TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH

MÔN HỌC
CS231 - NHẬP MÔN THỊ GIÁC MÁY TÍNH

BÁO CÁO ĐỒ ÁN
PHÂN LOẠI TRÁI CÂY TƯƠI

Giảng viên hướng dẫn: TS. Mai Tiến Dũng

Nhóm: 2

Sinh viên thực hiện: Nguyễn Khánh Tuấn Anh - 22520055
Trịnh Quốc Bảo - 22520125
Đinh Nhật Trường - 22521575

TP.HCM, Ngày 07 tháng 05 năm 2024

MỤC LỤC

BẢNG PHÂN CÔNG, ĐÁNH GIÁ THÀNH VIÊN:

Bảng 1. Bảng phân công, đánh giá thành viên

Họ và tên	MSSV	Phân công	Đánh giá
Nguyễn Khánh Tuấn Anh	22520055	Tìm hiểu trích xuất đặc trưng và data Tìm hiểu và thực hiện thuật toán LDA Viết báo cáo và làm slide Build flask để demo	10/10
Đinh Nhật Trường	22521575	Tìm hiểu trích xuất đặc trưng và data Tìm hiểu và thực hiện thuật toán LDA Viết báo cáo và làm slide Build flask để demo	10/10
Trịnh Quốc Bảo	22520125	Tìm hiểu trích xuất đặc trưng và data Tìm hiểu và thực hiện Random Forest Viết báo cáo và làm slide Build flask để demo	10/10

1 MỞ ĐẦU

1.1 Tổng quan

Phân loại trái cây tươi là một lĩnh vực quan trọng trong ngành công nghiệp thực phẩm và nông nghiệp. Việc phân loại thủ công tốn nhiều thời gian, công sức và có thể dẫn đến sai sót. Do vậy, ứng dụng thị giác máy tính vào việc phân loại trái cây tự động là một hướng đi tiềm năng, mang lại nhiều lợi ích.

1.2 Lý do chọn đồ án

Nhu cầu tiêu thụ trái cây tươi ngày càng cao, dẫn đến áp lực gia tăng cho việc phân loại và kiểm tra chất lượng sản phẩm.Phân loại thủ công tốn nhiều thời gian, công sức và chi phí, đồng thời tiềm ẩn nguy cơ sai sót cao. Hệ thống phân loại tự động giúp phân loại trái cây nhanh chóng, chính xác và hiệu quả hơn. Qua đó giúp giảm thiểu sai sót, tiết kiệm chi phí, nâng cao chất lượng sản phẩm và đảm bảo vệ sinh an toàn thực phẩm.

1.3 Cài đặt

IDE: Google Colab, Kaggle.

Ngôn ngữ lập trình: Python

2 NỘI DUNG

2.1 Phát biểu bài toán

Mô tả input: Trong nghiên cứu này, đầu vào là ảnh số có chứa trái cây cần phân loại.

Figure 1: Dữ liệu vào

Mô tả output: Đầu ra của bài toán là kết quả được phân loại là hỏng (rotten) hoặc tươi (fresh).

2.2 Cơ sở lý thuyết

2.2.1. Histogram ảnh màu

Trích xuất đặc trưng là một quá trình chuyển đổi dữ liệu đầu vào phức tạp thành một dạng đơn giản hơn để biểu diễn dữ liệu, phù hợp hơn cho việc học máy. Trong quá trình này,

dữ liệu được loại bỏ dư thừa và giữ lại các thông tin hữu ích cho bài toán.

Histogram là một phương pháp mô tả đặc trưng quan trọng trong lĩnh vực xử lý ảnh và nhận dạng đối tượng. Đặc biệt, Color Histogram được sử dụng để trích xuất đặc trưng từ một hình ảnh bằng cách tính toán phân bố của các giá trị màu sắc trong ảnh. Phương pháp này giúp biểu diễn một hình ảnh dưới dạng một vector, phản ánh phân bố của các màu sắc trong hình ảnh đó.

Có ba bước cơ bản để xây dựng một vector histogram cho hình ảnh:

1. **Xác định các khoảng giá trị (bins):**
   - Histogram chia không gian màu của hình ảnh thành các khoảng giá trị (bins). Mỗi bin đại diện cho một phạm vi của các giá trị màu. Việc lựa chọn số lượng bins là quan trọng, vì nó ảnh hưởng trực tiếp đến độ chi tiết và độ mịn của histogram.
2. **Duyệt qua tất cả các giá trị pixel và đếm số lượng pixel rơi vào mỗi khoảng bin:**
   - Trong bước này, ta sẽ duyệt qua từng pixel của hình ảnh và kiểm tra giá trị màu của pixel đó. Giá trị màu này sẽ được gán vào một bin tương ứng trong histogram. Quá trình này được thực hiện cho tất cả các pixel trong hình ảnh, tạo ra một phân bố số lượng pixel cho mỗi bin.
3. **Chuẩn hóa lại vector histogram:**
   - Sau khi đếm xong số lượng pixel cho mỗi bin, histogram có thể được chuẩn hóa (normalize) để đảm bảo rằng các giá trị của histogram nằm trong một phạm vi cố định, thường là từ 0 đến 1. Việc chuẩn hóa giúp histogram không phụ thuộc vào kích thước của hình ảnh và cho phép so sánh giữa các hình ảnh khác nhau.

2.2.2. Support Vector Machine (SVM)

SVM là một thuật toán học máy có giám sát dùng để phân chia dữ liệu thành các nhóm riêng biệt. Phương pháp máy vector hỗ trợ SVM ra đời từ lý thuyết học thống kê do Vapnik và Chervonekis xây dựng năm 1995. Mục tiêu của SVM là xây dựng một siêu phẳng giữa hai lớp sao cho khoảng cách từ nó tới các điểm gần siêu phẳng nhất của hai lớp là cực đại.

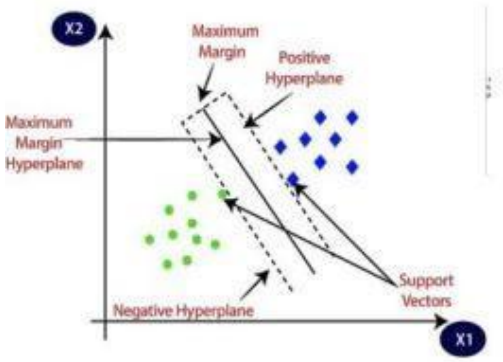

Figure 2: Thuật toán máy hỗ trợ (SVM)

Các vector đặc trưng Histogram thường được sử dụng làm đầu vào cho bộ phân loại SVM. Với dữ liệu huấn luyện được gắn nhãn, SVM học cách phân chia các vector đặc trưng của các lớp khác nhau bằng cách tìm một siêu phẳng tốt nhất để tách chúng. Siêu phẳng này được chọn để tối đa hóa khoảng cách từ các vector đặc trưng đến siêu phẳng.

Kernel là một phần quan trọng trong SVM vì nó cho phép SVM xử lý các bài toán phân lớp phi tuyến bằng cách ánh xạ dữ liệu từ không gian gốc sang một không gian mới có số chiều cao hơn. Thay vì tìm một siêu phẳng tuyến tính trong không gian gốc, SVM sẽ tìm một siêu phẳng tuyến tính trong không gian mới được ánh xạ bằng kernel.

2.2.3. Random Forest (RF)

- **Giới thiệu**

Random Forest là một thuật toán máy học có giám sát (supervised learning) được sử dụng phổ biến cho các bài toán phân loại (classification) và hồi quy (regression). Thuật toán này được đề xuất bởi Leo Breiman vào năm 2001 và sau đó được sử dụng và phát triển rộng rãi trong nhiều lĩnh vật khác nhau. Random Forest là một phần của họ các thuật toán "ensemble learning", có nghĩa là nó kết hợp nhiều mô hình học máy để tạo ra một mô hình mạnh mẽ hơn. Mục tiêu chính của Random Forest là đưa ra kết quả cuối cùng từ nhiều cây quyết (Decision Tree) định độc lập nhau.

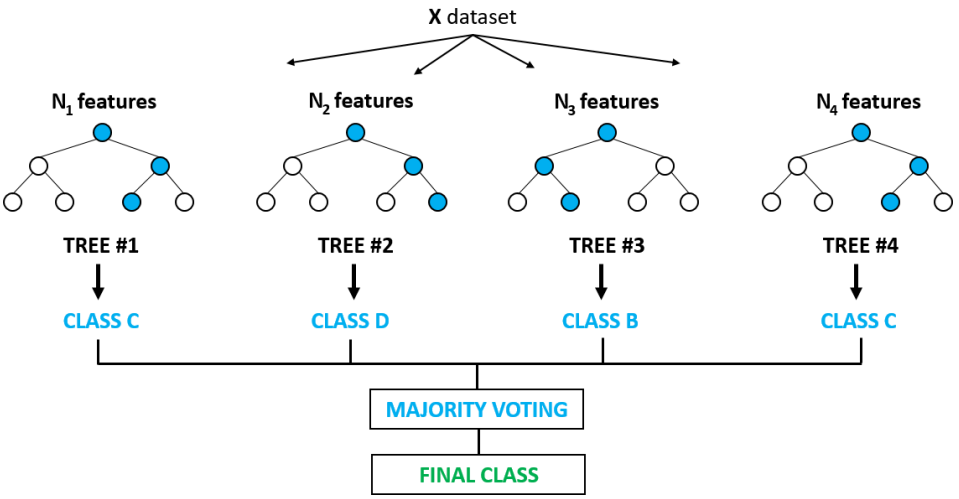

Figure 3: Rừng Ngẫu Nhiên (Random Forest)

- **Cơ chế hoạt động:**

- Bootstrap aggregation (Bagging)

+ Bootstrap aggregation hay còn được gọi lại Bagging là một kĩ thuật học máy nhằm để cải thiện hiệu suất và độ ổn định của của các thuật toán dự đoán. Với ý tưởng chính là kết hợp nhiều mô hình học máy đơn giản để tạo ra một mô hình phức tạp và mạnh mẽ hơn. Các thức hoạt động chính:

+ Lấy mẫu dữ liệu (Bootstrap Sampling): Từ tập dữ liệu gốc chứa N mẫu, tạo ra M tập dữ liệu con bằng phương pháp lấy mẫu ngẫu nhiên có hoàn lại (bootstrap sampling). Mỗi tập dữ liệu con cũng sẽ chứa N mẫu có độ dài giống như tập dữ liệu gốc, nhưng có thể bị trùng lặp mẫu hoặc không chứa một số mẫu khác.

+ Huấn luyện mô hình (Model): Đối với Random Forest, sẽ dùng N mô hình RF để đi huấn luyện với mỗi tập dữ liệu con cho bài toán dự đoán trái cây tươi hay hỏng.

+ Dự đoán (Prediction): Sau khi các mô hình RF đã được huấn luyện, sử dụng mô hình để đưa ra dự đoán cho mẫu mới

+ Tổng hợp lại kết quả: Ở bài toán này ta sẽ sử dụng phương pháp voting để xác định nhãn được chọn nhiều nhất làm nhãn cuối cùng cho bài toán này.

Với phương pháp Color Histogram, Random Forest sử dụng các vector đặc trưng Color Histogram làm dữ liệu đầu vào cho bài toán phân loại. Mỗi vector đặc trưng Histogram biểu diễn một biểu đồ phân phối tần suất của các đặc điểm màu sắc trong hình ảnh. Trong quá trình huấn luyện, Random Forest tạo ra một tập hợp các cây quyết định, mỗi cây sẽ học từ các mẫu dữ liệu biểu diễn bằng các vector đặc trưng Histogram. Khi có dữ liệu mới, mỗi cây sẽ đưa ra dự đoán của riêng mình dựa trên các đặc trưng Histogram của hình ảnh đó, và kết quả cuối cùng sẽ được quyết định thông qua phương pháp như voting để chọn ra lớp phân loại cuối cùng.

2.2.4. LDA (Linear Discriminant Analysis)

LDA là một phương pháp phân loại và giảm chiều dữ liệu cho bài toán classification. LDA đã được đề xuất bởi Ronald A. Fisher vào năm 1936 và sau đó được phát triển và mở rộng bởi nhiều nhà nghiên cứu khác. Mục tiêu của LDA là tìm một không gian mới với số chiều nhỏ hơn không gian ban đầu sao cho hình chiếu của các điểm trong cùng 1 lớp lên không gian mới này là gần nhau và ngược lại.

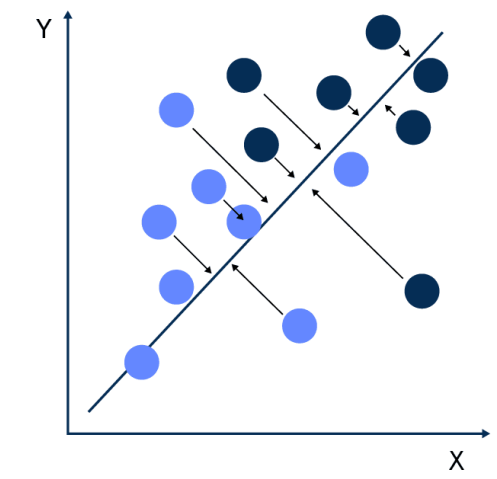

Figure 4: Thuật toán LDA

Cũng như thuật toán SVM ở trên thì đầu vào cũng là các vector Histogram. LDA sẽ tìm chiều dữ liệu con của vector Histogram mà cực đại hóa được giá trị trung bình giữa các lớp và cực tiểu hóa được độ lệch chuẩn trong mỗi lớp.

3 DATASET

3.1 Thông tin bộ dữ liệu sử dụng

- Dataset: Fruits fresh and rotten for classification
- Số lượng: 10901 training images (4740 fresh,6161 rotten) and 2698 test images (1164 fresh,1534 rotten)
- Số trái cây: 3 (Orange, Apple, Banana)

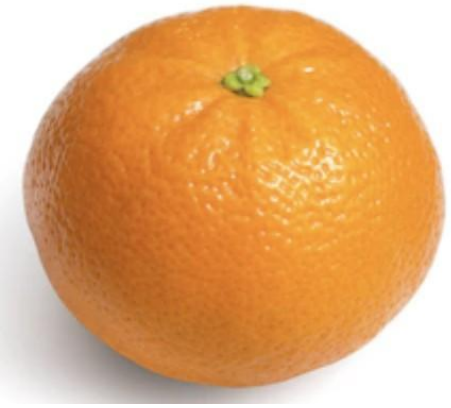

Figure 5: Ví dụ về một hình ảnh có trong bộ dữ liệu

3.2 Chuẩn bị dữ liệu cho phân loại trái cây tươi

Tiền xử lý dữ liệu

Để chuẩn bị dữ liệu cho việc đào tạo mô hình phân loại trái cây, nhóm sử dụng tập dữ liệu đã cho,chia thành 2 nhóm trái cây tươi (fresh), trái cây hỏng (rotten). Các thuật toán như SVM, LDA, Random Forest được dùng để phân chia dữ liệu thành các nhóm riêng biệt nên phải sử dụng mẫu positive và mẫu negative để mô hình có thể học được cách phân biệt đặc trưng Histogram giữa lớp này và lớp khác.

Trước khi tính toán đặc trưng,nhóm resize ảnh lại thành kích thước (224,224).

Tập dữ liệu được chia thành 3 phần train ,validation ,test.Tập dữ liệu val dùng để đánh giá và tinh chỉnh tham số mô hình.

Trích xuất đặc trưng

Histogram được sử dụng để trích xuất đặc trưng, các tham số trong đồ án này là bin trên mỗi kênh là 8. Vector đặc trưng được tính toán và thu được vector có độ dài là 512 cho từng mẫu dữ liệu huấn luyện. Sau đó gán nhãn 1 đối với mẫu fresh và 0 đối với mẫu rotten

4 ÁP DỤNG THUẬT TOÁN VÀ KẾT QUẢ THỰC NGHIỆM

4.1 Áp dụng thuật toán SVM

Chúng em sử dụng SVM để xây dựng mô hình phân loại dự đoán đối tượng có phải là trái cây tươi hay hỏng. SVM xây dựng một siêu phẳng hoặc một tập hợp các siêu phẳng trong không gian đa chiều được sử dụng cho phân loại.

Mô hình SVM có một số siêu tham số như kernel, gamma, C không thể học trực tiếp từ dữ liệu.Sau quá trình tinh chỉnh tham số với tập validation, chúng em quyết định sử dụng C = 0.1, gamma= 1, kernel= ‘poly’.

Thông tin cài đặt huấn luyện mô hình SVM

- Dữ liệu: 7631 ảnh
- Phân chia: train: 0.7, val: 0.3
- Kernel: poly
- Gamma: 1
- C=0.1

Đánh giá trên tập test

Sau khi tạo ra mô hình, chúng em sẽ đánh giá mô hình dựa trên 4 loại độ đo: accuracy score, precision score, recall score và F1 score.

- Accuracy score: khả năng dự đoán đúng của mô hình trên tổng các mẫu trong tập testing.
- Precision score: thể hiện sự chuẩn xác của việc dự đoán đúng, tỷ lệ càng cao thì mô hình nhận các positive càng chuẩn
- Recall score: thể hiện khả năng phát hiện tất cả các positive, tỷ lệ càng cao thì khả năng bỏ sót các positive càng thấp.
- F1 score: là số dung hòa giữa recall score và precision score.

Classification report for classifier SVC(C=0.1, gamma=1, kernel='poly'):

	precision	recall	f1-score	support
0	0.95	0.92	0.93	1534
1	0.90	0.93	0.92	1164
accuracy			0.93	2698
macro avg	0.92	0.93	0.93	2698
weighted avg	0.93	0.93	0.93	2698

Figure 6: Kết quả huấn luyện mô hình SVM

4.2 Áp dụng thuật toán Random Forest

Chúng em sử dụng thuật toán Random Forest cho bài toán toán phân loại trái cây tươi hay hỏng. Random Forest sẽ tạo ra nhiều cây quyết định sau đó dùng kĩ thuật voting để đưa ra kết quả cuối cùng. Random Forest sẽ dùng trong thư viện sklearn với các tham số như sau

Thông tin cài đặt huấn luyện mô hình Random Forest

Dữ liệu: 7631 ảnh

Phân chia: train: 0.7, val: 0.3

n_estimators=50

max_depth=10

max_features='sqrt'

min_samples_split=10

min_samples_leaf=4

random_state=42

Đánh giá trên tập test

Sau khi tạo ra mô hình, chúng em sẽ đánh giá mô hình dựa trên 4 loại độ đo: accuracy score, precision score, recall score và F1 score.

Classification report for classifier RandomForestClassifier(max_depth=10, min_samples_leaf=4, min_samples_split=10, n_estimators=50, random_state=42):

	precision	recall	f1-score	support
0	0.98	0.96	0.97	1534
1	0.95	0.97	0.96	1164
accuracy			0.97	2698
macro avg	0.97	0.97	0.97	2698
weighted avg	0.97	0.97	0.97	2698

Figure 7: Kết quả huấn luyện mô hình Random Forest

4.3 Áp dụng thuật toán LDA

Tính toán vector HOG cho mỗi ảnh và sau đó chuẩn hóa các vector HOG này. Tiếp theo, áp dụng LDA để tìm không gian con của tập đặc trưng sao cho tối đa hóa sự phân tách giữa các lớp dữ liệu. Sau khi chiếu dữ liệu lên không gian con này, sử dụng siêu phẳng quyết định để phân loại dữ liệu dựa trên các giá trị chiếu thu được. Sau đó tiến hành dự đoán ảnh trong tập test.

Cấu hình của LDA sẽ được để mặc định như trên thư viện sklearn với solver = "svd" dùng để phân biệt tuyến tính giữa các lớp.

Thông tin cài đặt huấn luyện mô hình LDA

covariance_estimator: None,

n_components: None,

priors: None,

shrinkage: None,

solver: 'svd',

store_covariance: False,

tol: 0.0001

Đánh giá trên tập test

Sau khi tạo ra mô hình, chúng em sẽ đánh giá mô hình dựa trên 4 loại độ đo: accuracy score, precision score, recall score và F1 score.

Classification report for classifier LinearDiscriminantAnalysis():

	precision	recall	f1-score	support
0	0.93	0.93	0.93	1534
1	0.91	0.91	0.91	1164
accuracy			0.92	2698
macro avg	0.92	0.92	0.92	2698
weighted avg	0.92	0.92	0.92	2698

Figure 8: Kết quả huấn luyện mô hình LDA

4.4 Một số kết quả phân loại thực tế

+ **Linear Discriminant Analysis(LDA)**

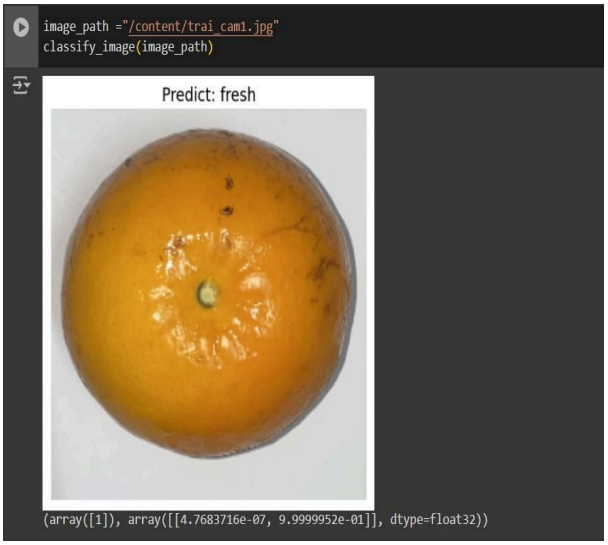

Figure 9: Kết quả dự đoán trái cam tươi

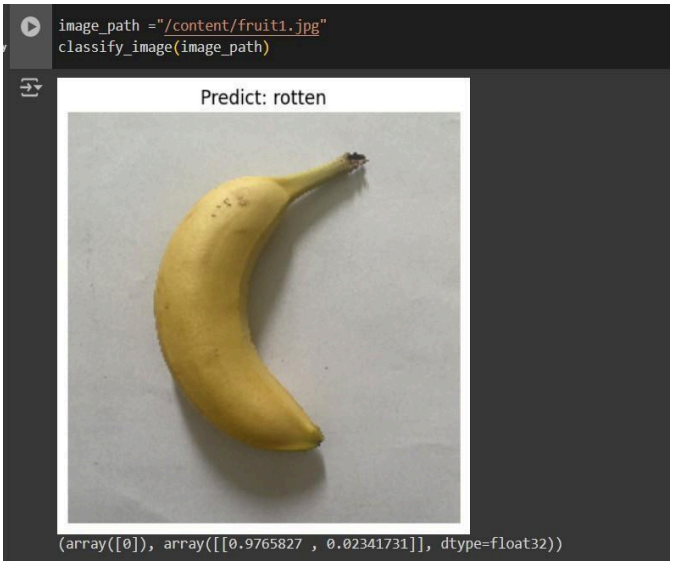

Figure 10: Kết quả dự đoán trái chuối hỏng

+ **Random Forest**

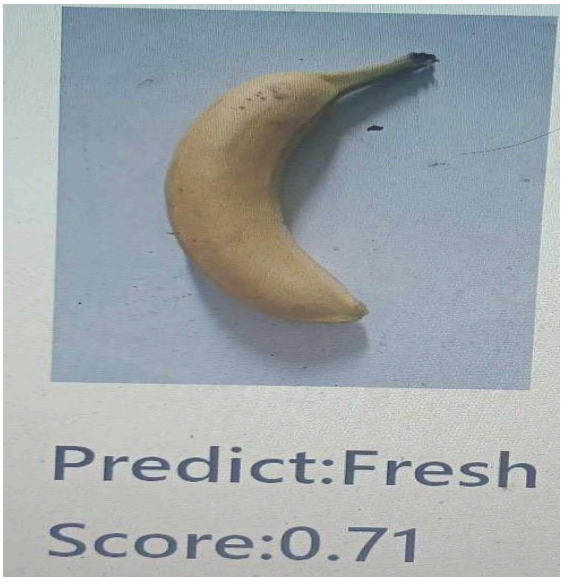

Figure 11: Kết quả dự đoán trái chuối tốt

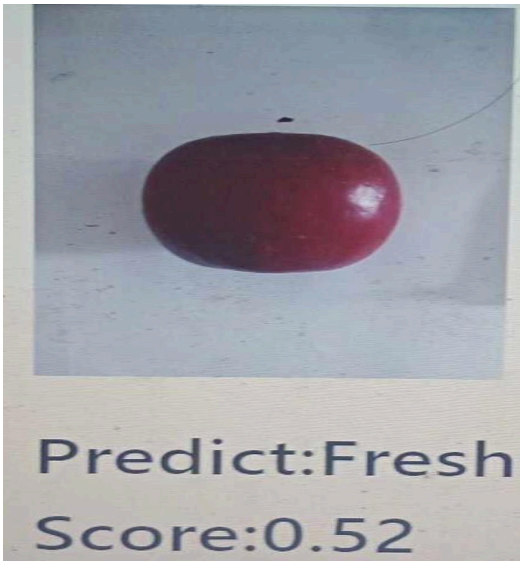

Figure 12: Kết quả dự đoán trái táo tốt

+ **Support Vector Machine (SVM):**

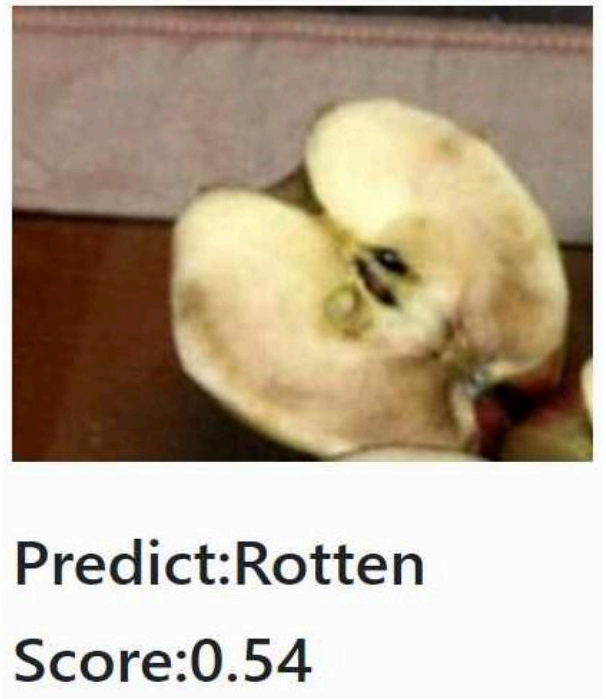

Figure 13: Kết quả dự đoán trái táo hỏng

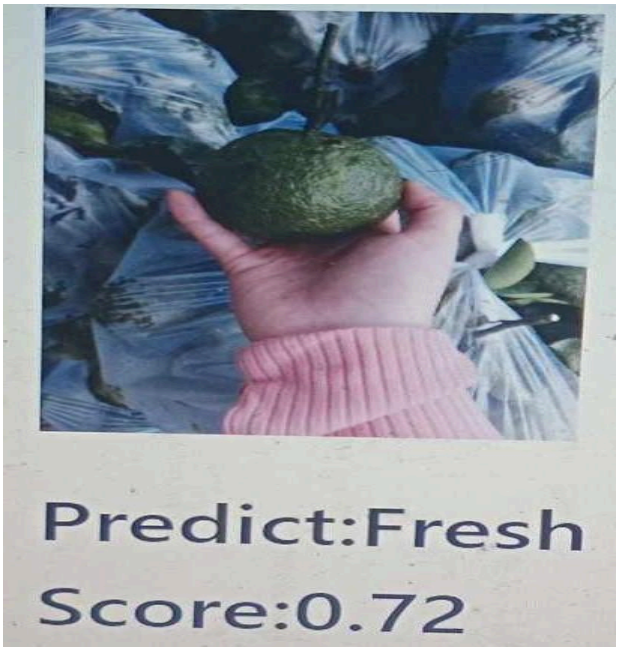

Figure 14: Kết quả dự đoán trái cam hỏng

5 TỔNG KẾT

Tổng kết quá trình tìm hiểu và thực nghiệm đối với bài toán phân loại trái cây tươi và hỏng, nhóm có đưa ra bảng so sánh và một số nhận xét đối với các mô hình sử dụng

Method	Accuracy	Precision	Recall	F1-score
SVM	0.927	0.902	0.932	0.917
Random Forest	**0.978**	**0.977**	**0.971**	**0.974**
LDA	0.92	0.92	0.92	0.92

Table 1: Kết quả đánh giá mô hình trên tập test

Nhận xét và đánh giá:

- Do sự phân phối màu sắc giữa ảnh thuộc các lớp tập train với tập test là khá giống nhau vì vậy sẽ cho ra các vector histogram màu có phân bố rất giống nhau nên sẽ cho ra kết quả như trên.
- Dựa vào bảng số liệu, ta có thể thấy được mô hình Random Forest có tỉ lệ chính xác (accuracy) cao nhất, đạt đến 97.8%, cũng như các chỉ số khác như độ chính xác (precision), tỷ lệ thu hồi (recall), và F1-score đều đạt được giá trị cao nhất so với các phương pháp SVM và LDA.
- Tóm lại, sự hiệu quả cao của Random Forest là do tính linh hoạt và khả năng làm việc tốt với các tập dữ liệu lớn và phức tạp. Phương pháp này sử dụng nhiều cây quyết định để tạo ra một mô hình mạnh mẽ hơn và giảm thiểu hiện tượng overfitting. Điều này giúp mô hình có khả năng áp dụng tốt cho dữ liệu mới và không cần điều chỉnh (tuning) nhiều. Trong khi đó, SVM và LDA thường phù hợp với các tập dữ liệu nhỏ hoặc có cấu trúc đơn giản hơn, với SVM hoạt động tốt trong không gian chiều cao và LDA hiệu quả khi các lớp có phân phối Gaussian và ma trận hiệp phương sai giống nhau.

Qua quá trình tìm hiểu, nhóm có đưa ra một số dự định cải tiến trong tương lai

- Triển khai web dự đoán trái cây tươi trên Flask
- Áp dụng các mô hình hiện đại của Deep Learning để so sánh
- Tạo thêm nhiều mẫu dữ liệu đa dạng hơn

6 THAM KHẢO

1. Vu T. Bài 34: Decision Trees (1): Iterative Dichotomiser 3. Tiep Vu’s blog.
2. GeeksforGeeks. Decision Tree. GeeksforGeeks.
3. What Is Random Forest? | IBM. Ibm.com.
4. Vu T. Bài 29: Linear Discriminant Analysis. Tiep Vu’s blog.
5. Wikipedia Contributors. Color histogram.