nd international conference on innovations in information, embedded and communication systems (ICIIECS)
Relay Switch (2021) relay switch. https://bit.ly/3rFfpK2. Accessed 21 Dec 2021
Rivai M, Ashari M, Mustaghfirin MA et al (2019) Drip irrigation system using bldc motor-driven direct pumping and soil moisture sensor. In: 2019 international conference on computer science, information technology, and electrical engineering (ICOMITEE). IEEE, pp 221–226
Sayanthan S, Thiruvaran T, Kannan N (2018) Arduino based soil moisture analyzer as an effective way for irrigation scheduling. In: 2018 IEEE international conference on information and automation for sustainability (ICIAfS). IEEE, pp 1–4
Shawn (2021) Soil moisture sensor-getting started with arduino. https://www.seeedstudio.com/blog/2020/01/10/what-is-soil-moisture-sensor-and-simple-arduino-tutorial-to-get-started/
Sirohi A, Malik A (2021) A hybrid model for the classification of sunflower diseases using deep learning. In: 2021 2nd international conference on intelligent engineering and management (ICIEM), pp 58–62. https://doi.org/10.1109/ICIEM51511.2021.9445342
Touch Display (2021) Touch display. https://www.raspberrypi.com/products/raspberry-pi-touch-display/. Accessed 21 Dec 2021
Uddin J, Reza ST, Newaz Q et al (2012) Automated irrigation system using solar power. In: 2012 7th international conference on electrical and computer engineering. IEEE, pp 228–231
Velmurugan S (2020) An iot based smart irrigation system using soil moisture and weather prediction. Int J Eng Res Technol
Water Usage for agriculture irrigation (2021) Water usage for agriculture irrigation. https://www.oecd.org/agriculture/topics/water-and-agriculture/#:~:text=Agriculture%20irrigation%20accounts%20for%2070,on%20the%20sector%20and%20beyond. Accessed 26 Jan 2022
Zhang S, Wang M, Shi W et al (2018) Construction of intelligent water saving irrigation control system based on water balance. IFAC-PapersOnLine 51(17):466–471