TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO CUỐI KỲ

HỌC PHẦN HỌC MÁY VÀ ỨNG DỤNG

ĐỀ TÀI:

HỆ THỐNG NHẬN DẠNG TRÁI CÂY

GIẢNG VIÊN HƯỚNG DẪN: TS. Phạm Công Thắng

HỌ VÀ TÊN SINH VIÊN	LỚP SINH HOẠT	LỚP HỌC PHẦN
Cao Kiều Văn Mạnh	20TCLC_KHDL	20.15
Võ Hoàng Bảo		
Nguyễn Tuấn Hưng		
Nguyễn Tiến Hùng		

ĐÀ NẴNG, 12/2023

BẢNG PHÂN CÔNG NHIỆM VỤ

Sinh viên thực hiện	Các nhiệm vụ	Đánh giá
Cao Kiều Văn Mạnh	Phân công nhiệm vụ, đảm bảo tiến độ	Hoàn thành
	Xây dựng mô hình, quy trình huấn luyện	
	Tích hợp mô hình vào hệ thống website	
	Viết báo cáo, chuẩn bị nội dung thuyết trình	
Võ Hoàng Bảo	Thu thập và xây dựng bộ dữ liệu	Hoàn thành
	Tích hợp phương pháp giải thích kết quả dự đoán của mô hình vào hệ thống	
	Viết báo cáo, chuẩn bị nội dung thuyết trình	
Nguyễn Tuấn Hưng	Xây dựng hệ thống website	Hoàn thành
	Tích hợp mô hình vào hệ thống website	
	Deploy hệ thống	
	Viết báo cáo, chuẩn bị nội dung thuyết trình	
Nguyễn Tiến Hùng	Thu thập và xây dựng bộ dữ liệu	Hoàn thành
	Thực hiện quy trình làm sạch và tiền xử lý dữ liệu	
	Viết báo cáo, chuẩn bị nội dung thuyết trình	

MỤC LỤC

DANH MỤC HÌNH ẢNH

DANH MỤC BẢNG

TÓM TẮT ĐỒ ÁN

Trong bối cảnh của sự bùng nổ mạnh mẽ của trí tuệ nhân tạo (AI) và sự tiến bộ đáng kể trong lĩnh vực công nghệ, việc áp dụng AI vào các lĩnh vực khác nhau của cuộc sống hàng ngày đã trở thành xu hướng không thể phủ nhận. Sự phát triển của các mô hình học máy và các học sâu trở thành một công cụ đắc lực giải quyết các bài toán trong lĩnh vực thị giác máy tính, đặc biệt là các bài toán nhận dạng và phân loại, một trong số đó là bài toán nhận dạng trái cây.

Đề tài "Hệ Thống Nhận Dạng Trái Cây" đã được đưa ra với hy vọng có thể ứng dụng thành công các mô hình học sâu hiện đại để xây dựng một hệ thống nhận dạng trái cây tự động, đặc biệt là đối với các loại trái cây phổ biến ở nước ta. Nhóm tiến hành xây dựng một bộ dữ liệu ảnh 20 loại trái cây khác nhau. Bên cạnh đó, nhóm đề xuất các mô hình học sâu dựa trên các kiến trúc Convolutional Neural Network như MobileNet, ResNet và DenseNet nhằm nhận diện các loại trái cây. Ngoài ra, nhóm cũng tiến hành xây dựng hệ thống website để tích hợp kết quả nghiên cứu, giúp người dùng có thể dễ dàng tương tác và sử dụng.

Qua quá trình thử nghiệm và đánh giá trên tập dữ liệu nhóm xây dựng, kết quả cho thấy sự hiệu quả của các mô hình với độ chính xác cao nhất lên đến **89.73%** cùng với hệ thống có tốc độ xử lý nhanh, với tốc độ xử lý nhanh nhất trung bình từ **3 - 5s/ảnh** cho quá trình phân loại và giải thích kết quả phân loại. Tuy nhiên, kết quả vẫn còn nhiều thiếu sót và hạn chế. Nhóm chúng em sẽ tiếp tục phát triển và hoàn thiện hơn trong tương lai.

1. TỔNG QUAN ĐỀ TÀI

1.1. Vấn đề:

Trong thời đại công nghệ hiện đại, ứng dụng của trí tuệ nhân tạo (AI) ngày càng nở rộ và mang lại nhiều tiện ích đối với cuộc sống hàng ngày. Một trong những ứng dụng tiêu biểu của AI là khả năng nhận diện hình ảnh, giúp con người nhanh chóng xác định thông tin từ hình ảnh một cách chính xác và hiệu quả.

Mặc dù có nhiều ứng dụng của nhận diện hình ảnh, nhưng việc nhận diện trái cây vẫn đặt ra nhiều thách thức đặc biệt. Sự đa dạng về hình dạng, màu sắc, và kích thước của các loại trái cây tạo ra khó khăn trong việc xây dựng mô hình nhận diện độ chính xác cao. Ngày càng nhiều người tiêu dùng quan tâm đến việc ăn uống lành mạnh và đa dạng. Do đó, nhu cầu tìm hiểu về thông tin dinh dưỡng của các loại trái cây qua việc chụp hình và nhận diện nhanh chóng đặt ra một thách thức lớn đối với ứng dụng nhận diện trái cây.

Từ thực tế trên, nhóm chúng em tiến hành nghiên cứu và phát triển đề tài "*Hệ thống nhận dạng trái cây (A system for recognizing fruit)*" nhằm giải quyết các vấn đề này.

1.2. Mục tiêu đề tài

Các mục tiêu chính của dự án lần này bao gồm: (1) thu thập và tổng hợp bộ dữ liệu trái cây có độ đa dạng cao; (2) thử nghiệm và xây dựng các mô hình, kỹ thuật học sâu để nhận dạng trái cây; (3) tiến hành tích hợp mô hình trên vào hệ thống website và cung cấp dịch vụ cho người dùng một cách chính xác, đơn giản và hiệu quả.

2. CƠ SỞ LÝ THUYẾT

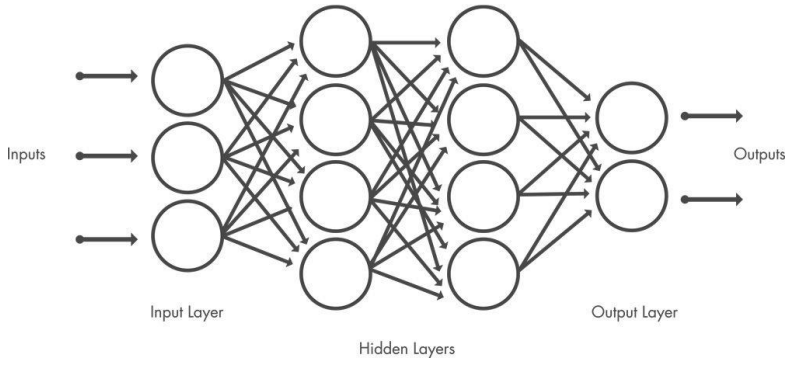
2.1. Ý tưởng

Bài toán nhận dạng và phân loại ảnh là một bài toán cơ bản nhất trong lĩnh vực thị giác máy tính. Là một trường hợp cụ thể của bài toán nhận dạng và phân loại, bài toán nhận dạng trái cây kế thừa các khó khăn của các bài toán gốc kèm theo các khó khăn của chính nó: số lượng hoa quả đa dạng theo mùa, vùng miền, địa hình… với vô số các loại hoa quả có hình dáng, màu sắc, kết cấu tương tự nhau.

Với sự phát triển của các mô hình học sâu, trong đó là CNN đang thể hiện hiệu suất tích cực của mình trong việc giải quyết các bài toán trong lĩnh vực thị giác máy tính, đặc biệt là các bài toán phân loại và nhận dạng. Chỉ cần có đủ dữ liệu, điều chỉnh hợp lý các tham số, huấn luyện hiệu quả, bài toán nhận dạng trái cây có thể dễ dàng được giải quyết bằng các mạng học sâu CNN. Từ đó tích hợp mô hình vào hệ thống dịch vụ để cung cấp cho người dùng.

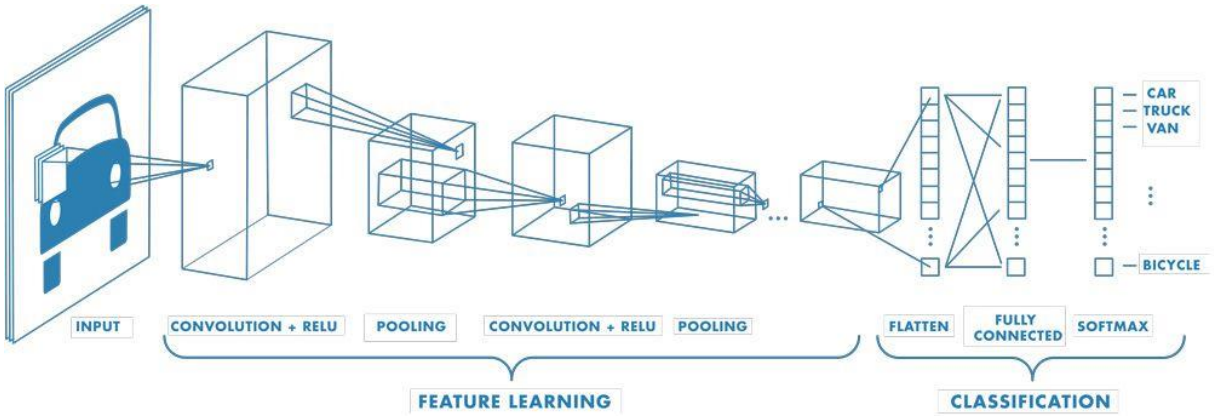
2.2. Convolutional Neural Network (CNN)

CNN hay mạng neural tích chập là một trong những mô hình hiện đại và phổ biến được áp dụng trong lĩnh vực thị giác máy tính, đặc biệt là các bài toán nhận dạng và phân loại. CNN coi hình ảnh đầu vào là một mảng pixel và nó phụ thuộc vào độ phân giải của hình ảnh. Dựa trên độ phân giải của hình ảnh, một đầu vào sẽ có 3 chiều đó chính là chiều cao H, chiều rộng W và độ sâu D. Một mạng CNN bao gồm một lớp input, một lớp output và các lớp ẩn ở giữa chúng.



Hình 1. Thứ tự các lớp trong CNN

Hình dưới đây là ví dụ về toàn bộ luồng CNN để xử lý hình ảnh đầu vào và phân loại các đối tượng dựa trên giá trị.



Hình 2. Ví dụ kiến trúc một mô hình CNN phân loại xe

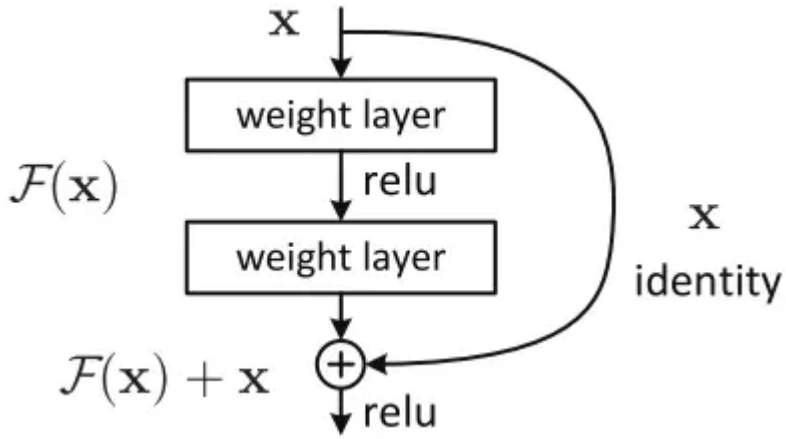
- **Convolution** hay tích chập đưa hình ảnh đầu vào qua một tập các bộ lọc tích chập (kernels), mỗi bộ lọc kích hoạt những đặc trưng nhất định từ hình ảnh. Việc áp dụng hình ảnh đầu vào với các bộ lọc khác nhau có thể thực hiện các hoạt động như phát hiện cạnh, làm mờ hoặc làm rõ nét.
- **Activation** ReLU (Rectified linear unit) là một trong những hàm kích hoạt phổ biến trong CNN cho phép việc huấn luyện nhanh và hiệu quả hơn bằng việc ánh xạ các giá trị âm về 0 và giữ các giá trị dương. Nó được gọi là lớp kích hoạt do việc chỉ có các đặc trưng nó kích hoạt được giữ lại và đưa vào lớp tiếp theo. Một số thay thế có thể được nhắc đến như tanh hay sigmoid tùy thuộc vào mô hình.
- **Pooling** sẽ giảm bớt số lượng tham số khi hình ảnh quá lớn. Không gian pooling còn được gọi là lấy mẫu con hoặc lấy mẫu xuống làm giảm kích thước của mỗi map nhưng vẫn giữ lại thông tin quan trọng. Các pooling có thể có nhiều loại khác nhau: Max Pooling, Average Pooling, Sum Pooling

Về kỹ thuật, trong mô hình CNN, mỗi hình ảnh đầu vào sẽ được chuyển qua 1 loạt các lớp tích chập với các bộ lọc (Kernels), tổng hợp lại các lớp được kết nối đầy đủ (Full Connected) và áp dụng hàm Softmax để phân loại đối tượng có giá trị xác suất giữa 0 và 1.

2.3. Kiến trúc ResNet

Một vấn đề phổ biến thường thấy ở các mô hình CNN có nhiều lớp tích chập đó là hiện tượng Vanishing Gradient, khi mà Gradients thường sẽ có giá trị nhỏ dần khi đi xuống các layer thấp hơn. Dẫn đến kết quả là các cập nhật thực hiện bởi Gradients Descent không làm thay đổi nhiều về trọng số của các layer đó và làm chúng không thể hội tụ và mạng sẽ không thu được kết quả tốt. Mạng ***ResNet*** ra đời để giải quyết vấn đề đó.

Giải pháp mà mạng ResNet đưa ra là sử dụng một kết nối "tắt" đồng nhất để xuyên qua một lớp hay nhiều lớp. Một khối như vậy được gọi là một Residual Block [4], được mô tả như hình ảnh sau.



Hình 3. Khối Residual Block [4]

ResNet gần như tương tự với các mạng CNN cơ bản với các lớp convolution, pooling, activation và fully-connected layer. Ảnh bên trên thể hiện cách thức Residual block được sử dụng trong mạng. Xuất hiện một mũi tên cong xuất phát từ đầu và kết thúc tại cuối khối dư. Hay nói cách khác là sẽ bổ sung Input X vào đầu ra của layer, hay chính là phép cộng mà ta thấy trong hình minh họa, việc này sẽ giúp giảm thiểu hiện tượng đạo hàm bằng 0, do vẫn còn cộng thêm X. Với H(x) là giá trị dự đoán, F(x) là giá trị thật (nhãn), chúng ta muốn H(x) bằng hoặc xấp xỉ F(x). Việc F(x) có được từ x như sau:

$$x \rightarrow weight_1 \rightarrow ReLU \rightarrow weight_2 \rightarrow F(x)$$

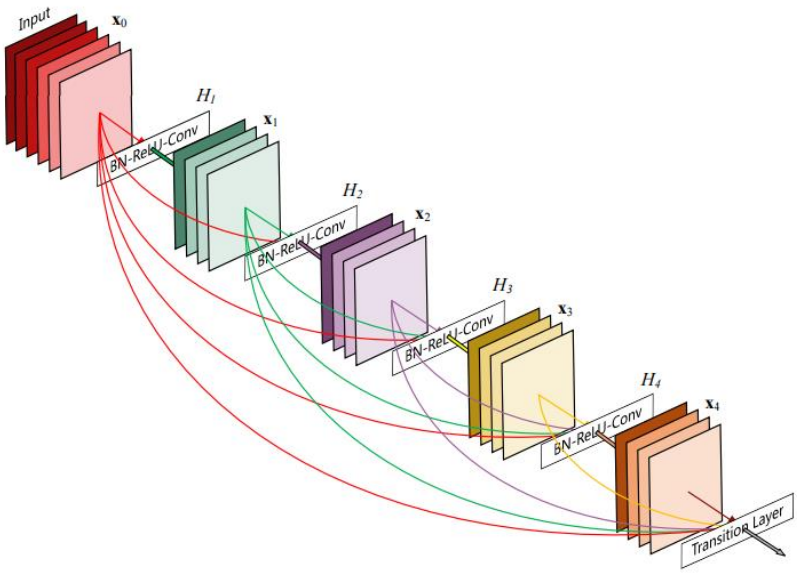
Giá trị H(x) có được bằng cách:

$$F(x) + x \rightarrow ReLU \rightarrow H(x)$$

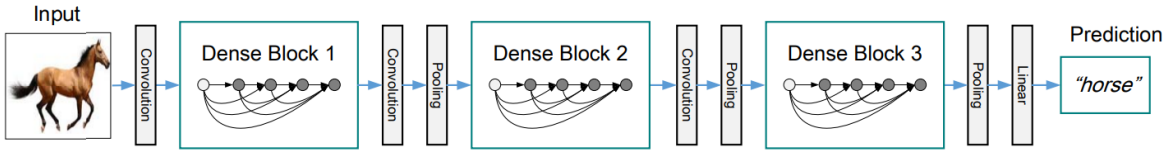
Việc xếp chồng các lớp sẽ không làm giảm hiệu suất mạng. Ta có thể đơn giản xếp chồng các ánh xạ đồng nhất lên mạng hiện tại và hiệu quả của kiến trúc không thay đổi. Với kiến trúc này, các lớp ở phía trên có được thông tin trực tiếp hơn từ các lớp dưới nên sẽ điều chỉnh trọng số hiệu quả hơn. Giải pháp ResNet là một giải pháp đơn giản tập trung vào cải tiến thông tin phản hồi thông qua độ dốc của mạng và tạo tiền đề cho hàng loạt biến thể của các kiến trúc sau này có thể được huấn luyện mạng nơ ron với độ sâu hàng nghìn lớp.

2.4. Kiến trúc DenseNet

Các nghiên cứu gần đây chứng minh rằng các mạng nơ ron tích chập sẽ có thể sâu hơn đáng kể, chính xác hơn và đạt hiệu suất huấn luyện cao hơn khi nếu nó chứa các kết nối ngắn hơn giữa các lớp gần input và output. Dựa vào ý tưởng đó, ***DenseNet (Densely connected convolutional network)*** ra đời, về ý tưởng, DenseNet hoạt động gần giống với mạng ResNet nhưng có một vài khác biệt cơ bản. Trong DenseNet, mỗi lớp sẽ kết nối với mọi lớp còn lại một cách feed-forward [5]. Do đó, với mạng CNN truyền thống nếu chúng ta có L layer thì sẽ có L connection, còn DenseNet sẽ có $\frac{L*(L+1)}{2}$ connection trực tiếp. Ở mỗi lớp, các ma trận đặc trưng của các lớp trước đó sẽ được sử dụng như là input, và ma trận đặc trưng của chính nó sẽ được làm input cho các lớp phía sau. Densenet có cấu trúc gồm các Dense block và các Transition layers được xếp chồng lên nhau: Dense block - Transition layers - Dense block - Transition layers -...



Hình 4. Kiến trúc của 1 Dense Block [5]



Hình 5. Một mạng DenseNet với 3 Dense Block [5]

Một số ưu điểm của Densenet:

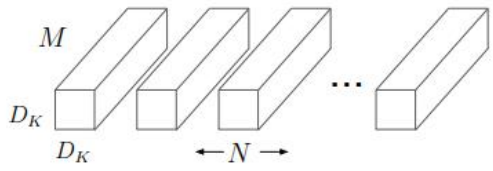
- Accuracy: Densenet huấn luyện tham số ít hơn 1 nửa so với mạng Resnet nhưng có cùng accuracy so trên ImageNet classification dataset.
- Overfitting: DenseNet chống overfitting rất hiệu quả.
- Giảm được vanishing gradient.
- Các feature học được từ các layer trước được sử dụng hiệu quả hơn.

2.5. Kiến trúc MobileNet

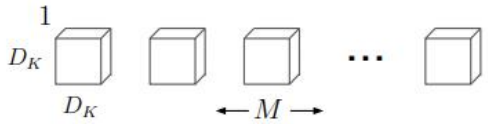
MobileNet là mô hình CNN được thiết kế với mục đích trở nên gọn nhẹ để ứng dụng vào các thiết bị di động và thiết bị nhúng. Với một mạng CNN truyền thống có các kích thước M, N là số input, output channel; Df là số chiều của ma trận đặc trưng đầu vào; Dk là số chiều của kernel, lúc đó số phép tính cần tính là khá lớn:

$$M * N * {D_f}^2 * {D_k}^2$$

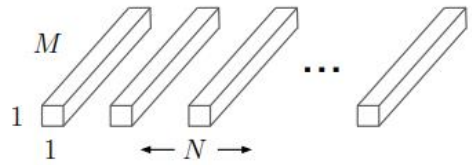
Thay vì tính tất cả như vậy, ý tưởng cốt lõi đầu tiên của MobileNet chính là ***Depthwise separable convolution*** [6]. Depthwise Separable Convolutions chia mạng CNN cơ bản ra làm hai phần: Deepwise Convolution và Pointwise Convolution.



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Hình 6. Sự khác biệt giữa các lớp convolution truyền thống với Depthwise và Pointwise Convolution

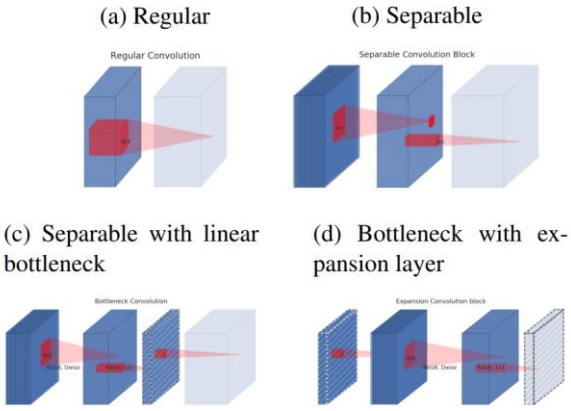
- **Deepwise Convolution**: thay vì tích chập 2 ma trận có cùng số kênh như CNN thông thường, ta tích chập đầu vào chỉ với số filter bằng với số channel nhưng

chỉ có chiều sâu là 1, thực hiện phép tích chập rời rạc trên từng channel. Với số phép tính là $M * {D_f}^2 * {D_k}^2$

- **Pointwise Convolution**: ta tiếp tục sử dụng kết quả từ Deepwise Convolution, ở bước Pointwise, ta sử dụng bộ lọc có kích thước 1x1, số lượng bộ lọc bằng số lượng channel mà ta muốn thu được. Kích thước không đổi, số channel thay đổi, số phép tính cần tính là $M * N * {D_k}^2$.

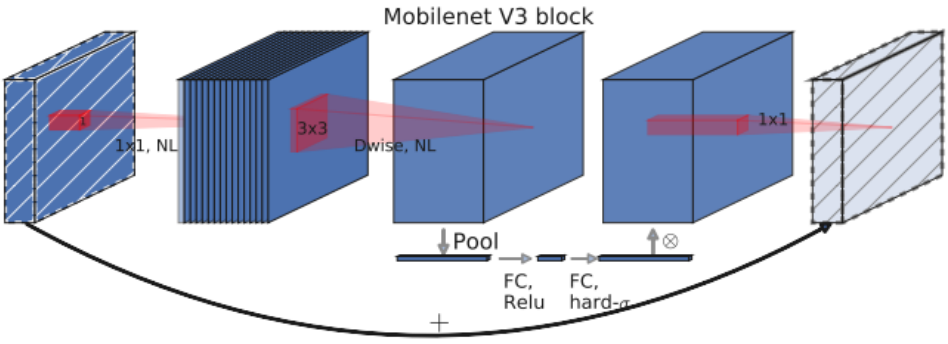
Tổng kết lại, Depthwise Separable Convolution có số lượng tham số bằng $\frac{1}{N} + \frac{1}{{D_k}^2}$ lần so với các mô hình CNN truyền thống. Nhờ có kỹ thuật này, MobileNet đã giảm số lượng tính toán, giảm số lượng các tham số, đồng thời có thể thực hiện trích xuất đặc trưng một cách tách biệt trên các channel khác nhau. Từ đó MobileNet còn phát triển thêm MobileNetV2 rồi tới MobileNetV3.

MobileNetV2 đề xuất thêm Linear Bottlenecks giúp giảm kích thước input và Inverted Residual Block giúp tăng độ chính xác của mô hình mà không cần đến chi phí lớn [7].



Hình 7. Sự phát triển của Separable Convolution[7]

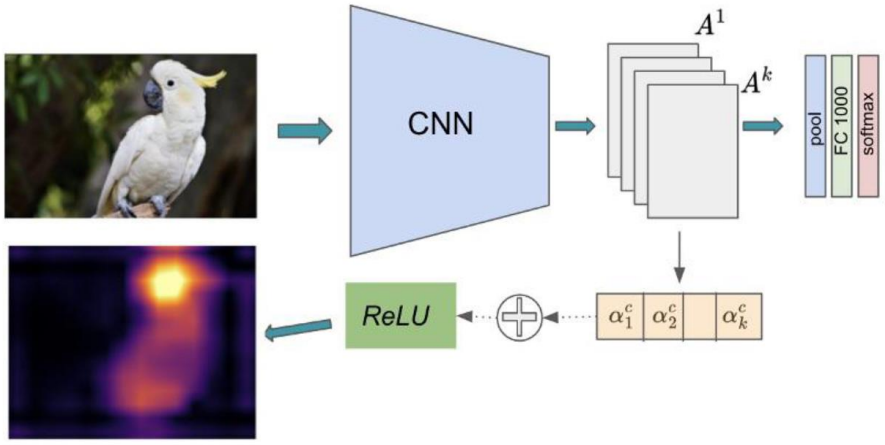
MobileNetV3 tiếp tục cải tiến bằng việc dùng Squeeze and Excite nhằm tăng lượng thông tin giữa các kênh [8].



Hình 8. Kiến trúc MobileNet V3[8]

2.6. Kỹ thuật GRAD-CAM

Grad-CAM là một trong những kỹ thuật giải thích đầu tiên được phát triển cho các mô hình xử lý ảnh và có thể được áp dụng lên các bất kỳ mạng CNN nào. Grad-CAM giúp tổng quát hóa kỹ thuật CAM khi CAM chỉ có thể dùng được cho một số kiến trúc nhất định [9]. Grad-CAM hoạt động dựa trên thông tin đạo hàm được cập nhật đi qua lớp tích chập cuối (hoặc bất kỳ) của mạng. Kết quả cho ra một heat map đánh dấu vùng của hình ảnh có ảnh hưởng cao nhất tới dự đoán của mô hình về lớp đã cho trước.



Hình 9. Sơ đồ hoạt động Grad-CAM [9]

Grad-CAM là một kĩ thuật giải thích hậu xử lý và không yêu cầu bất kỳ thay đổi nào về kiến trúc hay huấn luyện [7]. Thay vào đó, Grad-CAM truy cập vào lớp tích chập bên trong của mô hình để xác định khu vực có ảnh hưởng cao nhất đến sự dự đoán của mô hình. Bởi vì Grad-CAM chỉ dựa vào các bước chuyển tiếp qua mô hình, không có lan truyền ngược nên nó cũng có hiệu quả về mặt tính toán.

Hình 10. Mô tả kết quả đạt được của kỹ thuật Grad-CAM

3. GIẢI PHÁP TRIỂN KHAI

3.1. Giải pháp về dữ liệu

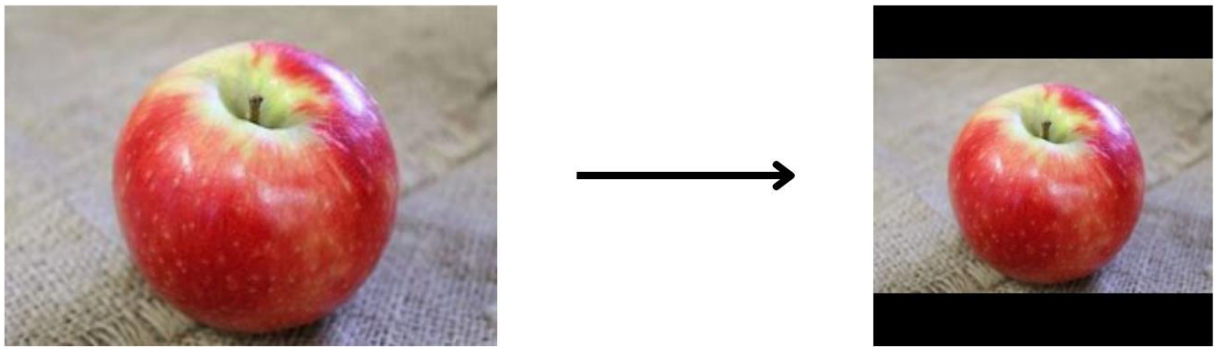
3.1.1. Thu thập dữ liệu hình ảnh trái cây

Nhóm tiến hành thu thập dữ hình ảnh trái cây từ hai tập "Fruits-262" [2] và "Fruits and Vegetables Image Recognition Dataset" [3] trên Kaggle. Từ hai bộ dữ liệu trên, nhóm tiến hành lấy hình ảnh của 20 loại trái cây sau cho đề tài: Apple (Táo), Avocado (Bơ), Banana (Chuối), Cucumber (Dưa leo), Dragonfruit (Thanh long), Durian (Sầu riêng), Grape (Nho), Guava (Ổi), Kiwi (Kiwi), Lemon (Chanh), Lychee (Vải), Mango (Xoài), Orange (Cam), Papaya (Đu đủ), Pear (Lê), Pineapple (Dứa), Pomegranate (Lựu), Strawberry (Dâu tây), Tomato (Cà chua) và Watermelon (Dưa hấu).

Sau đó nhóm tiến hành kiểm tra lại toàn bộ dữ liệu, lọc bỏ các hình ảnh trái cây không đúng loại, không đúng yêu cầu, không rõ ràng, mờ, chất lượng kém hoặc trộn lẫn với các loại trái cây khác.

3.1.2. Tiền xử lý dữ liệu

Sau khi kết thúc quá trình thu thập dữ liệu. Nhóm thực hiện một số bước tiền xử lý sau. Đầu tiên, nhóm tiến hành resize tất cả các ảnh về cùng kích thước 224x224 và định dạng chung để giảm độ phức tạp của mô hình. Trong quá trình resize, nhóm tiến hành thêm padding để giữ đúng tỉ lệ ảnh gốc, đảm bảo khi resize không bị ảnh hưởng về hình dạng của trái cây.



Hình 11. Mô tả cách thức resize ảnh

Sau đó, nhóm tiến hành chia bộ dữ liệu thành 3 tập train/test/validation với tỉ lệ 6/2/2 và đảm bảo phân bố ảnh của các lớp trái cây trong mỗi tập.

Cuối cùng, để tăng cường sự đa dạng của bộ dữ liệu, nhóm tiến hành làm giàu dữ liệu nhằm mô phỏng các thay đổi trên ảnh mà có thể xảy ra trong thực tế như xoay, lật, tăng giảm độ sáng, tương phản,v.v… (Chỉ thực hiện tăng cường dữ liệu trên tập huấn luyện).

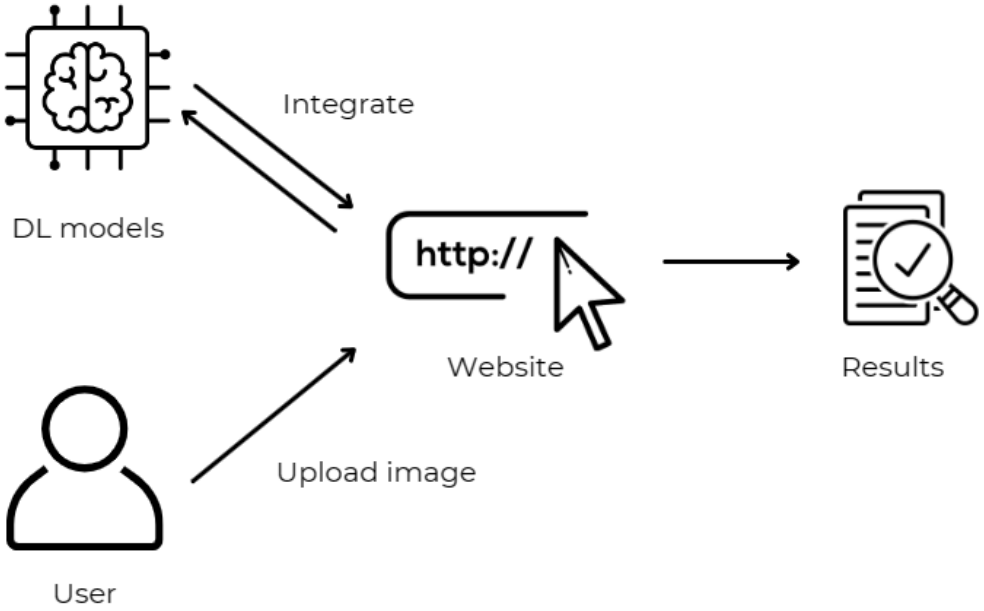
Hình 12. Sơ đồ quy trình tiền xử lý dữ liệu

3.2. Giải pháp về hệ thống

3.2.1. Xây dựng cấu trúc hệ thống và dịch vụ

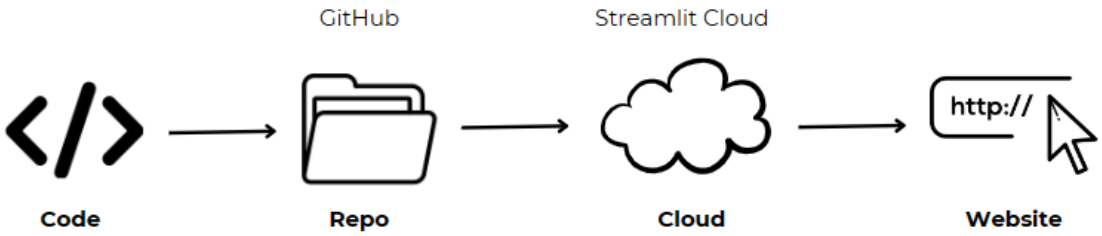
Nhằm cung cấp dịch vụ đến cho người dùng, nhóm tiến hành xây dựng hệ thống cung cấp dịch vụ bao gồm hai thành phần chính:

- Website: sử dụng framework Streamlit. Streamlit là một thư viện Python mã nguồn mở, cung cấp các hàm hỗ trợ giúp người dùng dễ dàng tạo ra một website cho phép tích hợp nhanh các mô hình Machine Learning/Deep Learning.
- Mô hình học sâu phân loại ảnh trái cây.



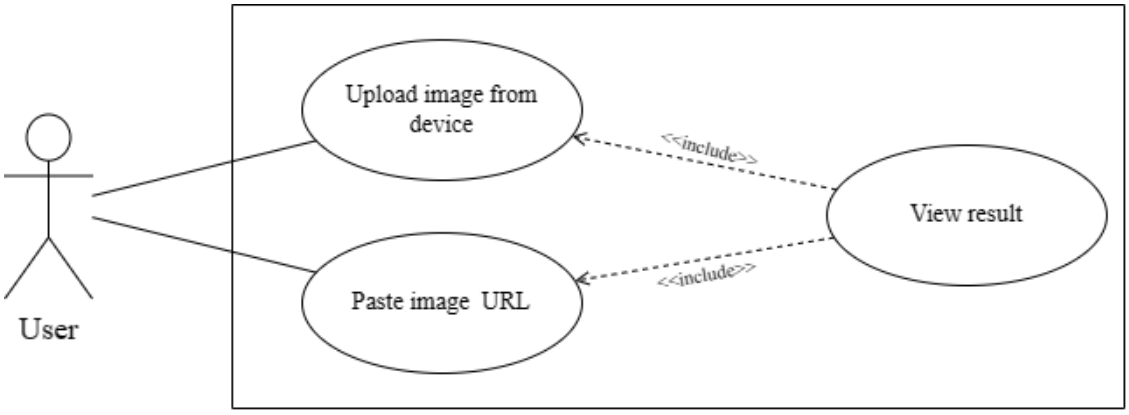
Hình 13. Sơ đồ tổng thể của hệ thống

Sau khi xây dựng xong hệ thống. Nhóm tiến hành deploy website thông qua dịch vụ Streamlit Community Cloud. Quy trình deploy hệ thống được mô tả như sơ đồ bên dưới.



Hình 14. Sơ đồ Deploy hệ thống

3.2.2. Sơ đồ usecase hệ thống



Hình 15. Sơ đồ usecase hệ thống

Chức năng	Mô tả
Upload image from device	Người dùng upload ảnh từ thiết bị lên hệ thống để nhận dạng.
Paste image URL	Người dùng có thể dán URL dẫn đến hình ảnh cần nhận dạng.
View result	Người dùng xem kết quả trả về từ mô hình bao gồm ảnh cần nhận dạng thuộc loại trái cây gì, độ tự tin của mô hình, hàm lượng calo có trong 100 gam của loại trái cây đó, hình ảnh giải thích kết quả nhận dạng của mô hình và tổng thời gian chạy của hệ thống dịch vụ.

Bảng 1. Phân tích chức năng hệ thống

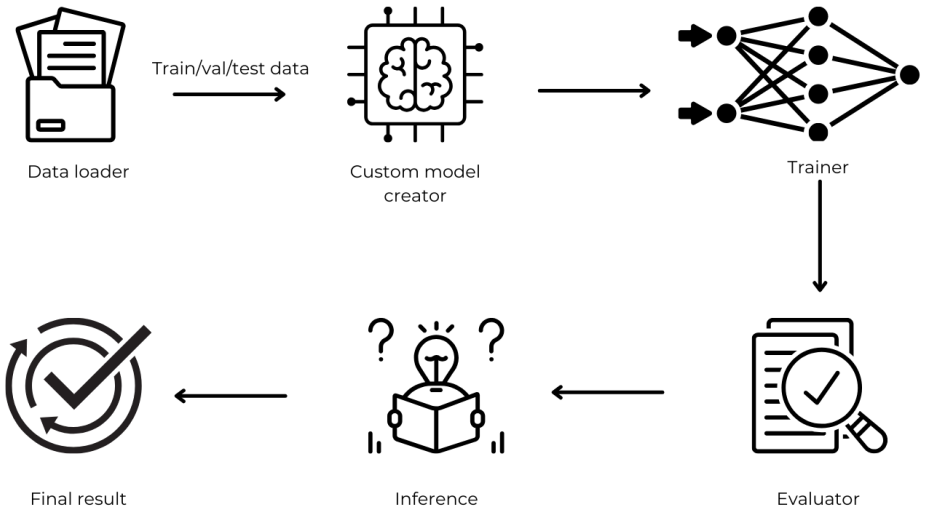
3.3. Giải pháp nhận dạng loại trái cây sử dụng deep learning

3.3.1. Xây dựng mô hình hệ thống

Trong đề tài này, nhóm tiến hành xây dựng lại 3 mô hình ResNet50, DenseNet121 và MobileNet. Cả 3 mô hình trên đều được xây dựng sử dụng framework Tensorflow. Thông tin cụ thể của ba mô hình như sau:

- ***Mô hình 1:*** sử dụng kiến trúc ResNet50.
- ***Mô hình 2:*** sử dụng kiến trúc DenseNet121.
- ***Mô hình 3:*** sử dụng kiến trúc MobileNet.

Bên cạnh việc xây dựng các kiến trúc mô hình ở trên, nhóm cũng đã tiến hành đã xây dựng các quy trình training/testing/inference hoàn chỉnh cho dự án này. Nhờ đó, trong tương lai, nhóm có thể dễ dàng thử nghiệm với kiến trúc mô hình mới, dữ liệu mới hoặc các kỹ thuật huấn luyện khác.



Hình 16. Sơ đồ quy trình xây dựng mô hình học sâu cho hệ thống

3.3.2. Huấn luyện mô hình

Nhóm tiến hành huấn luyện cả 3 mô hình trên cấu hình chung GPU NVIDIA V100 - 16GB vRAM; CPU Intel(R) Xeon(R) - 2.00GHz; RAM: 51 GB.

Các thông số huấn luyện của cả 3 mô hình được thiết lập như sau

- Kích thước ảnh đầu vào: 224x224 với batch size được sử dụng là 32.
- Tiến hành huấn luyện trong 60 epochs và dừng việc huấn luyện khi mô hình không được cải thiện trong vòng 15 epochs. Sử dụng các hàm hỗ trợ để lưu lại checkpoints tốt nhất của mô hình trong quá trình huấn luyện.
- Learning rate khởi tạo cho quá trình huấn luyện là 0.001 và tiến hành giảm learning rate đi một nửa sau mỗi 10 epochs.
- Optimizer được sử dụng là Adam optimizer. Hàm loss được sử dụng là Categorical Crossentropy.

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

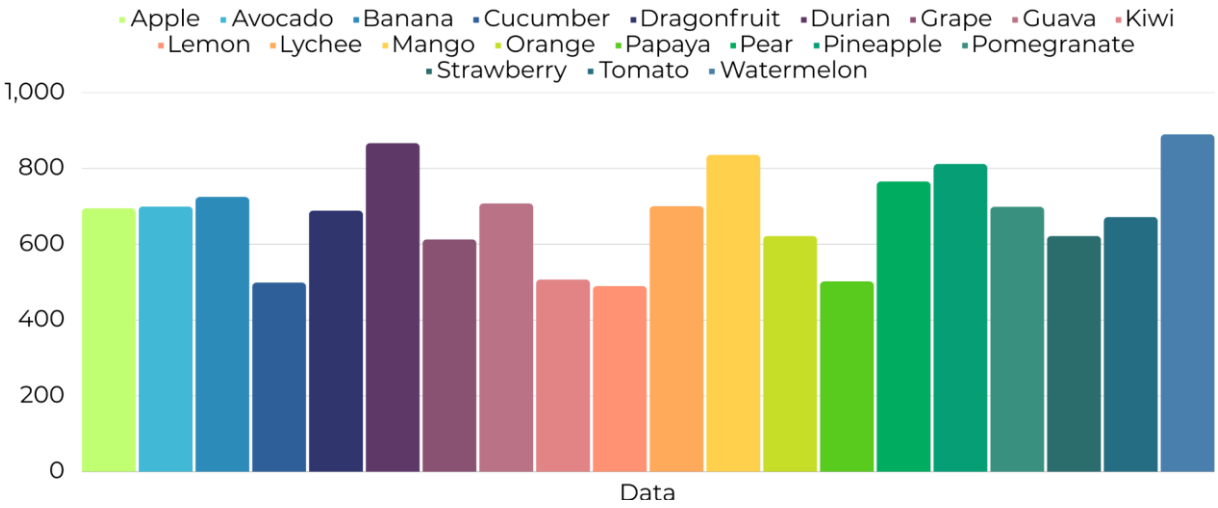
4.1. Kết quả thu thập và xử lý dữ liệu

Tổng kết, nhóm đã xây dựng được bộ dữ liệu bao gồm 13615 ảnh thuộc về 20 loại trái cây khác nhau. Thông tin cụ thể về dữ liệu được mô tả ở bảng sau.

STT	Tên tiếng Anh	Tên tiếng Việt	Tên lớp	Số lượng ảnh
1	Apple	Táo	0	695
2	Avocado	Bơ	1	700
3	Banana	Chuối	2	725
4	Cucumber	Dưa chuột	3	499

5	Dragonfruit	Thanh long	4	689
6	Durian	Sầu riêng	5	867
7	Grape	Nho	6	613
8	Guava	Ổi	7	708
9	Kiwi	Kiwi	8	507
10	Lemon	Chanh	9	490
11	Lychee	Vải thiều	10	701
12	Mango	Xoài	11	836
13	Orange	Cam	12	622
14	Papaya	Đu đủ	13	502
15	Pear	Lê	14	766
16	Pineapple	Dứa	15	812
17	Pomegranate	Lựu	16	699
18	Strawberry	Dâu	17	622
19	Tomato	Cà chua	18	672
20	Watermelon	Dưa hấu	19	890

Bảng 2. Số lượng dữ liệu đã thu thập



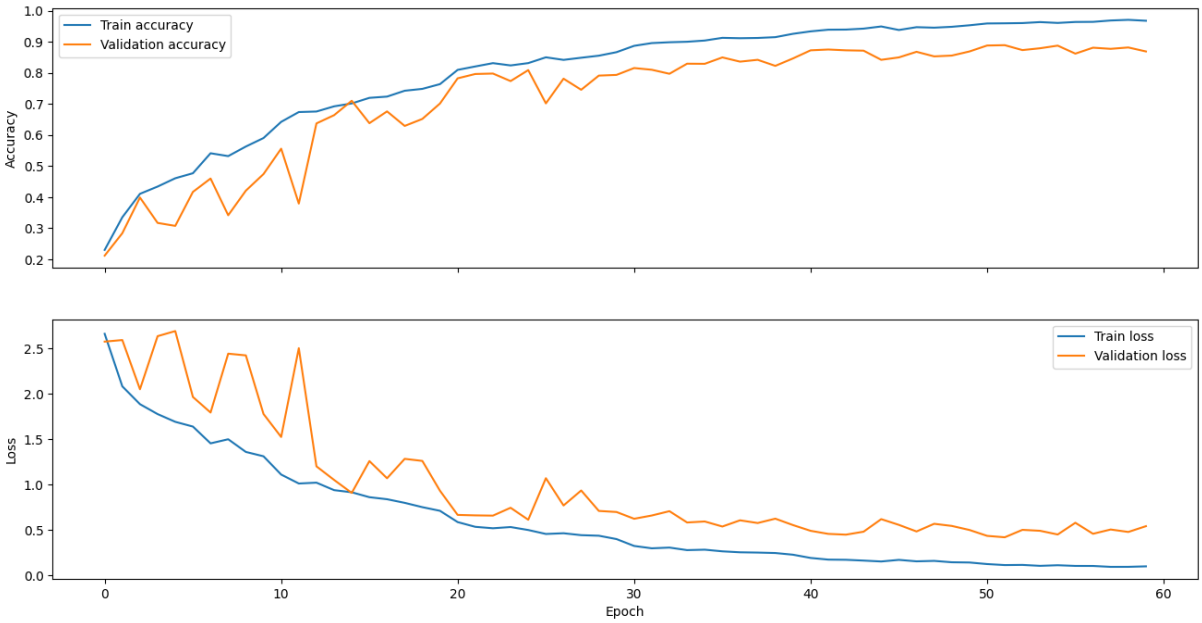
Hình 17. Biểu đồ số lượng các loại quả

Bộ dữ liệu trên được chia thành 3 tập train/test/validation theo tỷ lệ 6/2/2, ảnh của mỗi loại trái cây được chia đều cho các tập train/test/validation.

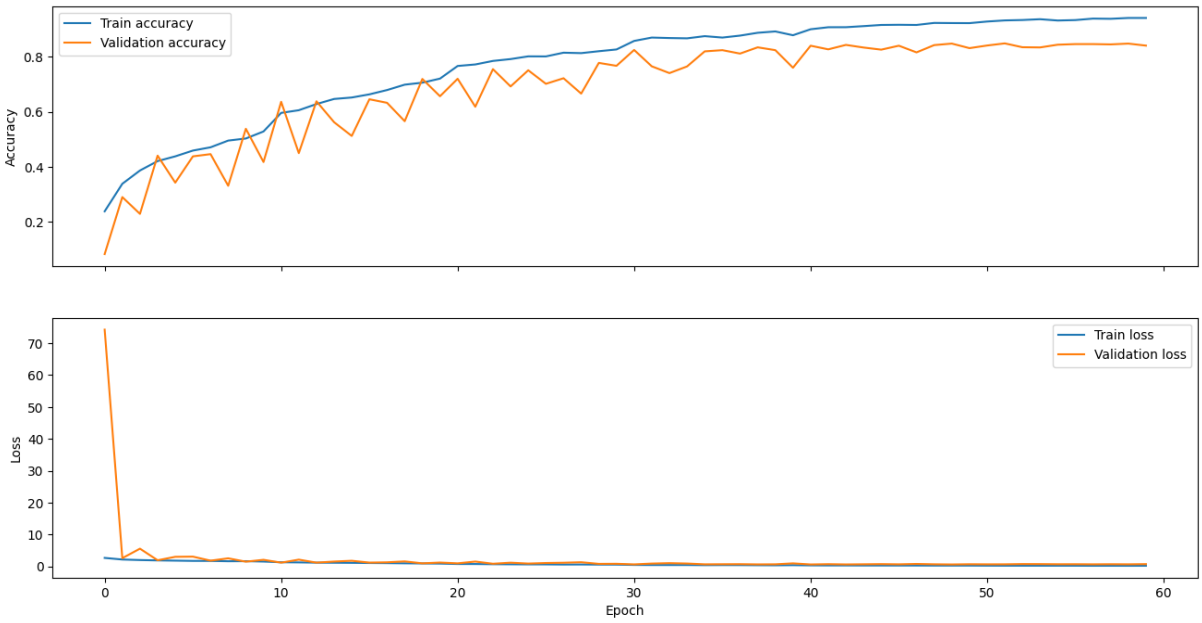
4.2. Kết quả huấn luyện và kiểm thử mô hình

4.2.1. Kết quả huấn luyện mô hình

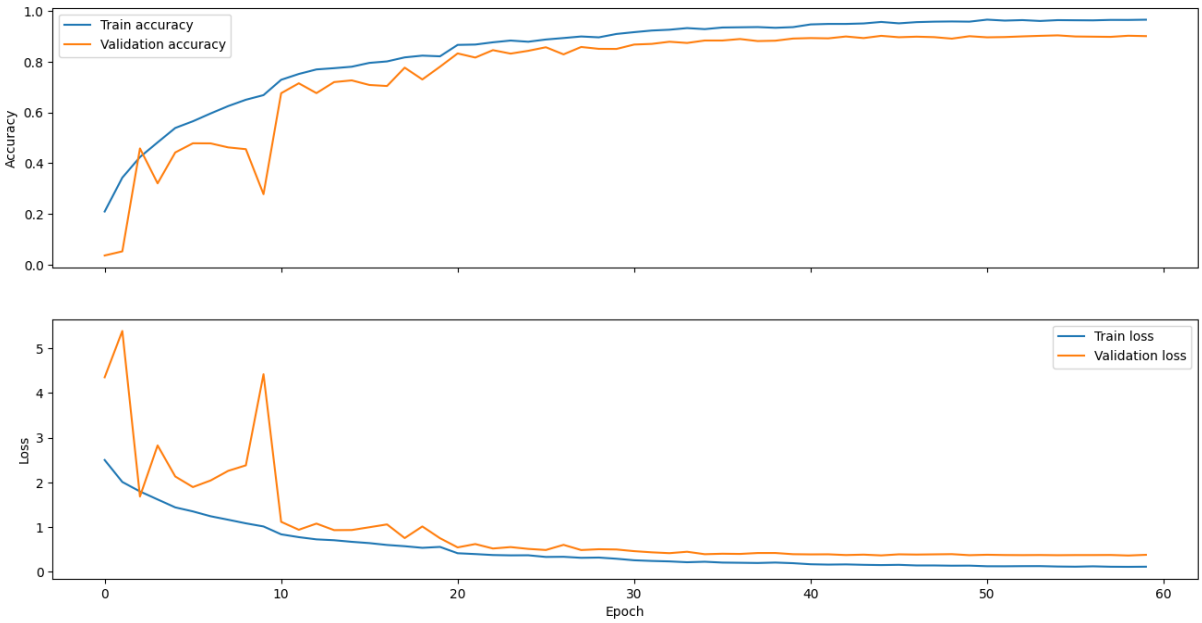
Sau đây là kết quả huấn luyện của 3 mô hình mà nhóm thử nghiệm.



Hình 18. Kết quả huấn luyện mô hình 1 (ResNet50)



Hình 19. Kết quả huấn luyện mô hình 2 (DenseNet121)



Hình 20. Kết quả huấn luyện mô hình 3 (MobileNet)

4.2.2. Kết quả kiểm thử mô hình

Nhóm tiến hành đánh giá cả 3 mô hình trên tập kiểm thử dựa trên các metrics sau: Accuracy, Precision, F1 và Recall.

Lớp	Mô hình 1 (ResNet50)			Mô hình 2 (DenseNet121)			Mô hình 3 (MobileNet)		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Apple	0.77	0.76	0.77	0.77	0.70	0.73	0.91	0.79	0.85
Avocado	0.87	0.90	0.88	0.81	0.80	0.80	0.84	0.91	0.87
Banana	0.92	0.92	0.92	0.86	0.91	0.88	0.92	0.92	0.92
Cucumber	0.84	0.87	0.86	0.81	0.77	0.79	0.82	0.89	0.85
Dragonfruit	0.93	0.96	0.95	0.87	0.97	0.92	0.94	0.99	0.96
Durian	0.95	0.99	0.97	0.91	0.99	0.95	0.97	0.98	0.97
Grape	0.94	0.96	0.95	0.95	0.95	0.95	0.95	0.96	0.96
Guava	0.84	0.85	0.85	0.78	0.71	0.74	0.86	0.88	0.87
Kiwi	0.99	0.93	0.96	0.94	0.93	0.94	0.98	0.93	0.95
Lemon	0.83	0.87	0.85	0.79	0.81	0.80	0.78	0.90	0.83
Lychee	0.95	0.89	0.92	0.91	0.86	0.89	0.88	0.91	0.90

Mango	0.83	0.73	0.78	0.79	0.71	0.75	0.87	0.71	0.78
Orange	0.91	0.94	0.92	0.95	0.85	0.90	0.91	0.94	0.92
Papaya	0.90	0.80	0.85	0.90	0.78	0.83	0.86	0.82	0.84
Pear	0.76	0.77	0.77	0.64	0.79	0.71	0.8	0.81	0.81
Pineapple	0.95	0.96	0.95	0.90	0.93	0.91	0.91	0.95	0.93
Pomegranate	0.89	0.86	0.88	0.83	0.83	0.83	0.91	0.84	0.88
Strawberry	0.88	0.97	0.92	0.91	0.93	0.92	0.93	0.97	0.95
Tomato	0.92	0.95	0.93	0.90	0.92	0.91	0.95	0.9	0.92
Watermelon	0.95	0.94	0.94	0.92	0.92	0.92	0.97	0.96	0.97
Accuracy	89.06%			85.42%			**89.84%**		

Bảng 3. Kết quả kiểm thử mô hình

Mô hình	**Kích thước mô hình (MB)**	**Inference time trên tập kiểm thử (s)**
Mô hình 1 (ResetNet50)	270.9	4
Mô hình 2 (DenseNet121)	82.25	6
Mô hình 3 (MobileNet)	**37.6**	**2**

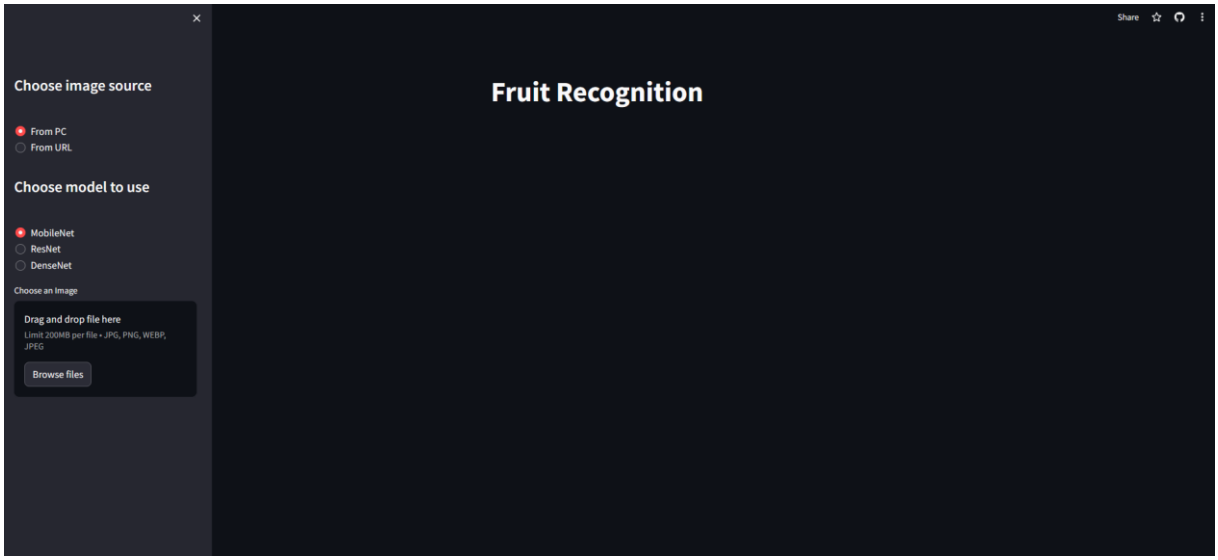
Bảng 4. Đánh giá kích thước và thời gian chạy trên tập kiểm thử

4.3. Kết quả xây dựng và kiểm thử hệ thống dịch vụ

4.3.1. Kết quả xây dựng hệ thống dịch vụ

❖ **Trang chủ**

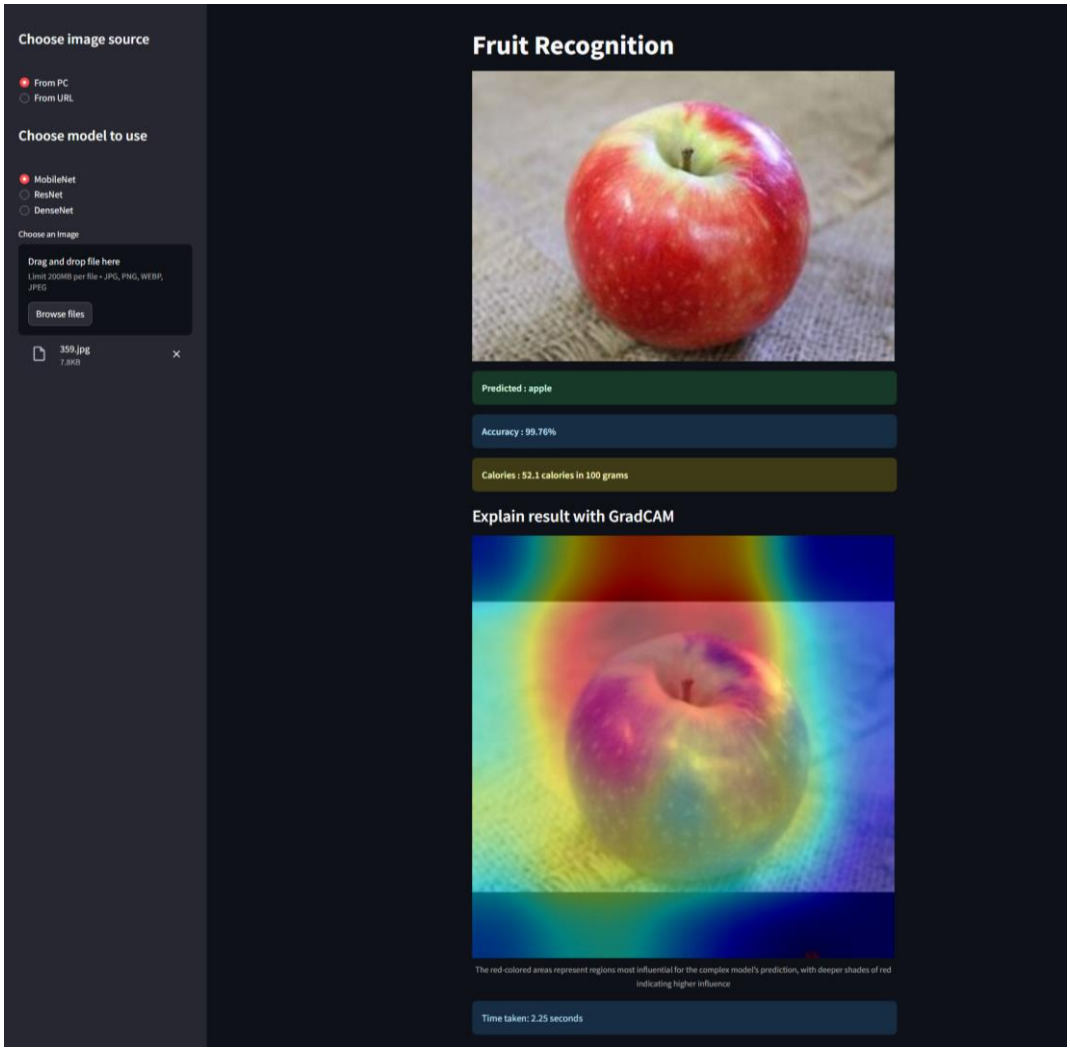
Trang chủ của giao diện hệ thống là nơi người dùng có thể tải ảnh từ thiết bị hoặc dán đường dẫn đến ảnh mà người dùng muốn nhận diện. Sau đó, người dùng có thể tiến hành lựa chọn mô hình muốn sử dụng và tiến hành nhận diện.



Hình 21. Giao diện trang chủ

❖ **Tính năng xem kết quả**

Sau khi đã upload ảnh, hệ thống truyền ảnh vào mô hình học sâu để đưa ra dự đoán và trả về cho website. Sau quá trình xử lý, kế quả hệ thống trả về cho người dùng bao gồm: lớp mà ảnh đầu vào được phân loại, độ tự tin của mô hình, số calo có trong 100 gam trái cây thuộc lớp được phân loại, kết quả giải thích đầu ra của kỹ thuật Grad-CAM và tổng thời gian xử lý của hệ thống.



Hình 22. Giao diện tính năng xem kết quả

4.3.2. Kết quả kiểm thử hệ thống dịch vụ

Cấu hình của Streamlit Cloud bao gồm các thông số sau:

- CPU: Intel Core i7.
- RAM: 1 GB.
- Ổ cứng: 1 GB.
- Hệ điều hành: Debian 11.

Nhóm tiến hành đánh giá thời gian chạy của 3 mô hình khi tích hợp vào hệ thống với 10 ảnh. Lưu ý, thời gian này bao gồm cả thời gian đưa ra kết quả giải thích bằng kỹ thuật Grad-CAM.

Mô hình	**Thời gian trung bình để đưa kết quả dự đoán (s)**
Mô hình 1 (ResetNet50)	4.71
Mô hình 2 (DenseNet121)	10.23
Mô hình 3 (MobileNet)	**2.46**

Bảng 5. Kết quả đánh giá thời gian chạy của hệ thống trong thực tế

5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết luận

Qua nghiên cứu này, nhóm đã xây dựng thành công được một bộ dữ liệu ảnh trái cây khá đa dạng cũng như đã tiến hành thử nghiệm và đề xuất việc sử dụng các mô hình học sâu CNN để xây dựng mô hình phân loại trái cây. Bên cạnh đó, nhóm cũng đã tiến hành đánh giá hiệu quả của các mô hình khi áp dụng vào hệ thống thực tế, từ đó xây dựng thành công hệ thống dịch vụ với độ chính xác cao, tốc độ xử lý nhanh và có khả năng mở rộng về sau.

Mặc dù đã đạt được một số kết quả tích cực, tuy nhiên nghiên cứu của nhóm vẫn còn một số điểm hạn chế và khó khăn sau:

- Về mặt dữ liệu: tuy bộ dữ liệu của nhóm có chất lượng khá tốt và độ đa dạng cao nhưng số lượng dữ liệu vẫn còn khá ít và trong đề tài này nhóm chỉ thực hiện trên 20 loại trái cây.
- Về mặt mô hình: tuy các mô hình đều đạt kết quả khá cao, nhưng cách tiếp cận hiện tại vẫn đang phụ thuộc khá nhiều vào dữ liệu.

5.2. Hướng phát triển

Với những kết quả đạt được hiện tại và những hạn chế trong dự án, trong tương lai, nhóm dự định sẽ tiếp tục phát triển thêm với các hướng đi sau:

- Về mặt dữ liệu: tiếp tục thu thập và mở rộng bộ dữ liệu với nhiều loại trái cây hơn cũng như thử nghiệm thêm nhiều phương pháp làm giàu dữ liệu.
- Về mặt mô hình: tiếp tục nghiên cứu và phát triển thêm các mô hình khác để giảm sự phụ thuộc vào dữ liệu và tăng cường tính tổng quát hóa của hệ thống.
- Về mặt hệ thống dịch vụ: cải thiện thêm giao diện và trải nghiệm người dùng. Thử nghiệm thêm các phương pháp tăng tốc hệ thống cũng như tiến hành đánh giá hệ thống một cách tổng quát và kĩ càng hơn.

TÀI LIỆU THAM KHẢO

[1] Yan Ju, Shan Jia, Jialing Cai, Haiying Guan, Siwei Lyu, "GLFF: Global and Local Feature Fusion for AI-synthesized Image Detection" (2023), https://arxiv.org/pdf/2211.08615.pdf

[2] Mihai Minut, "Fruits-262" (2021), kaggle.com/datasets/aelchimminut/fruits262

[3] Kritik Seth, "Fruits and Vegetables Image Recognition Dataset" (2021), kaggle.com/datasets/kritikseth/fruit-and-vegetable-image-recognition

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, "Deep Residual Learning for Image Recognition" (2015), https://arxiv.org/pdf/1512.03385.pdf

[5] Gao Huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger, "Densely Connected Convolutional Networks" (2018), https://arxiv.org/pdf/1608.06993.pdf

[6] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications" (2017), https://arxiv.org/pdf/1704.04861.pdf

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, Liang-Chieh Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks" (2018), https://arxiv.org/pdf/1801.04381v4.pdf

[8] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, Hartwig Adam, "Searching for MobileNetV3" (2019), https://arxiv.org/pdf/1905.02244.pdf

[9] Michael Munn, David Pitman, "Explainable AI for Practitioners" (2022)