Xây dựng tập dữ liệu nhận diện rau củ

Đặng Thị Thúy Hồng[1,2,3], Ngô Thị Phúc[1,2,3], Nguyễn Thị Nguyệt[1,2,3], Nguyễn Thiện Thuật[1,2,3], Nguyễn Gia Tuấn Anh[1,2,4], and Lưu Thanh Sơn[1,2,5]

[1] University of Information Technology, Ho Chi Minh City, Vietnam
[2] Vietnam National University, Ho Chi Minh City, Vietnam
[3] {20520523, 20521765, 20521689, 20521998}@uit.edu.vn
[4] anhngt@uit.edu.vn
[5] sonlt@uit.edu.vn

Tóm tắt nội dung Hiện nay, ở nước ta nói riêng và ở các nước đang phát triển có nền nông nghiệp là một trong các ngành sản xuất chủ yếu, quá trình thu hoạch, phân loại và đánh giá chất lượng các loại sản phẩm nông nghiệp, đặc biệt là các rau củ, chủ yếu còn phải thực hiện bằng các phương pháp thủ công. Đây là công việc không quá khó, nhưng tiêu tốn nhiều thời gian, công sức của con người và là rào cản đối với mở rộng phát triển quy mô nông nghiệp. Để đáp ứng nhu cầu về dữ liệu cho việc nghiên cứu bài toán nhận diện rau củ, nhóm đã thu thập và xây dựng bộ dữ liệu ảnh rau củ, cung cấp cho cộng đồng nghiên cứu nguồn dữ liệu đủ đáp ứng những yêu cầu cho việc nghiên cứu các mô hình học máy để giải quyết bài toán nhận diện rau củ.

Keywords: Nhận diện rau củ, rau củ, VID.

1 Giới thiệu

Trong ăn uống hằng ngày, rau củ cung cấp cho cơ thể nhiều chất hoạt tính sinh học, là nguồn cung cấp các chất dinh dưỡng bao gồm kali, chất xơ, folate (acid folic), vitamin A, vitamin C. Một chế độ ăn uống có nhiều rau củ trong khẩu phần ăn có thể làm giảm nguy cơ các bệnh tim mạch bao gồm bệnh đau tim và đột quỵ, đặc biệt có thể ngăn ngừa một số loại bệnh ung thư nhất định. Chính vì vậy rau củ chiếm một phần quan trọng trong chế độ ăn của chúng ta.

Hiện nay có rất nhiều người đặc biệt là gen Z đang theo đuổi chế độ ăn sạch, có nghĩa là tập trung ăn nhiều thực phẩm tươi, sạch như rau củ chẳng hạn. Nhưng khi chúng ta đi chợ, siêu thị. Để lựa mua rau củ thì rất ít hoặc khó để phân biệt được sự khác nhau của các loại rau củ này, hay thậm chí là không biết tên chúng là gì, mặc dù xuất hiện nhiều trong khẩu phần ăn. Do đó, nhóm đã quyết định chọn đề tài "Xây dựng tập dữ liệu nhận diện rau củ".

Mô hình này sẽ huấn luyện tập dữ liệu chứa các hình ảnh của các loại rau, củ có sẵn. Sau đó máy sẽ nhận biết những hình ảnh rau củ được đưa vào, cùng với đó đưa ra tên loại rau củ đó

- **Input**: Một ảnh chụp rau củ
- **Output**: 1 trong 15 nhãn rau củ của bộ dữ liệu.

VD:

– **Input**: Ảnh rau củ

Hình 1.

– **Output**: Tên: Papaya

2 Bộ dữ liệu

2.1 Nguồn dữ liệu và mô tả

Trong quá trình nghiên cứu, chúng tôi xây dựng bộ dữ liệu 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 Sau khi xử lý bộ dữ liệu thô, chúng tôi xử lý thủ công và tạo ra bộ dữ liệu bao gồm 2000 điểm dữ liệu với 15 loại rau củ tương ứng với 15 nhãn được miêu tả trong Bảng 1

Chúng tôi đã tiến hành chia bộ dữ liệu thành 2 tập là tập huấn luyện (train set) và tập kiểm thử (test set) với tỉ lệ lần lượt là 8:2.

Link bộ dữ liệu: BỘ DỮ LIỆU VID.

2.2 Bộ nhãn, quá trình gán nhãn

Về bộ nhãn, bộ dữ liệu của chúng tôi bao gồm 15 nhãn tương ứng với các thuộc tính, mỗi nhãn là một loại rau củ được mô tả như Bảng 1.

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. Như sau:

Bảng 1. Bảng chi tiết các thuộc tính của bộ dữ liệu VID.

Tên thuộc tính	Định nghĩa
Bean	Đậu
Bitter Gourd	Khổ qua
Botter Gourd	Bầu
Broccoli	Bông cải xanh
Cabbage	Bắp cải
Carrot	Cà rốt
Cauliflower	Bông cải trắng
Cucumber	Dưa leo
Papaya	Đu đủ
Potato	Khoai tây
Pumpkin	Bí ngô
Radish	Cải củ đen.
Tomato	Cà chua

– Ảnh chứa 1 trong 15 loại rau củ được đề cập trong Bảng 1. Tối thiểu là 1 loại, tối đa là 15 loại
– Chấp nhận ảnh của các loại rau củ bị cắt mất một phần, bị mờ trong khoảng chấp nhận được (< 0.4) hoặc loại rau củ đó bị cắt ra thành từng phần nhưng vẫn nhận dạng được
– 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
– Trong khi gán nhãn thì phải loại bỏ cảm xúc cá nhân

Khi gán nhãn, phải nhận được sự đồng thuận từ 3/4 thành viên trở lên, nếu không điểm dữ liệu sẽ được bỏ đi và thay bằng một hình ảnh khác.

Hướng dẫn gán nhãn: HƯỚNG DẪN GÁN NHÃN.

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:

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

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

Hình 2. Công thức Cohen của Kappa

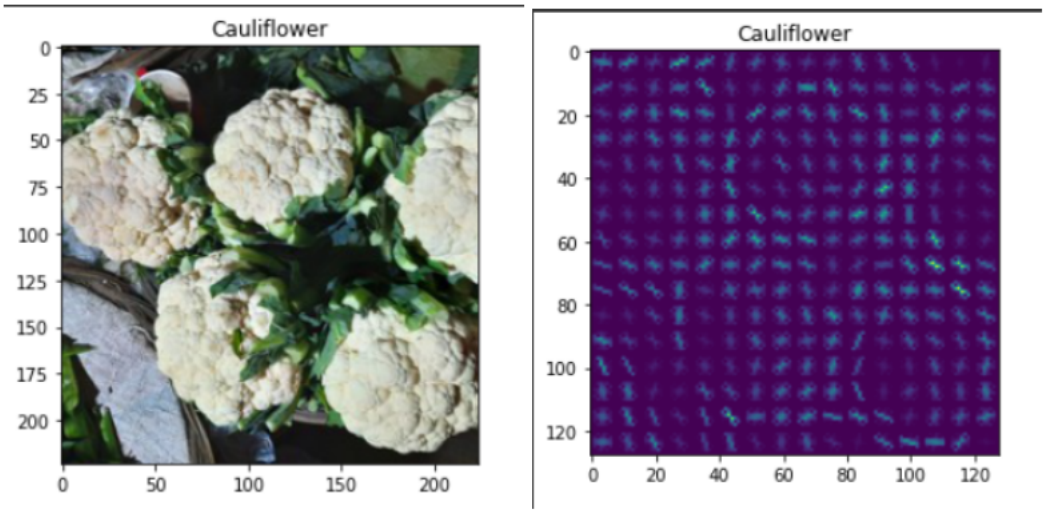
Trong trường hợp thử với tập train 1200 bức ảnh, chúng tôi nhận được kết quả rất khả quan, khoảng 0.75. Điều này chứng tỏ tất cả người gán đã có mức độ đồng thuận trong việc gán nhãn là rất cao. Chúng tôi tiếp tục thực hiện đề tài với quy định gán nhãn đã thống nhất.

2.3 Phương pháp 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 Equalzition 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 hình ảnh

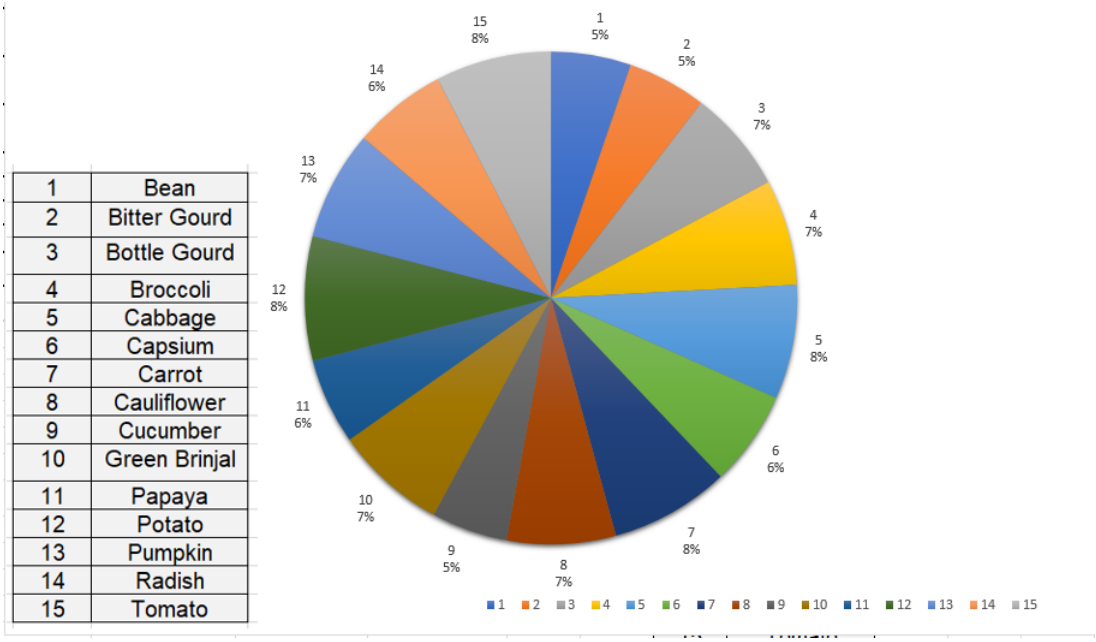


Hình 3. Hình ảnh trước và sau khi xử lý

Source code xử lý ảnh: SOURCE CODE XỬ LÝ ẢNH.

2.4 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 4. Phân bố số lượng rau củ

- 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 luyệ
- Tập dữ liệu có kích thước nhỏ khi so với các công trình khác
- Các nhãn có số lượng điểm dữ liệu chênh lệch nhau. Tập train và tập test chia theo đúng tỉ lệ 8:2
- Số lượng hình ảnh rau củ của các thuộc tính phân bố không đều. Các hình ảnh thuộc các tập Cabbage, Carrot, Potato, Tomato chiếm số lượng cao nhất; mặc khác các hình thuộc các tập Bean, Bitter Gourd, Cucumber chiến số lượng ít. Điều này có thể dẫn đến sự nhầm lẫn cho các phương pháp học máy khi dự đoán hình ảnh để phân loại rau củ (ví dụ các mô hình học máy có thể gây nhầm lẫn giữa hình ảnh thuộc tập Bean và hình ảnh thuộc tập Cabbage)

2.5 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:

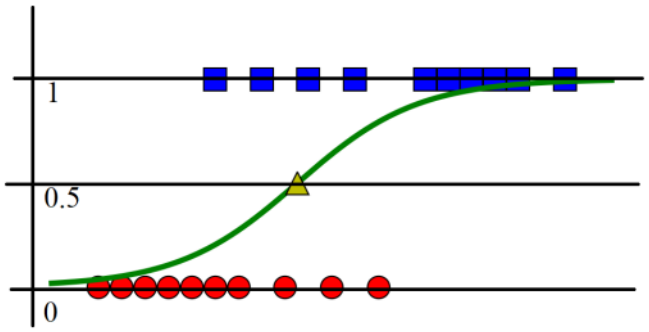
- Việc thu thập hình ảnh từ nhiều nguồn trên internet không đảm bảo sự đồng đều về chất lượng, kích thước, chủ thể, khiến quá trình tiền xử lý, làm sạch dữ liệu trở nên khó khăn hơn.

– Có một số lượng lớn dữ liệu bị trùng 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 15 loại rau củ trong đề tài này.

3 Đề xuất phương pháp máy học

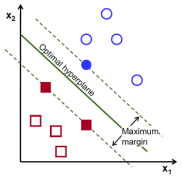
3.1 Mô hình máy học

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 rau củ trong hình ảnh. Đầu ra dự đoán của Logistic Regression có dạng viết chung là: f(x) = (wTx) Trong đó được gọi là logistic function.



Hình 5. Mô hình Logistic Regression

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.

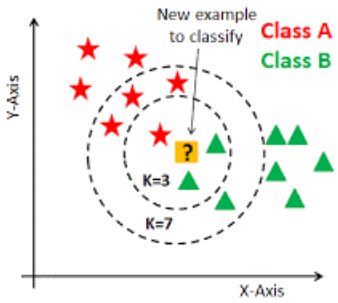


Hình 6. Mô hình Support Vector Machine

Hàm quyết định phân dữ liệu vào lớp thứ i là: fi(x) = wTi x + bi 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(x), giá trị ngưỡng b xác định khoảng cách giữa siêu phẳng và gốc.

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



Hình 7. Mô hình K-Nearest Neighbors

3.2 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 kì với nhiều yêu cầu khác nhau

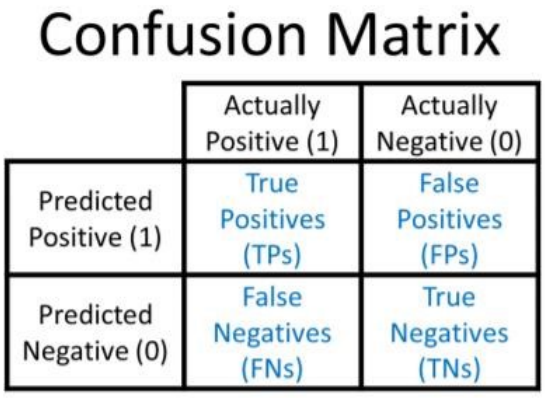
- Matplotlib: là thư viện dùng để vẽ đồ thị 2D
- Seaborn: là một thư viện trực quan hóa dữ liệu.
- PIL: có thể mở, lưu, xử lý đặc điểm hình ảnh với nhiều định dạng ảnh khác nhau.
- Skimage: là một thư viện xử lý ảnh nguồn mở bao gồm các thuật toán để phân đoạn, thao tác không gian màu, phân tích, lọc, 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.

3.3 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.



Hình 8. Confusion Matrix

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

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

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:

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

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

Hình 9.

Bài toán phân loại rau củ là phân lớp đa lớp 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.

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.

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

Hình 10.

Đề tài của chúng tôi dùng thêm Accuracy vì đây là độ đo rất phù hợp với bài toán phân lớp đa nhãn.

4 Kết luận

Nhìn chung, bộ dữ liệu VID đa dạng về các nhãn nhưng chưa có sự đa dạng về hình ảnh của mỗi nhãn. Ảnh của rau củ bị mờ, bị cắt một phần hoặc thậm chí rau củ bị biến dạng, bị cắt xẻ thành nhiều phần khiến đội ngũ gán nhãn bỏ khá nhiều công sức trong lúc gán. Với số lượng ảnh như hiện tại được sử dụng làm nhãn, trong thời gian sắp tới nhóm em sẽ mở rộng bộ dữ liệu để thỏa mãn được nhu cầu về một bộ dữ liệu lớn để phục vụ việc nghiên cứu.

Về phương diện ứng dụng, các mô hình học máy khi được huấn luyện trên bộ dữ liệu này vẫn chưa đủ khả năng nhận diện được hình ảnh trong thực tế do các yếu tố về màu sắc, hình dáng, các đặc điểm của rau củ được đề cập trong Hướng dẫn gán nhãn của bộ dữ liệu đã được làm đơn giản hóa nhiều so với thực tế. Trong thời gian tới nhóm sẽ tiếp tục nghiên cứu để có thể tạo ra một bộ dữ liệu đảm bảo được đầy đủ các yếu tố về hình ảnh của các loại rau củ nhằm đánh giá một cách toàn diện mọi khía cạnh của mô hình nhận diện.

Tổng kết đề tài, nhóm chúng em xin cảm ơn thầy Lưu Thanh Sơn và thầy Nguyễn Gia Tuấn Anh nói riêng và khoa Khoa học và Kỹ thuật Thông tin nói chung đã tạo điều kiện cho chúng em được tiếp cận, nghiên cứu về đề tài khá thực tế và hữu ích đối với các GenZ như chúng em.

Tài liệu

1. Slide bài giảng môn học trên Courses.uit.edu.vn
2. Hands On Machine Learning with Scikit Learn and TensorFlow (Aurélien Géron)
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. 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
9. 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
10. How to train YOLO v3, v4 for custom objects detection | using colab free GPU. https://www.youtube.com/watch?v=hTCmL3S4Obw
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. Histogram of Oriented Gradients explained using OpenCV cập nhật gần nhất 20/06/2020, lần cuối truy cập 23/12/2021. (learnopencv.com)
13. Thuật toán HOG (Histrogram of oriented gradient) cập nhật gần nhất 22/11/2019. https://phamdinhkhanh.github.io/2019/11/22/HOG.html
14. Các bài viết liên quan tới Học máy thống kê. https://ndquy.github.io/categories/machine-learning/