TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN NGÀNH CÔNG NGHỆ THÔNG TIN

MÔN KHOA HỌC DỮ LIỆU ẢNH: NỀN TẢNG VÀ ỨNG DỤNG



BÁO CÁO HỆ THỐNG NHẬN DIỆN LÁ CÂY THÔNG MINH TRÊN THIẾT BỊ DI ĐỘNG

GVHD: TS Dương Việt Hằng

SV thực hiện: Hồ Hoàng Việt Tiến 230201056

Bùi Thị Thanh Hồng 230201043

Tháng 12 Năm 2024

**Mục lục**

[1 Giới thiệu 3](#_Toc185847393)

[2 Ứng dựng nhận diện lá cây trong thực tế 3](#_Toc185847394)

[2.1 Nông nghiệp thông minh 3](#_Toc185847395)

[2.2 Giáo dục và nghiên cứu 3](#_Toc185847396)

[3 Phương pháp và thuật toán nhận diện 4](#_Toc185847397)

[3.1 CONVOLUTIONAL NEURAL NETWORK 4](#_Toc185847398)

[3.2 EFFICIENT NETS: 8](#_Toc185847399)

[3.3 PHƯƠNG PHÁP THỰC HIỆN 10](#_Toc185847400)

[4 Thách thức trong nhận diện lá cây 11](#_Toc185847401)

[4.1 Đa dạng về đặc điểm của lá cây 12](#_Toc185847402)

[4.2 Ảnh hưởng của môi trường 12](#_Toc185847403)

[4.3 Hiệu suất trên thiết bị di động 12](#_Toc185847404)

[5 Giao diện ứng dụng: 12](#_Toc185847405)

[6 Kết quả 13](#_Toc185847406)

[6.1 Triển khai thành công trên thiết bị di động: 13](#_Toc185847407)

[6.2 Độ chính xác cao (trên 90%): 13](#_Toc185847408)

[6.3 Khả năng hoạt động tốt trong môi trường thực tế: 13](#_Toc185847409)

[6.4 So sánh hiệu quả với các mô hình CNN khác: 13](#_Toc185847410)

[7 Kết luận và tính nhân rộng 13](#_Toc185847411)

[7.1 Kết luận 13](#_Toc185847412)

[7.2 Tính Nhân rộng 14](#_Toc185847413)

# Giới thiệu

Bài toán nhận diện lá cây đã được nghiên cứu rộng rãi với nhiều phương pháp, trong đó CNN truyền thống đạt nhiều kết quả đáng kể. Đề tài này sử dụng EfficientNet để phát triển ứng dụng nhận diện lá trên thiết bị di động, nhằm so sánh hiệu suất của EfficientNet với CNN truyền thống về tốc độ xử lý, độ chính xác và khả năng triển khai trên nền tảng di động.

# Ứng dựng nhận diện lá cây trong thực tế

## Nông nghiệp thông minh

Trong nông nghiệp, việc nhận diện chính xác các loại cây trồng, cây dại, hoặc các loại bệnh trên lá cây là yếu tố quan trọng giúp nâng cao năng suất và chất lượng sản phẩm. Hệ thống nhận diện lá cây thông minh mang lại nhiều lợi ích:

* + 1. Xác định nhanh loại cây trồng:
       - Hỗ trợ nông dân và kỹ thuật viên xác định các giống cây trồng chỉ bằng cách chụp ảnh lá cây, thay vì dựa vào kiến thức kinh nghiệm truyền thống.
       - Giảm thời gian và chi phí cần thiết để tham vấn các chuyên gia thực vật học.
    2. Phát hiện sớm bệnh lá:
       - Kết hợp với cơ sở dữ liệu về các bệnh thường gặp trên lá cây, hệ thống có thể cảnh báo nông dân về các dấu hiệu bệnh như đốm lá, cháy lá, hoặc vàng lá.
       - Đưa ra gợi ý về các biện pháp phòng ngừa và xử lý phù hợp, tối ưu hóa quy trình chăm sóc cây trồng.
    3. Hỗ trợ phân loại cỏ dại:
       - Nhận diện nhanh cỏ dại xâm lấn, từ đó hỗ trợ nông dân kiểm soát và loại bỏ, đảm bảo cây trồng phát triển tốt hơn.

## Giáo dục và nghiên cứu

* + 1. Hỗ trợ học sinh, sinh viên:
       - Làm công cụ học tập trực quan giúp học sinh, sinh viên nhận diện và phân biệt các loại cây, học hỏi về các đặc điểm sinh học của từng loài thực vật.
       - Tạo ra hứng thú học tập thông qua trải nghiệm thực tế khi sử dụng ứng dụng trên thiết bị di động.
    2. Hỗ trợ nghiên cứu thực vật:
       - Giúp các nhà khoa học và chuyên gia thực vật học nhanh chóng nhận diện và phân loại các loài cây trong tự nhiên mà không cần mang theo các tài liệu cồng kềnh.
       - Tích hợp với cơ sở dữ liệu thực vật để lưu trữ và chia sẻ thông tin về các loài mới được phát hiện.

# Phương pháp và thuật toán nhận diện

## CONVOLUTIONAL NEURAL NETWORK

CNN (Convolutional Neural Network) là một loại mạng nơ-ron nhân tạo phổ biến, đặc biệt hiệu quả trong các bài toán về xử lý ảnh và nhận diện đối tượng. CNN được thiết kế với kiến trúc gồm nhiều lớp (layers) giúp tự động học các đặc trưng (features) từ dữ liệu đầu vào qua từng tầng, từ đó tạo ra khả năng nhận diện và phân loại đối tượng. Kiến trúc của CNN bao gồm các loại lớp chính sau:

* Convolutional Layer: Đây là lớp lõi của CNN, có nhiệm vụ thực hiện phép tích chập (con- volution) trên ảnh đầu vào để trích xuất các đặc trưng như cạnh, góc và các chi tiết quan trọng. Tầng này giúp CNN phát hiện các đặc điểm riêng biệt của hình ảnh qua từng lớp.
* Pooling Layer: Lớp này thường đi sau Convolutional Layer và có nhiệm vụ giảm kích thước của các đặc trưng, giúp giảm số lượng tham số và tính toán trong mô hình. Các phép pooling phổ biến như Max Pooling sẽ chọn giá trị lớn nhất trong vùng lân cận, giữ lại các đặc điểm quan trọng trong khi loại bỏ nhiễu.
* Fully Connected Layer: Đây là lớp kết nối đầy đủ, thường nằm ở cuối mô hình CNN, có nhiệm vụ sử dụng các đặc trưng đã học từ các lớp trước để phân loại hoặc dự đoán kết quả cuối cùng. Lớp này hoạt động giống như mạng nơ-ron truyền thống (MLP), tổng hợp thông tin từ các đặc trưng và đưa ra quyết định cuối cùng.

Sự kết hợp của các lớp Convolutional, Pooling và Fully Connected giúp CNN tự động học và tối ưu các đặc trưng từ ảnh đầu vào, làm cho CNN trở thành một công cụ mạnh mẽ trong các ứng dụng về nhận diện hình ảnh và phân loại đối tượng.

* + 1. CONVOLUTIONAL LAYER:

Convolutional Layer là lớp quan trọng nhất trong CNN, thực hiện phép tích chập giữa ảnh đầu vào và các bộ lọc (kernel) để tạo ra các feature maps. Quá trình này giúp trích xuất các đặc trưng quan trọng từ ảnh, như cạnh, đường viền, hoặc màu sắc.

Trong Convolutional Layer, mỗi bộ lọc sẽ di chuyển (quét) qua ảnh đầu vào theo một khoảng cách nhất định (stride) và áp dụng phép nhân tích chập trên các vùng ảnh. Kết quả của phép tích chập là một feature map, thể hiện các đặc trưng mà bộ lọc đó phát hiện được. Các đặc trưng có thể bao gồm:

* + - * Cạnh và đường viền: Các bộ lọc có thể nhận biết cạnh của các vật thể trong ảnh, phát hiện đường viền, giúp nhận diện hình dáng và kết cấu.
      * Màu sắc và mẫu hình: Một số bộ lọc có thể phát hiện màu sắc cụ thể hoặc các mẫu hình phức tạp hơn trong ảnh.

Mỗi bộ lọc trong Convolutional Layer học một loại đặc trưng khác nhau từ ảnh, nhờ vào quá trình tối ưu hóa trong quá trình huấn luyện. Khi kết hợp nhiều bộ lọc, mô hình có thể nhận biết và phân tích các đặc trưng từ mức độ đơn giản đến phức tạp, từ đó xây dựng một biểu diễn chi tiết cho ảnh đầu vào.

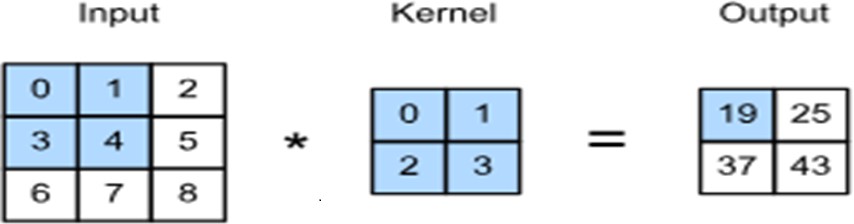


Figure 1: Convolutional Layer

* + 1. POOLING LAYER:

POOLING LAYER giúp làm giảm số lượng tham số cần học trong mô hình, từ đó giúp tăng khả năng tổng quát (generalization) của mô hình. Điều này có nghĩa là mạng sẽ học được các đặc trưng mạnh mẽ hơn, không bị nhạy cảm với sự thay đổi nhỏ trong dữ liệu đầu vào (ví dụ: vị trí của đối tượng trong hình ảnh).

Một số lý do tại sao Pooling Layer lại quan trọng: Giảm độ phức tạp tính toán: Giảm kích thước của dữ liệu đầu vào, giúp giảm thời gian tính toán.

Chống overfitting: Giảm số lượng tham số, giúp mô hình ít bị học quá mức trên dữ liệu huấn luyện.

Bất biến với dịch chuyển nhỏ: Pooling giúp mô hình học được các đặc trưng mà không bị ảnh hưởng bởi sự dịch chuyển nhỏ của đối tượng trong ảnh.

Một số loại pooling phổ biến: Max Pooling, Average Pooling

* + - * Max Pooling là phương pháp phổ biến nhất trong CNN. Với mỗi cửa sổ (window) có kích thước cố định (thường là 2x2 hoặc 3x3), Max Pooling chọn giá trị lớn nhất trong cửa sổ đó làm đại diện cho vùng đó.
      * Average Pooling hoạt động tương tự như Max Pooling, nhưng thay vì chọn giá trị lớn nhất, nó tính giá trị trung bình trong vùng được lọc.
    1. FULLY-CONNECTED:

Lớp Fully-Connected (FC) là lớp kết nối đầy đủ trong một mạng nơ-ron, có nhiệm vụ kết nối tất cả các đặc trưng đã được trích xuất từ các lớp trước đó và gộp lại để đưa ra kết quả cuối cùng của mô hình, như là phân loại, dự đoán hay nhận diện.

Trong mạng nơ-ron, lớp Fully-Connected có cấu trúc đặc biệt, trong đó mỗi nơ-ron trong lớp FC được kết nối với tất cả các nơ-ron từ lớp trước đó. Điều này có nghĩa là, đầu vào của mỗi nơ-ron trong lớp FC không chỉ phụ thuộc vào một đặc trưng cụ thể mà là sự kết hợp của tất cả các đặc trưng từ lớp trước đó. Quá trình này giúp mô hình học và kết hợp các đặc trưng đã trích xuất để tạo ra đầu ra chính xác hơn.

Cách thức hoạt động của lớp Fully-Connected:

* + - * Nhận đầu vào từ lớp trước: Mỗi nơ-ron trong lớp FC nhận đầu vào từ tất cả các nơ-ron của lớp trước đó. Đầu vào này là kết quả của quá trình tính toán và trích xuất đặc trưng từ các lớp convolutional và pooling trước đó.
      * Kết hợp thông qua trọng số và bias: Các nơ-ron trong lớp FC sẽ kết hợp đầu vào thông qua các trọng số (weights) và bias (độ lệch) để tính toán đầu ra. Trọng số xác định mức độ quan trọng của mỗi đầu vào, trong khi bias giúp điều chỉnh kết quả tính toán.
      * Tính toán đầu ra: Kết quả của phép tính này sẽ được chuyển qua một hàm kích hoạt (activation function) như ReLU, sigmoid, hoặc softmax để tính toán đầu ra của lớp FC, từ đó mô hình đưa ra kết quả cuối cùng.

Lớp Fully-Connected giúp mô hình học các mối quan hệ phức tạp giữa các đặc trưng đã trích xuất, giúp nâng cao khả năng phân loại hoặc dự đoán.

* + 1. Ưu điểm: CNN (Convolutional Neural Network) mang lại nhiều ưu điểm nổi bật nhờ khả năng tự động học và trích xuất đặc trưng từ dữ liệu mà không cần phải qua các bước tiền xử lý thủ công phức tạp. Các đặc điểm chính của CNN bao gồm:

1. Học đặc trưng tự động: CNN có khả năng học các đặc trưng từ dữ liệu đầu vào một cách tự động thông qua các lớp Convolutional và Pooling. Điều này giúp giảm thiểu công sức và thời gian cần thiết cho việc chuẩn bị dữ liệu thủ công, như cắt xén hoặc trích xuất đặc trưng.
2. Xác định mối quan hệ không gian giữa các điểm ảnh: Các lớp convolution giúp xác định mối quan hệ không gian giữa các điểm ảnh trong ảnh đầu vào. Điều này cho phép CNN học được các đặc trưng cục bộ, chẳng hạn như cạnh, hình dáng hoặc các mẫu hình đặc trưng của đối tượng.
3. Chia sẻ tham số: Trong CNN, các bộ lọc (kernel) được sử dụng chung cho tất cả các vùng trong ảnh, giúp giảm số lượng tham số cần phải học. Việc này không chỉ giảm thiểu tài nguyên tính toán mà còn giúp tránh hiện tượng overfitting (quá khớp), đồng thời cải thiện hiệu quả huấn luyện.
4. Khả năng tổng quát hóa tốt: CNN có khả năng nhận diện đối tượng bất kể các biến đổi như tỷ lệ, xoay hay dịch chuyển. Điều này giúp cải thiện khả năng nhận dạng trong nhiều tình huống và môi trường khác nhau, làm cho mô hình trở nên mạnh mẽ và linh hoạt trong các ứng dụng thực tế.
5. Ứng dụng rộng rãi: CNN có thể được ứng dụng trong nhiều bài toán khác nhau, bao gồm:
   * Phân loại ảnh: Phân loại ảnh thành các lớp khác nhau, chẳng hạn như nhận diện hình ảnh động vật, người, cảnh vật.
   * Nhận diện đối tượng: Xác định và phân loại các đối tượng cụ thể trong ảnh.
   * Phân đoạn ảnh: Xác định và phân chia các khu vực quan trọng trong ảnh.
   * Nhận diện khuôn mặt: Phát hiện và nhận diện các khuôn mặt trong ảnh hoặc video.

Nhờ những đặc điểm trên, CNN đã trở thành một công cụ mạnh mẽ và không thể thiếu trong lĩnh vực xử lý ảnh và học sâu (deep learning).

* + 1. Nhược điểm: CNN truyền thống như VGG và ResNet đã đạt được nhiều thành tựu ấn tượng trong các bài toán nhận diện ảnh, tuy nhiên, chúng vẫn tồn tại một số hạn chế:

1. Yêu cầu nhiều tham số và tài nguyên tính toán:
   * Các mô hình CNN truyền thống, đặc biệt là VGG và ResNet, yêu cầu số lượng tham số rất lớn, điều này dẫn đến việc tiêu tốn rất nhiều bộ nhớ và tài nguyên tính toán. Khi tăng độ sâu của mạng (số lượng lớp), số tham số tăng theo cấp số nhân, gây khó khăn trong việc triển khai và huấn luyện.
2. Cần điều chỉnh thủ công số lớp và kích thước:
   * Để đạt được hiệu suất cao, CNN truyền thống thường yêu cầu thử nghiệm và điều chỉnh thủ công số lượng lớp, kích thước của từng lớp và các tham số khác. Quá trình này đòi hỏi một lượng lớn thời gian và tài nguyên, đồng thời không đảm bảo tối ưu cho tất cả các trường hợp.
3. Không tối ưu về sự cân bằng giữa độ rộng, độ sâu và độ phân giải:
   * Các mạng CNN truyền thống như VGG và ResNet không tối ưu hóa sự cân bằng giữa độ sâu, độ rộng và độ phân giải của mô hình. Điều này có thể dẫn đến overfitting (quá khớp) nếu mô hình quá phức tạp hoặc không đủ khả năng tổng quát nếu quá đơn giản, khiến cho hiệu suất mô hình không ổn định.
4. Khó triển khai trên các thiết bị di động hoặc nhúng:
   * Vì yêu cầu về tài nguyên tính toán và bộ nhớ lớn, CNN truyền thống gặp khó khăn khi triển khai trên các thiết bị di động hoặc thiết bị nhúng, nơi tài nguyên hạn chế. Điều này khiến việc sử dụng các mô hình như VGG hoặc ResNet cho các ứng dụng di động trở nên không thực tế, vì chúng cần nhiều bộ nhớ và khả năng xử lý mạnh mẽ mà các thiết bị này không có.

## EFFICIENT NETS:

* + 1. Giới thiệu: EfficientNet là một kiến trúc mạng neural do Google phát triển, nhằm tối ưu hóa việc sử dụng tài nguyên trong các mạng nơ-ron tích chập (CNN). Mục tiêu của EfficientNet là đạt được hiệu quả cao nhất về tài nguyên tính toán, giúp mô hình vừa nhỏ gọn vừa đạt được độ chính xác tốt.

EfficientNet sử dụng phương pháp Compound Scaling để mở rộng mô hình một cách cân bằng và thông minh giữa ba yếu tố chính: độ sâu, chiều rộng, và độ phân giải. Thay vì chỉ tăng kích thước theo một chiều như trong nhiều mô hình truyền thống, Compound Scaling điều chỉnh đồng thời cả ba yếu tố này với các tham số tối ưu đã được nghiên cứu. Phương pháp này giúp mô hình mở rộng hiệu quả mà không làm tăng đột ngột yêu cầu về tài nguyên tính toán, đảm bảo cân bằng giữa hiệu suất và chi phí xử lý.

EfficientNet đã cho thấy khả năng vượt trội trong việc tối ưu hóa tài nguyên, phù hợp cho nhiều ứng dụng yêu cầu mô hình nhẹ, đặc biệt trên các thiết bị di động và nhúng.

* + 1. Compound Scaling Compound Scaling là phương pháp mở rộng mô hình được sử dụng trong kiến trúc EfficientNet để tối ưu hóa hiệu suất của mạng nơ-ron tích chập (CNN) mà vẫn đảm bảo hiệu quả sử dụng tài nguyên. Compound Scaling mở rộng mô hình đồng đều trên ba thành phần chính:
       - Depth (Độ sâu): Đây là độ sâu của mô hình, thể hiện qua số lượng các lớp (layers) trong mạng. Việc tăng độ sâu giúp mô hình học được nhiều đặc trưng phức tạp hơn, nhưng nếu quá sâu sẽ dẫn đến yêu cầu tài nguyên tính toán cao và có thể gây hiện tượng quá khớp (overfitting).
       - Width (Chiều rộng): Chiều rộng của mô hình là số lượng neurons trong mỗi lớp. Tăng chiều rộng giúp mô hình có khả năng lưu trữ nhiều thông tin hơn tại mỗi lớp, giúp tăng cường khả năng biểu diễn của mạng.
       - Resolution (Độ phân giải): Độ phân giải của ảnh đầu vào là kích thước của hình ảnh mà mô hình nhận được. Việc tăng độ phân giải giúp mô hình có thể phát hiện các chi tiết nhỏ trong ảnh, nhưng đồng thời cũng đòi hỏi nhiều tài nguyên tính toán hơn.

Thay vì chỉ tăng một trong ba yếu tố này, EfficientNet sử dụng Compound Scaling để mở rộng đồng đều cả ba yếu tố theo các tham số đã được tối ưu hóa. Cách tiếp cận này giúp mô hình đạt hiệu suất cao hơn mà không tiêu tốn quá nhiều tài nguyên tính toán, tạo ra sự cân bằng lý tưởng giữa hiệu quả và độ chính xác.

* + 1. Công thức: a. d là hệ số mở rộng độ sâu.

1. w là hệ số mở rộng chiều rộng.
2. r là hệ số mở rộng độ phân giải.
3. *α, β, γ* là các hệ số tỷ lệ
4. *ϕ* một tham số điều chỉnh (compound coefficient) để kiểm soát mức độ mở rộng tổng thể của mô hình.

**Depth:** *d* = *αϕ* **Width:** *w* = *βϕ* **Resolution:** *r* = *γϕ* **Subject to:**

*s.t.α · β*2 *· γ*2 *≈* 2

*α ≥* 1*, β ≥* 1*, γ ≥* 1

* + 1. Các phiên bản EfficientNet-B0: Đây là phiên bản cơ bản nhất của mô hình EfficientNet, sử dụng bộ tham số cơ sở với các giá trị =1, =1, =1. EfficientNet-B0 có độ sâu, độ rộng và độ phân giải tiêu chuẩn, là nền tảng để phát triển các phiên bản lớn hơn. Từ phiên bản này, các phiên bản tiếp theo được xây dựng thông qua quá trình mở rộng kích thước mô hình dựa trên bộ tham số đã được tối ưu hóa.

EfficientNet-B1: Phiên bản đầu tiên áp dụng kỹ thuật Compound Scaling. Nhóm tác giả đã tìm ra bộ tham số tối ưu khi cố định giá trị là 1, với =1.2, =1.1, và =1.15. Sau khi xác định được các giá trị tối ưu này, các tham số , , được giữ cố định làm hằng số. Bằng cách thay đổi giá trị của , mô hình sẽ được mở rộng một cách có hệ thống để thu được các phiên bản từ EfficientNet-B1 đến EfficientNet-B7, mỗi phiên bản sẽ tăng dần kích thước và độ phức tạp theo một tỷ lệ nhất định.

Quá trình mở rộng EfficientNet từ B0 đến B7 giúp mô hình đạt được sự cân bằng tốt nhất giữa độ chính xác và hiệu suất tính toán. Compound Scaling cho phép EfficientNet tối ưu hóa tài nguyên bằng cách mở rộng đồng thời theo ba chiều (độ rộng, độ sâu, độ phân giải), thay vì chỉ mở rộng một chiều như các mô hình truyền thống.

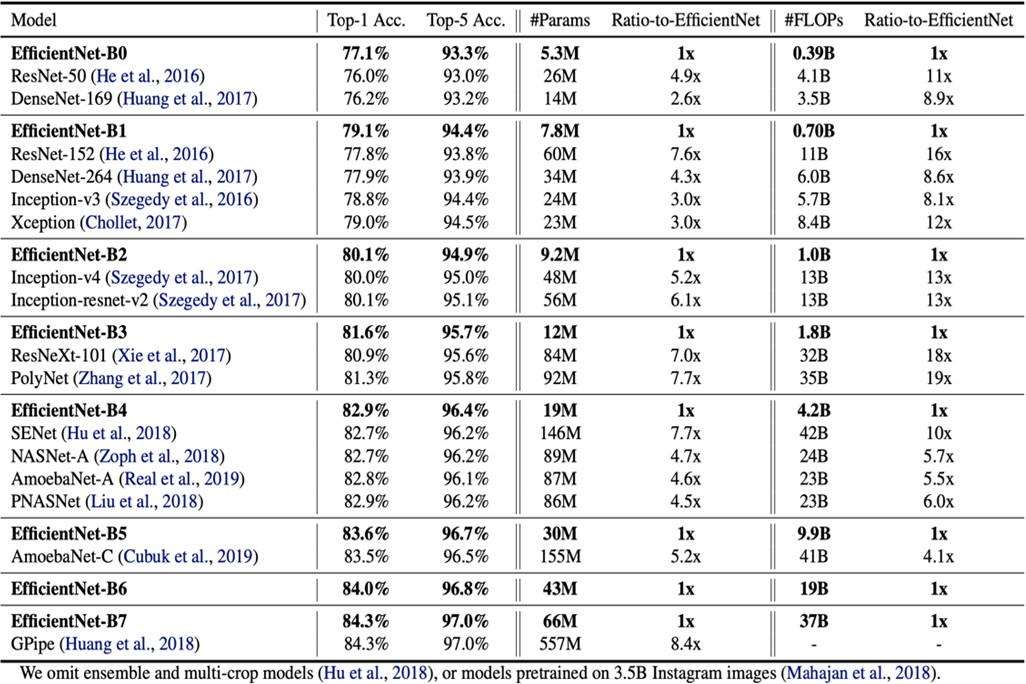


Figure 2: Quá trình thực nghiệm

## Mobile Net:

* + 1. Giới thiệu
* MobileNet là một họ kiến trúc mạng nơ-ron tích chập được thiết kế bởi Google, tối ưu hóa cho việc sử dụng trên các thiết bị di động và nhúng với tài nguyên hạn chế. Các mô hình MobileNet tập trung vào việc giảm thiểu số lượng tham số và phép tính, đồng thời vẫn duy trì độ chính xác cao trên các tác vụ thị giác máy tính.
  + 1. Các phiên bản chính
* **MobileNetV1 (2017):** Phiên bản đầu tiên giới thiệu khái niệm tích chập chiều sâu (depthwise separable convolution) để giảm đáng kể số lượng phép tính và kích thước mô hình.
* **MobileNetV2 (2018):** Cải thiện hiệu suất bằng cách sử dụng khối dư nghịch đảo (inverted residual blocks) và kết nối tắt tuyến tính (linear bottlenecks).
* **MobileNetV3 (2019):** Tối ưu hóa thêm bằng cách sử dụng tìm kiếm kiến trúc (architecture search) và các kỹ thuật lượng tử hóa (quantization).
  + 1. Ưu điểm
* **Kích thước nhỏ gọn:** MobileNet có kích thước nhỏ hơn đáng kể so với các mô hình truyền thống như VGG hay ResNet, giúp tiết kiệm không gian lưu trữ và băng thông.
* **Hiệu quả tính toán:** Sử dụng tích chập chiều sâu và các kỹ thuật tối ưu hóa khác để giảm số lượng phép tính, cho phép triển khai trên các thiết bị có khả năng xử lý hạn chế.
* **Độ chính xác cao:** Mặc dù có kích thước nhỏ và hiệu quả cao, MobileNet vẫn đạt được độ chính xác cạnh tranh trên nhiều tác vụ thị giác máy tính.
* **Khả năng tùy chỉnh:** MobileNet cho phép điều chỉnh độ sâu và chiều rộng của mạng để cân bằng giữa hiệu suất và độ chính xác, phù hợp với các ứng dụng khác nhau.

# Dataset

1. FLAVIA: Số lượng ảnh: 1.907 ảnh lá cây. Số lượng loại cây: 32 loài cây khác nhau. Kích thước hình ảnh: 1600x1200 pixels. Định dạng ảnh: JPEG.

Hình ảnh được chụp với nền trắng, tạo điều kiện thuận lợi cho việc xử lý và nhận diện.

1. CHIA DỮ LIỆU TRAINING: Chia dữ liệu thành 3 phần:
   * Tập huấn luyện (Training Set): 70% dữ liệu – Dùng để huấn luyện mô hình.
   * Tập xác nhận (Validation Set): 15% dữ liệu – Dùng để tối ưu hóa và điều chỉnh tham số của mô hình.
   * Tập kiểm tra (Test Set): 15% dữ liệu – Dùng để đánh giá hiệu năng mô hình sau khi huấn luyện.

# Phương pháp thực hiện

* + 1. Xây dựng mô hình
       - Mô tả quá trình huấn luyện mô hình: Quá trình huấn luyện mô hình được thực hiện trên Google Colab, nơi cung cấp tài nguyên tính toán mạnh mẽ, bao gồm GPU, giúp tăng tốc quá trình huấn luyện và tối ưu hóa mô hình. Chúng tôi đã sử dụng mô hình EfficientNet – một mô hình mạng nơ-ron tích chập (Convolutional Neural Network) có kiến trúc hiệu quả, phù hợp với các ứng dụng yêu cầu tối ưu hóa hiệu suất.
       - Công cụ và thư viện: Quá trình huấn luyện được thực hiện bằng cách sử dụng thư viện TensorFlow/Keras, một thư viện phổ biến và mạnh mẽ trong lĩnh vực học sâu. TensorFlow/Keras hỗ trợ các công cụ dễ sử dụng và cung cấp các hàm chuyên biệt để xây dựng, huấn luyện và đánh giá mô hình.
       - Lưu mô hình đã huấn luyện: Sau khi hoàn tất quá trình huấn luyện, chúng tôi lưu mô hình dưới định dạng .h5. Đây là định dạng phổ biến, cho phép lưu trữ toàn bộ trọng số và cấu trúc của mô hình, giúp dễ dàng tái sử dụng hoặc chuyển đổi sang các định dạng khác.
    2. Chuyển đổi mô hình sang TensorFlow Lite:
       - Mục đích chuyển đổi: Mô hình ban đầu có thể có kích thước lớn và cần nhiều tài nguyên để vận hành. Do đó, việc chuyển đổi mô hình sang định dạng TensorFlow Lite giúp giảm thiểu kích thước và tối ưu hóa tốc độ thực thi, phù hợp cho các ứng dụng di động.
       - Công cụ chuyển đổi: Sử dụng TensorFlow Lite Converter, chúng tôi tiến hành chuyển đổi mô hình EfficientNet từ định dạng .h5 sang định dạng TensorFlow Lite (.tflite). Công cụ này hỗ trợ nhiều kỹ thuật tối ưu hóa như lượng tử hóa (quantization) để giảm kích thước mô hình và tăng tốc độ suy luận mà không làm giảm đáng kể độ chính xác của mô hình.
    3. Tích hợp mô hình TensorFlow Lite vào ứng dụng di động (Android):
       - Tích hợp mô hình: Sau khi chuyển đổi thành công sang định dạng .tflite, mô hình được tích hợp vào ứng dụng di động trên nền tảng Android.
       - Thư viện TensorFlow Lite: Để thực thi mô hình, ứng dụng sử dụng thư viện TensorFlow Lite, một thư viện chuyên biệt của TensorFlow dành cho các thiết bị di động và nhúng. Thư viện này cho phép chạy các mô hình học sâu trực tiếp trên thiết bị di động, mang đến khả năng xử lý nhanh chóng và tiết kiệm tài nguyên.
       - Kết quả: Việc tích hợp mô hình vào ứng dụng giúp cải thiện khả năng xử lý của ứng dụng di động mà không yêu cầu tài nguyên tính toán cao, đồng thời đảm bảo độ chính xác và hiệu suất của mô hình đã huấn luyện.

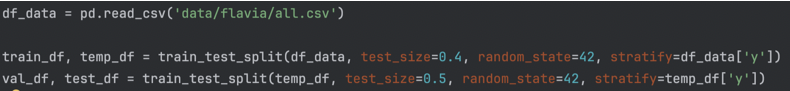
# Cài đặt

* Cài đặt các thư viện cần thiết

A black background with white text

Description automatically generated

* Import dữ liệu và nhãn



A screenshot of a computer program

Description automatically generated

* Tạo data generator và tăng cường dữ liệu train

A screenshot of a computer program

Description automatically generated 

* Xây dựng mô hình:

A computer screen with text

Description automatically generated

* Huấn luyện mô hình

A screen shot of a computer code

Description automatically generated

* Đánh giá mô hình trên tập test:

* Chuyển mô hình về tflite:

A screen shot of a computer

Description automatically generated

# Thách thức

Mặc dù nhận diện lá cây bằng các thuật toán học sâu (deep learning), đặc biệt là EfficientNet, đã đạt được nhiều kết quả ấn tượng, nhưng vẫn còn tồn tại nhiều thách thức đáng kể. Những thách thức này không chỉ liên quan đến việc tối ưu hóa mô hình mà còn đến từ các yếu tố thực tế trong việc triển khai hệ thống. Dưới đây là những thách thức chính:

## Khó khăn trong quá trình training theo EfficientNet-B0

* Trong lúc xây dựng và huấn luyện mô hình bằng EfficientNet-B0 nhóm đã gặp khá nhiều vấn đề về dữ liệu, vì flavia là dataset nhỏ, mỗi cây chỉ có khoảng 60 ảnh dẫn đến việc bị overfitting. Nhóm đã cố gắng thử nhiều cách nhưng không đem lại kết quả tốt.
* Chuyển đổi mô hình sử dụng MobileNet: Tuy dữ liệu training trên EfficientNet-B0 không đem lại kết quả tốt nhưng với MobileNet lại có kết quả khá ấn tượng với tỉ lệ chính xác trên tập test là 96%, nên nhóm quyết định làm demo trên mô hình được xây bởi MobileNet.

## Đa dạng về đặc điểm của lá cây

* Lá cây có hình dạng, kích thước rất đa dạng giữa các loài. Một số loài cây có lá tương tự nhau, dễ gây nhầm lẫn trong quá trình nhận diện.
* Lá cây có thể thay đổi hình dáng do tác động của môi trường như gió, nước hoặc ánh sáng.
* Màu sắc lá thay đổi theo mùa, từ xanh tươi vào mùa xuân và hè đến vàng úa hoặc đỏ vào mùa thu, làm khó khăn cho hệ thống nhận diện.
* Các hoa văn trên lá (như vân lá) đôi khi rất nhỏ hoặc không rõ ràng, dẫn đến việc nhận diện sai.

## Hiệu suất trên thiết bị di động

* Thiết bị di động có giới hạn về tài nguyên phần cứng như bộ xử lý, bộ nhớ và thời lượng pin. Điều này làm giảm khả năng chạy các mô hình lớn và phức tạp như EfficientNet.
* Việc đảm bảo mô hình nhận diện hoạt động nhanh chóng trên thiết bị di động là một thách thức, đặc biệt khi phải xử lý các hình ảnh có độ phân giải cao hoặc trong thời gian thực.

# Ứng dụng điện thoại

# Giao diện gồm 3 phần chính:

# Hiển thị hình ảnh được chụp hoặc được chọn từ gallery

# Khu vực hiển thị thông tin nhãn của ảnh

# Khu vực chứa nút chụp hoặc chọn ảnh

# A black rectangular frame with a white screen Description automatically generated

# Các đoạn code chính trong ứng dụng mobile:

# Load mô hình:

# A screen shot of a computer program Description automatically generated

# Tiền xử lý ảnh:

# A computer screen shot of a program Description automatically generated

# Nhận diện ảnh:

# A computer screen shot of a program Description automatically generated

# Kết quả

## Triển khai thành công trên thiết bị di động:

Ứng dụng nhận diện lá cây sẽ được triển khai trên các thiết bị di động với hiệu suất tối ưu. Điều này đòi hỏi mô hình phải được tối giản và chuyển đổi sang định dạng TensorFlow Lite hoặc các định dạng tương tự để phù hợp với các thiết bị di động có tài nguyên hạn chế về bộ nhớ và xử lý.

## Độ chính xác cao (~96%):

Mô hình đã hoạt động tốt và có độ chính xác cao trên tập test cũng như những bức ảnh thực tế

## Tính Nhân rộng

Hệ thống có thể được mở rộng để phục vụ nhiều lĩnh vực khác nhau, như:

* Hỗ trợ người nông dân trong việc nhận diện các loại cây trồng, theo dõi tình trạng sức khỏe của cây qua lá, và phát hiện sớm bệnh dịch.
* Giúp nhận diện và phân loại các loài cây trong khu vực rừng, hỗ trợ giám sát đa dạng sinh học và bảo vệ rừng.
* Nhận diện các loại cây thuốc dựa trên lá, hỗ trợ các nhà dược học hoặc người dân địa phương tìm kiếm và sử dụng dược liệu tự nhiên.