

Smart Fruit Ripeness Detection Integrating Image Processing and Temperature Sensing Technologies

Tejas Kumar V, Talapaneni Varshith Chowdary, Vikram R Patel

Under The Guidance of Dr.K Badarinath

RV College of Engineering

Keywords—Image Processing, Nodemcu, Fruit Detection, Rgb Technology, Smart System

Abstract:

This paper proposes the design of the system, to detect the ripeness of a fruit (in particular bananas) using temperature readings and image processing techniques. The temperature module MLX90614 measures the temperature of the fruit and the image processing technique analyses an image of the fruit to determine its color. By combining these two readings, the program can determine whether the fruit is ripe or not. The program collects temperature readings from a Nodemcu device and then analyses an image of the fruit using OpenCV library to get the average color of the fruit. Then, it converts the average color from BGR to RGB and passes it through a function that converts the RGB color to a single value. This single value is then used along with the temperature readings to determine if the fruit is ripe or not. In summary, the project provides a way to automatically detect the ripeness of a fruit by analyzing its temperature and color using image processing techniques. This could potentially be useful for fruit processing and harvesting industries

Introduction:

Detection of ripening using image processing is a technique that involves analyzing digital images of fruits and vegetables to detect changes in color, texture, and other visual characteristics that indicate ripeness. This technology has gained a lot of attention in recent years, as it has many potential applications, including improving the accuracy and efficiency of fruit sorting and grading, reducing food waste by identifying ripe fruit for timely harvest and providing

real-time monitoring of fruit ripening for better quality control, the basic process of detecting ripening using image processing involves capturing images of the fruit, processing those images using software algorithms, and analyzing the resulting data to detect changes in color, texture, and other visual characteristics.

We have used python for image processing which has different types of libraries in it. For example, we have used serial library to import temperature data from node mcu to python. We use cv2 library for the finding the rgb values of the fruit and we used NumPy library for performing mathematical operations. We upload the code to the PyCharm (python compiler).

Motivation and Objectives

1. Quality control: Fruit producers and distributors need to ensure that their products are of high quality and meet the desired level of ripeness. A detection system that can accurately assess the ripeness of fruit can help ensure that only high-quality products are sold to consumers

2. Improved efficiency: Currently, the ripeness of fruit is often assessed manually, which is time-consuming and can lead to inconsistencies in assessment. An automated detection system can help streamline the process and reduce the time and cost associated with manual assessment.

3. Reduced waste: Overripe fruit is often wasted, leading to financial losses for producers and contributing to food waste. By detecting the ripeness of fruit, an automated system can help ensure that fruit is harvested and sold at the optimal time, reducing waste.

4. Improved consumer experience: Consumers want to buy fruit that is ripe and ready to eat. A detection

system that accurately assesses the ripeness of fruit can help ensure that consumers are getting high quality products that are ready to be consumed.

Problem Statement & Proposed System:

To determine the ripeness of the fruit by monitoring the temperature using contactless temperature module sensor and analyzing the color changes of the fruit by image processing technique (masked filter) and extracting the rgb values by setting upper bound and lower bound yellow color rgb values. Using the MLX90614 sensor and image processing technique for detecting fruit ripeness has several potential advantages over current systems, like Nondestructive the MLX90614 sensor measures the infrared radiation emitted by the fruit, which means that it can detect ripeness without the need for destructive sampling. This can help reduce waste and increase efficiency in the fruit production process. The MLX90614 sensor and image processing technique can provide more accurate measurements of fruit ripeness than some current systems. Overall, the use of the MLX90614 sensor and image processing technique for detecting fruit ripeness has the potential to improve accuracy, reduce waste, and increase efficiency in the fruit production process, while also being cost-effective and versatile.

SYSTEM ARCHITECTURE

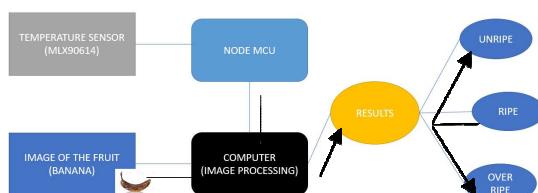


Figure 1: Block diagram of system Architecture

As shown in fig-1 the system is used to determine the ripeness of the fruit by monitoring the temperature using contactless temperature module sensor and analyzing the color changes of the fruit by image processing technique (masked filter) and extracting the rgb values by setting upper bound and lower bound yellow color rgb values.

MLX90614 sensor measures the infrared radiation emitted by the fruit, which means that it can detect ripeness without the need for destructive sampling. This can help reduce waste and increase efficiency in the fruit production process. The fruit temperature values are transmitted to computer using Nodemcu, (an Arduino board also can be used) by establishing serial communication between Nodemcu PC for further Processing.

The PC camera is used to capture the image of the fruit under test, in this case "banana". The captured image is analyzed using image processing technique using OpenCV library and NumPy, with python as the programming language. The proposed algorithm is implemented successfully, to test the state of the fruit using RGB values of the image and temperature values. The result of the state will be like, fruit is ripe, fruit is unripe, overripened.

Design & Implementation:

Hardware Implementation:

As shown in fig-4 Temperature sensor (Fig-3) is connected to Nodemcu using I2C protocol. Nodemcu (Fig-2) is connected to PC using a serial connection via a serial port (UART protocol is used). The data received from the Arduino is read, decoded, and converted to temperature value. The "schedule" library is used to call the main function regularly, to read the updated temperatures from the board.

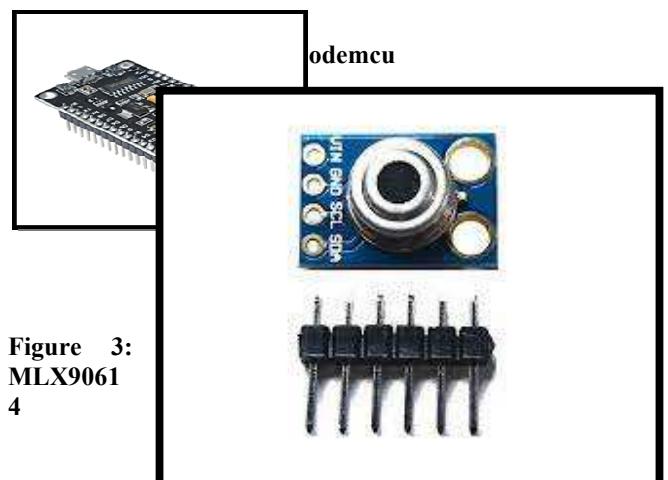


Figure 3:
MLX90614
4

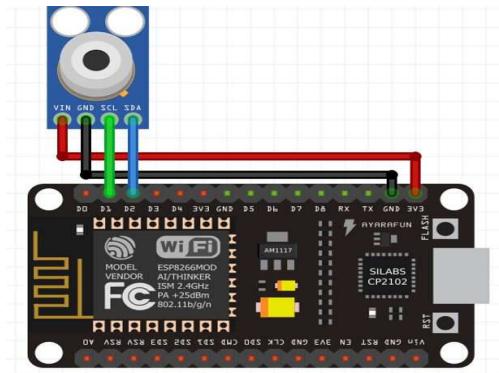
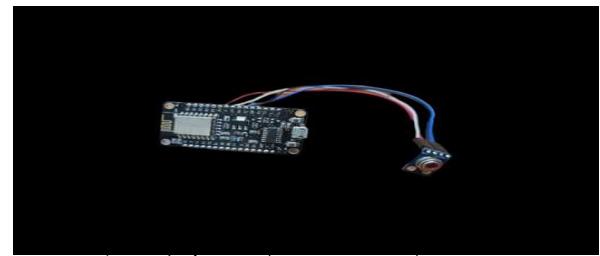


Figure 4: Temperature sensor connection to Nodemcu

Arduino code:

```
#include <Wire.h>
#include <Adafruit_MLX90614.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
#include <Adafruit_Sensor.h>
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
void setup () {
  Serial.begin(9600);
  while (! Serial);
  if (! mlx.begin ()) {
    Serial.println("Error connecting to MLX sensor.
Check wiring.");
    while (1);
  }
}
void loop () {
  float temperature = mlx.readObjectTempF();
  Serial.println(temperature);
  delay (1000);
}
```



implement the functionalities.

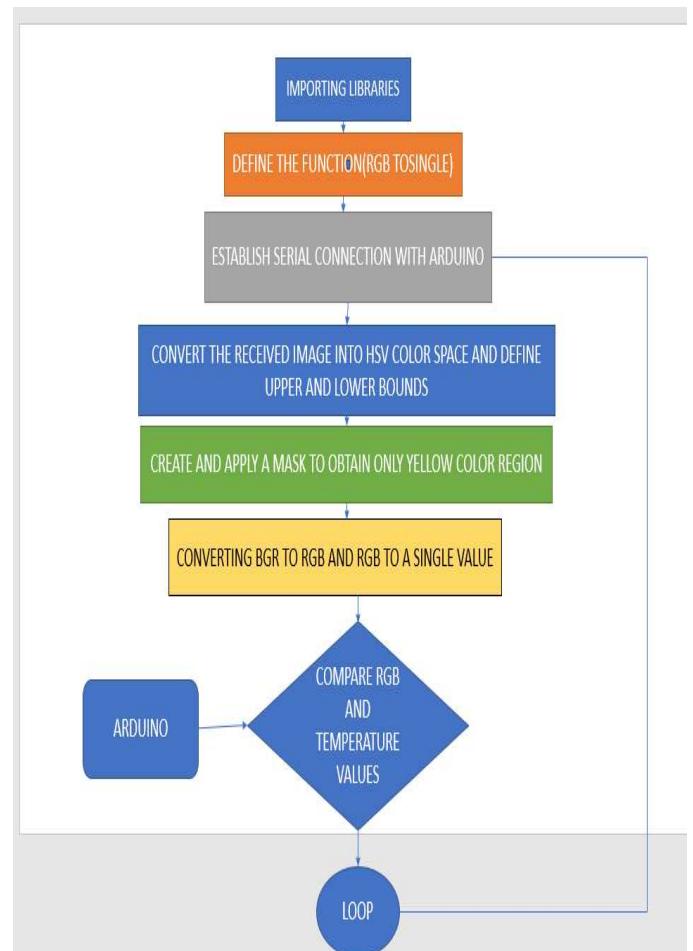


Figure 6: Flowchart of Model Working

The image can be captured live using computers camera/externally connected camera, and fed to the program. An image is loaded using OpenCV (cv2) and converted to the HSV color space. A range of yellow colors in the HSV color space is defined. The image is thresholded to keep only the yellow pixels. The average color of the yellow pixels in the image is computed. The average color is converted from BGR to RGB. The `rgb_to_single` function is used to convert the average color to a single value between 0

and 9, , which is used later to determine fruit ripeness

Depending on the temperature and color value, the ripeness of the fruit is determined.

Python code:

```

import serial
import time
import schedule
import cv2
import numpy as np
import glob

def rgb_to_single(rgb):
    r = rgb[0] / 255.0
    g = rgb[1] / 255.0
    b = rgb[2] / 255.0
    dec = r * 0.3 + g * 0.59 + b * 0.11
    # Scale the value and convert to an integer
    single = ((dec * 9))
    return single

def main_func():
    Arduino = serial.Serial('COM11', 9600)
    print ('Established serial connection to Arduino')
    arduino_data = arduino.readline()
    decoded_values = str(arduino_data[0:len(arduino_data)].decode("utf-8"))
    list_values = decoded_values.split('x')
    for item in list_values:
        list_in_floats.append(float(item))
    print ('collected readings from Arduino: {list_in_floats}')
    temperature = list_in_floats[0]
    img = cv2.imread(r'C:\Users\OMEN\Desktop\el bananas\vikq.png')
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    lower_yellow = np.array ([20, 100, 100])
    upper_yellow = np.array ([30, 255, 255])
    mask = cv2.inRange(hsv, lower_yellow, upper_yellow)
    res = cv2.bitwise_and (img, img, mask=mask)
    avg_color_per_row = np.average (res, axis=0)
    avg_color = np.average (avg_color_per_row, axis=0)
    avg_color = avg_color[::-1]
    print ("RGB values:")
    int_list = avg_color
    str_list = [str(i) for i in int_list]
    result = ",".join(str_list)
    print(result)
    r = rgb_to_single(int_list)
    print(r)
    if temperature < 59:
        print ("Fruit is not ripe")
    elif temperature >= 59 and temperature < 68:
        if r == 0:
            print ("Fruit is not ripe")

```

```

        elif r>0 and r<=1:
            print ("Fruit is Over-ripened")
        else:
            print ("Fruit is ripe")
        else:
            if r == 0:
                print ("Fruit is not ripe")
            elif r > 0 and r<=1:
                print ("Fruit is Over-ripened")
            else:
                print ("Fruit is ripe")
        arduino_data = 0
        list_in_floats. Clear ()
        list_values. Clear ()
        arduino. close ()
        print ('Connection closed')
        print ('<----->')
        list_values = []
        list_in_floats = []
        print ('Program started')
        schedule. every (1). seconds.do(main_func)
        while True:
            schedule.run_pending ()
            time. sleep (1)

```

Explanation of python code:

The above code establishes a connection with an Arduino board and retrieves data from it. It captures an image and processes it to determine the ripeness of a fruit based on temperature and color. The image is converted to the HSV color space, and a specific

range of yellow color is defined. The program then applies a mask to isolate the yellow regions in the image and calculates the average color within that masked area. The average color is converted from the BGR to RGB format and displayed. Additionally, the RGB values are combined into a single decimal value using the `rgb_to_single` function. Based on the received temperature data and the calculated color value, the code determines whether the fruit is ripe, over-ripened, or not ripe. The code uses the `schedule` library to run the main analysis function repeatedly at one-second intervals. It maintains the Arduino connection throughout the process and clears the necessary variables after each iteration.

Result And Analysis

Object(image)	Results
	<pre> Program started Established serial connection to Arduino Collected readings from Arduino: [91.2] RGB values: 0.0,0.0,0.0 0.0 Fruit is not ripe Connection closed <-----> </pre>
	<pre> Program started Established serial connection to Arduino Collected readings from Arduino: [91.49] RGB values: 0.019502958579881656,0.00742603550295858,0.0012294142011834319 0.0002705821441002436 Fruit is Over-ripened Connection closed <-----> </pre>
	<pre> Program started Established serial connection to Arduino Collected readings from Arduino: [91.71] RGB values: 237.73553368403202,192.20708184852109,51.36517902596383 6.7190473443068095 Fruit is ripe Connection closed <-----> </pre>

Figure 7: Results Obtained For various inputs

Related Works:

The same work can be done using other components. An example is shown in a journal. This gives a complete description on the construction of a scanning system which is based on a lidar sensor and servo motor and obtain a point cloud data from oil palm fresh fruit bunch. The mains components are lidar lit v3, Arduino uno and servo motors(2). Here the lidar collects the intensity value that reflects from the virescens oil palm. This data is stored in a csv file for further analysis. Normally the ripeness of the fruit

is checked by human vision but for tall trees or for the fruits with a small change during ripening, we can't predict the ripening. With the use of servo motor, lidar collects the data of reflected intensity values and using lora sensor this data is passed to Arduino Uno and with r studio software the representation of point cloud data can be done. The experts developed a web application to show the type of oil palm fruit bunch. Microsoft Excel data streamer enables live stream data from the lidar sensors. Researchers have calculated the mean intensity values for a ripen and unripen bunch such that they can compare those values and differentiate a new bunch of oil palm fruit using the model. And the model gave very good results.

Acknowledgement:

We are extremely grateful for the RVCE faculty, Dr. K. Badrinath for giving us this golden opportunity to work on this Project that involved several concepts which helped in improving our knowledge. We would also like to extend our sincere thanks to everyone who has helped and supported us throughout the engineering project. We would like to thank them for supporting us and giving us suggestions whenever we needed.

Conclusion:

In conclusion, this project successfully integrates image processing and temperature sensing technologies to create a smart fruit ripeness detection system. By combining these two technologies, the system provides an accurate and efficient method for determining the ripeness of fruits. The image processing component utilizes advanced algorithms and the OpenCV library to analyse fruit images and extract color information. It applies a specific color range to isolate the fruit's color and calculates the average color value. This information is crucial in determining the fruit's ripeness. Simultaneously, the temperature sensing technology measures the fruit's temperature using a dedicated module. Temperature readings play a vital role in assessing fruit ripeness, as temperature changes are often associated with the maturation process. The integration of image processing and temperature sensing enables the system to make informed decisions about fruit ripeness. By considering both color and temperature data, the system provides accurate and reliable results. The application of this smart fruit ripeness detection system holds great potential in various

industries, such as fruit processing, distribution, and harvesting. It allows for automated and efficient quality control processes, optimizing fruit selection and minimizing waste. Overall, this project demonstrates the effectiveness of combining image processing and temperature sensing technologies to develop a smart and reliable fruit ripeness detection system with significant implications for the agricultural and food industries.

Future Scope:

This project can be extended for implementing code on raspberry Pi and a camera module for to capture image to for image processing directly and displaying the state of the fruit, the application of this project could be in the agricultural industry, where farmers can use it to determine the ripeness of fruits such as bananas. the temperature sensor can also provide useful data in monitoring the storage conditions of the fruit, ensuring that they are stored at the correct temperature to maintain their quality, also it is used in fruit packing industry in classifier to maintain quality control and increased efficiency.

References:

- [1] Lidar and its use in Agriculture (BOB HOWARD) [2015] B App Sc (App Geol) Grad Dip (Computer Studies) Grad Dip (Environmental Management), pp 1-4
- [2] Development of Viescens Fresh Fruit Bunch Ripeness prediction using LiDAR for Smart Agriculture (Abdul Dzulijalal, Mohd Sallehin, Siti Rahmah, Azinda Abu) [2021] IEEE Region 10 Symposium (TENSYMP) | ©2021 IEEE DOI: 10.1109/TENSYMP52854.2021.9550981. pp 1-8
- [3] Sustainability of Airborne and Terrestrial Laser Scanning for Mapping Tree Crop Structural Metrics for Improved Orchard Management (Dan Wu, Kasper Johansen, Stuart Phinn and Andrew Robson) [2020]- Received: 8 April 2020; Accepted: 18 May 2020; Published: 21 May 2020 pp 15-20
- [4] Debangshi, 2022. LiDAR Sensor: Applications in Agriculture. Chronicle of Bioresource Management 6(3), 068-070.