Title: - Design & development of Autonomous Farming With Plant health Indication System

Abstract

Nearly three quarters of Indian families depend on agricultural income. Almost all crops suffer from plant diseases and insects like Tungro virus, moths and butterflies. Farmers find it difficult to identify actual disease. According to Indian crop survey, on an average scale of 80 percent of crops get damaged due to disease and insects. If the problems are known before then diseases can be prevented. There is a scope for improvement in these fields using IoT Technology. So, We are designing an IoT based robot that will monitor the crop and also the environment around the crop. This system uses machine learning technique to identify the problem and takes measures to prevent diseases and insects that harm the crops. Different sensors are used to study the environment, and a camera to detect the plant type and disease. A number of prototype guidance systems have been developed but have not yet proceeded to commercialization. Our crop monitoring system is efficient and affordable which will help the Indian farmers.

Introduction

Agriculture is a major occupation in India. Some of the major problems in the Indian agriculture are, rising of input costs, lack of skilled labours, scarcity of water resources and crop monitoring. To overcome these problems, the automation technology was used in agriculture.

The automation in agriculture could help farmers to reduce their efforts. The IoT plays a vital role in various fields such as industrial, medical, automobiles etc., The IoT technology is gradually increasing its productivity in agricultural field. The IoT technology is being developed for irrigation system, soil health monitoring, environment monitoring, etc., All of these functions have not yet performed using a single IoT platform.

Agronomic crops and plants are something on which economy exceedingly depends. In this 21st century, as the demography is growing rapidly, it is of genuine need to produce agronomic crops and plants. Bleeding edge developing solicitations progressively creating without extending its domain measure. Reusing a comparative land is one of the responses for more production, at any rate the farmland reusing technique does not work out each time on account of land conditions. We find a couple of issues with reusing of land to have more production. The genuine purposes for these are breaking down and tainting which harms the farmland inside out. The degradation of soil and being unusable in view of crumbling, around three million hectares of agrarian lands are lost every year. One of the effects goes to plants and crops disease.

This disease hampers production most significantly. The utilization of advancement like Internet of Things (IoT) in agronomic crops and plants could have exceptional impact. To detect the disease manually in plants is a critical job in farming field, as having ailment in plants are very regular. On the other hand, appropriate measures are not taken under consideration so it causes genuine impacts on plants because of which crop and plant quality, amount or efficiency is hindered. Discovery of plant infection through some programmed procedure is helpful as it lessens a substantial work of observing in huge homesteads of products. We have proposed a system for image processing procedure which is utilized for programmed location and characterization of plant leaf ailments automatically.

It additionally covers study on various illness characterization methods that can be used for plant leaf ailment identification. Together with measuring the infection, this model also

compares the soil quality with plants health via image processing, which is a vital viewpoint for infection location in plant leaf disease.

Overview of system

Certain important factors such as temperature, humidity, light and the level of carbon dioxide has an impact on the productivity of plant growth. Therefore, continuous monitoring of these environmental factors gives information to the user, how each factor affects growth and how to maximize the growth of plants. In recent years, precision agriculture has become the trend in agriculture.

Here the focus is mainly on understanding the environment through the interpretation of wide variety of data. The main idea of the system is to monitor the plants whether they get required amount of water and light. If there is enough moisture in the soil, the same will be reported to the user.

This will help the user to give the resources to the plants every day without much manual effort and constantly monitor the health of a plant from a remote location. Improvement of agricultural field has become biggest challenge for countries like India, so new technologies are to be adopted. We have implemented a novel methodology of physical parameter monitoring, data integration to the cloud, alert generation and predicting the future values. We have used Temperature humidity sensor, Soil moisture sensor and Light intensity sensor.
These sensors have been installed in the agriculture field to collect the data, and thus data is stored into the cloud using ESP 32 IoT cloud platform.

Problem statement

Causes of Crop Failure

Adverse Climatic Conditions

- Adverse climatic conditions will most probably top the list of the causes of crop failure.
- Adverse weather conditions include conditions that are too harsh for crops to survive, including extremely cold or extremely hot temperatures.
- These adverse weather conditions cause the crops to either dry up due to the scorching sun or fail to grow due to extremely cold conditions.

Unpredictable Weather Conditions

- In the recent past, the climatic and weather conditions have been quite unpredictable.
- The sequence of the cultivation seasons has been interrupted by the constant change in weather conditions.
- For instance, an extended period of drought, prolonged wet season, flash floods, and a complete change of season.
- The unpredictable weather conditions are a result of global warming and other human activities.

Pests and Diseases

- Several pests affect the growth of crops in the fields. Some of these pests tend to be expensive to curb.
- For instance, most farmers in the developing world will watch their crops being consumed by pests such as armyworms, stalk borer, Black cutworm, and Asiatic Garden beetles simply because they do not have money to purchase pesticides. Several diseases also lead to crop failure.
- Some of these diseases include leaf blight, Pythium, and southern rust. Pests and diseases, if not detected and dealt with early enough, may lead to massive crop failure.

Poor Farming Practices

- Poor farming methods and techniques will also lead to crop failure. The poor farming methods are mostly as a result of lack of knowledge of modern farming techniques and lack of funds to embrace the technology in farming.
- The farming methods that could lead to crop failure include mono-cropping and failure to apply fertilizers and pesticides.
- Application and practice of the new and the superior farming methods go a long way in curbing and reducing crop failure.

Human Activities

- Several human activities affect the prosperity of crops in the fields.
- The application of harsh chemicals could lead to the wilting of the crops. Other activities that could lead to crop failures include poor disposal of industrial waste products which may increase levels of greenhouse gases in the atmosphere.
- The effects of these gases, such as sulfur dioxide, lead to the fall of acid rain and blockage of leaf pores. These phenomena lead to crop wilting and the result is crop failure.

Objective

- Our project is concerned with the farmer and the cultivation land.
- It uses a camera and all the sensors which are used to monitor the agricultural land.
- It can be used from the plantation period to harvesting period.
- It detects plant type and calculates the area of the field itself.
- The robot monitors the field, checks the soil health based on moisture level and Ph level.
- The plant health is judged based on its leaf color.
- The robot prompts farmer, if any problem associated with the field is found in local language.

Scope of Project

- This Project will prompt further headway in field of farming.
- By decreasing human efforts in unfavourable climatic conditions, the plants wellbeing will lie in safe hands.
- This project will open entryways to another mechanical progression in observation and will at last prompt advancement in agriculture field.

Methodology

- The agricultural robot will be using a chassis as a base to connect and assemble everything on it, will be consisting of four motors. Two of which are high torque motors and the rest being gear motors. The robot is capable of doing four separate functions.
    1. Crop monitoring
    2. Soil fertility check
    3. Irrigation System
    4. Environment monitoring
- These functions will be working in different modes. Programming of different modes is done separately.
- The sharp sensor gives input to the robot by measuring the length and breadth of the field.

- Arduino is programmed in such a way that, after getting the data of length and breadth of field, mode will be selected in which the robot is made to work.
- Mode 1. Detect the plant type: This mode checks the plant type by its leaf size and pattern using machine learning technique. There on deciding the plant type, we are going to check other conditions based on plant type.
- Mode 2. Detect plant and soil health: When we know the plant type, we can check the condition of the plant in accordance of it's need. Some crop needs more nutrients in soil where as some crop needs less and color of some deficiency shown on the plant leaf may vary in color and visibility.
- Mode 3. Detect the Moister level of the soil: The moister level for all the crops is not same. Some may need more moister level where as some need less moister level. So, we have an effective irrigation system, where it will be working on accordance with plant's need.
- Mode 4. Checking the Environment condition: Every crop need its own likely environment where it will grow faster and yields more. So, when we check the environment, we come to know that the crop yield more or not. If not, we can take some precautions.
- To check the fertility of the soil we are going to use the soil moister sensor and Ph sensor.
- Based on the output of these two sensors and the color of the crop leaves, we can decide the fertility of the soil and the health of the plant.
- The Ph sensor can decide on soil nature and moister level where as the color of leaf can tell us about the fertility of the soil and its health.

CHAPTER NO 2

LITERATURE REVIEW

SL No	Author	Journals	Title
1.	Dhanasekar J, 2 Sengottuvel P, 3N Srikanth	International Journal of Pure and Applied Mathematics Volume 116 No. 20 2017, 457-461 ISSN: 1311-8080 (printed version); ISSN: 1314-3395	AUTONOMOUS FARMING ROBOT WITH PLANT HEALTH INDICATION
2.	Madan Veer, Akash Chavan, Tushar Chavan	International Journal of Computer Applications (0975 – 8887) National Conference on Advancements in Computer & Information Technology (NCACIT-2016)	Autonomous Farming Robot for Plant Health Indication
3.	Bhure Amit P 2 , Mangnale Shivkumar 3 Pandharkar Suraj R	International Journal of Advanced Technology in Engineering and Science www.ijates.com Volume No.03, Issue No. 01, January 2015	AUTONOMOUS FARMING ROBOT WITH PLANT HEALTH INDICATION
4.	Nitin Gawade2 , Sharvari shedge3 , Amol padale	INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING Vol. 4, Issue 4, April 2016	Autonomous Farming Robot for Plant Health Indication Using Image Processing

Paper 1 :

- Agribusiness is extremely work concentrated field and just field where the robot is not utilized by and by.
- Presently a-days numerous ventures are attempting to diminish this human work by making machine in terms of robot.
- Vision based line direction of robot is finished by a strategy named as posture (having mix of picture and Its edge) directs a robot in succession field.
- So along these lines the stage which we have executed causes the robot to utilize system named as stance for its column direction.
- Here we are outlining independent savvy cultivating robot which shows the plant well being by watching the shade of their leaves as far as picture preparing which is finished by raspberry pi and some confinement to plant structure.
- The robot additionally takes note of the encompassing natural states of the plant as far as temperature and stickiness with the goal that the robot will choose about wellbeing of plat and will show on the LCD.

Paper 2:

- Agriculture is very labour intensive field and only field where the robot is not used presently.
- Now-a- days many industries are trying to reduce this human labor by making machine in terms of robot. Vision based row guidance of robot is done by a technique named as pose (having combination of image and its angle) helps to guide a robot in a row field.
- So thus the platform which we have implemented helps the robot to use technique named as pose for its row guidance.
- Here we are designing autonomous intelligent farming robot which indicates the plant health by observing the colour of their leaves in terms of image processing which is done by raspberry pi and some limitation to plant height.
- The robot also notes the surrounding environmental conditions of the plant in terms of temperature and humidity so that the robot will decide about health of plat and will display on the LCD.

Paper 3:

- Agriculture is very labor intensive field and only field where the robots are not involved.
- Now-a- days many industries are trying to reduce this human labor by making robots and machines. A vision-based row guidance method is presented to guide a robot platform which is designed independently to drive through the row crops in a field according to the design concept of open architecture.
- Then, the offset and heading angle of the robot platform are detected in real time to guide the platform on the basis of recognition of a crop row using machine vision.
- And the control scheme of the platform is proposed to carry out row guidance. Here we are designing a autonomous intelligent farming robot which indicates the plant health by observing the color of their leaves and based on the height of the plant.
- The robot also notes the surrounding environmental conditions of the plant like temperature, moisture and humidity so that the robot will decide about health of plat and will display on the LCD.
- The robot has also watering mechanism it will water the plants according to their needs by observing soil moisture and humidity.
- It will also tell when the cutting process should take place by observing the leaf color.

Paper4:

- Agriculture is very labor intensive field and only field where the robots are not involved.
- Now-a- days many industries are trying to reduce this human labor by making robots and machines. A vision-based row guidance method is presented to guide a robot platform which is designed independently to drive through the row crops in a field according to the design concept of open architecture.
- Then, the offset and heading angle of the robot platform are detected in real time to guide the platform on the basis of recognition of a crop row using machine vision.
- And the control scheme of the platform is proposed to carry out row guidance. Here we are designing a autonomous intelligent farming robot which indicates the plant health by observing the color of their leaves and based on the height of the plant.

- The robot also notes the surrounding environmental conditions of the plant like temperature, moisture and humidity so that the robot will decide about health of plat and will display on the LCD.
- The robot has also watering mechanism it will water the plants according to their needs by observing soil moisture and humidity.
- It will also tell when the cutting process should take place by observing the leaf color.

Paper5:

S. Zhang et al [2] represent IoT based plant disease recognition system implementing k-mean and super pixel clustering and pyramid of histograms of orientation gradients (PHOG). The authors firstly applied super-pixel clustering algorithm to perform partition of colored unhealthy leaf images. Then, k-mean clustering algorithm was used to fragment the diseased image from each super-pixel and PHOG vectors were extracted from three color components and gray scaled image. This proposed model is applied in leaf of apple and cucumber plant with the accuracy of 90.43% and 92.15%.

Paper 6:

A. Camargo et al [3] proposed a Multi class classification model to classify the leaf disease detected using threshold segmentation. The features of shape, texture, gray level, histogram of frequencies, fractal dimension are extracted and processed by multi class support vector machine where the accuracy had been 91.3%.

Paper7:

M. Dhakate et al [4] represented a methodology implementing image processing and neural networks to identify and recognize diseases in pomegranate plant which are fruit Spot, Fruit rot, bacterial blight, and leaf spot. The authors further applied GLMC method for texture extraction and classified the diseases by artificial neural networking where the proposed model showed 90% accuracy.

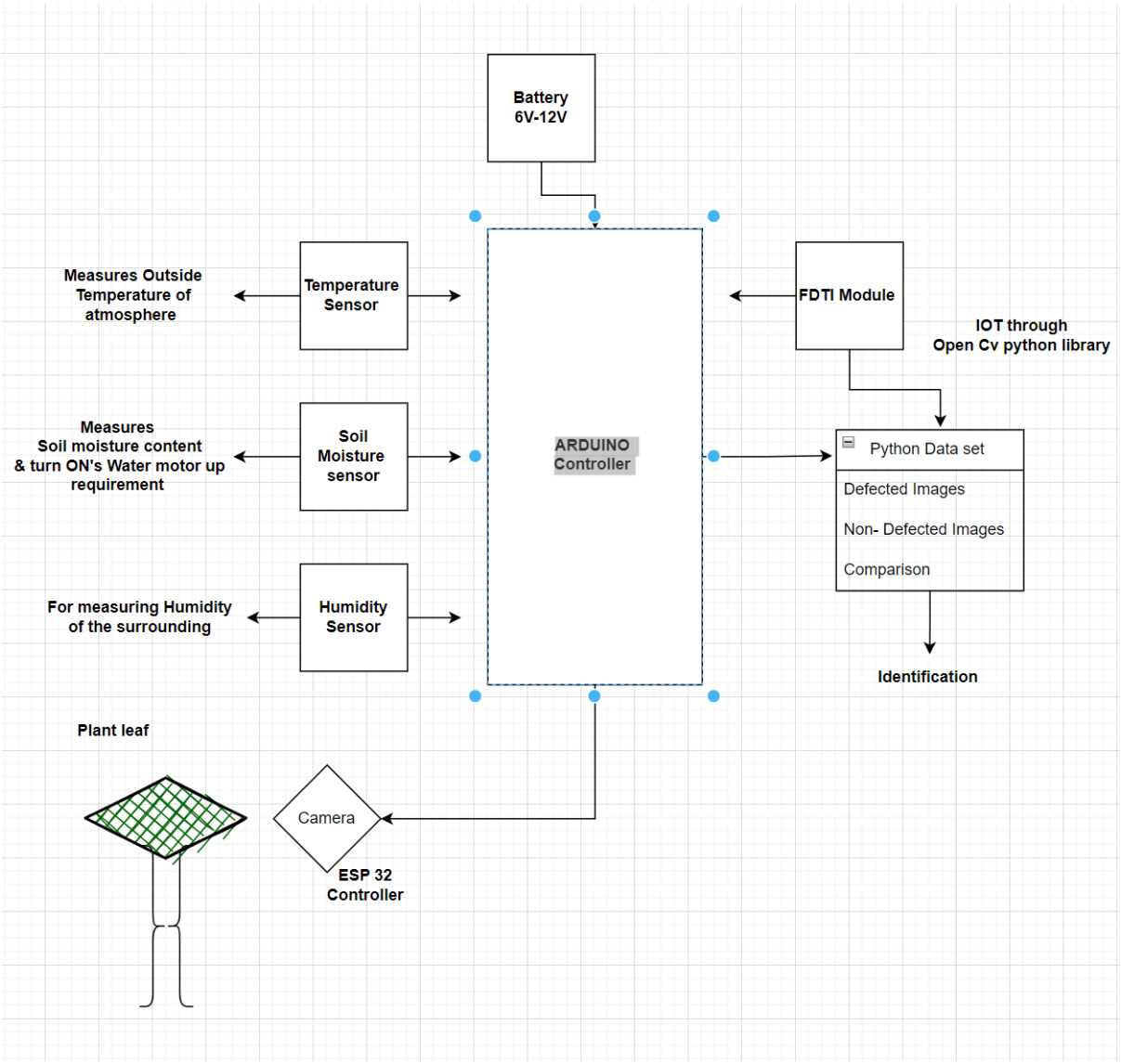
Literature Gap Identified

- We had researched many of the journals most of the projects are on simple monitoring system based on Image aqua-station using raspberry pie controller.
- The main disadvantage is controller cost Raspberry comes with price of 9000-11000 per piece.
- In our project we are trying to develop the setup with minimum cost so that each & every farmer can make use of this.
- ESP 32 Controller Comes with 750/- cost these in turn reduces thousands of cost to the framers.

CHAPTER NO 3

PROPOSED SYSTEM

Schematic Diagram



Construction of the project

- Esp 32 is Controller which can be used for monitoring of a specific object or body, it comes with an inbuilt Camera & wi-fi Module. Camera screening can be done via Serial monitoring by connecting to a server port.
- The project comes with a humidity, moisture & temperature sensor.

- These sensors measure the surrounding atmosphere and operates according to collaboratives. Like example,
- If the temperature of surrounding is high it turns on the fan to maintain the constant temperature to efficiently grow the plantation. Or
- It providers spraying of water according to humidity or moisture of the soil or atmosphere condition by turning on the water motor upon the requirement using Motor drivers provided.
- Motor drivers are used to control the actuation of Motor ONN & OFF for a particular temperature, humidity & moisture of the soil.
- Form Images processing and defect identification
- Data set will be prepared with defective and non-defective Images by creating identical folders for each defect, when the camera module captures the Image of leaf, data from the code will be compiled with real time image and identification will done on the basis of defect present on the leaf. Further this action will be solved by spraying the fertilizer for the defected leaf.

Note

- We can set the operations of spraying and switching ONN or OFF of the fan with an empirical value of surrounding.
- Example: - if the temperature of the atmosphere reaches more then 45 degrees then the Motor of blower or fan is turned ONN when the temperature comes below 30 degrees automatic cut-off of motor will be coded in the controller.
- Water spraying also works same as above, depending up on the moisture os the soil or surrounding etc.

Features & Future Scope

Features

1. Fully programmed framework so decreases the human work.
2. Saves time.
3. A ton of precision. Harvesting, weeding.

Future Scope

- We can build robots exactness of identification of leaf shading effectively by utilizing high quality camera.
- Wire-less System. We can make this framework remote by utilizing RF connectors.
- The framework can additionally altered for picking organic products, and real cutting procedure by the framework.

Components

- LDR sensor
- Soil Moisture
- Temperature Sensors
- DC Motor
- Rotating mini Blower or fan
- Camera Module Esp 32 with inbuilt Wi-fi.
- Fan or blower
- Arduino

All the data are screened on Android mobile or serial monitor Apps in the PC . Like health wise.

Software Details & data Set

3.1 Resources And Consumables Required

- The resources used for development of the automatic shaft
- With the rise of deep learning and computer vision, we can automate object detection.
- We can build deep learning and computer vision models, that can detect and locate objects, calculate the distance between them, predict their future stages, etc.
- Object detection has a wide range of applications in computer vision and machine learning.

Now that we have understood object detection, let us move to a slightly advanced technique called image segmentation. We can easily understand the difference between object detection and image segmentation by analyzing.

- Both methods try to identify and locate the objects in an image. In object detection, this is achieved using bounding boxes.
- The algorithm or model will locate the objects by drawing a rectangle-bounding box around them. In image segmentation, each pixel of the image is annotated.
- That means, given an image, the segmentation model tries to do the pixel-wise classification by classifying all the pixels of an image into meaningful classes of objects.
- This is also known as a dense prediction because it predicts the meaning of each pixel by identifying and understanding what object they belong to.

“The return format of image Segmentation is called a mask: an image that has the same size as the original image, but for each pixel, it has simply a boolean indicating whether the object is present or not present.”

- We will be using this technique in this project study. To know more about image segmentation.
- Now we got an idea about object detection and image segmentation. Let us move further and understand the problem statement.
-

ML Formulation

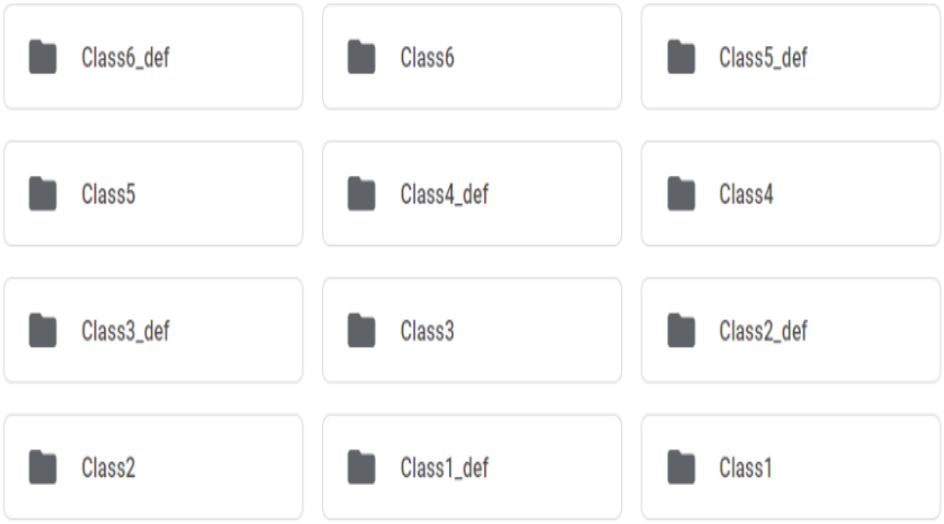
- The problem can be formulated as an image segmentation task. Given an image of a product, we need to plot the segmentation mask for that. If the product is defective, the segmentation map should be able to locate that defect.

Performance metric

- One of the most commonly used metrics in segmentation problems is the Intersection-Over-Union (IoU) score. Refer to the below image, which clearly shows how the IoU score is calculated.
- IoU is the area of overlap between the predicted segmentation and the true segmentation divided by the area of union between the predicted segmentation and the original segmentation.

Understanding the data

- The dataset contains two folders — train and test. The train set consists of six classes of images. Each class of images was separated into two folders, of which one folder contained 1000 non-defective images and the other contained 130 defective images.
- The below figure shows the folders inside the train folder.
- The folder name ending with "_def" contains the defective images of the corresponding class, and those without "_def" represent the non-defective images.
- The test folder contains a set of 120 defective images whose segmentation map is to be predicted.



Data set folder for defected & non defected leaf

Data preparation

- Preparing the image data and segmentation masks
- Now we need to prepare the image data and the corresponding segmentation mask for each image.
- We have the images divided into twelve folders. In this section images with defective parts are stored.
- The first image represents a defective product and the second one represents a non-defective image.
- Now we need to prepare the segmentation maps for these images.
- The segmentation map should detect the defective parts in the image. For the above images, the expected segmentation maps will be like this.
- The data for defected part is obtained using the get_data function which will be designed later in the part.
- one can use this information and draw an segmentation mask using the skimage function, which will also be done in later process.

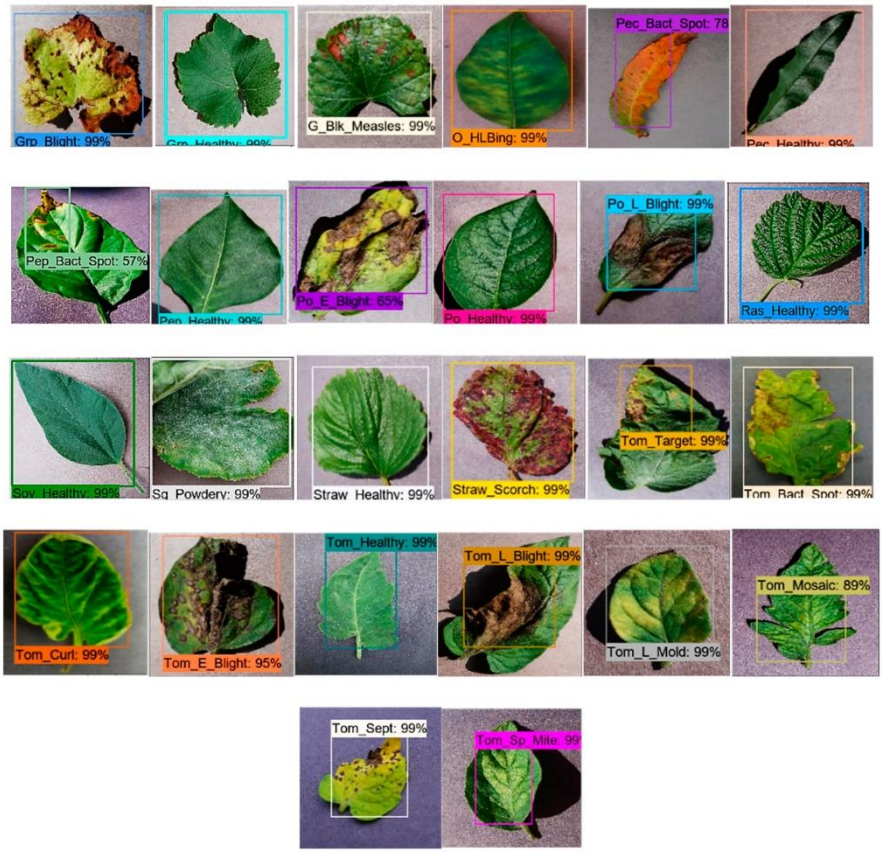
Loading the data

The structured data is obtained in the form as shown below.

	Unnamed: 0	image	mask
0	0	woboto/train/train/Class2_def/96.png	woboto/train/train/Class2_def/masks/96.png
1	1	woboto/train/train/Class2_def/95.png	woboto/train/train/Class2_def/masks/95.png
2	2	woboto/train/train/Class2_def/94.png	woboto/train/train/Class2_def/masks/94.png
3	3	woboto/train/train/Class2_def/93.png	woboto/train/train/Class2_def/masks/93.png
4	4	woboto/train/train/Class2_def/92.png	woboto/train/train/Class2_def/masks/92.png

The column “images” contains the full file path for each image and the “mask” column contains the corresponding mask images.

Defected images of leaf for data set



Data set images for open Cv python storage

Modeling

- Now that we got all the data, the next step is to find a model that can generate the segmentation masks for images. Let me introduce the UNet model, which is very popular for image segmentation tasks.
- UNet architecture contains two paths- the contracting path and the expanding path. The below figure will give a better understanding of the Unet architecture.

The model structure resembles the English letter "U" and hence the name Unet. The left side of the model contains the contraction path (also called the encoder) and it helps to capture the context in the image. The encoder is just a traditional stack of convolutional and max-pooling layers. Here we can see that the pooling layers reduce the height and width of the image and increase the depth or number of channels. At the end of the contraction path, the model will understand the shapes, patterns, edges, etc which are present in the image, but it loses the information of "where" it is present.

Since our problem is to obtain the segmentation maps for images, the information we obtain from the contracting path alone will not be sufficient. We need a high-resolution image as the output in which all pixels are classified.

"If we use a regular convolutional network with pooling layers and dense layers, we will lose the "WHERE" information and only retain the "WHAT" information which is not what we want. In the case of segmentation, we need both "WHAT" as well as "WHERE" information."

So, we need to upsample the image to retain the "where" information. This Is done in the expanding path which is on the right side. The expanding path (also called the decoder) is used for the localization of the captured context using upsampling techniques. There are various upsampling techniques like bilinear interpolation, nearest neighbor approach, transposed convolution, etc.

Training

After this training will be done, and the model is also decided. Now let us train the model.

Training procedure with an example

Since the number of non-defective images is much higher than the defective images, we are taking only a sample from the non-defective images for better

results. The model was compiled using adam optimizer and we used the dice loss as the loss function. The performance metric used was iou score.

Esp-32 connection for Open Cv

OpenCV is an open-sourced image processing library that is very widely used not just in industry but also in the field of research and development.

Here for object detection, we have used the cvlib Library. The library uses a pre-trained AI model on the COCO dataset to detect objects. The name of the pre-trained model is YOLOv3.

Introduction to open cv

OpenCV (Open Source Computer Vision Library: http://opencv.org) is an open-source library that includes several hundreds of computer vision algorithms. The document describes the so-called OpenCV 2.x API, which is essentially a C++ API, as opposed to the C-based OpenCV 1.x API (C API is deprecated and not tested with "C" compiler since OpenCV 2.4 releases)

- But first, make sure to get familiar with the common API concepts used thoroughly in the library.

API Concepts

All the OpenCV classes and functions are placed into the cv namespace. Therefore, to access this functionality from your code, use the cv:: specifier or using namespace cv; directive:

```
#include "opencv2/core.hpp"
...
cv::Mat H = cv::findHomography(points1, points2, cv::RANSAC, 5);
...
```

or

```
#include "opencv2/core.hpp"
using namespace cv;
...
Mat H = findHomography(points1, points2, RANSAC, 5 );
...
```

Automatic Memory Management

OpenCV handles all the memory automatically.

First of all, std::vector, cv::Mat, and other data structures used by the functions and methods have destructors that deallocate the underlying memory buffers when needed. This means that the destructors do not always deallocate the buffers as in case of Mat. They take into account possible data sharing. A destructor decrements the reference counter associated with the matrix data buffer. The buffer is deallocated if and only if the reference counter reaches zero, that is, when no other structures refer to the same buffer. Similarly, when a Mat instance is copied, no actual data is really copied. Instead, the reference counter is incremented to memorize that there is another owner of the same data.

Automatic Allocation of the Output Data

OpenCV deallocates the memory automatically, as well as automatically allocates the memory for output function parameters most of the time. So, if a function has one or more input arrays (cv::Mat instances) and some output arrays, the output arrays are automatically allocated or reallocated. The size and type of the output arrays are determined from the size and type of input arrays. If needed, the functions take extra parameters that help to figure out the output array properties.

The array frame is automatically allocated by the >> operator since the video frame resolution and the bit-depth is known to the video capturing module. The array edges is automatically allocated by the cvtColor function. It has the same size and the bit-depth as the input array. The number of channels is 1 because the color conversion code cv::COLOR_BGR2GRAY is passed, which means a color to grayscale conversion. Note that frame and edges are allocated only once during the first execution of the loop body since all the next video frames have the same resolution. If you somehow change the video resolution, the arrays are automatically reallocated.

The key component of this technology is the cv::Mat::create method. It takes the desired array size and type. If the array already has the specified size and type, the method does nothing. Otherwise, it releases the previously allocated data, if any (this part involves decrementing the reference counter and comparing it with zero), and then allocates a new buffer of the required size. Most functions call the cv::Mat::create method for each output array, and so the automatic output data allocation is implemented.

Some notable exceptions from this scheme are cv::mixChannels, cv::RNG::fill, and a

few other functions and methods. They are not able to allocate the output array, so you have to do this in advance.

Saturation Arithmetic's

As a computer vision library, OpenCV deals a lot with image pixels that are often encoded in a compact, 8- or 16-bit per channel, form and thus have a limited value range. Furthermore, certain operations on images, like color space conversions, brightness/contrast adjustments, sharpening, complex interpolation (bi-cubic, Lanczos) can produce values out of the available range. If you just store the lowest 8 (16) bits of the result, this results in visual artifacts and may affect a further image analysis. To solve this problem, the so-called saturation arithmetics is used. For example, to store r, the result of an operation, to an 8-bit image, you find the nearest value within the 0..255 range:

I(x,y)=min(max(round(r),0),255)

Similar rules are applied to 8-bit signed, 16-bit signed and unsigned types. This semantics is used everywhere in the library. In C++ code, it is done using the cv::saturate_cast<> functions that resemble standard C++ cast operations.

Fixed Pixel Types. Limited Use of Templates

Templates is a great feature of C++ that enables implementation of very powerful, efficient and yet safe data structures and algorithms. However, the extensive use of templates may dramatically increase compilation time and code size. Besides, it is difficult to separate an interface and implementation when templates are used exclusively.

This could be fine for basic algorithms but not good for computer vision libraries where a single algorithm may span thousands of lines of code. Because of this and also to simplify development of bindings for other languages, like Python, Java, Matlab that do not have templates at all or have limited template capabilities, the current OpenCV implementation is based on polymorphism and runtime dispatching over templates. In those places where runtime dispatching would be too slow (like pixel access operators), impossible (generic cv::Ptr<> implementation), or just very inconvenient (cv::saturate_cast<>()) the current implementation introduces small template classes, methods, and functions. Anywhere else in the current OpenCV version the use of templates is limited.

Consequently, there is a limited fixed set of primitive data types the library can operate on. That is, array elements should have one of the following types:

8-bit unsigned integer (uchar)

8-bit signed integer (schar)

16-bit unsigned integer (ushort)

16-bit signed integer (short)

32-bit signed integer (int)

32-bit floating-point number (float)

64-bit floating-point number (double)

a tuple of several elements where all elements have the same type (one of the above). An array whose elements are such tuples, are called multi-channel arrays, as opposite to the single-channel arrays, whose elements are scalar values.

Input Array and Output Array

Many OpenCV functions process dense 2-dimensional or multi-dimensional numerical arrays. Usually, such functions take cv::Mat as parameters, but in some cases it's more convenient to use std::vector<> (for a point set, for example) or cv::Matx<> (for 3x3 homography matrix and such). To avoid many duplicates in the API, special "proxy" classes have been introduced. The base "proxy" class is cv::InputArray. It is used for passing read-only arrays on a function input. The derived from InputArray class cv::OutputArray is used to specify an output array for a function. Normally, you should not care of those intermediate types (and you should not declare variables of those types explicitly) - it will all just work automatically. You can assume that instead of InputArray/OutputArray you can always use cv::Mat, std::vector<>, cv::Matx<>, cv::Vec<> or cv::Scalar. When a function has an optional input or output array, and you do not have or do not want one, pass cv::noArray().

Error Handling

OpenCV uses exceptions to signal critical errors. When the input data has a correct format and belongs to the specified value range, but the algorithm cannot succeed for some reason (for example, the optimization algorithm did not converge), it returns a special error code (typically, just a boolean variable).

The exceptions can be instances of the cv::Exception class or its derivatives. In its turn, cv::Exception is a derivative of std::exception. So it can be gracefully

handled in the code using other standard C++ library components.

Finally,

Multi-threading and Re-enter ability

That is, the same function or the same methods of different class instances can be called from different threads. Also, the same Mat can be used in different threads because the reference-counting operations use the architecture-specific atomic instructions.

Connection of ESP 32 for Open Cv

- Pins description and the method to program ESP32 Camera Module using FTDI Module.
- We will also set up the Arduino IDE for the ESP32 Camera Module.
- We will also upload the firmware and then work on the object detection & identification part.
- The script for object detection will be written in the python programming language, thus we will also have to install Python and its required Libraries.

In an earlier ESP32 CAM Based project we learned about Face Detection System & also Color Detection System using Python & OpenCV. This project also requires the use of OpenCV for Object Detection & Identification.

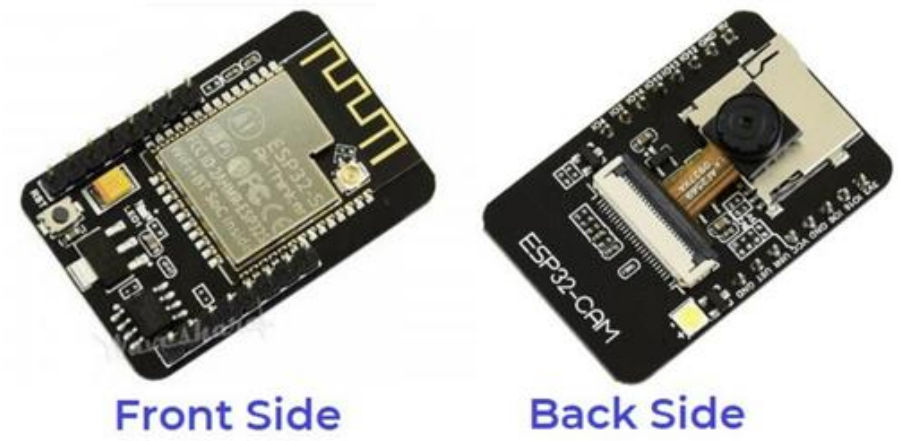
Components

The following is the list of Bill of Materials for building an ESP32 CAM Based Object Detection & Identification System. The ESP32 CAM when combined with another hardware & firmware track and identify the object.

- ESP 32 CAM Board
- FTDI Module
- USB Cable
- Jumper Wires

ESP32 CAM Module

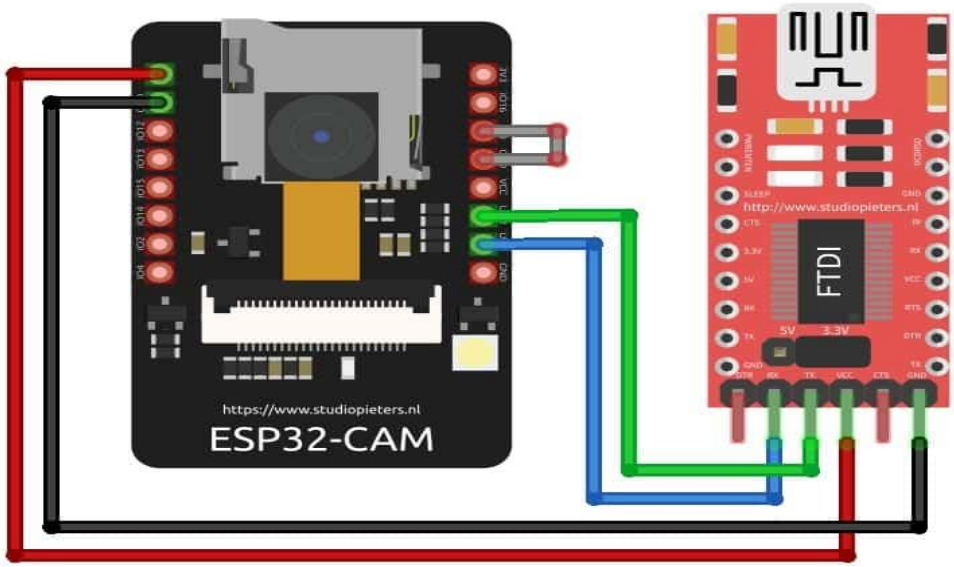
The ESP32 Based Camera Module developed by AI-Thinker. The controller is based on a 32-bit CPU & has a combined Wi-Fi + Bluetooth/BLE Chip. It has a built-in 520 KB SRAM with an external 4M PSRAM. Its GPIO Pins have support like UART, SPI, I2C, PWM, ADC, and DAC.



- The module combines with the OV2640 Camera Module which has the highest Camera Resolution up to 1600 × 1200.
- The camera connects to the ESP32 CAM Board using a 24 pins gold plated connector. The board supports an SD Card of up to 4GB. The SD Card stores capture images.

ESP32-CAM FTDI Connection

- For programming the board, you need any USB-to-TTL Converter Module or an FTDI Module. There are so many FTDI Module available based on CP2102 or CP2104 Chip or any other chip.
- For getting started with ESP32 CAM Module, make a following connection between FTDI Module and ESP32 CAM module.



Pin connection ESP32 with FTDI

`

ESP32-CAM	FTDI PROGRAMMER
GND	GND
5V	VCC (5V)
U0R	TX
U0T	RX
GPIO 0	GND

Table connection of ESP 32 Cam module

Connect the 5V & GND Pin of ESP32 to 5V & GND of FTDI Module. Similarly, connect the Rx to UOT and Tx to UOR Pin. And the most important thing, you need to short the IO0 and GND Pin together. This is to put the device in programming mode. Once programming is done you can remove it.

Major Component Description & it's Uses

1. Arduino Board

In this Project Arduino is used as Main Controller, in simple Arduino is an open-source electronics platform based on easy-to-use hardware and software to Control And manipulate the date Received with internal components like sensors and motors.

Arduino Board are able to read inputs Like- light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of instructions to the microcontroller on the board.

Uses In our Project

- In This project Arduino is used as a external controller where it propagates all the activity inheriting of Battery, like Receiving and transmitting the data of BMS from Internal Gas Sensors, Temperature sensor & Current amplification sensor.
- These data are read as Input and are sent to the cloud as outputs for evaluation of each cell via Bolt IOT Wi-fi Module to the cloud to manage the Battery Condition.
- Following slides gives a brief description of Arduino and its Pin configuration.

Board	Name	Arduino UNO R3
	SKU	A000066
Microcontroller	ATmega328P	
USB connector	USB-B	
Pins	Built-in LED Pin	13
	Digital I/O Pins	14
	Analog input pins	6
	PWM pins	6
Communication	UART	Yes
	I2C	Yes
	SPI	Yes
Power	I/O Voltage	5V

	Input voltage (nominal)	7-12V
	DC Current per I/O Pin	20 mA
	Power Supply Connector	Barrel Plug
Memory	ATmega328P	2KB SRAM, 32KB FLASH, 1KB EEPROM

Table 1 Arduino Specification

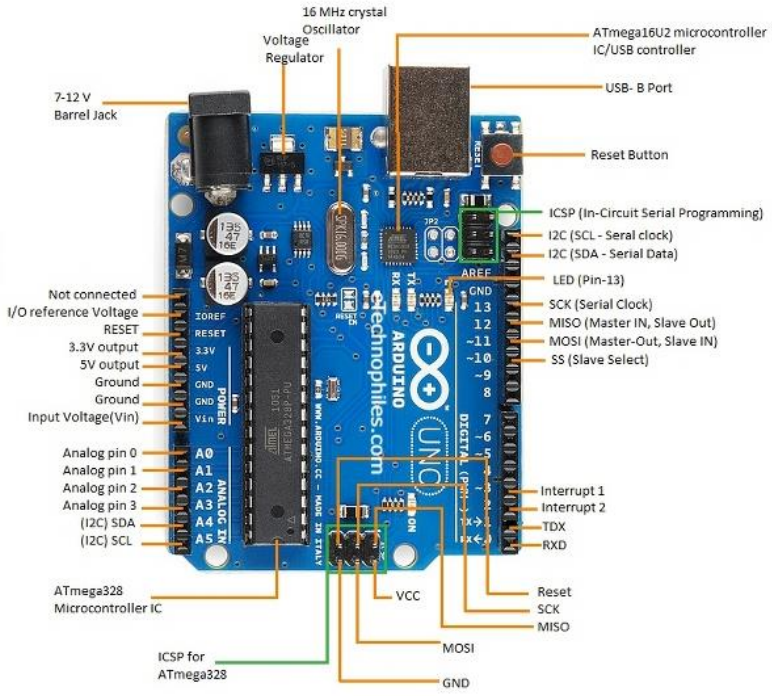


Figure 3 Arduino Pin Description

2. The Arduino IoT Cloud is a online platform that makes it easy for you to create, deploy and monitor IoT projects.

ESP32 / ESP8266 is used for IOT to create , Deploy & monitoring of Battery

- The Arduino IoT Cloud supports a wide range of third party boards based on the ESP32 and ESP8266 microcontrollers with support for Wi-Fi. To set them up, simply choose the third party option in the device setup.
- Connecting ESP32 & ESP8266 to Arduino Cloud IoT.

The goals of this project are:

1. Send data to the cloud from the board.
2. Turn ON/OFF a motor fan on the board through the cloud when Maximum temperature is induced in the battery.

We will also need the following components for the circuit:

- LED
- 220-ohm resistor
- Breadboard
- Jumper Wires

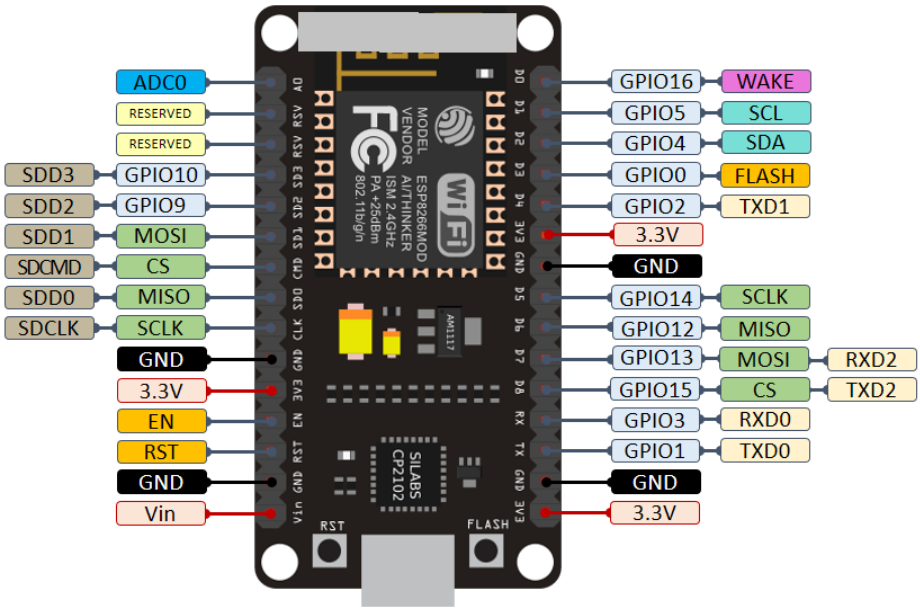


Figure 4 ESP 8266 Pin Description

2.1 Circuit connection of Board to ESP8266 for Cloud Interaction IDE design.

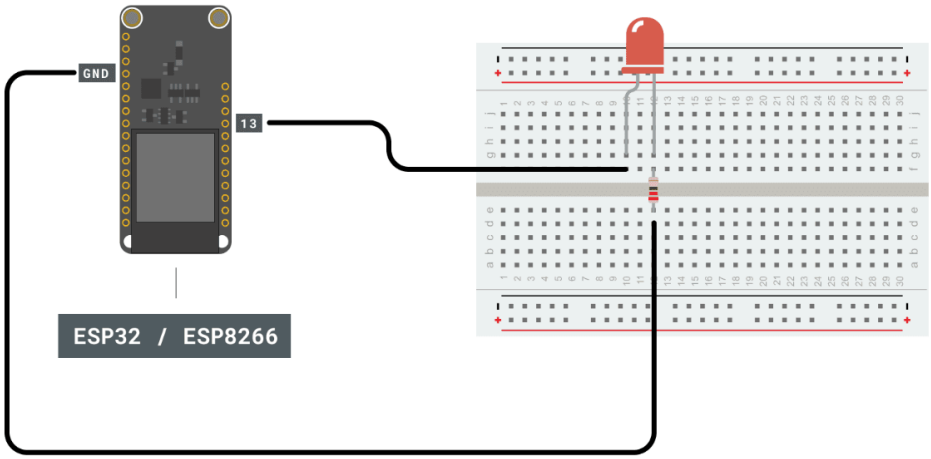


Figure 5 ESP 8266 Pin Connection

Step 1: Setting up the device

1. Once in the cloud, click on the "Devices" tab. Then, click on the "Add device" button.
2. Then, click on "Set up a 3rd party device".
3. Now we need to select what board we are using. Select either ESP32 or ESP8266, and then choose the board from the drop-down menu.
4. Now the board needs a name. We choose My_ESP32, but you can have a lot more imagination than that!
5. You will now receive your Device ID and Secret key. Please note that the secret key cannot be recovered, so make sure you note it down. You can also download a PDF with the information.

When you have saved it, tick the box at the bottom, and click on "Continue".

You have now configured your ESP32 board with the Arduino Cloud IoT. Your device will now appear in the list of devices.

Step 2: Creating a Thing

- The next step is to create a Thing. This is done by navigating to the "Things" tab.
- Then, we need to create a new Thing, by clicking on the "Create Thing" button. We can then rename our Thing something appropriate to what we are doing, in this case we simply chose ESP32 Project.

- Now, we need to link our device with our Thing. This is done by clicking on the button in the "Device" section. This will open a window, where your ESP32 / ESP8266 should be available to select.
- Once the device is linked, we need to create two variables: random_value and led_switch. Click on "Add variable" button. This will open a window where you need to fill in variable information. Let's create the random_value first. The data type is int, permission is read only and update policy is on change. Once done, click on the "Add variable" button.
- Now, let's also add the led switch variable. The data type for this variable is Boolean, the permission is read & write, and update policy is on change. Once done, click on the "Add variable" button.

Step 3: Adding credentials

- Now, we need to enter the credentials for our network, plus the secret key generated during the device configuration.
- First, click on the button in the "Network Section".
- Then, enter the credentials (network name, network password and secret key). Click "Save" when finished.
- The next step is to program the board. To do so, we need to go to the "Sketch" tab.

Programming the board includes Data set as follows

1. Voltage or current value, if there is deflection in the value then the board receives the signal via sensor and necessary action will be taken via cloud code program like power cut-off from the battery.
2. Temperature value, if the temperature of the battery exceed or deceased then the cooling fan will be turned ON/OFF upon situation.
3. Gas value, if the battery gets over heated then the battery releases hazardous gases which permanently damages the battery if it is turned off at proper time. Then there will be buzzer indication via cloud.

3. NTC Thermistor Temperature Sensor Module & IDE design

- It is very sensitive to ambient temperature. It is generally used to detect the temperature of the surrounding environment.
- Through potentiometer adjustment, it is possible to change the temperature detection threshold. DO output can be directly connected to the micro controller to detect high and low, by detecting temperature changes in the environment.

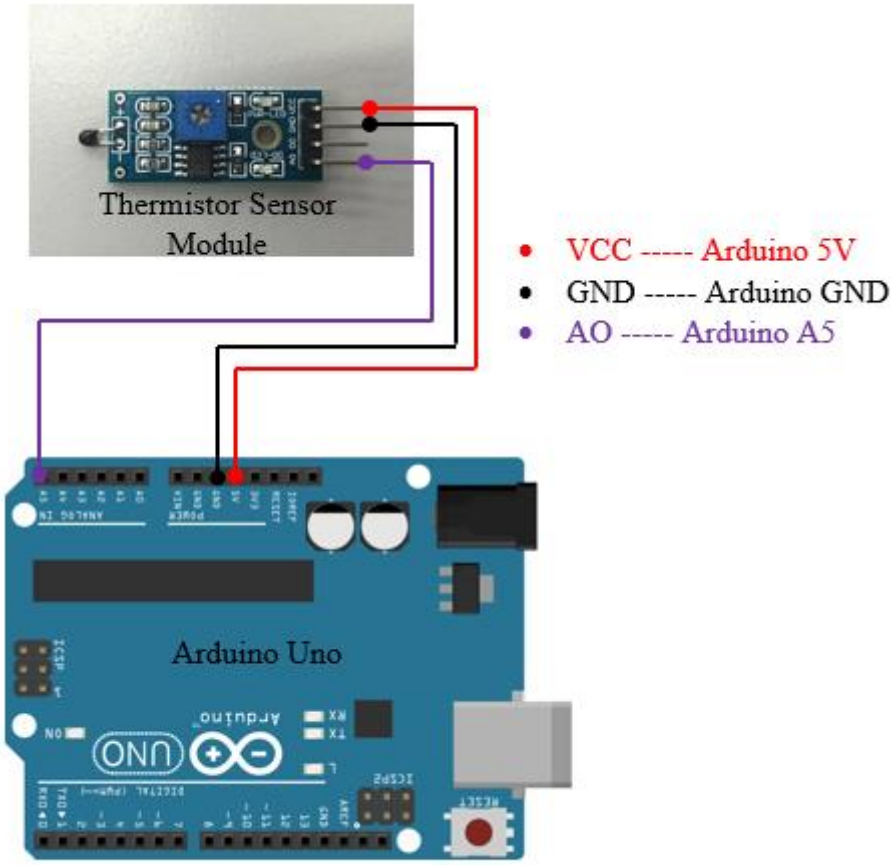


Figure 6 Temperature sensor Pin Diagram

- Working voltage: 3.3V-5V
- Output form: DO digital switching outputs (0 and 1) and AO analog voltage output
- Fixed bolt hole for easy installation
- Small PCB board size: 3.2cm x 1.4cm
- uses a wide voltage comparator LM393

Soil Moisture Sensor Module

This is an easy to use digital soil moisture sensor. Just insert the sensor in the soil and it can can measure moisture or water level content in it. It gives a digital output of 5V when moisture level is high and 0V when the moisture level is low in the soil.

Specifications:-

Operating voltage: 3.3V~5V

Dual output mode,analog output more accurate

A fixed bolt hole for easy installation

With power indicator (red) and digital switching output indicator (green)

Having LM393 comparator chip, stable

Panel PCB Dimension: Approx.3cm x 1.5cm

Soil Probe Dimension: Approx. 6cm x 3cm

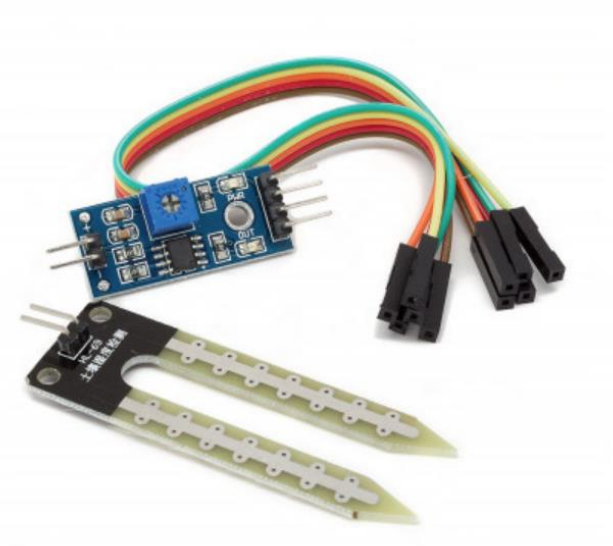
Cable Length: Approx.21cm

VCC: 3.3V-5V

GND: GND

DO: digital output interface(0 and 1)

AO: analog output interface



Pin connection of sensor

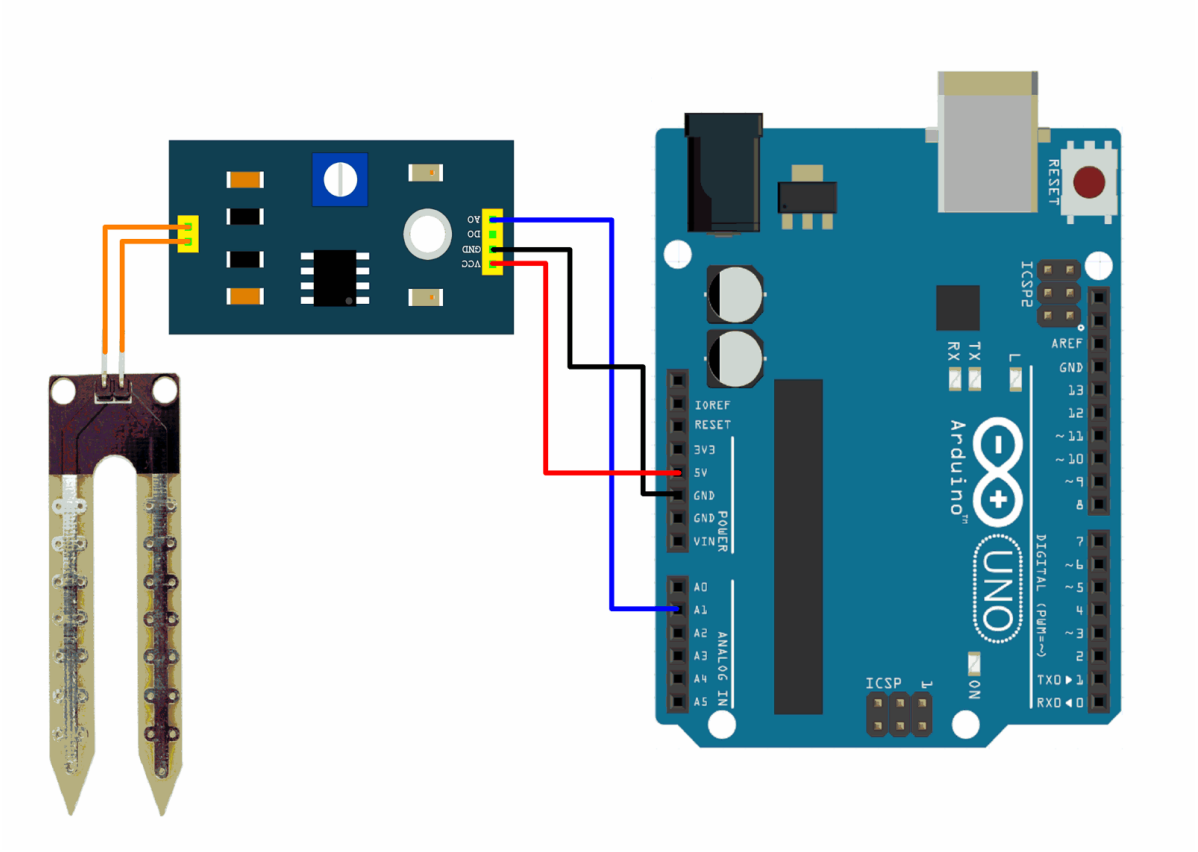


Figure pin connection

Humidity Sensor (DHT22)

The DHT-22 (also named as AM2302) is a digital-output, relative humidity, and temperature sensor. It uses a capacitive humidity sensor and a thermistor to measure the surrounding air, and sends a digital signal on the data pin.

DHT22 pins	
1	VCC
2	DATA
3	NC
4	GND

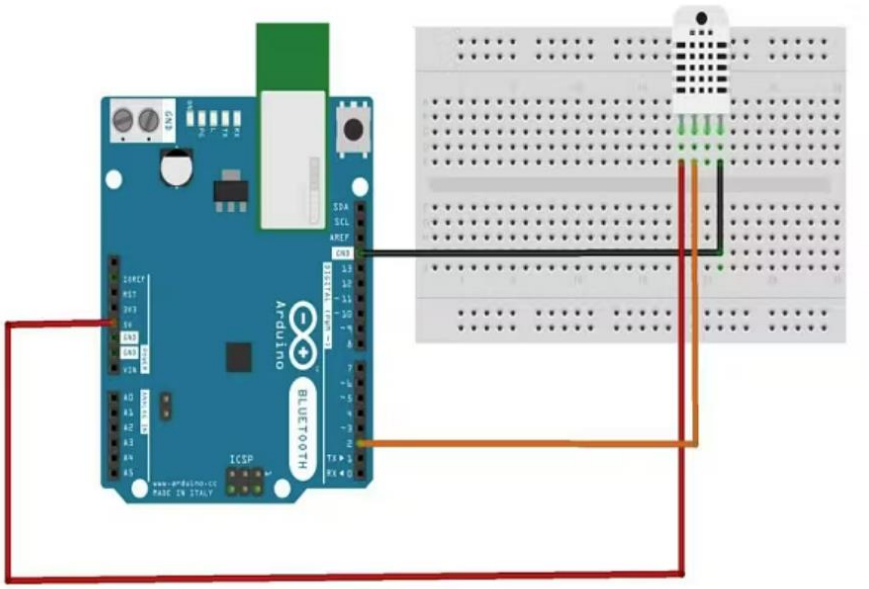


Figure pin connection

Technical Details

Power − 3-5V

Max Current − 2.5mA

Humidity − 0-100%, 2-5% accuracy

Temperature − 40 to 80°C, ±0.5°C accuracy

Installing ESP32CAM Library

Here we will not use the general ESP webserver example rather another streaming process. Therefore, we need to add another ESPCAM library. The esp32cam library provides an object-oriented API to use OV2640 camera on ESP32 microcontroller. It is a wrapper of esp32-camera library.

- Once downloaded add this zip library to Arduino Libray Folder. To do so follow the following steps:
- Open Arduino -> Sketch -> Include Library -> Add .ZIP Library… -> Navigate to downloaded zip file -> add

Source Code/Program for ESP32 CAM Module

- Here is a source code for Object Detection & Identification with ESP32 Camera & OpenCV.
- Write the code according to compiler and execute the program

- Before Uploading the code you have to make a small change to the code. Change the SSID and password variable and in accordance with your WiFi network.

- Now compile and upload it to the ESP32 CAM Board. But during uploading, you have to follow few steps every time.

- Make sure the IO0 pin is shorted with the ground when you have pressed the upload button.
- If you see the dots and dashes while uploading press the reset button immediately
- Once the code is uploaded, remove the I01 pin shorting with Ground and press

the reset button once again.

- If the output is the Serial monitor is still not there then press the reset button again.

Python Library Installation

- For the live stream of video to be visible on our computer we need to write a Python script that will enable us to retrieve the frames of the video.

- Open Python
- Then Go to the command prompt and install NumPy, OpenCV and cvlib libraries.

    - type: pip install numpy and press enter. After the installation is done.
    - type: pip install opencv-python and press enter.
    - type: pip install cvlib and press enter, close the command prompt.

In our python code we have used urllib.request to retrieve the frames from the URL and the library for image processing is OpenCV. For Object detection, we have used the Cvlib library that uses an AI model for detecting objects. Since the whole process requires a good amount of processing power, thus we have used multiprocessing which utilizes multiple cores of our CPU. Refere openCv python liberies details

Possible Outcomes

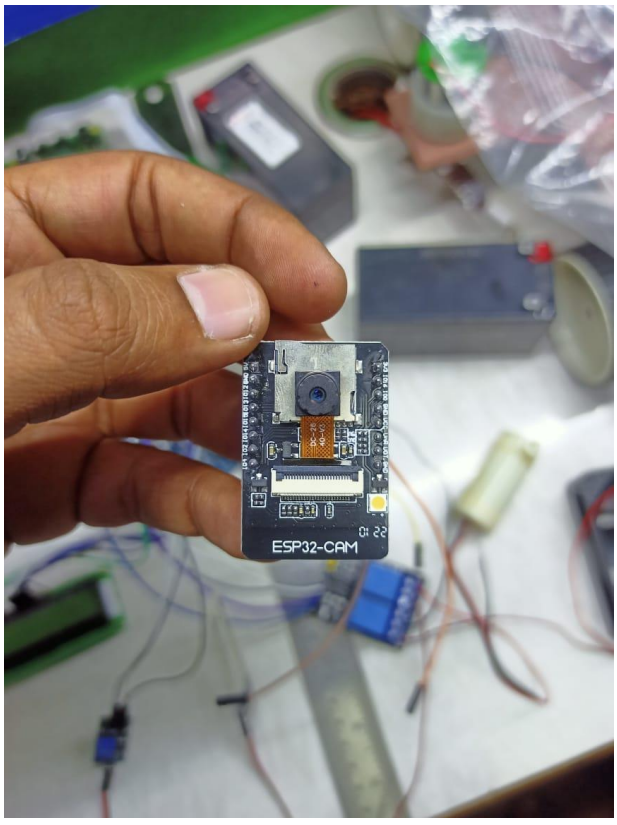
- Technology is making giant strides, meaning that farmers who get ahead of the curve and invest now, can start to become more efficient, produce healthier crops and increase their margins.
- Agricultural robots automate slow, repetitive, and dull tasks for farmers, allowing them to focus more on improving overall production yields, which will be vital as the world's population increases.
- Farm automation practices can make agriculture more profitable while also reducing the ecological footprint of farming at the same time.
- Site-specific application software can reduce the amount of pesticides and fertilizer used while also reducing greenhouse gas emissions.

Advantages of proposed system

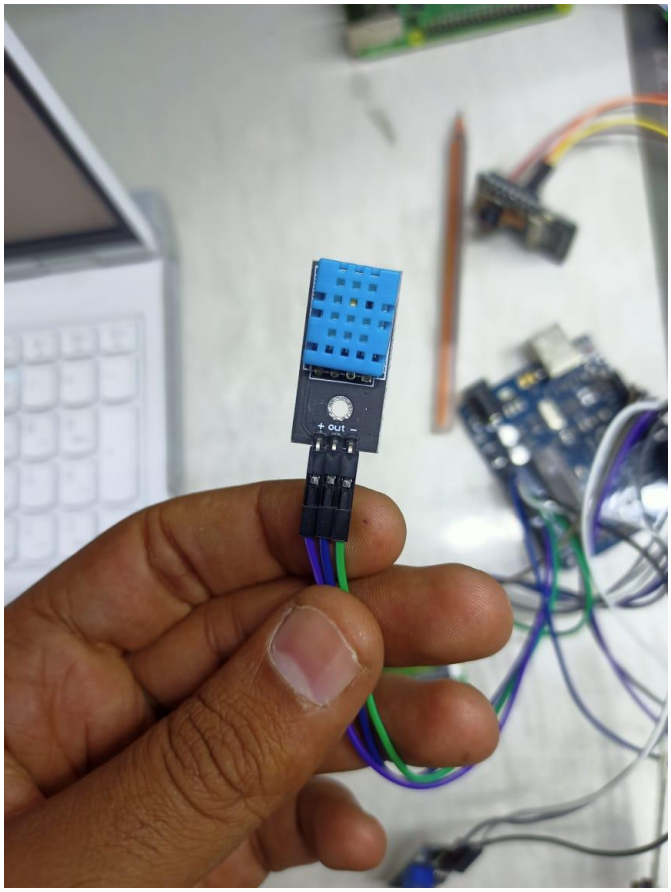
1. Reduces labor cost
2. Reduces human intervention.
3. Minimum time consumption
4. Higher plant growth rates.
5. Less failures crops.
6. Saves much water considering atmosphere sepal value .

Hardware development

Cooling fan



ESP 32 CAM + IOT WI-fi Module



Temperature + Humidity sensor

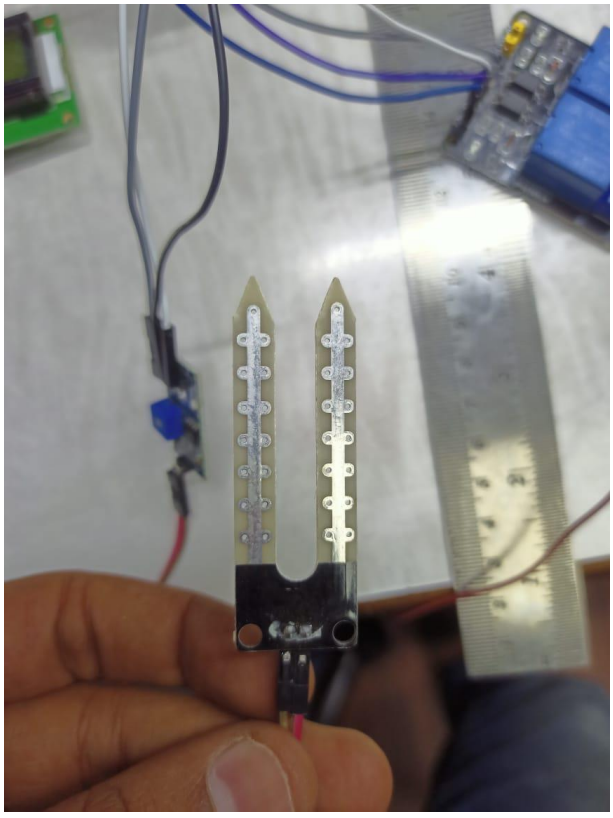
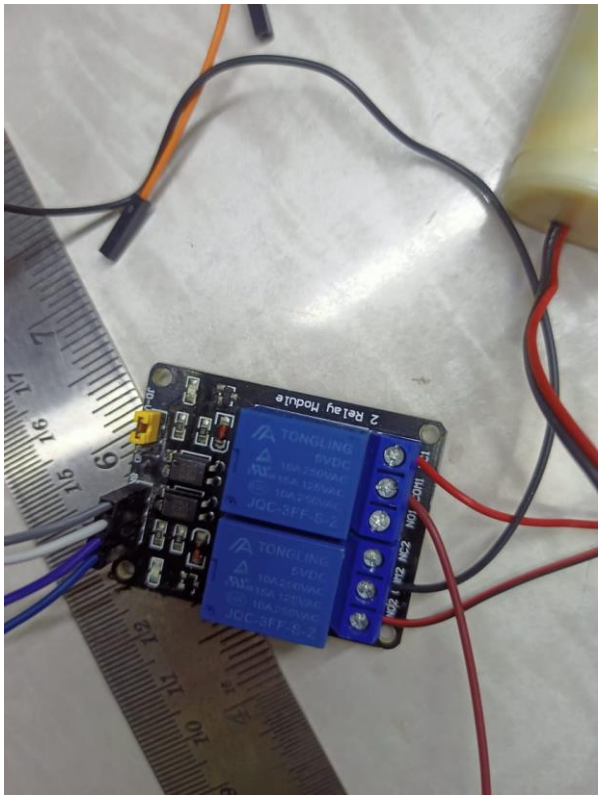
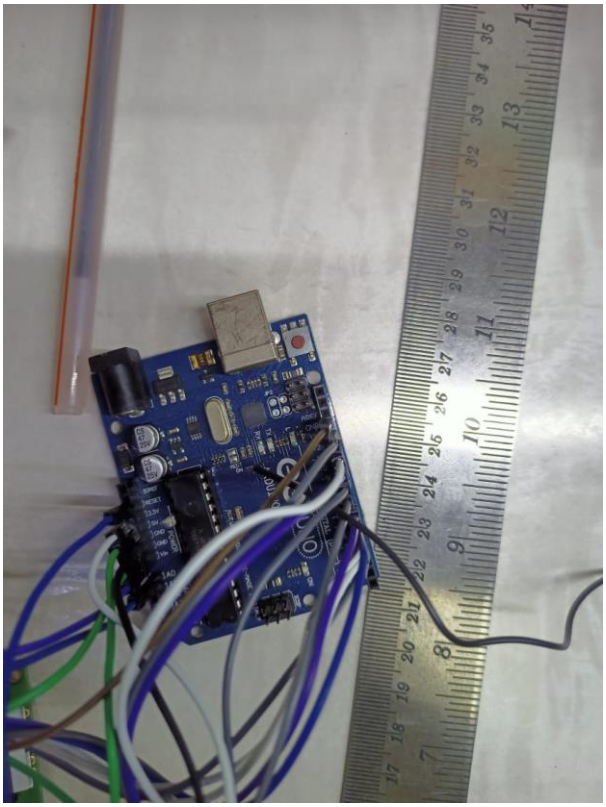


Figure Soil Moisture sensor

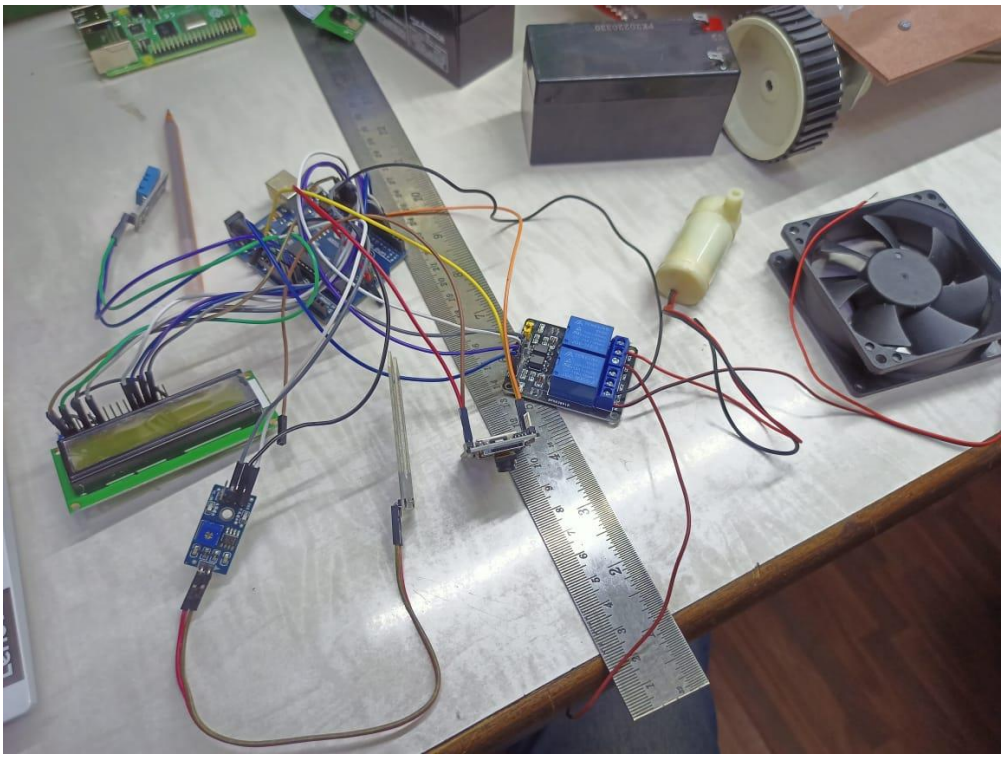
Submersible water motor



Relay



Arduino Uno Micro-Controller



Wire connection of all electronic sensor

Code for the installation

```
#include "DHT.h"
#include <LiquidCrystal.h>
//int IRSensor = 9; // connect ir sensor module to Arduino pin 9
#define Relay 6
#define sensorPin 9
int LED = 13; // conect LED to Arduino pin 13

LiquidCrystal lcd(3, 4, 5, 6, 7, 8);

#define DHTPIN A0     // what pin we're connected to

// Uncomment whatever type you're using!
#define DHTTYPE DHT11   // DHT 11
//#define DHTTYPE DHT22   // DHT 22  (AM2302)
//#define DHTTYPE DHT21   // DHT 21 (AM2301)

// Initialize DHT sensor for normal 16mhz Arduino
DHT dht(DHTPIN, DHTTYPE);

void setup() {
  Serial.begin(9600);
  //  pinMode(IRSensor, INPUT); // IR Sensor pin INPUT
  pinMode(sensorPin, INPUT);
  pinMode(LED, OUTPUT); // LED Pin Output
  Serial.println("DHTxx test!");
  lcd.begin(16, 2);
  dht.begin();
  lcd.print("Plant Health");
  delay(2000);
  lcd.clear();
}

void loop() {
```

```
Soil();
// Wait a few seconds between measurements.
delay(2000);
// Reading temperature or humidity takes about 250 milliseconds!
// Sensor readings may also be up to 2 seconds 'old' (its a very slow sensor)
float h = dht.readHumidity();
// Read temperature as Celsius
float t = dht.readTemperature();
// Read temperature as Fahrenheit
float f = dht.readTemperature(true);

// Check if any reads failed and exit early (to try again).
if (isnan(h) || isnan(t) || isnan(f)) {
  Serial.println("Failed to read from DHT sensor!");
  return;
}
// Compute heat index
// Must send in temp in Fahrenheit!
float hi = dht.computeHeatIndex(f, h);
lcd.print("Temperature:");
lcd.print(t);// display the temperature
lcd.print((char)223);
lcd.setCursor(0, 1);
lcd.print("humidity:");
lcd.print(h); // display the humidity
delay(1000);
lcd.clear();
Serial.print("Humidity:");
Serial.print(h);
Serial.print(" %\t");
Serial.print("Temperature:");
Serial.print(t);
Serial.print(" *C ");
Serial.print(f);
```

```
  Serial.print(" *F\t");
  Serial.print("Heat index: ");
  Serial.print(hi);
  Serial.println(" *F");
}
void Soil()
{
  int sensorStatus = digitalRead(sensorPin); // Set the GPIO as Input
  Serial.print(sensorStatus);
  if (sensorStatus == 1) // Check if the pin high or not
  {
    digitalWrite(Relay, LOW); // LED LOW
    Serial.println("Motion Ended!"); // print Motion Detected! on the serial monitor window
  }
  else
  {
    //else turn on the onboard LED
    digitalWrite(Relay, HIGH); // LED High
    Serial.println("Motion Detected!"); // print Motion Ended! on the serial monitor window
  }

}
```

Proposed Plan of action

Sr. No.	Activity	Year 2022-23											
		July	Aug	Sept	Oct	Nov	Dec	Jan	Feb	Mar	Apr	May	June
1	Literature Survey												
2	Problem Definition												
3	Material Selection												
4	Analytical calculation												
5	3D Designing												
6	Market survey												
7	Purchase of component												
8	Fabrication development												
9	Code development												
10	Testing												
11	Thesis Writing												

Expenditure

LDR sensor = 450/-

Soil Moisture = 650/-

Temperature Sensors = 950/-

Water motor = 750/-

Rotating mini Blower or fan = 450/-

Camera Module Esp 32 with inbuilt Wi-fi. = 1250/-

FDIT module = 650/-

Battery = 950/-

Jumper wires = 100/-

Other extra needed = 500/-

Arduino = 950/-

Total = 7650/-

References

[1] AndrewEnglish, Patrick Ross, David Ball, “Vision Based Guidance for Robot Navigation in Agriculture, Member, IEEE, Peter Corke, Fellow,IEEE-2014

[2] Chung L. Chang “Zigbee-assisted Mobile Robot Gardener”, Senior Member, IEEE,andJia H. Jhu IEEE - 2013

[3] XinLiu, Qian Zhang*, RuPeng Luan, Feng Yu Applications of Perceptual HashAlgorithm in Agriculture Images, 2013 IEEE.

[4] ” XUE Jinlin, XULiming “Autonomous Agricultural Robot and ItsRowGuidance ” IEEE 2010.

[5] Sanjay B. Patilet al “Leaf Disease Severity MeasurementUsing Image Processing” / International Journal of Engineering and Technology Vol.3 (5), 297- 301 2011

[6] SajjadYaghoubi, Negar Ali Akbarzadeh, ShadiSadeghiBazargani“Autonomous Robots for AgriculturalTasks and Farm Assignment and FutureTrends in Agro Robots”, International Journal of Mechanical &MechatronicsEngineering IJMME-IJENSjune2013.

[7] Pedersen S. M,Fountas S and Blackmore S “Agricultural Robots – Applications and Economic Perspective”, University of Copenhagen,Institute of Food and Resource Economics.

[8] ” liming wang, Jianboshi, Gang song and I-fan shen“Object detection combining recognitionand segmentation”, Fudanuniversity shanghai