*Bìa của bạn ở đây!*

# LỜI CẢM ƠN

Lời đầu tiên em xin trân thành cảm ơn quý thầy cô trong khoa đã trợ giúp chúng em một cách tận tâm khi chúng em thực hiện đề tài này. Đặc biệt em xin gửi lời cảm ơn đến thầy người đã tận tình hướng dẫn, giúp đỡ, chỉ bảo em trong suốt thời gian thực hiện khóa luận tốt nghiệp.

Đồng thời em xin trân trọng cảm ơn những tình cảm quý báu mà các thầy cô trong Trường Đại học Bách Khoa Đà Nẵng truyền đạt cho em những những kinh nghiệm, kỹ thuật và cách thức trong việc xây dựng đề tài này.

Tuy nhiên, do thời gian còn hạn chế em chưa thể phát huy hết những ý tưởng, khả năng hỗ trợ của ngôn ngữ lập trình và kỹ thuật lập trình vào đề tài này. Trong thời gian thực hiện đề tài không tránh khỏi những thiếu xót. Em rất mong nhận được sự đóng góp từ quý thầy cô và các bạn.

Em xin chân thành cảm ơn.

Đà Nẵng, ngày … tháng … năm 2023

Sinh viên thực hiện

# LỜI CAM ĐOAN

Em xin cam đoan, bài báo cáo này là công trình nghiên cứu của bản thân dưới sự hướng dẫn của thầy cô trong khoa. Các kết quả báo cáo được nêu trong chuyên đề tốt nghiệp này là trung thực và chưa từng được công bố trong các công trình nào khác.

Những tài liệu trong chuyên đề phục vụ cho phân tích và nhận xét được thu thập từ các nguồn khác nhau và có ghi rõ trong phần tài liệu tham khảo. Bên cạnh đó, chuyên đề còn sử dụng những nhận xét, đánh giá và số liệu từ nghiên cứu của các tác giả, cơ quan nghiên cứu khác cũng được nêu rõ trong phần tài liệu tham khảo ghi rõ nguồn gốc.

Nếu phát hiện có bất kỳ gian lận nào em xin chịu trách nhiệm về nội dung báo cáo của mình.

Đà Nẵng, ngày…tháng…năm 2023

Sinh viên thực hiện

# MỤC LỤC

[LỜI CẢM ƠN 2](#_Toc138705554)

[LỜI CAM ĐOAN 3](#_Toc138705555)

[MỤC LỤC 4](#_Toc138705556)

[DANH MỤC HÌNH ẢNH 6](#_Toc138705557)

[LỜI NÓI ĐẦU 7](#_Toc138705558)

[**CHƯƠNG I. CƠ SỞ LÝ THUYẾT 8**](#_Toc138705559)

[1.1 Tổng quan về Deep Learning 8](#_Toc138705560)

[1.2 Mạng nơron tích chập (CNN - Convolutional Neural Network) 8](#_Toc138705561)

[1.3 Kỹ thuật Transfer Learning 9](#_Toc138705562)

[1.4 Công cụ Teachable Machine 11](#_Toc138705563)

[1.5 Ngôn ngữ Python 12](#_Toc138705564)

[1.6 Ngôn ngữ Javascript và thư viện React Native 14](#_Toc138705565)

[1.7 API 16](#_Toc138705566)

[**CHƯƠNG II. HUẤN LUYỆN MÔ HÌNH 18**](#_Toc138705567)

[2.1 Thu thập dữ liệu về 30 món ăn Việt Nam 18](#_Toc138705568)

[2.1.1 Bộ dữ liệu thu thập 18](#_Toc138705569)

[2.1.2 Phương pháp thu thập 18](#_Toc138705570)

[2.2 Tiền xử lý dữ liệu 19](#_Toc138705571)

[2.2.1 Chuẩn hóa kích cỡ ảnh 19](#_Toc138705572)

[2.2.2 Giảm nhiễu ảnh 19](#_Toc138705573)

[2.2.3 Tăng cường ảnh 19](#_Toc138705574)

[2.3 Huấn luyện mô hình với MobileNetV2 và Transfer Learning 20](#_Toc138705575)

[2.3.1 Cấu trúc mạng MobileNetV2 20](#_Toc138705576)

[2.3.2 Huấn luyện Transfer Learning với mạng MobileNetV2 22](#_Toc138705577)

[2.4 Huấn luyện mô hình với Teachable Machine 27](#_Toc138705578)

[**CHƯƠNG III. XÂY DỰNG ỨNG DỤNG NHẬN DẠNG 30**](#_Toc138705579)

[3.1 Phát biểu bài toán 30](#_Toc138705580)

[3.2 Phân tích thiết kế 31](#_Toc138705581)

[3.2.1 Xác định Actor 31](#_Toc138705582)

[3.2.2 Biểu đồ Use case 31](#_Toc138705583)

[3.2.3 Biểu đồ hoạt động 32](#_Toc138705584)

[3.3 Xây dựng hệ thống 33](#_Toc138705585)

[3.4 Cài đặt hệ thống 35](#_Toc138705586)

[3.5 Đánh giá thực nghiệm 38](#_Toc138705587)

[**CHƯƠNG IV. KẾT LUẬN 40**](#_Toc138705588)

[4.1 Những nội dung đạt được 40](#_Toc138705589)

[4.2 Những hạn chế 40](#_Toc138705590)

[4.3 Hướng phát triển 41](#_Toc138705591)

[**TÀI LIỆU THAM KHẢO 42**](#_Toc138705592)

# DANH MỤC HÌNH ẢNH

[Hình 1: Tổng quan về Deep Learning 8](#_Toc138705593)

[Hình 2: Mô hình mạng nơron tích chập CNN 9](#_Toc138705594)

[Hình 3: Kỹ thuật Transfer Learning 10](#_Toc138705595)

[Hình 4: Công cụ Teachable Machine 11](#_Toc138705596)

[Hình 5: Ngôn ngữ lập trình Python 12](#_Toc138705597)

[Hình 6: Ngôn ngữ lập trình Javascript 14](#_Toc138705598)

[Hình 7: Thư viện React Native 14](#_Toc138705599)

[Hình 8: Mối liên kết giữa Server và Client qua API 16](#_Toc138705600)

[Hình 9: Bộ dữ liệu 6.000 ảnh về 30 món ăn Việt Nam 18](#_Toc138705601)

[Hình 10: Cấu trúc mạng MobileNet 20](#_Toc138705602)

[Hình 11: Cấu trúc mạng MobileNetV2 với bottleneck 22](#_Toc138705603)

[Hình 12: Code nạp dữ liệu ảnh train và val vào chương trình để huấn luyện 23](#_Toc138705604)

[Hình 13: Code tải mô hình MobileNetV2 24](#_Toc138705605)

[Hình 14: Code thêm các tầng Fully Connected để dự đoán trên dữ liệu món ăn 25](#_Toc138705606)

[Hình 15: Code cấu hình trọng số mô hình và huấn luyện nhận dạng món ăn 25](#_Toc138705607)

[Hình 16: Code lưu lại mô hình nhận dạng món ăn đã huấn luyện 26](#_Toc138705608)

[Hình 17: Kết quả dự đoán sau khi huấn luyện Transfer Learning với MobileNetV2 26](#_Toc138705609)

[Hình 18: Các bước huấn luyện mô hình với Teachable Machine 27](#_Toc138705610)

[Hình 19: Huấn luyện dữ liệu món ăn trên Teachable Machine 27](#_Toc138705611)

[Hình 20: Dự đoán kết quả món ăn sau khi huấn luyện với Teachable Machine 28](#_Toc138705612)

[Hình 21: Lưu lại mô hình sau khi huấn luyện Teachable Machine 29](#_Toc138705613)

[Hình 22: Biểu đồ User case tổng quát nhận dạng đồ ăn 31](#_Toc138705614)

[Hình 23: Biều đồ tuần tự chức năng dự đoán món ăn 32](#_Toc138705615)

[Hình 24: Hệ thống nhận dạng món ăn 33](#_Toc138705616)

[Hình 25: Kết quả dự đoán trả về dưới dạng Json 34](#_Toc138705617)

[Hình 26: Màn hình chính của ứng dụng 35](#_Toc138705618)

[Hình 27: Màn hình liệt kê hình ảnh có trong thư viện di động 36](#_Toc138705619)

[Hình 28: Màn hình hiển thị thông tin kết quả dự đoán 37](#_Toc138705620)

[Hình 29: Màn hình hiển thị công thức nấu đồ ăn đã dự đoán 38](#_Toc138705621)

# LỜI NÓI ĐẦU

Ngày nay, việc nhận dạng đồ ăn trên các thiết bị di động ngày càng trở nên phổ biến, giúp cho việc tìm kiếm thông tin và chia sẻ về các món ăn trở nên thuận tiện hơn bao giờ hết. Với sự bùng nổ của deep learning, việc sử dụng một số công nghệ mới như mạng neuron sâu để nhận dạng thức ăn dựa trên hình ảnh đã thể hiện tính hiệu quả và độ chính xác cao.

Nắm bắt được nhu cầu trên nên em quyết định thực hiện việc xây dựng ứng dụng di động nhận dạng đồ ăn với deep learning, bao gồm cả frontend và backend, cách kết nối với API image recognition, cũng như các kỹ thuật xử lý dữ liệu trong quá trình nhận dạng.

Thông qua đó cũng là cơ hội để em tìm hiểu và nghiên cứu chuyên sâu về các mô hình mạng neuron sâu như VGG16, VGG19 và MobileNet V2, cùng với các kỹ thuật như transfer learning, fine-tuning để điều chỉnh các mô hình nhằm tăng cường độ chính xác nhận dạng. Các phương pháp và thuật toán sử dụng để xử lý dữ liệu trong quá trình nhận dạng cũng sẽ được trình bày chi tiết.

Kết quả sau khi thực hiện đề tài và bài báo cáo này sẽ cho em có thêm các kiến thức cần thiết về tiềm năng và triển vọng của ứng dụng nhận dạng đồ ăn trên thiết bị di động sử dụng deep learning, cũng như những thách thức và khó khăn chúng ta cần đối mặt trong quá trình phát triển ứng dụng liên quan đến học máy cũng như học sâu.

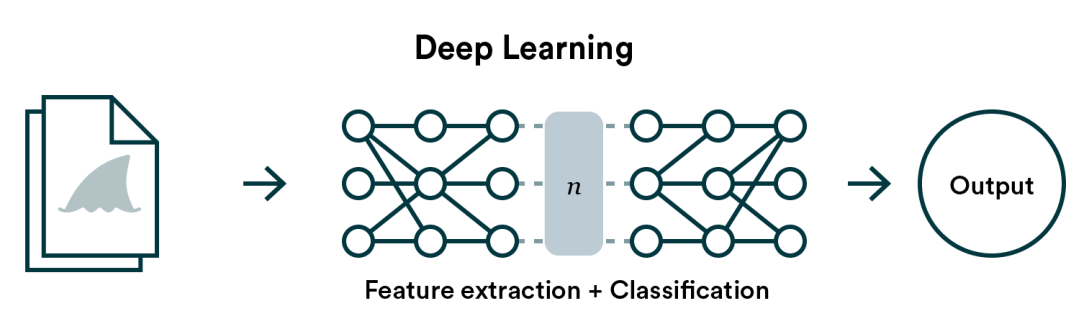
Lời cuối cùng, em xin trân thành cảm ơn thầy cô trong khoa Công nghệ thông tin Trường đại học Bách Khoa Đà Nẵng đã tận tâm theo dõi, chỉ bảo và hướng dẫn chúng em một cách chu đáo trong suốt quá trình chúng em nghiên cứu và thực hiện đề tài tốt nghiệp này.

Em xin cảm ơn!

# CHƯƠNG I. CƠ SỞ LÝ THUYẾT

## 1.1 Tổng quan về Deep Learning

Deep learning là một phương pháp của trí tuệ nhân tạo (AI), chuyên sử dụng mạng neuron nhân tạo với nhiều lớp ẩn để học và xử lý dữ liệu. Deep learning có thể tự động phát hiện và học các đặc trưng (features) của dữ liệu, từ đó giúp máy tính hiểu được dữ liệu mang tính chất bất định và chưa được biểu diễn dưới dạng các công thức toán học cụ thể.



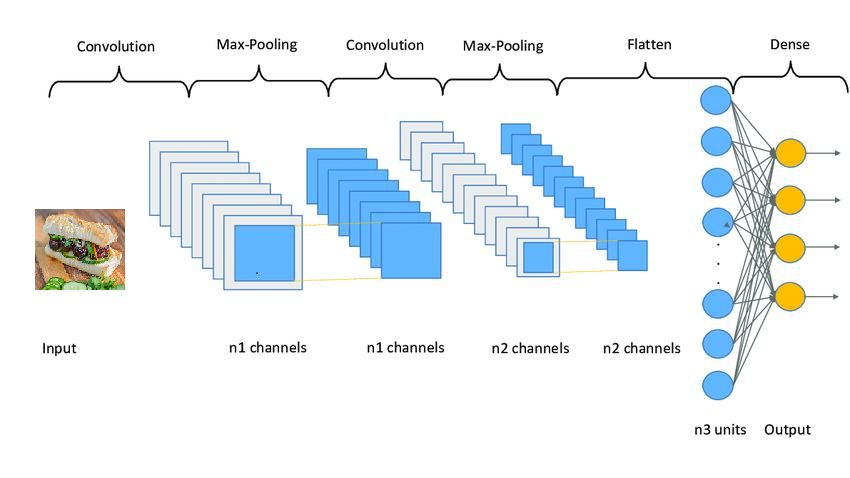
Hình 1: Tổng quan về Deep Learning

Các mô hình deep learning thường được huấn luyện trên các tập dữ liệu lớn và đa dạng, nhằm giúp chúng học được các đặc trưng của dữ liệu một cách tổng quát hơn. Các ứng dụng của deep learning rất đa dạng, từ nhận dạng ảnh, xử lý ngôn ngữ tự nhiên, đến dự báo và nhận diện giọng nói.

Với sự phát triển của các mô hình deep learning, đặc biệt là mạng neuron sâu (deep neural networks), các bài toán đã đạt được độ chính xác và hiệu quả cao hơn rất nhiều so với những phương pháp truyền thống trước đây. Tuy nhiên, việc huấn luyện các mô hình deep learning vẫn đòi hỏi sự đầu tư kỹ thuật và tài nguyên tính toán lớn.

## 1.2 Mạng nơron tích chập (CNN - Convolutional Neural Network)

Mạng CNN - Convolutional Neural Network là một trong những mô hình mạng phổ biến trong các hệ thống nhận dạng, xử lý ảnh, âm thanh và nhiều lĩnh vực khác. Mạng CNN có khả năng xây dựng liên kết chỉ sử dụng một phần cục bộ trong ảnh kết nối đến nút trong lớp tiếp theo thay vì toàn bộ ảnh như trong mạng neural network truyền thẳng.



Hình 2: Mô hình mạng nơron tích chập CNN

Mạng CNN thường được chia thành một số lớp cơ bản như lớp tích chập (Convolutional), lớp lấy mẫu (Pooling), lớp kích hoạt phi tuyến hàm ReLU (Rectified Linear Unit) và lớp kết nối đầy đủ (Fully connected layer). Lớp tích chập giúp mạng học được các đặc trưng cục bộ của ảnh thông qua việc quét qua ảnh với các filter. Trong khi đó, lớp lấy mẫu giúp giảm kích thước đầu vào và giữ lại các giá trị quan trọng nhất.

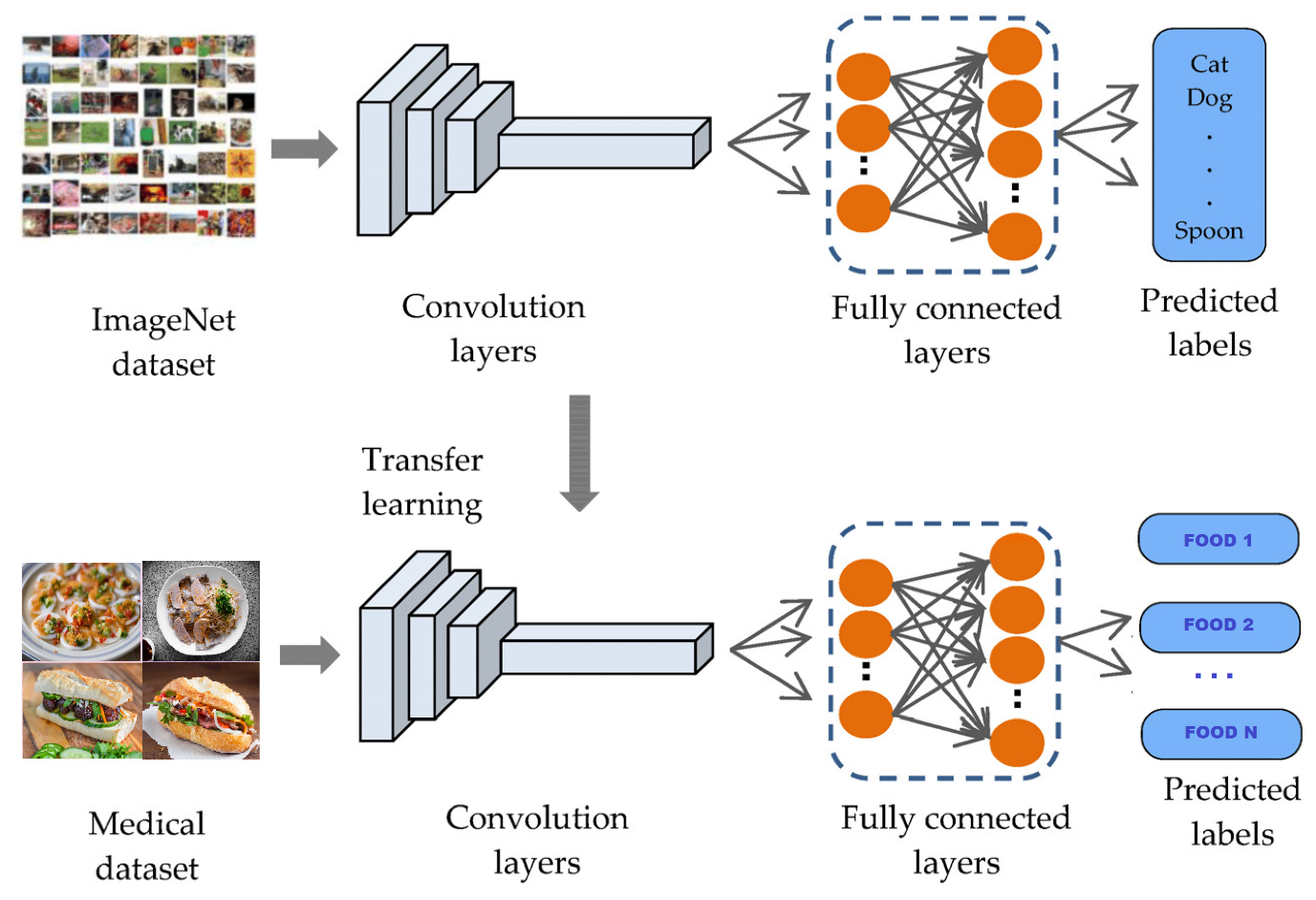
Lớp kích hoạt phi tuyến hàm ReLU được sử dụng để kích hoạt các neuron trong mạng CNN, giúp mạng học được các đặc trưng phi tuyến tính và giảm việc vanishing gradient. Cuối cùng, lớp kết nối đầy đủ được sử dụng để kết nối đầu ra từ các lớp trước đó với các lớp đầu ra.

Từ khả năng học được các đặc trưng cục bộ của ảnh thông qua lớp tích chập và khả năng giảm kích thước đầu vào thông qua lớp lấy mẫu, mạng CNN đã đạt được kết quả tốt trong các bài toán nhận dạng ảnh, xử lý âm thanh và nhiều lĩnh vực khác.

## 1.3 Kỹ thuật Transfer Learning

Transfer learning và fine-tuning là hai kỹ thuật quan trọng trong học sâu (deep learning) giúp giảm thiểu thời gian và tài nguyên cần thiết để huấn luyện mạng nơ-ron.

Transfer learning là quá trình sử dụng mô hình đã được huấn luyện trước đó trên một tác vụ khác, sau đó "transfer" lại các đặc trưng đã học được bằng cách sử dụng một số phương pháp như cắt ngang (cut-off), bỏ bớt các lớp phía trên hoặc sử dụng nguyên mô hình đã huấn luyện để giúp huấn luyện những tác vụ mà dữ liệu huấn luyện không đủ. Điều này giúp tiết kiệm thời gian huấn luyện và tăng độ chính xác của quá trình huấn luyện.



Hình 3: Kỹ thuật Transfer Learning

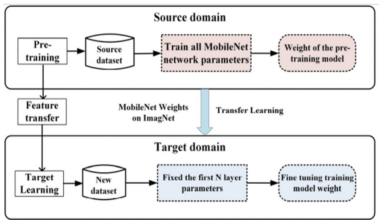
Trong hình trên, chúng ta cần huấn luyện mô hình với tập dữ liệu là các món ăn mà ta đã thu thập được trước đó. Khi áp dụng kỹ thuật Transfer Learning ta sẽ tận dụng lại mô hình đã được huấn luyện trước đó (pre-trained model). Mô hình đã được huấn luyện trước với bộ dữ liệu ImageNet và đã cho ra kết quả rất tốt khi kiểm tra.

Tuy nhiên, khi sử dụng mô hình đó vào bài toán phân loại món ăn ta chỉ tận dụng các lớp (tầng) trích chọn đặc chưng (các tầng đầu tiên ) để trích xuất ra các đặc trưng của món ăn như, màu sắc, độ dài, hình dạng…. Thế nhưng, các tầng dự đoán, phân loại về món ăn thì lại đặt ở tầng cuối cùng chính là các lớp đầy đủ ở phía mô hình của chúng ta cần xây dựng.

Phương pháp Tramsfer Learning giúp giảm thiểu thời gian cần thiết để huấn luyện một mạng nơ-ron từ đầu và tăng độ chính xác của mô hình vì mô hình đã được huấn luyện sẵn trên những tác vụ khác. Phù hợp với các bài toán phân loại mà dự liệu của ta thu thập được quá ít, không đủ để miêu tả một cách chính xác khi xây dựng một mô hình CNN mới từ đầu.

## 1.4 Công cụ Teachable Machine

Teachable Machine sử dụng phương pháp Transfer Learning để tìm các mẫu và xu hướng trong hình ảnh hoặc mẫu âm thanh, đồng thời tạo ra một mô hình phân loại đơn giản và dễ dàng trong vòng vài giây. Để xây dựng các lớp phân loại hình ảnh, Teachable Machine dựa trên mạng nhận dạng hình ảnh được đào tạo trước gọi là MobileNet - Mạng này đã được đào tạo để nhận ra hàng ngàn đối tượng (chẳng hạn như mèo, chó, ô tô, trái cây và chim, đồ ăn).



Hình 4: Công cụ Teachable Machine

Hình trên cho thấy quá trình đào tạo TL-MobileNet dựa trên mô hình học chuyển giao. Các trọng số và đặc trưng của mô hình MobileNet được đào tạo trước trong bộ miền nguồn (Source domain), sau đó chúng được chuyển sang miền đích (Target domain) để phân loại. Miền đích không sử dụng khởi tạo ngẫu nhiên để bắt đầu quá trình học dữ liệu ngay từ đầu và các tham số mô hình được chia sẻ giữa miền nguồn và miền đích, vì vậy phương pháp này sẽ giúp cải thiện hiệu quả học tập.

Với Teachable Machine chúng ta có thể tạo các mẫu dữ liệu trực tiếp từ webcam hoặc các dữ liệu đã có từ trước đó để đào tạo mô hình. Công cụ này của Google sử dụng thuật toán KNN (K-Nearest Neighbors) một trong những thuật toán học máy có giám sát, đơn giản và dễ triển khai - để phân loại hình ảnh. Giao diện Teachable Machine mà chúng ta sử dụng chỉ là front-end do Google tạo ra, nó không bao gồm bất cứ thông tin mã nguồn nào về Back-end, hay nói cách khác thông tin về các lớp mạng ẩn không được Google công bố nên chúng ta không thể biết được về chúng.

## 1.5 Ngôn ngữ Python

Python là một ngôn ngữ lập trình bậc cao được sử dụng cho các mục đích lập trình đa năng, do tác giả Guido van Rossum tạo ra và lần đầu ra mắt vào năm 1991. Python là ngôn ngữ dễ sử dụng, trong Python có các đặc tính như: dễ đọc, dễ viết và dễ nhớ. Đặc tính này giúp cho những người mới theo học lập trình dễ dàng tiếp cận và học tập ngôn ngữ Python.



Hình 5: Ngôn ngữ lập trình Python

Ngôn ngữ lập trình Python cung cấp cho chúng ta các thư viện tiện ích liên quan đến việc xây dựng các mô hình học máy và học sâu. Các thư viện đó phải kể đến như: Tensorflow, Keras, Numpy, PyTorch, Pandas, Matplotlib… Ngoài ra, Python còn cung cấp thêm các thư viện để xây dựng 1 web server phục vụ cho mục đích triển khai các mô hình học máy cho các thiết bị cùng sử dụng.

**Một vài thư viện của Python được sử dụng trong đề tài**

* Thư viện TensorFlow:

TensorFlow là một thư viện mã nguồn mở được phát triển bởi Google, được sử dụng rộng rãi trong lĩnh vực học máy và trí tuệ nhân tạo. Nó có thể được sử dụng để xây dựng và huấn luyện các mô hình học máy và mạng nơ-ron sâu.

* Thư viện Keras:

Keras là một thư viện học sâu mã nguồn mở được phát triển bởi François Chollet, cũng là một phần của TensorFlow. Nó cung cấp một API bậc cao để xây dựng, huấn luyện và triển khai các mạng nơ-ron sâu.

* Thư viện NumPy:

NumPy là một thư viện Python cho tính toán khoa học. Nó cung cấp các đối tượng mảng và các hàm tính toán trên mảng, cung cấp cho người dùng các phép tính toán phức tạp để xử lý dữ liệu.

* Thư viện OpenCV:

OpenCV (Open Source Computer Vision Library) là một thư viện mã nguồn mở được sử dụng rộng rãi trong lĩnh vực thị giác máy tính và xử lý ảnh. Nó cung cấp các công cụ để xử lý ảnh và video, nhận dạng đối tượng, trích xuất đặc trưng, và rất nhiều thứ khác.

* Thư viện Flask:

Flask là một framework Web miễn phí và mã nguồn mở cho Python. Nó được sử dụng để xây dựng các ứng dụng web theo cách phi truyền thống, cung cấp các tính năng cơ bản như routing, xử lý HTTP, khung nhìn và sự kiểm soát chuỗi công việc.

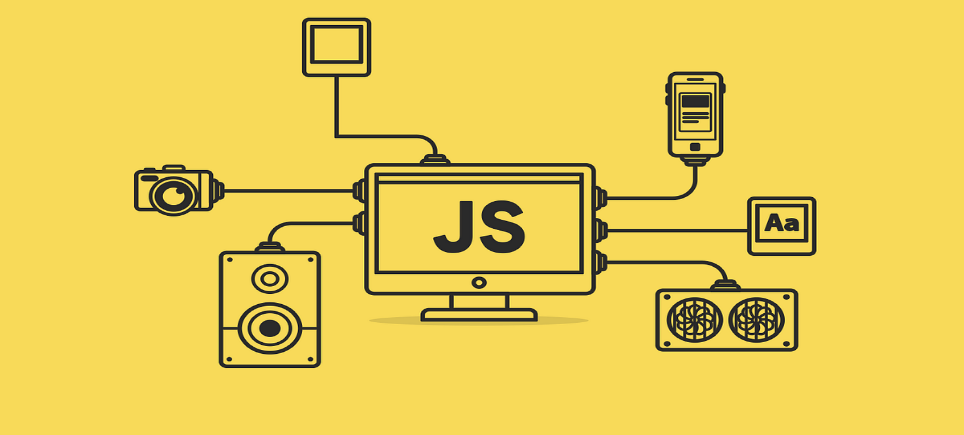
* Thư viện Pillow:

Pillow là một thư viện mã nguồn mở cho Python đóng vai trò là một "fork" của thư viện PIL (Python Imaging Library). Nó cung cấp các công cụ để xử lý ảnh, bao gồm mở, lưu, chỉnh sửa và chuyển đổi ảnh, cắt, ghép, thay đổi kích thước, v.v. Thư viện này rất hữu ích cho các ứng dụng xử lý ảnh và đồ họa.

Tóm lại, các thư viện trên là các công cụ quan trọng đối với việc xây dựng một mô hình học máy, thị giác máy tính, tính toán khoa học, và phát triển khai các mô hình AI lên web với Python. Chúng được sử dụng rộng rãi trong cộng đồng Python, và cung cấp cho người dùng các tính năng mạnh mẽ để xây dựng các ứng dụng và giải quyết các vấn đề cho các dự án của họ.

## 1.6 Ngôn ngữ Javascript và thư viện React Native

JavaScript là ngôn ngữ lập trình phía client-side (mặt khách) được sử dụng rộng rãi để xây dựng các ứng dụng web tương tác. Nó có thể được sử dụng để xử lý các sự kiện trên trang web, thay đổi dữ liệu trong HTML và CSS, và thực hiện các thao tác phía client-side khác.



Hình 6: Ngôn ngữ lập trình Javascript

React Native là một framework để phát triển ứng dụng di động cho cả IOS và Android bằng ngôn ngữ lập trình JavaScript. Nó cho phép người dùng xây dựng các ứng dụng di động một cách nhanh chóng và hiệu quả bằng việc sử dụng các điểm mạnh của JavaScript và thư viện UI của React.



Hình 7: Thư viện React Native

Trong quá trình xây dựng ứng dụng di động, chúng ta sẽ sử dụng React Native để xây dựng ra một giao diện chương trình ở phía người dùng cuối. Thông qua React Native ứng dụng sẽ được cấp quyền chọn ảnh cũng như gửi ảnh từ điện thoại di động lên phía server để yêu cầu dự đoán món ăn và nhận lấy kết quả dự đoán hiển thị ra cho phía người sử dụng thấy.

**Một vài package của React Native được sử dụng trong đề tài:**

* react-navigation

Được sử dụng trong React Native để quản lý các đường dẫn (routing) trong ứng dụng, cho phép người dùng tạo các màn hình và chuyển đổi giữa các màn hình dễ dàng hơn. Nó cung cấp cho người dùng một loạt các phương thức để tạo và quản lý các màn hình, bao gồm cả các phương thức để chuyển đổi giữa các màn hình.

* axios

Sử dụng trong việc gửi các yêu cầu HTTP từ client-side của ứng dụng React Native đến server-side. Với axios, người dùng có thể tạo các yêu cầu HTTP GET, POST, PUT, DELETE, PATCH… để tương tác với các API trên server-side.

* expo

Cung cấp các dịch vụ, công cụ và thư viện cho việc phát triển, kiểm thử và triển khai ứng dụng React Native. Expo hỗ trợ người dùng trong việc cập nhật phiên bản, ứng dụng và hỗ trợ các thiết bị di động.

* expo-image-picker

Được sử dụng để lấy ảnh từ bộ nhớ trong thiết bị hoặc chụp ảnh từ camera và trả về đối tượng ảnh (image object). Package này cung cấp các phương thức đơn giản để lấy ảnh và sử dụng trong ứng dụng.

* react-native-tab-view

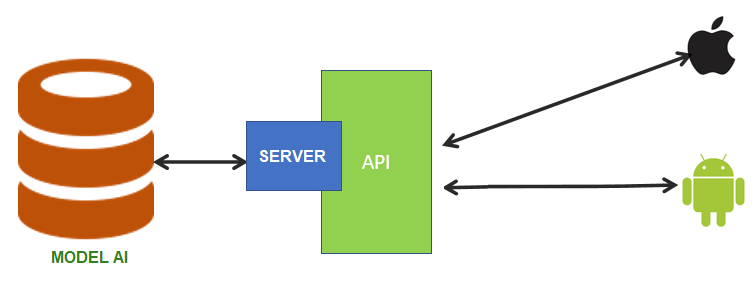
Được sử dụng để tạo các tab trong ứng dụng React Native. Với react-native-tab-view, người dùng có thể tạo nhiều màn hình và chuyển đổi giữa chúng bằng cách sử dụng các tab riêng biệt.

* react-native-loading-spinner-overlay

Được sử dụng để hiển thị spinner hoặc progress bar trong ứng dụng React Native. Package này cung cấp các phương thức đơn giản để hiển thị spinner trong quá trình tải dữ liệu hoặc khi đang chờ xử lý.

## 1.7 API

API là một từ viết tắt của cụm từ "Application Programming Interface". Đây là một tập hợp các quy tắc, giao thức, và công nghệ được sử dụng để trao đổi dữ liệu giữa ứng dụng, hệ thống hoặc máy tính khác nhau mà không cần phải chia sẻ mã nguồn hoặc giao diện người dùng.



Hình 8: Mối liên kết giữa Server và Client qua API

API định nghĩa các phương thức và giao thức mà một ứng dụng hoặc hệ thống khác có thể sử dụng để truy cập vào các chức năng của ứng dụng hoặc hệ thống này. Điều này cho phép các ứng dụng của bên thứ ba truy cập vào các tính năng của ứng dụng hoặc hệ thống bằng cách sử dụng API, mà không cần phải có quyền truy cập vào mã nguồn của ứng dụng hoặc hệ thống.

Trong ứng dụng của chúng ta hoạt động giựa trên 2 phần: Phần Client phía giao diện người dùng di động và phần Server phía máy chủ. Client sẽ đảm nhiệm công việc với người dùng cuối đó là hiển thị giao diện trên thiết bị di động và giúp người dùng chọn gửi ảnh lên Server. Phía Server sẽ đảm nhiệm công việc nhận ảnh gửi lên từ Client và đưa vào mô hình đã train để thực hiện dự đoán kết quả, sau khi dự đoán thành công sẽ trả về kết quả là thông tin dự đoán món ăn cho người dùng ở phía Client.

Để 2 ứng dụng là Client và Server có thể kết nối và trao đổi dữ liệu với nhau, ta cần có API – đóng vai trò như cầu nối dữa Client – Server. Khi này, việc gửi nhận hình ảnh cũng như trả về kết quả sẽ thông qua API làm trung gian. Điều này sẽ không yêu cầu người dùng phải truy cập lên trực tiếp mã nguồn của Server mới có thể sử dụng được model để dự đoán món ăn và đưa ra kết quả.

# CHƯƠNG II. HUẤN LUYỆN MÔ HÌNH

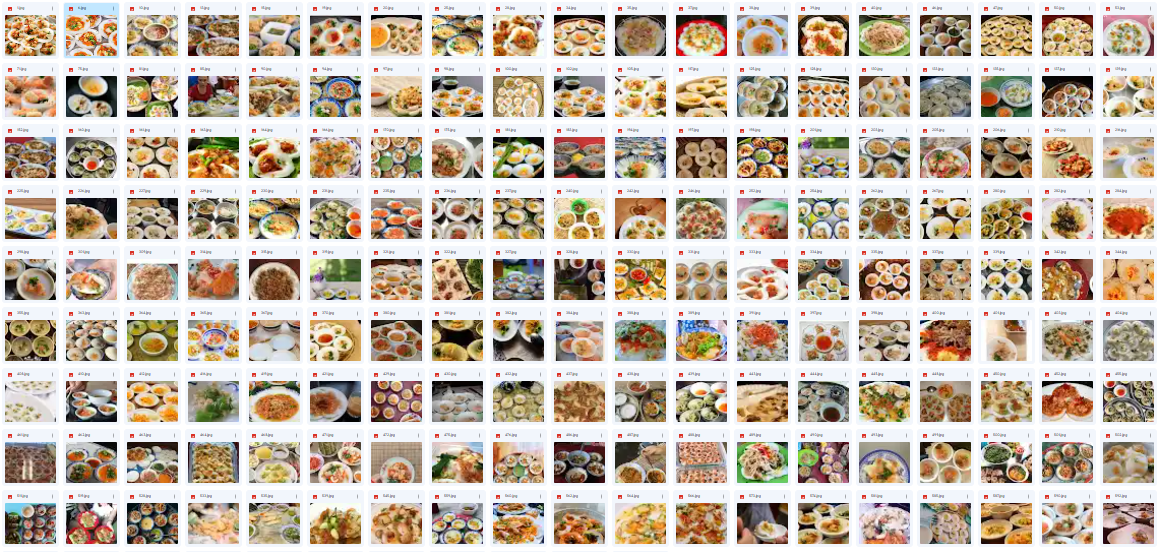
## 2.1 Thu thập dữ liệu về 30 món ăn Việt Nam

### 2.1.1 Bộ dữ liệu thu thập

Các dự liệu về ẩm thực Việt Nam rất đa dạng. Chúng ta có thể tìm kiếm và thu thập trong thực tế hoặc trên không gian internet. Vì sự hạn chế về thời gian cũng như các công cụ xử lý dữ liệu nên em quyết định chọn ra 30 món ăn để thu thập dữ liệu. nên các món ăn được chọn bao gồm 30 món ăn đặc trưng của từng miền: Bánh bèo, Bánh bột lọc, Bánh căn, Bánh canh, Bánh chưng, Bánh cuốn, Bánh đúc, Bánh giò, Bánh khọt, Bánh mì, Bánh pía, Bánh tét, Bánh tráng nướng, Bánh xèo, Bún bò huế, Bún đậu mắm tôm, Bún mắm, Bún riêu, Bún thịt nướng, Cá kho tộ , Canh chua, Cao lầu, cháo lòng, Cơm tấm, Gỏi cuốn, Hủ tiếu, Mì quảng, Nem chua, Phở, Xôi xéo.

### 2.1.2 Phương pháp thu thập

Bộ dữ liệu thu thập được gồm 30 món ăn Việt Nam với tổng cộng 6.000 hình ảnh được chụp với mọi góc cạnh của món ăn. Bộ dữ liệu được tìm kiếm và thu thập từ nhiều nguồn khác nhau trên internet, như các blog ẩm thực, trang mạng xã hội về ẩm thực, trang web trao đổi kinh nghiệm nấu nướng và ẩm thực.



Hình 9: Bộ dữ liệu 6.000 ảnh về 30 món ăn Việt Nam

Thông tin thu thập được bao gồm các tài nguyên đa dạng như hình ảnh, video và các bài viết về các món ăn, thông tin về nguyên liệu cần có và cách nấu chế biến của mỗi món ăn. Bên cạnh đó, các thông tin về lịch sử phát triển của món ăn, văn hóa ẩm thực đặc trưng của từng miền đất nước cũng được thu thập để cung cấp cho người sử dụng đầy đủ, toàn diện và chi tiết nhất về các món ăn Việt Nam.

Tuy nhiên, khi áp dụng bộ dữ liệu này vào mô hình để huấn luyện, em sẽ chỉ chọn ra các hình ảnh của từng món ăn và đưa chúng vào từng thư mục riêng biệt để chuẩn bị cho quá trình huấn luyện tiếp theo.

## 2.2 Tiền xử lý dữ liệu

Tiền xử lý dữ liệu ảnh của các món ăn là một bước quan trọng trong quá trình xử lý dữ liệu dạng ảnh, đặc biệt là trong trường hợp thu thập dữ liệu ảnh liên quan đến 30 món ăn như trường hợp này.Tiền xử lý dữ liệu ảnh là bước quan trọng trong việc chuẩn hóa, tối ưu và nâng cao độ tin cậy của dữ liệu ảnh trước khi đưa vào đào tạo mô hình. Việc tiền xử lý dữ liệu ảnh trong trường hợp thu thập dữ liệu ảnh về 30 món ăn Việt Nam cũng rất cần thiết để đạt được độ chính xác cao trong quá trình xử lý dữ liệu.

### 2.2.1 Chuẩn hóa kích cỡ ảnh

Trong quá trình thu thập dữ liệu, có thể thu thập được những bức ảnh với kích cỡ khác nhau. Việc mở to hay thu nhỏ kích cỡ ảnh sẽ làm thay đổi khối lượng thông tin trên mỗi pixel, dẫn đến mất mát thông tin trong quá trình xử lý ảnh. Việc tiền xử lý dữ liệu ảnh cho phép chuẩn hóa kích cỡ của ảnh, giúp bảo toàn được nhiều thông tin trên các pixel.

### 2.2.2 Giảm nhiễu ảnh

Trong quá trình thu thập, những hình ảnh có thể bị mờ, nhòe, hoặc bị nhiễu. Khi không xử lý nhiễu ảnh, các nhiễu sẽ gây ra ảnh hưởng lớn đến độ chính xác của mô hình xử lý dữ liệu ảnh. Việc tiền xử lý nhằm loại bỏ các nhiễu không cần thiết, giúp cải thiện chất lượng ảnh, tăng độ chính xác của mô hình xử lý dữ liệu ảnh.

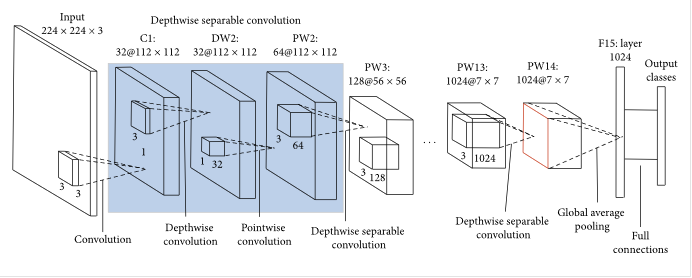
### 2.2.3 Tăng cường ảnh

Trong quá trình tiền xử lý dữ liệu ảnh, có thể sử dụng các kỹ thuật tăng cường ảnh như xoay, lật, thay đổi độ sáng tối, cắt tỉa ảnh... để tạo ra những phiên bản ảnh mới nhằm gia tăng đa dạng và số lượng dữ liệu cho mô hình đào tạo. Tăng cường ảnh có thể giúp cải thiện khả năng tổng quát hóa (generalization) của mô hình và giúp hạn chế overfitting.

## 2.3 Huấn luyện mô hình với MobileNetV2 và Transfer Learning

### 2.3.1 Cấu trúc mạng MobileNetV2

MobileNet là một mạng học sâu được thiết kế để sử dụng trên các thiết bị có tài nguyên tính toán hạn chế, như điện thoại thông minh. MobileNet có thể được tăng tốc độ huấn luyện và tính toán bằng cách sử dụng các convolution dùng chung (Depthwise Separable Convolution) để giảm số lượng tham số và tính toán.



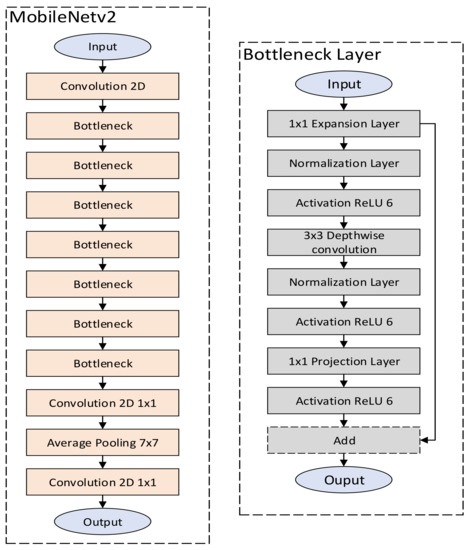
Hình 10: Cấu trúc mạng MobileNet

MobileNetV2 là một phiên bản cải tiến của kiến trúc MobileNet, được phát triển bởi nhóm nghiên cứu tại Google. Kiến trúc này là một mạng CNN (Convolutional Neural Network) cho việc phát hiện đối tượng và nhận diện hình ảnh trên các thiết bị di động với tốc độ và độ chính xác cao.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Layer | Output Size | Stride | Input Size | Operator |
| 0 | 112x112x3 |  | 112x112x3 | - |
| 1 | 112x112x32 | 1 | 112x112x3 | conv3x3 + Batch Normalization + ReLU |
| 2 | 112x112x16 | 1 | 112x112x32 | bottleneck |
| 3 | 56x56x24 | 2 | 112x112x16 | bottleneck |
| 4 | 56x56x24 | 1 | 56x56x24 | bottleneck |
| 5 | 28x28x32 | 2 | 56x56x24 | bottleneck |
| 6 | 28x28x32 | 1 | 28x28x32 | bottleneck |
| 7 | 14x14x64 | 2 | 28x28x32 | bottleneck |
| 8 | 14x14x64 | 1 | 14x14x64 | bottleneck |
| 9 | 14x14x64 | 1 | 14x14x64 | bottleneck |
| 10 | 14x14x96 | 1 | 14x14x64 | bottleneck |
| 11 | 7x7x160 | 2 | 14x14x96 | bottleneck |
| 12 | 7x7x160 | 1 | 7x7x160 | bottleneck |
| 13 | 7x7x320 | 1 | 7x7x160 | conv1x1 + Batch Normalization + ReLU |
| 14 | 1x1x1280 | 1 | 7x7x320 | conv1x1 + Batch Normalization + ReLU |
| 15 | 1x1x1000 | 1 | 1x1x1280 | Global Avg Pooling |
| 16 | 1x1x1000 |  | 1x1x1000 | Softmax |

Cấu trúc mạng MobilenetV2 bao gồm một phần thân mạng (body architecture) và một phần đầu mạng (head architecture). Phần thân mạng của MobilenetV2 sử dụng kỹ thuật phân tách convolution (separable convolution), kết hợp với các kỹ thuật khác như bottleneck, depthwise convolution, inverted residual và linear bottleneck để giảm số lượng tham số và tăng tốc độ xử lý.

Cụ thể, phần thân mạng của MobilenetV2 được thiết kế thành các khối bottleneck, trong đó có hai loại khối: khối bình thường và khối mở rộng (expansion block). Khối bình thường bao gồm các lớp Convolution, BatchNormalization và ReLU, trong khi khối mở rộng bao gồm thêm một lớp Convolution để mở rộng số kênh của đầu vào trước khi tiến hành các lớp Convolution, BatchNormalization và ReLU.



Hình 11: Cấu trúc mạng MobileNetV2 với bottleneck

Các khối bottleneck được kết hợp với kỹ thuật shortcut (residual connection) để giúp mô hình học tốt hơn và giảm thiểu hiện tượng vanishing gradient. Ngoài ra, MobilenetV2 cũng sử dụng các kỹ thuật như squeeze-and-excitation và global average pooling để tăng cường khả năng phân biệt đối tượng và tối ưu hóa mô hình.

### 2.3.2 Huấn luyện Transfer Learning với mạng MobileNetV2

Trong quá trình thu thập dữ liệu ảnh, chúng ta đã thu thập 30 món ăn ở Việt Nam (tương ứng với 30 nhãn huấn luyện). Toàn bộ dữ liệu được lưu trong thư mục **images**, trong thư mục này chúng ta lại có 2 thư mục là **images/Train** và **images/Val** – nhằm phân dữ liệu cho quá trình huấn luyện và quá trình xác thực.

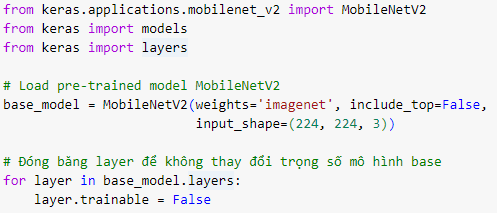
|  |
| --- |
| images  Train  Banh mi  Banh canh  ….  Pho  Val  Banh mì  Banh canh  ….  Pho |

Để thực hiện huấn luyện cho các dữ liệu được lưu trữ trong các thư mục trên, trước tiên ta sẽ phải nạp các dữ liệu là từng hình ảnh tương ứng vào một mảng chứa các ảnh dùng để huấn luyện trong thư mục **images/Train** và nạp vào một mảng chứa các ảnh dùng để xác thực trong thư mục **images/Val**.



Hình 12: Code nạp dữ liệu ảnh train và val vào chương trình để huấn luyện

Khi thực hiện kỹ thuật Transfer Learning dựa trên mạng MobileNetV2, khi đó mạng MobileNetV2 được sử dụng như là một mô hình base để giúp huấn luyện một mô hình mới trên các tập dữ liệu riêng biệt (món ăn). Thay vì huấn luyện một mô hình từ đầu trên tập dữ liệu mới, chúng ta có thể tái sử dụng mô hình MobileNetV2, đã được huấn luyện sẵn trên tập dữ liệu ImageNet, để trích xuất các đặc trưng (features) từ các hình ảnh. Sau đó, các đặc trưng này được đưa vào các lớp Fully Connected để tạo ra một mô hình mới cho bài toán phân loại riêng của chúng ta.



Hình 13: Code tải mô hình MobileNetV2

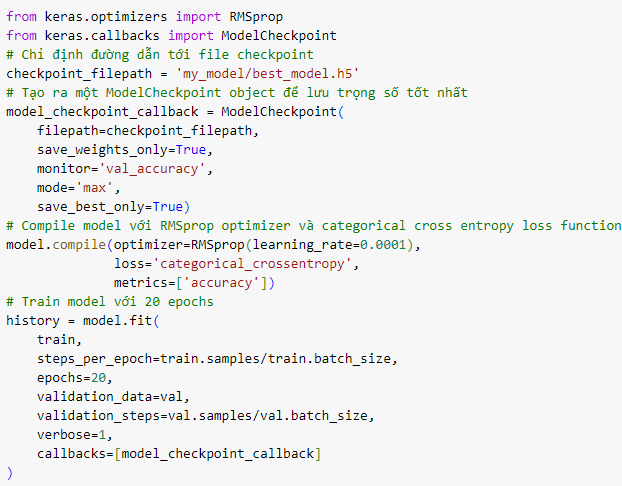
Bước đầu tiên trong việc huấn luyện Transfer Learning với MobileNetV2 là tạo mô hình base bằng cách tải mô hình đã được huấn luyện trên tập dữ liệu ImageNet trong thư viện Keras. Mô hình có kích thước đầu vào là (224 x 224 x 3) và mục đích của ta chỉ sử dụng mô hình để phục vụ cho công việc trích chọn đặc trưng của dữ liệu nên chúng ta loại bỏ lớp fully connected cuối cùng của mô hình để tạo mô hình đơn giản hơn. Ngoài ra, ta sẽ đóng băng các layer của mô hình MobileNetV2 để tránh việc train lại các trọng số của mô hình đó.

Sau khi đã tạo mô hình base, chúng ta sẽ thêm các lớp fully connected lên đầu của mô hình để phù hợp với bài toán phân loại món ăn Việt Nam. Các lớp mới bao gồm Global Average Pooling Layer và lớp Flatten để giảm chiều của feature maps xuống 1 chiều và hai lớp fully connected với số lượng neurons phù hợp với số lượng lớp cần phân loại.



Hình 14: Code thêm các tầng Fully Connected để dự đoán trên dữ liệu món ăn

Tiếp theo, chúng ta sẽ compile mô hình và huấn luyện nó trên tập dữ liệu đã chuẩn bị. Việc huấn luyện này bao gồm việc lựa chọn thuật toán tối ưu và hàm mất mát phù hợp, cùng với tham số khác nhau như epoch và batch size. Sau quá trình huấn luyện, chúng ta sẽ đánh giá mô hình trên tập dữ liệu test để xác định độ chính xác của mô hình.



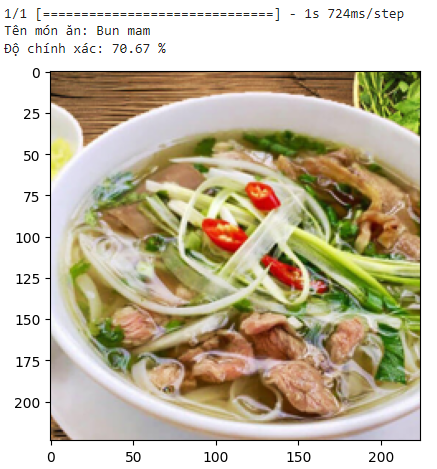
Hình 15: Code cấu hình trọng số mô hình và huấn luyện nhận dạng món ăn

Sau khi quá trình huấn luyện thành công, ta sẽ lưu lại mô hình để có thể dùng triển khai lên các hệ thống, ứng dụng cho phép nhận dạng món ăn.



Hình 16: Code lưu lại mô hình nhận dạng món ăn đã huấn luyện

Kết thúc quá trình huấn luyện, mô hình đã được huấn luyện trên tập dữ liệu phân loại 30 món ăn Việt Nam và có thể được sử dụng để phân loại các món ăn từ các ảnh mới.



Hình 17: Kết quả dự đoán sau khi huấn luyện Transfer Learning với MobileNetV2

Trong hình trên, mô hình dự đoán mất khoảng 1 giây và đưa ra được tên món ăn với độ chính xác khoảng 70%.

## 2.4 Huấn luyện mô hình với Teachable Machine

Như đã đề cập ở phần [1.3 Công cụ Teachable Machine,](#_1.4_Công_cụ) công cụ học máy này sử dụng phương pháp học chuyển giao, một kỹ thuật ML để tìm các mẫu và xu hướng trong hình ảnh hoặc mẫu âm thanh, đồng thời tạo ra một mô hình phân loại đơn giản và dễ dàng trong vòng vài giây. Khi cầ huấn luyện một mô hình nhận dạng món ăn với Teachable Machine ta chỉ cần thực hiện 3 bước:

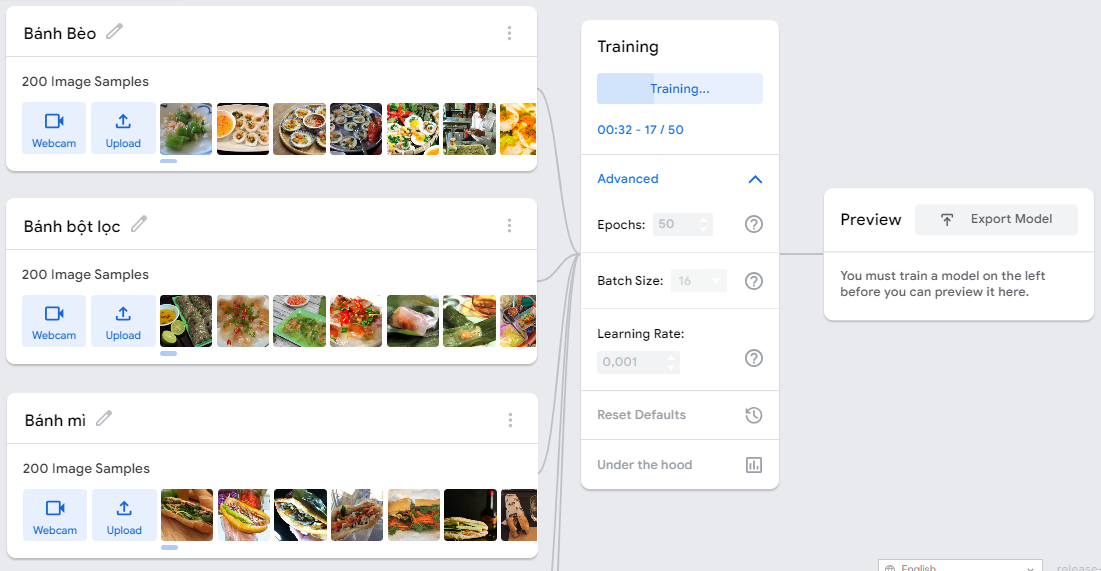


Hình 18: Các bước huấn luyện mô hình với Teachable Machine

Bước 1: Thu thập dữ liệu các món ăn ( đã thu thập 6000 ảnh cho 30 món ăn )

Bước 2: Sử dụng nguồn dữ liệu đã thu thập được ở bước 1 tải dữ liệu huấn luyện lên website [https://teachablemachine.withgoogle.com/](https://teachablemachine.withgoogle.com/%20%20)

Bước 3: Lưu lại mô hình đã được huấn luyện và thực hiện nhận diện món ăn với mô hình vừa huấn luyện

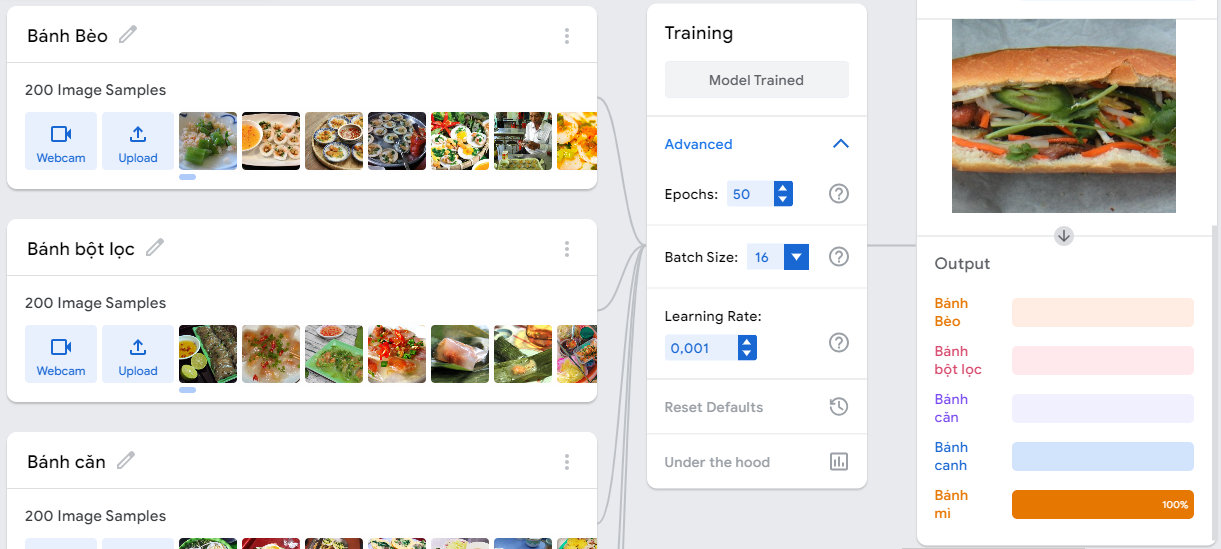


Hình 19: Huấn luyện dữ liệu món ăn trên Teachable Machine

Trong hình trên, chúng ta đã tải tất cả các dữ liệu hình ảnh liên quan đến 30 món ăn lên trên Teachable Machine và phân chúng theo tên của từng món ăn. Ví dụ: Bánh bèo, bánh bột lọc, bánh mì… Sau khi thực hiện tải đầy đủ dữ liệu về các món ăn cần huấn luyện, ta sẽ thực hiện tinh chỉnh tham số cho mô hình để có thể huấn luyện ra một mô hình nhận dạng món ăn với độ chính xác cao nhất.

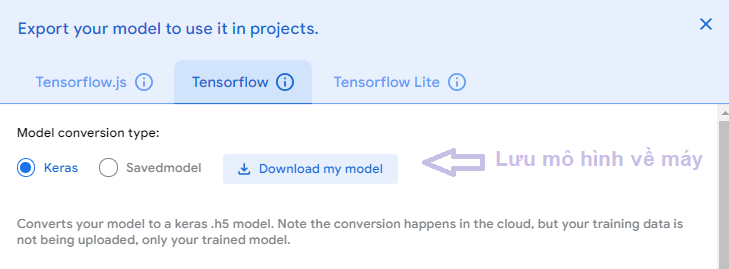
Khi làm việc với công cụ Teachable Machine, em đã thực nghiệm nhiều tham số cho mô hình thông qua một số lần train trước đó. Qua đó các tham số để phục vụ huấn luyện mô hình sẽ là: Epochs = 50, Batch size = 16, Learning Rate = 0.001. Với các tham số trên khi mô hình được huấn luyện thành công nó sẽ có độ chính xác sấp sỉ bằng 80%.

Sau khi huấn luyện mô hình thành công, chúng ta cũng có thể thực hiện dự đoán trực tiếp với mô hình vừa huấn luyện bằng cách gửi ảnh cần dự đoán lên để mô hình huấn luyện hoặc mở webcam để thực hiện dự đoán kết quả theo thời gian thực.



Hình 20: Dự đoán kết quả món ăn sau khi huấn luyện với Teachable Machine

Trong hình trên, chúng ta thực hiện việc dự đoán mô hình thông qua cách chọn và tải một hình ảnh Bánh Mì lên mô hình vừa huấn luyện đề thực hiện đưa ra dự đoán, khi đó mô hình nhận dạng chính xác với tên món ăn là Bánh mì và độ chính xác dự đoán là 100%.



Hình 21: Lưu lại mô hình sau khi huấn luyện Teachable Machine

Cuối cùng, để sử dụng và triển khai mô hình lên các ứng dụng khác, chúng ta cần tải về mô hình có định dạng .h5 để có thể sử dụng mô hình huấn luyện này trong thư viện Keras.

# CHƯƠNG III. XÂY DỰNG ỨNG DỤNG NHẬN DẠNG

## 3.1 Phát biểu bài toán

Xây dựng một ứng dụng di động nhận diện 30 món ăn Việt Nam là một bài toán thực tế và có tính ứng dụng cao trong lĩnh vực ẩm thực. Với sự phát triển của công nghệ hiện nay, chúng ta có thể tận dụng trí tuệ nhân tạo để tạo ra một ứng dụng giúp người dùng có thể nhận diện chính xác các món ăn Việt Nam thông qua ảnh chụp hoặc camera trực tiếp.

Bài toán này đòi hỏi chúng ta phải xây dựng một hệ thống phân loại ảnh hiệu quả và chính xác, đồng thời tập hợp dữ liệu lớn về các món ăn truyền thống của Việt Nam. Ứng dụng này cũng cần được thiết kế để có thể đáp ứng được nhu cầu của người dùng với các tính năng như nhận dạng tên món ăn, cung cấp thông tin chi tiết về từng món ăn và cách chế biến của từng món ăn được nhận dạng.

Một trong những thách thức khi xây dựng ứng dụng này là sự đa dạng của các loại món ăn trong ẩm thực Việt Nam, nhất là đối với những người mới tiếp cận. Có rất nhiều loại món ăn với các cách chế biến và hương vị khác nhau nằm trong nhóm 30 món ăn cần nhận diện. Việc tách biệt được chúng sẽ đòi hỏi xử lý ảnh tốt, phân tích rất chính xác sự khác biệt về hình dạng, màu sắc, thành phần và cách chế biến của từng món ăn.

Để hoàn thành bài toán này một cách hiệu quả, chúng ta cần sử dụng các thuật toán học máy và học sâu tiên tiến để xử lý ảnh và phân loại món ăn. Chúng ta cần phải xây dựng được một mô hình học máy với độ đánh giá cao nhằm mục đích đưa ra các dự đoán chính xác khi có yêu cầu nhận dạng món ăn từ người sử dụng. Không chỉ có thế, ứng dụng cần được thiết kế để có thể đáp ứng được nhu cầu của người dùng.

Nhờ vào bài toán này, người dùng sẽ có cơ hội trải nghiệm các món ăn Việt Nam truyền thống và đa dạng hơn. Ngoài ra, ứng dụng cũng có thể là một công cụ hữu ích cho những người đang học về ẩm thực Việt Nam và muốn tìm hiểu thêm về các món ăn truyền thống của đất nước chúng ta.

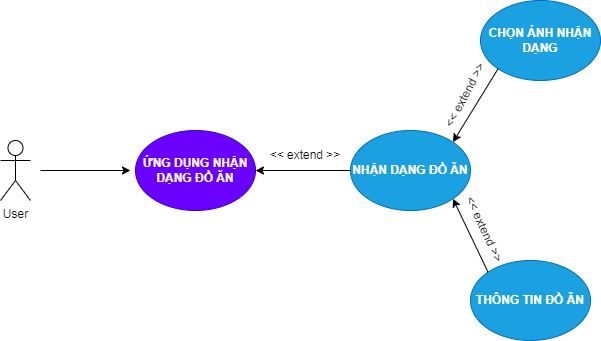
## 3.2 Phân tích thiết kế

### 3.2.1 Xác định Actor

Hệ thống chỉ có Actor duy nhất là người dùng ứng dụng (user):

* Người dùng có thể sử dụng thuộc phạm vi trong nước và các người dùng quốc tế không giới hạn độ tuổi, giới tính…
* Người dùng là những người có nhu cầu tìm hiểu về ẩm thực Việt Nam hoặc có nhu cầu nhận biết và phân loại các món ăn ở Việt Nam.
* Người dùng là thành phần sử dụng chính của ứng dụng, người dùng được phép chọn ảnh từ thiết bị di động của mình và được phép thực hiện chức năng dự đoán món ăn cho ảnh.

### 3.2.2 Biểu đồ Use case

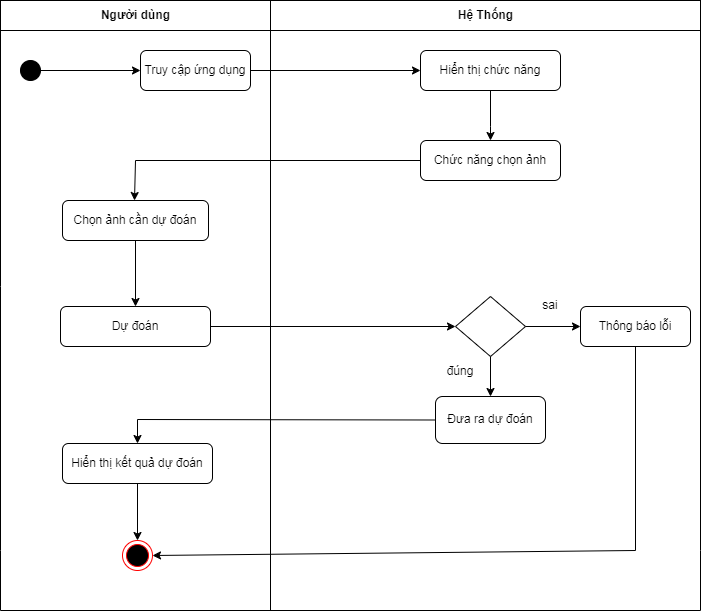


Hình 22: Biểu đồ User case tổng quát nhận dạng đồ ăn

**Mô tả:**

|  |  |
| --- | --- |
| **Use-case** | **Nội dung** |
| *Tên use case* | Nhận dạng đồ ăn |
| *Mô tả* | - Cho phép người dùng có thể truy cập ứng dụng và thực hiện chọn ảnh và nhận dạng món ăn Việt Nam  - Người dùng có thể xem thông tin kết quả dự đoán món ăn |
| *Actor* | Người dùng |
| *Điều kiện kích hoạt* | Khách hàng chọn vào chức năng chọn ảnh để dự đoán |
| *Tiền điều kiện* | Khách hàng đã truy cập vào ứng dụng |
| *Hậu điều kiện* | Đưa ra thông tin của món ăn được dự đoán |
| *Luồng sự kiện* | 1. Người dùng truy cập vào ứng dụng  2. Ứng dụng hiển thị chức năng chọn ảnh  3. Người dùng chọn ảnh và gửi ảnh dự đoán  4. Ứng dụng trả về kết quả thông tin món ăn  5. Kết thúc use case |

### 3.2.3 Biểu đồ hoạt động



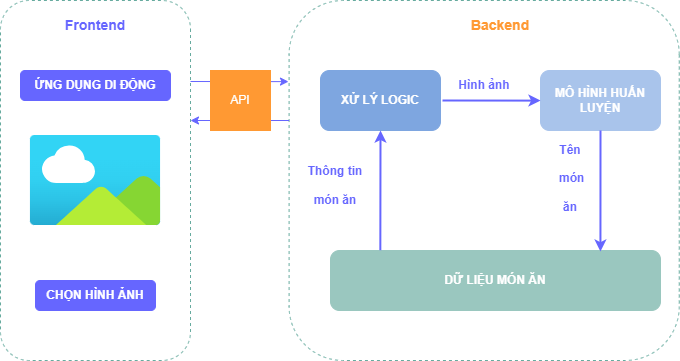
Hình 23: Biều đồ tuần tự chức năng dự đoán món ăn

**Mô tả các bước:**

|  |  |
| --- | --- |
| **Các bước** | **Mô tả** |
| Bước 1 | Người dùng truy cập ứng dụng |
| Bước 2 | Hệ thống hiển thị giao diện chức năng |
| Bước 3 | Người dùng chọn ảnh trong thư viện cần dự đo |
| Bước 4 | Hệ thống kiểm tra ảnh của người dùng |
| Bước 4.1 | Nếu ảnh không đúng định dạng hoặc ảnh bị thiếu thì hệ thống thông báo lỗi |
| Bước 4.2 | Nếu ảnh hợp lệ thì hệ thống thực hiện đưa ảnh vào dự đoán |
| Bước 5 | Hệ thống trả về kết quả về thông tin dự đoán hoặc thông báo lỗi |
| Bước 6 | Kết thúc |

## 3.3 Xây dựng hệ thống

Hệ thống nhận diện món ăn sẽ bao gồm 2 thành phần chính đó là: Frontend và Backend tương ứng với 2 phần đó là giao diện người dùng và phần xử lý nhận dạng.



Hình 24: Hệ thống nhận dạng món ăn

Phía giao diện người dùng còn được hiểu là Frontend, ứng dụng sẽ cho phép người dùng truy cập ứng dụng và chọn ảnh món ăn cần dự đoán và đưa thực hiện gửi ảnh lên phía Backend sau đó nhận thông tin kết quả về mô tả món ăn, cách nấu món ăn, tên món ăn, ảnh món ăn... hiển thị trực tiếp cho người dùng sử dụng.

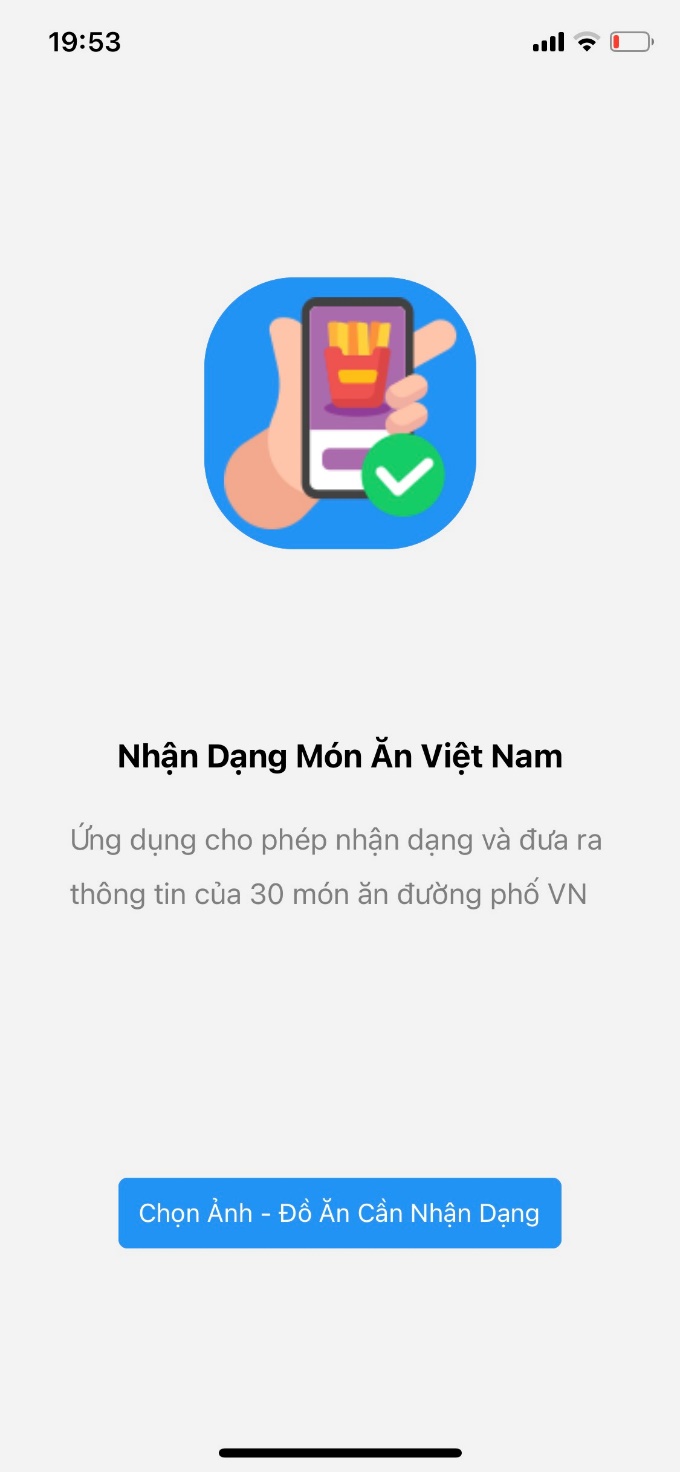
Phía xử lý và nhận dạng còn được hiểu là Backend sẽ tiếp nhận hình ảnh của người dùng gửi từ phía Fontend lên và đưa ảnh vào mô hình để dự đoán ra tên ảnh và tiếp tục đối sánh tên ảnh trong cơ sở dữ liệu để trả về thông tin chi tiết của món ăn được dự đoán.



Hình 25: Kết quả dự đoán trả về dưới dạng Json

Để truyền tải và tiếp nhận thông tin dữ liệu giữa 2 ứng dụng Frontend và Backend ta sẽ sử dụng thông qua API. Các thông tin gửi lên Backend phải được gửi đúng với địa chỉ Endpoint và các thông tin trả về sẽ là các chuỗi json bao gồm nhiều thuộc tính dữ liệu cho phía Frontend tiếp nhận và xử lý chuỗi json đó hiển thị ra cho người dùng.

## 3.4 Cài đặt hệ thống



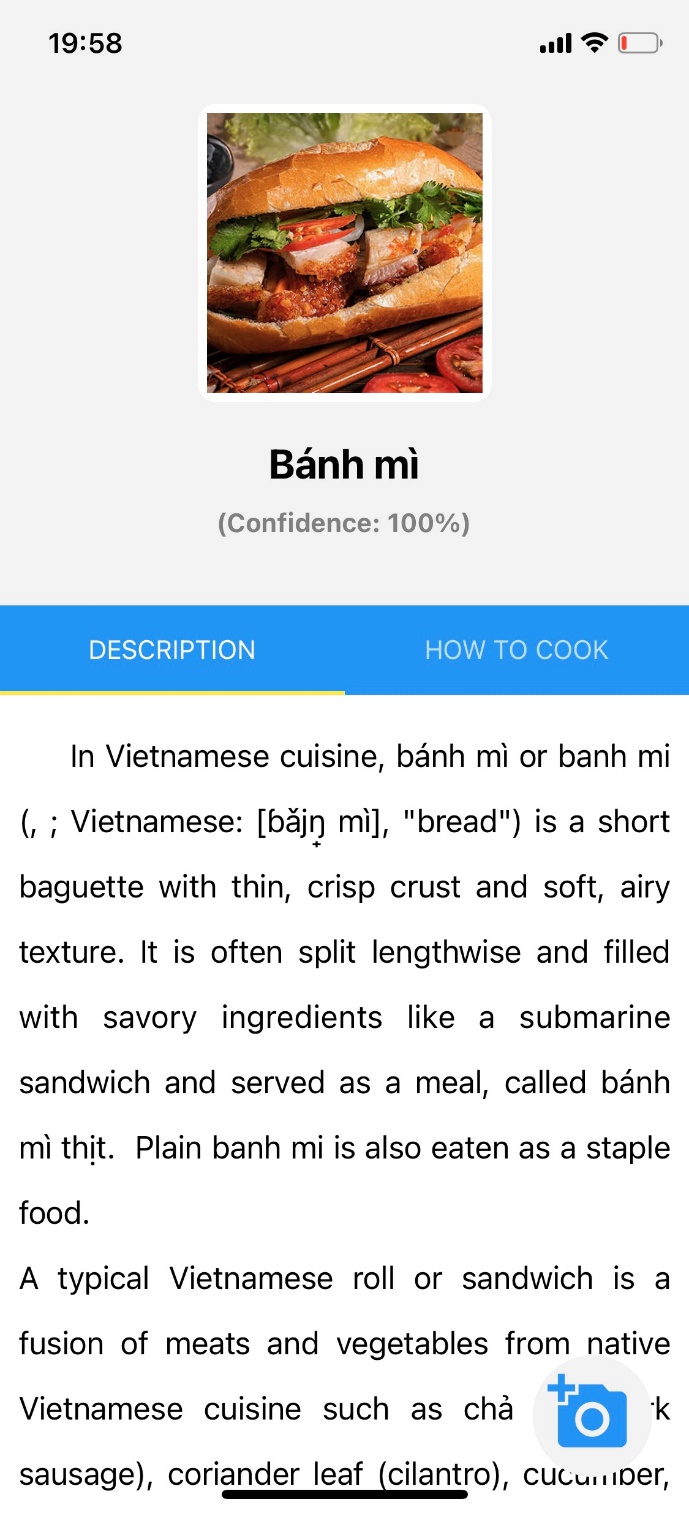
Hình 26: Màn hình chính của ứng dụng

Khi người dùng truy cập vào ứng dụng lần đầu tiên, ứng dụng sẽ chuyển người dùng tới màn hình chính.Màn hình chính bao gồm các thông tin miêu tả về ứng dụng và một nút chọn ảnh cho phép người dùng có thể truy cập thư viện ảnh trong thiết bị di động.



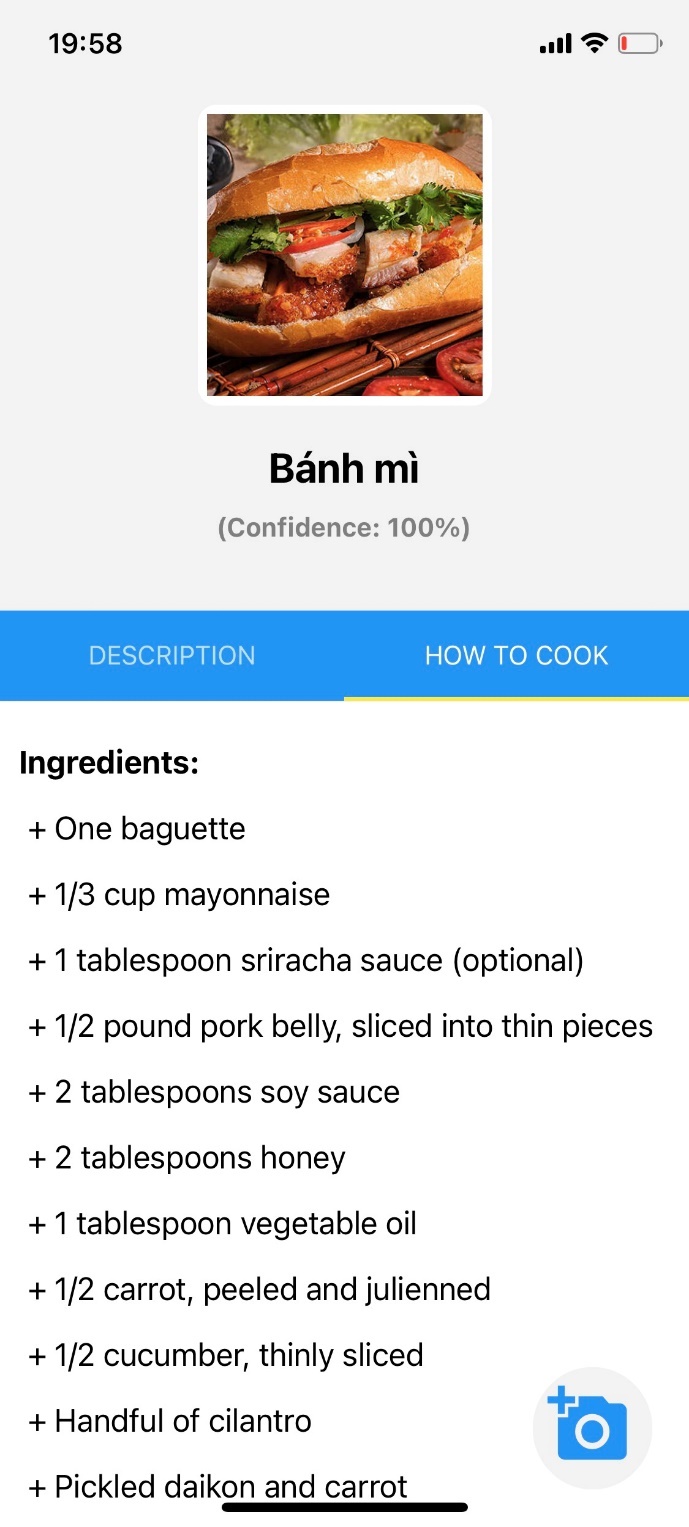
Hình 27: Màn hình liệt kê hình ảnh có trong thư viện di động

Khi người dùng thực hiện nhấn vào nút chọn ảnh ở màn hình chính giao diện màn hình liệt kê sẽ được hiển thị. Tại màn hình này, sẽ cung cấp cho ta tất cả các hình ảnh có trong thư viện mà ta cần thực hiện cho quá trình tra cứu và dự đoán các thông tin của món ăn.



Hình 28: Màn hình hiển thị thông tin kết quả dự đoán

Khi thực hiện chọn hỉnh ảnh từ thư viện ảnh trong máy, hệ thống sẽ tự động gửi hình ảnh đó lên phía Server để thực hiện dự đoán. Ứng dụng lúc này sẽ chuyển đến màn hình hiển thị thông tin kết quả dự đoán. Tại đây, mặc định các thông tin về mô tả món ăn, tên món ăn, hình ảnh, độ chính xác dự đoán… sẽ được hiển thị



Hình 29: Màn hình hiển thị công thức nấu đồ ăn đã dự đoán

Mặc định khi kết quả dự đoán được trả về sẽ hiển thị tab thông tin mô tả về món ăn đó. Ứng dụng sẽ chuyển đến màn hình xem công thức nấu đồ ăn đã được dự đoán khi người dùng thực hiện chuyển từ tab "DESCRIPTION" sang tab "HOW TO COOK"

## 3.5 Đánh giá thực nghiệm

Các kết quả thử nghiệm thực tế cho thấy ứng dụng có khả năng nhận dạng đến 90% các món ăn thường gặp và đưa ra kết quả trong vòng 3 giây với độ chính xác trung bình khoảng > 80%. Tuy nhiên, mô hình vẫn chưa phân biệt được rõ ràng các món ăn có điểm tương đồng nhau như: Phở - bún, bánh cuốn – bánh bột lọc…Ngoài ra, mô hình vẫn hạn chế trong việc dự đoán cho các món ăn khác vì chỉ có thể nhận dạng và đưa ra kết quả về thông tin của 30 món ăn đã được huấn luyện.

Về phần ứng dụng, được thiết kế giao diện đơn giản, dễ sử dụng và tối ưu hóa tính ứng dụng của ứng dụng đã tạo ra một trải nghiệm tuyệt vời cho người dùng. Điểm đáng khen của ứng dụng này là khả năng cung cấp rất nhiều thông tin chi tiết về cách chế biến món ăn, giúp người dùng hiểu rõ hơn về nguyên liệu cần thiết và cách chế biến món ăn một cách dễ hiểu và rõ ràng.

Thế nhưng, để đảm bảo tính ứng dụng cao trong thực tế và tận dụng tối đa các ưu điểm của của Deep Learning, mô hình cần phải được cập nhật liên tục và cần được huấn luyện thêm nữa để tăng độ chính xác khi phải gặp các dữ liệu mới. Ứng dụng cần được cập nhật thường xuyên và tiếp tục phát triển hơn nữa để tăng trải nghiệm người dùng.

Tổng thể, ứng dụng nhận dạng đồ ăn bằng trên di động sử dụng Deep Learning là một ứng dụng rất hữu ích và tiện lợi cho người dùng, đáp ứng được nhu cầu của họ tìm kiếm và khám phá các món ăn mới. Ứng dụng gây ấn tượng mạnh với tính ứng dụng và các tính năng chính của nó, đồng thời, cũng nhấn mạnh rằng cần phải được cập nhật thường xuyên để đảm bảo một trải nghiệm tốt hơn cho người dùng.

# CHƯƠNG IV. KẾT LUẬN

## 4.1 Những nội dung đạt được

* **Về mặt lý thuyết:**
  + Tìm hiểu được các kiến thưc lý thuyết liên quan đến AI và học máy, học sâu. Nắm bắt được cách hoạt động của các mô hình mạng và các cấu trúc mạng trong học sâu.
  + Tìm hiểu chi tiết về mạng MobileNet và MobileNetV2 và xây dựng được các mô hình huấn luyện dựa trên kỹ thuật Transfer Learning
  + Học hỏi thêm được các công cụ hữu ích phục vụ cho việc huấn luyện và đạo tạo các mô hình liên quan đến nhận dạng ảnh.
  + Hiểu biết thêm về các ngôn ngữ lập trình Python, Javascript khi phát triển phần mềm và ứng dụng di động thông qua chúng.
  + Nắm được nguyên lý, phân tích thiết kế một ứng dụng phần mềm hoàn chỉnh và các quy trình từng bước thực hiện xây dựng 1 ứng dụng phần mềm.
* **Về mặt ứng dụng:**
  + Đã xây dựng thành công được ứng dụng cho phép người dùng có thể nhận dạng ảnh và thông tin món ăn từ chính thiết bị di động của mình.
  + Ứng dụng khá thực tế khi trợ giúp cho những người cần tìm hiểu về ẩm thực Việt Nam cả trong nước và quốc tế.

## 4.2 Những hạn chế

Ngoài những kết quả đã đạt được trong quá trình thực hiện đề tài. Hệ thống còn một số hạn chế sau:

* Số lượng món ăn còn hạn chế chỉ có thể nhận dạng cho 30 món ăn đã huấn luyện trước
* Tốc độ dự đoán và đưa ra kết quả của mô hình vẫn chưa tối ưu nhất
* Việc dự đoán đôi khi vẫn có nhầm lẫn và độ chính xác dự đoán chưa là tuyệt đối
* Ứng dụng còn thiếu nhiều chức năng để tăng tính trải nghiệm người dùng
* Giao diện ứng dụng đôi khi sẽ bị lệch hoặc vỡ giao diện khi người dùng sử dụng các thiết bị di động quá lỗi thời.
* Ứng dụng hiện tại chỉ cung cấp các thông tin kết quả trả về bằng Tiếng Anh

## 4.3 Hướng phát triển

Để giải quyết những hạn chế trên về mô hình cũng như tối ưu hóa trải nghiệm người dùng, em đề xuất một số hướng phát triển cho ứng dụng nhận dạng món ăn ở Việt Nam như sau:

* Mở rộng cơ sở dữ liệu với số lượng món ăn đa dạng và phong phú.
* Nâng cao tính ứng dụng bằng cách cải tiến hệ thống nhận dạng hình ảnh.
* Tăng tốc việc nhận dạng bằng kỹ thuật đa luồng
* Cập nhật các giao diện với các kích thước phù hợp với tất cả các thiết bị
* Hỗ trợ đa ngôn ngữ có thể linh hoạt và truyền tải đầy đủ nội dung cho người dùng.

Tổng thể, ứng dụng nhận dạng món ăn ở Việt Nam là một bước phát triển đáng kể trong công nghệ ẩm thực Việt Nam và em sẽ tiếp tục được triển ứng dụng này sao cho ứng dụng được tối ưu hóa trải nghiệm người dùng và mở rộng khả năng nhận dạng các loại món ăn đa dạng và phong phú hơn trong tương lai.

# TÀI LIỆU THAM KHẢO

[1] Lược sử Deep Learning

Xem tại: <https://machinelearningcoban.com/2018/06/22/deeplearning/>

[2] Nhận dạng hình ảnh thực phẩm bằng phương pháp deep learning

Xem tại: <http://www.tailieumienphi.vn/doc/nhan-dang-hinh-anh-thuc-pham-bang-phuong-phap-deep-learning-ffeiuq.html>

[3] Deep learning approaches in food recognition

Xem tại: <https://arxiv.org/ftp/arxiv/papers/2004/2004.03357.pdf>

[4] Introduction to Convolutional Neural Networks (CNN)

Xem tại: <https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>

[5] Deep Learning with Python : Beginners Guide to Deep Learning

Xem tại: <https://www.edureka.co/blog/deep-learning-with-python/>

[6] MobileNetV2: Inverted Residuals and Linear Bottlenecks

Xem tại: <https://arxiv.org/pdf/1801.04381v4.pdf>

[7] Deploying Deep Learning Model using Flask API

Xem tại: <https://medium.com/@draj0718/deploying-deep-learning-model-using-flask-api-810047f090ac>

[8] Machine Learning for Food Recognition with Android Demo

Xem tại: <https://hackernoon.com/machine-learning-for-food-recognition-with-android-demo-b62cf2964f63>