****

TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỒ HỒ CHÍ MINH

**KHOA CÔNG NGHỆ THÔNG TIN**

□&□

**BÁO CÁO KẾT THÚC HỌC PHẦN**

**MÁY HỌC**

**XÂY DỰNG MÔ HÌNH PHÂN LOẠI TRÁI CÂY**

Giảng viên hướng dẫn: **ThS. Huỳnh Thành Lộc**

Sinh viên thực hiện:

1. Nguyễn Quốc Minh 22DH112165
2. Lê phạm Hoàng Vũ 22DH114826
3. Trường Triều Dĩ 2DH110505

***Thành phố Hồ Chí Minh, tháng 11 năm 2024***

**A yellow oval with red text

Description automatically generated**

TRƯỜNG ĐẠI HỌC NGOẠI NGỮ - TIN HỌC THÀNH PHỒ HỒ CHÍ MINH

**KHOA CÔNG NGHỆ THÔNG TIN**

□&□

**BÁO CÁO KẾT THÚC HỌC PHẦN**

**MÁY HỌC**

**XÂY DỰNG MÔ HÌNH PHÂN LOẠI TRÁI CÂY**

Mã lớp học phần: **241123018401**

Năm học: **2024 – 2025**

Học kỳ: **1**

Sinh viên thực hiện:

1. Nguyễn Quốc Minh 22DH 112165
2. Lê phạm Hoàng Vũ 22DH114826
3. Trường Triều Dĩ 22DH110505

***Thành phố Hồ Chí Minh, tháng 11 năm 2024***

phụ lục

[DANH MỤC BẢNG 2](#_Toc182939991)

[CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI 3](#_Toc182939992)

[1.1. Giới thiệu bài toán 3](#_Toc182939993)

[1.2. Các công trình liên quan 4](#_Toc182939994)

[CHƯƠNG 2. CHUẨN BỊ DỮ LIỆU 6](#_Toc182939995)

[2.1. Giới thiệu tập dữ liệu 6](#_Toc182939996)

[2.2. Tiền xử lý và trực quan hóa dữ liệu 7](#_Toc182939997)

[2.2.1. **Tiền xử lý dữ liệu** 7](#_Toc182939998)

[2.2.2. **CNN** 7](#_Toc182939999)

[2.2.3. **MobileNetV2** 9](#_Toc182940000)

[2.2.4. **ResNet50** 11](#_Toc182940001)

[2.2.5. **KNN** 13](#_Toc182940002)

[2.3. Trích chọn đặc trưng 15](#_Toc182940003)

[2.3.1. **Trích xuất đặc trưng cạch và màu của các loại quả** 15](#_Toc182940004)

[CHƯƠNG 3. XÂY DỰNG MÔ HÌNH 20](#_Toc182940005)

[3.1. CNN 20](#_Toc182940006)

[3.1.1. **Thư viện** 20](#_Toc182940007)

[3.1.2. **Mô tả thuật toán** 20](#_Toc182940008)

[3.2. MobileNetV2 22](#_Toc182940009)

[3.2.1. **Thư viện** 22](#_Toc182940010)

[3.2.2. **Mô tả thuật toán** 22](#_Toc182940011)

[3.3. Resnet50 23](#_Toc182940012)

[3.3.1. **Thư viện** 23](#_Toc182940013)

[3.3.2. **Mô tả thuật toán** 23](#_Toc182940014)

[3.4. KNN 25](#_Toc182940015)

[3.4.1. **Thư viện** 25](#_Toc182940016)

[3.4.2. **Mô tả thuật toán** 25](#_Toc182940017)

[3.5. Thước đo đánh giá mô hình 27](#_Toc182940018)

[3.5.1. **Đánh giá mô hình CNN** 27](#_Toc182940019)

[3.5.2. **Đánh giá mô hình MobileNetV2** 28](#_Toc182940020)

[3.5.3. **Đánh giá mô hình ResNet50** 30](#_Toc182940021)

[3.5.4. **Đánh giá mô hình KNN** 31](#_Toc182940022)

[3.5.5. Biểu đồ đánh giá giữa các thuật toán trên 32](#_Toc182940023)

[CHƯƠNG 4. KẾT QUẢ THỰC NGHIỆM 33](#_Toc182940024)

[4.1. Kết Quả các thuật toán 33](#_Toc182940025)

[4.1.1. **CNN** 33](#_Toc182940026)

[4.1.2. **MobileNetV2** 34](#_Toc182940027)

[4.1.3. **RetNet50** 35](#_Toc182940028)

[4.1.4. **KNN** 35](#_Toc182940029)

[4.2. So sánh 36](#_Toc182940030)

[4.2.1. **CNN** 36](#_Toc182940031)

[4.2.2. **MobileNetV2** 36](#_Toc182940032)

[4.2.3. **RetNet50** 37](#_Toc182940033)

[4.2.4. **KNN** 37](#_Toc182940034)

[4.2.5. **Kết luận** 38](#_Toc182940035)

[CHƯƠNG 5. KẾT LUẬN 39](#_Toc182940036)

[5.1. Kết quả đạt được 39](#_Toc182940037)

[5.1.1. **So sánh hiệu suất các mô hình:** 39](#_Toc182940038)

[5.1.2. **Đánh giá và rút ra bài học:** 39](#_Toc182940039)

[5.1.3. **Ứng dụng tiềm năng:** 39](#_Toc182940040)

[5.2. Những khó khăn, hạn chế 40](#_Toc182940041)

[5.2.1. **Khó Khăn** 40](#_Toc182940042)

[5.2.2. **Hạn Chế** 40](#_Toc182940043)

[TÀI LIỆU THAM KHẢO 41](#_Toc182940044)

[PHỤ LỤC 41](#_Toc182940045)

danh mục hình

[Hình 0: Tiền xử lí dữ liệu 1 7](#_Toc182938378)

[Hình 1 : Biểu đồ độ chính xác (CNN) 1 7](#_Toc182938379)

[Hình 2: Biểu đồ độ mất mát ( CNN) 1 8](#_Toc182938380)

[Hình 3:Biểu đồ độ chính xác(MobileNetV2) 1 9](#_Toc182938381)

[Hình 4:Biểu đồ độ mất mát(MobileNetV2) 1 10](#_Toc182938382)

[Hình 5: Biểu đồ độ chính xác( ResNet50 ) 1 11](#_Toc182938383)

[Hình 6: Biểu đồ độ mất mát ( ResNet50 ) 1 12](#_Toc182938384)

[Hình 7: Biểu đồ độ chính xác( KNN ) 1 13](#_Toc182938385)

[Hình 8: Biểu đồ độ mất mát ( KNN ) 1 14](#_Toc182938386)

[Hình 9 : Đặc trưng cạnh và màu của cà chua 1 15](#_Toc182938387)

[Hình 10: Đặc trưng cạnh và màu của cam 1 15](#_Toc182938388)

[Hình 11: Đặc trưng cạnh và màu của chuối 1 16](#_Toc182938389)

[Hình 12: Đặc trưng cạnh và màu của dưa hấu 1 16](#_Toc182938390)

[Hình 13: Đặc trưng cạnh và màu của lê 1 17](#_Toc182938391)

[Hình 14: Đặc trưng cạnh và màu của nho 1 17](#_Toc182938392)

[Hình 15: Đặc trưng cạnh và màu của quýt 1 18](#_Toc182938393)

[Hình 16: Đặc trưng cạnh và màu của táo 1 18](#_Toc182938394)

[Hình 17: Đặc trưng cạnh và màu của thơm 1 19](#_Toc182938395)

[Hình 18: Đặc trưng cạnh và màu của xoài 1 19](#_Toc182938396)

[Hình 19: Ma trận đánh giá của CNN 1 28](#_Toc182938397)

[Hình 20: Ma trận đánh giá MobileNetV2 1 29](#_Toc182938398)

[Hình 21: Ma trận đánh giá ResNet50 1 30](#_Toc182938399)

[Hình 22: Ma trận đánh giá KNN 1 31](#_Toc182938400)

[Hình 23: Mô hình đánh giá giữa các thuật toán 1 32](#_Toc182938401)

[Hình 24:Kết quả dự đoán của mô hình CNN 1 33](#_Toc182938402)

[Hình 25: Kết quả dự đoán của mô hình MobileNetV2 1 34](#_Toc182938403)

[Hình 26: Kết quả dự đoán của mô hình ResNet50 1 35](#_Toc182938404)

[Hình 27: Kết quả dự đoán của mô hình KNN 1 35](#_Toc182938405)

# DANH MỤC BẢNG

[Bảng 1: Các mô hình phổ biến 1 5](#_Toc182936111)

# GIỚI THIỆU ĐỀ TÀI

## Giới thiệu bài toán

Giới thiệu nhiệm vụ (Task) của đề tài.

Phân loại trái cây tự động là một hướng nghiên cứu đầy tiềm năng, đặc biệt hữu ích trong bối cảnh chuỗi cung ứng thực phẩm ngày càng mở rộng và yêu cầu tính hiệu quả cao. Với khả năng nhận diện chính xác các loại trái cây mà không cần sự can thiệp của con người, công nghệ này có thể cải thiện quy trình giám sát và quản lý hàng hóa, giúp việc sắp xếp và đóng gói diễn ra nhanh chóng hơn và giảm thiểu sai sót. Điều này đặc biệt có giá trị trong các siêu thị và cửa hàng tạp hóa, nơi khối lượng lớn trái cây cần được phân loại liên tục và chính xác.

Một ứng dụng quan trọng của hệ thống phân loại tự động là hỗ trợ thanh toán không cần quét mã vạch, giúp tiết kiệm thời gian cho cả người mua và nhân viên thu ngân. Thay vì dựa vào việc quét mã truyền thống, công nghệ nhận diện hình ảnh có thể xác định loại và số lượng trái cây trên quầy thanh toán thông qua hệ thống camera tích hợp.

Trong nghiên cứu này, nhóm chỉ tập trung vào 10 loại trái cây được tiêu thụ nhiều nhất trên thế giới: cà chua, chuối, dưa hấu, táo, nho, cam, xoài, quýt, lê, và dứa. Đây là những loại trái cây phổ biến trong các cửa hàng và siêu thị lớn, do đó việc nghiên cứu giải pháp phân loại chúng sẽ có tính ứng dụng thực tế cao. Những thách thức của bài toán bao gồm việc xử lý các biến đổi về ánh sáng, hình dạng không đồng đều, và màu sắc tương tự giữa một số loại trái cây, tất cả đều là những yếu tố cần được xem xét kỹ lưỡng để đảm bảo độ chính xác cao trong thực tế.

Việc sử dụng các thuật toán học máy, đặc biệt là những mô hình như mạng nơ-ron tích chập (CNN), hứa hẹn sẽ mang lại hiệu quả vượt trội nhờ khả năng trích xuất đặc trưng mạnh mẽ từ dữ liệu hình ảnh. Các công nghệ bổ sung như thị giác máy tính và học sâu cũng sẽ góp phần tối ưu hóa hiệu quả của hệ thống, giảm thời gian xử lý và cải thiện khả năng phân loại tự động trong chuỗi cung ứng.

## Các công trình liên quan

Giới thiệu các mô hình (thuật toán) phổ biến thường được sử dụng để giải quyết bài toán của đề tài.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên mô hình | Nguyên lý hoạt động | Ưu điểm | Nhược điểm |
| 1 | K-Nearest Neighbors (KNN) | Dựa trên khoảng cách để phân loại dựa vào nhãn của các điểm gần nhất trong tập dữ liệu. | Đơn giản, dễ hiểu, dễ triển khai. | |  | | --- | | Tốc độ xử lý chậm khi dữ liệu lớn; nhạy cảm với dữ liệu nhiễu. | |
| 2 | Support Vector Machine (SVM) | |  | | --- | | Tìm siêu phẳng tối ưu phân tách các lớp dữ liệu với khoảng cách biên lớn nhất. | |  | | |  | | --- | | Hiệu quả với các bài toán phân loại phi tuyến; linh hoạt. | | Cần chọn kernel phù hợp, thời gian tính toán lâu với dữ liệu lớn. |
| 3 | Decision Trees | Sử dụng cấu trúc cây để phân chia dữ liệu dựa vào các đặc trưng và ngưỡng quyết định. | Dễ hiểu, dễ giải thích, hiệu quả với dữ liệu nhiều chiều. | Dễ bị overfitting nếu không cắt tỉa cây đúng cách |
| 4 | Random Forests | Kết hợp nhiều cây quyết định để đưa ra dự đoán chính xác hơn (ensemble learning). | Giảm thiểu overfitting, độ chính xác cao. | Thời gian tính toán lâu khi số lượng cây lớn |
| 5 | Logistic Regression | Dự đoán xác suất một mẫu thuộc về một lớp dựa trên hàm sigmoid áp dụng lên hồi quy tuyến tính. | Đơn giản, hiệu quả với các bài toán phân loại tuyến tính. | Không hiệu quả với các bài toán phân loại phi tuyến phức tạp. |
| 6 | Neural Networks | Mạng lưới các nơ-ron có các lớp ẩn để học và phân loại dữ liệu thông qua quá trình tối ưu hóa. | Có khả năng học các mẫu phức tạp và phi tuyến. | Cần nhiều dữ liệu và tài nguyên tính toán; khó tinh chỉnh. |
| 7 | Convolutional Neural Networks (CNN) | Trích xuất đặc trưng và phân loại dữ liệu (đặc biệt là hình ảnh) thông qua các lớp tích chập. | Hiệu quả cao với bài toán nhận diện và phân loại hình ảnh. | Cần nhiều dữ liệu và tài nguyên tính toán; phức tạp để triển khai |
| 8 | Gradient Boosting | Xây dựng các mô hình cây quyết định tuần tự, mỗi cây mới cải thiện lỗi của cây trước đó. | Hiệu quả cao, phổ biến trong các cuộc thi học máy. | Dễ bị overfitting nếu không điều chỉnh siêu tham số đúng cách. |
| 9 | Naive Bayes | Dựa trên định lý Bayes với giả định các đặc trưng là độc lập. | Đơn giản, nhanh, hoạt động tốt với dữ liệu nhỏ và khi giả định độc lập gần đúng. | Giả định độc lập giữa các đặc trưng không thực tế, làm giảm độ chính xác. |

Bảng 1: Các mô hình phổ biến

# CHUẨN BỊ DỮ LIỆU

## Giới thiệu tập dữ liệu

Tập dữ liệu phân loại trái cây này được xây dựng để phục vụ cho các bài toán phân loại hình ảnh trong học máy. Tập dữ liệu bao gồm nhiều ảnh của các loại trái cây phổ biến, với mỗi loại được lưu trữ trong một thư mục riêng biệt. Mỗi ảnh biểu thị hình ảnh của một loại trái cây duy nhất, được chụp từ các góc độ khác nhau để tăng tính đa dạng và thực tế của tập dữ liệu.

**Các thông tin chi tiết về tập dữ liệu:**

* **Số loại trái cây**: Tập dữ liệu bao gồm các loại trái cây như chuối, táo, cam, nho, dưa hấu, xoài, quýt, lê, và thơm. Mỗi loại trái cây được coi là một lớp trong bài toán phân loại.
* **Số lượng ảnh**: Mỗi thư mục chứa hàng chục ảnh của một loại trái cây, tổng số ảnh của cả tập dữ liệu có thể lên tới hàng trăm hoặc hàng nghìn ảnh.
* **Định dạng ảnh**: Ảnh thường có định dạng .jpg hoặc .png, và kích thước của ảnh có thể khác nhau. Trước khi đưa vào mô hình, ảnh sẽ được tiền xử lý và resize về kích thước cố định (ví dụ: 64x64 hoặc 100x100).
* **Mục tiêu**: Mục tiêu của tập dữ liệu là huấn luyện một mô hình học máy có thể phân loại chính xác loại trái cây trong ảnh. Đây là một ví dụ của bài toán phân loại hình ảnh đa lớp (multi-class classification).

**Ứng dụng của tập dữ liệu:** Tập dữ liệu phân loại trái cây có thể được sử dụng trong các hệ thống nhận diện và quản lý thực phẩm, các ứng dụng bán lẻ tự động, hoặc trong nghiên cứu về thị giác máy tính (computer vision). Đây là một tập dữ liệu giúp người học và các nhà nghiên cứu thực hành các mô hình học sâu như CNN, ResNet, MobileNet, hoặc các thuật toán cơ bản như Naive Bayes kết hợp PCA, đồng thời tìm hiểu các kỹ thuật tiền xử lý ảnh và tối ưu hóa mô hình trong lĩnh vực phân loại hình ảnh.

## Tiền xử lý và trực quan hóa dữ liệu

### **Tiền xử lý dữ liệu**

A screenshot of a computer program

Description automatically generated

Hình 0: Tiền xử lí dữ liệu

### **CNN**

A graph with blue and orange lines

Description automatically generated

Hình 1 : Biểu đồ độ chính xác (CNN)

A graph of a graph with blue lines and orange lines

Description automatically generated

Hình 2: Biểu đồ độ mất mát ( CNN)

### **MobileNetV2**

A graph of blue and orange lines

Description automatically generated

Hình 3:Biểu đồ độ chính xác(MobileNetV2)

A graph of loss and loss

Description automatically generated

Hình 4:Biểu đồ độ mất mát(MobileNetV2)

‘

### **ResNet50**

A graph of a line graph

Description automatically generated with medium confidence

Hình 5: Biểu đồ độ chính xác( ResNet50 )

A graph with orange lines and blue lines

Description automatically generated

Hình 6: Biểu đồ độ mất mát ( ResNet50 )

### **KNN**

A graph with a line going up

Description automatically generated

Hình 7: Biểu đồ độ chính xác( KNN )

A graph with red lines

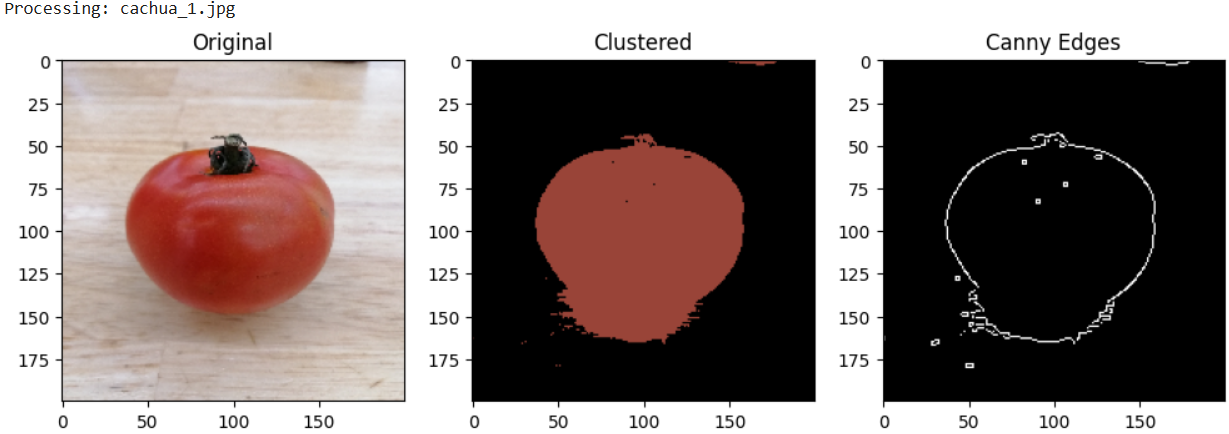
Description automatically generated

Hình 8: Biểu đồ độ mất mát ( KNN )

## Trích chọn đặc trưng

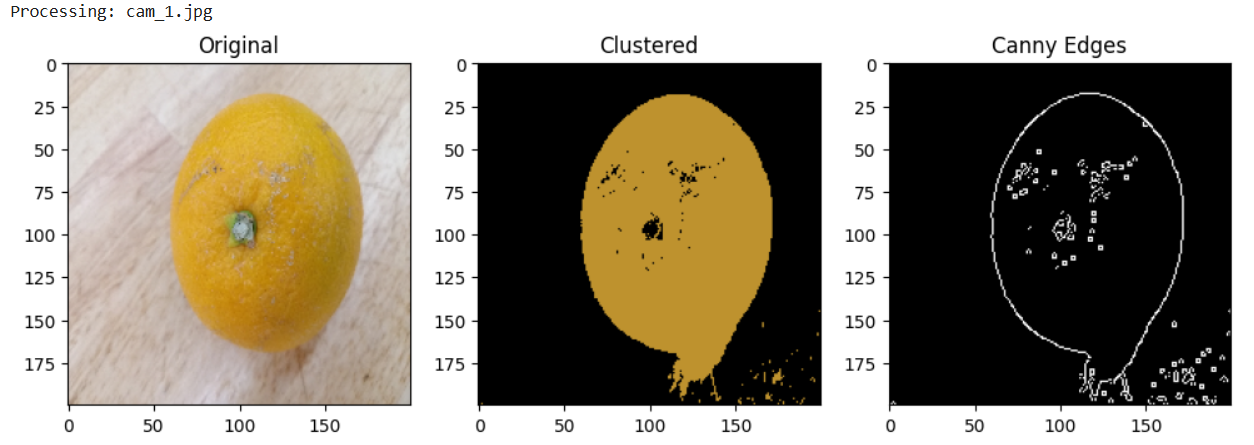
### **Trích xuất đặc trưng cạch và màu của các loại quả**

#### 2.3.1.1 Cà chua



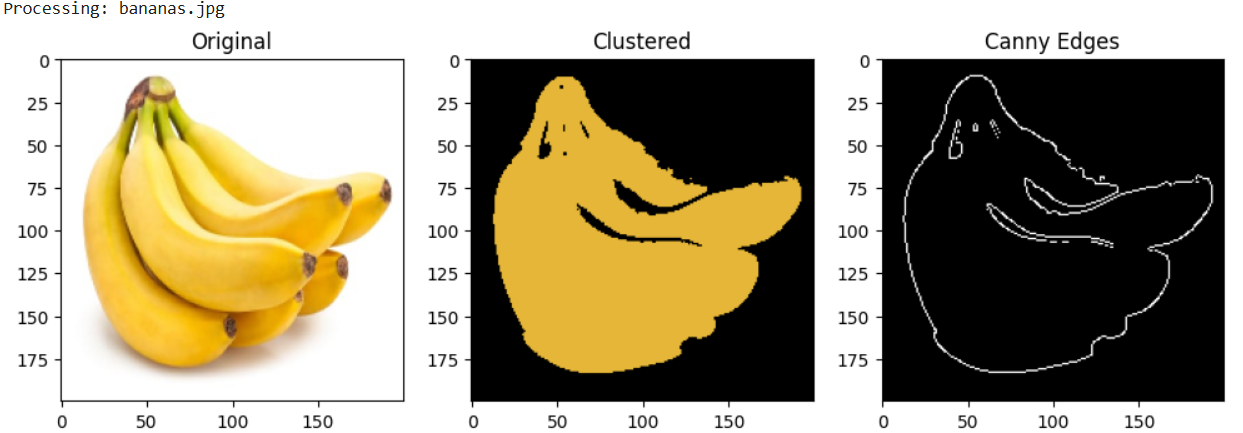
Hình 9 : Đặc trưng cạnh và màu của cà chua

#### 2.3.1.2 Cam



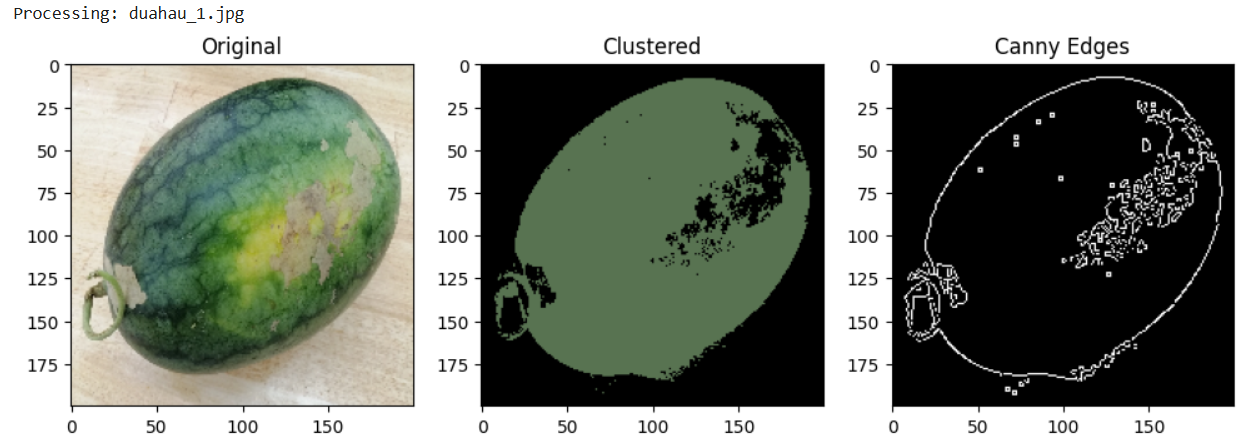
Hình 10: Đặc trưng cạnh và màu của cam

#### 2.3.1.3 Chuối



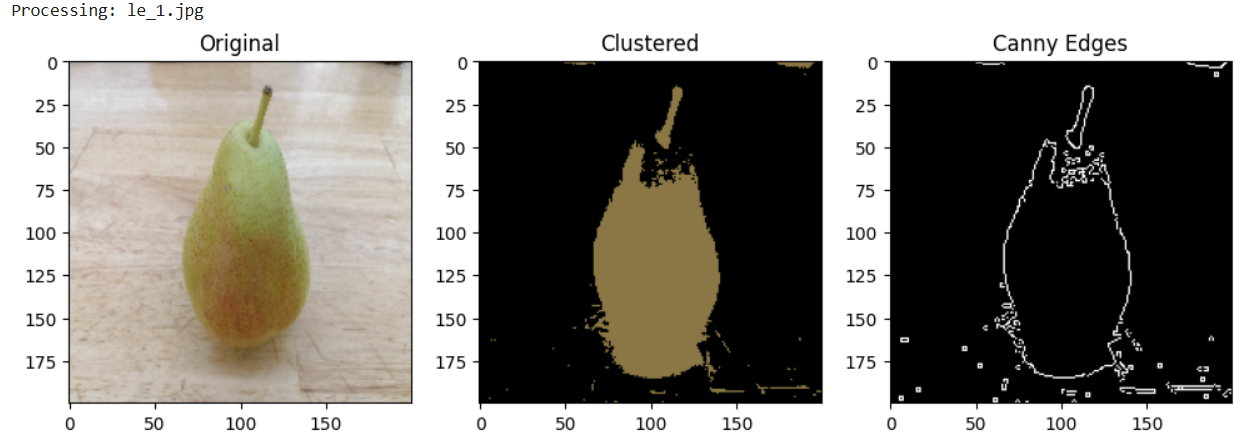
Hình 11: Đặc trưng cạnh và màu của chuối

#### 2.3.1.4 Dưa hấu



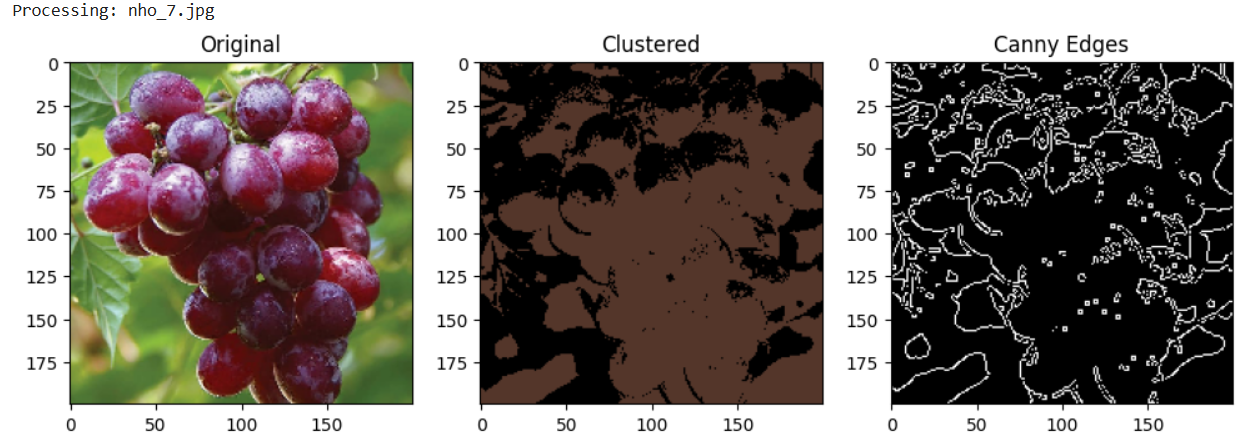
Hình 12: Đặc trưng cạnh và màu của dưa hấu

#### 2.3.1.5 Lê



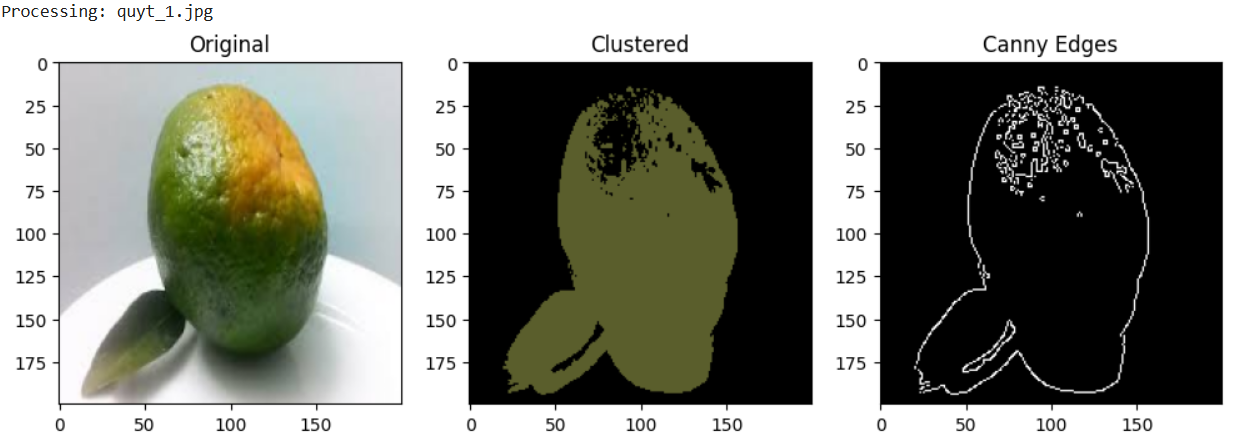
Hình 13: Đặc trưng cạnh và màu của lê

#### 2.3.1.6 Nho



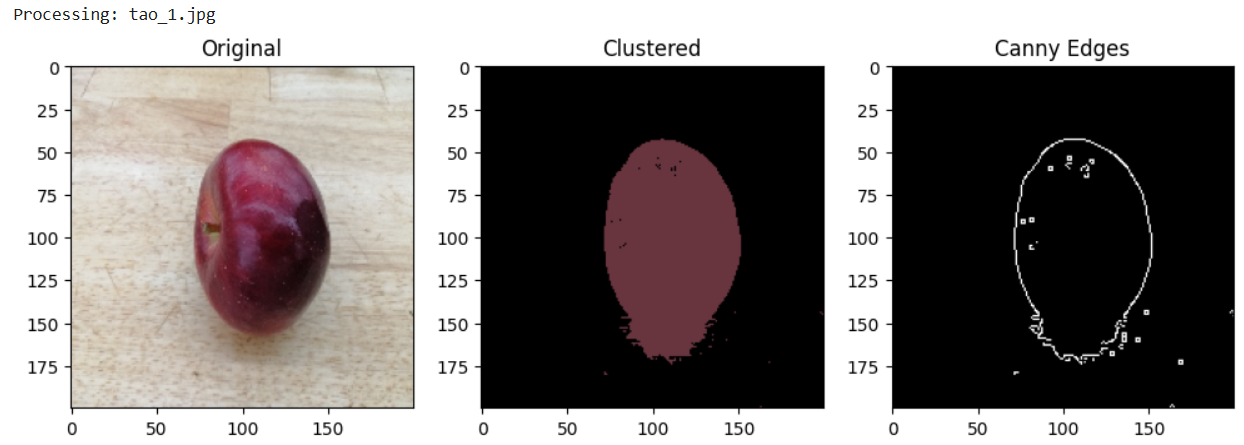
Hình 14: Đặc trưng cạnh và màu của nho

#### 2.3.1.7 Quýt



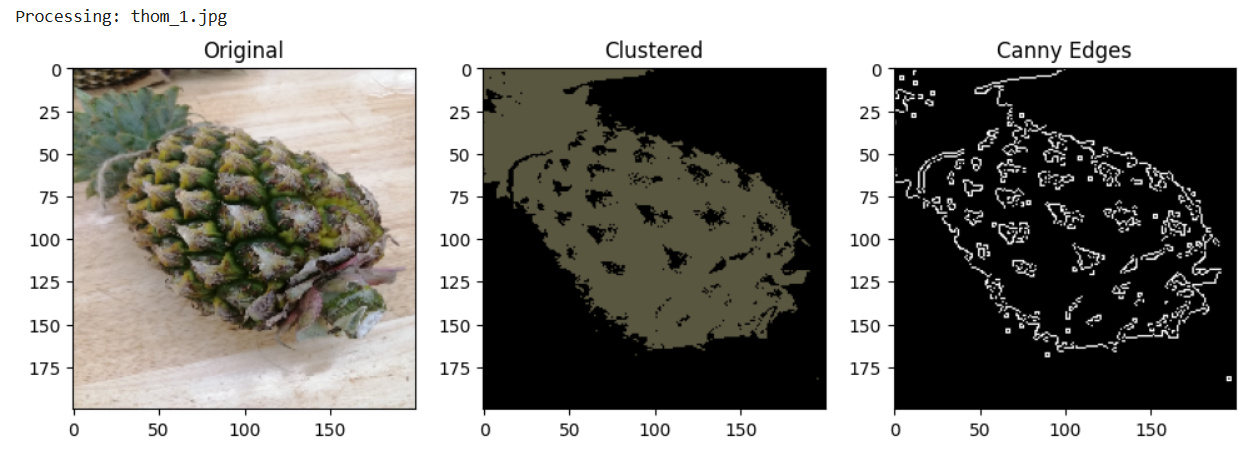
Hình 15: Đặc trưng cạnh và màu của quýt

#### 2.3.1.8 Táo



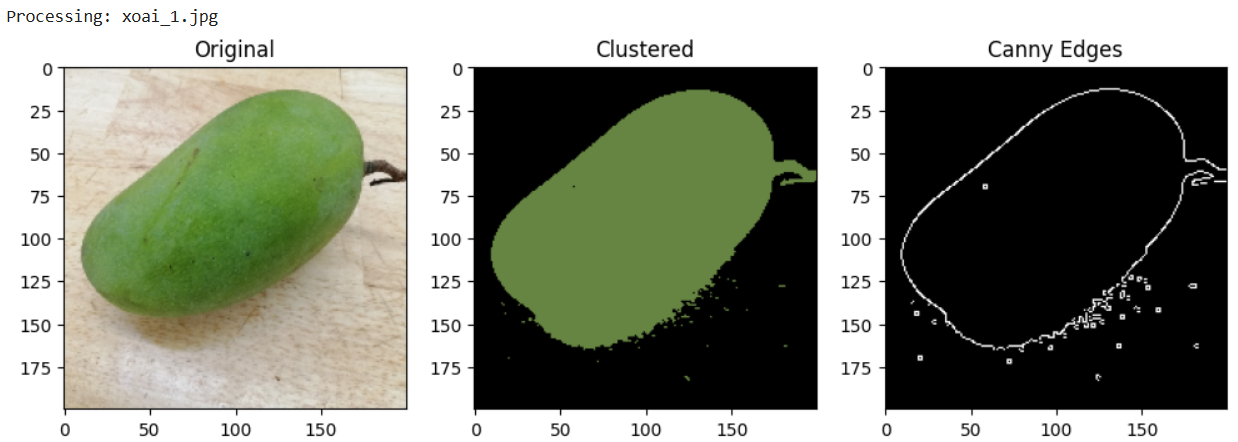
Hình 16: Đặc trưng cạnh và màu của táo

#### 2.3.1.9 Thơm



Hình 17: Đặc trưng cạnh và màu của thơm

#### 2.3.1.10 Xoài



Hình 18: Đặc trưng cạnh và màu của xoài

# XÂY DỰNG MÔ HÌNH

(Tùy thuộc vào hướng triển khai của đề tài mà nhóm trình bày chi tiết việc cài đặt các thuật toán và các thư viện được sử dụng)

## CNN

### **Thư viện**

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import LabelEncoder

from tensorflow.keras.utils import to\_categorical

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout, BatchNormalization, GlobalAveragePooling2D, Add, Activation,Input

from tensorflow.keras.preprocessing.image import ImageDataGenerator

### **Mô tả thuật toán**

# Xây dựng kiến trúc mô hình

# Xây dựng kiến trúc mô hình cải tiến

model = Sequential([

Conv2D(32, (3, 3), activation='relu', input\_shape=(64, 64, 3)),

BatchNormalization(),

MaxPooling2D(pool\_size=(2, 2)),

Conv2D(64, (3, 3), activation='relu'),

BatchNormalization(),

MaxPooling2D(pool\_size=(2, 2)),

Conv2D(128, (3, 3), activation='relu'),

BatchNormalization(),

MaxPooling2D(pool\_size=(2, 2)),

Conv2D(256, (3, 3), activation='relu'),

BatchNormalization(),

MaxPooling2D(pool\_size=(2, 2)),

Flatten(),

Dense(512, activation='relu'),

Dropout(0.5),

Dense(len(categories), activation='softmax') # số lớp đầu ra bằng số loại trái cây

])

# Compile mô hình

model.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

model.summary()

# Cấu hình data augmentation

datagen = ImageDataGenerator(

rotation\_range=30,

width\_shift\_range=0.2,

height\_shift\_range=0.2,

shear\_range=0.2,

zoom\_range=0.2,

horizontal\_flip=True,

fill\_mode="nearest"

)

# Huấn luyện mô hình với data augmentation

batch\_size = 32

history = model.fit(datagen.flow(X\_train, y\_train, batch\_size=batch\_size),

validation\_data=(X\_test, y\_test),

steps\_per\_e

# Lấy ngẫu nhiên một ảnh trong tập kiểm tra

idx = random.randint(0, len(X\_test) - 1)

sample\_image = X\_test[idx]

sample\_label = y\_test[idx]

# Dự đoán

prediction = model.predict(np.expand\_dims(sample\_image, axis=0))

predicted\_label = le.inverse\_transform([np.argmax(prediction)])

true\_label = le.inverse\_transform([np.argmax(sample\_label)])

# Hiển thị kết quả dự đoán

plt.imshow(sample\_image)

plt.title(f"Thực tế: {true\_label[0]}, Dự đoán: {predicted\_label[0]}")

plt.axis('off')

plt.show()

## MobileNetV2

### **Thư viện**

from tensorflow.keras.applications import MobileNetV2

from tensorflow.keras.models import Model

### **Mô tả thuật toán**

x = base\_model.output

x = Flatten()(x)

x = Dense(128, activation='relu')(x)

x = Dropout(0.5)(x)

output = Dense(len(categories), activation='softmax')(x)

model\_mobilenet = Model(inputs=base\_model.input, outputs=output)

# Compile mô hình

model\_mobilenet.compile(optimizer='adam', loss='categorical\_crossentropy', metrics=['accuracy'])

model\_mobilenet.summary()

# huấn luyện mô hình MobileNetV2

history\_mobilenet = model\_mobilenet.fit(

datagen.flow(X\_train, y\_train, batch\_size=batch\_size),

validation\_data=(X\_test, y\_test),

steps\_per\_epoch=len(X\_train) // batch\_size,

epochs=50

)

def predict\_image(img, model, categories):

# Resize ảnh về kích thước (64, 64)

img\_resized = cv2.resize(img, (64, 64))

# Chuyển ảnh từ BGR sang RGB và chuẩn hóa về phạm vi [0, 1]

img\_normalized = cv2.cvtColor(img\_resized, cv2.COLOR\_BGR2RGB) / 255.0

# Thêm chiều batch để phù hợp với input của mô hình

img\_input = np.expand\_dims(img\_normalized, axis=0)

# Dự đoán với mô hình

prediction = model.predict(img\_input)

# Lấy chỉ số của lớp có xác suất cao nhất

predicted\_class = np.argmax(prediction, axis=1)[0]

# Lấy tên loại trái cây dựa trên chỉ số

predicted\_label = categories[predicted\_class]

return predicted\_label, img\_normalized

# Đường dẫn đến ảnh bạn muốn dự đoán

test\_image\_path = '/content/drive/MyDrive/TraiCay/chuoi/banana.jpg'

test\_img = cv2.imread(test\_image\_path)

# Dự đoán và lấy nhãn, ảnh chuẩn hóa

predicted\_label, img\_to\_show = predict\_image(test\_img, model\_mobilenet, categories)

# Hiển thị ảnh với nhãn dự đoán

plt.imshow(img\_to\_show)

plt.title(f"Dự đoán: {predicted\_label}")

plt.axis('off')

plt.show()

## Resnet50

### **Thư viện**

from tensorflow.keras.applications import ResNet50

from tensorflow.keras.optimizers import Adam

### **Mô tả thuật toán**

def residual\_block(x, filters, kernel\_size=3, stride=1):

# Lưu lại đầu vào ban đầu để dùng trong phép cộng

shortcut = x

# Lớp tích chập đầu tiên

x = Conv2D(filters, kernel\_size, strides=stride, padding='same')(x)

x = BatchNormalization()(x)

x = Activation('relu')(x)

# Lớp tích chập thứ hai

x = Conv2D(filters, kernel\_size, strides=1, padding='same')(x)

x = BatchNormalization()(x)

# Nếu cần thiết, thay đổi kích thước của shortcut để phù hợp với output

if stride != 1:

shortcut = Conv2D(filters, kernel\_size=1, strides=stride, padding='same')(shortcut)

shortcut = BatchNormalization()(shortcut)

# Cộng shortcut và đầu ra của lớp tích chập rồi áp dụng hàm kích hoạt ReLU

x = Add()([x, shortcut])

x = Activation('relu')(x)

return x

# Khởi tạo đầu vào

input\_layer = Input(shape=(64, 64, 3))

# Thêm các lớp Conv2D và Residual Block

x = Conv2D(64, kernel\_size=3, strides=1, padding='same', activation='relu')(input\_layer)

x = MaxPooling2D(pool\_size=2)(x)

# Thêm các Residual Blocks

x = residual\_block(x, 64)

x = residual\_block(x, 64)

x = residual\_block(x, 128, stride=2) # Giảm kích thước với stride=2

x = residual\_block(x, 128)

# Global Average Pooling và lớp Fully Connected cuối cùng

x = GlobalAveragePooling2D()(x)

output\_layer = Dense(len(categories), activation='softmax')(x)

# Tạo mô hình

model\_resnet\_custom = Model(inputs=input\_layer, outputs=output\_layer)

# Compile mô hình

model\_resnet\_custom.compile(optimizer=Adam(), loss='categorical\_crossentropy', metrics=['accuracy'])

# Huấn luyện mô hình

history\_resnet\_custom = model\_resnet\_custom.fit(X\_train, y\_train, epochs=10, validation\_data=(X\_test, y\_test))

# Chọn ngẫu nhiên 9 ảnh từ tập kiểm tra để hiển thị

num\_images = 9

indices = np.random.choice(range(len(X\_test)), num\_images, replace=False)

plt.figure(figsize=(12, 12))

for i, idx in enumerate(indices):

# Lấy ảnh từ tập kiểm tra

image = X\_test[idx]

true\_label = categories[y\_test[idx].argmax()] # Nhãn thực tế

# Dự đoán nhãn của ảnh

pred\_label\_idx = model\_resnet\_custom.predict(image.reshape(1, 64, 64, 3)).argmax()

pred\_label = categories[pred\_label\_idx]

# Hiển thị ảnh với nhãn dự đoán và nhãn thực tế

plt.subplot(3, 3, i + 1)

plt.imshow(image)

plt.title(f"Thực tế: {true\_label}\nDự đoán: {pred\_label}")

plt.axis('off')

plt.suptitle("Dự đoán trái cây bằng ResNet Custom Model", fontsize=16)

plt.show()

## KNN

### **Thư viện**

from sklearn.decomposition import PCA

from sklearn.neighbors import KNeighborsClassifier

from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix

import seaborn as sns

### **Mô tả thuật toán**

# Chuyển đổi dữ liệu huấn luyện và kiểm tra thành dạng phẳng

flat\_train\_data = X\_train.reshape(len(X\_train), -1)

flat\_test\_data = X\_test.reshape(len(X\_test), -1)

# Giảm chiều với PCA để tăng hiệu quả tính toán

pca = PCA(n\_components=100) # Sử dụng 100 thành phần chính

pca\_train\_data = pca.fit\_transform(flat\_train\_data)

pca\_test\_data = pca.transform(flat\_test\_data)

# Khởi tạo và huấn luyện mô hình KNN

k = 5 # Chọn số lượng hàng xóm K (có thể thử nghiệm với các giá trị khác)

knn\_model = KNeighborsClassifier(n\_neighbors=k)

knn\_model.fit(pca\_train\_data, y\_train.argmax(axis=1))

# Dự đoán trên tập kiểm tra

knn\_predictions = knn\_model.predict(pca\_test\_data)

# Đánh giá hiệu suất

accuracy = accuracy\_score(y\_test.argmax(axis=1), knn\_predictions)

print(f"Accuracy: {accuracy \* 100:.2f}%")

# Báo cáo chi tiết

print("Classification Report:")

print(classification\_report(y\_test.argmax(axis=1), knn\_predictions, target\_names=categories))

# Ma trận nhầm lẫn

conf\_matrix = confusion\_matrix(y\_test.argmax(axis=1), knn\_predictions)

plt.figure(figsize=(10, 8))

sns.heatmap(conf\_matrix, annot=True, fmt="d", cmap="Blues", xticklabels=categories, yticklabels=categories)

plt.xlabel("Predicted Label")

plt.ylabel("True Label")

plt.title("Confusion Matrix - KNN")

plt.show()

# Chọn ngẫu nhiên ảnh từ tập kiểm tra để hiển thị

num\_images = 10

indices = np.random.choice(range(len(X\_test)), num\_images, replace=False)

plt.figure(figsize=(12, 12))

for i, idx in enumerate(indices):

# Dự đoán nhãn cho ảnh đã chọn

image = X\_test[idx]

true\_label = categories[y\_test[idx].argmax()] # Nhãn thực tế

pca\_image = pca.transform(image.reshape(1, -1)) # Áp dụng PCA lên ảnh

pred\_label = categories[knn\_model.predict(pca\_image)[0]] # Nhãn dự đoán

# Hiển thị ảnh

plt.subplot(4, 3, i + 1)

plt.imshow(image)

plt.title(f"Thực tế: {true\_label}\nDự đoán: {pred\_label}")

plt.axis('off')

plt.suptitle("Dự đoán trái cây bằng KNN", fontsize=16)

plt.show()

## Thước đo đánh giá mô hình

### **Đánh giá mô hình CNN**

Accuracy: 98.02%

Độ mất mát (Loss): 0.0198

Classification Report:

precision recall f1-score support

cachua 0.95 1.00 0.97 52

cam 1.00 1.00 1.00 60

chuoi 0.50 0.67 0.57 3

duahau 1.00 1.00 1.00 55

le 0.98 0.98 0.98 43

nho 0.96 1.00 0.98 55

quyt 1.00 1.00 1.00 52

tao 0.97 0.93 0.95 40

thom 1.00 0.96 0.98 52

xoai 1.00 0.95 0.98 42

accuracy 0.98 454

macro avg 0.94 0.95 0.94 454

weighted avg 0.98 0.98 0.98 454

A screenshot of a graph

Description automatically generated

Hình 19: Ma trận đánh giá của CNN

### **Đánh giá mô hình MobileNetV2**

Accuracy: 98.46%

Độ mất mát (Loss): 0.0154

Classification Report:

precision recall f1-score support

cachua 0.95 1.00 0.97 52

cam 1.00 1.00 1.00 60

chuoi 0.67 0.67 0.67 3

duahau 1.00 1.00 1.00 55

le 0.98 0.98 0.98 43

nho 0.96 1.00 0.98 55

quyt 1.00 1.00 1.00 52

tao 1.00 0.93 0.96 40

thom 1.00 0.98 0.99 52

xoai 1.00 0.98 0.99 42

accuracy 0.98 454

macro avg 0.96 0.95 0.95 454

weighted avg 0.99 0.98 0.98 454

A screenshot of a graph

Description automatically generated

Hình 20: Ma trận đánh giá MobileNetV2

### **Đánh giá mô hình ResNet50**

Mô hình ResNet50 - Độ chính xác trên tập kiểm tra: 63.66%

Độ mất mát (Loss): 3.6060

Classification Report:

precision recall f1-score support

cachua 0.85 0.63 0.73 52

cam 1.00 0.55 0.71 60

chuoi 0.02 1.00 0.04 3

duahau 1.00 0.96 0.98 55

le 0.00 0.00 0.00 43

nho 0.71 1.00 0.83 55

quyt 0.86 0.60 0.70 52

tao 0.00 0.00 0.00 40

thom 1.00 0.92 0.96 52

xoai 0.92 0.79 0.85 42

accuracy 0.64 454

macro avg 0.64 0.65 0.58 454

weighted avg 0.73 0.64 0.67 454

A screenshot of a graph

Description automatically generated

Hình 21: Ma trận đánh giá ResNet50

### **Đánh giá mô hình KNN**

Accuracy: 90.53%

Độ mất mát (Loss): 0.0947

Classification Report:

precision recall f1-score support

cachua 0.94 0.90 0.92 52

cam 0.98 1.00 0.99 60

chuoi 1.00 0.67 0.80 3

duahau 0.93 0.93 0.93 55

le 0.75 0.95 0.84 43

nho 0.95 0.67 0.79 55

quyt 0.91 1.00 0.95 52

tao 0.90 0.95 0.93 40

thom 0.89 0.81 0.85 52

xoai 0.89 0.98 0.93 42

accuracy 0.91 454

macro avg 0.91 0.89 0.89 454

weighted avg 0.91 0.91 0.90 454

A screenshot of a graph

Description automatically generated

Hình 22: Ma trận đánh giá KNN

### Biểu đồ đánh giá giữa các thuật toán trên

A graph with different colored squares

Description automatically generated

Hình 23: Mô hình đánh giá giữa các thuật toán

# KẾT QUẢ THỰC NGHIỆM

*(Tùy thuộc vào hướng triển khai của đề tài mà nhóm trình bày các kết quả thực nghiệm liên quan. Cần có phần biện luận để giải thích, so sánh các kết quả thực nghiệm.)*

## Kết Quả các thuật toán

### **CNN**



Hình 24:Kết quả dự đoán của mô hình CNN

### **MobileNetV2**

A bunch of bananas on a white background

Description automatically generated

Hình 25: Kết quả dự đoán của mô hình MobileNetV2

### **RetNet50**

A close up of a fruit

Description automatically generated

Hình 26: Kết quả dự đoán của mô hình ResNet50

### **KNN**

A close up of a fruit

Description automatically generated

Hình 27: Kết quả dự đoán của mô hình KNN

## So sánh

### **CNN**

#### Đặc điểm

**Cấu trúc**: Gồm các lớp tích chập (convolutional layers), lớp phi tuyến (activation layers, thường là ReLU), và lớp pooling để giảm chiều dữ liệu nhưng vẫn giữ lại các đặc trưng quan trọng.

**Khả năng học đặc trưng**: CNN tự động học các đặc trưng không gian từ dữ liệu hình ảnh, từ những đặc trưng cơ bản như cạnh và góc cho đến các đặc trưng phức tạp ở các lớp sâu hơn.

#### Ưu điểm

**Hiệu suất cao trên hình ảnh**: CNN đã chứng minh hiệu quả vượt trội trong các bài toán xử lý hình ảnh, đặc biệt là phân loại và nhận diện đối tượng.

**Khả năng mở rộng**: Có thể mở rộng và tùy chỉnh số lớp để phù hợp với yêu cầu độ chi tiết của bài toán.

#### Nhược điểm

**Yêu cầu phần cứng cao**: Để huấn luyện một CNN sâu cần GPU mạnh và nhiều bộ nhớ.

**Thời gian huấn luyện dài**: Nếu mạng có nhiều lớp, thời gian huấn luyện có thể kéo dài đáng kể.

### **MobileNetV2**

#### Đặc điểm

**Kiến trúc**: Dựa trên cấu trúc MobileNetV1 nhưng cải thiện với các khối bottleneck depthwise separable convolution và ReLU6, giúp tăng cường hiệu suất và giảm số lượng tham số.

**Độ gọn nhẹ**: Được thiết kế để chạy nhanh trên thiết bị di động và hệ thống nhúng, phù hợp cho các ứng dụng thời gian thực.

#### Ưu điểm

**Hiệu suất cao trên thiết bị nhỏ**: Tối ưu hóa để hoạt động mượt mà trên CPU, giúp giảm tải cho hệ thống và tiết kiệm năng lượng.

**Tốc độ suy luận nhanh**: Thời gian dự đoán ngắn hơn, phù hợp cho các ứng dụng yêu cầu phản hồi ngay lập tức như hệ thống bán hàng tự động.

#### Nhược điểm

**Độ chính xác**: Dù hiệu suất khá tốt, MobileNetV2 có thể không đạt được độ chính xác cao bằng các mô hình lớn hơn nếu không được tinh chỉnh phù hợp.

**Khó khăn khi xử lý chi tiết phức tạp**: Với những hình ảnh cần nhiều lớp sâu để phân tích chi tiết, MobileNetV2 có thể không phát huy tối đa khả năng.

### **RetNet50**

#### Đặc điểm

**Kiến trúc residual**: Sử dụng các khối residual giúp thông tin truyền qua nhiều lớp một cách hiệu quả, giải quyết vấn đề biến mất gradient khi huấn luyện mạng sâu.

**Số lớp**: ResNet50 có 50 lớp, cân bằng giữa độ sâu và hiệu suất, đảm bảo trích xuất đặc trưng phức tạp.

#### Ưu điểm

**Khả năng học sâu**: Hiệu quả trong việc phân loại hình ảnh có độ phức tạp cao nhờ khả năng học được đặc trưng từ nhiều lớp sâu.

**Độ chính xác cao**: Trong nhiều bài toán, ResNet50 cho kết quả vượt trội về độ chính xác.

#### Nhược điểm

**Yêu cầu tài nguyên cao**: Cần phần cứng mạnh để đạt được tốc độ huấn luyện và suy luận tối ưu.

**Thời gian xử lý**: Mặc dù hiệu suất cao, việc huấn luyện và triển khai có thể mất thời gian dài.

### **KNN**

#### Đặc điểm

**Nguyên lý**: Tìm kiếm các điểm dữ liệu gần nhất trong không gian đặc trưng để phân loại mẫu mới.

**Đơn giản và dễ triển khai**: Không cần giai đoạn huấn luyện phức tạp, chỉ cần lưu trữ dữ liệu và so sánh khoảng cách khi dự đoán.

#### Ưu điểm

**Đơn giản**: Cấu trúc thuật toán đơn giản, dễ dàng áp dụng vào bài toán nhỏ.

**Không cần huấn luyện**: Tiết kiệm thời gian ở giai đoạn chuẩn bị mô hình.

#### Nhược điểm

**Không tối ưu cho dữ liệu lớn**: Khi số lượng mẫu và chiều dữ liệu tăng, tốc độ suy luận giảm đáng kể.

**Hiệu suất kém trên hình ảnh phức tạp**: KNN không phù hợp với dữ liệu hình ảnh có độ phân giải cao và nhiều chi tiết cần xử lý.

### **Kết luận**

**ResNet50** và **CNN** cơ bản là lựa chọn tối ưu nếu bạn cần độ chính xác cao và tài nguyên máy tính đủ mạnh.

**MobileNetV2** là lựa chọn phù hợp khi cần sự cân bằng giữa hiệu suất và tốc độ trên các thiết bị có tài nguyên hạn chế.

**KNN** thích hợp để thử nghiệm nhanh, nhưng không phải là lựa chọn tối ưu cho hệ thống thực tế với dữ liệu phức tạp.

**Kết luận tổng quát**: MobileNetV2 là lựa chọn tối ưu cho bài toán, trong khi các mô hình khác như KNN và CNN cũng có thể được sử dụng trong các trường hợp cụ thể. ResNet50 cần được tối ưu hóa thêm hoặc không phù hợp với bài toán này.

# KẾT LUẬN

## Kết quả đạt được

Trong quá trình thực hiện nghiên cứu, các kết quả đạt được bao gồm:

### **So sánh hiệu suất các mô hình:**

* **MobileNetV2** chứng tỏ là mô hình có độ chính xác cao nhất, đạt **98.46%**. Đây là mô hình phù hợp nhất cho bài toán nhờ hiệu suất vượt trội và thời gian huấn luyện tối ưu.

* **CNN** đạt hiệu suất rất tốt với độ chính xác **98.02%**, đứng thứ hai sau MobileNetV2. Kết quả này thể hiện khả năng xử lý hiệu quả của mô hình đối với bài toán, tuy nhiên chưa vượt trội so với MobileNetV2.
* **KNN** mặc dù là mô hình đơn giản, nhưng vẫn cho kết quả ấn tượng với độ chính xác **90.53%**, đứng thứ ba. Điều này cho thấy mô hình truyền thống này vẫn có thể hoạt động hiệu quả nếu được áp dụng phù hợp với tập dữ liệu.
* **ResNet50** mặc dù là một mạng sâu tiên tiến, lại cho kết quả thấp nhất với độ chính xác **63.66%**. Điều này có thể do cần điều chỉnh thêm hoặc mô hình không phù hợp với tập dữ liệu đang sử dụng.

### **Đánh giá và rút ra bài học:**

* Các kết quả này giúp nhận diện rõ mô hình nào phù hợp nhất cho bài toán phân loại/dự đoán cụ thể.
* Đồng thời, qua việc triển khai các mô hình khác nhau, chúng ta đã hiểu rõ hơn về sự ảnh hưởng của cấu trúc mạng và dữ liệu đến hiệu suất của từng phương pháp.

### **Ứng dụng tiềm năng:**

* Với độ chính xác cao của MobileNetV2, mô hình này có thể được áp dụng trong các hệ thống thực tế để thực hiện nhiệm vụ phân loại một cách hiệu quả.
* Các mô hình khác như KNN hoặc CNN có thể được áp dụng trong những trường hợp yêu cầu khác nhau hoặc dùng để so sánh.

## Những khó khăn, hạn chế

### **Khó Khăn**

1. Chất lượng dữ liệu: Bộ dữ liệu ảnh trái cây có thể chứa các ảnh có chất lượng khác nhau, độ phân giải khác nhau, điều kiện ánh sáng khác nhau, góc chụp khác nhau. Điều này có thể làm giảm hiệu suất của mô hình.
2. Overfitting: Mô hình CNN có thể bị overfitting nếu dữ liệu huấn luyện không đủ đa dạng. Overfitting xảy ra khi mô hình học quá tốt trên dữ liệu huấn luyện và không khái quát hóa tốt trên dữ liệu mới.
3. Thời gian huấn luyện: Huấn luyện các mô hình phức tạp như CNN, MobileNetV2, ResNet50 có thể tốn nhiều thời gian, đặc biệt nếu bộ dữ liệu lớn và tài nguyên tính toán hạn chế.
4. Lựa chọn siêu tham số: Việc lựa chọn các siêu tham số như số lượng lớp, số lượng epoch, learning rate, batch size có thể ảnh hưởng đáng kể đến hiệu suất của mô hình. Cần phải thử nghiệm và điều chỉnh để tìm ra các giá trị tối ưu.
5. Kích thước bộ dữ liệu: Nếu bộ dữ liệu quá nhỏ, mô hình có thể không học được các đặc trưng quan trọng của các loại trái cây và dẫn đến hiệu suất thấp.

### **Hạn Chế**

1. Phân loại các loại trái cây tương tự: Mô hình có thể gặp khó khăn trong việc phân biệt các loại trái cây có hình dạng và màu sắc tương tự nhau, ví dụ như cam và quýt.
2. Ảnh có nhiều đối tượng: Nếu một ảnh chứa nhiều loại trái cây khác nhau, mô hình có thể không phân loại chính xác.
3. Trái cây bị che khuất: Nếu một phần của trái cây bị che khuất, mô hình có thể gặp khó khăn trong việc nhận dạng.
4. Khả năng mở rộng: Việc mở rộng mô hình để phân loại thêm các loại trái cây mới có thể yêu cầu thu thập thêm dữ liệu và huấn luyện lại mô hình.

# TÀI LIỆU THAM KHẢO

# PHỤ LỤC

**Bảng phân công công việc của các thành viên trong nhóm**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Nội dung công việc** | 22DH 112165  Nguyễn Quốc Minh | 22DH110505  Trường Triều Dĩ | 22DH114826  Lê Phạm Hoàng Vũ |
| 1 | Viết báo cáo | 35% | 35% | 30% |
| 2 | Thực hiện mô hình CNN | 40% | 30% | 30% |
| 3 | Thực hiện mô hình MobileNetV2 | 20% | 50% | 30% |
| 4 | Thực hiện mô hình ResNet50 | 30% | 30% | 40% |
| 5 | Thực hiện mô hình KNN | 30% | 30% | 30% |

*)*