Internet of Things: A Survey-Humidity And Temperature Level Finding Using Image Process Techniques

V.Sanjay[1], Saravanakumar S[2]
Dept of CSE
[1]Kumaraguru college of Technology,Coimbatore.
[2]Adithya institute of technology , Coimbatore

Abstract- *The main objective is to find the humidity and temperature level through the plant's leaf color in a simple and fastest way. This project is useful for students, green house managers and researchers for getting the results accurately in simple manner. By an image processing technique with subject status the device displays the results in accurate way. The system is intended not only for the students but also for green house managers, home owners, nursery owners, gardeners etc. The device allows the detection of a specific color in a live stream video content composed of infinite frames at different time instants detect the color of every frame one by one. Finally the result will be given to the user based on the color of the leaf. The whole result will be under the control of the detection of the leaf's color by image processing technique using object detection and color identification by OpenCV with python*

Keywords- Internet of Things (IoT); Threats; Security; Privacy

I. INTRODUCTION

1.1 General

Humidity (atmospheric moisture) is an important factor of the environment for plant growth and development. The significance of humidity for plants was ignored for many years by physiologists who were intent on assessing the more dramatic effects of radiation, temperature, and mineral nutrition on growth of plants. However, during the last decade there has been a significant amount of amount of humidity research, leading to a recognition and acceptance of the important role that humidity has upon plant growth. Certain vegetation patterns are closely linked with atmospheric moisture levels. Home owners and greenhouse managers have long recognized the importance of increased humidity for better plant growth. They have developed procedures to increase or decrease humidity for certain plants at particular time.

In this system we provide information about find the humidity level of the plant to the people. This system shows the accurate result of the humidity level by the plant's leaf through its RGB Color. So this is extremely handy for people and also for plants to stop loses of too much of water level by identify the humidity level. By this process, stomata open (In order to make stomata open, it is important to keep the humidity level in high) will be keep in the plant which is useful to make photosynthesis possible. Hence this is very useful to greenhouse managers as well as agricultural students.

1.2 Objective

The main objective of our project is to design a portable device that can read humidity and temperature level by using the image process techniques as well as DHT11 sensor (Humidity and Temperature sensor) by reading the data from the plant leaf and store the data in cloud.

1.3 Scope

Our project entitled humidity level and temperature reader device is to read the level of humidity and temperature. The project will be using a DHT11 sensor where the sensitivity is very high but cost efficient. The operation of this device is that once the leaf's color is captured by the camera, The device will run and the LCD will display both the reading of humidity level and temperature level as well as the data will be stored in the cloud. The accuracy of the humidity and temperature is observed by identifying the leaf's color. The limitation of our project is that the device is specifically display the reading of humidity and temperature on the LCD provided and save it in the cloud. The device is portable and available everywhere. By maintaining the correct humidity level in the crops are used to avoid more susceptible to a variety of diseases, fungus and parasites that thrive in humid conditions.

So that's why we are developed this device for the peoples who do home gardening, terrace gardening, nursery

workers and agricultural students. This device will help those people to easily get the humidity and temperature level.

II. LITERATURESURVEY

2.1 Introduction to Image Processing

So Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

- Importing the image via image acquisition tools;
- Analyzing and manipulating the image;
- Output in which result can be altered image or report

2.2 Algorithm used in Image Processing

The algorithm used in Image processing is template matching. The template matching process uses pixels, samples, models or textures as patterns. The recognition function is used to calculate the difference between the features of the image input and the stored template.

2.3 Recent works done on Image Processing

During last few years lots of work have been done in Image processing, like predict the pattern recognition, object recognition, Disease identification, computer vision and many more. Proposing new ideas like these regarding image processing and understanding to result out useful information used for further processing automatically.

2.4 Future Scope

Todeal image processing technology will advance and the visual system of man can be replicated. The future trend in remote sensing will be towards improved sensors. Graphics data is becoming increasingly important in image processing app1ications. The future image processing applications of satellite based imaging ranges from planetary exploration to surveillance applications.

III. SYSTEM ANALYSIS

3.1 Existing System

To The existing systems concentrate mainly with finding the temperature and humidity levels separately using the sensors such as DHT11 and hydrometer (humidity sensor). The sensor needs input pins and it requires careful timing to grab the data

The downside of DHT11 sensor is you can only get new data every two seconds. It is slow in measurement. It has limited measurement range only. The sensors such as DHT11 and hydrometer are could not able to access or not flexible by the people who are do the above mentioned works such as gardening, research purpose and etc.

The humidity sensor senses, measures and reports both moisture and air temperature but not the humidity. The earliest devices give the data about the humidity level and temperature only by the sensors and not by the color of the plant's parts.

Many of the above-mentioned sensors or devices is especially dedicated to a particular set of people, for example, research persons and agricultural students events but it is only reach in who is in the field of science because they only know about thesesensors.

Disadvantages of Existing system

- Even though these devices are working with some peoples that couldn't reach toeveryone.
- These devices are using separate sensors for collect the data and could not save the data in cloud.
- These devices didn't give a flexible toeveryone.
- Needs papers to record the data.
- Takes much more time.

3.2 Proposed system

Many of the above-mentioned sensors or devices is especially dedicated to a particular set of people, for example, research persons and agricultural students events but it is only reach in who is in the field of science because they only know about thesesensors. It is a leaf classification system based on hybrid spatial features involves color and textures descriptors and back propagation network classifier.

Advantages of Proposed system

- The device finds the humidity and temperature resultsinstantly.
- It removes human errors.
- The system provides an unbiasedresult.

- Thus, the system excludes human efforts and savesresources.
- Reduce the time andcost
- Paper less examination
- Results are verified immediately
- Accurate results.

IV. SOFTWARE DESCRIPTION

4.1 Object Detection with Deep Learning and OpenCV

While this original blog post demonstrated how we can categorize an image into one of ImageNet's 1,000 separate class labels it could not tell us where an object resides in image. In order to obtain the bounding box (x, y)-coordinates for an object in a image we need to instead apply object detection. Object detection can not only tell us what is in an image but also where the object is as well. In the remainder of today's blog post we'll discuss how to apply object detection using deep learning and OpenCV.

In the first part of today's post on object detection using deep learning we'll discuss Single Shot Detectors and MobileNets. When combined together these methods can be used for super-fast, real-time object detection on resource constrained devices (including the Raspberry Pi, smartphones, etc.) From there we'll discover how to use OpenCV's

Dnn- module to load a pre-trained object detection network.

This will enable us to pass input images through the network and obtain the output bounding box (x, y)-coordinates of each object in the image. Finally we'll look at the results of applying the MobileNet Single Shot Detector to example input images. In a future blog post we'll extend our script to work with real-time video streams as well.

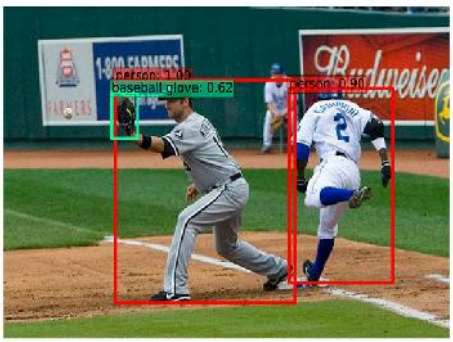


Figure 4.1.1: Single Shot Detectors for Object Detection

Examples of object detection using Single Shot Detectors (SSD) from Liu et al. When it comes to deep learning-based object detection there are three primary object detection methods that you'll likely encounter:

- Faster R-CNNs (Girshick et al., 2015)
- You Only Look Once (YOLO) (Redmon and Farhadi, 2015)
- Single Shot Detectors (SSDs) (Liu et
- al., 2015)

Faster R-CNNs are likely the most "heard of" method for object detection using deep learning; however, the technique can be difficult to understand (especially for beginners in deep learning), hard to implement, and challenging to train. Furthermore, even with the "faster" implementation R-CNNs (where the "R" stands for "Region Proposal") the algorithm can be quite slow, on the order of 7 FPS. If we are looking for pure speed then we tend to use YOLO as this algorithm is much faster, capable of processing 40-90 FPS on a Titan X GPU. The super fast variant of YOLO can even get up to 155 FPS.

The problem with YOLO is that it leaves much accuracy to be desired. SSDs, originally developed by Google, are a balance between the two. The algorithm is more straightforward (and I would argue better explained in the original seminal paper) than Faster R-CNNs.

We can also enjoy a much faster FPS throughput than Girshick et al. at 22-46 FPS depending on which variant of the network we use. SSDs also tend to be more accurate than YOLO. To learn more about SSDs, please refer to Liu et al.

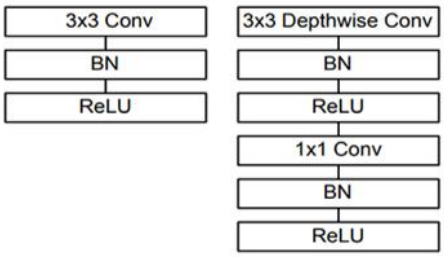


Figure 4.1.2: MobileNets: Efficient (deep) Neural Networks

(Left) Standard convolutional layer with batch normalization and ReLU. (Right) Depthwise separable convolution with depthwise and pointwise layers followed by batch normalization and ReLU (figure and caption from Liu et al.).When building object detection networks we normally use existing network architecture, such as VGG or ResNet, and then use it inside the object detection pipeline. The problem is

that these network architectures can be very large in the order of 200-500MB.

Network architectures such as these are unsuitable for resource constrained devices due to their sheer size and resulting number of computations. Instead, we can use MobileNets (Howard et al., 2017), another paper by Google researchers. We call these networks "MobileNets" because they are designed for resource constrained devices such as your smartphone. MobileNets differ from traditional CNNs through the usage of depthwise separable convolution (**Figure 4.1.2** above).

The general idea behind depthwise separable convolution is to split convolution into two stages:

1. A 3×3 depthwise convolution.
2. Followed by a 1×1 pointwise convolution.

This allows us to actually reduce the number of parameters in our network. The problem is that we sacrifice accuracy. MobileNets are normally not as accurate as their larger big brothers. For more details on MobileNets please see Howard et al.

Combining MobileNets and Single Shot Detectors for fast, Efficient Deep-Learning Based Object Detection

If we combine both the MobileNet architecture and the Single Shot Detector (SSD) framework, we arrive at a fast, efficient deep learning-based method to object detection. The model we'll be using in this blog post is a Caffe version of the original TensorFlow implementation by Howard et al. and was trained by chuanqi.

The MobileNet SSD was first trained on the COCO dataset (Common Objects in Context) and was then fine-tuned on PASCAL VOC reaching 72.7% mAP (mean average precision). We can therefore detect 20 objects in images (+1 for the background class), including airplanes, bicycles, birds, boats, bottles, buses, cars, cats, chairs, cows, dining tables, dogs, horses, motorbikes, people, potted plants, sheep, sofas, trains, and tv monitors.

Components Required

1. Raspberry Pi
2. Power Cable
3. WiFi or Internet

4.5.1 Steps for building Raspberry Pi Data Logger on Cloud

Step 1: Signup for ThingSpeak

For creating your channel on ThingSpeak you first need to sign up on ThingSpeak. In case if you already have account on ThingSpeak just sign in using your id and password.

For creating your account go to www.thinspeak.com Click on signup if you don't have account and if you already have account click on sign in. After clicking on signup fill your details. After this verifies your E-mail id and click on continue.

Step 2: Create a Channel for Your Data

Once you Sign in after your account verification, Create a new channel by clicking "New Channel" button. After clicking on "New Channel", enter the Name and Description of the data you want to upload on this channel. For example I am sending my CPU data (temperature), so I named it as CPU data.

Now enter the name of your data (like Temperature or pressure) in Field1. If you want to use more than one Field you can check the box next to Field option and enter the name and description of your data.

After this click on save channel button to save your details.

Step 3: Getting API Key in ThingSpeak

To send data to ThingSpeak, we need an unique API key, which we will use later in our python code to upload our CPU data to ThingSpeak Website.

Click on "API Keys" button to get your unique API key for uploading your CPU data. Now copy your "Write API Key". We will use this API key in our code.

Step 4: Python Code for Raspberry Pi

Just make a file with any name and .py extension and copy-paste the code and save the file. Don't forget to replace the API key with yours. You can run the python file. Assuming you already installed python in Raspberry pi using command .

Case 1: If you are using monitor screen then just run the code. Now install all libraries.
Case 2: If you are using "Putty" then you should follow other commands after install all.

V. PROJECTDESCRIPTION

The agricultural land mass is more than just being a feeding sourcing in today's world. Indian economy is highly dependent of agricultural productivity. Therefore in field of agriculture, detection of humidity and temperature in plants plays an important role. To detect a plant humidity and temperature in very initial stage, use of automatic detection technique is beneficial. However, during the last decade there has been a significant amount of amount of humidity research, leading to a recognition and acceptance of the important role that humidity has upon plant growth. In such scenarios early detection could have been fruitful.

The existing method for plant humidity detection is simply observed by experts through sensors which identification and detection of plant humidity is done. For doing so, a large team of experts as well as continuous monitoring of plant is required, which costs very high when we do with large farms. At the same time, in some countries, farmers do not have proper facilities or even idea that they can contact to experts. Due to which consulting experts even cost high as well as time consuming too. In such conditions, the suggested technique proves to be beneficial in monitoring large fields of crops. Automatic detection of the humidity and temperature by just taking images on the plant leaves makes it easier as well as cheaper and it protect the plants before they get affect by some other disease.

One of the main reasons for plant disease is low humidity level or extreme humidity level. This supports machine vision to provide image based automatic process control, inspection, and robot guidance to avoid such scenario which is mentioned in above.

5.2 Block Diagram of Raspberry Pi

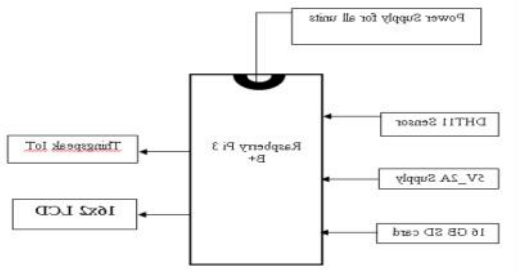


Figure 5.2 Raspberry Pi 3 B+ Connections to DHT11, LCD and Power Supply

5.2.1 Logical Block Diagram

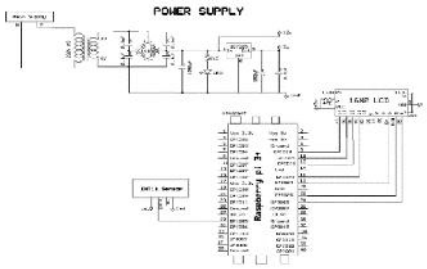


Figure 6.2.1 logical block diagram to connect the power supply with raspberry pi 3 B+ with DHT11

6.2.2 Raspberry Pi 3 B+ and with Software Setting Up

Figure 6.2.2 Raspberry Pi 3 B+

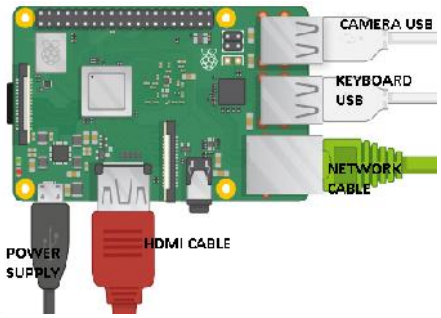


Figure 5.2.2.1 setting up the raspberry pi

5.3 MODULES

5.3.1 Camera Settings

In order to perform runtime operations, the device's web camera is used. To capture a video, we need to create a Video Capture object. Its argument can be either the device index or the name of a video file. Device index is just the number to specify which camera. Normally one camera will be connected, so we simply pass 0. You can select the second camera by passing 1 and so on. After that, you can capture frame-by-frame. But at the end, don't forget to release the capture. Moreover if anyone wants to apply this color detection technique on any image it can be done with little modifications in the code which I'll discuss later.

5.3.2 Capturing frames

The infinite loop is used so that the web camera captures the frames in every instance and is open during the entire course of the program. After capturing the live stream frame by frame we are converting each frame in BGR color space (the default one) to HSV color space. There are more than 150 color-space conversion methods available in Open CV. But we will look into only two which are most widely used ones, BGR to Gray and BGR to HSV. For color conversion, we use the function cv2.cvt Color (input image, flag) where flag determines the type of conversion. For BGR to HSV, we use the flag cv2.COLOR_BGR2HSV. Now we know how to convert BGR image to HSV, we can use this to extract a colored object. In HSV, it is easier to represent a color than RGB color-space. In specifying the range, we have specified the range of blue color. Whereas you can enter the range of any color you wish.

5.3.3 Masking technique

The mask is basically creating some specific region of the image following certain rules. Here we are creating a mask that comprises of an object in blue color. After that I have used a bitwise and on the input image and the threshold image so that only the blue colored objects are highlighted and stored in res.

We then display the frame, res and mask on 3 separate windows using imshow() function.

5.3.4 Display the frame

As imshow() is a function of HighGui it is required to call waitKey regularly, in order to process its event loop.The function waitKey() waits for key event for a “delay” (here, 5 milliseconds). If you don’t call waitKey, HighGui cannot process windows events like redraw, resizing, input event etc. So just call it, even with a 1ms delay.

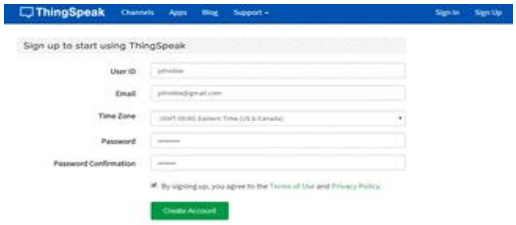
5.3.5 Summarizing the process

- Take each frame of the video.
- Input will be given by plant’s leaf.
- The process will happen to detect the leaf object and leaf color.
- Based on the color the humidity will be find
- The data will be stored in the thingspeak platform

Screenshot:

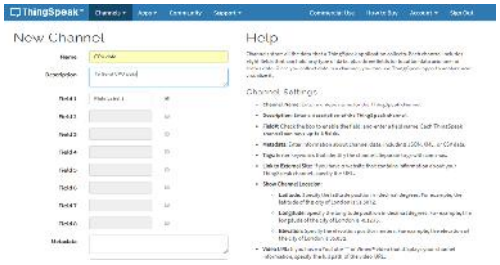
1. Create an account in Thingspeak platform

2. Signup for Thingspeak

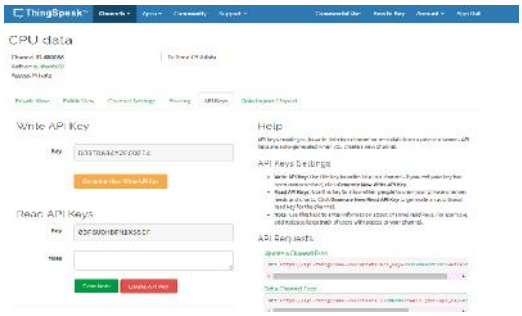


After this we have to verify the email id and click on to continue.

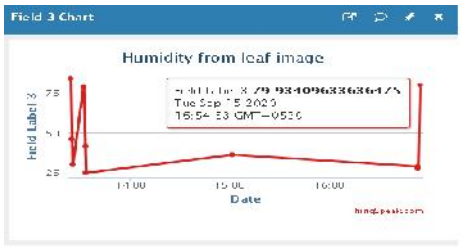
3. Create channel for the data



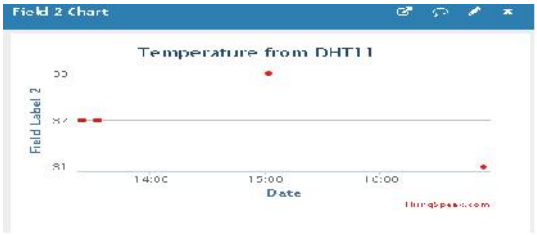
4. Getting API key for thingspeak



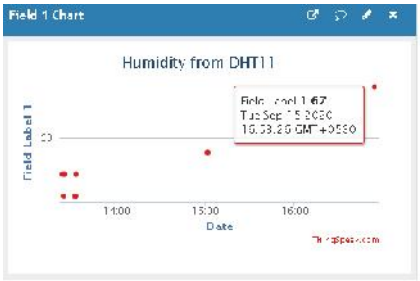
5. Humidity data based on leaf color stored in cloud



6. Temperature data based on DHT11 sensor stored in cloud

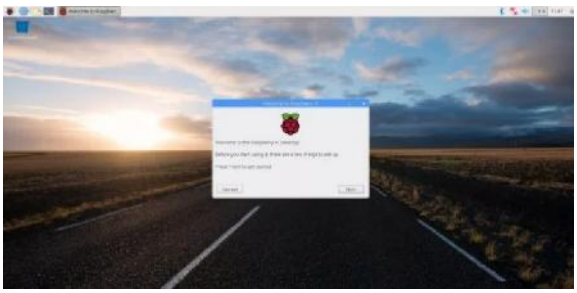


6. Humidity data based on DHT11 sensor stored in cloud



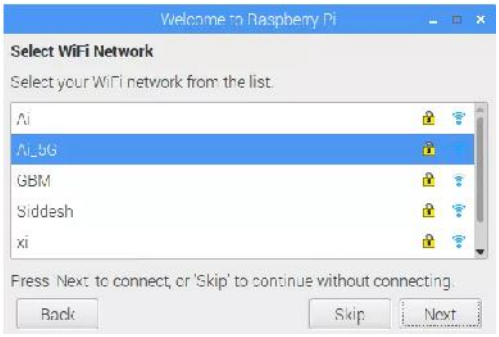
Getting started with raspberry pi

1. Initial configuration of raspbian pi



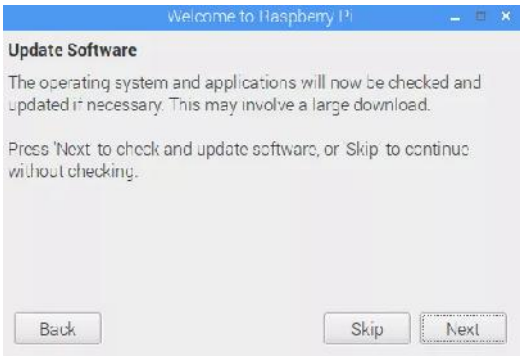
2. Setting new password

3. Connect to WiFi

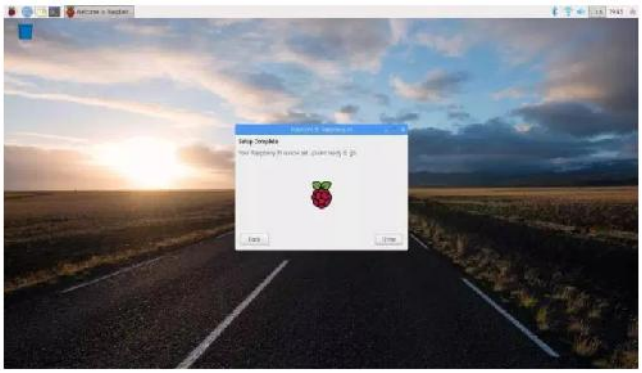


4. Setup Background

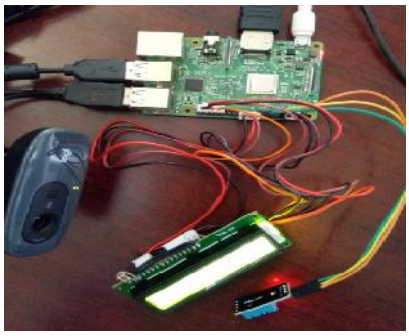
5. Update Raspbian



6. Setup complete

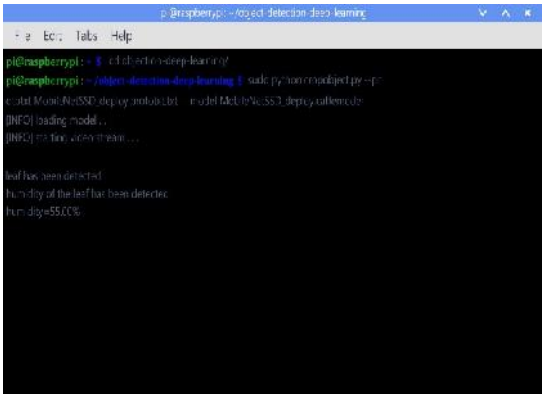


7. Setting up the Things

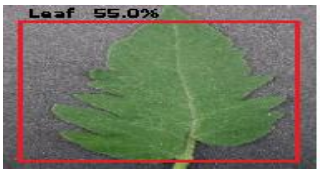


8. Open terminal and run the project

8. Give the input



9. Output of the executed input



10. Output in the LCD Display

VI. CONCLUSION

In the present study, we have discussed the various Image processing techniques which can support to find out the humidity and temperature level information. Since the device of Image processing brings a lot of advantages in higher learning institution, it is recommended to apply these techniques in the areas like computer vision, Face detection, Digital video processing, Remote sensing, Bio medical image enhancement and analyse, Biometric verification, signature recognition and etc.

VII. FUTURE WORK

The Main objective of this project is to collect the data of humidity and temperature level through the plants leaf color and save that data in the cloud. Device main objective is to offer a quick and easy way. The future Government will take this project to develop the crop production in the good level. This application is mainly meant to open the doors for the people who are interest to change the agricultural level in the next level by stop the effects on plants. This project will be a good one to house owners, greenhouse holders and the

agricultural department people because it designed clearly to understand by everyone.

REFERENCES

[1] J. G. ArnalBarbedo, “Digital image processing techniques for detecting, quantifying and classifying plant diseases,” SpringerPlus, vol. 2, article 660, pp. 1–12, 2013.
[2] J. K. Patil and R. Kumar, “Feature extraction of diseased leaf images,” Journal of Signal & Image Processing, vol. 3, no. 1, p. 60, 2012.
[3] J. R. Rohr, T. R. Raffel, J. M. Romansic, H. McCallum, and P. J. Hudson, “Evaluating the links between climate, disease spread, and amphibian declines,” Proceedings of the National Academy of Sciences of the United States of America, vol. 105, no. 45, pp. 17436–17441, 2008.
[4] K. A. Garrett, S. P. Dendy, E. E. Frank, M. N. Rouse, and S. E. Travers, “Climate change effects on plant disease: genomes to ecosystems,” Annual Review of Phytopathology, vol. 44, pp. 489–509, 2006.
[5] S. A. Miller, F. D. Beed, and C. L. Harmon, “Plant disease diagnostic capabilities and networks,” Annual Review of Phytopathology, vol. 47, pp. 15–38, 2009.
[6] S. M. Coakley, H. Scherm, and S. Chakraborty, “Climate change and plant disease management,” Annual Review of Phytopathology, vol. 37, no. 1, pp. 399–426, 1999.