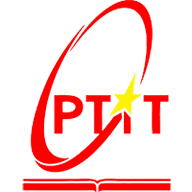
**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**CƠ SỞ TẠI THÀNH PHỐ HỒ CHÍ MINH**

****

**XỬ LÝ ẢNH**

**ĐỀ TÀI: NHẬN DIỆN BỆNH CỦA LÁ CÂY**

**Giảng viên : ThS. HUỲNH TRUNG TRỤ**

**Nhóm 23 : VŨ HOÀNG MẠNH – N20DCCN001**

**Lớp : D20CQCNPM01-N**

**TP. Hồ Chí Minh ngày 21 tháng 12 năm 2023**

**LỜI CẢM ƠN**

Đề tài “Nhận diện bệnh của lá cây” là nội dung nhóm em chọn để nghiên cứu và làm đồ án kết thúc môn Xử lý ảnh.

Để hoàn thành quá trình nghiên cứu và hoàn thiện đồ án này, nhóm em xin chân thành cảm ơn sâu sắc đến Thầy Huỳnh Trung Trụ. Thầy đã chỉ dạy và hướng dẫn nhóm em trong quá trình nghiên cứu để nhóm hoàn thiện đồ án này.

Mặc dù đã cố gắng hoàn thành đồ án với tất cả sự nỗ lực nhưng chắc chắn không tránh khỏi những thiếu sót, do thời gian tìm hiểu và trình độ hiểu biết còn hạn hẹp. Nhóm em rất mong được sự tận tình chỉ bảo của quý thầy cô.

TP. Hồ Chí Minh, ngày 21 tháng 12 năm 2023

Sinh viên thực hiện

Vũ Hoàng Mạnh

1. **GIỚI THIỆU**

Việc duy trì sức khỏe của cây trồng là một phần quan trọng trong nền nông nghiệp. Tuy nhiên, nhận diện và điều trị các bệnh của cây thường đòi hỏi kiểm tra thường xuyên và sự can thiệp kịp thời của nhân viên nông nghiệp. Đây không chỉ là một quá trình tốn kém về thời gian và nhân lực mà còn đôi khi không hiệu quả.

Với mục tiêu giải quyết vấn đề này, nhóm em đã lựa chọn đề tài “Nhận diện bệnh của lá cây”. Đề tài này được lựa chọn để tận dụng sức mạnh của trí tuệ nhân tạo, đặc biệt là machine learning và computer vision, để xây dựng một hệ thống nhận diện bệnh của lá cây dựa trên hình ảnh.

Mục tiêu chính của đề tài là phát triển một hệ thống tự động có khả năng nhận diện các bệnh thường gặp trên lá cây thông qua việc phân loại và đưa ra dự đoán chính xác về tình trạng sức khỏe của cây. Qua đó, giúp nhân viên nông nghiệp xác định và can thiệp kịp thời để bảo vệ và cải thiện sức khỏe của cây trồng một cách hiệu quả.

Nhóm em đã sử dụng các mô hình học máy và công nghệ xử lý ảnh như Xception và DenseNet để xây dựng một hệ thống phân loại bệnh của lá cây. Bằng cách huấn luyện mô hình trên tập dữ liệu chứa các hình ảnh của lá cây bị nhiễm bệnh và lá cây khỏe mạnh, nhóm em đã tạo ra một mô hình có khả năng nhận diện và phân loại các bệnh phổ biến trên lá cây với độ chính xác cao.

1. **CƠ SỞ LÝ THUYẾT**
2. **Trí tuệ nhân tạo**
3. *Định nghĩa*

Trí tuệ nhân tạo là một lĩnh vực của khoa học máy tính tập trung vào việc tạo ra các hệ thống hoạt động như con người, có khả năng tự học, tự thích nghi và thực hiện các nhiệm vụ thông minh mà trước đó chỉ có con người mới có khả năng thực hiện.

1. *Lý thuyết và ứng dụng*

Học Máy (Machine Learning): Đây là một phần quan trọng của trí tuệ nhân tạo, chủ yếu tập trung vào việc xây dựng các mô hình có khả năng học từ dữ liệu mà không cần phải được lập trình cụ thể. Các thuật toán học máy giúp mô hình hóa dữ liệu và tạo ra các dự đoán hoặc quyết định dựa trên dữ liệu đó.

Xử lý ảnh và Computer Vision: Lĩnh vực này tập trung vào việc phân tích và hiểu về hình ảnh và video, từ việc xử lý cơ bản như lấy thông tin từ hình ảnh đến việc nhận diện vật thể, phân loại, và nhận diện khuôn mặt hoặc đối tượng.

Deep Learning: Là một phương pháp của học máy, đặc biệt chú trọng vào việc sử dụng các mạng nơ-ron sâu để tự học các đặc trưng từ dữ liệu, dẫn đến khả năng nhận diện và phân loại chính xác cao trong nhiều tác vụ.

Ứng dụng trong đề tài: Trong đề tài của nhóm em, nhóm đã sử dụng trí tuệ nhân tạo, đặc biệt là machine learning và computer vision, để xây dựng một hệ thống có khả năng nhận diện và phân loại bệnh của lá cây từ hình ảnh, áp dụng các nguyên lý và phương pháp của AI để giải quyết vấn đề cụ thể trong lĩnh vực nông nghiệp.

1. **Các thuật toán**
2. *Học máy (Machine Learning)*

Deep Learning: Trong đề tài của nhóm em, nhóm đã sử dụng mô hình Deep Learning như Xception và DenseNet. Đây là những mô hình được huấn luyện sẵn với khả năng nhận diện và phân loại hình ảnh, thường được sử dụng trong các bài toán Computer Vision như việc nhận diện bệnh trên lá cây.

1. *Xử lý ảnh và Computer Vision*

Convolutional Neural Networks (CNNs): Đây là một loại mạng nơ-ron sử dụng phép tích chập, được thiết kế để xử lý và học từ dữ liệu hình ảnh. Trong đề tài của nhóm em, Xception và DenseNet là các mô hình CNNs phổ biến được sử dụng để nhận diện bệnh trên lá cây từ hình ảnh.

1. *Xử lý dữ liệu*

ImageDataGenerator: Đây là một công cụ trong thư viện Keras giúp tạo ra các dữ liệu đầu vào cho việc huấn luyện mô hình. Nó có thể thực hiện các phép biến đổi và tăng cường dữ liệu hình ảnh, giúp tăng độ đa dạng của dữ liệu và cải thiện hiệu suất của mô hình.

1. *Huấn Luyện Mô Hình*

Optimizer Adam: Là một thuật toán tối ưu hóa thông dụng trong huấn luyện mạng nơ-ron, giúp cải thiện việc hội tụ của mô hình và tối ưu hóa hàm mất mát.

Categorical Crossentropy: Là một hàm mất mát thường được sử dụng trong bài toán phân loại nhiều lớp, giúp đánh giá hiệu suất của mô hình.

1. **THIẾT KẾ ỨNG DỤNG**
2. **Tổng quan về kiến trúc của hệ thống**
3. *Ứng Dụng Streamlit*

Streamlit App: Đây là một framework Python cung cấp môi trường dễ sử dụng để xây dựng và triển khai các ứng dụng web cho các dự án Data Science và Machine Learning. Trong đề tài của nhóm em, ứng dụng Streamlit được sử dụng để tạo giao diện người dùng cho việc tải lên hình ảnh, chẩn đoán và hiển thị kết quả của việc nhận diện bệnh trên lá cây.

1. *Mô Hình Nhận Diện Bệnh Trên Lá Cây*

Xception và DenseNet Ensemble Model: Trong kiến trúc của mô hình, nhóm em đã sử dụng hai mô hình mạng nơ-ron sâu (Deep Learning) được huấn luyện trước (pre-trained) là Xception và DenseNet. Nhóm đã kết hợp hai mô hình này để tạo thành một mô hình ensemble, tận dụng sức mạnh của từng mô hình để cải thiện độ chính xác và độ tin cậy trong việc nhận diện các bệnh trên lá cây.

1. *Tiền Xử Lý Dữ Liệu*

ImageDataGenerator và Pillow: Nhóm em đã sử dụng thư viện ImageDataGenerator trong Keras để tiền xử lý dữ liệu hình ảnh, bao gồm việc thay đổi kích thước, tăng cường dữ liệu và chuẩn hóa. Bên cạnh đó, thư viện Pillow được sử dụng để xử lý và đọc dữ liệu hình ảnh.

1. *Chức Năng Của Ứng Dụng*

Upload và Hiển thị hình ảnh: Ứng dụng cho phép người dùng tải lên hình ảnh của lá cây bị nhiễm bệnh. Sau khi tải lên, hình ảnh sẽ được hiển thị trên giao diện người dùng để người dùng có thể xem trước trước khi thực hiện chẩn đoán.

Chẩn đoán và hiển thị kết quả: Sau khi tiền xử lý hình ảnh, mô hình ensemble (Xception và DenseNet) sẽ được áp dụng để chẩn đoán và nhận diện bệnh trên lá cây từ hình ảnh đã tải lên. Kết quả của quá trình chẩn đoán sẽ được hiển thị cho người dùng.

1. **Tiền xử lý dữ liệu**
2. *Chuẩn bị*

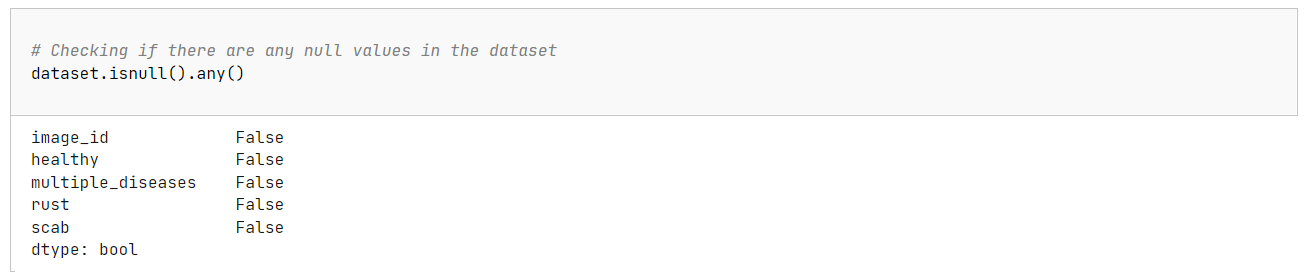
Đọc tập Training data và import các thư viện cần thiết:



1. *Data Exploration and Visualization (Tìm hiểu và mô hình hóa dữ liệu):*

* Data Exploration:

Thực hiện check có giá trị null nào trong dataset không:



Kết quả cho thấy không có giá trị null nào trong các cột: image\_id, healthy, multiple\_diseases, rust, scab.

Thực hiện check kiểu dữ liệu của các cột trong dataset:

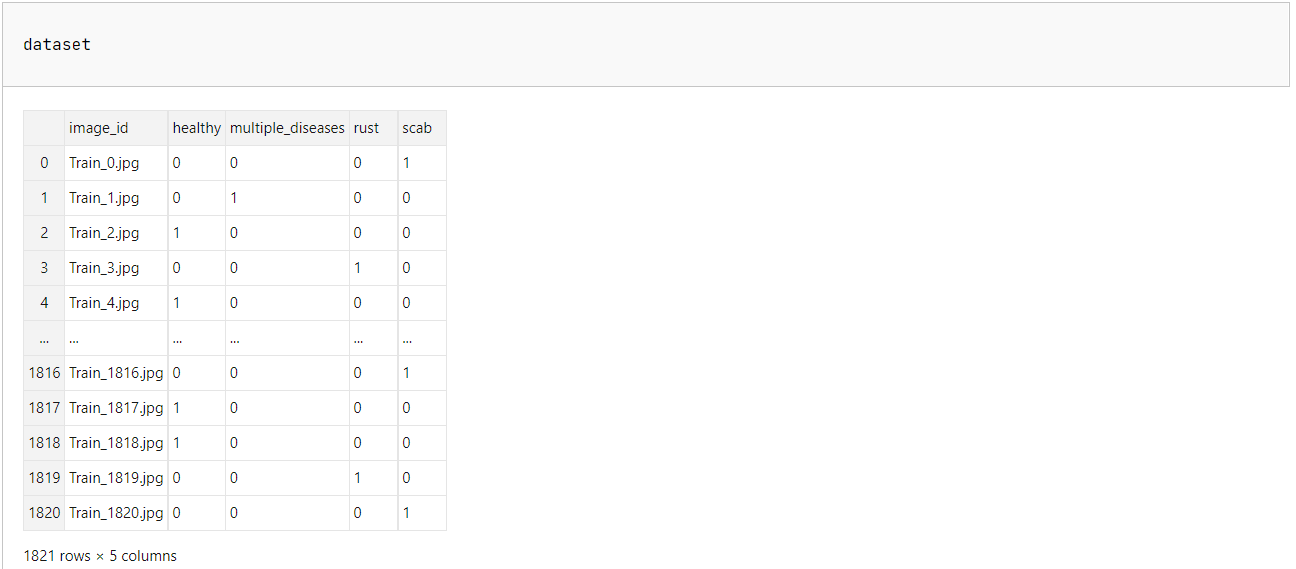


Kết quả: kiểu dữ liệu của image\_id là object (chuỗi), của healthy, multiple\_diseases, rust, scab là int64 (số nguyên).

Cuối cùng là thêm đuôi .jpg cho mỗi image\_id:



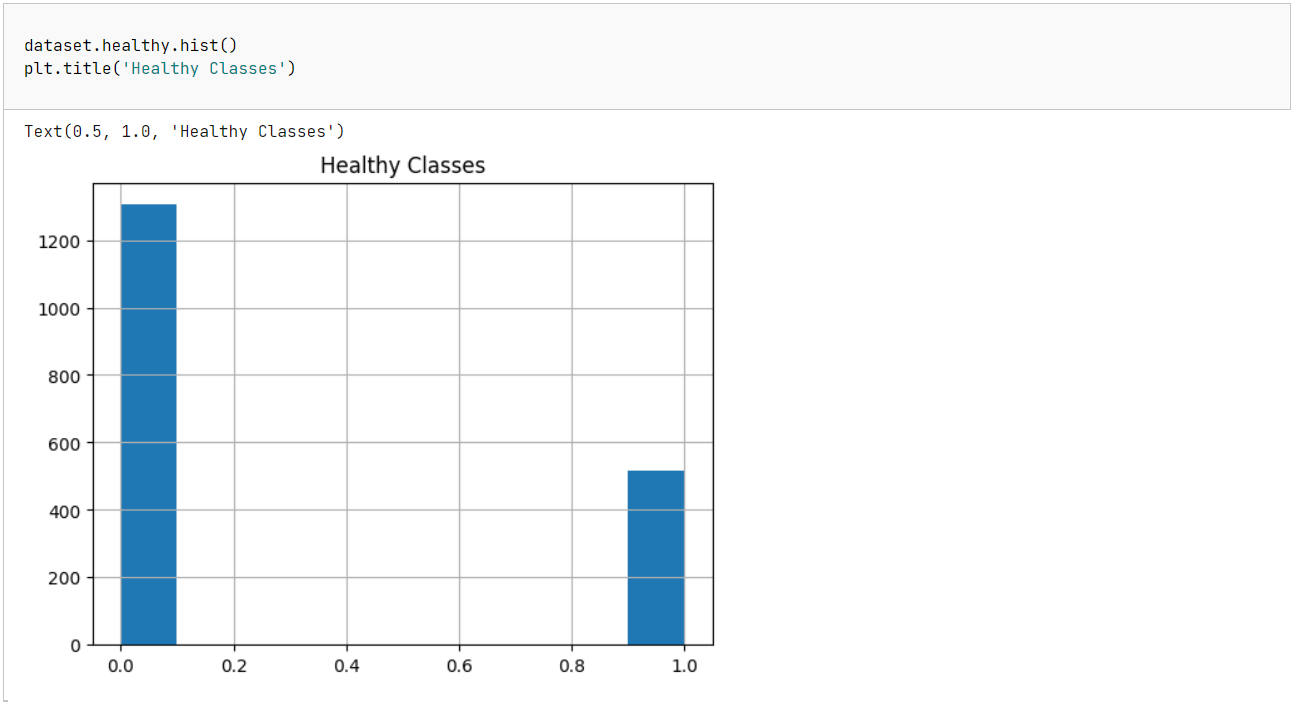
Bảng dataset:



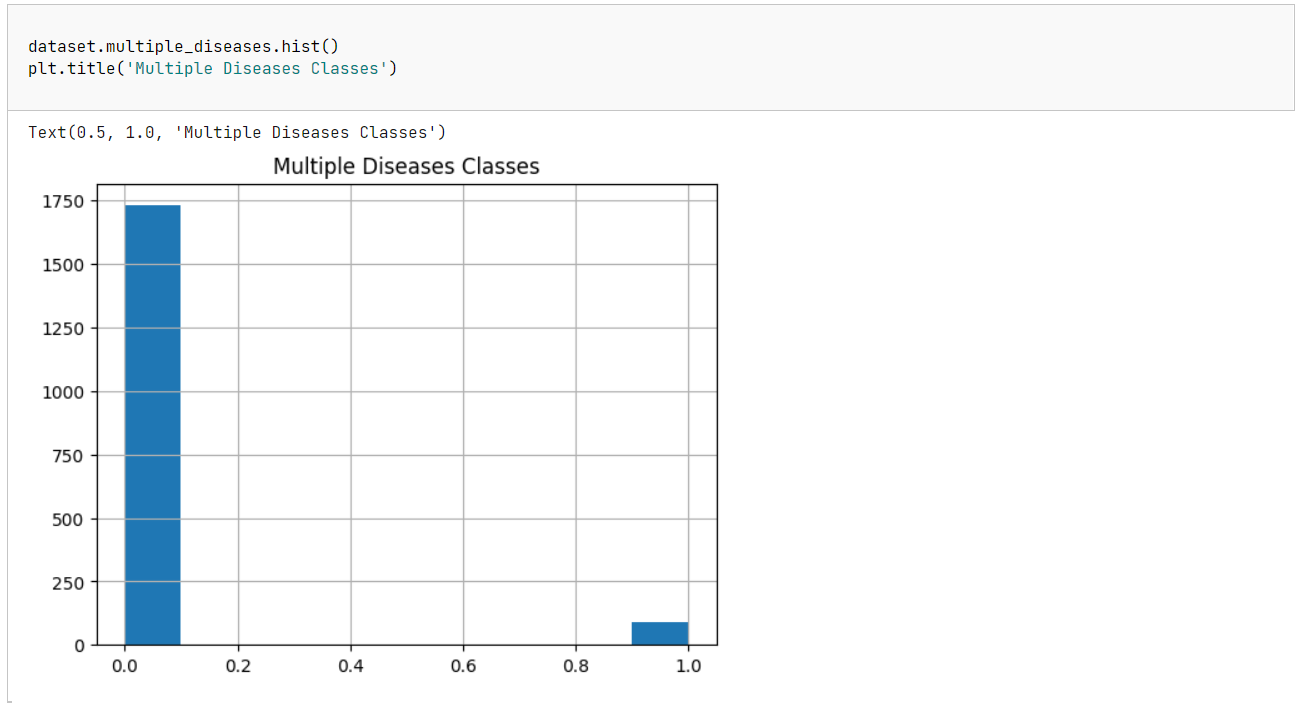
* Data Visualization:

Classes Distribution: Phân phối các lớp trong dataset và vẽ biểu đồ histogram:

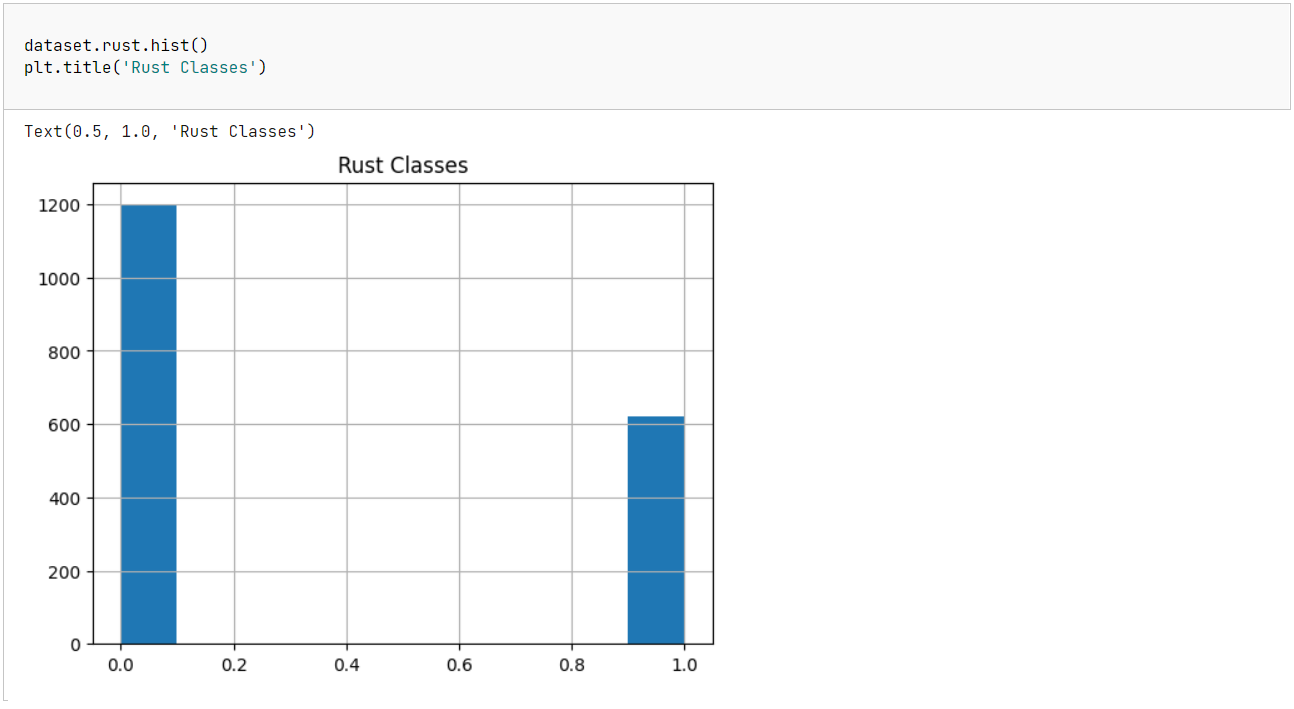
Thực hiện vẽ biểu đồ histogram cho lớp healthy:



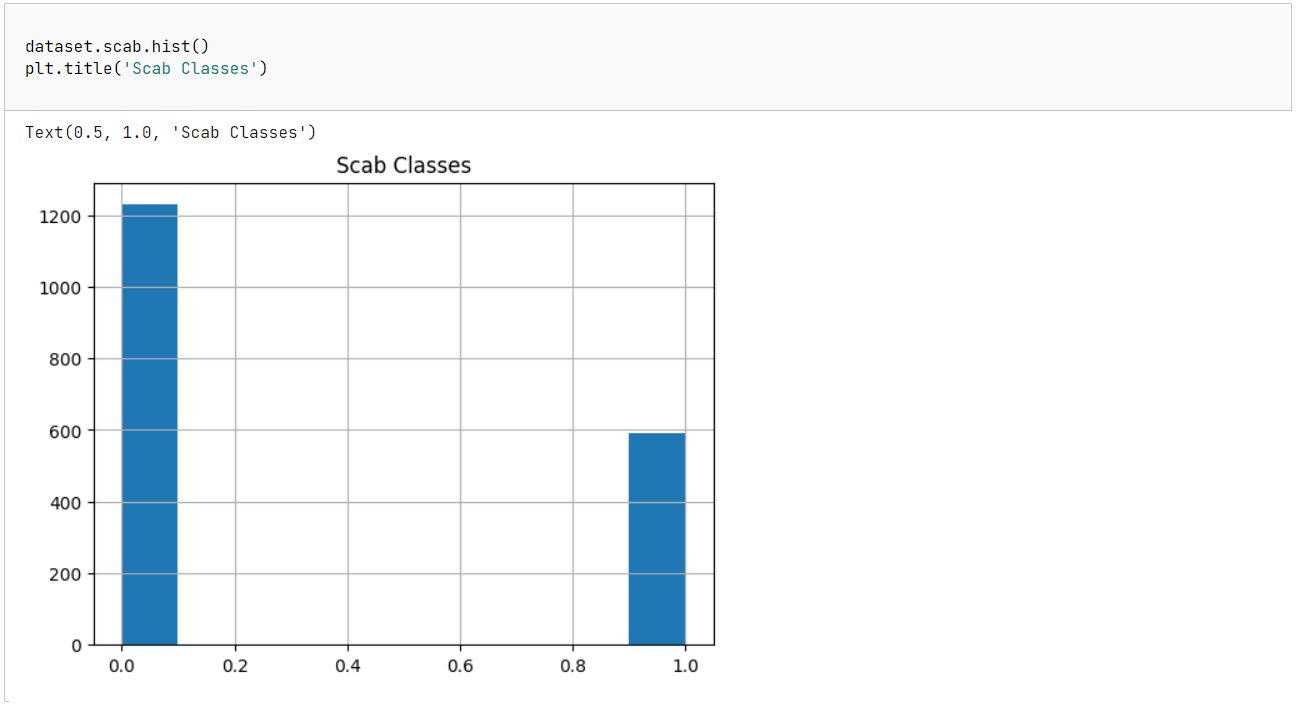
Thực hiện vẽ biểu đồ histogram cho lớp multiple\_diseases:



Thực hiện vẽ biểu đồ histogram cho lớp rust:



Thực hiện vẽ biểu đồ histogram cho lớp scab:

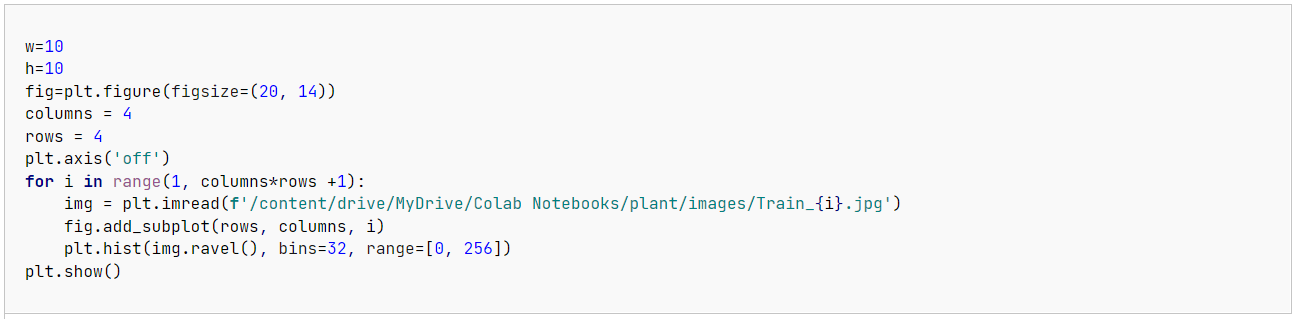


Class Image Visualisation: Thực hiện phân loại ảnh và gán nhãn tương ứng. Ở đây nhóm em đã thực hiện bằng cách: Thiết lập chiều rộng (w = 10) và chiều cao (h = 10), sau đó tạo 1 figure có kích thước 20x14. Sau đó tạo 1 table để hiển thị hình ảnh có kích thước 4x4 rồi đọc ảnh từ link tương ứng với giá trị I, sau đó thêm tiêu đề tương ứng với phân loại ảnh lá cây:

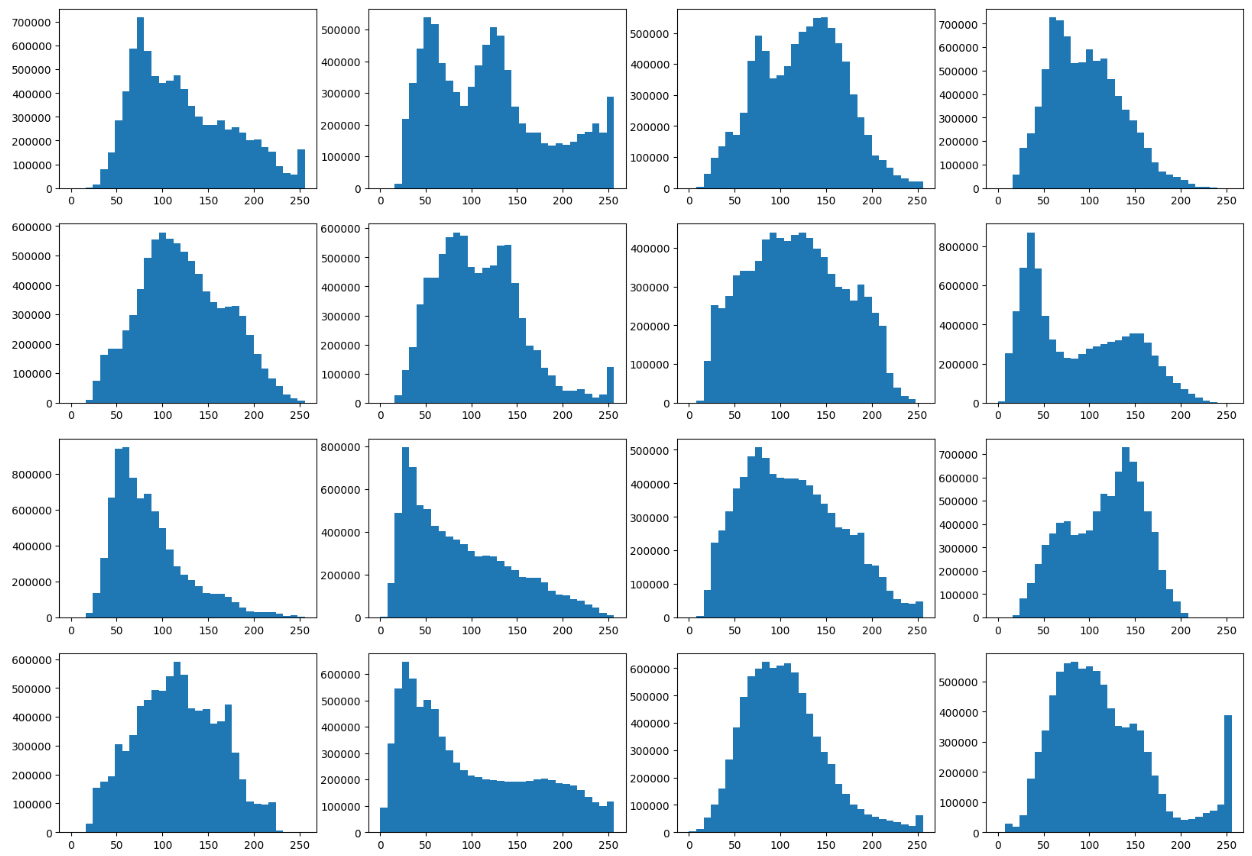


* Image Segmentation: Phân đoạn hình ảnh

Nhóm em đã tạo ra mỗi biểu đồ histogram cho mỗi hình ảnh trong table 4x4 trên bằng cách tính toán histogram cho từng ảnh:

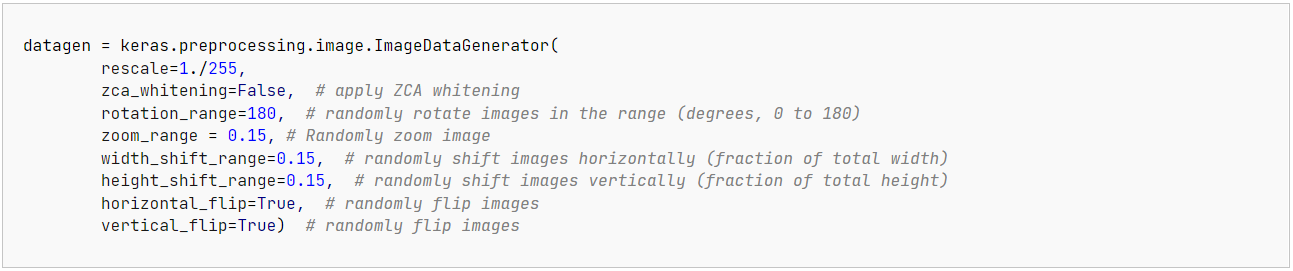


Kết quả thu được là histogram sẽ phân phối các giá trị pixel theo khoảng màu từ 0 đến 256, chia thành 32 bins:

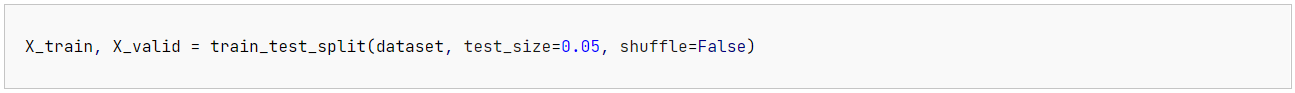


* Making Training Data: Chuẩn bị dữ liệu huấn luyện:

Tạo ImageDataGenerator: Ở đây nhóm em đã sử dụng công cụ ImageDataGenerator của Tensoflow/Keras để tạo ra nhiều phiên bản mới của mỗi ảnh với các phép biến đổi ảnh như: rescale (thay đổi scale của ảnh), rotation (quay ảnh), zoom (phóng to/thu nhỏ), width/height shift (dịch ảnh theo chiều dài/rộng), horizontal/vertical flip (xoay ảnh ngang/dọc ngẫu nhiên) và shear (cắt ảnh):



Chia data thành tập huấn luyện và tập kiểm tra: Ở đây em đã dùng train\_test\_split từ sklearn để chia data thành 2 phần: X\_train (dùng để huấn luyện) và X\_valid (dùng để kiểm tra). test\_size=0.05 chỉ định rằng 5% data sẽ được sử dụng cho việc kiểm tra và shuffle=False để không xáo trộn data.



Making a Tensorflow Dataset: đọc dữ liệu hình ảnh từ dataframe và tạo generator để đưa dữ liệu vào mô hình. Nhóm em đưa 8 ảnh vào mô hình (batch) mỗi lần huấn luyện, sau đó tạo ra generator từ data trong dataframe của datagen đã được tăng cường dữ liệu qua các tham số biến đổi ảnh. Em đang để chế độ class\_mode ở đây là raw nhằm trả về mảng numpy của nhãn không đổi và shuffle=False để dữ liệu không bị xáo trộn sau mỗi lần đưa ảnh.

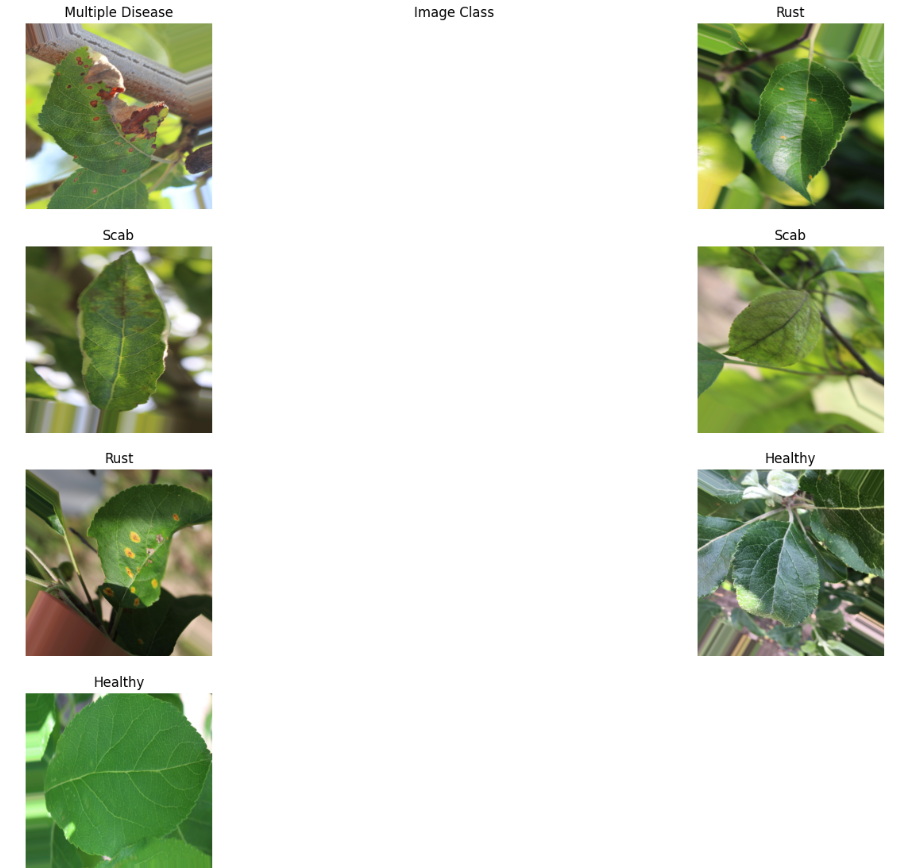


Ở đây kết quả thu được là có 1821 hình ảnh được xác nhận và 92 hình ảnh cho tập validation.

Generator Images Visualisations: hiển thị 1 lát cắt của dữ liệu được tạo ra từ generator để lấy 1 batch hình ảnh và nhãn từ dữ liệu huấn luyện. Ở mỗi vòng lặp sẽ lấy 1 batch gồm 8 ảnh và nhãn tương ứng từ generator, sau đó hiển thị ảnh và đặt tiêu đề cho ảnh tương ứng với nhãn của nó:



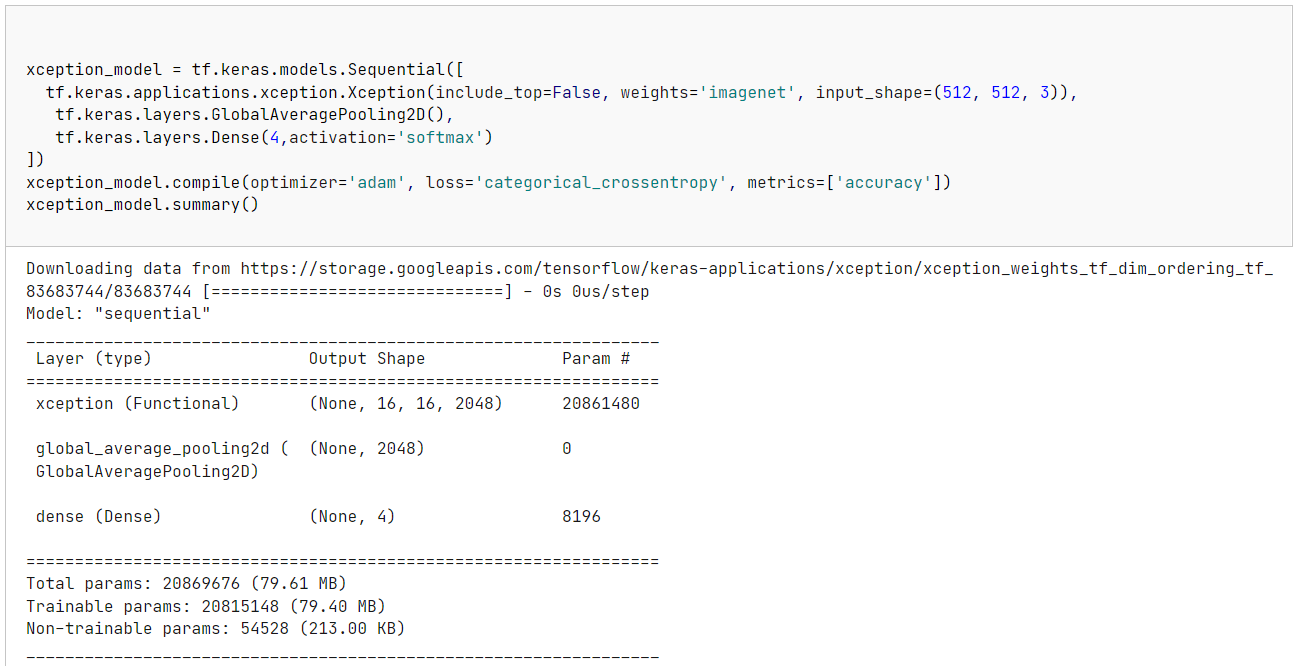
Ví dụ kết quả:



1. **Xây dựng các mô hình**

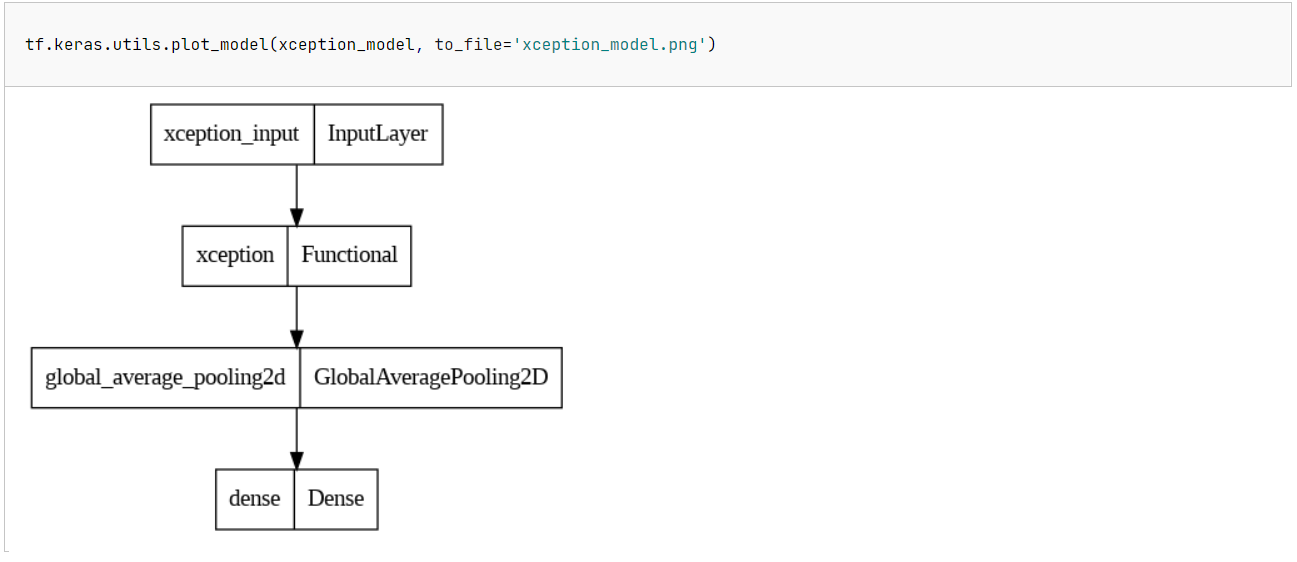
* Making The Models: Xây dựng mô hình bằng Sequential API của Keras để xây dựng một chuỗi các layer mạng nơ-ron.

Xception Model: Nhóm em dùng Xception để làm layer đầu tiên cho mô hình, include\_top=False để loại bỏ các layer fully connected ở đầu mô hình. Sau layer Xception, là layer GlobalAveragePooling2D, giúp chuyển đổi các tensor có kích thước (16, 16, 2048) thành tensor có kích thước (2048), tức là sử dụng trung bình toàn cục trên các feature map. Cuối cùng là Dense với activation function là 'softmax' để chuyển đổi output thành xác suất, từ đó dự đoán ảnh thuộc lớp nào. Có 4 node đại diện cho các lớp (Healthy, Multiple Disease, Rust, Scab).

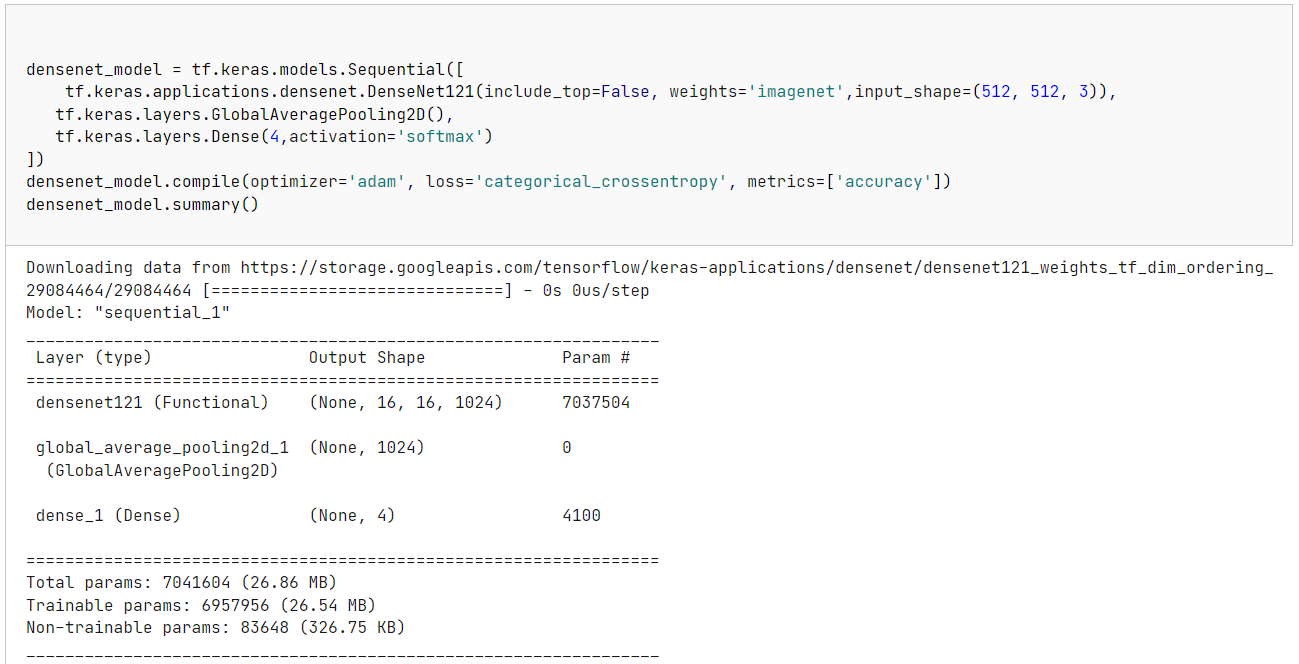


Ở trên là tổng số tham số của mô hình (Total params), số tham số có thể được huấn luyện (Trainable params) và số tham số không cần huấn luyện (Non-trainable params) (trọng số đã được tải từ trọng số đã được huấn luyện trước).

Dưới đây là biểu đồ biểu diễn cấu trúc của mô hình Xception, bao gồm: các layer và các kết nối:



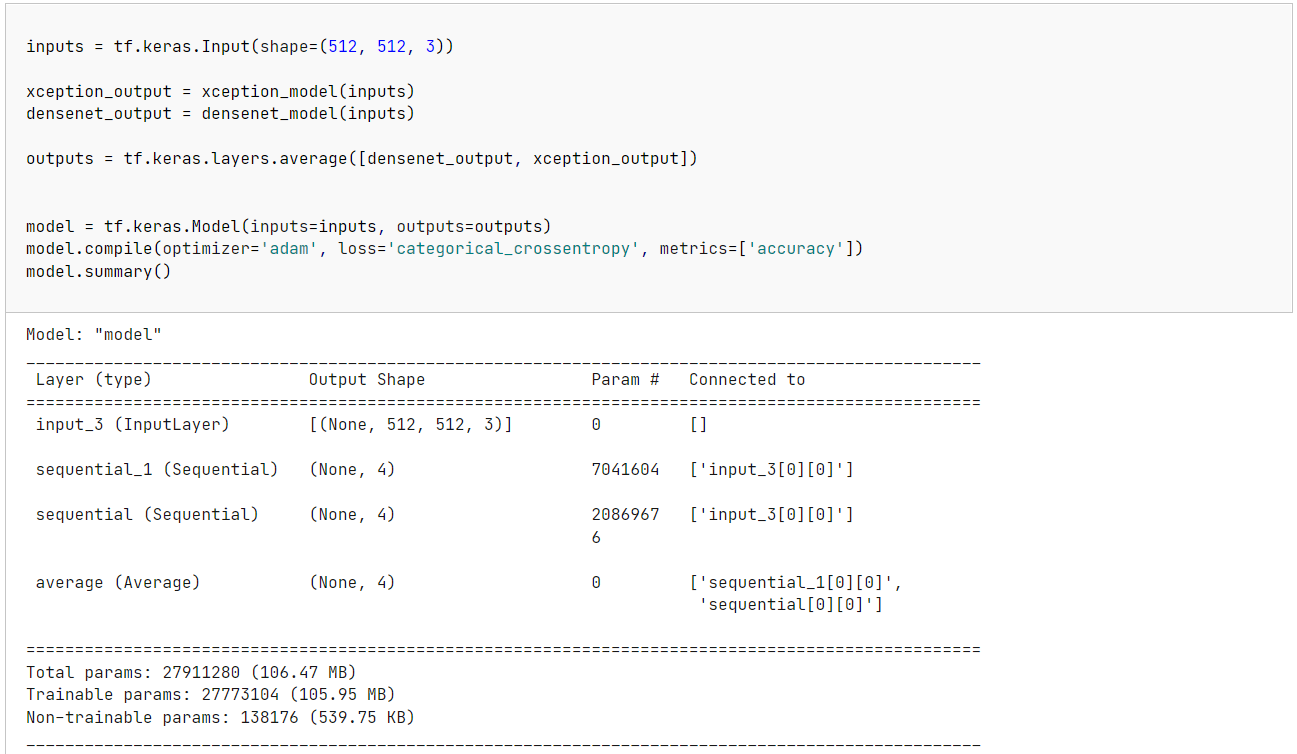
DenseNet121: Mô hình thứ 2 nhóm em sử dụng đó là DenseNet. Sau khi tải mô hình, em thêm một lớp GlobalAveragePooling2D() để chuyển đổi các feature map từ lớp cuối cùng của DenseNet thành vector đặc trưng duy nhất. Cuối cùng, em thêm một lớp Dense với hàm kích hoạt softmax để phân loại ảnh thành 4 lớp khác nhau. Hàm compile cấu hình quá trình huấn luyện với bộ tối ưu hóa adam, hàm loss là categorical cross-entropy và metrics để đánh giá là accuracy (độ chính xác).



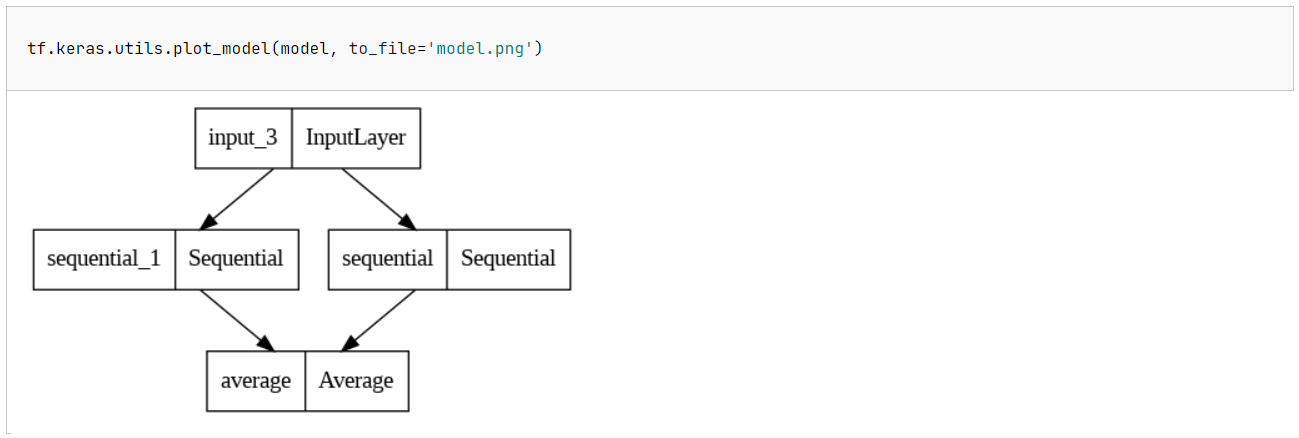
Dưới đây là biểu đồ biểu diễn cấu trúc của mô hình DenseNet121, bao gồm: các layer và các kết nối:



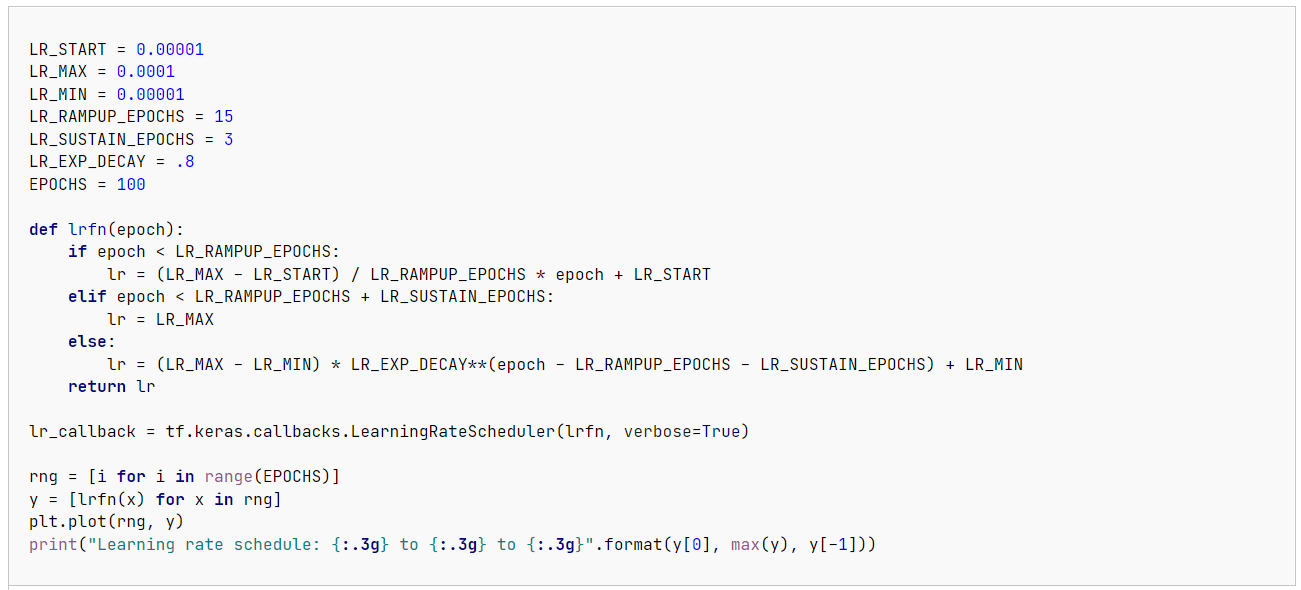
* Ensembling the Models: Kết hợp 2 mô hình neural network đã được huấn luyện (Xception và DenseNet) để tạo thành một mô hình kết hợp. Em sử dụng dự đoán của cả 2 mô hình Xception và DenseNet, sau đó sử dụng hàm average để tính trung bình dự đoán 2 mô hình:



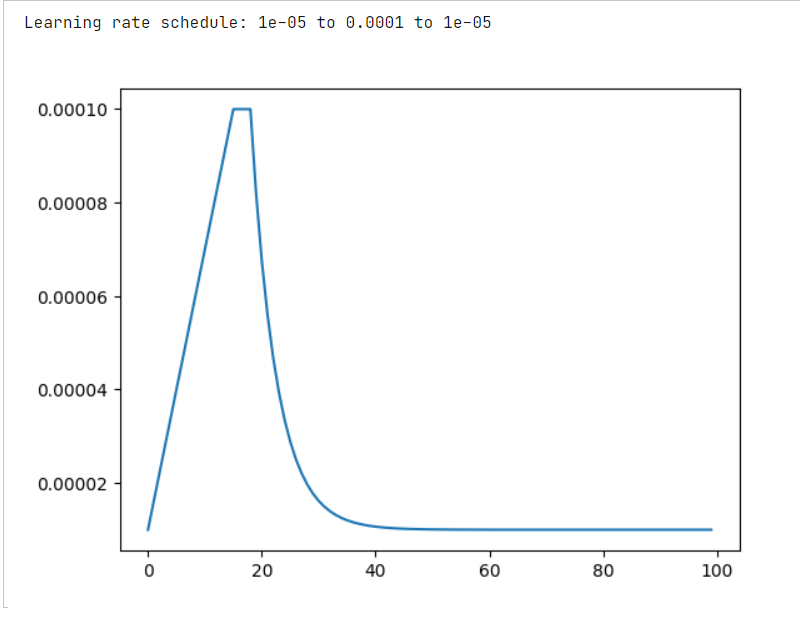
Biểu đồ biểu diễn cấu trúc của mô hình kết hợp:



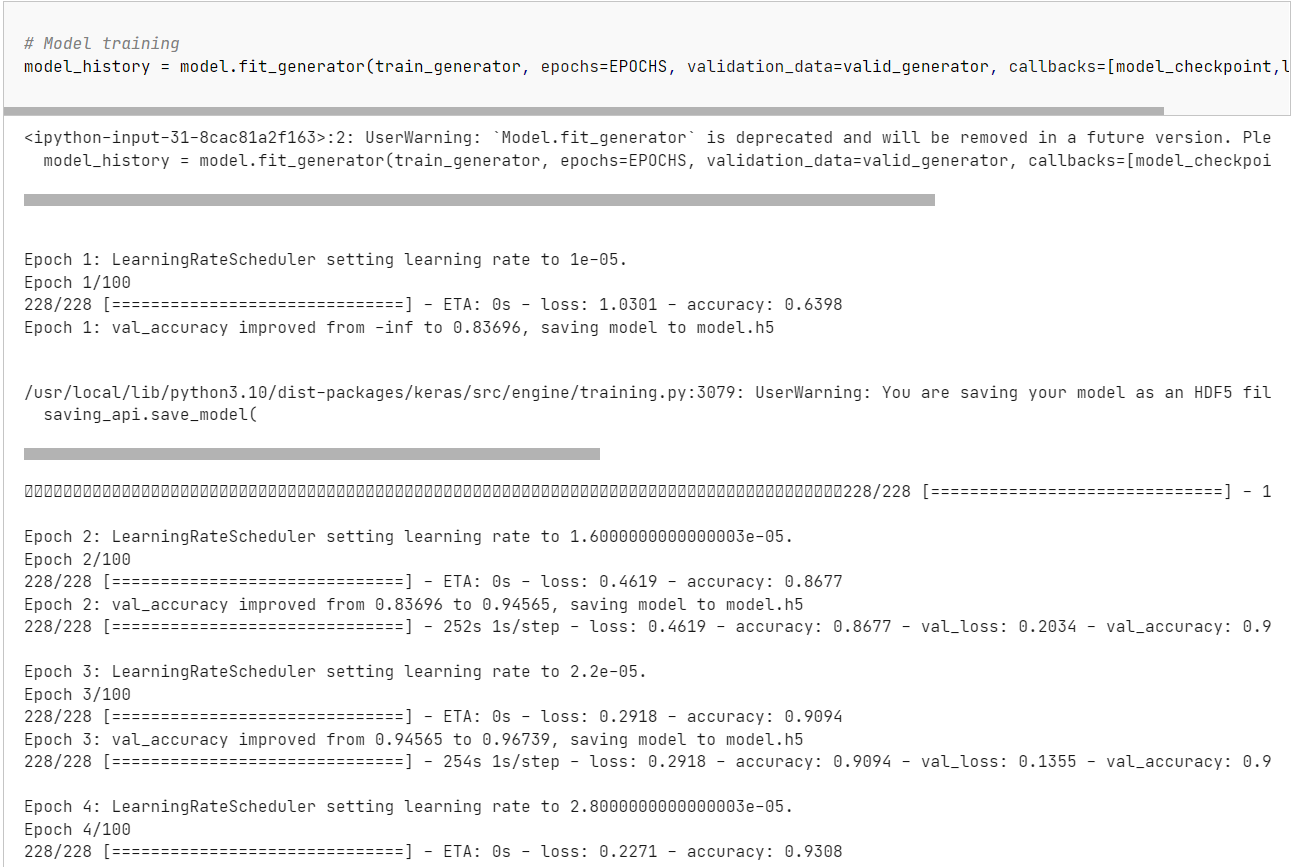
Setting the hyperparameters & Callbacks: Thiết lập hyperparamenters để kiểm soát quá trình đào tạo và Callbacks cho quá trình huấn luyện mô hình:



Kết quả:



* Training the Model:



Saving History: Lưu lịch sử huấn luyện vào file CSV: Dữ liệu lịch sử huấn luyện của mô hình được lưu dưới dạng file CSV (ModelHistory.csv) bằng cách chuyển đổi thông tin từ biến model\_history.history thành một DataFrame của thư viện Pandas. Các thông tin bao gồm: accuracy, loss, val\_accuracy, và val\_loss trong suốt quá trình huấn luyện.



Model History Visualisation: Vẽ đồ thị biểu diễn lịch sử huấn luyện mô hình, gồm 4 biểu đồ: accuracy plot (biểu đồ độ chính xác), loss plot (biểu đồ giảm thiểu loss), validation accuracy plot (biểu đồ độ chính xác trên tập validation), và validation loss plot (biểu đồ giảm thiểu loss trên tập validation).



* Predicting Classes



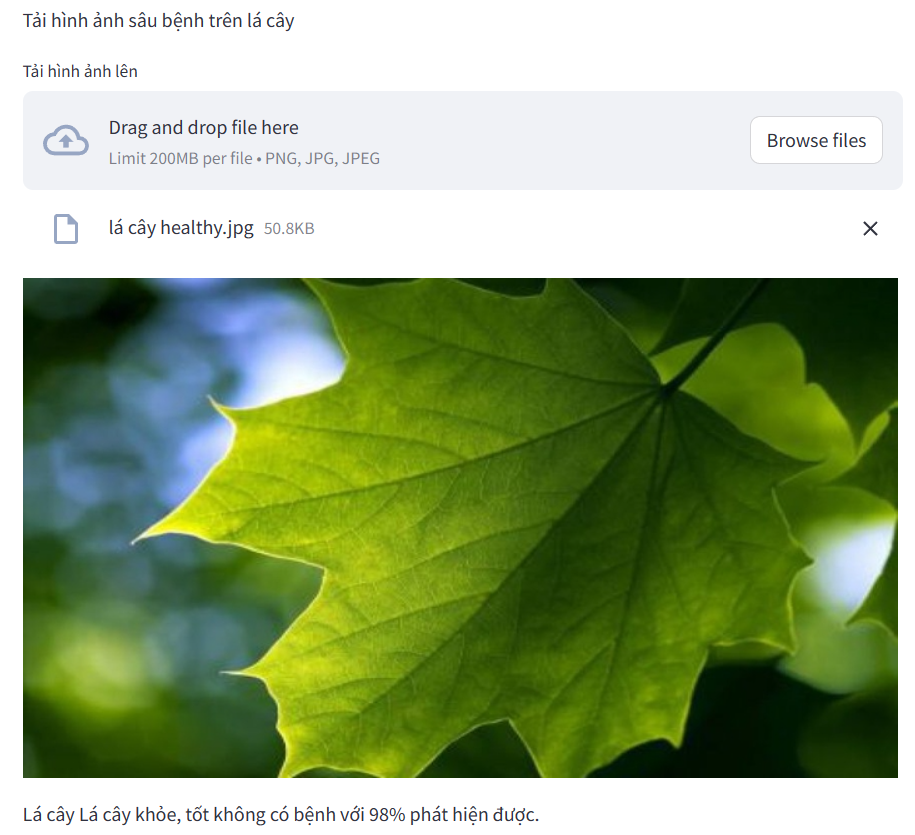


1. **KẾT QUẢ CHẠY CHƯƠNG TRÌNH**

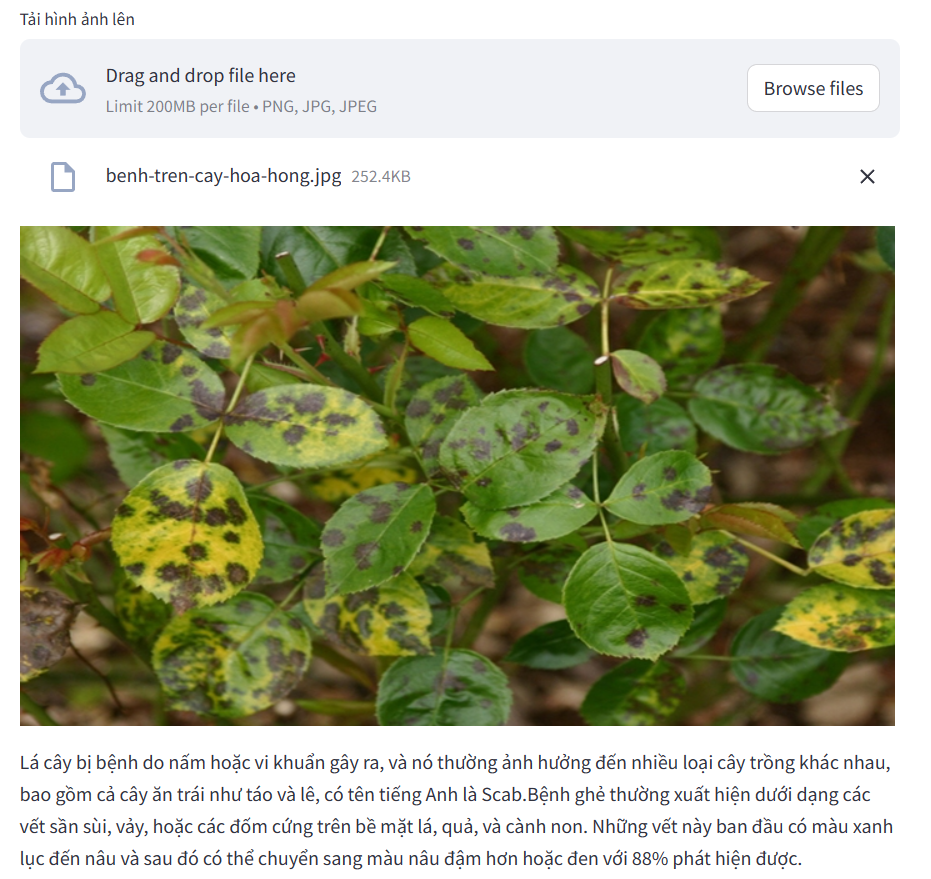
Giao diện của chương trình:

****

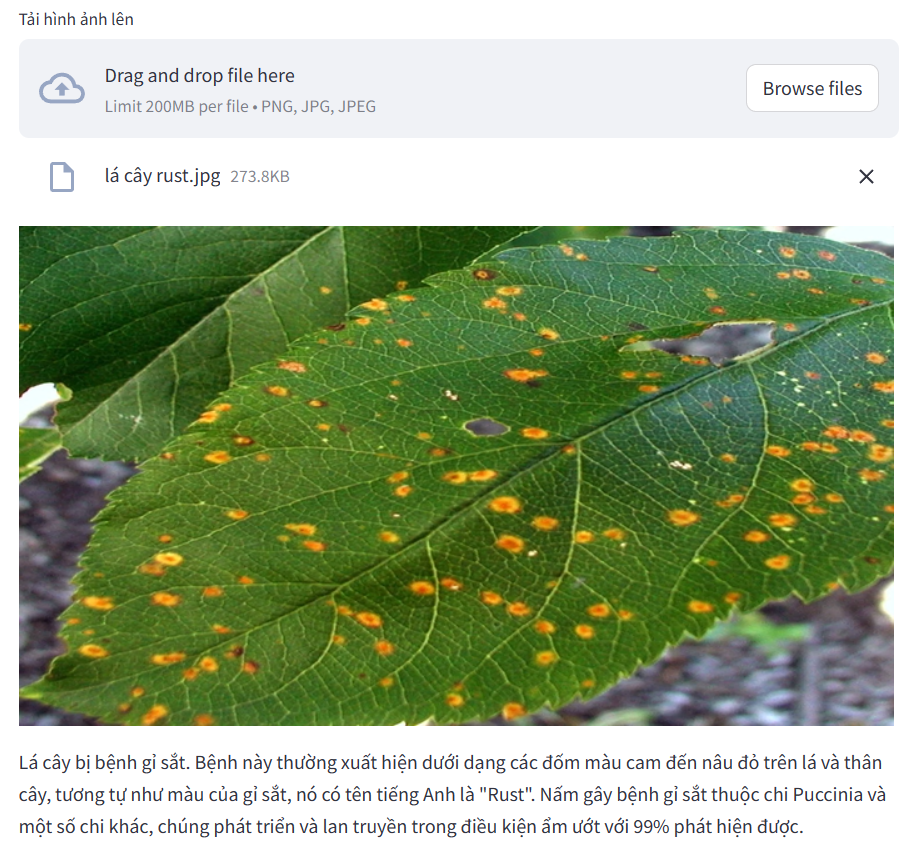
Kết quả 1:



Kết quả 2:



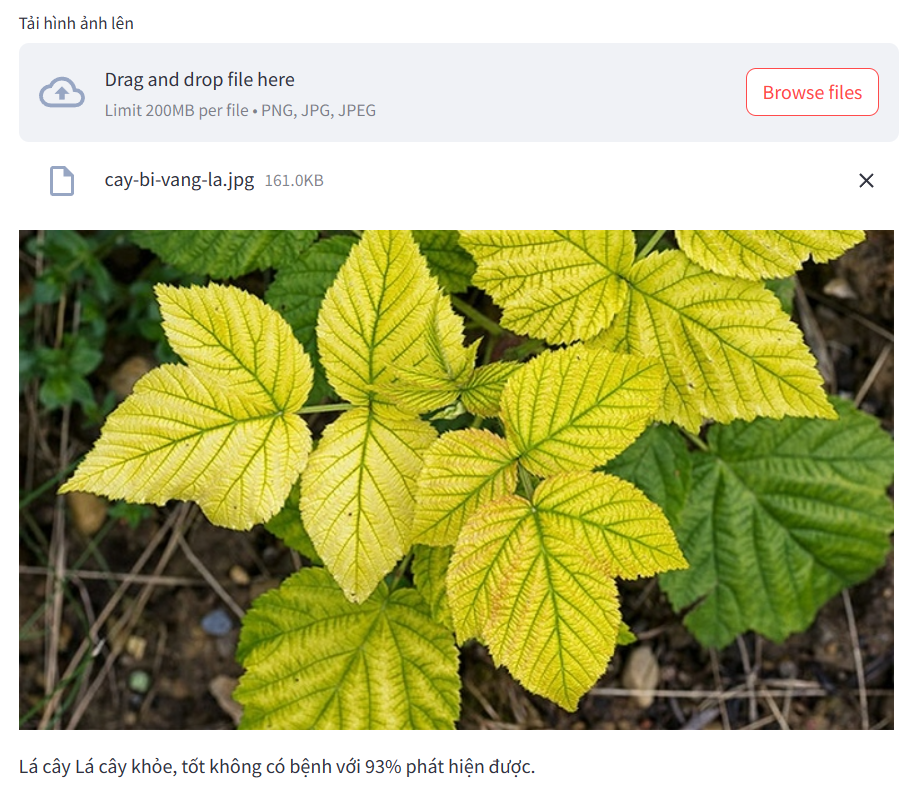
Kết quả 3:



1. **NHẬN XÉT ĐÁNH GIÁ**
2. Phân tích số liệu dự đoán của các mô hình

Hiện tại dataset của nhóm em chỉ mới phân loại bệnh của lá cây thành 4 nhóm là: healthy (lá cây khỏe mạnh), multi\_diseases (lá cây bị nhiều loại bệnh cùng lúc), rust (lá cây bị bệnh gỉ sắt) và scab (lá cây bị nấm). Đối với những mẫu lá cây có tình trạng bệnh thuộc 1 trong 4 nhóm trên thì tỉ lệ dự đoán của mô hình khá cao, trong khoảng từ 80% trở lên. Ngược lại, nếu như mẫu lá cây cần nhận diện không nằm trong 4 nhóm trên thì không thể dự đoán được, và kết quả sẽ được phân vào loại lá cây khỏe mạnh:

Ví dụ về trường hợp mẫu lá cây cần nhận diện bệnh không thuộc 1 trong 4 nhóm lá cây sẵn có:



Đây là 1 mẫu lá cây bị bệnh vàng lá. Tuy nhiên do trong dataset của nhóm em chưa có nhóm bệnh vàng lá nên chương trình sẽ phân loại mẫu lá cây vào nhóm khỏe mạnh.

1. Đánh giá về ứng dụng

Đề tài của nhóm em còn gặp nhiều vấn đề như chưa đa dạng về mẫu bệnh của lá cây, tỉ lệ dự đoán đối với mẫu bệnh lá cây có trong dataset khá cao (trên 80%). Tuy nhiên đối với mẫu bệnh lá cây chưa có trong dataset thì tỉ lệ dự đoán không thể tin tưởng được. Ngoài ra chương trình có thể được nâng cấp hơn với 1 vài tính năng như thêm camera để người dùng có thể xem trực tiếp, nhận diện lá cây đó thuộc loại lá cây gì, ngoài ra có thể đề xuất phương án chữa bệnh cho lá cây,…

Về mặt thực tiễn nhóm em thấy đề tài này nếu như thành công và được phát triển thêm nhiều tính năng, thân thiện với người dùng sẽ rất hữu ích cho các vấn đề về nông nghiệp, lâm nghiệp, …, đặc biệt là với những người dùng không có chuyên môn như nông dân, người trồng cây cảnh tại nhà thì họ có thể nhanh chóng biết được cây của mình đang mắc bệnh gì và có được các phương pháp chữa bệnh, chăm sóc cây cơ bản.

Trên đây là toàn bộ phần báo cáo đề tài “Nhận diện bệnh của lá cây”. Trong quá trình thực hiện nhóm em còn gặp nhiều thiếu xót và nhóm rất mong nhận được những lời nhận xét của giảng viên dạy. Nhóm em xin chân thành cảm ơn.

**TÀI LIỆU THAM KHẢO**

1. TỰ ĐỘNG NHẬN DẠNG MỘT SỐ LOẠI SÂU BỆNH TRÊN LÁ BƯỞI SỬ DỤNG CÔNG NGHỆ ẢNH - Nguyễn Minh Triết1 , Trương Quốc Bảo2 và Trương Quốc Định3 (1: Công ty Cổ phần Dịch vụ Bưu chính viễn thông Sài Gòn; 2: Khoa Công nghệ, Trường Đại học Cần Thơ; 3: Khoa Công nghệ Thông tin và Truyền thông, Trường Đại học Cần Thơ)
2. [Kaggle Competition -Plant Pathology 2020 - FGVC7] Phân loại bệnh lá cây – Pham Thi Hong Anh
3. Face-Recognition-with-OpenCV-Python-DL – Trần Văn Huy
4. NHẬN DIỆN BỆNH TRÊN LÁ DỰA TRÊN MẠNG NEURON TÍCH CHẬP – Ngô Tấn Đăng – Phạm Trần Đăng Khoa – Hồ Sĩ Quỳnh Đức – TP.HCM 2022