

**ĐẠI HỌC ĐÀ NẴNG  
TRƯỜNG ĐẠI HỌC BÁCH KHOA  
KHOA ĐIỆN TỬ - VIỄN THÔNG**



**BÁO CÁO BÀI TẬP 4  
NHÓM 10**

**Sinh viên thực hiện:**     Đinh Văn Quang     20DT1  
   Hà Phước Phúc     20DT2  
   Nguyễn Văn Quý     21DT2  
**Lớp học phần:**             20.38  
**Giảng viên hướng dẫn:** TS. Hoàng Lê Uyên Thục

**Đà Nẵng, 9/2024**

## **Bài 4: Tự động đánh giá độ ngọt của quả xoài dựa vào màu sắc của vỏ.**

### **Cơ sở dữ liệu:**

<https://drive.google.com/drive/folders/1tZZ5MCpfXG0zwehCI-uFIFpkjn4kltL>

**Yêu cầu báo cáo gồm các bước sau:**

### **1. Giới thiệu bài toán**

Mục tiêu của bài toán là tự động đánh giá độ ngọt của quả xoài dựa trên màu sắc của vỏ. Giả sử rằng độ ngọt của quả xoài có liên hệ với sắc màu đỏ trong vỏ quả, tức là cường độ đỏ càng cao thì quả càng ngọt và ngược lại. Dựa trên cơ sở này, ta sẽ thực hiện phân loại quả xoài thành hai lớp: "quả ngọt" và "quả chua" dựa trên hình ảnh của vỏ xoài.

### **2. Mô tả dữ liệu**

- **Nguồn dữ liệu:** Bộ dữ liệu gồm hình ảnh của các quả xoài ở các mức độ chín khác nhau, dữ liệu hình ảnh có 22 mẫu.

- **Kích thước dữ liệu:** Mỗi ảnh có kích thước và độ phân giải khác nhau. Để chuẩn hoá dữ liệu, chúng ta sẽ đưa tất cả ảnh về cùng một kích thước tiêu chuẩn.

- **Nội dung dữ liệu:** Bộ dữ liệu gồm các hình ảnh của quả xoài với nhiều mức độ chín khác nhau, từ đó có thể phân loại dựa trên màu sắc vỏ.

- **Thông tin độ ngọt:** Đánh giá độ ngọt và chua qua màu sắc của quả xoài. Độ ngọt quả xoài phụ thuộc vào màu sắc màu đỏ trong vỏ (cường độ đỏ càng cao thì quả càng ngọt)

+ Quả xoài ngọt: 1, 2, 3, 4, 5, 6, 7, 8, 9, 19, 20, 21

+ Quả xoài chua: 10, 11, 12, 13, 14, 15, 16, 17, 18, 22

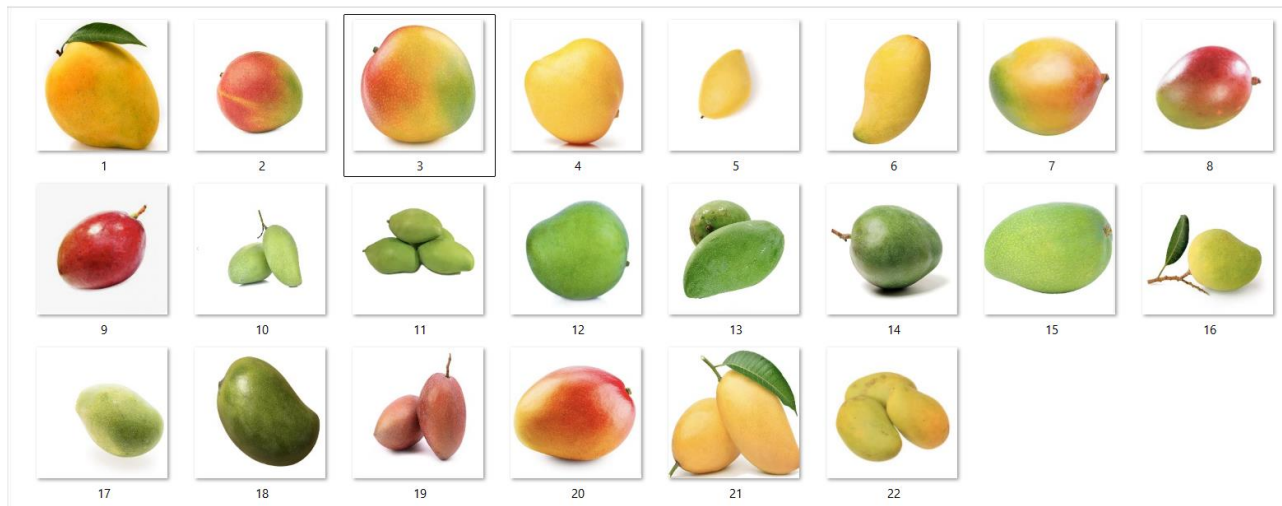
**3. Thực hiện chuẩn hoá dữ liệu: đưa tất cả ảnh về cùng một kích thước. Tiến hành gán nhãn dữ liệu: quả chua/quả ngọt, với giả sử độ ngọt của xoài phụ thuộc vào sắc màu đỏ trong vỏ (cường độ đỏ càng cao thì quả càng ngọt, và ngược lại)**

- **Chuẩn hoá dữ liệu:** Đưa tất cả ảnh về cùng một kích thước 300x300px, để đảm bảo tính đồng nhất trong quá trình xử lý.

- **Gán nhãn dữ liệu:** Dựa vào cường độ ánh sáng màu đỏ, chia thư mục Mango ban đầu thành 2 thư mục nhỏ lần lượt chứa ảnh của các quả ngọt (Sweet) và chua (Sour) như đã mô tả ở phần trên.

**4. Chọn 1 quả xoài bất kỳ trong mỗi lớp để test, phần còn lại để train. Copy và dán ảnh (thu nhỏ) các quả xoài vào đây**

- Chọn quả xoài số 2 để test, phần còn lại để train



**5. Chọn đặc trưng là tỷ lệ cường độ màu đỏ trên tổng cường độ của 3 thành phần màu đỏ (red), lục (green), lam (blue). Tính đặc trưng. Giải thích cách làm + code**

$$\text{Tỷ lệ cường độ màu đỏ} = \frac{\text{Cường độ đỏ}}{\text{Cường độ đỏ} + \text{Cường độ lục} + \text{Cường độ lam}}$$

## a. Đoạn code thực thi chương trình:

```
1 import os
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from PIL import Image
5
6 folder_sour = 'Mango/Sour'
7 folder_sweet = 'Mango/Sweet'
8
9 images_sour = os.listdir(folder_sour)
10 images_sweet = os.listdir(folder_sweet)
11
12 # Tạo mảng lưu các giá trị chỉ số của các hình
13 RGB_sour = np.zeros(len(images_sour))
14 RGB_sweet = np.zeros(len(images_sweet))
15
16 # Hàm tính chỉ số R, G, B
17 def calc_RGB(I):
18     R = np.sum(I[:, :, 0])
19     G = np.sum(I[:, :, 1])
20     B = np.sum(I[:, :, 2])
21     return R / (R + G + B)
22
23 # Tính chỉ số của tất cả các hình trong thư mục
24 for i, img in enumerate(images_sour):
25     I = np.array(Image.open(os.path.join(folder_sour, img)))
26     RGB_sour[i] = calc_RGB(I)
27
28 for i, img in enumerate(images_sweet):
29     I = np.array(Image.open(os.path.join(folder_sweet, img)))
30     RGB_sweet[i] = calc_RGB(I)
31
32 # Vẽ đồ thị
33 plt.figure()
34 plt.stem(RGB_sour, np.zeros(len(images_sour)), 'g', markerfmt='og', label='sour mango')
35 plt.stem(RGB_sweet, np.zeros(len(images_sweet)), 'r', markerfmt='or', label='sweet mango')
36 plt.legend()
37 plt.title('feature of sweet and sour mango')
38 plt.xlabel('R/(R+G+B)')
39 plt.show()
40
41 # Test theo công thức
42 I_test = np.array(Image.open("Mango/Sour/22.png"))
43 testRGB = calc_RGB(I_test)
44
45 if testRGB > 0.3487:
46     print("xoài ngọt")
47 else:
48     print("xoài chua")
49
```

## b. Giải thích cách làm, code:

### Bước 1. Import các thư viện cần thiết:

**os** để làm việc với đường dẫn file và thư mục

**numpy** để xử lý mảng và tính toán

**matplotlib.pyplot** để vẽ đồ thị

**PIL** (Python Imaging Library) để đọc và xử lý ảnh

**Bước 2.** Đọc danh sách file ảnh trong 2 thư mục Sour và Sweet

**Bước 3.** Tạo mảng để lưu trữ cái chỉ số R, G, B:

```
12 # Tạo mảng lưu các giá trị chỉ số của các hình
13 RGB_sour = np.zeros(len(images_sour))
14 RGB_sweet = np.zeros(len(images_sweet))
15
```

- `len(images_sour)` sẽ trả về số lượng ảnh xoài chua
- `len(images_sweet)` sẽ trả về số lượng ảnh xoài ngọt
- Mảng 1 chiều (1D) chứa các số 0, với số lượng phần tử bằng số lượng ảnh xoài chua, xoài ngọt

**Bước 4.** Tạo hàm tính các chỉ số RGB:

```
16 # Hàm tính chỉ số R, G, B
17 def calc_RGB(I):
18     R = np.sum(I[:, :, 0])
19     G = np.sum(I[:, :, 1])
20     B = np.sum(I[:, :, 2])
21     return R / (R + G + B)
22
```

- Tham số đầu vào `I`, là một mảng numpy 3D đại diện cho một ảnh màu RGB
- `I[:, :, 0]` truy cập tất cả các pixel của kênh màu đỏ trong ảnh (0 là chỉ định kênh Red)
- `R = np.sum(I[:, :, 0])` là tổng các giá trị pixel trong kênh Red
- Tương tự đối với kênh Green và Blue
- Kết quả trả về là tỷ lệ giữa tổng cường độ ánh sáng đỏ và tổng cường độ ánh sáng màu

**Bước 5.** Tính chỉ số RGB cho tất cả ảnh

```
23 # Tính chỉ số của tất cả các hình trong thư mục
24 for i, img in enumerate(images_sour):
25     I = np.array(Image.open(os.path.join(folder_sour, img)))
26     RGB_sour[i] = calc_RGB(I)
27
```

- `enumerate(images_sour)` tạo ra một iterator cung cấp chỉ số và giá trị của mỗi phần tử trong `images_sour`
- Vòng lặp này sẽ chạy qua tất cả các file ảnh trong thư mục xoài chua

- $I$  là một mảng 3D đại diện cho ảnh
- $i$  chạy từ 0 đến khi load hết ảnh trong folder
- $RGB\_sour[i] = calRGB(I)$ , tính tỉ lệ  $R/(R+G+B)$  cho ảnh thứ  $i$  và lưu vào mảng  $RGB\_sour$
- Tương tự đối với tính chỉ số RGB cho bộ ảnh ngọt

**Bước 6.** Vẽ đồ thị tỷ lệ  $R/(R+G+B)$  để xác định giá trị ngưỡng  $x^*$

```

32 # Vẽ đồ thị
33 plt.figure()
34 plt.stem(RGB_sour, np.zeros(len(images_sour)), 'g', markerfmt='og', label='sour mango')
35 plt.stem(RGB_sweet, np.zeros(len(images_sweet)), 'r', markerfmt='or', label='sweet mango')
36 plt.legend()
37 plt.title('feature of sweet and sour mango')
38 plt.xlabel('R/(R+G+B)')
39 plt.show()
40

```

- Mảng  $RGB\_sour$  chứa giá trị  $R/(R+G+B)$  cho các ảnh xoài chua (*trực hoành*)
- Trục tung: Tất cả các điểm đều ở vị trí 0 (không mang ý nghĩa số liệu cụ thể)
- 'g': Màu xanh lá cây cho thân cột
- $markerfmt='og'$ : Điểm đánh dấu hình tròn ('o') màu xanh lá cây ('g')
- $label='sour mango'$ : Nhãn cho chú thích đồ thị
- Tương tự đồ thị stem cho xoài ngọt
- $plt.legend()$ : Hiển thị chú thích (legend) cho đồ thị.

**Bước 7.** Kiểm tra ảnh test, so với ngưỡng đã chọn:

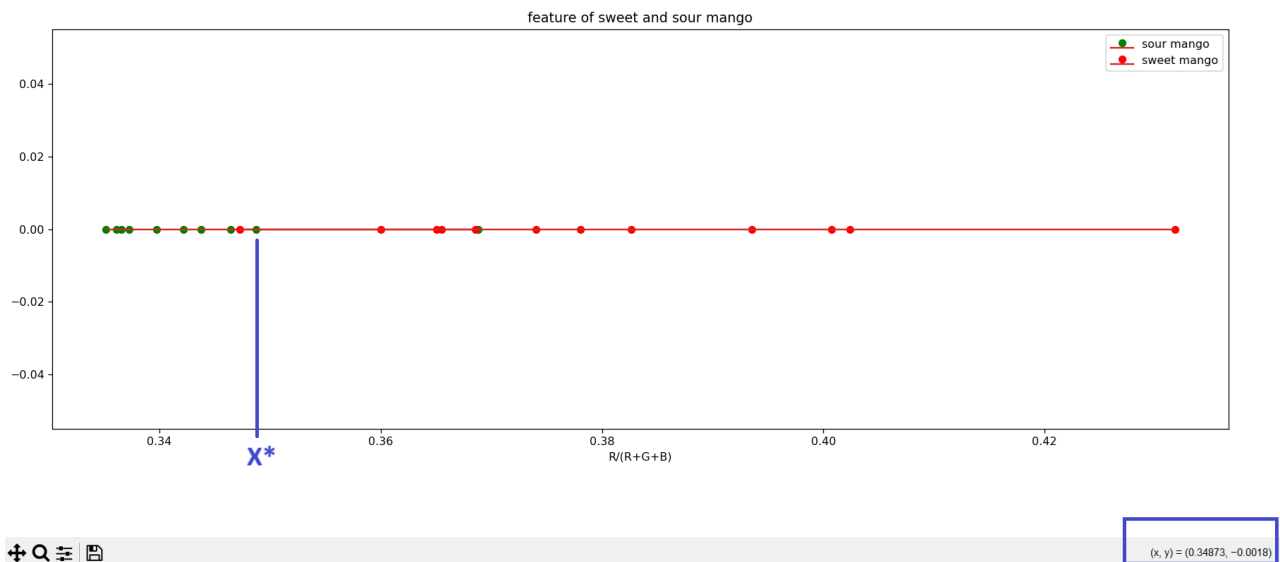
```

41 # Test theo công thức
42 I_test = np.array(Image.open("Mango/Sweet/2.png"))
43 testRGB = calc_RGB(I_test)
44
45 if testRGB > 0.3487:
46     print("xoài ngọt")
47 else:
48     print("xoài chua")
49

```

- Xác định tỷ lệ  $R/(R+G+B)$  của ảnh test và so với ngưỡng  $x^*$  rồi đưa ra kết luận đó là xoài chua hay xoài ngọt

## 6. Chọn ngưỡng



- Từ dữ liệu trên, vẽ được đồ thị stem thể hiện tỷ lệ  $R/(R+G+B)$  của xoài chua và ngọt

- Đưa ra dự đoán giá trị  $X^* = 0.3487$

- Dựa vào đồ thị, các đánh dấu hình tròn màu xanh (xoài chua) thì có xu hướng nằm về phía trái  $X^*$  thể hiện ít sắc đỏ trong quả hơn so với xoài ngọt.

## 7. Kiểm tra

- Ở bước 7 mục 6 đã đưa ảnh 2.png và để test



**Hình 2.png**

## 8. Ghi kết quả

- Theo đánh giá khách quan thì hình 2.png có sắc đỏ cao và dự đoán quả xoài này là xoài ngọt
- Kết quả đánh giá từ mô hình, cho thấy 2.png là quả xoài ngọt

```
# Test theo công thức
I_test = np.array(Image.open("Mango/Sweet/2.png"))
testRGB = calc_RGB(I_test)

if testRGB > 0.3487:
    print("Ảnh test là quả xoài ngọt")
else:
    print("Ảnh test là quả xoài chua")
```



main x

C:\Users\LENOVO\AppData\Local\Programs\Python\Python311\python.exe C:\Users\LENOVO\Desktop\Do\_Ngot\_Xoai\main.py

Ảnh test là quả xoài ngọt

Process finished with exit code 0

## 9. Nhận xét

- Mô hình phân loại dựa trên tỷ lệ cường độ màu đỏ đã đạt được một mức độ chính xác nhất định.
- Tính toán và đưa ra dự đoán giá trị  $X^*$  chưa thực sự tối ưu, có những ảnh quả xoài vừa xanh vừa vàng dẫn đến mô hình không thể dự đoán được chính xác đó là xoài ngọt hay xoài chua
- Việc sử dụng ngưỡng để phân loại cần được tinh chỉnh thêm dựa trên dữ liệu thực tế để cải thiện hiệu suất.
- Mô hình trên có tập dữ liệu train còn hạn chế.
- Những hạn chế của mô hình đơn giản dựa trên tỷ lệ cường độ màu đỏ có thể không đủ chính xác trong trường hợp màu sắc của vỏ xoài bị ảnh hưởng bởi nhiều yếu tố khác, như ánh sáng hay trạng thái chín tự nhiên của quả.

### Tài liệu tham khảo

- [1] <https://answers.opencv.org/question/198437/detecting-percentage-of-red-green-blue-cyan-magenta-yellow/>
- [2] <https://www.geeksforgeeks.org/k-nearest-neighbours/>
- [3] [https://en.wikipedia.org/wiki/RGB\\_color\\_model](https://en.wikipedia.org/wiki/RGB_color_model)
- [4] <https://realpython.com/python-opencv>



Source code: [Module1 AI 24 25/Do Ngot Xoai at main - haphucc/Module1 AI 24 25](#)