Artificial Intelligence and Machine Learning Operated Pesticide Sprayer

Onkar Kasture[1], B Shivakumar[2], Aham Warikoo[3], Shweta Nangare[4]

[1234]Student, Dept. of Mechanical Engineering, PCCOE, Maharashtra, India

---***---

Abstract - *Robotics has influenced all modern day infrastructures and developments, and is crucial to find its way through traditional agricultural practices that are still mostly done by manpower and human interference. Pesticides being chemical agents, if sprayed excessively on agricultural/organic matter may highly influence its health, and also cause soil degradation to a very high extent leading to degradation of farm fertility, as most of the farmers being un aware of the damages that such excessive insecticide sprays can cause to land and plant health. Thus, our prime motive is to develop a robot governed by Artificial Intelligence and Machine Learning algorithm dedicating itself to spray pesticides only where it is essential, by incorporating machine vision. Thus, analyzing plants, monitoring their health status, and spraying pesticides if and only if plant is prone to infestation or attack.*

Key Words: Artificial Intelligence, Automation, Robotics, Embedded-Electronics, Actuators, Raspberry-Pi, Machine Vision

1. INTRODUCTION

In farms, the farmers need to spray pesticides and have to stay up in the sun and spend a lot of time. Since they cannot look at each plant separately, they tend to spray pesticide on all the plants, which also wastes the pesticides because some plants might not need it and it damages the soil. In case of larger field, the labour costs also increase. Therefore, to minimise these issues we have designed our project.

2. PROBLEM STATEMENT

To develop and design an AIML operated pesticide sprayer which can identify healthy and unhealthy leaves with help of artificial intelligence and machine learning model to help reduce excess wastage of insecticides and help reduce labor and time costs of farmers along with keeping the soil fertility at optimal level.

3. OBJECTIVE

Our prime objective to create a robot which can automatically detect the leaves which are under the attack of insects and require spraying of pesticides so that further damage can be prevented. It is of prime importance to facilitate automation in the field of agriculture, thus boosting accuracy, efficiency and to achieve automation in practices that can be tiring and cumbersome to human workers.

4. METHODOLOGY

The methodology opted for our project is by firstly studying research papers and doing a thorough background verification regarding the need, scope and history of the project. Initially a conceptual design depicting the physical structure and look along with the mechanisms and working of our model are developed. The design of the structure for various loading conditions were analysed. The governing AI models are first decided, finalised, programmed and tested for errors. Then the model are further trained by providing sample photos and datasets to facilitate proper development of the neural network and later on the circuits are designed and simulated using Proteus 8 professional software and the materials are selected according to the design specifications to which our project conforms with.

5. LITERATURE STUDY

5.1 Selection of Project

In their paper [1] talk about how pesticides impact human health and environment. How they affect aquatic ecosystem. This paper also talks about the different ways of ecological effects, detection methods of pesticides, degradation of pesticides in water, soil and their process of metabolism. Types of various pesticides and their chemical structures are differentiated. The methods of pesticides extractions and detection were given special focus. This paper talks about the impacts of their consumption by mother on the new-born. Consistent usage of these chemicals not only degrade the productivity of soil but also the multiple lives by interfering the food chain. To provide a proper impactful solution to this problem we designed our project. [1]

5.2 Concept and Actual Methodology of the Project

This paper offers clear and conceptual idea about the Methodology. This particular paper mainly focuses on early identification of plant disease by monitoring the plants. It talks about 1] Identification 2] capturing and comparing 3] decision making. Our project revolves around these three main points. The plants are first identified. Then through the sensors or scanners, various images are taken and sent to the processor. After taking the images as input, the images are compared. If the picture is found contaminated, then the sprayer comes into the frame to spray. [2]

5.3 System Design of Robot

Module used here is Raspberry PI, which is called single board computer. Being a fast processor with low power consumption makes it beneficiary to use. In addition to it, it is highly reliable with compactness. Hereby, we are using python, which is object oriented programming language. This paper talks about HDMI cable, Raspberry PI zero, Open Source Computer Vision, Camera in brief. Image processing techniques detect plant diseases. A digital camera captures the images of the plant, which is further interfaced with the raspberry pi board. [3]

5.4 Component Selection:

The various components and electronic parts, which are required for smooth functioning of the AI model, integrated with the machine vision system i.e. camera module interfaced with pi is obtained from researching various papers and getting insights from this paper. [4]

5.5 Material and Specification Selection

This paper talks about existing system and advancement in crop spraying and various methodologies of spraying. That involves Fixed Nozzle Spacing, Optimal Spray Coverage and One Target-One Shoot (OTOS). The more use of pesticide can degrade soil and this happens since farmers use manual labor work since they are unskilled. They used a movable spraying gadget as trial instrument to execute one target one shoot spraying strategy, as it is more efficient than other ones. Selection of nozzle plays vital role in design as it decides the amount of spray and fertility of soil. This was amount-spraying mechanism.

In robotic mechanism, this paper mentions that, the plan of case o for the robot was started. Initial test is for planning a case, which would carry a heap of until 2-5 Kgs. Iron is utilized at first as the metal of skeleton. Yet, case is weighing around 1Kg. So for avoiding the stumbling load of the gadget, iron was not preferred as the metal of skeleton; tough aluminium is used in most piece of the suspension and iron has used just in certain areas as a fragile metal rather than aluminium. [5]

6. EMBEDDED ELECTRONICS AND SIMULATION

The robot is a 4-wheeled vehicle operated by DC-motors coupled with the wheels. The total load acting on the wheels would be 3 kg, with each wheel taking a load of 0.75kg or 750 grams. Thus by taking into consideration, the total load of 750 grams acting on the motors, the force acting on each motor would approximately be 14.715 Newton. Therefore, from design calculations, the net Torque acting due to this force is 2.943 Nm. Thus, from above criteria and design calculations, standard 12V DC motor, 8500 rpm is selected.

Body of the robot supports a vertical arm with a horizontal arm coupled to it at one end with the rear end of the horizontal arm (housing the camera unit and the spraying end) held above the plant under inspection at the top of the plant. In addition, the body supports a storage tank (1L) to store liquid pesticide and a mini-liquid pump (6V DC, 0.35 A, 5W, 60ml/min discharge) to pump pesticide from tank to nozzle via hosepipe.

The actuators like the DC-motors and Liquid-pesticide pump are all controlled by interfacing them with Raspberry Pi-4 model -B SBC (Single Board Computer) which is triggered to respond in first hand by the input received from camera module (machine vision). Photos captured by the "Machine Vision System", after being processed by the well-trained neural-network model inside the Raspberry-Pi-4 micro-controller, will decide whether to energize the pesticide pump or not, based on plant health situation.

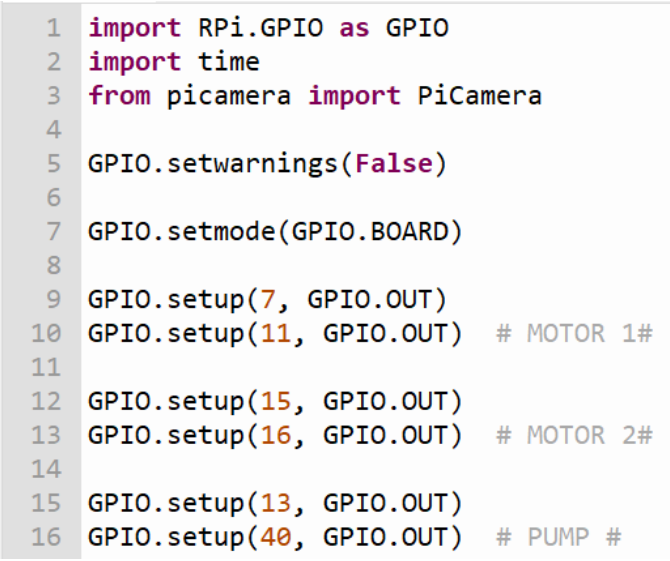
If plant is found healthy, the robot will simply move on to the next plant for analysis, but if found infested by pests, then the

Pi is set to actuate the pesticide pump and pesticide is sprayed on to the infested zone via nozzle.

Communication starts from the Camera module of the raspberry pi. The machine vision system of our model will first capture images of the plant under inspection by clicking its pictures and storing it in the pi memory allocated to it. This images stored in the pi directory will be accessed by our AI program.

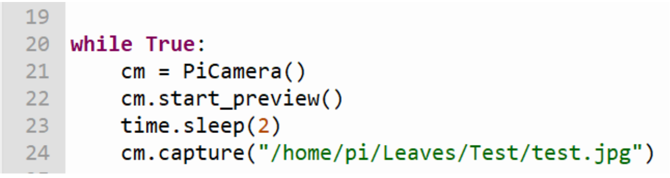
All the essential actuators are controlled by the response received from the processed output of AI algorithm. Based on the category under which the images are captured by the PI-Camera of the raspberry pi SBC by classification model. The raspberry pi is programmed to further actuate corresponding actuators in the system. The main controlling of the actuators in our system like the DC Motors and the DC pumps are mainly controlled by LM298H-bridge motor driver module. It has the ability to control two Motors at a time it has two input sections and two output sections and requires an additional DC voltage supply for amplification of voltage to actuate our actuators. We are using a 14.8V 2200mah Li-Ion battery to power our motor driver module to achieve an overall reduction in system weight.

The DC motors and the DC Pesticide pump are interfaced with LM298N Motor controller module and the controllers are further interfaced with Raspberry Pi. The Raspberry Pi is then interfaced with the camera module to get input pictures of plants to be analyzed. Thus, the flow of the process is starting from the camera module to raspberry pi and from raspberry pi to the motor controllers and from the motor controllers to the motors and the pumps. These processes keep on repeating itself iteration after iteration until the entire row of the agricultural farmland is completely analyzed by the AI algorithm.



```
import RPi.GPIO as GPIO
import time
from picamera import PiCamera

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)

GPIO.setup(7, GPIO.OUT)
GPIO.setup(11, GPIO.OUT)   # MOTOR 1#

GPIO.setup(15, GPIO.OUT)
GPIO.setup(16, GPIO.OUT)   # MOTOR 2#

GPIO.setup(13, GPIO.OUT)
GPIO.setup(40, GPIO.OUT)   # PUMP #
```

Fig-1: Libraries included for Pin Configuration and Pin Designation Instructions

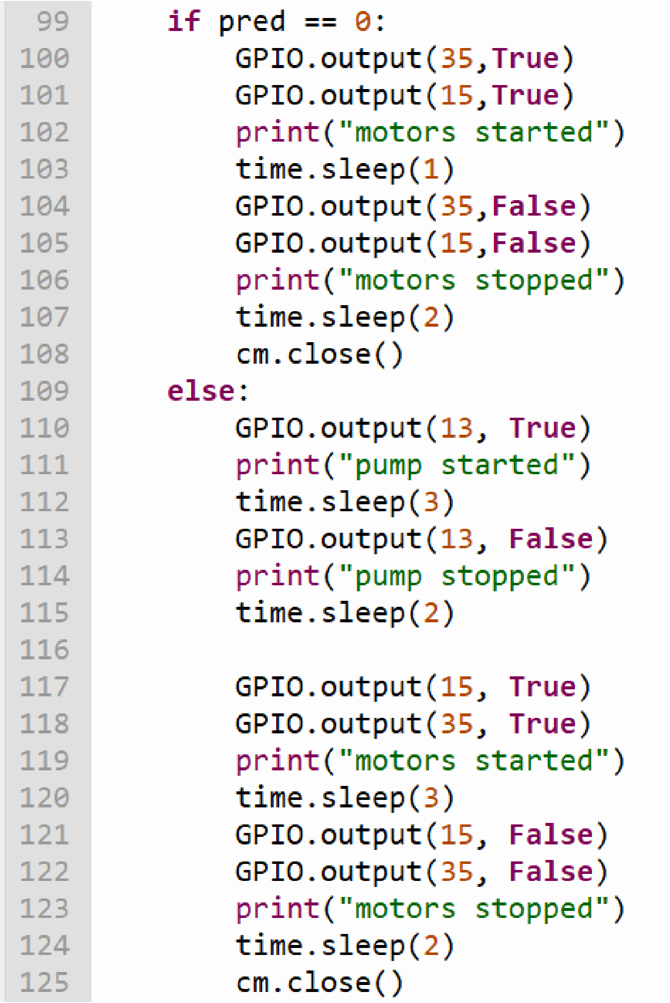


```

while True:
    cm = PiCamera()
    cm.start_preview()
    time.sleep(2)
    cm.capture("/home/pi/Leaves/Test/test.jpg")
```

Fig-2: Camera photo capturing instructions

Based on the logic confirmed to the above literature we will set the pin numbers corresponding to the motors and the pump correspondingly as high or low thus controlling the on/off timing as per the field requirements.

Now the decision making whether the plants are to be sprayed with pesticides by activating the pump or to simply move on to the next plant by activating the motor is done by using an if-else logical statement. After the analysis done by the AI algorithm, if the predicting variable corresponds to logic 1 then the leaves are considered to be unhealthy and the pump is activated corresponding to the if else logical statement. Else, the predicting variable corresponds to logic 0, then the motors are activated and the robot moves to the next plant for further analysis.



```
    if pred == 0:
        GPIO.output(35,True)
        GPIO.output(15,True)
        print("motors started")
        time.sleep(1)
        GPIO.output(35,False)
        GPIO.output(15,False)
        print("motors stopped")
        time.sleep(2)
        cm.close()
    else:
        GPIO.output(13, True)
        print("pump started")
        time.sleep(3)
        GPIO.output(13, False)
        print("pump stopped")
        time.sleep(2)

        GPIO.output(15, True)
        GPIO.output(35, True)
        print("motors started")
        time.sleep(3)
        GPIO.output(15, False)
        GPIO.output(35, False)
        print("motors stopped")
        time.sleep(2)
        cm.close()
```

Fig-3: Logic for Motor/Pump Actuation

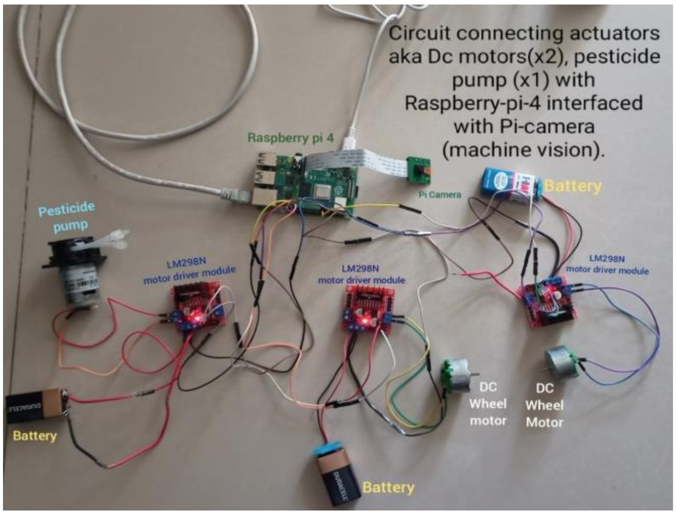

Circuit connecting actuators aka Dc motors(x2), pesticide pump (x1) with Raspberry-pi-4 interfaced with Pi-camera (machine vision).
Raspberry pi 4
Pi Camera
Battery
Pesticide pump
LM298N motor driver module
LM298N motor driver module
LM298N motor driver module
DC Wheel motor
DC Wheel Motor
Battery
Battery

Fig-4: Completed Final Embedded Electronic Circuit

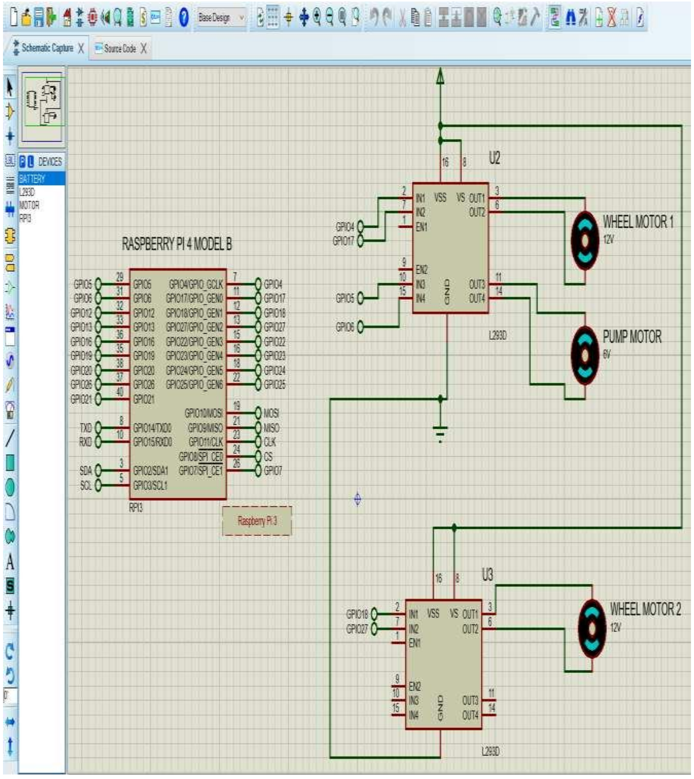
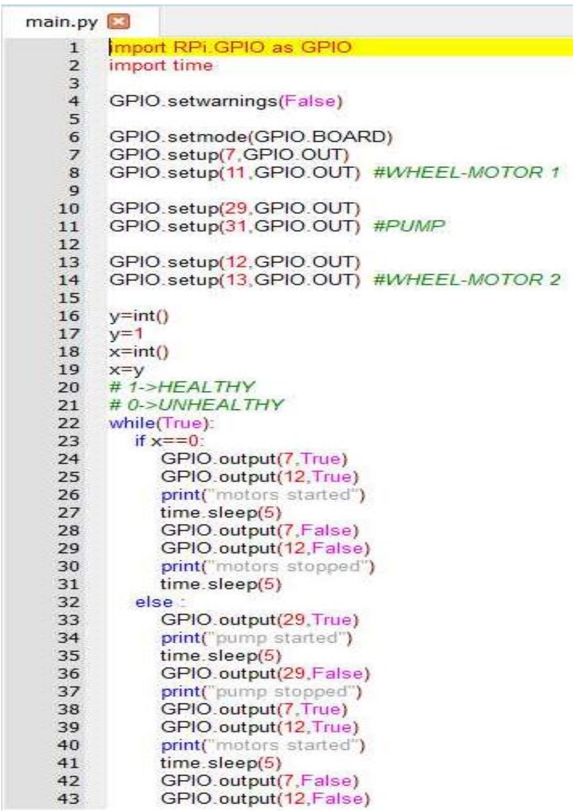


Fig-5: Circuit Simulation on Proteus 8 Professional


main.py

```
import RPi.GPIO as GPIO
import time

GPIO.setwarnings(False)

GPIO.setmode(GPIO.BOARD)
GPIO.setup(7,GPIO.OUT)
GPIO.setup(11,GPIO.OUT)  #WHEEL-MOTOR 1

GPIO.setup(29,GPIO.OUT)
GPIO.setup(31,GPIO.OUT)  #PUMP

GPIO.setup(12,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)  #WHEEL-MOTOR 2

y=int()
y=1
x=int()
x=y
# 1->HEALTHY
# 0->UNHEALTHY
while(True):
    if x==0:
        GPIO.output(7,True)
        GPIO.output(12,True)
        print("motors started")
        time.sleep(5)
        GPIO.output(7,False)
        GPIO.output(12,False)
        print("motors stopped")
        time.sleep(5)
    else :
        GPIO.output(29,True)
        print("pump started")
        time.sleep(5)
        GPIO.output(29,False)
        print("pump stopped")
        GPIO.output(7,True)
        GPIO.output(12,True)
        print("motors started")
        time.sleep(5)
        GPIO.output(7,False)
        GPIO.output(12,False)
```

Fig-6: Back-end Source code for Circuit Simulation on Proteus 8 Professional

7. ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Artificial Intelligence is defined as use of neural networks to train a computer to think like a human brain. While the concept is simple on surface, the actual application is much more nuanced. We are actually using a form of artificial intelligence and not the exact term. We are using Machine Learning, which is defined as training a model through continuous feeding of dataset such that the model is able to understand the patterns and differences and is then able to perform them autonomously. What this means is that we provide the model (code) with known dataset images and known labels. The model then understands the difference between the various labels and develops a logic. This logic is then used to perform the process at much faster rate and autonomously.

7.1 Dataset

Test
Train

Fig-7: AIML model dataset folders

Here we have two folders in our main dataset folder, test and train. The train folder includes all the known images along with their labels; for example, the names of the images contain the information regarding whether the leave in the image is healthy or unhealthy. The model then develops a pattern of similarities between two same categories and difference between two different categories. That means the model will learn to identify which leave is considered healthy and why and how it connects with other healthy leaves. Same with unhealthy leaves, it will learn why unhealthy leave is unhealthy and it will understand the difference between healthy and unhealthy leaves.

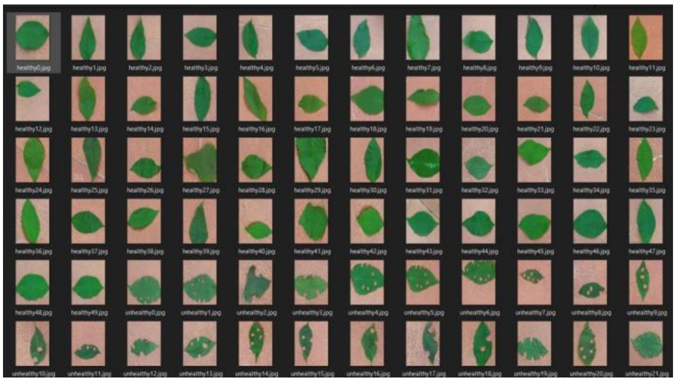


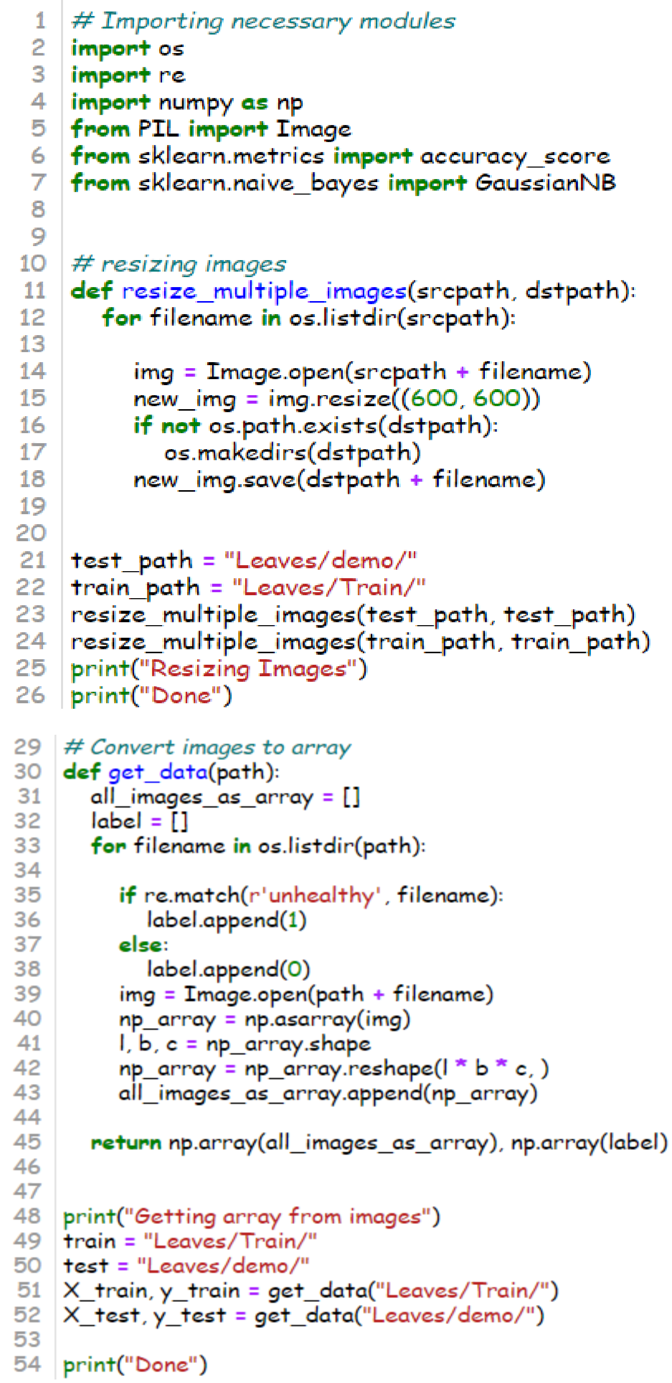
Fig-8: AIML dataset images

This processing takes a lot of time and computational power, doing this locally is generally faster but then you need to have a powerful graphics-processing unit, which are quite expensive.

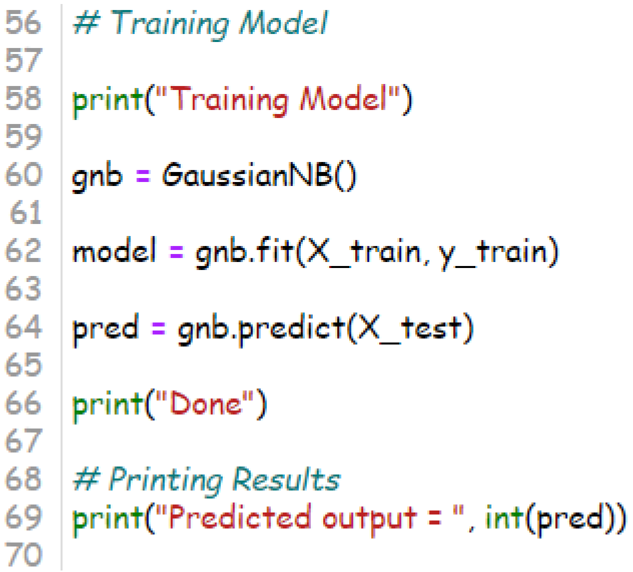
If the machine used for this training of model is slow and weak in terms of computational power, then it might take days or even weeks for the process to finish.

To solve this issue, we are using Google Collab Platform, which is a free platform provided by Google for developers and students which is used for doing the heavy processing. We used GPU based workbooks and the model was trained there, it took around 6 hours for the training to complete since the dataset was also needed to be uploaded to google servers.

7.2 Code



```python
# Importing necessary modules
import os
import re
import numpy as np
from PIL import Image
from sklearn.metrics import accuracy_score
from sklearn.naive_bayes import GaussianNB


# resizing images
def resize_multiple_images(srcpath, dstpath):
    for filename in os.listdir(srcpath):

        img = Image.open(srcpath + filename)
        new_img = img.resize((600, 600))
        if not os.path.exists(dstpath):
            os.makedirs(dstpath)
        new_img.save(dstpath + filename)


test_path = "Leaves/demo/"
train_path = "Leaves/Train/"
resize_multiple_images(test_path, test_path)
resize_multiple_images(train_path, train_path)
print("Resizing Images")
print("Done")

# Convert images to array
def get_data(path):
    all_images_as_array = []
    label = []
    for filename in os.listdir(path):

        if re.match(r'unhealthy', filename):
            label.append(1)
        else:
            label.append(0)
        img = Image.open(path + filename)
        np_array = np.asarray(img)
        l, b, c = np_array.shape
        np_array = np_array.reshape(l * b * c, )
        all_images_as_array.append(np_array)

    return np.array(all_images_as_array), np.array(label)


print("Getting array from images")
train = "Leaves/Train/"
test = "Leaves/demo/"
X_train, y_train = get_data("Leaves/Train/")
X_test, y_test = get_data("Leaves/demo/")

print("Done")
```



```python
# Training Model

print("Training Model")

gnb = GaussianNB()

model = gnb.fit(X_train, y_train)

pred = gnb.predict(X_test)

print("Done")

# Printing Results
print("Predicted output = ", int(pred))

```

Fig-9: Complete AIML code

Here we have the complete code used for the model and machine learning. The code is well captioned for easy understanding of what a particular block of code does and what function it serves.

Following are the steps involved in the code:

1. Importing necessary modules: Here we include all the modules and libraries we will need throughout the code, these modules are downloaded and complied internally whenever it runs on a system so no manual overlook is required.

2. Resizing images: Since the resolution of our camera output images is 1920x1080, it will take a long time to training using that high resolution while the improvement in accuracy will be marginal.

   Hence, we resize the images to 600x600 so that it is easier for the GPU to process them.

3. Converting images to array: The computer does not understand what an image is and how it looks; hence, it is necessary for us to specify the images in a language the computer can understand. Hence, we convert the images into an array (matrix) which includes colour values of all the 600 pixels in the images.

4. Training the model: This is where the magic happens, the array is fed to the code, which then understands the similarities and differences between the pixel color values and develops a logic around that.

5. Results: The logic developed above is then tested with the images from test folder whose data is unknown to the code but known to the accuracy

testing code block. The output of the model is then matched with the known/expected output and the accuracy is calculated.

In our code, we found the average accuracy to be around 85%, which is enough for our usage.

CONCLUSION

The objective of this research work was to develop a low-cost agricultural robot to spray fertilizers and pesticides in agriculture fields as well as for general crop monitoring. The prototype system is a four-wheeled robot that consists of a mobile base, a spraying mechanism and a camera for crop health and growth monitoring. Tests conducted on the robot showed that it could perform as required under real-world usage scenarios. While the productivity of the robot in terms of crop coverage is slightly lower than a human worker, the labor cost savings afforded by the robot is much greater as it functions completely in an autonomous mode and only requires the operator's control to place the robot at the start of the crop path. Furthermore, the robot targets and sprays the liquid fertilizer and pesticide only on the plants and not in the spaces between individual plants, as opposed to the human workers with knapsack sprayer that flood the entire crop path with liquid fertilizer and pesticide. Thus, the system is able to not only reduce the labor requirements and costs, but also result in greater resource savings and reduction of the contamination in underground water sources due to leeching process.

ACKNOWLEDGEMENT

We wish to express our deep sense of gratitude and honour towards our respected guide Mrs. J.P.Wagh for her inspiring guidance and constant encouragement. Her committed devotion, dedication and encouragement with full faith on us were like a lamp in our path that keeps us constant throughout project work.

Special thanks to Prof. Ishan Sathone for his guidance and support at various stages during the course of our project completion. We also express our honour and gratitude to our Head of Mechanical Department Dr. P. A. Deshmukh for consistent encouragement for completing our project work successfully.

We are also thankful to our respected Principal Dr. Govind N. Kulkarni. We are thankful to all Teaching and Non-Teaching staff member of the institute and our classmate who had made us enthusiastic for the project work.

REFERENCES

[1] Pesticides pollution: Classifications, human health impact, extraction and treatment techniques - Mohamed.A.Hassaan, Ahmed.El.Nemr

[2] Development of Smart Pesticide Spraying Robot - Pvr Chaitanya, Dileep Kotte, A. Srinath, K. B. Kalyan

[3] Plant Pathology Detection and Control Using Raspberry-Pi - T. Thamil Azhagi. K. Swetha , M. Shravani & A. T. Madhavi

[4] Pesticide Spraying Agricultural Robot - Arjun Prakash, Bharathi G., Manasa and Gayathri

[5] Leaf Disease Diagnosis and Pesticide Spraying Using Agricultural Robot (AGROBOT) - G.R. Gayathiri, Dr.M.G.Sumithra

[6] Agricultural Robot - A pesticide spraying device - P. Rajesh Kanna, R.Vikram

[7] Design and Operation of Agriculture Based Pesticide Spraying Robot – Amruta Sulakhe, M.N. Karanjkar

[8] Smart Pesticide Spraying Robot – Punit Kanse, Kaustubh Masekar, Chetan Pokale, Prof V. N. Patil.