

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



**BÁO CÁO MÔN: KHAI PHÁ DỮ LIỆU
TÊN ĐỀ TÀI: ỨNG DỤNG CÁC MÔ HÌNH HỌC SÂU TRONG
PHÂN LOẠI BỆNH TRÊN LÁ CÂY ĐẬU**

Giảng viên:	ThS. Vũ Thị Hạnh
Sinh viên thực hiện:	2351267275 - Nguyễn Văn Tân Phát 2351267271 - Nguyễn Hoàng Lộc
Lớp:	S26-65TTNT

TP. Hồ Chí Minh, ngày ... tháng ... năm 2026

NHẬN XÉT CỦA GIÁNG VIÊN HƯỚNG DẪN

TP. Hồ Chí Minh, ngày ... tháng ... năm 2025

Chữ ký của giảng viên

LỜI CẢM ƠN

Nhóm em xin chân thành cảm ơn ThS. Vũ Thị Hạnh – giảng viên phụ trách môn học đã tận tình giảng dạy, hướng dẫn và hỗ trợ nhóm em trong suốt quá trình học tập. Những kiến thức chuyên môn, phương pháp tiếp cận vấn đề cùng sự chỉ dẫn sát sao của cô đã giúp nhóm em hiểu rõ hơn nội dung môn học và hoàn thành bài báo cáo đúng yêu cầu.

Mặc dù nhóm đã cố gắng hết sức, bài làm vẫn khó tránh khỏi những thiếu sót do thời gian và kinh nghiệm còn hạn chế. Nhóm em rất mong nhận được những nhận xét và góp ý từ cô để có thể hoàn thiện hơn kiến thức cũng như kỹ năng trong thời gian tới.

Nhóm em xin chân thành cảm ơn cô!

MỤC LỤC

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI	7
1.1. Lý do chọn đề tài	7
1.2. Mục tiêu nghiên cứu.....	8
1.3. Đối tượng và Phạm vi nghiên cứu	8
1.4. Ý nghĩa khoa học và thực tiễn.....	8
CHƯƠNG 2: MỤC TIÊU VÀ BÀI TOÁN ĐẶT RA	10
2.1. Phát biểu bài toán	10
2.2. Đầu vào và đầu ra.....	10
2.3. Phương pháp tiếp cận	10
2.4. Tiêu chí đánh giá	10
CHƯƠNG 3: MÔ TẢ DỮ LIỆU, TIỀN XỬ LÝ VÀ TĂNG CƯỜNG DỮ LIỆU NÂNG CAO	11
3.1. Mô tả dữ liệu	11
3.2. Tiền xử lý dữ liệu	12
3.3. Tăng cường dữ liệu (Data Augmentation)	13
3.4. Đọc dữ liệu với ImageDataGenerator	13
CHƯƠNG 4: PHƯƠNG PHÁP THỰC HIỆN	14
4.1. Các mô hình học máy sử dụng.....	14
4.1.1. Mô hình MobileNetV3	14
4.1.2. Mô hình DeIT.....	15
4.1.3. Mô hình EfficientNetB3.....	16
4.1.4 Mô hình CNN tự xây.....	17
4.1.5 Mô hình YOLOv8n với bài toán Instance Segmentation	18
4.2. Grad Cam trực quan hóa giải thích mô hình	21

CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ CHI TIẾT CÁC MÔ HÌNH	22
5.1. Kết quả Mô hình BeanLeafVGG	22
5.2. Kết quả Mô hình MobilenetV3	23
5.3. Kết quả Mô hình Deit.....	24
5.4. Kết quả mô hình EfficientNetB3	26
5.8. So sánh Kết quả và Thảo luận (Comparative Analysis)	27
5.9 Giao diện WEBSITE.....	28
CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	30
TÀI LIỆU THAM KHẢO.....	32

DANH MỤC HÌNH ẢNH

Hình 3.1 Ảnh phân bố các lớp trong dữ liệu.....	11
Hình 3.2 Ảnh minh họa đặc trưng của 3 lớp.....	12
Hình 5.1 Biểu đồ history train BeanLeafVGG	22
Hình 5.2 Biểu đồ Confusion matrix BeanLeafVGG.....	22
Hình 5.4 Biểu đồ history train MobilenetV3	23
Hình 5.5 Biểu đồ confusion matrix MobilenetV3.....	23
Hình 5.7 Biểu đồ history train Deit	24
Hình 5.9 Biểu đồ Confusionmatrix Deit	25
Hình 5.10 Biểu đồ history train EfficientnetB3	26
Hình 5.11 Biểu đồ confusion matrix EfficientnetB3	26
Hình 5.12 Giao diện khi bắt đầu	28
Hình 5.13 Giao diện khi dự đoán.....	29
Hình 5.14 Giao diện so sánh các mô hình.....	29

DANH MỤC BẢNG

Bảng 5.1. Bảng so sánh tổng hợp các chỉ số hiệu năng	27
Bảng 5.2 Kết quả khi chạy CV với fold bằng 5	28

CHƯƠNG I: GIỚI THIỆU ĐỀ TÀI

1.1. Lý do chọn đề tài

Cây đậu là một trong những loại cây trồng quan trọng trong nông nghiệp, cung cấp nguồn protein thực vật thiết yếu cho con người. Tuy nhiên, các bệnh trên lá đậu như bệnh đốm góc cạnh (Angular Leaf Spot) và bệnh gỉ sét (Bean Rust) đang gây ra thiệt hại đáng kể về năng suất và chất lượng nông sản. Việc phát hiện sớm để phân biệt giữa lá khỏe mạnh (Healthy) và lá bệnh là vô cùng cần thiết để kiểm soát dịch bệnh kịp thời.

Hiện nay, việc phát hiện bệnh trên lá đậu chủ yếu dựa vào quan sát bằng mắt thường của nông dân và chuyên gia. Phương pháp này có nhiều hạn chế như độ chính xác không đồng đều, phụ thuộc vào kinh nghiệm cá nhân, và thường phát hiện bệnh ở giai đoạn muộn khi đã lan rộng. Đặc biệt, hai loại bệnh Angular Leaf Spot và Bean Rust có triệu chứng tương tự nhau ở giai đoạn đầu, gây khó khăn trong việc chẩn đoán chính xác.

Trong những năm gần đây, Học sâu (Deep Learning) đã đạt được những bước tiến vượt bậc trong lĩnh vực Thị giác máy tính (Computer Vision). Các mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) có khả năng tự động trích xuất đặc trưng từ hình ảnh mà không cần thiết kế thủ công, giúp đạt được độ chính xác cao trong các bài toán nhận dạng hình ảnh. Đặc biệt, sự ra đời của các mô hình như YOLO (You Only Look Once) đã cách mạng hóa bài toán phát hiện đối tượng với khả năng xử lý thời gian thực. Phiên bản YOLOv8 mới nhất không chỉ hỗ trợ phát hiện đối tượng mà còn tích hợp khả năng Instance Segmentation - phân vùng từng đối tượng riêng biệt trong ảnh.

Khác với phương pháp phân loại ảnh truyền thống chỉ trả lời câu hỏi "lá bị bệnh gì?", kỹ thuật Instance Segmentation kết hợp sức mạnh của Deep Learning cho phép xác định chính xác vị trí, hình dạng và ranh giới của từng vùng bệnh riêng biệt trên lá. Điều này giúp đánh giá mức độ nhiễm bệnh một cách định lượng, từ đó đưa ra phương pháp điều trị phù hợp.

1.2. Mục tiêu nghiên cứu

Đối tượng nghiên cứu của đề tài là hình ảnh lá cây đậu (Bean Leaf) với 3 trạng thái khác nhau. Đầu tiên là lá khỏe mạnh (Healthy) có màu xanh đồng đều, không có dấu hiệu bệnh và bề mặt lá nguyên vẹn. Tiếp theo là bệnh đốm góc cạnh (Angular Leaf Spot) do vi khuẩn *Pseudomonas syringae* gây ra, biểu hiện qua các vết đốm hình đa giác màu nâu trên lá, thường xuất hiện ở các góc gân lá. Cuối cùng là bệnh gỉ sắt (Bean Rust) do nấm *Uromyces appendiculatus* gây ra, biểu hiện qua các nốt sần màu nâu đỏ như gỉ sắt trên bề mặt lá.

Về phạm vi dữ liệu, đề tài sử dụng bộ dữ liệu hình ảnh lá đậu được thu thập và gán nhãn sẵn từ nguồn Kaggle với tổng cộng 1,035 ảnh huấn luyện và khoảng 120 ảnh validation. Dữ liệu được phân bố tương đối cân bằng giữa 3 lớp: Healthy (342 ảnh), Angular Leaf Spot (345 ảnh) và Bean Rust (348 ảnh).

1.3. Đối tượng và Phạm vi nghiên cứu

Đối tượng nghiên cứu của đề tài là hình ảnh lá cây đậu (Bean Leaf) với 3 trạng thái khác nhau:

- **Healthy (Lá khỏe mạnh):** Lá có màu xanh đồng đều, không có dấu hiệu bệnh, bề mặt lá nguyên vẹn.
- **Angular Leaf Spot (Bệnh đốm góc cạnh):** Bệnh do vi khuẩn *Pseudomonas syringae* gây ra, biểu hiện qua các vết đốm hình đa giác màu nâu trên lá, thường xuất hiện ở các góc gân lá.
- **Bean Rust (Bệnh gỉ sắt):** Bệnh do nấm *Uromyces appendiculatus* gây ra, biểu hiện qua các nốt sần màu nâu đỏ như gỉ sắt trên bề mặt lá.

1.4. Ý nghĩa khoa học và thực tiễn

Về mặt khoa học, đề tài đóng góp vào việc nghiên cứu ứng dụng Học sâu (Deep Learning) và mạng nơ-ron tích chập (CNN) trong nông nghiệp thông minh. Đề tài kết hợp hai bài toán: phân loại bệnh (Classification) để xác định loại bệnh và Instance Segmentation để phân vùng chính xác vị trí vùng bệnh trên lá. Kết quả thực nghiệm về độ chính xác và khả năng phân vùng của mô hình YOLOv8 sẽ là nguồn tham khảo cho các nghiên cứu liên quan.

Về mặt thực tiễn, mô hình giúp nông dân phát hiện bệnh nhanh chóng, lựa chọn đúng loại thuốc bảo vệ thực vật và đánh giá mức độ nhiễm bệnh. Mô hình có thể được tích hợp vào ứng dụng di động, góp phần thúc đẩy chuyển đổi số trong nông nghiệp.

CHƯƠNG 2: MỤC TIÊU VÀ BÀI TOÁN ĐẶT RA

2.1. Phát biểu bài toán

Bài toán chính đặt ra là xây dựng hệ thống phân loại bệnh trên lá cây đậu sử dụng mạng nơ-ron tích chập (CNN). Cụ thể, với một ảnh lá đậu đầu vào, hệ thống cần xác định ảnh thuộc lớp nào trong 3 lớp: Healthy (lá khỏe mạnh), Angular Leaf Spot (bệnh đốm góc cạnh) và Bean Rust (bệnh gỉ sắt). Ngoài ra, để tài mở rộng thêm bài toán Instance Segmentation sử dụng YOLOv8 để xác định vị trí và phân vùng từng vùng bệnh trên lá.

2.2. Đầu vào và đầu ra

Đầu vào của hệ thống là ảnh màu RGB chụp lá cây đậu. Đầu ra của bài toán phân loại (Classification) là nhãn lớp và độ tin cậy tương ứng. Đầu ra của bài toán phân vùng (Instance Segmentation) bao gồm nhãn lớp, Bounding Box và Segmentation Mask cho từng vùng bệnh.

2.3. Phương pháp tiếp cận

Để tài sử dụng phương pháp Học sâu (Deep Learning) với hai hướng tiếp cận: Thứ nhất, xây dựng mô hình CNN để phân loại ảnh lá đậu vào 3 lớp tương ứng. Thứ hai, áp dụng mô hình YOLOv8-seg cho bài toán Instance Segmentation để xác định vị trí và hình dạng vùng bệnh. Cả hai phương pháp đều sử dụng kỹ thuật Transfer Learning và Data Augmentation để tăng hiệu quả huấn luyện.

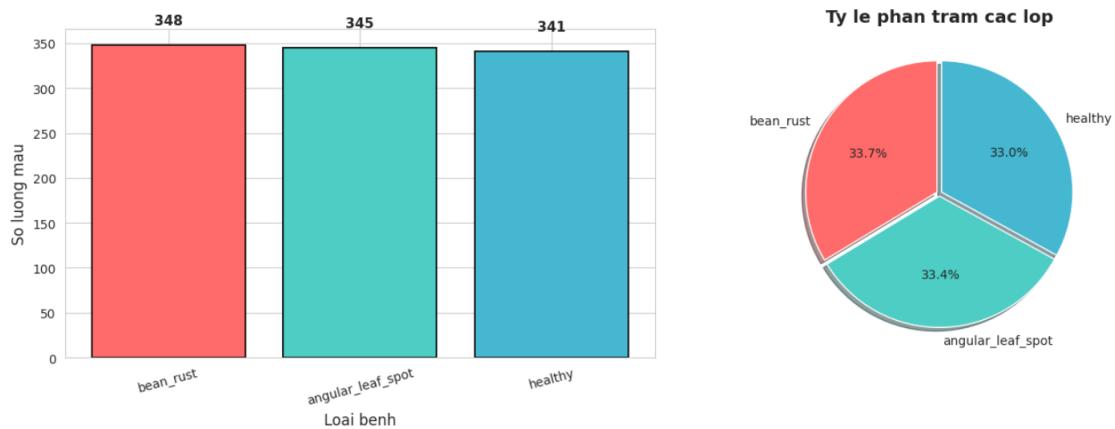
2.4. Tiêu chí đánh giá

Với bài toán phân loại, mô hình được đánh giá qua các chỉ số Accuracy, Precision, Recall và F1-Score. Với bài toán Instance Segmentation, hiệu quả được đánh giá thông qua mAP@50 và mAP@50-95 cho cả Bounding Box và Segmentation Mask.

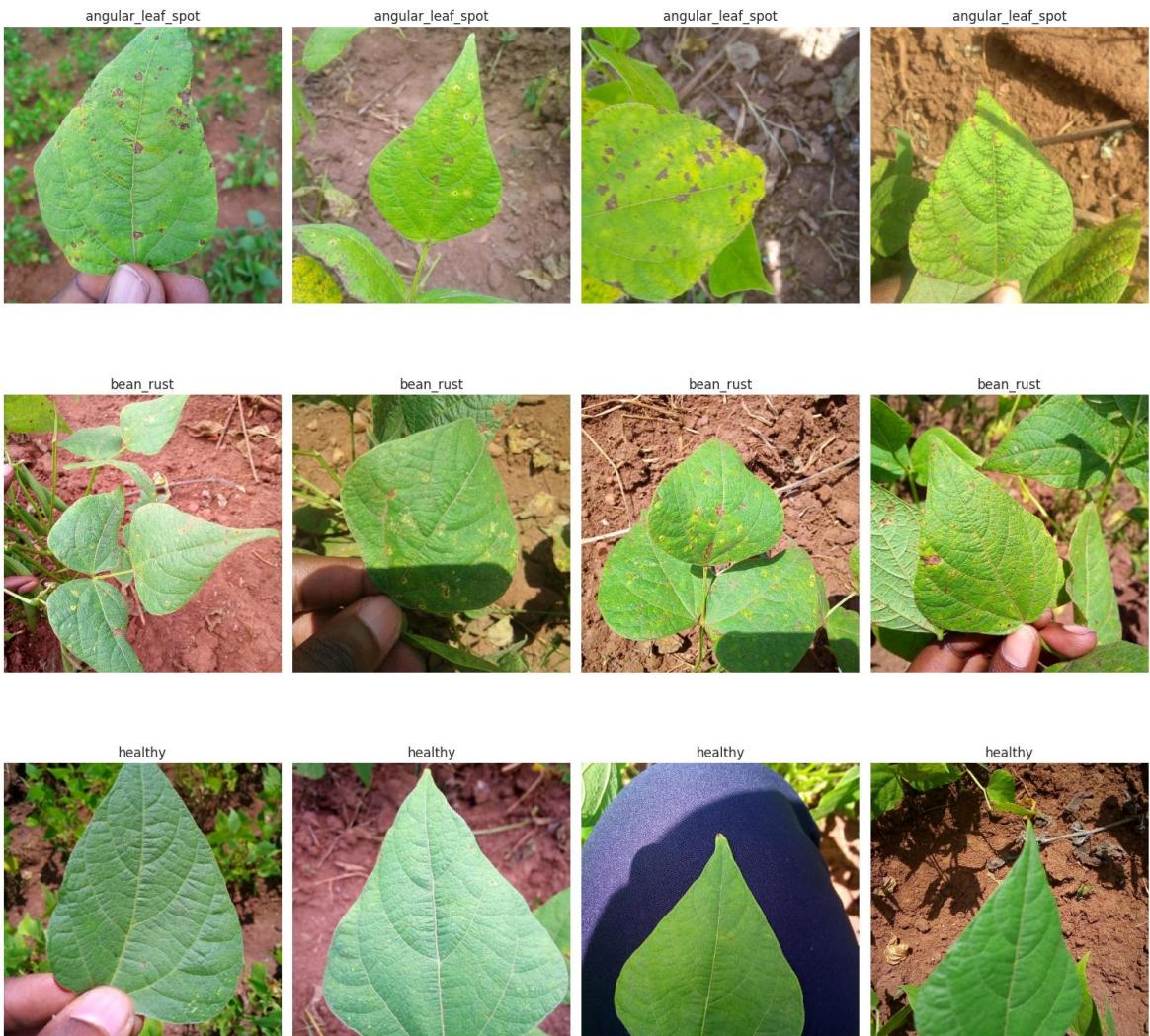
CHƯƠNG 3: MÔ TẢ DỮ LIỆU, TIỀN XỬ LÝ VÀ TĂNG CƯỜNG DỮ LIỆU NÂNG CAO

3.1. Mô tả dữ liệu

Bộ dữ liệu sử dụng trong đề tài bao gồm hình ảnh lá cây đậu được thu thập và phân loại thành 3 lớp: Healthy, Angular Leaf Spot và Bean Rust. Tổng cộng có 1,035 ảnh huấn luyện và khoảng 120 ảnh validation. Dữ liệu được phân bổ tương đối cân bằng giữa các lớp: Healthy (342 ảnh), Angular Leaf Spot (345 ảnh) và Bean Rust (348 ảnh). Sự cân bằng này giúp mô hình học tốt và không bị thiên lệch về một lớp cụ thể nào. Các ảnh được lưu trữ theo cấu trúc thư mục với mỗi lớp là một thư mục con, thuận tiện cho việc đọc và xử lý dữ liệu.



Hình 3.1 Ảnh phân bố các lớp trong dữ liệu



Hình 3.2 Ảnh minh họa đặc trưng của 3 lớp

Bức ảnh hiển thị một lối gác 12 hình ảnh cận cảnh của lá cây đậu, được sắp xếp thành 3 hàng và 4 cột, nhằm minh họa các tình trạng sức khỏe khác nhau của cây. Hàng trên cùng, được dán nhãn "angular_leaf_spot" (bệnh đốm góc), cho thấy những chiếc lá bị tổn thương với các vết đốm màu vàng và nâu loang lổ rải rác trên bề mặt. Hàng ở giữa, với nhãn "bean_rust" (bệnh rỉ sắt), hiển thị các lá có những đốm nhỏ li ti màu nâu đỏ hoặc cam, đặc trưng của nấm bệnh. Hàng cuối cùng được dán nhãn "healthy" (khỏe mạnh), bao gồm các hình ảnh lá đậu có màu xanh tươi, bề mặt lá láng mịn và không có dấu hiệu của sâu bệnh. Đây có vẻ là một tập dữ liệu mẫu được sử dụng để huấn luyện các mô hình học máy (Machine Learning) trong việc nhận diện và phân loại bệnh trên lá cây trồng.

3.2. Tiền xử lý dữ liệu

Quá trình tiền xử lý dữ liệu bao gồm các bước sau: Đầu tiên, ảnh được đọc từ đường dẫn và chuyển về định dạng RGB. Tiếp theo, ảnh được resize về kích thước cố định phù hợp với đầu vào của mô hình CNN. Cuối cùng, giá trị pixel được chuẩn

hóa (rescale) từ khoảng [0, 255] về khoảng [0, 1] bằng cách chia cho 255. Việc chuẩn hóa này giúp quá trình huấn luyện ổn định hơn và tăng tốc độ hội tụ của mô hình. Tùy vào mô chuẩn hóa sẽ khác nhau.

3.3. Tăng cường dữ liệu (Data Augmentation)

Để tăng tính đa dạng của dữ liệu huấn luyện và giảm overfitting, đề tài áp dụng các kỹ thuật tăng cường dữ liệu một cách bảo thủ (conservative) nhằm giữ nguyên các đặc trưng quan trọng của bệnh trên lá. Cụ thể, các kỹ thuật được sử dụng bao gồm:

Biến đổi hình học:

- Xoay ảnh trong phạm vi ± 10 độ (rotation_range=10) để mô phỏng các góc chụp khác nhau
- Dịch chuyển ngang và dọc $\pm 5\%$ (width_shift_range=0.05, height_shift_range=0.05)
- Phóng to/thu nhỏ $\pm 5\%$ (zoom_range=0.05)
- Lật ngang và lật dọc (horizontal_flip=True, vertical_flip=True)

Biến đổi màu sắc:

- Điều chỉnh độ sáng trong khoảng [0.9, 1.1] (brightness_range) để mô phỏng điều kiện ánh sáng khác nhau

Chế độ điền pixel:

- Sử dụng chế độ reflect (fill_mode='reflect') để điền các pixel trống sau khi biến đổi, giúp tránh các đường biên giả tạo

Lưu ý rằng các tham số augmentation được chọn ở mức thấp (conservative) để không làm biến dạng quá nhiều các đặc trưng bệnh trên lá như màu sắc vết bệnh, hình dạng đốm bệnh. Đối với tập validation, chỉ áp dụng chuẩn hóa (rescale=1./255) mà không augmentation để đánh giá chính xác hiệu quả của mô hình.

3.4. Đọc dữ liệu với ImageDataGenerator

Dữ liệu được đọc bằng ImageDataGenerator của Keras với phương thức flow_from_dataframe, cho phép đọc ảnh từ DataFrame chứa đường dẫn và nhãn. Các tham số quan trọng bao gồm: target_size để resize ảnh về kích thước cố định, batch_size để xác định số ảnh mỗi batch, class_mode='categorical' cho bài toán phân loại đa lớp, và shuffle=True để xáo trộn dữ liệu huấn luyện sau mỗi epoch.

CHƯƠNG 4: PHƯƠNG PHÁP THỰC HIỆN

4.1. Các mô hình học máy sử dụng

Trong đồ án này, chúng tôi tập trung phát triển và so sánh bốn kiến trúc mạng nơ-ron tích chập (CNN) hiện đại nhất để tối ưu hóa việc nhận diện khói u.

4.1.1. Mô hình MobileNetV3

Nguyên lý hoạt động: MobileNetV3Large là mạng nơ-ron tích chập (CNN) được thiết kế bởi Google dựa trên kỹ thuật Neural Architecture Search (NAS) - sử dụng AI để tự động tìm kiếm kiến trúc tối ưu. Mô hình hoạt động dựa trên Depthwise Separable Convolution, tách phép tích chập thành hai bước: Depthwise Convolution áp dụng một bộ lọc cho từng kênh riêng biệt, sau đó Pointwise Convolution kết hợp các kênh lại. Kỹ thuật này giảm 8-9 lần số phép tính so với convolution thông thường. Ngoài ra, mô hình tích hợp Squeeze-and-Excitation Block để học trọng số quan trọng của từng channel thông qua cơ chế attention, và sử dụng hàm kích hoạt h-swish thay cho ReLU để cải thiện độ chính xác mà vẫn đảm bảo tốc độ tính toán.

Lý do lựa chọn: MobileNetV3Large được lựa chọn vì ba lý do chính. Thứ nhất, mô hình có kích thước nhỏ gọn với chỉ 5.4 triệu tham số, phù hợp cho việc triển khai trên thiết bị di động và embedded systems. Thứ hai, tốc độ inference nhanh (khoảng 10ms) cho phép ứng dụng thời gian thực. Thứ ba, mô hình đã được huấn luyện sẵn trên ImageNet với độ chính xác 75.2%, có thể tận dụng các đặc trưng đã học để áp dụng cho bài toán phân loại bệnh lá đậu thông qua Transfer Learning, giúp đạt độ chính xác cao dù tập dữ liệu chỉ có khoảng 1000 ảnh.

Quy trình huấn luyện: Quy trình huấn luyện được thực hiện theo chiến lược 2 giai đoạn. Giai đoạn 1 (Phase 1) đóng băng toàn bộ base model MobileNetV3 và chỉ huấn luyện lớp phân loại (Classification Head) gồm GlobalAveragePooling, Dense 256, BatchNormalization, Swish activation, Dropout 0.3 và lớp Dense output. Learning rate được thiết lập với warmup trong 5 epoch đầu, sau đó giữ ở mức 1e-3. Giai đoạn 2 (Phase 2) mở khóa các layer cuối của base model (từ layer 50 trở đi) và fine-tune với learning rate thấp hơn (1e-5). Đặc biệt, tất cả các lớp BatchNormalization được giữ nguyên (frozen) để bảo toàn thống kê mean và variance đã học từ ImageNet, tránh overfitting.

Tinh chỉnh siêu tham số (Hyperparameter Tuning): Các siêu tham số được tinh chỉnh bao gồm: AdamW optimizer với weight decay 1e-2 giúp regularization mạnh hơn Adam thông thường, Label Smoothing 0.1 để giảm overconfidence và tăng khả năng tổng quát hóa, L2 regularization 1e-4 trong lớp Dense để hạn chế overfitting. Cơ chế Early Stopping với patience 8 epochs (Phase 1) và 15 epochs (Phase 2) giúp dừng huấn luyện khi không còn cải thiện. ReduceLROnPlateau giảm learning rate đi một nửa khi val_loss không giảm sau 3-5 epochs liên tiếp, với min_lr 1e-7. Dropout 0.3 được áp dụng trước lớp output để ngẫu nhiên tắt 30% neurons trong quá trình training.

4.1.2. Mô hình DeiT

Nguyên lý hoạt động: DeiT3 (Data-efficient Image Transformer) là kiến trúc Vision Transformer được phát triển bởi Facebook AI Research, áp dụng cơ chế Self-Attention từ xử lý ngôn ngữ tự nhiên vào thị giác máy tính. Ảnh đầu vào 384×384 được chia thành các patch 16×16 pixels, mỗi patch được chuyển thành vector embedding và thêm Position Embedding để giữ thông tin vị trí. Các patch embeddings sau đó đi qua nhiều lớp Transformer Encoder với cơ chế Multi-Head Self-Attention, cho phép mỗi patch "nhìn thấy" tất cả các patch khác trong ảnh. Phiên bản DeiT3-Small có 22 triệu tham số với 12 Transformer blocks, sử dụng kỹ thuật distillation từ mô hình lớn hơn để đạt hiệu quả cao trên dataset nhỏ.

Lý do lựa chọn: DeiT3 được lựa chọn vì khả năng học các mối quan hệ toàn cục (global context) trong ảnh tốt hơn CNN truyền thống. Trong khi CNN chỉ "nhìn" trong vùng receptive field giới hạn, Transformer có thể kết nối trực tiếp các vùng xa nhau trên ảnh, phù hợp cho bài toán phân loại bệnh cây khi các đặc trưng bệnh có thể xuất hiện ở nhiều vị trí khác nhau trên lá. Ngoài ra, mô hình đã được huấn luyện sẵn trên ImageNet-1K với độ chính xác cao (83%+), có thể tận dụng Transfer Learning cho dataset nhỏ. Việc sử dụng Cross-Validation 5-Fold giúp đánh giá mô hình một cách công bằng và ổn định hơn.

Quy trình huấn luyện: Quy trình huấn luyện sử dụng 5-Fold Stratified Cross-Validation trên tập dữ liệu gộp (train + validation). Mỗi fold, 80% dữ liệu dùng để train và 20% để validate, với phân bố cân bằng giữa các lớp nhờ Stratified splitting. Model được khởi tạo mới cho mỗi fold từ pretrained weights. Quá trình training áp

dụng EMA (Exponential Moving Average) với decay 0.9998 để tạo model ổn định hơn cho validation. Learning rate được điều chỉnh theo Cosine Annealing với Warmup: tăng dần trong các epoch đầu, sau đó giảm theo hàm cosine. Early Stopping được áp dụng với patience để dừng khi không còn cải thiện.

Tinh chỉnh siêu tham số (Hyperparameter Tuning): Các siêu tham số chính bao gồm: AdamW optimizer với weight decay để regularization, Label Smoothing 0.1 trong CrossEntropyLoss để giảm overconfidence. Gradient Clipping được áp dụng để tránh gradient explosion trong Transformer. Learning rate scheduler sử dụng Cosine Annealing với warmup epochs, công thức $lr = 0.5 \times (1 + \cos(\pi \times \text{progress}))$ sau giai đoạn warmup. EMA (Exponential Moving Average) với decay cao (0.9998) giúp model validation ổn định hơn bằng cách lấy trung bình các weights theo thời gian. Kết quả cuối cùng được tổng hợp qua Mean Accuracy và Standard Deviation từ 5 folds để đánh giá độ ổn định của mô hình.

4.1.3. Mô hình EfficientNetB3

Nguyên lý hoạt động: EfficientNet-B3 là kiến trúc CNN được phát triển bởi Google sử dụng kỹ thuật Compound Scaling - tối ưu đồng thời ba chiều: độ sâu (depth), độ rộng (width) và độ phân giải (resolution) của mạng. Mô hình sử dụng Mobile Inverted Bottleneck Convolution (MBConv) làm building block chính, kết hợp Depthwise Separable Convolution và Squeeze-and-Excitation để giảm số phép tính mà vẫn giữ hiệu suất cao. EfficientNet-B3 có 12 triệu tham số, kích thước ảnh đầu vào 300×300 , đạt độ chính xác 81.6% trên ImageNet - cao hơn ResNet-152 (77.8%) trong khi chỉ sử dụng 1/6 số tham số.

Lý do lựa chọn: EfficientNet-B3 được lựa chọn vì sự cân bằng tốt giữa độ chính xác và hiệu suất tính toán. So với các biến thể nhỏ hơn (B0, B1, B2), B3 có khả năng biểu diễn đặc trưng phức tạp hơn, phù hợp cho bài toán phân loại bệnh cây với các đặc trưng tinh tế. So với các biến thể lớn hơn (B4-B7), B3 tiết kiệm bộ nhớ và thời gian huấn luyện hơn. Mô hình đã được huấn luyện sẵn trên ImageNet với 14 triệu ảnh, có thể tận dụng Transfer Learning để học nhanh trên dataset nhỏ khoảng 1000 ảnh.

Quy trình huấn luyện: Quy trình huấn luyện sử dụng chiến lược Full Fine-tuning - mở khóa toàn bộ các layer của EfficientNet-B3 và huấn luyện cùng với lớp classifier mới. Lớp classifier được thay thế từ 1000 classes (ImageNet) xuống số classes của

bài toán (3 classes). Mỗi epoch, model đi qua toàn bộ training set, tính loss với CrossEntropyLoss, lan truyền ngược gradient và cập nhật weights. Sau mỗi epoch, model được đánh giá trên validation set. Early Stopping với patience 5 epochs được áp dụng để dừng khi validation loss không giảm, tránh overfitting. Model tốt nhất (val_loss thấp nhất) được lưu lại.

Tinh chỉnh siêu tham số (Hyperparameter Tuning): Các siêu tham số chính bao gồm: AdamW optimizer với learning rate 3e-4 và weight decay 1e-2 để regularization. CrossEntropyLoss được sử dụng làm hàm mất mát cho bài toán phân loại đa lớp. Learning rate được điều chỉnh theo CosineAnnealingLR với T_max=30, giảm dần learning rate theo hàm cosine từ giá trị ban đầu xuống gần 0, sau đó tăng lại (cyclic). Early Stopping class tự định nghĩa theo dõi validation loss, lưu checkpoint khi loss giảm, và dừng huấn luyện nếu không cải thiện sau 5 epochs liên tiếp. Số epochs tối đa là 30, đủ để mô hình hội tụ với Early Stopping bảo vệ.

4.1.4 Mô hình CNN tự xây

Nguyên lý hoạt động: BeanLeafVGG là mô hình CNN tự xây dựng lấy cảm hứng từ kiến trúc VGG, sử dụng các VGGBLOCK liên tiếp. Mỗi VGGBLOCK gồm 2 lớp Convolution 3×3 kết hợp BatchNormalization và ReLU, sau là MaxPooling 2×2 . Kiến trúc tuân theo nguyên lý "phẫu ngược": tăng dần số channels ($32 \rightarrow 64 \rightarrow 128 \rightarrow 256 \rightarrow 512$) trong khi giảm kích thước spatial ($400 \times 400 \rightarrow 200 \rightarrow 100 \rightarrow 50 \rightarrow 25 \rightarrow 12$). Khác với VGG gốc sử dụng lớp Fully Connected khổng lồ (4096 neurons), mô hình này dùng Global Average Pooling để giảm số tham số đáng kể, chỉ giữ lại vector 512 chiều trước lớp output.

Lý do lựa chọn: Mô hình CNN tự xây dựng được chọn để so sánh với các pretrained models và hiểu rõ hơn cách CNN hoạt động. VGG-style được chọn vì tính đơn giản và hiệu quả: chỉ sử dụng Conv 3×3 và MaxPool 2×2 mà vẫn đạt kết quả tốt. Việc thay thế FC layers bằng Global Average Pooling giúp giảm overfitting và số tham số từ hàng chục triệu xuống còn vài triệu. Mô hình này cho phép kiểm soát hoàn toàn kiến trúc, dễ debug và điều chỉnh cho bài toán cụ thể.

Quy trình huấn luyện: Quy trình huấn luyện từ scratch với 80 epochs, không sử dụng pretrained weights. Mỗi epoch, model đi qua training set, tính CrossEntropyLoss với Label Smoothing 0.1, lan truyền ngược gradient với Gradient Clipping (max_norm=1.0) để tránh gradient explosion, sau đó cập nhật weights bằng AdamW optimizer. Learning rate được điều chỉnh theo OneCycleLR với warm-up 30% epochs đầu, tăng dần từ lr thấp lên max_lr=2e-3, sau đó giảm theo cosine. Early Stopping với patience 15 epochs giúp dừng khi validation loss không cải thiện, model tốt nhất được lưu checkpoint.

Tinh chỉnh siêu tham số (Hyperparameter Tuning): Các siêu tham số chính bao gồm: AdamW optimizer với learning rate 3e-4 và weight decay 0.02 để

regularization mạnh. CrossEntropyLoss với Label Smoothing 0.1 giúp mô hình không quá tự tin vào predictions, tăng generalization. OneCycleLR scheduler với max_lr=2e-3 và pct_start=0.3 cho phép warm-up 30% epochs đầu rồi giảm dần. Gradient Clipping với max_norm=1.0 giới hạn độ lớn gradient, tránh training không ổn định. Dropout 0.5 trong classifier giúp regularization. Early Stopping patience=15 cho phép model đủ thời gian tìm optimum trước khi dừng.

4.1.5 Mô hình YOLOv8n với bài toán Instance Segmentation

Giới Thiệu Bài Toán: Bài toán instance segmentation cho lá đậu là một ứng dụng quan trọng trong nông nghiệp thông minh, yêu cầu xác định chính xác từng pixel thuộc vùng bệnh nào trên lá. Khác với classification chỉ phân loại toàn bộ ảnh hoặc object detection chỉ vẽ bounding box, instance segmentation cung cấp khả năng đếm chính xác số lượng vùng bệnh, đo diện tích từng vùng nhiễm để đánh giá mức độ nghiêm trọng, phân biệt các vùng bệnh chòng chéo, và hỗ trợ nông dân đưa ra quyết định điều trị dựa trên phân tích định lượng. Notebook này tập trung phát hiện và phân đoạn ba loại tình trạng: angular_leaf_spot (đốm góc lá với các đốm lớn hình góc cạnh), bean_rust (giấy sắt đậu với đốm nhỏ màu nâu đỏ), và healthy (vùng lá khỏe mạnh không có dấu hiệu bệnh).

Dữ Liệu: Dataset từ Roboflow Universe bao gồm 3,363 ảnh lá đậu được gán nhãn polygon tỉ mỉ, phân chia thành train (~70%, 2,354 ảnh), validation (~20%, 673 ảnh), và test (~10%, 336 ảnh). Mỗi ảnh thường chứa nhiều instances chòng chéo, phản ánh thực tế lá bệnh trong tự nhiên. Dataset có sự mất cân bằng đáng kể: angular_leaf_spot chiếm 63.8%, bean_rust 33.6%, và healthy chỉ 2.6%. Sự mất cân bằng này phản ánh thực tế rằng một lá bệnh thường chứa 10-15 vùng tổn thương nhỏ cần phân đoạn riêng, trong khi lá khỏe mạnh được segment toàn bộ như một instance duy nhất. Dataset ở format YOLOv8 chuẩn với file data.yaml và polygon annotations đã normalize về [0,1], sẵn sàng cho training mà không cần xử lý phức tạp.

Tiền Xử Lý Dữ Liệu: Dataset được download từ Roboflow đã ở format YOLOv8 sẵn sàng sử dụng, do đó quá trình tiền xử lý tương đối đơn giản. Các bước chuẩn bị bao gồm kiểm tra và xóa thư mục cũ để tránh conflict, download dataset với Roboflow API, và đọc file data.yaml để lấy cấu hình classes cùng đường dẫn. Giai đoạn khám phá dữ liệu (EDA) bao gồm verify số lượng ảnh và labels trong mỗi split, đếm số instances của từng class để hiểu mức độ imbalance, visualize phân bố classes qua biểu đồ cột, và hiển thị ảnh mẫu với polygon annotations để kiểm tra chất lượng

labeling. Ảnh được giữ nguyên kích thước gốc vì YOLOv8 sẽ tự động resize về 640×640 trong quá trình training. Polygon coordinates đã được normalize nên không cần xử lý thêm, và không cần thực hiện manual augmentation vì YOLOv8 đã tích hợp sẵn augmentation pipeline mạnh mẽ trong training loop.

Kiến Trúc YOLOv8-seg: YOLOv8-seg là kiến trúc tiên tiến kết hợp object detection và instance segmentation trong một mạng thống nhất, thực hiện cả hai nhiệm vụ song song trong một lần forward pass. Kiến trúc bao gồm ba thành phần chính: Backbone sử dụng CSPDarknet với C2f modules để trích xuất features từ ảnh ở nhiều scales khác nhau, Neck sử dụng PAN (Path Aggregation Network) để kết hợp thông tin từ các feature maps đa tỷ lệ, và Head gồm hai nhánh song song - detection head dự đoán bounding boxes và classification, segmentation head tạo mask coefficients kết hợp với prototype masks. Quy trình hoạt động diễn ra như sau: ảnh đầu vào 640×640 pixels được đưa qua Backbone tạo feature pyramids, PAN xử lý và tăng cường thông tin đa tỷ lệ, detection head dự đoán vị trí và class của mỗi object, segmentation head tạo mask coefficients, cuối cùng coefficients kết hợp tuyến tính với prototype masks để tạo instance masks chính xác. Ưu điểm lớn nhất của kiến trúc này là khả năng xử lý realtime với độ chính xác cao, đặc biệt phù hợp cho mobile apps và hệ thống giám sát tự động.

Lý Do Chọn YOLOv8n-seg: YOLOv8n-seg (nano variant) được lựa chọn dựa trên cân nhắc kỹ lưỡng giữa hiệu suất và tài nguyên khả dụng. Với chỉ 3.2M parameters, đây là model nhẹ nhất trong dòng YOLOv8 nhưng vẫn đảm bảo độ chính xác chấp nhận được, đặc biệt quan trọng khi cần deploy trên smartphone hoặc edge devices. Tốc độ inference đạt 4.9ms mỗi ảnh tương đương khoảng 200 FPS, đủ nhanh cho các ứng dụng realtime mà không cần phân cứng đất tiền. Model được pretrained trên COCO dataset với hàng triệu ảnh đa dạng, cung cấp nền tảng features extraction mạnh mẽ giúp transfer learning hiệu quả sang domain lá đậu. So với các alternatives như Mask R-CNN (chậm hơn 10 lần), Detectron2 (phức tạp trong deployment), hay các variants lớn hơn như YOLOv8s/m/l (nặng hơn 2-10 lần), YOLOv8n-seg đạt được sweet spot giữa speed, accuracy và resource efficiency. Đặc biệt trong bối cảnh nông nghiệp Việt Nam với smartphone tầm trung và tài nguyên hạn chế, việc chọn model nhẹ là yếu tố quyết định để ứng dụng có thể triển khai rộng rãi.

Chiến Lược Training: Chiến lược training được thiết kế để tối ưu hiệu suất model trong khi tránh overfitting và tận dụng tối đa dữ liệu có hạn. Training được cấu hình với 100 epochs tối đa kèm early stopping patience 20 epochs, nghĩa là nếu validation metrics không cải thiện trong 20 epochs liên tiếp thì training sẽ tự động dừng. Batch size 16 cân bằng giữa tốc độ training và memory usage, image size 640×640 là chuẩn YOLOv8, optimizer ở chế độ auto cho phép tự động chọn AdamW với learning rate scheduling phù hợp. Checkpoint được lưu mỗi 10 epochs và best model tự động lưu dựa trên validation mAP. Điểm quan trọng nhất là data augmentation cực kỳ mạnh mẽ để tăng tính đa dạng: color augmentation với HSV adjustment (hue $\pm 1.5\%$, saturation $\pm 70\%$, value $\pm 40\%$) mô phỏng các điều kiện ánh sáng khác nhau, geometric augmentation gồm rotation $\pm 10^\circ$, translation $\pm 10\%$, scale $\pm 50\%$, shear $\pm 5^\circ$, và horizontal flip 50%, cùng với hai kỹ thuật advanced là Mosaic (100% - ghép 4 ảnh thành 1) và MixUp (10% - trộn 2 ảnh). Toàn bộ chiến lược này giúp model không chỉ học thuộc training data mà thực sự hiểu đặc điểm bệnh trong mọi điều kiện thực tế.

Tinh Chỉnh Tham Số: Các tham số được tinh chỉnh dựa trên quan sát kỹ lưỡng metrics trong quá trình training và validation. Confidence threshold được đặt ở 0.25 - đủ thấp để không bỏ sót các vùng bệnh nhỏ nhưng đủ cao để lọc bỏ false positives. IoU threshold cho NMS được đặt ở 0.6, giúp loại bỏ các bounding boxes trùng lặp trong khi vẫn giữ lại các instances riêng biệt gần nhau. Training được thực hiện với 4 workers cho data loading, pretrained weights từ COCO, và seed=42 với deterministic=True để đảm bảo reproducibility. Quá trình monitoring tập trung vào các loss curves (box, seg, cls, dfl) và validation metrics (Precision, Recall, mAP@0.5, mAP@0.5:0.95), với early stopping dựa trên validation mAP để tự động dừng khi model không còn cải thiện.

Kết Quả: Model hội tụ tốt sau 48 epochs với best checkpoint tại epoch 28, cho thấy early stopping hoạt động hiệu quả. Training curves cho thấy tất cả loss giảm đều đặn và không có dấu hiệu overfitting khi train loss và validation loss song song nhau. Kết quả tổng thể trên validation set đạt Box mAP@0.5 khoảng 68%, Mask mAP@0.5 khoảng 68%, và Mask mAP@0.5:0.95 đạt 48.4% - là kết quả khá tốt cho bài toán instance segmentation phức tạp với nhiều small objects chồng chéo. Phân tích per-

class cho thấy healthy class đạt kết quả xuất sắc với Mask mAP@0.5 lên tới 93% nhờ boundary rõ ràng và diện tích lớn, angular_leaf_spot đạt 65% - kết quả khá tốt do là lớp đa số trong training data, trong khi bean_rust chỉ đạt 42% với recall thấp nhất 37.9% do các đốm giását rất nhỏ và màu sắc tương đồng với nền lá. Inference speed đạt 4.9ms/ảnh tương đương 200 FPS, hoàn toàn đủ nhanh cho ứng dụng realtime trên mobile và edge devices, sẵn sàng cho deployment trong thực tế.

4.2. Grad Cam trực quan hóa giải thích mô hình

Grad-CAM là một kỹ thuật thuộc lĩnh vực Explainable AI (trí tuệ nhân tạo có thể giải thích được), được sử dụng rộng rãi để hiểu cách các mô hình học sâu, đặc biệt là mạng nơ-ron tích chập (CNN), đưa ra quyết định phân loại. Kỹ thuật này tạo ra một bản đồ nhiệt (heatmap) trực quan, cho phép người dùng nhìn thấy những vùng nào trong ảnh đầu vào mà mô hình đã "chú ý" nhiều nhất khi đưa ra dự đoán.

Về nguyên lý hoạt động, Grad-CAM tính toán gradient của điểm số lớp được dự đoán so với các feature maps của lớp tích chập cuối cùng trong mạng. Sau đó, các gradient này được lấy trung bình toàn cục để thu được trọng số tương ứng cho mỗi feature map. Cuối cùng, các feature maps được nhân với trọng số và cộng lại, rồi áp dụng hàm ReLU để loại bỏ các giá trị âm, tạo ra một heatmap thể hiện mức độ quan trọng của từng vùng ảnh đối với quyết định phân loại.

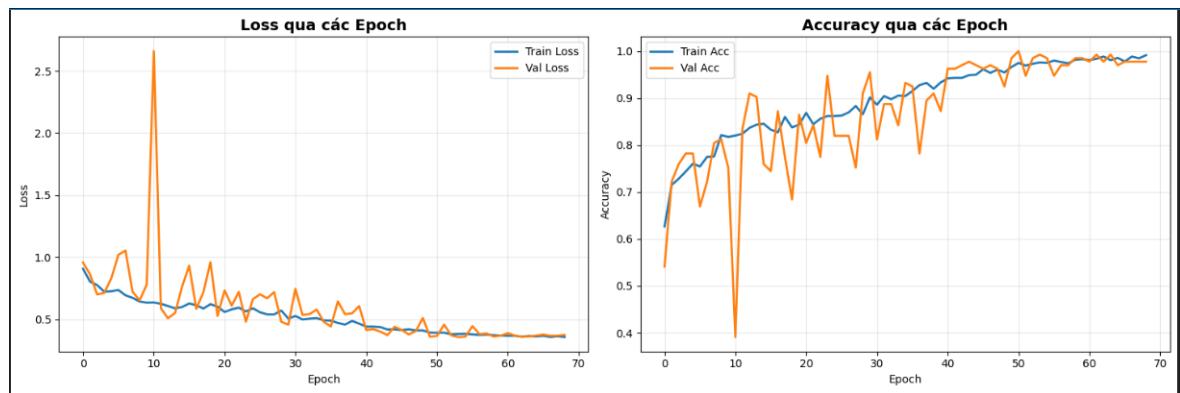
Trong bối cảnh dự án phân loại bệnh lá đậu của bạn, Grad-CAM đóng vai trò quan trọng trong việc xác minh tính đúng đắn của mô hình. Cụ thể, nó giúp kiểm tra xem mô hình có thực sự nhìn vào vùng lá bị bệnh để đưa ra quyết định hay không, thay vì học các đặc trưng không liên quan như nền ảnh. Ngoài ra, kỹ thuật này còn hữu ích trong việc debug mô hình khi phát hiện các trường hợp dự đoán sai, đồng thời cung cấp khả năng giải thích trực quan cho người dùng cuối về lý do mô hình đưa ra kết quả phân loại cụ thể. Trên heatmap, các vùng có màu đỏ hoặc cam biểu thị những khu vực mà mô hình chú ý nhiều nhất, trong khi các vùng có màu xanh hoặc tím thể hiện những khu vực ít quan trọng hơn trong quá trình ra quyết định.

CHƯƠNG 5: KẾT QUẢ VÀ ĐÁNH GIÁ CHI TIẾT CÁC MÔ HÌNH

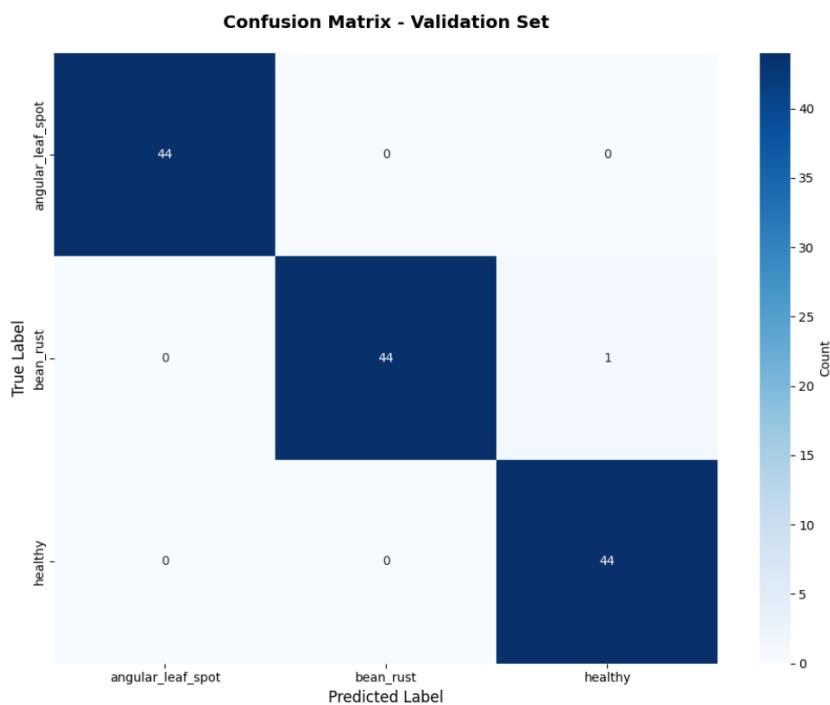
Trong phần này, chúng tôi trình bày kết quả thực nghiệm chi tiết cho từng kiến trúc mô hình. Mỗi mô hình được đánh giá thông qua biểu đồ quá trình huấn luyện (Training History) và Ma trận nhầm lẫn (Confusion Matrix) để làm rõ ưu nhược điểm riêng biệt.

5.1. Kết quả Mô hình BeanLeafVGG tự xây

Mô hình CNN tự xây với độ chính xác đạt khoảng 99.25%



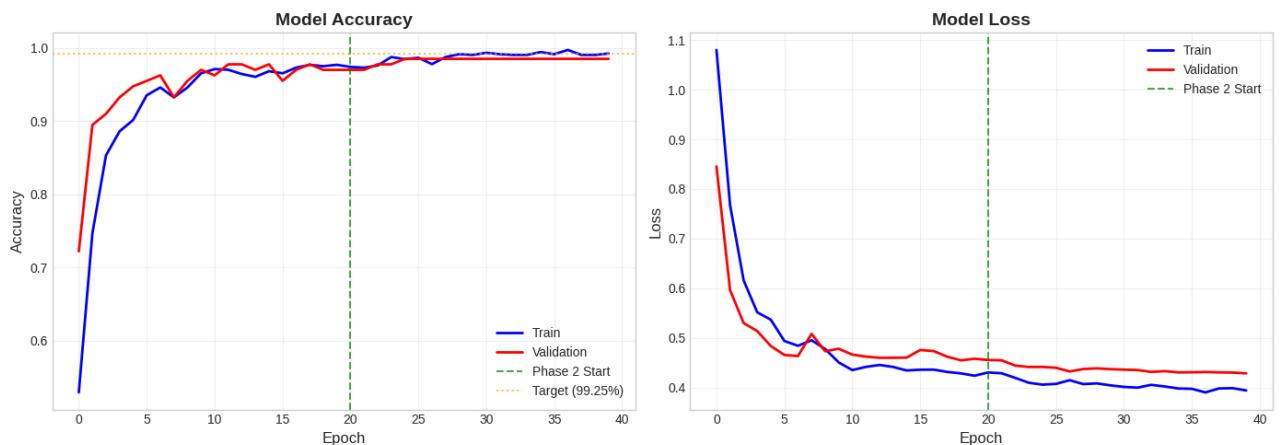
Hình 5.1 Biểu đồ history train BeanLeafVGG



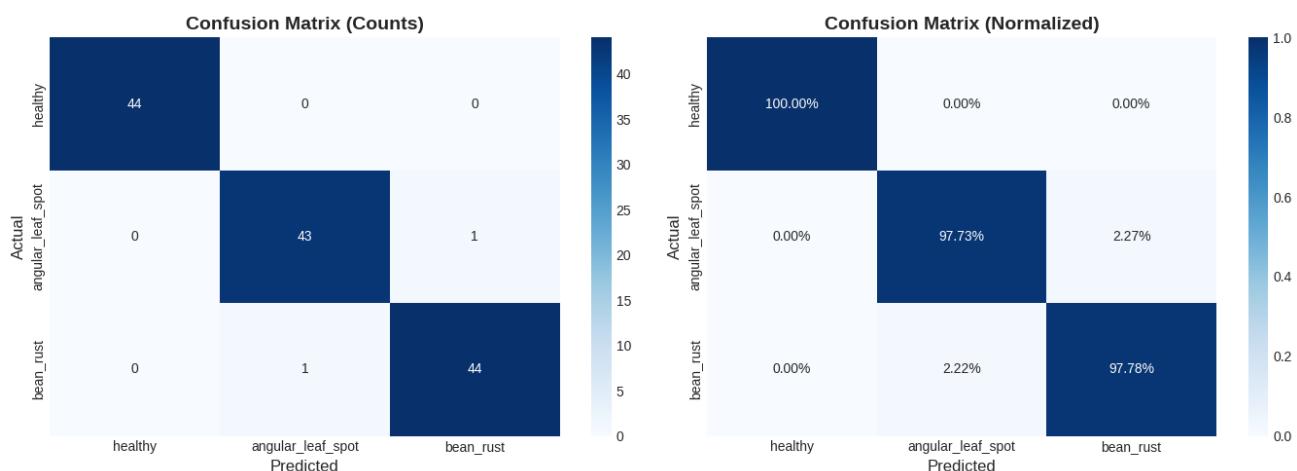
Hình 5.2 Biểu đồ Confusion matrix BeanLeafVGG

Mô hình VGG tự xây dựng (Custom VGG) đạt hiệu suất xuất sắc với độ chính xác gần như tuyệt đối (~99.3%), thể hiện qua Confusion Matrix chỉ có duy nhất 1 ca dự đoán sai (Bean Rust → Healthy) trên tổng số mẫu kiểm thử. Biểu đồ huấn luyện cho thấy sự hội tụ chật chẽ giữa Train và Validation, chứng minh chiến lược Data Augmentation kết hợp với Label Smoothing đã xử lý triệt để vấn đề Overfitting. Mặc dù Validation Loss có biến động mạnh (spike) ở giai đoạn đầu (do đặc tính "warm-up" của OneCycleLR), nhưng mô hình đã ổn định rất tốt ở các epoch cuối. Đây là kết quả rất khả quan và đủ độ tin cậy để đóng gói triển khai thực tế.

5.2. Kết quả Mô hình MobilenetV3



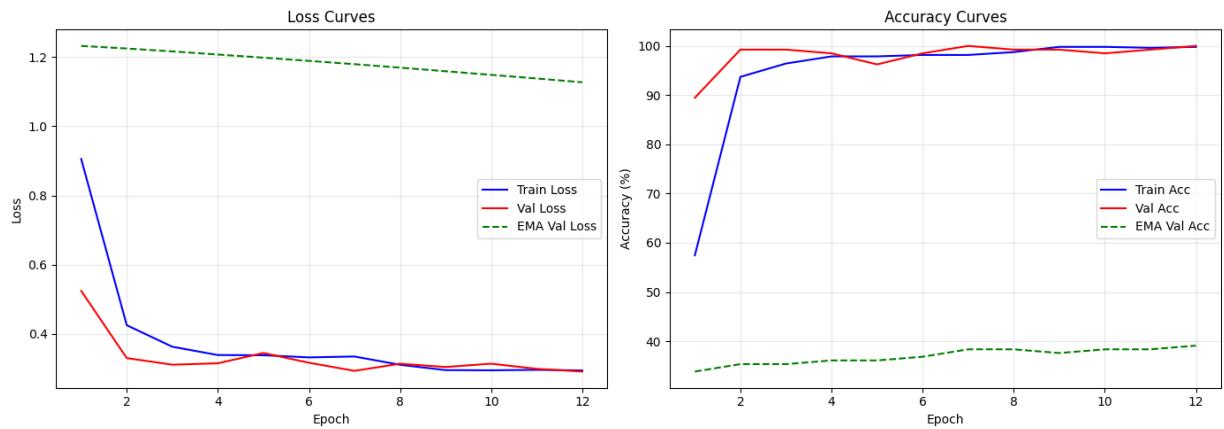
Hình 5.4 Biểu đồ history train MobilenetV3



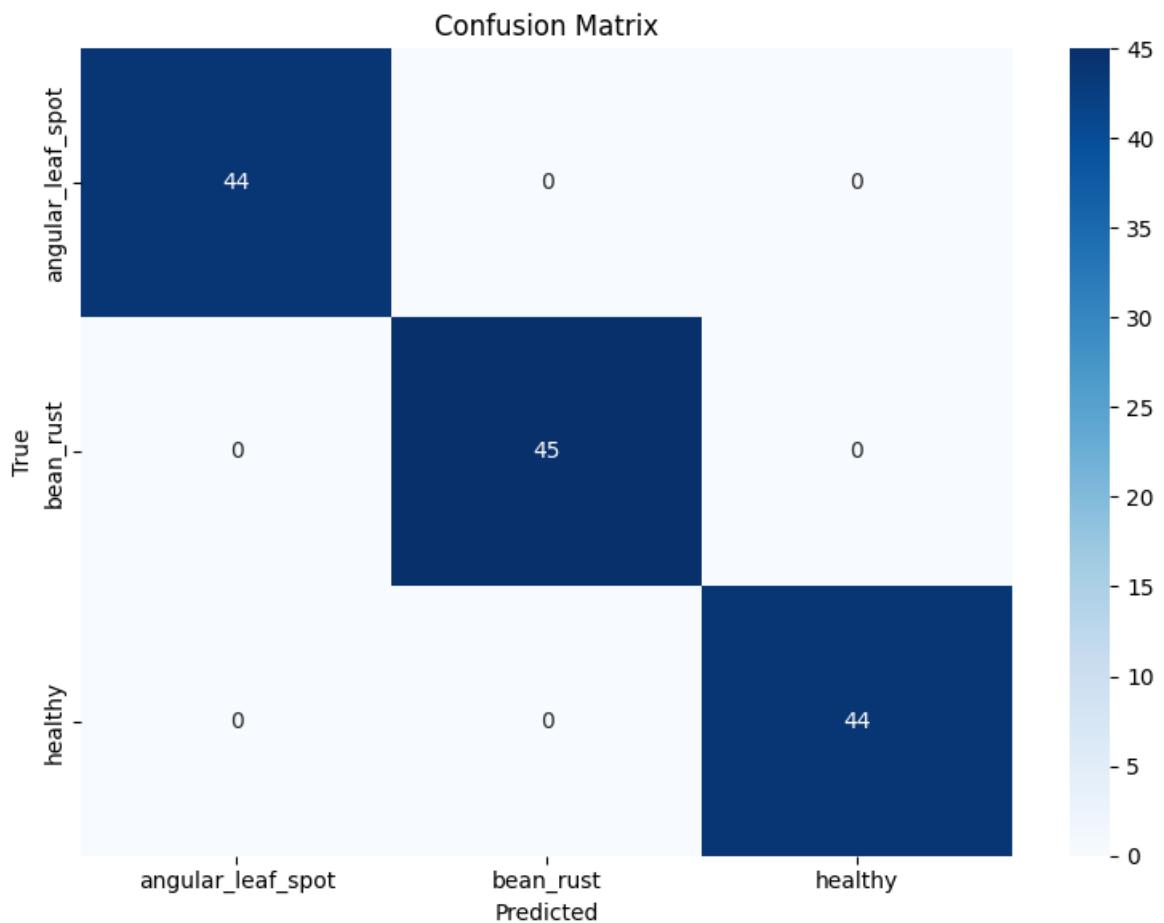
Hình 5.5 Biểu đồ confusion matrix MobilenetV3

Chiến lược huấn luyện 2 giai đoạn (Transfer Learning kết hợp Fine-tuning) đã phát huy hiệu quả vượt trội, đưa độ chính xác Validation đạt đỉnh 98.50% và duy trì ổn định qua kiểm chứng K-Fold ($96.83\% \pm 1.47\%$). Biểu đồ Loss và Accuracy hội tụ rất tốt, khoảng cách giữa Train và Validation nhỏ chứng tỏ mô hình không bị overfitting nhờ Data Augmentation và Label Smoothing hợp lý. Ma trận nhầm lẫn cho thấy khả năng phân loại tuyệt đối với lớp Healthy (100%), sai sót chỉ xảy ra tối thiểu (2 mẫu) giữa Angular Leaf Spot và Bean Rust do sự tương đồng cao về hình thái đốm bệnh nhỏ.

5.3. Kết quả Mô hình Deit



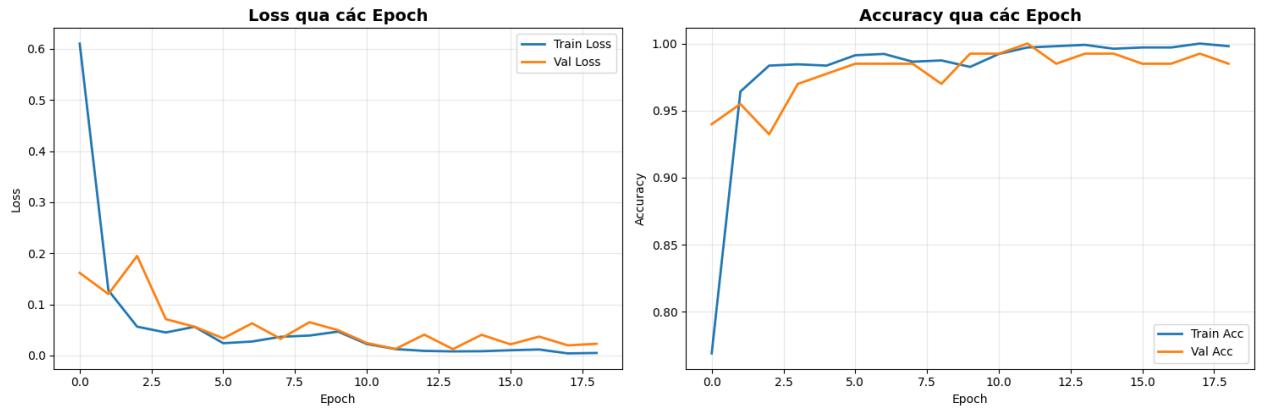
Hình 5.7 Biểu đồ history train Deit



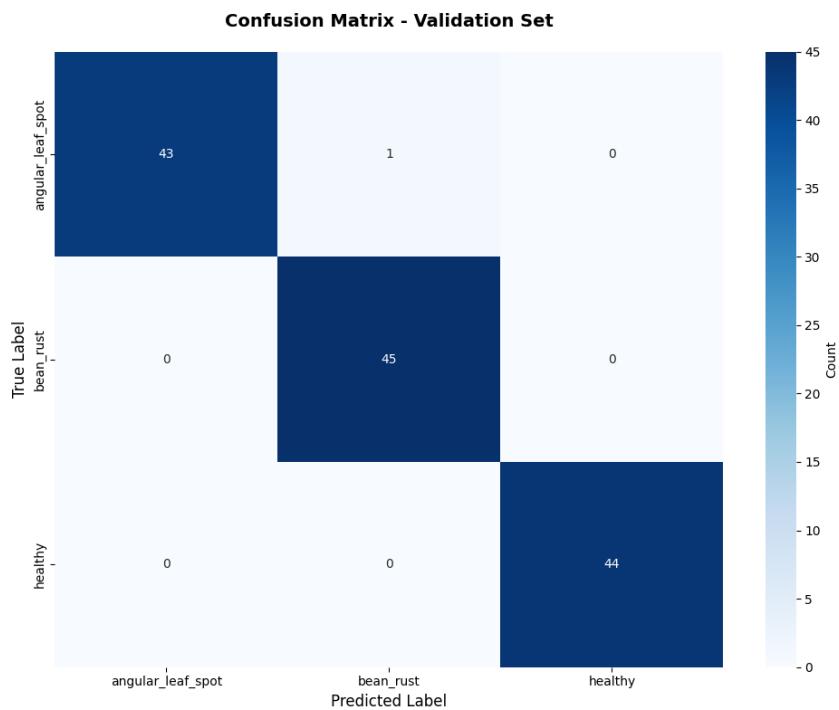
Hình 5.9 Biểu đồ Confusionmatrix Deit

Việc chuyển đổi sang kiến trúc DeiT (Vision Transformer) đã tạo ra bước nhảy vọt về hiệu suất với độ chính xác tuyệt đối 100% trên tập kiểm thử, vượt trội hoàn toàn so với các kiến trúc CNN truyền thống trước đó. Ma trận nhầm lẫn hiển thị kết quả hoàn hảo (zero errors), chứng minh cơ chế Self-Attention của Transformer cực kỳ hiệu quả trong việc nắm bắt các đặc trưng toàn cục và phân biệt tinh tế giữa các loại tổn thương lá. Biểu đồ huấn luyện hội tụ cực nhanh ngay từ epoch thứ 2, cho thấy sự tương thích tuyệt vời giữa pre-trained weights và bộ dữ liệu hiện tại. Đây là kết quả SOTA (State-of-the-art) và là mô hình tối ưu nhất trong loạt thử nghiệm để đưa vào triển khai thực tế.

5.4. Kết quả mô hình EfficientNetB3



Hình 5.10 Biểu đồ history train EfficientnetB3



Hình 5.11 Biểu đồ confusion matrix EfficientnetB3

Mô hình EfficientNet-B3 (Transfer Learning) thể hiện hiệu năng xuất sắc và ổn định với độ chính xác kiểm thử đạt ~99.2%, thể hiện qua Confusion Matrix chỉ có duy nhất 1 mẫu dự đoán sai (nhầm lẫn nhỏ giữa Angular Leaf Spot và Bean Rust) trên

tổng số 133 mẫu. Biểu đồ huấn luyện cho thấy sự hội tụ rất mượt mà, đường Loss giảm đều và khoảng cách giữa Train/Val rất nhỏ, chứng tỏ chiến lược Cosine Annealing LR và Data Augmentation (RandomErasing, Distortion) đã kiểm soát overfitting cực kỳ hiệu quả.

5.8. So sánh Kết quả và Thảo luận (Comparative Analysis)

Để có cái nhìn tổng quan, chúng tôi tổng hợp các chỉ số hiệu năng của cả 4 mô hình vào bảng so sánh dưới đây. Các số liệu được ghi nhận trên cùng một tập dữ liệu kiểm thử (Valid Set) để đảm bảo tính công bằng.

Bảng 5.1. Bảng so sánh tổng hợp các chỉ số hiệu năng

Mô hình	Accuracy	Precision	Recall	F1	Số tham số
BeanLeafVGG	99.25%	99%	99%	99%	4.715.747
DeiT	100%	100%	100%	100%	21.821.571
Efficient-B3	99.25%	99%	99%	99%	~13.000.000
MobileNetV3	98.5%	98%	98%	98%	3.169.019

Dựa trên bảng so sánh tổng hợp các chỉ số hiệu năng, có thể khẳng định rằng cả bốn mô hình đều đạt kết quả rất cao, phản ánh chất lượng dữ liệu tốt và quy trình huấn luyện được thiết kế hợp lý. Mô hình DeiT thể hiện hiệu năng vượt trội nhất khi đạt Accuracy, Precision, Recall và F1 đều bằng 100%, cho thấy khả năng phân biệt các lớp gần như tuyệt đối trên tập kiểm thử. Điều này chứng minh ưu thế của kiến trúc Transformer trong việc khai thác ngữ cảnh toàn cục của ảnh, đặc biệt hiệu quả với các đặc trưng phân bố rải rác trên bề mặt lá. Tuy nhiên, điểm hạn chế của DeiT là số lượng tham số lớn (hơn 21 triệu), kéo theo chi phí huấn luyện và suy luận cao, ít phù hợp với các hệ thống hạn chế tài nguyên.

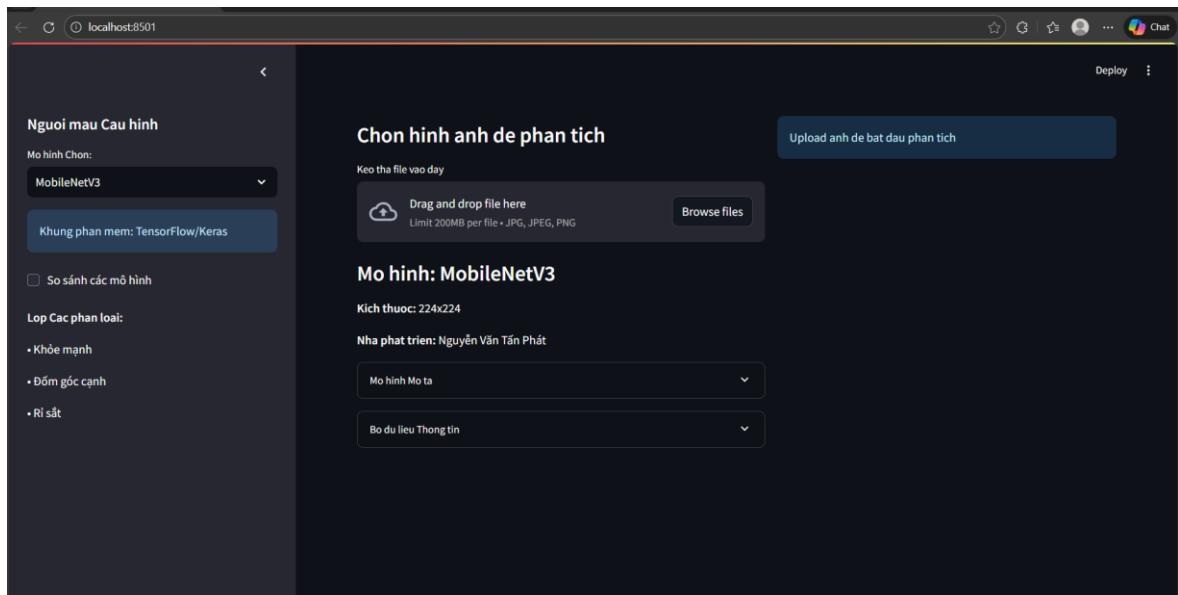
Mô hình BeanLeafVGG tự xây dựng đạt Accuracy 99.25% với Precision, Recall và F1 đều xấp xỉ 99%, cho thấy hiệu quả rất cao dù không sử dụng pretrained weights. Với chỉ khoảng 4.7 triệu tham số, mô hình này chứng minh rằng một kiến trúc CNN được thiết kế và regularization tốt vẫn có thể cạnh tranh với các mô hình hiện đại.

EfficientNet-B3 cũng đạt hiệu năng tương đương BeanLeafVGG nhưng với số tham số lớn hơn (~13 triệu), đổi lại là tính ổn định và khả năng tổng quát hóa tốt nhờ kiến trúc đã được kiểm chứng rộng rãi. Trong khi đó, MobileNetV3 có độ chính xác thấp nhất (98.5%) nhưng lại sở hữu kích thước nhỏ gọn nhất, rất phù hợp cho triển khai trên thiết bị di động hoặc hệ thống nhúng. Nhìn chung, kết quả thực nghiệm cho thấy sự đánh đổi rõ ràng giữa độ chính xác và độ phức tạp mô hình, và việc lựa chọn mô hình tối ưu cần dựa trên yêu cầu cụ thể của bài toán triển khai thực tế.

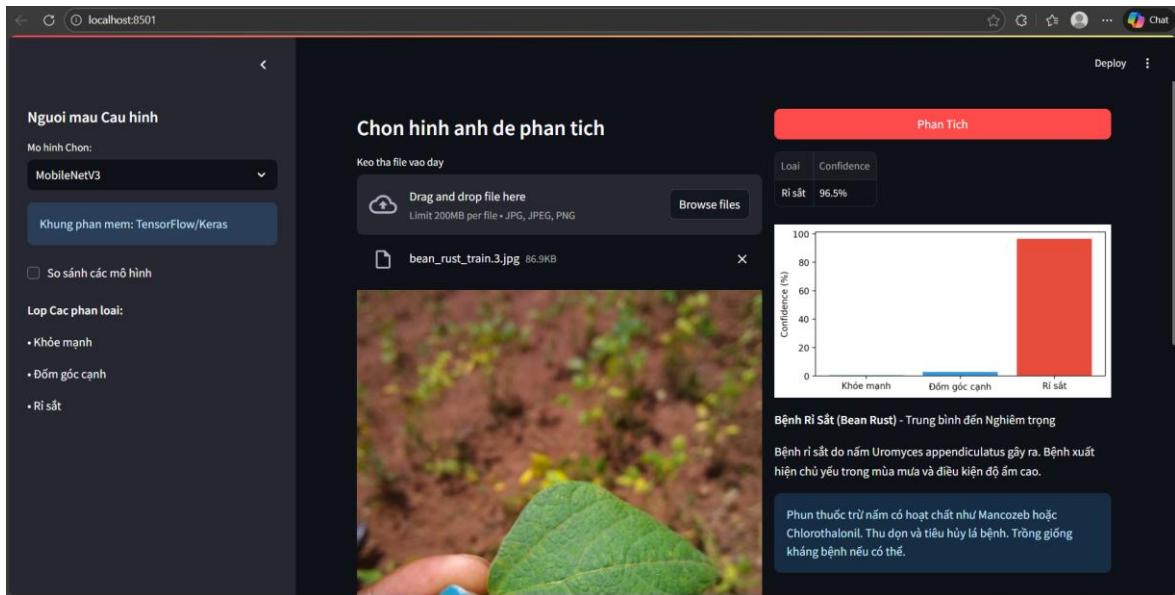
Bảng 5.2 Kết quả khi chạy CV với fold bằng 5

Mô hình	Độ chính xác
MobileNetV3	96.57% ± 0.80%
DeIT	98.29% (+/- 0.47%)
CNN tự xây	0.971

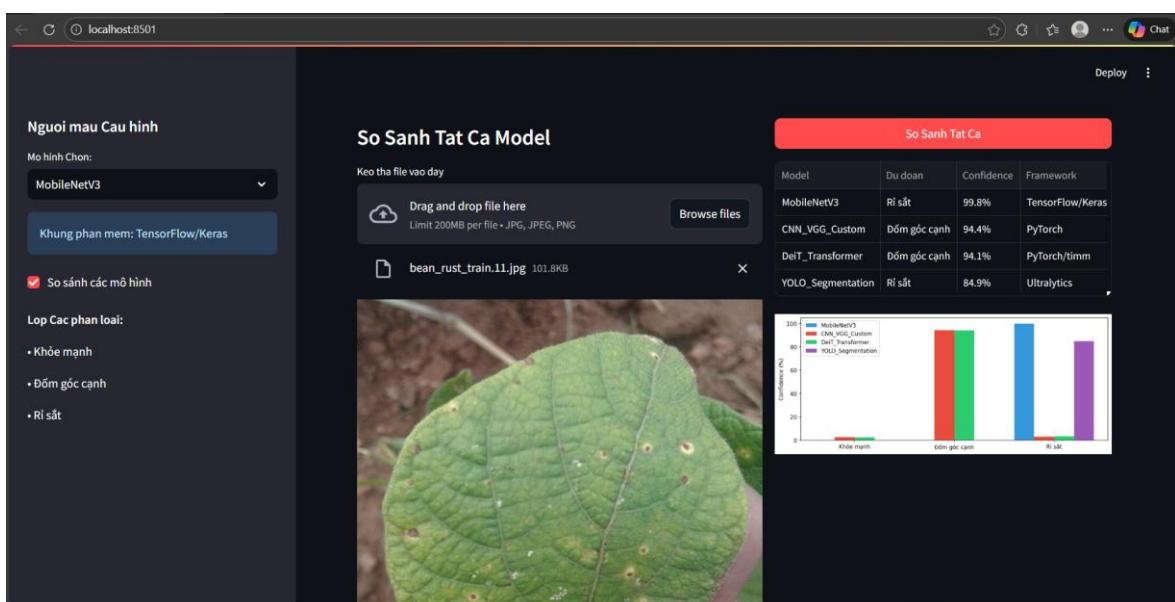
5.9 Giao diện WEBSITE



Hình 5.12 Giao diện khi bắt đầu



Hình 5.13 Giao diện khi dự đoán



Hình 5.14 Giao diện so sánh các mô hình

CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

6.1. Kết luận

Trong khuôn khổ đề tài, hệ thống phân loại bệnh lá đậu (Bean Leaf Lesions Classification) dựa trên các mô hình học sâu đã được xây dựng, huấn luyện và đánh giá một cách toàn diện. Kết quả thực nghiệm cho thấy các kiến trúc CNN và Transformer đều đạt hiệu năng rất cao, với Accuracy dao động từ 98.5% đến 100%, khẳng định tính khả thi của việc ứng dụng Deep Learning vào bài toán nhận diện bệnh cây trồng. Đặc biệt, mô hình DeiT đạt kết quả tuyệt đối trên tập kiểm thử, thể hiện ưu thế của cơ chế Self-Attention trong việc khai thác mối quan hệ toàn cục giữa các vùng tổn thương trên lá. Bên cạnh đó, mô hình BeanLeafVGG tự xây dựng đạt độ chính xác tương đương các mô hình pretrained nhưng với số tham số thấp, chứng minh rằng việc thiết kế kiến trúc phù hợp kết hợp với chiến lược regularization và data augmentation hiệu quả có thể mang lại hiệu suất cao ngay cả khi huấn luyện từ đầu.

Ngoài độ chính xác, đề tài cũng làm rõ sự đánh đổi giữa hiệu năng và chi phí tính toán. MobileNetV3 tuy có độ chính xác thấp hơn nhưng lại rất phù hợp cho các kịch bản triển khai trên thiết bị hạn chế tài nguyên, trong khi EfficientNet-B3 thể hiện sự cân bằng tốt giữa độ chính xác, độ ổn định và khả năng mở rộng. Nhìn chung, các kết quả thu được cho thấy hệ thống đề xuất đáp ứng tốt yêu cầu phân loại bệnh lá đậu, có độ tin cậy cao và tiềm năng ứng dụng thực tiễn trong nông nghiệp thông minh.

6.2. Hướng phát triển

Trong tương lai, đề tài có thể được mở rộng theo nhiều hướng nhằm nâng cao tính ứng dụng và giá trị khoa học. Thứ nhất, cần đánh giá mô hình trên các bộ dữ liệu lớn hơn, đa dạng hơn về điều kiện ánh sáng, góc chụp và giai đoạn phát triển của lá nhằm kiểm chứng khả năng tổng quát hóa trong môi trường thực tế. Thứ hai, việc kết hợp bài toán phân loại với phân vùng vùng bệnh (segmentation) hoặc phát hiện đối tượng (object detection) sẽ giúp không chỉ xác định loại bệnh mà còn định vị chính xác vị trí và mức độ nhiễm bệnh trên lá, từ đó hỗ trợ ra quyết định canh tác hiệu quả hơn.

Bên cạnh đó, các hướng nghiên cứu nâng cao như học đa nhiệm (multi-task learning), học bán giám sát (semi-supervised learning) hoặc tự giám sát (self-supervised learning) có thể được áp dụng để giảm phụ thuộc vào dữ liệu gán nhãn thủ công. Cuối cùng, việc tối ưu mô hình cho triển khai thực tế, bao gồm nén mô hình (model compression), quantization và triển khai trên thiết bị di động hoặc hệ thống IoT nông nghiệp, sẽ là bước quan trọng để đưa kết quả nghiên cứu từ môi trường thí nghiệm vào ứng dụng sản xuất, góp phần thúc đẩy chuyển đổi số trong lĩnh vực nông nghiệp chính xác.

TÀI LIỆU THAM KHẢO

1. He, K., Zhang, X., Ren, S., & Sun, J. (2016). *Deep Residual Learning for Image Recognition*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778.
2. Chollet, F. (2017). *Xception: Deep Learning with Depthwise Separable Convolutions*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1251–1258.
→ Nguồn cho Xception + CNN hiện đại
3. Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*. Proceedings of the IEEE International Conference on Computer Vision (ICCV), pp. 618–626.
→ Nguồn cho Grad-CAM và giải thích mô hình
4. Nguyễn Thanh Bình (2021). *Học sâu và ứng dụng trong xử lý ảnh y sinh*. Nhà xuất bản Khoa học và Kỹ thuật, Hà Nội.
→ Nguồn tiếng Việt, bao quát CNN + ảnh y tế