Guava Size and Health Prediction with AIOT

INTRODUCTION TO IoT
COURSE PROJECT REPORT

Submitted by:
GROUP 3: DSAI
Kapil Soni – 211020424
Sahil Nimsarkar – 211020440
Priyanshu Soni - 211020432

Dr. Shyama Prasad Mukherjee International Institute of Information Technology, Naya Raipur

July-2022

Abstract:

The agricultural sector in India is still the primary source of livelihood for about 58 percent of India's population. As we are the 9th largest exporter of agriculture products, we need farmers equipped with the latest technology and innovations to compete with foreign competitors. As there is a large scale of workers in this field which leads to underemployment which stops the country's growth nowadays, India is more focusing on service sectors. Most farmers should concentrate on cash crops as they are meant for profits. Due to this, we choose Guava as our fruit. It is the fourth most widely grown food crop in India.

Research problem: To estimate Guava's best size and health without human intervention through Machine Learning Computer Vision Models.

Limitations: We could not implement the machine learning model in our Raspberry pi as it only has 1GB RAM, and our model to estimate the size of Guava required more RAM.

Methodology: First, we connect our sensor camera to our Raspberry pi and check its proper function. Then we have to collect data on guava images and their sizes. Then implement our data to train our Open CV model to detect Guava and our ML model to estimate its size.

Result: Our prototype got an accuracy of 92 percent, and our project is ready to be implemented on ground level.

Introduction:

As we have chosen guava as our project, we used Raspberry pi 3B, an HP webcam, and a power bank to set up our project on the guava tree. The guava tree was available on our campus beside the substation. As Raspberry Pi can be used as a computer, it connects with the webcam properly, and we started to do our project.
With time we would have lost motivation, but then we saw our batch mates doing research papers under Debanjan Sir, which gave us all the motivation to do something for the society and welfare of our farmers.
To solve the problem, we have to estimate the size of the guava. First, we have to detect guava, and then to YOLO coordinates of guava in the detected image, and we could apply our machine learning model.

This project contributes greatly to the society as it helps the farmer in many ways like:

- Reduce man labour.
- Maximise Yield and Income
- Farmer can observe guava from his home.
- He would get to know right time to pick up guava.

Objectives:

- To detect the guava in the camera, send it to your mail through Raspberry Pi.
- To estimate the size of the guava through image by extracting its detected coordinates and have four parameters for a machine learning algorithm.
- By the size and health data, we can estimate whether the guava is healthy or not and pick the guava from a tree at the right time.

Methodology:

Our methodology would consist of 3 significant components a Raspberry Pi B3, a Webcam, and a Power Bank to power the Raspberry Pi. With this, we will set up our Webcam first, as it is our sensor, and see if the camera is working correctly. To protect the Raspberry Pi setup from the rain, we covered it in a box.

Fig 1.1 - Project Image

Fig 1.2 - Protected Project

Then to train our model, we would need data, so we took 20-30 images of guava by Webcam with a 20 cm distance between the camera and guavas so that image ratios and resolutions are the same. We also measured the circumference of the guava to detect its size, as healthy and ready-to-eat guava has a circumference of around 22-25 cm.

Fig 1.3 - Capturing the Guava

Fig 1.4 - Photo of Guava

After that, we started to train our YoloV4 model to detect guava through our Webcam with these images and then implement it in our Raspberry Pi to send farmers a mail of the photos of guava only when new guava is detected in the frame or size of guava grows.

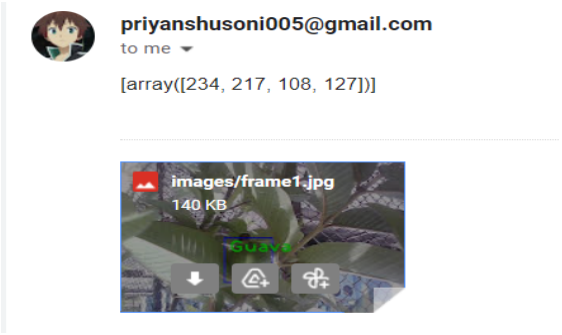

Fig 1.5 - Mailed coordinates

Fig 1.6 - Guava Detection

We used our previously collected data to extract images of guava. Then by calculating YOLO boundary points, there will be four parameters in our machine learning model. All photos must have the same distance from the camera to improve accuracy. We have also split our data set into 70:30 ratio rules for training and testing models.

Sno.	A	B	C	D	Circumference
1	238	240	120	76	8.4
2	113	220	181	242	10.2
3	254	177	108	113	10.8
4	259	0	117	102	9.1
5	350	7	90	102	8.8
6	169	94	95	106	9.3
7	234	217	108	127	10.4
8	325	201	114	123	10.2
9	253	157	102	101	8.3
10	270	193	230	231	9.1

Fig 1.7 - Our DataSet

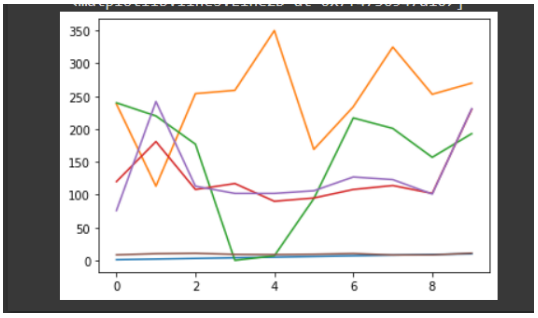

Fig 1.8 - Plotting our Dataset graphs

We started implementing machine learning algorithms like Linear Regression, Polynomial Regression, Decision Tree Regression, and Random Forest Regression and compared their r2 score and accuracy.

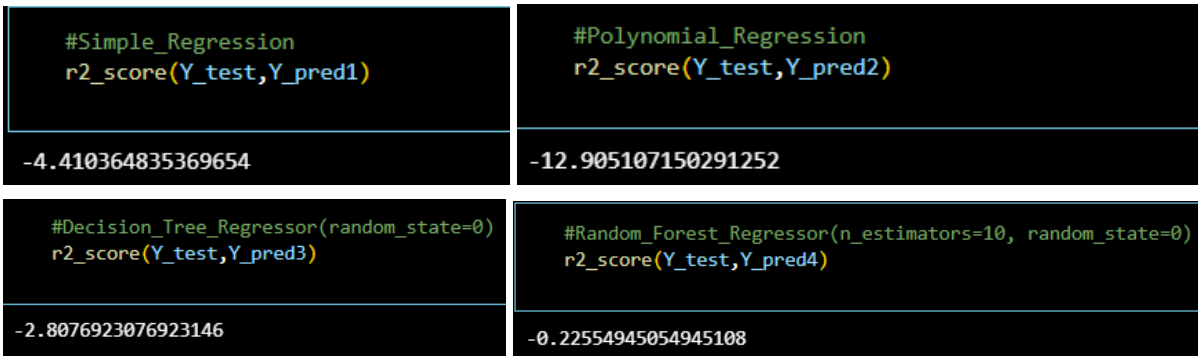

Fig 1.9 - Compared R2 Scores

After this, we implemented Random Forest into all our datasets to train the model and implemented it in our initial code. Now, these new coordinates will be saved in a new file and then would be used to predict the size of the fresh guava.

```
from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators = 10, random_state = 0)
regressor.fit(X, Y)
```

Fig 1.10 - Using Random Forest Regression

A conceptual diagram smoothly guides you through the entire process of our project for better understanding.

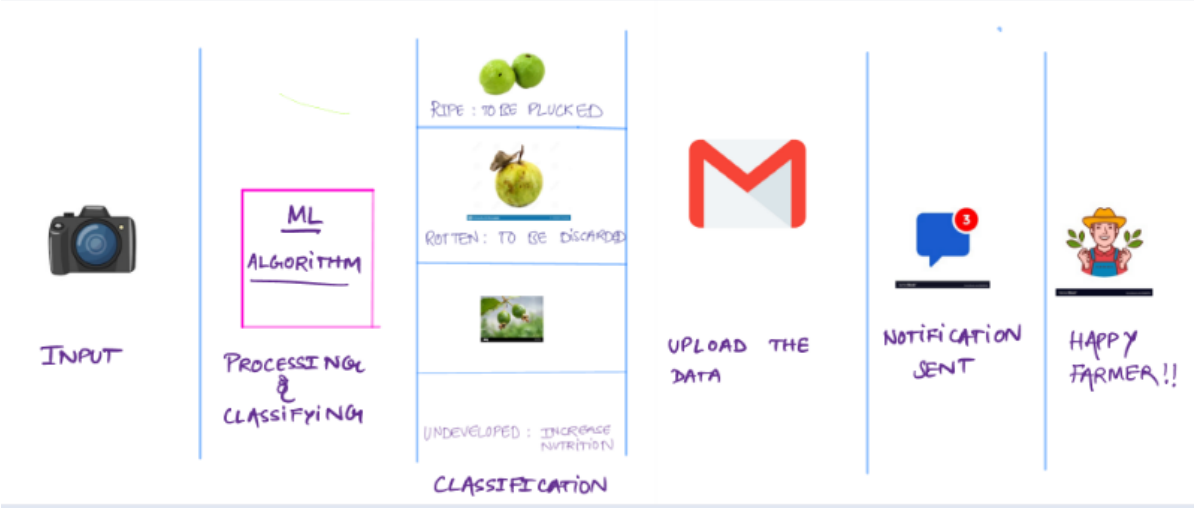

Fig 2.0 - Conceptual Diagram

Working Explanation:

The project is implemented when the Camera and Raspberry Pi setup are installed, with an excellent view of growing guava. The program in the RPi is coded such that it will keep monitoring guava fruit. It sends the dimensions of the guava whenever a change is detected in its size from its prior stored data. This data is then sent to the farmer's smartphone using email for real-time growth tracking of the fruit. The constant picturization of the growth of the fruit provides a database for the farmer for future use and contributes to the data analytics of the project.

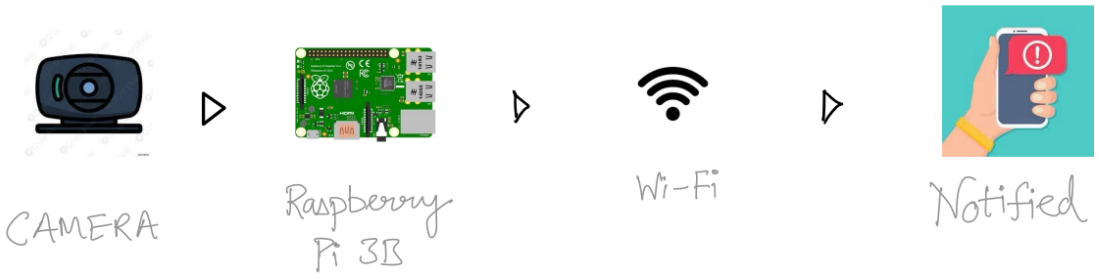

Fig 2.1 - Flowchart Diagram

Design of Sensor Node:

Components used:

Raspberry Pi 3B is our central point for collecting and analyzing information. It detects whether the detected object is guava and sends the specifications of the YOLO-bounding box and its estimated size to the farmer's smartphone over the WiFi.

Fig 3.1 - Raspberry Pi

HP webcam: It is designed for taking images while attached to any device. It is our primary input and constantly takes the information of guava for analysis.

Fig 3.2 - HP Web Camera

We have used Raspberry Pi3B and an HP webcam for our project powered using a mobile power bank. Raspberry Pi3B is chosen due to its high computational power, inbuilt WiFi module, and relatively pocket-friendly and portable size. The HP webcam is connected to the RPi using the USB-B port connected to our laptop(server) over the WiFi. The sensor node diagram provides a clearer view of the process.

Data Analysis:

We got these as our results by further doing three test cases and running it,.

```
print(Y_pred4)
print(Y_test)
```

```
[10.    9.67  9.66]
1    10.2
8     8.3
3     9.1
Name: Circumference, dtype: float64
```

Fig 4.1 - Prediction results

Plotting its graph with three more results as in our data set guava's circumference is only around 8 to 13 cms.

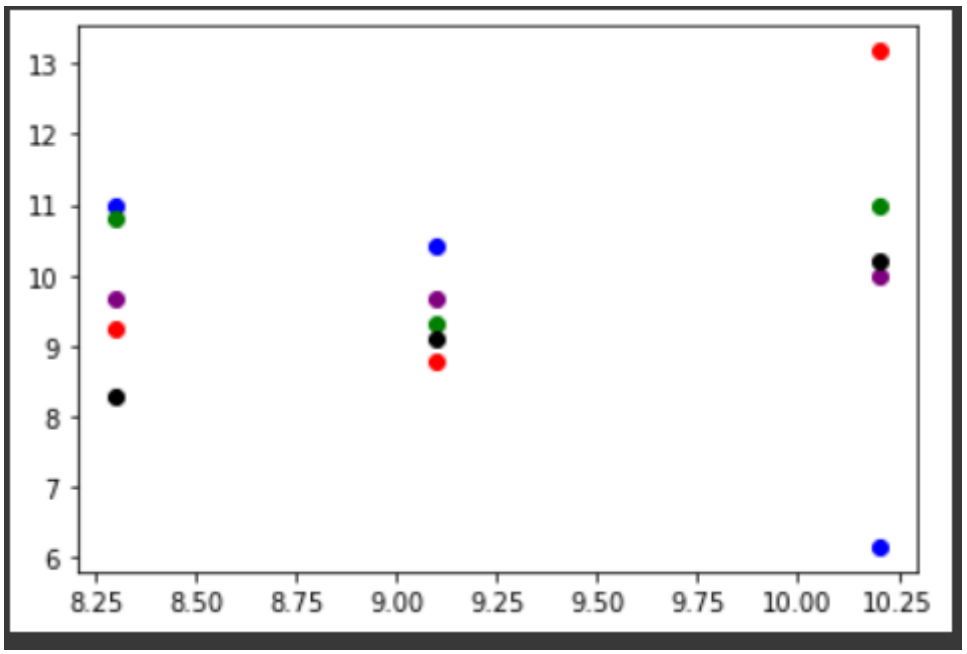

Fig 4.2 - Scatter Plot

Result:

After implementation of the prototype using the above-proposed methodology, the following result is obtained:

It has been deployed on the guava tree near the Electrical Maintenance Room, IIIT-NR and with further experiments we are improving accuracy of the model

Comparison:

https://www.researchgate.net/publication/325121428_IoT_based_Growth_Monitoring_System_of_Guava_Psidium_guajava_L_Fruits

In this paper, they used a tape to measure the dimensions of Guava and an optoelectronic sensor which can be removed by any natural instance of rain or by and living being.

But in our solution, we used a Webcam and fixed it to the tree so that it would be difficult for animals and insects to remove and be stiff and safe under any climatic conditions.

Conclusion:

We have developed an AIOT-based automated crop management system, equipped with on-site cameras to facilitate live detection. Farmers will be able to take in all the affecting parameters in real-time to take informed decisions about harvesting of their crop i.e. guava here. It is directly linked to the mailing system, allowing for an easy interaction. The data collected from this prototype can be further used to detect sizes of guava depending on region helping farmers make better strategies and take smarter decisions.

Future Scopes:

- Make a local host website to send data regularly that would be easy for data visualisation for farmers.
- Also add more dataset to increase accuracy of the model.
- This algorithm can aslo be trained for size detection of different fruits also that are somewhat circular in shape.
- We could also include disease detection models of guava so that it would aslo indicate health of the guava.

Learning outcomes:

Making this project pushed us to expand our knowledge. Some of our new learnings are given below:

- We learned about Raspberry Pi and all its features and components and how to implement its use and deploy it on the ground.
- Then we learned a good level of computer vision(Deep learning models) YOLO, R-CNN, and also to use and implement them.
- In the end, we also got to learn regression models and how to use and implement them.
- With this immense knowledge, we also got the experience of showing updates on time and finishing a project by a deadline.
- With this, we also build team spirit qualities to make sure to do everything for our project.

References:

- IoT based Growth Monitoring System of Guava (Psidium guajava L.) Fruits: https://www.researchgate.net/publication/325121428_IoT_based_Growth_Monitoring_System_of_Guava_Psidium_guajava_L_Fruits
- A Computer Vision System for Guava Disease Detection and Recommend Curative Solution Using Deep Learning Approach https://www.researchgate.net/publication/340058587_A_Computer_Vision_System_for_Guava_Disease_Detection_and_Recommend_Curative_Solution_Using_Deep_Learning_Approach
- GrowFruit: An IoT-Based Radial Growth Rate Monitoring Device for Fruit: https://ieeexplore.ieee.org/document/9566792
- YoloV4 repository: https://github.com/Tianxiaomo/pytorch-YOLOv4
- https://www.youtube.com/watch?v=Vg9rrOFmwHo
- https://www.youtube.com/watch?v=Vg9rrOFmwHo

Source Codes:

1) Main Code:

```
import cv2

import mail_demo

import requests

import numpy as np


def image_resize(image, width = None, height = None, inter = cv2.INTER_AREA):

    # initialize the dimensions of the image to be resized and

    # grab the image size

    dim = None

    (h, w) = image.shape[:2]


    # if both the width and height are None, then return the

    # original image

    if width is None and height is None:

        return image
```

```
    # check to see if the width is None
    if width is None:
        # calculate the ratio of the height and construct the
        # dimensions
        r = height / float(h)
        dim = (int(w * r), height)

    # otherwise, the height is None
    else:
        # calculate the ratio of the width and construct the
        # dimensions
        r = width / float(w)
        dim = (width, int(h * r))

    # resize the image
    resized = cv2.resize(image, dim, interpolation = inter)

    # return the resized image
    return resized

classlabels = [] # empty list of python
file_name = 'Labels.txt'
with open(file_name, 'rt') as fpt:
    classlabels = fpt.read().rstrip('\n').split('\n')

config_file = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
frozen_model = 'frozen_inference_graph.pb'

model = cv2.dnn_DetectionModel(frozen_model, config_file)
```

```
model.setInputSize(320, 320)
model.setInputScale(1.0/127.5) ## 255/2 = 127.5
model.setInputMean((127.5, 127.5, 127.5)) ## mobilenet => [-1, 1]
model.setInputSwapRB(True)

# img = cv2.imread('jeff-david-king-ehWdBropexc-unsplash.jpg')
# img = image_resize(img, height=800)

video_file = "pexels-charles-parker-5825806.mp4"

cap = cv2.VideoCapture(0)
cap.set(3,640)
cap.set(4,480)

font_scale = 3
font = cv2.FONT_HERSHEY_PLAIN

mail_sent = 0
count = 0
lst=[]

while True:
    # images = requests.get("http://192.168.1.22:8080/shot.jpg")
    # video = np.array(bytearray(images.content), dtype = np.uint8)
    # frame = cv2.imdecode(video, -1)

    ret, frame = cap.read()
    # frame = image_resize(frame, height=1024)
```

```
    ClassIndex, confidece, bbox = model.detect(frame, confThreshold=0.52)


    print(ClassIndex)


    if(len(ClassIndex)!=0):
        for ClassInd, conf, boxes in zip(ClassIndex, confidece, bbox):
            ci = ClassInd[0]
            if(ClassInd<=80):
                cv2.rectangle(frame, boxes, (255,0,0), 2)
                cv2.putText(frame, classlabels[ci-1], (boxes[0]+10, boxes[1]+40), font, fontScale =
font_scale, color = (0,255,0), thickness = 3)
                if ClassInd==53:
                    lst.append(boxes)


    flatlist = []
    for sublist in ClassIndex:
        for element in sublist:
            flatlist.append(element)


    if 53 in flatlist:
        if mail_sent == 0:
            count+=1
            cv2.imwrite("images/frame%d.jpg" % count, frame)
            mail_demo.send_mail("images/frame%d.jpg" % count, lst)
            mail_sent = 1
            with open(r'file.txt', 'w') as fp:
                for item in lst:
                    fp.write("%s" % item)
    else:
        mail_sent = 0
```

```
    cv2.imshow("Output", frame)
    cv2.waitKey(1)
```

2) Code to Send Mail:

```
import smtplib
import os
import imghdr
from email.message import EmailMessage

EMAIL_ADDRESS = 'priyanshusoni005@gmail.com'
EMAIL_PASSWORD = 'kdlhniwgrhicwuqg'

msg = EmailMessage()
msg['Subject'] = 'Guava detected'
msg['From'] = EMAIL_ADDRESS
msg['To'] = 'sahiln21102@iiitnr.edu.in'
msg.set_content('A Guava has been detected in the tree , size will be mailed to you later')

smtp = smtplib.SMTP_SSL('smtp.gmail.com', 465)
smtp.login(EMAIL_ADDRESS, EMAIL_PASSWORD)




def send_mail(file,coord):
    msg.set_content(str(coord))
    with open(file, 'rb') as f:
```

```
        file_data = f.read()
        file_type = imghdr.what(f.name)
        file_name = f.name

    msg.add_attachment(file_data, maintype='image', subtype =file_type, filename=file_name)
    smtp.send_message(msg)

    # with smtplib.SMTP('localhost', 1025) as smtp:
    # with smtplib.SMTP_SSL('smtp.gmail.com', 465) as smtp:
    #     smtp.login(EMAIL_ADDRESS, EMAIL_PASSWORD)

    #     smtp.send_message(msg)

def send_mail2(ciord):
    msg.set_content(str(ciord))
    smtp.send_message(msg)

def disconnect_server():
    smtp.quit()
```

3) Code for Predicting Size

```
import numpy as np
import pandas as pd
import mail_demo
mail_sent = 0
count = 0

file1 = open("file.txt","r")
```

```
X_test = []


df=pd.read_csv("Data1.csv")
X=df[['A','B','C','D']]
Y=df['Circumference']


from sklearn.ensemble import RandomForestRegressor
regressor = RandomForestRegressor(n_estimators = 10, random_state = 0)
regressor.fit(X, Y)


for item in file1:
    item=item[1:-1]
    X_test=item.split(" ")


res = []
for ele in X_test:
    if ele.strip():
        res.append(ele)


for item in res:
    float(item)
res=np.array(res)
res=res.reshape(1, -1)
# print(res)
Y_pred=regressor.predict(res)
# print(Y_pred)
if mail_sent == 0:
    count+=1
    mail_demo.send_mail2(Y_pred)
    mail_sent = 1
```

4) Implementing all the regressions and Data Analysis

https://colab.research.google.com/drive/14OBNTTrmswJz9E4KHa0xft29_9n-MgpY?usp=sharing