

Xây Dựng Mô Hình Nhận Diện Rau Củ Thông Qua Hình Ảnh

1st Đặng Thị Thúy Hồng
Trường Đại học Công nghệ Thông tin
ĐHQG-TPHCM
20520523@gm.uit.edu.vn

2nd Ngô Thị Phúc
Trường Đại học Công nghệ Thông tin
ĐHQG-TPHCM
20521765@gm.uit.edu.vn

3rd Nguyễn Thị Nguyệt
Trường Đại học Công nghệ Thông tin
ĐHQG-TPHCM
20521689@gm.uit.edu.vn

I. GIỚI THIỆU

A. Tóm tắt

Trong thế giới hiện nay, việc tạo ra các thiết bị tích hợp công nghệ trí tuệ nhân tạo phục vụ cho đời sống con người đang nhận được sự ủng hộ và thúc đẩy rất lớn. Trong đề tài lần này, chúng tôi muốn xây dựng một mô hình phục vụ cho việc tạo ra những thiết bị thông minh như thế, cụ thể là thiết bị dành cho những trung tâm thương mại hoặc những nơi phân phối bán lẻ. Trong khuôn khổ đề án này, chúng tôi đã xây dựng mô hình phân lớp đa lớp để phân loại và nhận biết vật thể là rau củ với 9 nhãn, chúng tôi đã sử dụng nhiều phương pháp xử lý ảnh cũng như nhiều phương pháp học máy để tìm ra mô hình phù hợp nhất. Bước đầu đạt kết quả độ chính xác 83.44% với phương pháp SVM.

B. Tên đề tài:

Nhận diện rau củ thông qua hình ảnh

C. Bối cảnh:

Trong bối cảnh cuộc Cách mạng công nghiệp 4.0 phát triển nhanh chóng, máy học trở thành một công nghệ cực kỳ quan trọng, nó mở đường cho việc tự động hóa nhiều công việc mà con người mất nhiều công sức và tiền bạc để làm. Một trong số đó là việc hỗ trợ những trung tâm thương mại, những nơi phân phối bán lẻ trong việc giảm thiểu một số các chi phí.

Giả sử có thể tạo ra một thiết bị giống như một chiếc máy ảnh khi người mua hàng đưa mặt hàng rau củ mà mình vừa chọn lựa vào thì máy sẽ giúp họ xác định mặt hàng đó là gì, trọng lượng, giá tiền bao nhiêu. Việc này sẽ giúp ích được rất nhiều cho người mua hàng không cần phải tốn công sức xếp hàng đợi thanh toán, giúp các nơi phân phối bán lẻ giảm chi phí thuê nhân công tiết kiệm được một phần chi phí. Đặc biệt trong tình hình dịch Covid-19 vẫn chưa chấm dứt hoàn toàn sẽ giúp giảm thiểu tối đa sự tiếp xúc giữa người với người.

Trong đề tài này, chúng tôi thực hiện việc đào tạo máy học để có thể phát hiện 9 loại rau củ khác nhau qua hình ảnh. Sau khi hoàn thành, chúng tôi có thể phát triển, đào tạo mô hình này với nhiều loại rau củ hơn.

D. Định nghĩa bài toán:

Task: Phân loại rau củ có trong hình.

E: Tập dữ liệu rau củ.

P: Tỷ lệ nhận diện chính xác loại rau củ có trong hình.

Input: Một hình ảnh của loại rau củ.

Output: Nhận được dự đoán cho bức ảnh đó.

E. Mục tiêu:

- Nhận diện phân loại được 9 loại rau củ có trong bộ dữ liệu với độ chính xác khoảng 70%.
- Tìm hiểu và thực hiện được quy trình các bước để tạo một bộ dữ liệu cũng như cách xây dựng, huấn luyện và đánh giá mô hình máy học.

F. Ứng dụng:

Ứng dụng của nhóm: Một thiết bị như một máy ảnh. Khi người sử dụng đưa vật thể ra trước máy ảnh thì thiết bị sẽ giúp phân biệt và nhận diện vật thể đó.

G. Công trình liên quan:

Fruit and Vegetable Identification Using Machine Learning for Retail Applications [8]

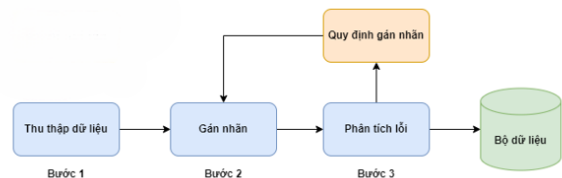
(Frida Femling – Chalmers University of Technology)

Là một trong những công trình mà chúng tôi đã tham khảo. Công trình sử dụng hình ảnh được chụp bằng máy quay video gắn vào hệ thống, hệ thống này giúp khách hàng gắn nhãn các loại trái cây và rau quả mong muốn với giá tùy vào trọng lượng của nó. Mục đích của hệ thống là giảm thiểu số lượng tương tác với máy tính của con người, tăng tốc quá trình nhận dạng và cải thiện khả năng sử dụng của giao diện người dùng đồ họa so với các hệ thống thủ công hiện có.

II. DỮ LIỆU

A. Thu thập dữ liệu

Nhóm chúng tôi đã xây dựng bộ dữ liệu VID dựa trên bộ dữ liệu Vegetable Images trên nền tảng Kaggle. Bên cạnh đó chúng tôi đã thu thập thêm hình ảnh từ Internet để hoàn thiện bộ dữ liệu của chúng tôi.



Hình 2.1 Quy trình xây dựng bộ dữ liệu VID

B. Thông tin bộ dữ liệu

Bộ dữ liệu:

VID là bộ dữ liệu chúng tôi xây dựng dựa trên bộ dữ liệu Vegetable Images có sẵn và công khai trên Kho lưu trữ Kaggle. Bên cạnh đó, chúng tôi bổ sung thêm vào bộ dữ liệu một lượng dữ liệu được khai thác bằng tay trên Internet.

Dataset	Total	Train	Validation	Test
Sample size	9000	5400	1800	1800
Bottle_Gourd	1000	600	200	200
Broccoli	1000	600	200	200
Cabbage	1000	600	200	200
Carrot	1000	600	200	200
Cauliflower	1000	600	200	200
Potato	1000	600	200	200
Pumpkin	1000	600	200	200
Radish	1000	600	200	200
Tomato	1000	600	200	200

Bảng 2.1 Thông tin bộ dữ liệu

Sau khi xử lý bộ dữ liệu thô, chúng tôi xử lý thủ công và tạo ra bộ dữ liệu bao gồm 9000 điểm dữ liệu với 9 loại rau củ tương ứng với 9 nhãn được miêu tả trong Bảng 1

Chúng tôi đã tiến hành chia bộ dữ liệu thành 3 tập là train set, validation set và test set với tỉ lệ lần lượt là 6:2:2.

Tên thuộc tính	Định nghĩa
Bottle_Gourd	Bầu
Broccoli	Bông cải xanh
Cabbage	Bắp cải
Carrot	Cà rốt
Cauliflower	Bông cải trắng
Potato	Khoai tây
Pumpkin	Bí ngô
Radish	Cải củ
Tomato	Cà chua

Bảng 2.2 Bảng chi tiết các thuộc tính của bộ dữ liệu VID

Link bộ dữ liệu: https://bit.ly/VID_dataset

C. Quá trình gán nhãn

Quá trình gán nhãn được thực hiện dựa trên quy định gán nhãn do chúng tôi xây dựng dựa theo sự thống nhất của tất cả thành viên nhóm.

STT	Mô tả
1	Ảnh chứa 1 trong 9 loại rau củ được đề cập trong Bảng 1. Tối thiểu là 1 loại, tối đa là 9 loại
2	Chấp nhận ảnh của các loại rau củ bị cắt mất một phần, bị mờ trong khoảng chấp nhận được hoặc loại rau củ đó bị cắt ra thành từng phần nhưng vẫn nhận dạng được
3	Chấp nhận hình ảnh những loại rau củ bị biến dạng và biến đổi về màu sắc, tuy nhiên loại rau củ đó phải nhận diện được dựa trên những thuộc tính khác

Bảng 2.3 Bảng quy định gán nhãn

Khi gán nhãn, phải nhận được sự đồng thuận từ 2/3 thành viên trở lên, nếu không điểm dữ liệu sẽ được bỏ đi và thay bằng một hình ảnh khác.

Hướng dẫn gán nhãn:

<https://bit.ly/20520523> 20521765 20521689 [Huong dan g an nhan](#)

Độ đồng thuận:

Trong quá trình gán nhãn, chúng tôi đã sử dụng công thức Kappa của Cohen để đánh giá mức độ đồng thuận, Công thức đó được phát biểu như sau^[16]:

Trong đó:

- Am là hệ số tương quan giữa đồng thuận kỳ vọng và đồng thuận thực tế

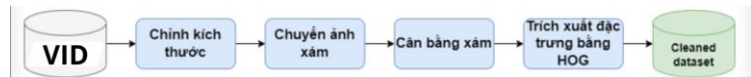
- P_0 là mức độ đồng thuận thực tế

- P_e là mức độ đồng thuận kỳ vọng

$$Am = \frac{P_0 - P_e}{1 - P_e}$$

Trong trường hợp thử với mẫu là 1200 bức ảnh, chúng tôi nhận được kết quả rất khả quan, khoảng 0.75. Điều này chứng tỏ tất cả người gán đã có mức độ đồng thuận trong việc gán nhãn là rất cao. Chúng tôi tiếp tục thực hiện đề tài với quy định gán nhãn đã thống nhất.

D. Phương pháp xử lý dữ liệu



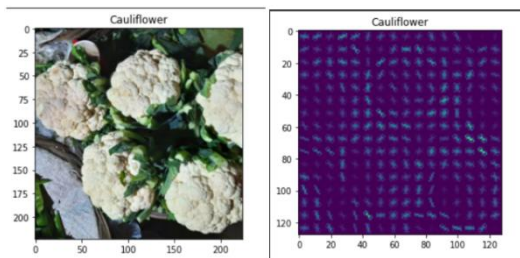
Hình 2.2 Sơ đồ xử lý dữ liệu

Các phương pháp xử lý dữ liệu chúng tôi đã sử dụng:

- Resize ảnh về cùng một kích thước.
- Phương pháp Grayscale (Chuyển ảnh xám sử dụng thư viện Pillow).
- Cân bằng sáng sử dụng Histogram Equalization sử dụng thư viện OpenCV
- Trích xuất đặc trưng bằng HOG (Histogram of Oriented Gradients).

Sau khi thử qua các phương pháp xử lý dữ liệu (Phát hiện cạnh Canny, Làm mờ GaussBlur, Tăng độ tương phản, trích xuất đặc trưng HOG) chúng tôi nhận thấy rằng sử dụng phương pháp trích xuất đặc trưng HOG là phương pháp mang lại hiệu quả trong việc huấn luyện mô hình, đem lại kết quả tốt nhất. Chính vì thế chúng tôi đã chọn phương pháp này.

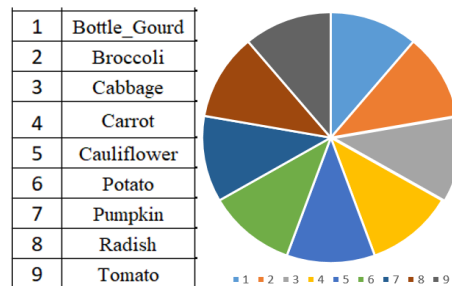
Trực quan hóa một số hình ảnh



Hình 2.3 Hình ảnh trước và sau khi xử lý

E. Đánh giá tập dữ liệu đã gán nhãn

Sau khi gán nhãn xong, chúng tôi có đánh giá về tập dữ liệu như sau:



Hình 2.4 Phân bố số lượng rau củ trong bộ dữ liệu

- Các điểm dữ liệu được gán nhãn đảm bảo đúng theo quy định gán nhãn
- Các điểm dữ liệu có nhiều kích thước khác nhau, cần được tiền xử lý trước khi thực hiện huấn luyện
- Tập dữ liệu có kích thước nhỏ khi so với các công trình khác
- Các nhãn có số lượng điểm dữ liệu bằng nhau. Tập train, tập validation và tập test chia theo đúng tỉ lệ 6:2:2

F. Thách thức trong quá trình xây dựng bộ dữ liệu

Trong quá trình xây dựng bộ dữ liệu chúng tôi đã gặp phải một số thách thức như sau:

- Việc thu thập hình ảnh từ nhiều nguồn trên internet không đảm bảo sự đồng đều về chất lượng, kích thước, chủ thể, khiến quá trình tiền xử lý, làm sạch dữ liệu trở nên khó khăn hơn
- Có một số lượng lớn dữ liệu bị trùng lặp sau khi tải về từ internet, chúng tôi phải thực hiện xử lý những dữ liệu trùng lặp thủ công.
- Trong quá trình tải, có một số lượng dữ liệu bị lẫn file rác, lẫn những hình ảnh không thuộc 9 loại rau củ trong đề tài này

III. PHƯƠNG PHÁP MÁY HỌC

A. Mô hình máy học

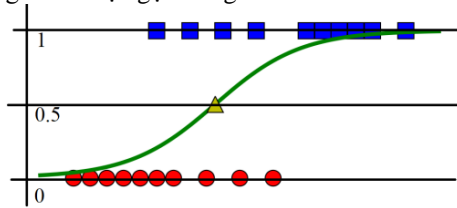
1. Logistic Regression

Logistic Regression là một thuật toán phân loại có đầu ra là số thực, thường được sử dụng nhiều cho các bài toán classification dùng để gán các đối tượng cho một tập hợp giá trị rời rạc. Ví dụ các bài toán như phân loại email, dương tính hay âm tính với Covid. Và chúng tôi đã áp dụng phương pháp này để phân loại loài vật trong hình ảnh.

Đầu ra dự đoán của Logistic Regression có dạng viết chung là:

$$f(\mathbf{x}) = \theta(\mathbf{w}^T \mathbf{x})$$

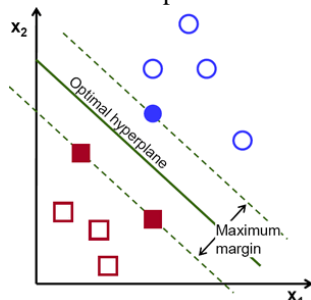
Trong đó θ được gọi là logistic function.



Hình 3.1 Mô hình Logistic Regression

2. Support Vector Machine (SVM)

Support Vector Machine là một thuật toán học có giám sát chủ yếu dùng cho việc phân loại. Thuật toán sẽ tìm ra hyper-plane phân chia các lớp ra thành hai phần riêng biệt.



Hình 3.2 Mô hình Support Vector Machine

Hàm quyết định phân dữ liệu vào lớp thứ i là:

$$f_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + b_i$$

Trong đó w là vector pháp tuyến n chiều và b là giá trị ngưỡng. Vector pháp tuyến w xác định chiều của siêu phẳng $f(\mathbf{x})$, giá trị ngưỡng b xác định khoảng cách giữa siêu phẳng và gốc.

3. K-Nearest Neighbors (KNN)

K-Nearest Neighbors là một trong những thuật toán học có giám sát đơn giản nhất trong Machine Learning, hoạt động tốt trong trường hợp phân loại với nhiều lớp. Thuật toán này sẽ thực hiện mọi tính toán khi cần dự đoán kết quả của dữ liệu mới.

- Khoảng cách Euclidean là một trong các độ đo khoảng cách được sử dụng phổ biến trong KNN.

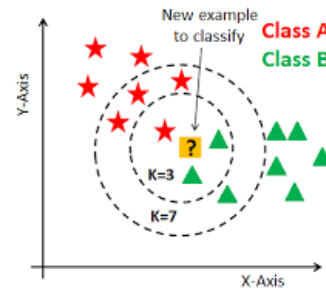
$$D(a, b) = \sqrt{\sum_{i=1}^n (b_i - a_i)^2}$$

Trong đó a_i là đặc trưng thứ i của dữ liệu.

- Khoảng cách Hamming là độ đo được sử dụng khi làm việc với dữ liệu rời rạc.

$$D(a, b) = \sum_{i=1}^n (a_i \neq b_i)$$

Trong đó n là số lượng đặc trưng của dữ liệu.



Hình 3.3 Mô hình K-Nearest Neighbors

B. Công cụ sử dụng

Nền tảng sử dụng: Google Colab.

Thư viện sử dụng:

- Sklearn: là thư viện cung cấp một tập các công cụ xử lý các bài toán machine learning và statistical modeling.
- Pandas: là thư viện dùng để thao tác, phân tích và dọn dẹp dữ liệu.
- Numpy: là thư viện chúng tôi dùng để xử lý mảng đa chiều, ma trận.
- Cv2: là thư viện cho phép đọc, ghi, thay đổi dữ liệu nhiều hình ảnh cùng một lúc.
- Math: module này cho phép truy cập vào các hàm lượng giác, hàm số, hàm logarit cho các số thực.
- Os: module này cho phép thao tác với tệp và thư mục.
- Random: module cho phép tạo ra một số ngẫu nhiên bất kỳ với nhiều yêu cầu khác nhau.
- Matplotlib: là thư viện dùng để vẽ đồ thị 2D.
- Seaborn: là một thư viện trực quan hóa dữ liệu.
- PIL: có thể mở, lưu, xử lý đặc điểm hình ảnh với nhiều định dạng ảnh khác nhau.
- Skimage: là một thư viện xử lý ảnh nguồn mở bao gồm các thuật toán để phân đoạn, thao tác không gian màu, phân tích, lọc, phát hiện tính năng, ...

Công cụ khác:

- GridSearchCV: Lấy một từ điển mô tả các tham số có thể được thử trên một mô hình để huấn luyện nó. Lưới tham số được định nghĩa như một từ điển, trong đó các khóa là các tham số và các giá trị là cài đặt cần kiểm tra.

C. Các phương pháp đánh giá

Confusion matrix: Giúp đánh giá được các giá trị cụ thể mỗi loại được phân loại như thế nào, lớp nào được phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào hay bị phân loại nhầm vào lớp khác.

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

Hình 3.4 Confusion Matrix

- True Positive (TP): Giá trị dự đoán khớp với giá trị thực tế. Nhân dương tính được dự đoán là dương tính.
- True Negative (TN): Giá trị dự đoán khớp với giá trị thực tế. Nhân âm tính được dự đoán là âm tính.
- False Positive (FP): Giá trị dự đoán bị sai. Nhân âm tính bị dự đoán là dương tính.
- False Negative (FN): Giá trị dự đoán bị sai. Nhân dương tính bị dự đoán là âm tính.

Chúng tôi sử dụng 2 độ đo sau đây để đánh giá mô hình:

1. F1-score:

F1-score là trung bình điều hòa của recall và precision. F1-score nằm trong khoảng (0,1] F1 càng cao, bộ phân loại càng tốt.

$$F1 = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Trong đó:

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

Bản chất bài toán phân loại loài vật là phân lớp đa lớp với các nhãn có trọng số như nhau, do vậy recall cao mà precision thấp thì không phải mô hình tốt và ngược lại, precision cao mà recall thấp cũng không phải mô hình tốt. Do vậy chúng tôi quyết định sử dụng F1-score.

2. Accuracy

Accuracy đánh giá mô hình bằng cách tính tỉ lệ giữa số điểm được dự đoán đúng và tổng số điểm trong tập dữ liệu kiểm thử.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Đề tài của chúng tôi có điểm dữ liệu của các nhãn bằng nhau, vậy nên chúng tôi dùng thêm Accuracy vì đây là độ đo rất phù hợp với bài toán phân lớp có bộ dữ liệu cân bằng.

IV. CÁC THỬ NGHIỆM TÍNH CHÍNH MÔ HÌNH

Sau khi chạy ba mô hình máy học bao gồm: Logistic Regression, Support Vector Machine, K - Nearest Neighbor với tham số mặc định, nhận thấy:

Thuật toán Support Vector Machine (82.18% - 82.22%) đã đem lại kết quả cao nhất trong ba thuật toán.

Trái ngược lại thì Thuật toán K - Nearest Neighbor đem lại kết quả rất thấp (59.22% - 63.11%).

Kết quả được thể hiện ở bảng 4.1.

Mô hình	Kết quả độ đo F1 (%)	Kết quả độ đo Accuracy (%)
Logistic Regression	74.38	74.44
Support Vector Machine	82.18	82.22
K - Nearest Neighbor	63.11	59.22

Bảng 4.1: Kết quả chạy mô hình Máy học với tham số mặc định

Chúng tôi tiếp tục tinh chỉnh tham số của mỗi phương pháp với phương pháp GridSearchCV để tăng hiệu suất mô hình và nhận được kết quả thể hiện ở bảng 4.2 cùng với bộ tham số thu được như sau:

- Thuật toán Logistic Regression với tham số 'C': 0.01, 'penalty': 'l2'.
- Thuật toán Support Vector Machine với tham số 'C':1, 'gamma':0.0001, 'kernel': 'rbf'.
- Thuật toán K - Nearest Neighbor với 'n_neighbors': 1, 'p':1.

Mô hình	Kết quả độ đo F1 (%)	Kết quả độ đo Accuracy (%)
Logistic Regression	70.15	70.39
Support Vector Machine	83.44	83.44
K - Nearest Neighbor	89.29	89.11

Bảng 4.2: Kết quả mô hình sau khi tinh chỉnh tham số

Sau khi chạy các mô hình với bộ tham số tìm được, nhận thấy:

Mô hình K - Nearest Neighbor đã được cải thiện so với khi chạy mô hình với tham số mặc định. Tăng lên đáng kể nhất từ 59.22% - 63.11% lên 89.11% - 89.29%.

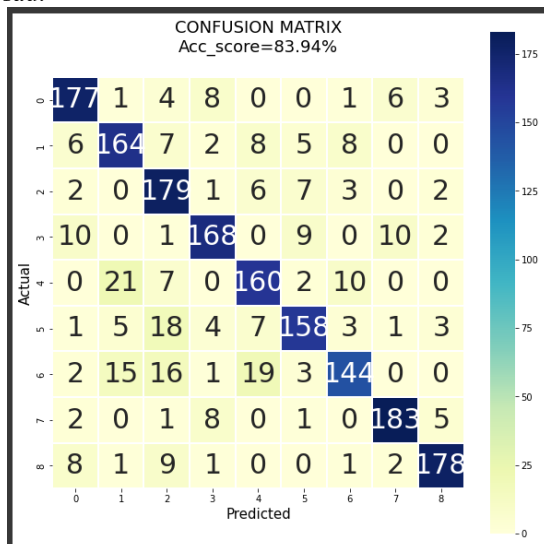
Mô hình Logistic Regression có giảm xuống từ 74.38% - 74.44% xuống 70.15% - 70.39%.

Thuật toán Support Vector Machine tăng lên không quá nhiều, từ 82.18% - 82.22% tăng lên 83.44% - 83.44%.

V. PHÂN TÍCH LỖI, HƯỚNG PHÁT TRIỂN

A. Phân tích lỗi

Trong ba thuật toán học máy mà chúng tôi đã áp dụng, chúng tôi nghĩ rằng Support Vector Machine mang lại kết quả chính xác nhất. Vì vậy chúng tôi chọn mô hình này để phân tích lỗi. Ma trận nhầm lẫn được sử dụng để hiểu rõ những lớp nào được phân loại đúng hay nhầm vào các lớp khác. Từ ma trận nhầm lẫn, ta có thể tính được tỉ lệ dự đoán trên các nhãn như sau:



Hình 5.1: Ma trận nhầm lẫn trên tập kiểm tra với thuật toán Support Vector Machine

- Tỉ lệ dự đoán đúng trên các hình ảnh loại Bottle_Gourd đạt $177/200 = 88.5\%$.
- Tỉ lệ dự đoán đúng trên các hình ảnh loại Broccoli đạt $164/200 = 82\%$.
- Tỉ lệ dự đoán đúng trên các hình ảnh loại Cabbage đạt $179/200 = 89.5\%$.
- Tỉ lệ dự đoán đúng trên các hình ảnh loại Carrot đạt $168/200 = 84\%$.
- Tỉ lệ dự đoán đúng trên các hình ảnh loại Cauliflower đạt $160/200 = 80\%$.
- Tỉ lệ dự đoán đúng trên các hình ảnh loại Potato đạt $158/200 = 79\%$.
- Tỉ lệ dự đoán đúng trên các hình ảnh loại Pumpkin đạt $144/200 = 72\%$.
- Tỉ lệ dự đoán đúng trên các hình ảnh loại Radish đạt $183/200 = 91.5\%$. Tỉ lệ cao nhất.
- Tỉ lệ dự đoán đúng trên các hình ảnh loại Tomato đạt $178/200 = 89\%$.

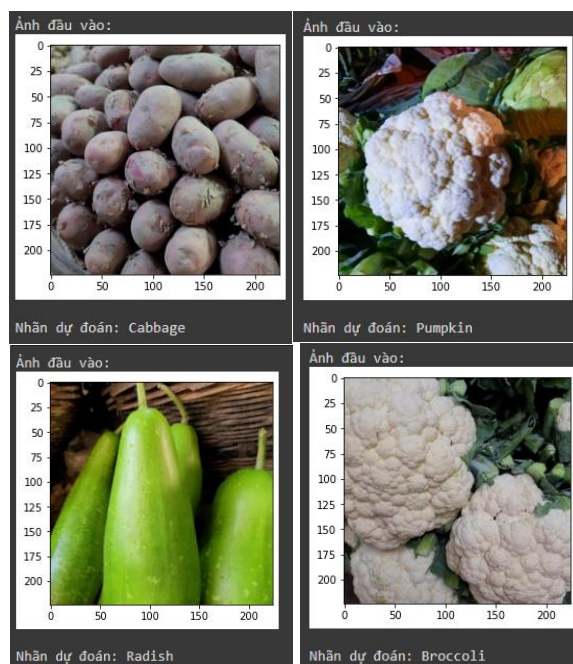
Theo như ma trận nhầm lẫn, ta nhận thấy:

- Trong các hình ảnh loại Bottle_Gourd (nhãn 0) bị dự đoán nhầm thì số lượng dự đoán nhầm thành loại Carrot (nhãn 3) là nhiều nhất (8).
- Trong các hình ảnh loại Broccoli (nhãn 1) bị dự đoán nhầm thì số lượng dự đoán nhầm thành loại

Cauliflower (nhãn 4) và loại Pumpkin (nhãn 6) là nhiều nhất (8).

- Trong các hình ảnh loại Cabbage (nhãn 2) bị dự đoán nhầm thì số lượng dự đoán nhầm thành loại Potato (nhãn 5) là nhiều nhất (7).
- Trong các hình ảnh loại Carrot (nhãn 3) bị dự đoán nhầm thì số lượng dự đoán nhầm thành loại Bottle_Gourd (nhãn 0) và loại Radish (nhãn 7) là nhiều nhất (10).
- Trong các hình ảnh loại Cauliflower (nhãn 4) bị dự đoán nhầm thì số lượng dự đoán nhầm thành loại Broccoli (nhãn 2) là nhiều nhất (21).
- Trong các hình ảnh loại Potato (nhãn 5) bị dự đoán nhầm thì số lượng dự đoán nhầm thành loại Cabbage (nhãn 2) là nhiều nhất (18).
- Trong các hình ảnh loại Pumpkin (nhãn 6) bị dự đoán nhầm thì số lượng dự đoán nhầm thành loại Cauliflower (nhãn 4) là nhiều nhất (19).
- Trong các hình ảnh loại Radish (nhãn 7) bị dự đoán nhầm thì số lượng dự đoán nhầm thành loại Carrot (nhãn 3) là nhiều nhất (8).
- Trong các hình ảnh loại Tomato (nhãn 8) bị dự đoán nhầm thì số lượng dự đoán nhầm thành loại Cabbage (nhãn 2) là nhiều nhất (9).

Một số hình ảnh bị dự đoán sai:



Hình 5.2 Một số hình ảnh ví dụ mô hình phân loại sai

Chúng tôi nhận thấy trong các hình ảnh bị phân loại sai có một số đặc điểm như sau:

- Số lượng rau củ trong ảnh nhiều dễ gây nhiễu.
- Một số hình ảnh của rau củ bị cắt mất một phần không còn nguyên dạng như ban đầu dễ bị nhầm sang loại rau củ khác có hình dáng gần tương đồng.
- Hình ảnh rau củ được chụp quá gần dễ gây nhầm lẫn.
- Những hình ảnh của loại Cauliflower thường bị nhầm lẫn thành loại Broccoli bởi vì 2 loại này có một số đặc điểm tương đồng nhau.

B. Hướng phát triển

Từ những kết quả đã đạt được, chúng tôi đã vạch ra những bước phát triển tiếp cho việc nghiên cứu đề tài này:

Trước tiên, chúng tôi sẽ xây dựng một bộ dữ liệu tốt hơn về mặt chất lượng lẫn số lượng, chú trọng hơn trong việc tiền xử lý dữ liệu, tinh chỉnh mô hình để có thể đạt độ chính xác cao hơn.

Cần phát triển bộ dữ liệu với nhiều nhãn hơn và đa dạng về hình ảnh hơn để hướng đến mục tiêu có thể ứng dụng vào thực tế, thỏa mãn được nhu cầu một bộ dữ liệu lớn để phục vụ nghiên cứu.

Nghiên cứu và sử dụng phương pháp object detection để tăng khả năng học cũng như dự đoán của mô hình. Đồng thời thử nghiệm và cài đặt thêm những mô hình deep learning cũng như machine learning khác.

Phát triển bài toán, mở rộng các nhãn đầu ra không chỉ là nhận diện các loại rau củ mà còn đưa ra các thông tin về khối lượng cũng như giá tiền của loại rau củ đưa vào máy quét.

VI. KẾT LUẬN

A. Kết quả đạt được

Đồ án đạt mục tiêu đã được đề ra từ đầu, đó là xây dựng được một mô hình cho kết quả đạt độ chính xác khoảng 70% (SVC đạt 83.44%)

Tìm hiểu, xây dựng được bộ dữ liệu VID bao gồm các hình ảnh của 9 loại rau củ, tìm hiểu và sử dụng được một số phương pháp tiền xử lý ảnh có thể ứng dụng trong các bài toán tương tự.

Tìm hiểu và tiếp cận được các phương pháp máy học cho bài toán phân lớp đa lớp. Có cái nhìn cơ bản tổng thể về Machine Learning.

B. Khó khăn gặp phải

Thiếu kinh nghiệm giải quyết các bài toán xử lý hình ảnh nói riêng và các bài toán Machine Learning nói chung dẫn đến việc gặp nhiều khó khăn trong quá trình thu thập cũng như xử lý dữ liệu. Phải thay đổi, chỉnh sửa bộ dữ liệu rất nhiều lần.

Hạn chế về kiến thức nền khiến cho việc tiếp cận các phương pháp xử lý ảnh cũng như tinh chỉnh mô hình trong python tương đối khó khăn.

C. Tổng kết đề tài

Nhóm chúng em xin cảm ơn cô Nguyễn Lưu Thùy Ngân, thầy Dương Ngọc Hào, thầy Lưu Thanh Sơn và thầy Phạm Huỳnh Phúc đã tạo điều kiện để chúng em được tiếp cận, nghiên cứu về đề tài khá thực tế và hữu ích đối với các GenZ như chúng em.

VII. TÀI LIỆU THAM KHẢO

- [1] Hands On Machine Learning with Scikit Learn and TensorFlow (Aurélien Géron)
- [2] Slide bài giảng môn Học máy thống kê trên Courses.uit.edu.vn
- [3] Thư viện phần mềm mã nguồn mở: <https://www.tensorflow.org>
- [4] <https://scikit-learn.org/stable/>
- [5] <https://pillow.readthedocs.io/en/stable/>
- [6] <https://seaborn.pydata.org/>
- [7] <https://matplotlib.org/>
- [8] Fruit and Vegetable Identification Using Machine Learning for Retail Applications (Frida Femling – Chalmers University of Technology) (https://www.researchgate.net/publication/328475022_Fruit_and_Vegetable_Identification_Using_Machine_Learning_for_Retail_Applications)
- [9] Dungnn15, “Xử Lý Ảnh Cơ Bản Với OpenCV Trong Python (P1)”, ngày đăng 15/06/2020, truy cập lần cuối 23/12/2021. (<https://codelearn.io/sharing/xu-ly-anh-voi-opencv-phan-1>)
- [10] Vu Tung Minh, “Beginner Cần Biết: Top 30 Thư Viện Python Tốt Nhất (Phần 1)”, ngày đăng 17/08/2020, truy cập lần cuối 28/12/2021. (<https://codelearn.io/sharing/top-30-libraries-packages-4-beginner-p1>)
- [11] Vu Tung Minh, “Beginner Cần Biết: Top 30 Thư Viện Python Tốt Nhất (Phần 1)”, ngày đăng 17/08/2020, truy cập lần cuối 28/12/2021. (<https://codelearn.io/sharing/top-30-libraries-packages-4-beginner-p1>)
- [12] How to train YOLO v3, v4 for custom objects detection | using colab free GPU <https://www.youtube.com/watch?v=hTCmL3S4Obw>
- [13] Histogram of Oriented Gradients explained using OpenCV cập nhật gần nhất 20/06/2020, lần cuối truy cập 23/12/2021. [Histogram of Oriented Gradients explained using OpenCV](https://www.learnopencv.com/histogram-of-oriented-gradients-explained-using-opencv/) ([learnopencv.com](https://www.learnopencv.com/histogram-of-oriented-gradients-explained-using-opencv/))
- [14] Thuật toán HOG (Histogram of oriented gradient) cập nhật gần nhất 22/11/2019. <https://phamdinhhkhanh.github.io/2019/11/22/HOG.html>
- [15] Các bài viết liên quan tới Học máy thống kê. <https://ndquy.github.io/categories/machine-learning/>
- [16] An Agreement Measure for Determining Inter-Annotator Reliability of Human Judgements on Affective Text (Plaban Kr. Bhowmick, Pabitra Mitra, Anupam Basu) [An Agreement Measure for Determining Inter-Annotator Reliability of Human Judgements on Affective Text](https://www.aclanthology.org/) ([aclanthology.org](https://www.aclanthology.org/))

Bảng phân công công việc

Họ và tên	MSSV	Nhiệm vụ
Đặng Thị Thúy Hồng	20520523	Hỗ trợ làm slide, Thuyết trình, Làm báo cáo word, Hỗ trợ tinh chỉnh mô hình, Xây dựng bộ dữ liệu.
Ngô Thị Phúc	20521765	Làm slide, Làm báo cáo word, Thuyết trình, Hỗ trợ tinh chỉnh mô hình, Xây dựng bộ dữ liệu.
Nguyễn Thị Nguyệt	20521689	Hỗ trợ làm slide, Làm báo cáo word, Thuyết trình, Viết code chạy model, Xây dựng bộ dữ liệu.