

BỘ NÔNG NGHIỆP VÀ MÔI TRƯỜNG
PHÂN HIỆU TRƯỜNG ĐẠI HỌC THỦY LỢI
BỘ MÔN CÔNG NGHỆ THÔNG TIN



BÁO CÁO KHAI PHÁ DỮ LIỆU CÔNG NGHỆ THÔNG TIN

ĐỀ TÀI : DỰ ĐOÁN CÂY BI BỆNH BẰNG PHƯƠNG PHÁP XỬ LÝ ẢNH

Giảng viên hướng dẫn:	<i>Vũ Thị Hạnh</i>
Lớp: S25-64CNTT	Thành viên:
	Nguyễn Lâm Mạnh Lê Quang Nhật

2025

Mục lục

CHƯƠNG 1: LỜI GIỚI THIỆU VỀ ĐỀ TÀI NHÓM 6	3
1.1. Lý do chọn đề tài	3
1.2 Mục tiêu nghiên cứu	4
1.3 Danh sách các loại bệnh đặc trưng của lá	4
.....	4
CHƯƠNG 2 TIẾN HÀNH XỬ LÝ DỮ LIỆU VÀ PHÂN TÍCH	6
2.1 Học chuyển giao (Transfer learning).....	6
2.2 Về mô hình MobileNetV2.....	6
2.3 Kiến trúc của mô hình MobileNetv2:	7
Chương 3: PHƯƠNG PHÁP VÀ HỆ THỐNG THỰC NGHIỆM.....	7
3.1 Mô tả bộ dữ liệu.....	7
3.2 Phân tích khám phá dữ liệu (EDA)	8
3.2.1 Tiền Xử Lý Dữ Liệu:	10
3.2.2 Huấn luyện mô hình:	12
3.2.6 Đánh giá & trực quan hóa (Evaluation & Visualization):	14
3.2.7 Đánh giá mô hình (Model Evaluation)	14
3.2.8 Đánh giá mô hình (Model Evaluation)	17
Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	18
4.1: Hiệu suất tổng thể:	18
4.2 Tối ưu mô hình	18
4.3 Hướng phát triển.....	18

CHƯƠNG 1: LỜI GIỚI THIỆU VỀ ĐỀ TÀI NHÓM 6

1.1. Lý do chọn đề tài



Hình 1: lá bệnh

Trong bối cảnh nông nghiệp hiện đại đang chuyển mình mạnh mẽ nhờ vào công nghệ số, việc ứng dụng trí tuệ nhân tạo (AI) vào phát hiện sớm bệnh cây trồng đóng vai trò then chốt trong việc nâng cao năng suất và chất lượng nông sản. Một trong những vấn đề phổ biến và gây thiệt hại lớn là bệnh trên lá cây – dấu hiệu đầu tiên phản ánh tình trạng sức khỏe của cây trồng.

Tuy nhiên, việc phát hiện bệnh lá bằng mắt thường đòi hỏi kinh nghiệm chuyên môn cao và không phải lúc nào cũng chính xác. Do đó, đề tài “Ứng dụng MobileNetV2 trong phân tích hình ảnh lá cây để phát hiện bệnh” được lựa chọn nhằm xây dựng một hệ thống nhẹ, hiệu quả, có thể triển khai trên thiết bị di động, giúp nông dân và kỹ sư nông nghiệp nhận diện bệnh nhanh chóng, chính xác và tiết kiệm chi phí.

MobileNetV2 là một kiến trúc mạng nơ-ron tích chập tối ưu cho các thiết bị có tài nguyên hạn chế, nhưng vẫn đảm bảo độ chính xác cao trong phân loại hình ảnh. Việc kết hợp MobileNetV2 với tập dữ liệu hình ảnh lá cây bị bệnh sẽ mở ra hướng đi mới trong việc giám sát cây trồng thông minh.

1.2 Mục tiêu nghiên cứu

Xây dựng mô hình học sâu sử dụng MobileNetV2 để phân loại bệnh lá cây dựa trên hình ảnh. Đánh giá hiệu quả mô hình qua các chỉ số: độ chính xác, độ nhạy, độ đặc hiệu và F1-score.

Triển khai mô hình trên thiết bị di động nhằm hỗ trợ người dùng cuối (nông dân, kỹ sư nông nghiệp).

1.3 Danh sách các loại bệnh đặc trưng của lá

Bệnh: Tomato__Tomato_Yellow_Leaf_Curl_Virus
Khả năng: 98.55%



Bệnh: Tomato__Tomato_Yellow_Leaf_Curl_Virus
Khả năng: 99.99%



Bệnh: Apple__Apple_scab
Khả năng: 72.70%



Bệnh: Potato__healthy
Khả năng: 53.32%



Bệnh: Tomato__Bacterial_spot
Khả năng: 48.22%



Bệnh: Tomato__Tomato_Yellow_Leaf_Curl_Virus
Khả năng: 99.58%



Hình 1.2: các loại lá bệnh

Dưới đây là tổng hợp các loại cây trồng và các bệnh lá thường gặp, được sử dụng làm tập dữ liệu huấn luyện trong đề tài:

Apple (Táo)

Apple Scab: Đốm nâu xám trên lá, gây rụng sớm.

Black Rot: Vết thối đen hình tròn, ảnh hưởng cả lá và quả.

Cedar Apple Rust: Đốm vàng cam, làm biến dạng lá.

Healthy: Lá xanh đều, không có vết đốm hay cháy.

Corn (Ngô/Bắp)

Cercospora Leaf Spot (Gray Leaf Spot): Đốm xám hình chữ nhật, gây khô lá.
Common Rust: Nốt phồng nâu đỏ (mụn gỉ sắt).
Northern Leaf Blight: Đốm nâu dài, lan rộng dọc theo gân lá.
Healthy: Lá xanh, không có vết bệnh.

Potato (Khoai tây)

Early Blight: Đốm nâu tròn có vòng đồng tâm.
Late Blight: Đốm nước lan rộng, thối đen toàn bộ lá.
Healthy: Lá mượt, không đốm.

Tomato (Cà chua)

Bacterial Spot: Đốm nước nhỏ do vi khuẩn, làm rách mép lá.
Early Blight: Đốm nâu sẫm có vòng tròn đồng tâm.
Late Blight: Đốm thối xám, lan rộng và gây rụng lá.
Leaf Mold: Mốc vàng mặt dưới lá.
Septoria Leaf Spot: Đốm nâu nhỏ li ti, nhiều ở phần dưới cây.
Spider Mites (Two-Spotted): Chấm vàng, lá héo dần.
Target Spot: Đốm tròn lớn, màu nâu đậm ở giữa.
Mosaic Virus: Lá biến dạng, loang màu xanh vàng.
Yellow Leaf Curl Virus: Lá cuộn tròn, mép vàng.
Healthy: Lá đều màu, không có vết bệnh.

Pepper Bell (Ớt chuông)

Bacterial Spot: Đốm sẫm, gây rách lá.
Healthy: Lá xanh đậm, không đốm.

Grape (Nho)

Black Rot: Đốm nâu, gây thối quả.
Esca (Black Measles): Vết cháy đen giữa lá.
Leaf Blight (Isariopsis Leaf Spot): Đốm nâu không đều, dễ rụng.
Healthy: Lá xanh bóng, không bệnh.

Strawberry (Dâu tây)

Leaf Scorch: Mép lá khô, đỏ sẫm.
Healthy: Lá xanh mượt, không cháy mép.

Soybean (Đậu nành)

Healthy: Lá xanh đều, không đốm.

Squash (Bí ngòi)

Powdery Mildew: Lớp phấn trắng phủ trên bề mặt lá.

Raspberry (Mâm xôi)

Healthy: Lá không bị biến màu.

CHƯƠNG 2 TIẾN HÀNH XỬ LÝ DỮ LIỆU VÀ PHÂN TÍCH

2.1 Học chuyển giao (Transfer learning)

Huấn luyện một mô hình CNN từ đầu đòi hỏi một lượng dữ liệu khổng lồ và năng lực tính toán rất lớn. Học chuyển giao là một giải pháp hiệu quả, cho phép chúng ta tận dụng "kiến thức" từ một mô hình đã được huấn luyện trước trên một bộ dữ liệu lớn (như ImageNet).

2.2 Về mô hình MobileNetV2

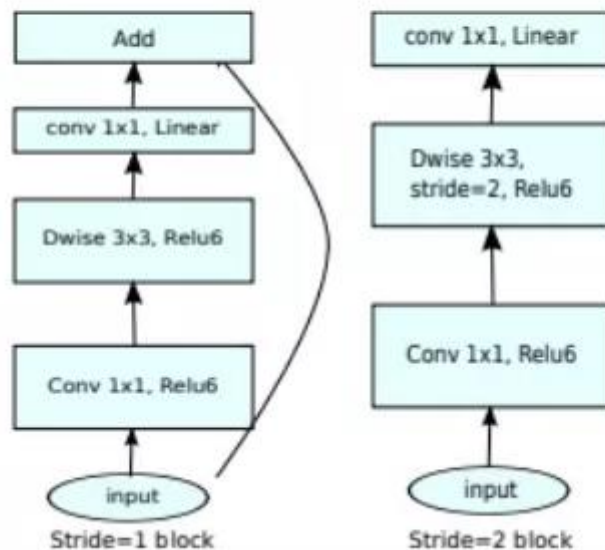
MobileNetV2 là một kiến trúc CNN nhẹ được phát triển bởi Google vào năm 2018, nhằm tối ưu hóa hiệu suất trên các thiết bị di động và hệ thống nhúng. Đây là phiên bản cải tiến của MobileNetV1, với nhiều thay đổi về cấu trúc giúp tăng độ chính xác và giảm độ phức tạp tính toán.

MobileNetV2 được thiết kế dựa trên hai khái niệm chính:

Inverted Residuals: Khác với kiến trúc ResNet truyền thống, MobileNetV2 sử dụng kết nối dư đảo ngược, trong đó đầu vào và đầu ra của khối có số chiều thấp, còn phần giữa có số chiều cao hơn.

Linear Bottleneck: Sau khi mở rộng chiều dữ liệu, MobileNetV2 sử dụng lớp tuyến tính thay vì lớp phi tuyến (ReLU) để tránh mất thông tin trong không gian đặc trưng thấp.

2.3 Kiến trúc của mô hình MobileNetV2:



Hình 2:

Kiến trúc MobileNetV2 bao gồm các thành phần chính sau:

Depthwise Separable Convolution: Là kỹ thuật tách lớp tích chập thành hai bước: tích chập theo từng kênh (depthwise) và tích chập điểm (pointwise). Điều này giúp giảm đáng kể số lượng tham số và phép toán.

Expansion Layer: Mở rộng chiều dữ liệu bằng tích chập 1x1.

Depthwise Convolution Layer: Áp dụng tích chập 3x3 cho từng kênh riêng biệt.

Projection Layer (Linear Bottleneck): Thu hẹp chiều dữ liệu bằng tích chập 1x1 và không sử dụng hàm kích hoạt ReLU.

Shortcut Connection: Kết nối đầu vào và đầu ra của khối nếu có cùng kích thước, giúp truyền thông tin hiệu quả hơn.

Chương 3: PHƯƠNG PHÁP VÀ HỆ THỐNG THỰC NGHIỆM

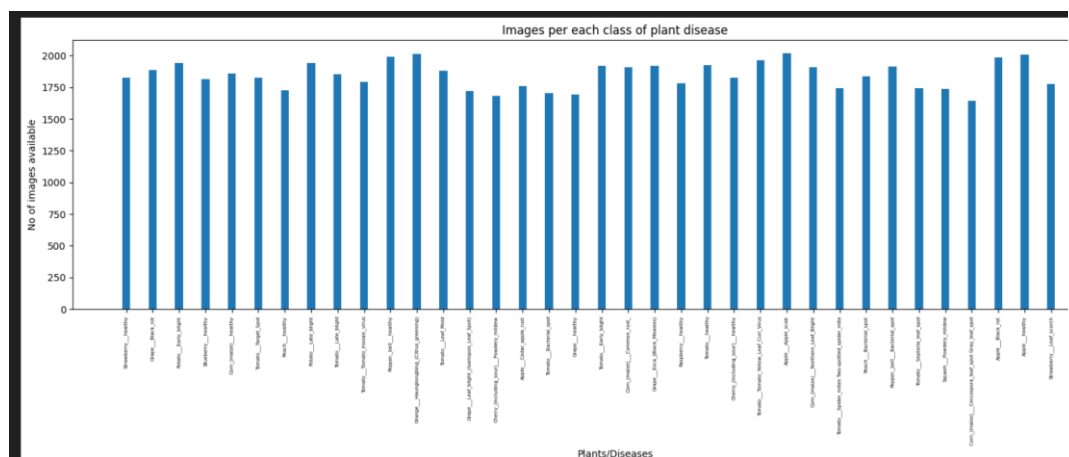
3.1 Mô tả bộ dữ liệu

Nguồn dữ liệu:

Bộ dữ liệu được sử dụng trong đề tài là "New plant diseases" do Paul Mooney công bố trên nền tảng Kaggle. Bộ dữ liệu này được phát triển nhằm hỗ trợ nghiên cứu về phân tích và dự đoán lá bị bệnh thông qua ảnh..

3.2 Phân tích khám phá dữ liệu (EDA)

Phân tích khám phá dữ liệu (Exploratory Data Analysis - EDA) là bước đầu tiên giúp ta hiểu rõ về cấu trúc, kích thước, đặc điểm phân bố cũng như tính chất của bộ dữ liệu. Bước này rất quan trọng đối với các bài toán nhận dạng hình ảnh vì nó giúp ta phát hiện sự mất cân bằng giữa các lớp, đánh giá chất lượng ảnh, và đề xuất hướng xử lý phù hợp

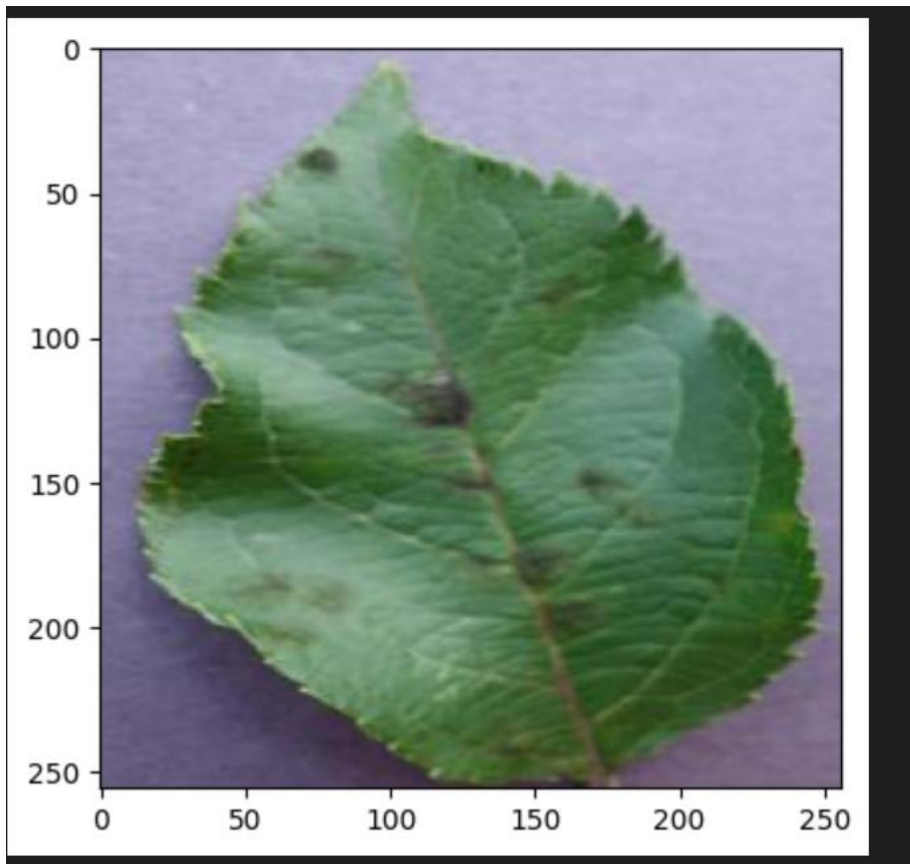


Số lượng ảnh của mỗi loại cây hoặc bệnh trong tập dữ liệu là rất đồng đều, hầu hết các lớp đều có khoảng 1700–2000 ảnh.

Không có lớp nào quá ít ảnh hoặc quá nhiều ảnh, cho thấy tập dữ liệu được cân bằng tốt.

Sự cân bằng này giúp mô hình học máy hoặc mạng nơ-ron khi huấn luyện không bị thiên lệch về bất kỳ loại bệnh nào, từ đó tăng độ chính xác và tính tổng quát của mô hình.

Việc mỗi lớp có số lượng ảnh tương đương cũng cho thấy tập dữ liệu được xây dựng có chủ đích và chất lượng tốt, phù hợp cho các bài toán phân loại bệnh cây trồng.



Trục X (ngang, ở dưới ảnh)

Biểu thị **chỉ số cột (pixel column index)** trong ảnh.

Giá trị từ **0 đến khoảng 250**, nghĩa là ảnh có **chiều rộng ~250 pixel**.

Mỗi con số thể hiện vị trí của pixel theo chiều ngang (từ trái sang phải). Ví dụ: pixel tại vị trí (100, 150) nằm ở cột 100, hàng 150.

Trục Y (dọc, bên trái ảnh)

Biểu thị **chỉ số hàng (pixel row index)** trong ảnh.

Giá trị từ **0 đến ~250**, tức ảnh có **chiều cao ~250 pixel**.

Mỗi con số thể hiện vị trí của pixel theo chiều dọc (từ trên xuống dưới).

Khu vực trung tâm trong trường hợp này là một **chiếc lá** (có vài đốm đen, có thể biểu hiện bệnh trên lá).

Mỗi điểm nhỏ trong ảnh (pixel) chứa giá trị **màu RGB** (Red, Green, Blue). Tổng hợp lại tạo thành màu sắc của bức ảnh mà bạn nhìn thấy.

tổng thể: ảnh này được hiển thị với **kích thước 256×256 pixel (xấp xỉ)** — một kích thước phổ biến trong **xử lý ảnh và học sâu (deep learning)**.

Các chỉ số (0–255 ở cả hai trục) giúp bạn xác định **tọa độ của từng pixel** trong ma trận ảnh khi thao tác bằng Python (NumPy hoặc OpenCV).

3.2.1 Tiền Xử Lý Dữ Liệu:

-Giới thiệu lý thuyết về tiền xử lý dữ liệu:

Tiền xử lý dữ liệu (Data Preprocessing) là giai đoạn đầu tiên trong quá trình xây dựng mô hình học máy hoặc học sâu (Deep Learning).

Mục tiêu của giai đoạn này là:

Làm sạch dữ liệu (data cleaning)

Chuyển đổi dữ liệu về định dạng phù hợp

Tăng chất lượng và độ chính xác của mô hình

Dữ liệu thực tế thường nhiều, thiếu, không đồng nhất hoặc không cân bằng, nên nếu bỏ qua bước này, hiệu năng mô hình có thể bị giảm nghiêm trọng.

-Các bước tiền xử lý dữ liệu cơ bản

Bước 1: Thu thập dữ liệu (Data Collection)

Nguồn: cảm biến, ảnh, video, file CSV, cơ sở dữ liệu, web, API, v.v.

Mục tiêu: đảm bảo dữ liệu đủ lớn và đại diện cho bài toán.

Bước 2: Làm sạch dữ liệu (Data Cleaning)

Loại bỏ dữ liệu thiếu (Missing values): dùng trung bình, trung vị hoặc loại bỏ dòng/cột.

Xử lý dữ liệu nhiễu (Noise Handling): dùng lọc, làm trơn, hoặc loại bỏ ngoại lệ.

Phát hiện và loại bỏ ngoại lệ (Outliers): dùng Z-score, IQR, hoặc mô hình thống kê.

Bước 3: Biến đổi dữ liệu (Data Transformation)

Chuẩn hóa dữ liệu (Normalization): đưa giá trị về cùng thang đo (VD: [0,1] hoặc [-1,1]).

Chuẩn hóa phân phối (Standardization): đưa về dạng phân phối chuẩn (z-score).

Mã hóa dữ liệu phân loại (Encoding): one-hot, label encoding,...

Bước 4: Giảm chiều dữ liệu (Dimensionality Reduction)

PCA, t-SNE, hoặc AutoEncoder để giảm số lượng đặc trưng, tránh overfitting.

Bước 5: Chia tập dữ liệu (Data Splitting)

Tách dữ liệu thành train / validation / test (thường theo tỉ lệ 70/20/10 hoặc 80/10/10).

-Tiền xử lý cơ bản cho CNN:

Resize ảnh: đưa tất cả ảnh về cùng kích thước 224x224 (ResNet), 32x32 (CIFAR), 299x299 (Inception)

Chuẩn hóa giá trị pixel ổn định độ sáng, giúp hội tụ nhanh chia cho 255 để về $[0,1]$ hoặc chuẩn hóa theo mean/std

Chuyển đổi kênh màu đảm bảo đúng định dạng đầu vào RGB, Grayscale, HSV,...

Tăng cường dữ liệu (Data Augmentation) giảm overfitting, tạo thêm dữ liệu lật ảnh, xoay, dịch chuyển, phóng to, cắt ngẫu nhiên

Chuẩn hóa đặc trưng (Feature Normalization) tăng tốc hội tụ, giảm sai lệch sử dụng mean/std từ ImageNet nếu dùng transfer learning

-Tiền xử lý trong CNN hiện đại:

Color jitter / Random brightness: thay đổi độ sáng, tương phản, bão hòa

Random cropping & resizing: cắt ngẫu nhiên vùng ảnh trước khi resize

Normalization theo tập dữ liệu lớn: dùng mean/std của ImageNet

AutoAugment / RandAugment: tự động chọn phép biến đổi tối ưu

Mixup / CutMix: trộn hai ảnh với nhau để tăng tính khái quát

Label smoothing: giảm overconfidence của mô hình

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1280)	0
dropout (Dropout)	(None, 1280)	0
dense (Dense)	(None, 38)	48,678

Total params: 2,306,662 (8.80 MB)

Trainable params: 48,678 (190.15 KB)

Non-trainable params: 2,257,984 (8.61 MB)

InputLayer (input_layer_1) (None, 224, 224, 3) Ảnh đầu vào RGB có kích thước 224x224, 3 kênh màu.

MobileNetV2 (mobilenetv2_1.00_224) (None, 7, 7, 1280) Đây là mô hình MobileNetV2 được dùng làm **feature extractor** (trích xuất đặc trưng). Các tham số bị "đóng băng" (non-trainable).

GlobalAveragePooling2D (None, 1280) Chuyển tensor 3D ($7 \times 7 \times 1280$) → vector 1D (1280). Giúp giảm số tham số và tránh overfitting.

Dropout (None, 1280) Giảm overfitting bằng cách ngẫu nhiên tắt một số neuron trong quá trình huấn luyện.

Dense (fully connected) (None, 38) Lớp đầu ra (output layer) gồm 38 neuron → mô hình này được huấn luyện để **phân loại 38 lớp**.

3.2.2 Huấn luyện mô hình:

Epoch 1/20				
150/150	<div></div>	58s 367ms/step	- accuracy: 0.1067	- loss: 4.3379
Epoch 2/20				
150/150	<div></div>	52s 345ms/step	- accuracy: 0.3269	- loss: 2.5003
Epoch 3/20				
150/150	<div></div>	54s 362ms/step	- accuracy: 0.4798	- loss: 1.8457
Epoch 4/20				
150/150	<div></div>	51s 341ms/step	- accuracy: 0.6044	- loss: 1.3204
Epoch 5/20				
150/150	<div></div>	48s 322ms/step	- accuracy: 0.6617	- loss: 1.1271
Epoch 6/20				
150/150	<div></div>	48s 321ms/step	- accuracy: 0.6815	- loss: 1.0913
Epoch 7/20				
150/150	<div></div>	50s 331ms/step	- accuracy: 0.6767	- loss: 1.0771
Epoch 8/20				
150/150	<div></div>	48s 322ms/step	- accuracy: 0.7231	- loss: 0.9344
Epoch 9/20				
150/150	<div></div>	47s 316ms/step	- accuracy: 0.7619	- loss: 0.7940
Epoch 10/20				
150/150	<div></div>	49s 324ms/step	- accuracy: 0.7387	- loss: 0.8454
Epoch 11/20				
150/150	<div></div>	51s 342ms/step	- accuracy: 0.7804	- loss: 0.7249
Epoch 12/20				
150/150	<div></div>	52s 345ms/step	- accuracy: 0.7448	- loss: 0.8177
Epoch 13/20				
150/150	<div></div>	52s 344ms/step	- accuracy: 0.7748	- loss: 0.7259
Epoch 14/20				
150/150	<div></div>	51s 339ms/step	- accuracy: 0.7673	- loss: 0.7740

Hình 3.2.5: huấn luyện mô hình

Mỗi epoch → mô hình được huấn luyện qua 150 batch dữ liệu.

Tổng cộng huấn luyện 20 lần (20 epoch) trên toàn bộ tập dữ liệu.

Epoch 1/20: đang ở epoch thứ 1 trên tổng 20 epoch.

150/150: có tổng 150 batch, mô hình đã xử lý hết.

58s 367ms/step: mất 58 giây cho epoch này, mỗi batch trung bình 367ms.

accuracy: 0.1067: độ chính xác (accuracy) sau epoch này là 10.67%.

loss: 4.3379: giá trị hàm mất mát (loss) là 4.34.

Mô hình CNN đang học đúng hướng.

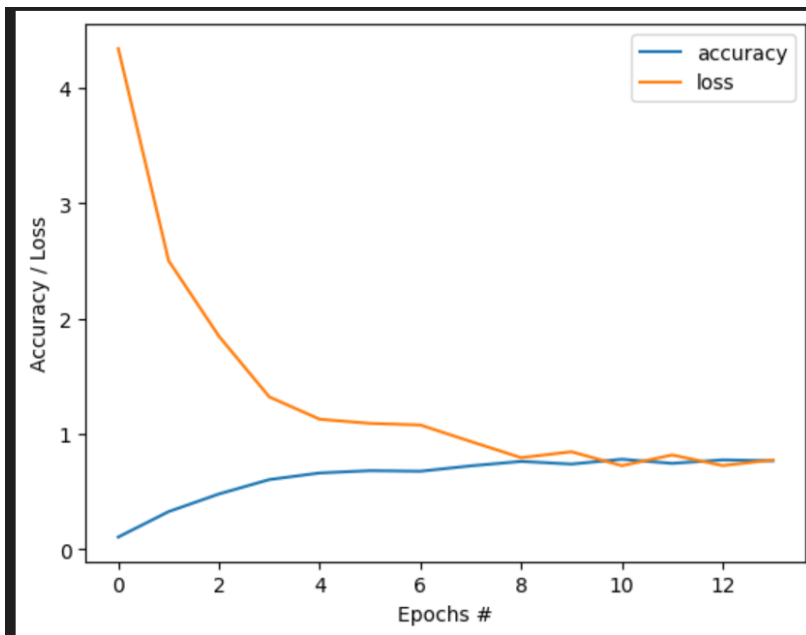
Sau 14 epoch, accuracy đạt khoảng **~77%** — khá tốt, có thể cải thiện thêm bằng:

Tăng số epoch lên 25–30.

Dùng **data augmentation** hoặc **learning rate scheduler**.

Áp dụng **Batch Normalization** hoặc **Dropout** để tránh overfitting.

3.2.6 Đánh giá & trực quan hóa (Evaluation & Visualization):



Hình 3.2.7: Đánh giá

3.2.7 Đánh giá mô hình (Model Evaluation)

```
model.predict(train_generator)
```

Mô hình dự đoán xác suất thuộc về từng lớp cho từng mẫu trong tập train.

Ví dụ: [0.8, 0.1, 0.1] nghĩa là xác suất cao nhất cho lớp 0.

```
np.argmax(y_pred,axis=1)
```

Lấy chỉ số có xác suất cao nhất làm nhãn dự đoán (predicted class).

```
train_generator.classes
```

Lấy nhãn thật (true labels) tương ứng với mỗi ảnh trong tập huấn luyện.

`train_generator.class_indices.keys()`

Lấy danh sách tên các lớp

```
2197/2197 696s 316ms/step
=== Kết quả đánh giá mô hình trên tập TRAIN ===
Accuracy : 79.83%
Precision: 85.96%
Recall : 79.83%
F1-score : 79.94%

=== Báo cáo chi tiết từng lớp ===
```

	precision	recall	f1-score	support
Apple__Apple_scab	0.65	0.90	0.75	2016
Apple__Black_rot	0.92	0.86	0.89	1987
Apple__Cedar_apple_rust	1.00	0.41	0.58	1760
Apple__healthy	0.84	0.93	0.88	2008
Blueberry__healthy	0.85	0.95	0.90	1816
Cherry_(including_sour)__Powdery_mildew	0.98	0.88	0.93	1683
Cherry_(including_sour)__healthy	0.93	0.95	0.94	1826
Corn_(maize)__Cercospora_leaf_spot_Gray_leaf_spot	0.96	0.71	0.82	1642
Corn_(maize)__Common_rust	0.96	0.98	0.97	1907
Corn_(maize)__Northern_Leaf_Blight	0.81	0.95	0.88	1908
Corn_(maize)__healthy	0.99	0.99	0.99	1859
Grape__Black_rot	0.69	0.98	0.81	1888
Grape__Esca_(Black_Measles)	0.99	0.55	0.71	1920
Grape__Leaf_blight_(Isariopsis_Leaf_Spot)	0.92	0.98	0.95	1722
Grape__healthy	0.98	0.96	0.97	1692
...				
accuracy			0.80	70295

Hình 3.2.8:Đánh giá mô hình

Accuracy (Độ chính xác)

79.83% — mô hình dự đoán đúng gần 80% ảnh trong tập huấn luyện

Precision (Độ chính xác của dự đoán): 85.96% — khá cao, nghĩa là mô hình ít dự đoán nhầm

Recall (Độ bao phủ): 79.83% — tương đồng với accuracy, mô hình bỏ sót một phần mẫu đúng

F1-score: 79.94% — cân bằng giữa độ chính xác và độ bao phủ

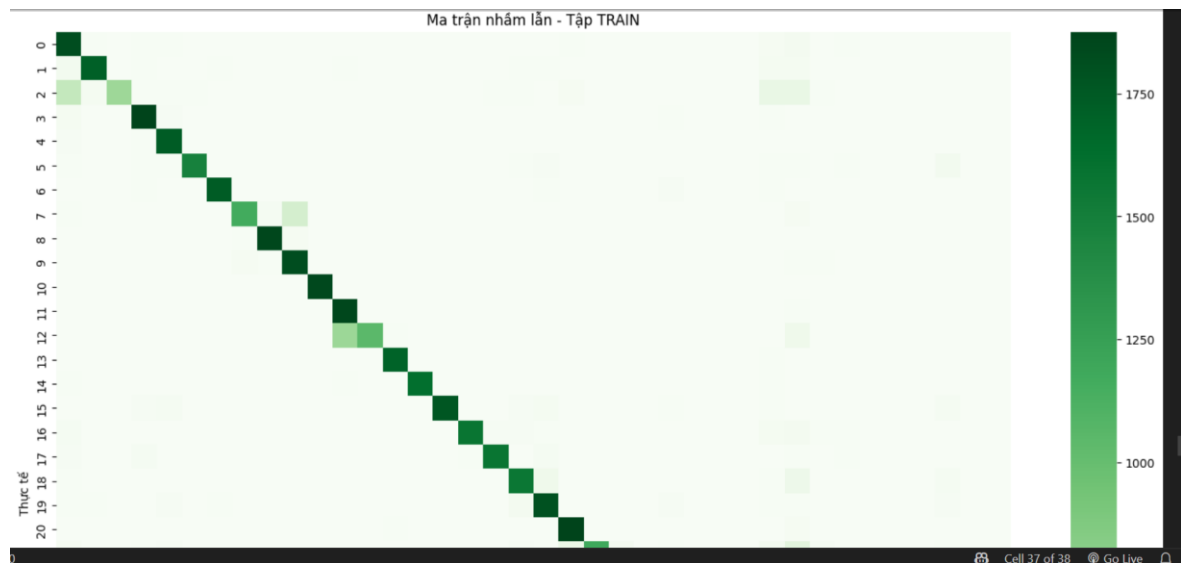
Precision: Tỷ lệ dự đoán đúng cho lớp đó. Nếu thấp → mô hình hay dự đoán nhầm vào lớp khác.

Recall: Tỷ lệ phát hiện đúng các mẫu của lớp đó. Nếu thấp → mô hình hay bỏ sót lớp này.

F1-score:

Giá trị cân bằng giữa precision và recall, phản ánh tổng thể độ tốt của mô hình cho lớp đó.

Support: Số lượng mẫu thật của lớp đó trong tập dữ liệu.



Hình 3.2.9: tiến hành dự đoán hình ảnh lá cây bị bệnh

1. Accuracy : 79.83%

Accuracy (độ chính xác) = số dự đoán đúng chia cho tổng số mẫu.

Với 38 lớp nhãn, 79.83% nghĩa là khoảng 3/4 các mẫu trong tập train được mô hình dự đoán chính xác.

Đây là kết quả trên tập train, không phải trên test, nên nó phản ánh khả năng học của mô hình với dữ liệu đã thấy.

2. Precision: 85.96%

Precision (độ chính xác theo lớp) đo lường tỉ lệ dự đoán đúng trên tổng số dự đoán trung bình có trọng số theo số mẫu từng lớp.

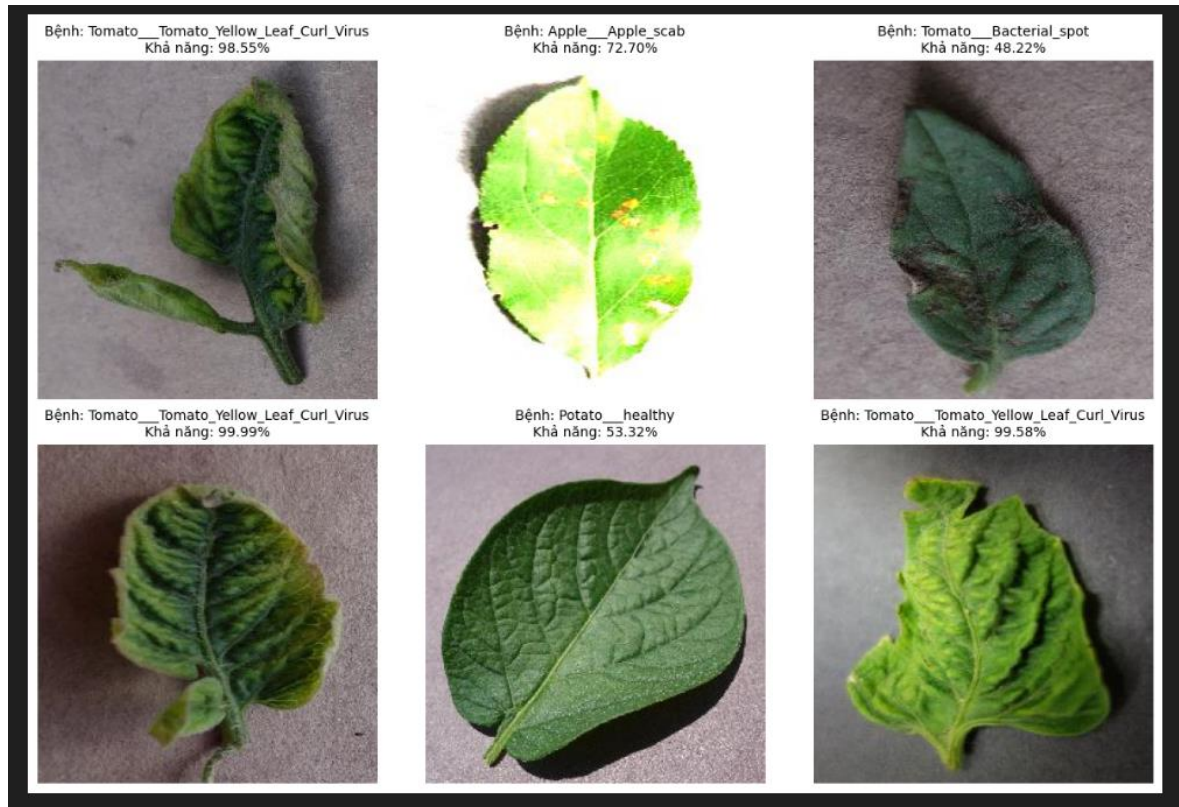
85.96% nghĩa là khi mô hình dự đoán một lớp nào đó, khoảng 85% dự đoán là chính xác.

3. Recall: 79.83% Recall (độ nhạy) = số dự đoán đúng chia cho tổng số mẫu thực sự của lớp đó.

Với 79.83%, tức là mô hình phát hiện đúng khoảng 3/4 mẫu thuộc mỗi lớp.

4. F1-score: 79.94% Với Precision và Recall, F1-score 79.94% là một con số hợp lý, phản ánh cân bằng giữa độ chính xác khi dự đoán và khả năng tìm đúng mẫu thực tế.

3.2.8 Đánh giá mô hình (Model Evaluation)



Hình 3.3: Tổng quan các loại bệnh

Tomato — Yellow Leaf Curl Virus (98.55%) Lá cà chua bị virus làm xoắn vàng mép — hình ảnh khá điển hình

Apple — Apple Scab (72.70%): Lá táo có vết đốm sạm, giống triệu chứng bệnh Apple Scab.

Tomato — Bacterial Spot (48.22%) Độ tin cậy thấp hơn (chưa chắc chắn), có thể nhầm với bệnh khác.

Tomato — Yellow Leaf Curl Virus (99.99%) Dự đoán rất chắc chắn, lá cuộn tròn và mép vàng rõ ràng.

Potato — Healthy (53.32%) Lá khoai tây bình thường, nhưng xác suất chưa cao → mô hình hơi phân vân.

Tomato — Yellow Leaf Curl Virus (99.58%) Rất chắc chắn, triệu chứng bệnh giống các mẫu bị virus ở trên.

Chương 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1: Hiệu suất tổng thể:

Mô hình đã học được khá tốt đặc trưng hình ảnh của các loại bệnh lá, phân biệt được phần lớn các lớp với độ chính xác khá cao.

Một số lớp đạt độ chính xác và F1-score trên 90–95%, đặc biệt là các lớp có đặc trưng bệnh rõ ràng như “Corn___healthy” hay “Cherry___healthy”.

4.2 Tối ưu mô hình

Data Augmentation: Xoay, lật, thay đổi ánh sáng, zoom,... giúp tăng tính đa dạng của dữ liệu và giảm overfitting.

Transfer Learning: Dùng các mô hình mạnh như EfficientNet, MobileNetV3, ResNet50, DenseNet đã huấn luyện sẵn (pretrained).

Fine-tuning: Huấn luyện lại các lớp cuối của mạng để mô hình thích nghi với dữ liệu lá bệnh.

4.3 Hướng phát triển

Thử mô hình Vision Transformer (ViT) hoặc ConvNeXt cho độ chính xác cao hơn.

Kết hợp mô hình phát hiện vùng (object detection) như YOLO / Faster R-CNN để xác định vị trí vùng bệnh trên lá, không chỉ là phân loại.

Dùng Explainable AI (Grad-CAM) để hiển thị vùng mô hình “nhìn” khi quyết định bệnh → giúp kiểm chứng và tăng độ tin cậy.

