



HCMUTE

**Báo Cáo Nhận Diện Trái Cây
Sử Dụng Phương Pháp CNN
Trí Tuệ Nhân Tạo**

GVHD: Nguyễn Trường Thịnh

Họ Và Tên : Võ Nguyễn Phú Thịnh

MSSV : 19146271

Khoa : Đào Tạo Chất Lượng Cao

Ngành : Cơ Điện Tử

Tp. Hồ Chí Minh, 6-2022

MỤC LỤC

TÓM TẮT ĐỀ TÀI	1
Chương 1: TỔNG QUAN VỀ ĐỀ TÀI	2
1.1 Tổng quan về lĩnh vực nghiên cứu, các công trình nghiên cứu trong và ngoài nước đã công bố	2
1.1.1 Tổng quan về lĩnh vực nghiên cứu	2
1.2 Mục đích đề tài	4
1.3 Nhiệm vụ của đề tài và giới hạn đề tài	5
1.3.1 Nhiệm vụ của đề tài	5
1.3.2 Giới hạn của đề tài	5
1.4 Phương pháp nghiên cứu	5
Chương 2: CƠ SỞ LÝ THUYẾT	7
2.1 Nghiên cứu về cơ sở lý thuyết chung và khái niệm	7
2.1.1 Khái niệm trí tuệ nhân tạo (Artificial Intelligence)	7
2.1.2 Khái niệm học máy (Machine Learning)	8
2.1.3 Khái niệm học sâu (Deep Learning)	9
2.1.4 Tổng quan về thư viện Tensorflow	9
2.2 Mạng nơ ron nhân tạo và thuật toán CNN	10
2.2.1 Khái niệm mạng nơ ron nhân tạo (Artificial Neural Networks)	10
2.2.2 Thuật toán CNN – Convolutional Neural Network	11
Chương 3: TRÍCH XUẤT DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH	13
3.1 Giới thiệu về Google Colaboratory	13
3.2 Trích xuất và xử lý dữ liệu	13
3.2.1 Đánh giá sơ bộ và tiền xử lý dữ liệu	13
3.2.2 Truy xuất dữ liệu từ drive và chia thành các tập train, test	14
3.3 Xây dựng mô hình	16
CHƯƠNG 4: HUẤN LUYỆN, ĐÁNH GIÁ MÔ HÌNH VÀ KẾT QUẢ THU ĐƯỢC	17
4.1 Huấn luyện mô hình	17

4.2 Đánh giá mô hình	17
4.3 Thử nghiệm kết quả thu được.....	19
4.4 Chạy thử mô hình real-time bằng java trên môi trường google colab.....	21
Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	22
5.1 Kết luận	22
5.2 Hướng phát triển.....	22
PHỤ LỤC	23
TÀI LIỆU THAM KHẢO	23

Danh mục hình ảnh

Hình 1. Robot Sophia mô phỏng hành vi của con người.....	3
Hình 2. Một số loại trái cây thường gặp.....	3
Hình 3. Phân loại trái cây theo phương pháp thủ công	4
Hình 4. 70 classes từ tập dữ liệu.....	14
Hình 5. Đồ thị biểu diễn độ chính xác của mô hình	18

TÓM TẮT ĐỀ TÀI

Hiện nay, ở nước ta nói riêng và ở các nước đang phát triển có nền nông nghiệp là một trong các ngành sản xuất chủ yếu, quá trình thu hoạch, phân loại và đánh giá chất lượng các loại sản phẩm nông nghiệp, đặc biệt là các loại hoa quả, chủ yếu còn phải thực hiện bằng các phương pháp thủ công. Đây là công việc không quá khó, nhưng tiêu tốn nhiều thời gian, công sức của con người và là rào cản đối với mở rộng phát triển quy mô sản xuất nông nghiệp. Trong thời gian gần đây, nhờ có sự phát triển mạnh mẽ về khả năng tính toán của các thể hệ máy tính hiện đại cũng như sự bùng nổ về dữ liệu thông qua mạng lưới Internet trải rộng, ta đã chứng kiến nhiều sự đột phá trong lĩnh vực Học máy, đặc biệt là trong lĩnh vực Thị giác máy tính. Sự quay lại và phát triển vượt bậc của các phương pháp Học sâu đã giúp Thị giác máy tính đạt được những thành tựu đáng kể trong lĩnh vực Nhận dạng ảnh, trong đó có bài toán nhận dạng hoa quả.

Đề tài nghiên cứu “Nhận dạng và phân loại hoa quả trong ảnh màu” 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 hoa quả tự động, đặc biệt là đối với các loại hoa quả phổ biến tại nước ta.

Bộ cơ sở dữ liệu ảnh là một trong các thành phần quan trọng hàng đầu trong các phương pháp Học máy nói chung, được sử dụng để phục vụ cho quá trình tính toán tham số và huấn luyện, tinh chỉnh các mô hình. Thông thường, bộ dữ liệu càng lớn và càng được chọn lọc tỉ mỉ cẩn thận thì độ chính xác của mô hình càng được cải thiện, nhưng trong phạm vi đề tài văn này sẽ được hạn chế cả về số lượng loại hoa quả sẽ nhận dạng cũng như số lượng ảnh chụp cho mỗi loại hoa quả đó. Cụ thể: - Số lượng hoa quả sẽ nhận dạng: 70 loại hoa quả phổ biến ở nước ta như nho, táo, chuối, thanh long... - Số lượng ảnh gốc cho mỗi loại quả: 300-500 ảnh Sau khi đã thu thập đủ số lượng ảnh gốc cho các loại hoa quả, ta sẽ sử dụng các thuật toán chỉnh sửa ảnh, như làm nghiêng ảnh, chèn thêm nhiễu hoặc ghép ảnh với nền khác, để tạo thêm ảnh mới nhằm tăng cường kích thước cơ sở dữ liệu (CSDL).

Chương 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 Tổng quan về lĩnh vực nghiên cứu, các công trình nghiên cứu trong và ngoài nước

1.1.1 Tổng quan về lĩnh vực nghiên cứu

Trí tuệ nhân tạo hay trí thông minh nhân tạo (Artificial intelligence – viết tắt là AI) là một ngành thuộc lĩnh vực khoa học máy tính (Computer science). Là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người. Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc ứng dụng các hệ thống học máy (machine learning) để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính.

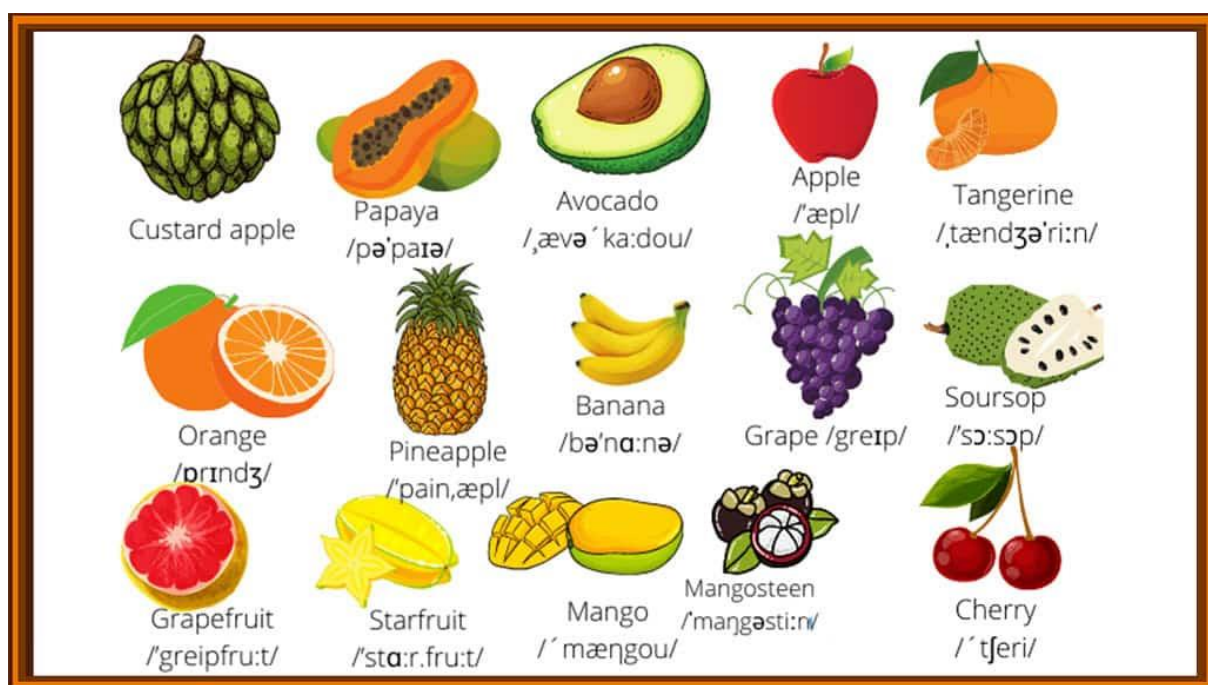
Những năm gần đây, trí tuệ nhân tạo (artificial intelligence, AI) dần nổi lên như một minh chứng cho cuộc cách mạng công nghiệp lần thứ tư, sau động cơ hơi nước, năng lượng điện và công nghệ thông tin. Trí tuệ nhân tạo đã và đang trở thành nhân tố cốt lõi trong các hệ thống công nghệ cao. Thậm chí, nó đã len lỏi vào hầu hết các lĩnh vực của đời sống mà có thể chúng ta không nhận ra. Hệ thống nhận diện khuôn mặt người trong ảnh của Facebook, hệ thống xử lý ảnh biển số xe trong bãi đỗ xe, trợ lý ảo chatbot, hệ thống trí tuệ nhân tạo được tạo ra để phân tích dữ liệu, ghi chú và báo cáo từ những gì thu thập từ bệnh nhân, nghiên cứu bên ngoài và chuyên môn lâm sàng giúp chọn lựa cách điều trị phù hợp, gần đây nhất là robot có tuệ nhân tạo Sophia mô phỏng các hành vi của con người... chỉ là một vài ứng dụng nổi bật trong vô vàn những ứng dụng của trí tuệ nhân tạo.

Những năm gần đây, sự phát triển của các hệ thống tính toán cùng lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn đã giúp trí tuệ nhân tạo tiến thêm một bước dài. Một lĩnh vực mới được ra đời được gọi là học sâu (deep learning, DL). Deep learning đã giúp máy tính thực thi những việc vào mười năm trước tưởng chừng là không thể: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết, giao tiếp với con người, chuyển đổi ngôn ngữ, hay thậm chí cả sáng tác văn thơ và âm nhạc.



Hình 1. Robot Sophia mô phỏng hành vi của con người

Nhận diện trái cây là việc làm cần thiết nhằm đảm bảo phân loại các loại trái cây khác nhau dựa trên hình ảnh bên ngoài của nó. Nhằm mục đích có thể dễ dàng vận chuyển, phân loại, qua đó giảm lượng thời gian cần xử lý. Bài toán nhận diện hoa quả kế thừa các khó khăn của các bài toán gốc và kèm theo là các khó khăn riêng của nó, như: Số lượng khổng lồ về chủng loại hoa quả theo mùa, theo vùng miền, địa hình, . . . Với vô số các loại hoa quả có hình dáng, sắc màu, kết cấu khá giống nhau. Dải biến thiên màu sắc theo chu kỳ phát triển của quả từ lúc còn xanh đến lúc chín, hay sự đa dạng về hình dáng của cùng một loại quả do ảnh hưởng của thời tiết, điều kiện sinh trưởng thổ nhưỡng của đất,



Hình 2. Một số loại trái cây phổ biến

Phương pháp phổ biến nhất hiện nay là phân loại trái cây thủ công dựa vào vỏ ngoài và kích thước của quả để phân loại trái cây chín hay chưa. Việc phân loại như vậy rất tốn kém thời gian cũng như nguồn nhân lực.



Hình 3. Phân loại trái cây theo phương pháp thủ công

Trước kia, người ta thường sử dụng phương pháp làm phân loại truyền thống bằng việc các công nhân sẽ phân loại trên dây chuyền. Nhược điểm của phương pháp này là khả năng phân loại không cao do sức người hạn chế, tốn nhiều thời gian, và nếu với số lượng trái cây lớn thì phương này phải sử dụng nhiều nhân công lao động, tốn nhiều chi phí về con người và khó có thể đạt năng suất cao. Từ đó sự ứng dụng rộng rãi của trí tuệ nhân tạo vào việc phân loại trái cây ngày càng phổ biến với nhiều nghiên cứu ứng dụng đã đạt được một số thành tựu nhất định trong lĩnh vực này.

1.2 Mục đích đề tài

Hiện tại, ứng dụng của AI vào việc nhận diện trái cây đã dần phổ biến nhưng nó vẫn chưa được phổ biến đến nhiều nơi trên thế giới. Có thể do việc xây dựng cũng như thi công một hệ thống đòi hỏi sự phức tạp và chi phí xây dựng đắt đỏ. Việc áp dụng trí thông minh nhân tạo vào đời sống sẽ giúp giảm thiểu thời gian cũng như công sức và tiền bạc.

Trong đề tài này sẽ tiếp bước các nghiên cứu về lĩnh vực trí tuệ nhân tạo ứng dụng trong việc nhận diện trái cây như đã nêu ở trên trong ngành khoa học máy tính để hoàn thiện và cải tiến, để người dùng có thể tiếp cận và sử dụng như một phần trong cuộc sống thường ngày. Vì vậy mục đích nghiên cứu là: nghiên cứu,

ứng dụng và đưa ra những phương pháp của trí tuệ nhân tạo trong việc nhận diện trái cây sử dụng mạng nơ ron nhân tạo tích chập (CNN).

1.3 Nhiệm vụ của đề tài và giới hạn đề tài

1.3.1 Nhiệm vụ của đề tài

Nghiên cứu và ứng dụng các lý thuyết về trí tuệ nhân tạo (Artificial Intelligence), học máy (Machine Learning), học sâu (Deep Learning).

Xây dựng mô hình Mạng nơ-ron tích chập (Convolutional Neural Network) ứng dụng cho phân loại trái cây, xử lý tập dữ liệu bao gồm 47745 hình ảnh về 70 loại trái cây phổ biến trên thế giới.

Tối ưu mô hình đã xây dựng ở trên đưa ra kết quả dự đoán với độ chính xác không dưới 80%, có thể phân loại trái cây thời gian thực (realtime) bằng camera trên máy tính.

Sử dụng google colaboratory để xử lý dữ liệu, xây dựng mô hình mạng nơ-ron tích chập (Convolutional Neural Network), đánh giá mô hình đưa ra độ chính xác và xử lý dữ liệu thời gian thực.

1.3.2 Giới hạn của đề tài

Đề tài chỉ dừng lại ở việc nghiên cứu và ứng dụng trí tuệ nhân tạo (Artificial Intelligence) nên vẫn còn rất nhiều thiếu sót.

Nhận diện 70 loại trái cây trên thế giới, chưa có khả năng nhận diện chính xác hoàn toàn các loại trái cây, chưa nhận diện các loại trái cây khác bên ngoài cơ sở dữ liệu.

Độ chính xác của mô hình không quá cao do chưa có được tập dữ liệu tối ưu và toàn diện (sử dụng tập dữ liệu có sẵn trên kaggle)

www.kaggle.com/datasets/moltean/fruits

Nhận diện trái cây thời gian thực bằng camera máy tính và google colaboratory.

1.4 Phương pháp nghiên cứu

Từ các cơ sở lý thuyết nền tảng đã học từ bộ môn Trí tuệ nhân tạo (Artificial Intelligence) và Học máy (Machine Learning), áp dụng vào việc xây dựng mô hình và xử lý dữ liệu đầu vào.

Nghiên cứu các tài liệu khoa học liên quan đến trí tuệ nhân tạo, các nghiên cứu đã thực hiện trước đây, các đề tài và nghiên cứu khoa học trong và ngoài nước, tham khảo các tài liệu trên Internet và các nguồn khác nhau.

Chương 2: CƠ SỞ LÝ THUYẾT

2.1 Nghiên cứu về cơ sở lý thuyết chung và khái niệm

2.1.1 Khái niệm trí tuệ nhân tạo (Artificial Intelligence)

Đầu tiên chúng cần phải hiểu được khái niệm AI : **Trí Tuệ Nhân Tạo** - **AI (Artificial Intelligence)** hoặc trí thông minh nhân tạo là công nghệ mô phỏng các quá trình suy nghĩ và học tập của con người cho máy móc, đặc biệt là hệ thống máy tính. Trí tuệ nhân tạo này do con người lập trình ra với mục đích tự động hóa các hành vi thông minh như con người, từ đó cắt giảm bớt nhân công là con người và có tính chuẩn xác cao hơn. Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc ứng dụng các hệ thống học máy (machine learning) để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính.

Sự khác biệt của trí tuệ nhân tạo so với các lập trình logic trước kia chính là khả năng suy nghĩ độc lập của chúng, thay vì việc mọi thứ được lập trình sẵn và cỗ máy đó sẽ thực hiện các thao tác theo logic được con người đặt ra, **Ai - Trí Tuệ Nhân Tạo** sẽ tự xem xét tình huống và đưa ra phương án tối ưu nhất, qua đó tiết kiệm chi phí cũng như vận hành cho công việc hiệu quả hơn. Ngoài ra khả năng tự tính toán đó sẽ khiến Ai đưa ra những ý kiến mới, giúp con người thêm nhiều ý tưởng hơn trong phát triển. Tuy rằng trí thông minh nhân tạo có nghĩa rộng như là trí thông minh trong các tác phẩm khoa học viễn tưởng, nó là một trong những ngành trọng yếu của tin học. Trí thông minh nhân tạo liên quan đến cách cư xử, sự học hỏi và khả năng thích ứng thông minh của máy móc.

Công nghệ AI được chia làm 4 loại chính:

Loại 1: Công nghệ AI phản ứng.

Công nghệ AI phản ứng có khả năng phân tích những động thái khả thi nhất của chính mình và của đối thủ, từ đó, đưa ra được giải pháp tối ưu nhất.

Một ví dụ điển hình của công nghệ AI phản ứng là Deep Blue. Đây là một chương trình chơi cờ vua tự động, được tạo ra bởi IBM, với khả năng xác định

các nước cờ đồng thời dự đoán những bước đi tiếp theo của đối thủ. Thông qua đó, Deep Blue đưa ra những nước đi thích hợp nhất.

Loại 2: Công nghệ AI với bộ nhớ hạn chế

Đặc điểm của công nghệ AI với bộ nhớ hạn chế là khả năng sử dụng những kinh nghiệm trong quá khứ để đưa ra những quyết định trong tương lai. Công nghệ AI này thường kết hợp với cảm biến môi trường xung quanh nhằm mục đích dự đoán những trường hợp có thể xảy ra và đưa ra quyết định tốt nhất cho thiết bị.

Ví dụ như đối với xe không người lái, nhiều cảm biến được trang bị xung quanh xe và ở đầu xe để tính toán khoảng cách với các xe phía trước, công nghệ AI sẽ dự đoán khả năng xảy ra va chạm, từ đó điều chỉnh tốc độ xe phù hợp để giữ an toàn cho xe.

Loại 3: Lý thuyết trí tuệ nhân tạo

Công nghệ AI này có thể học hỏi cũng như tự suy nghĩ, sau đó áp dụng những gì học được để thực hiện một việc cụ thể. Hiện nay, công nghệ AI này vẫn chưa trở thành một phương án khả thi.

Loại 4: Tự nhận thức

Công nghệ AI này có khả năng tự nhận thức về bản thân, có ý thức và hành xử như con người. Thậm chí, chúng còn có thể bộc lộ cảm xúc cũng như hiểu được những cảm xúc của con người. Đây được xem là bước phát triển cao nhất của công nghệ AI và đến thời điểm hiện tại, công nghệ này vẫn chưa khả thi.

2.1.2 Khái niệm học máy (Machine Learning)

Học máy hay máy học trong tiếng Anh là Machine learning, viết tắt: ML. Học máy (ML) là một công nghệ phát triển từ lĩnh vực trí tuệ nhân tạo. Các thuật toán ML là các chương trình máy tính có khả năng học hỏi về cách hoàn thành các nhiệm vụ và cách cải thiện hiệu suất theo thời gian. ML vẫn đòi hỏi sự đánh giá của con người trong việc tìm hiểu dữ liệu cơ sở và lựa chọn các kỹ thuật phù hợp để phân tích dữ liệu. Đồng thời, trước khi sử dụng, dữ liệu phải sạch, không có sai lệch và không có dữ liệu giả.

Có 2 loại học máy (Machine Learning):

+Học có giám sát

Trong học có giám sát, máy tính học cách mô hình hóa các mối quan hệ dựa trên dữ liệu được gán nhãn (labeled data). Sau khi tìm hiểu cách tốt nhất để mô hình hóa các mối quan hệ cho dữ liệu được gán nhãn, các thuật toán được huấn luyện được sử dụng cho các bộ dữ liệu mới. Ứng dụng của kỹ thuật học có giám sát: Xác định tín hiệu hay biến số tốt nhất để dự báo lợi nhuận trong tương lai của cổ phiếu hoặc dự đoán xu hướng thị trường chứng khoán.

+Học không giám sát

Trong học không giám sát, máy tính không được cung cấp dữ liệu được dán nhãn mà thay vào đó chỉ được cung cấp dữ liệu mà thuật toán tìm cách mô tả dữ liệu và cấu trúc của chúng. Ứng dụng của học không giám sát: Phân loại các công ty thành các nhóm công ty tương đồng dựa trên đặc điểm của chúng thay vì sử dụng tiêu chuẩn của các nhóm ngành hoặc các quốc gia.

2.1.3 Khái niệm học sâu (Deep Learning)

Deep Learning là tập hợp con của Machine Learning và nó có tác dụng hỗ trợ cho máy tính tự huấn luyện chính nó để có thể thực hiện mọi tác vụ tương tự như con người. Điều này chính là giúp máy tính bắt chước con người cách học hỏi và suy nghĩ. Trí tuệ nhân tạo có thể được hiểu đơn giản là được cấu thành từ các lớp xếp chồng lên nhau, trong đó mạng thần kinh nhân tạo nằm ở dưới đáy, Machine learning nằm ở tầng tiếp theo và Deep learning nằm ở tầng trên cùng.

2.1.4 Tổng quan về thư viện Tensorflow

Với sự bùng nổ của lĩnh vực Trí Tuệ Nhân Tạo – A.I. trong thập kỷ vừa qua, machine learning và deep learning rõ ràng cũng phát triển theo cùng. Và ở thời điểm hiện tại, TensorFlow chính là thư viện mã nguồn mở cho machine learning nổi tiếng nhất thế giới, được phát triển bởi các nhà nghiên cứu từ Google. Việc hỗ trợ mạnh mẽ các phép toán học để tính toán trong machine learning và deep learning đã giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn nhiều. Các hàm được dựng sẵn trong thư viện cho từng bài toán cho phép TensorFlow xây dựng được nhiều neural network. Nó còn cho phép bạn tính toán song song trên nhiều máy tính khác nhau, thậm chí trên nhiều CPU, GPU trong cùng 1 máy hay tạo ra các dataflow graph – đồ thị luồng dữ liệu để dựng nên các model. Kiến trúc TensorFlow hoạt động được chia thành 3 phần:

- Tiền xử lý dữ liệu

- Dựng model
- Train và ước tính model

Tensor đại diện cho các loại dữ liệu được đưa vào trong Tensorflow. Mỗi thuộc tính trong tensor sẽ có những đặc điểm và tính năng khác nhau với những thuộc tính cơ bản sau đây:

- Rank: Trong các cấu trúc dữ liệu, thuộc tính bậc được hiểu một cách đơn giản chính là sự phân cấp bậc và là căn cứ cho việc phân loại các tensor. Mỗi tensor khi được phân bậc sẽ có tên gọi khác nhau, cụ thể bậc 0 là Scalar, bậc 1 là Vector bậc 2 là Matrix, các bậc cao hơn nữa sẽ được gọi là n-tensor.
- Shape: Đây là thuộc tính chiều của tensor các cấu trúc dữ liệu.
- Type: Kiểu dữ liệu của các element và là thuộc tính type duy nhất có trong tensor. Một tensor chỉ có một loại type duy nhất cho toàn bộ các element có trong tensor. Vì vậy mà cấu trúc dữ liệu có tính thống nhất.

2.2 Mạng nơ ron nhân tạo và thuật toán CNN

2.2.1 Khái niệm mạng nơ ron nhân tạo (Artificial Neural Networks)

Mạng nơ ron nhân tạo (Artificial Neural Networks) là một mô hình xử lý thông tin, cấu thành từ các lớp nơ ron, được ra đời trên cơ sở mô phỏng hoạt động não bộ của sinh vật. Mạng nơ ron nhân tạo gắn kết nhiều nơ ron theo một mô hình nhất định, được trải qua huấn luyện để rút ra được kinh nghiệm, và sử dụng các kinh nghiệm đã có để xử lý các thông tin mới. Mạng nơ ron nhân tạo thường áp dụng vào giải các bài toán nhận dạng mẫu, hoặc dự đoán.

Mạng nơ ron nhân tạo, Artificial Neural Network (ANN) là một mô hình xử lý thông tin phỏng theo cách thức xử lý thông tin của các hệ nơ ron sinh học. Nó được tạo nên từ một số lượng lớn các phần tử (nơ ron) kết nối với nhau thông qua các liên kết (trọng số liên kết) làm việc như một thể thống nhất để giải quyết một vấn đề cụ thể nào đó. Một mạng nơ ron nhân tạo được cấu hình cho một ứng dụng cụ thể (nhận dạng mẫu, phân loại dữ liệu,...) thông qua một quá trình học từ tập các mẫu huấn luyện. Về bản chất học chính là quá trình hiệu chỉnh trọng số liên kết giữa các nơ ron.

2.2.2 Thuật toán CNN – Convolutional Neural Network

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến. Nó giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay.

Các convolutional layer có các parameter(kernel) đã được học để tự điều chỉnh lấy ra những thông tin chính xác nhất mà không cần chọn các feature.

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau và sử dụng các hàm nonlinear activation như ReLU và tanh để kích hoạt các trọng số trong các node. Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo.

Mỗi một lớp sau khi thông qua các hàm kích hoạt sẽ tạo ra các thông tin trừu tượng hơn cho các lớp tiếp theo. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi neural đầu vào (input node) cho mỗi neural đầu ra trong các lớp tiếp theo.

Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer) hay mạng toàn vẹn (affine layer). Còn trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution.

Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Như vậy mỗi neuron ở lớp kế tiếp sinh ra từ kết quả của filter áp đặt lên một vùng ảnh cục bộ của neuron trước đó.

Mỗi một lớp được sử dụng các filter khác nhau thông thường có hàng trăm hàng nghìn filter như vậy và kết hợp kết quả của chúng lại. Ngoài ra có một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu).

Trong quá trình huấn luyện mạng (training) CNN tự động học các giá trị qua các lớp filter dựa vào cách thức mà bạn thực hiện. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw pixel > edges > shapes > facial > high-level features. Layer cuối cùng được dùng để phân lớp ảnh.

Cách thức hoạt động của thuật toán Deep Learning diễn ra như sau: Các dòng thông tin sẽ được trải qua nhiều lớp cho đến lớp sau cùng. Lấy quy trình học của con người làm ví dụ cụ thể. Qua các lớp đầu tiên sẽ tập trung vào việc học các khái niệm cụ thể hơn trong khi các lớp sâu hơn sẽ sử dụng thông tin đã học để nghiên cứu và phân tích sâu hơn trong các khái niệm trừu tượng. Quy trình xây dựng biểu diễn dữ liệu này được gọi là trích xuất tính năng. Để có thể dễ hình dung về Deep Learning chúng ta sẽ tìm hiểu cách nó hoạt động thông qua một số ví dụ sau. Hãy bắt đầu với một ví dụ đơn giản về Deep Learning ở cấp độ khái niệm. Hãy cùng suy nghĩ làm thế nào để chúng ta có thể nhận biết được một hình nào đó là hình vuông.

Có thể đầu tiên bạn sẽ kiểm tra xem hình đó có 4 cạnh hay không, nếu nó đúng chúng ta sẽ kiểm tra tiếp 4 cạnh này có được kết nối với nhau thành 1 hình tứ giác hay không, nếu đúng chúng ta sẽ kiểm tra tiếp 4 cạnh này có vuông góc với nhau không và chúng có kích thước bằng nhau không. Nếu tất cả đều đúng thì kết quả nó là hình vuông Nhìn chung thì cũng không có gì phức tạp cả nó chỉ là 1 hệ thống phân cấp các khái niệm. Chẳng hạn như ví dụ ở trên chúng ta đã chia nhiệm vụ xác định hình vuông thành những nhiệm vụ nhỏ và đơn giản hơn. Deep Learning cũng hoạt động tương tự như vậy nhưng ở quy mô lớn hơn.

Chương 3: TRÍCH XUẤT DỮ LIỆU VÀ XÂY DỰNG MÔ HÌNH

3.1 Giới thiệu về Google Colaboratory

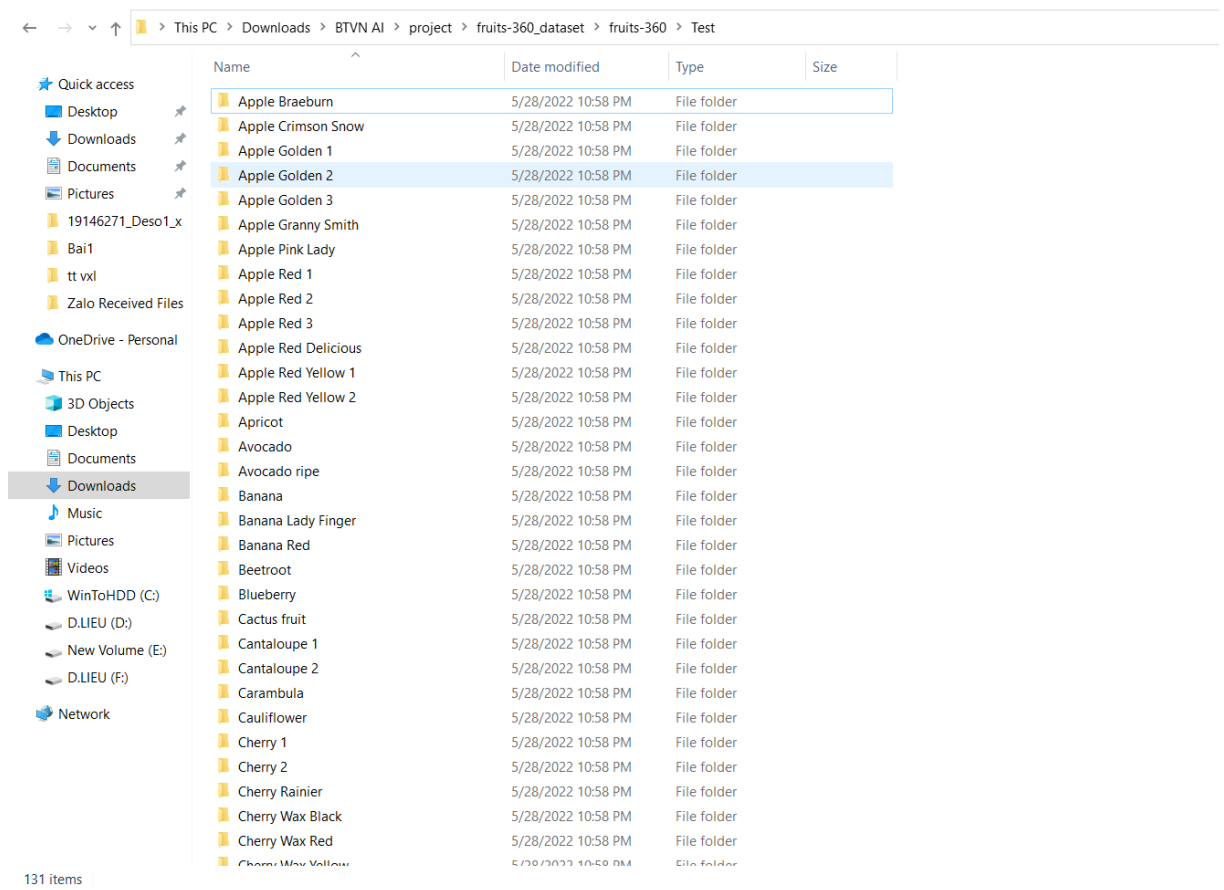
Google Colaboratory (gọi tắt là Google Colab hay Colab) là một sản phẩm của Google Research. Colab dựa trên Jupyter Notebook, người dùng có thể viết và thực thi đoạn mã python thông qua trình duyệt và đặc biệt rất phù hợp với data analysis, machine learning và giáo dục.

Google Colab là một sản phẩm từ Google Research, nó cho phép chạy các dòng code python thông qua trình duyệt, đặc biệt phù hợp với Data analysis, machine learning và giáo dục. Colab không cần yêu cầu cài đặt hay cấu hình máy tính, mọi thứ có thể chạy thông qua trình duyệt, có thể sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được cung cấp. Colab cung cấp nhiều loại GPU, thường là Nvidia K80s, T4s, P4s and P100s, tuy nhiên người dùng không thể chọn loại GPU trong Colab, GPU trong Colab thay đổi theo thời gian.

3.2 Trích xuất và xử lý dữ liệu

3.2.1 Đánh giá sơ bộ và tiền xử lý dữ liệu

Sau khi tải tập dữ liệu bao gồm 70 class và 47745 dữ liệu hình ảnh từ kaggle ta có thể đánh giá sơ qua bộ dữ liệu.



Hình 4. 70 classes từ tập dữ liệu.

3.2.2 Truy xuất dữ liệu từ drive và chia thành các tập train, test

Khai báo một số thư viện cần thiết:

+ Code + Text

```
import tensorflow as tf
from tensorflow import keras
from keras.models import Sequential
from keras.layers.convolutional import Conv2D, MaxPooling2D
from keras.layers import Flatten, Dense, Dropout, Activation
from google.colab import drive
import matplotlib.pyplot as plt
import numpy as np
```

Truy xuất tập dữ liệu đã tải từ google drive:

```
[ ] drive.mount('/content/drive',force_remount=True)
from tensorflow.keras.preprocessing.image import ImageDataGenerator
train_datagen = ImageDataGenerator(rescale=1./255,
                                   shear_range=0.2,
                                   zoom_range=0.2,
                                   horizontal_flip=True)

train=train_datagen.flow_from_directory('/content/drive/MyDrive/train10/Training',
                                       target_size=(256,256),
                                       batch_size=32,
                                       class_mode='categorical')

test=train_datagen.flow_from_directory('/content/drive/MyDrive/test10/Test',
                                       target_size=(256,256),
                                       batch_size=32,
                                       class_mode='categorical')
```

Mounted at /content/drive
Found 35755 images belonging to 70 classes.
Found 11990 images belonging to 70 classes.

Phân chia dữ liệu đã tải thành 2 tập là train và test với tập test có giá trị bằng 25% giá trị của tập train bằng `train_test_split` từ thư viện `sklearn`.

Có thể thấy được sau khi tách ta được 35755 dữ liệu thuộc 70 classes ở tập dữ liệu train và 11990 dữ liệu thuộc 70 classes ở tập dữ liệu test.

3.3 Xây dựng mô hình

Xây dựng mô hình CNN bằng Sequential từ thư viện keras.models với 1 lớp input với dữ liệu đầu vào là dữ liệu với kích thước 256x256x3, 3 lớp ẩn và lớp output với 70 classes đầu ra:

```
model=Sequential()
model.add(Conv2D(128,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same',input_shape=(256,256,3)))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(64,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D((2,2)))
model.add(Conv2D(32,(3,3),activation='relu',kernel_initializer='he_uniform',padding='same'))
model.add(MaxPooling2D((2,2)))
model.add(Flatten())
model.add(Dense(128,activation='relu',kernel_initializer = 'he_uniform'))
#model.add(Dropout(0,2))
model.add(Dense(70,activation='Softmax'))
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.callbacks import EarlyStopping
#opt = SGD(lr = 0.01, momentum = 0.9)
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy',metrics = ['accuracy'])
callbacks=[EarlyStopping(monitor='val_loss',patience=100)]
```

CHƯƠNG 4: HUẤN LUYỆN, ĐÁNH GIÁ MÔ HÌNH VÀ KẾT QUẢ THU ĐƯỢC

4.1 Huấn luyện mô hình

Huấn luyện với mô hình đã tạo với tập huấn luyện là train, tập thẩm định là test, batch size =64, số lần huấn luyện là 8 lần:

```
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['accuracy'])
callbacks=[EarlyStopping(monitor='val_loss', patience=100)]
history=model.fit(train,
                  steps_per_epoch=len(train),
                  batch_size = 64,
                  epochs=10,
                  validation_data=test,
                  validation_steps=len(test),
                  callbacks=callbacks,
                  verbose = 1)
```

Kết quả thu được ở năm lần huấn luyện thu được ta có thể thấy độ chính xác đối với tập train là 90% và độ chính xác của mô hình khi chạy với tập thẩm định

```
Epoch 1/8
648/648 [=====] - 9415s 15s/step - loss: 2.2061 - accuracy: 0.4328 - val_loss: 0.9558 - val_accuracy: 0.7341
Epoch 2/8
648/648 [=====] - 443s 684ms/step - loss: 0.2851 - accuracy: 0.9110 - val_loss: 0.7096 - val_accuracy: 0.8194
Epoch 3/8
648/648 [=====] - 439s 678ms/step - loss: 0.1370 - accuracy: 0.9597 - val_loss: 0.7081 - val_accuracy: 0.8501
Epoch 4/8
648/648 [=====] - 440s 679ms/step - loss: 0.0799 - accuracy: 0.9755 - val_loss: 0.3884 - val_accuracy: 0.9107
Epoch 5/8
648/648 [=====] - 438s 677ms/step - loss: 0.0840 - accuracy: 0.9763 - val_loss: 0.2917 - val_accuracy: 0.9245
Epoch 6/8
648/648 [=====] - 440s 679ms/step - loss: 0.0605 - accuracy: 0.9830 - val_loss: 0.5715 - val_accuracy: 0.8852
Epoch 7/8
648/648 [=====] - 438s 676ms/step - loss: 0.0613 - accuracy: 0.9841 - val_loss: 0.4342 - val_accuracy: 0.9226
Epoch 8/8
648/648 [=====] - 438s 676ms/step - loss: 0.0591 - accuracy: 0.9849 - val_loss: 0.4348 - val_accuracy: 0.9281
```

là gần 70%

4.2 Đánh giá mô hình

Từ kết quả mô hình sau khi huấn luyện như trên ta có thể đánh giá sơ qua kết quả của mô hình như sau: khi huấn luyện độ chính xác chính xác của mô hình khá cao đạt gần 93% ở kết quả thu được nhưng khi thẩm định thì kết quả thu được độ chính xác chỉ giao động ở 80% cho thấy được chất lượng mô hình và data chưa thực sự tối ưu nhưng vẫn trong khoảng chấp nhận được.

✓
1m



```
#đánh giá chất lượng của mô hình và vẽ lại  
score = model.evaluate(test,verbose=0)  
print('Sai số kiểm tra là: ',score[0])  
print('Độ chính xác kiểm tra là: ',score[1])
```

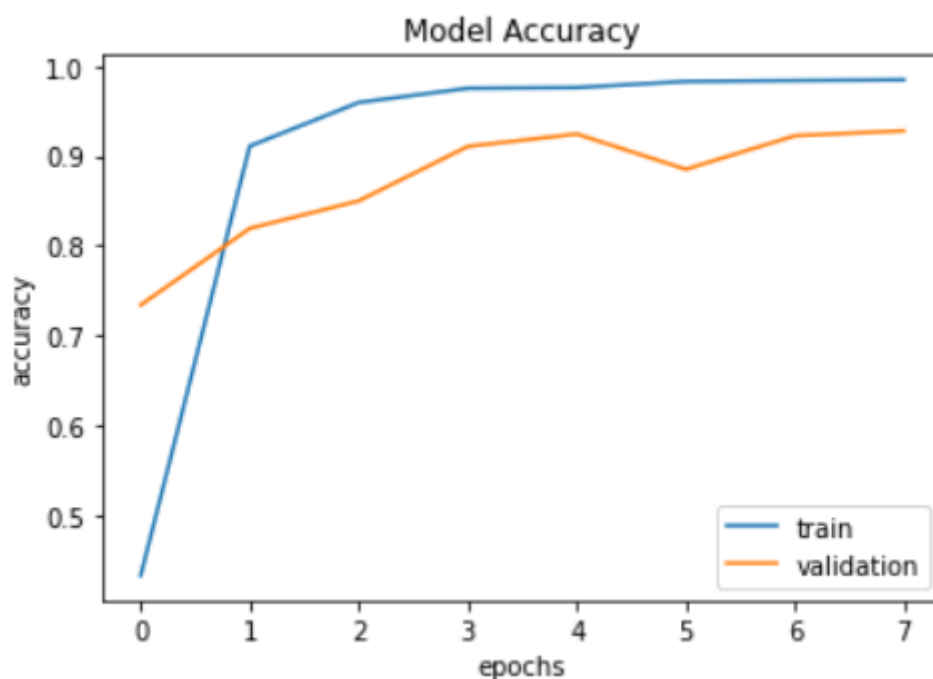


```
Sai số kiểm tra là: 0.4282713830471039  
Độ chính xác kiểm tra là: 0.9273504018783569
```

Tiến hành vẽ biểu đồ thể hiện giá thu được sau mỗi lần huấn luyện với thư viện matplotlib:

```
plt.plot(history.history['accuracy'])  
plt.plot(history.history['val_accuracy'])  
plt.title('Model Accuracy')  
plt.ylabel('accuracy')  
plt.xlabel('epochs')  
plt.legend(['train', 'validation'],loc='upper_left')
```

Kết quả thu được:



Hình 5. Đồ thị biểu diễn độ chính xác của mô hình

4.3 Thử nghiệm kết quả thu được

Ta tiến hành thử nghiệm kết quả thu được của mô hình đã huấn luyện bằng cách tải mô hình đã huấn luyện bằng load_model từ thư viện keras

```
[ ] from tensorflow.keras.models import load_model
    model=load_model('model_fruit70.h5')
```

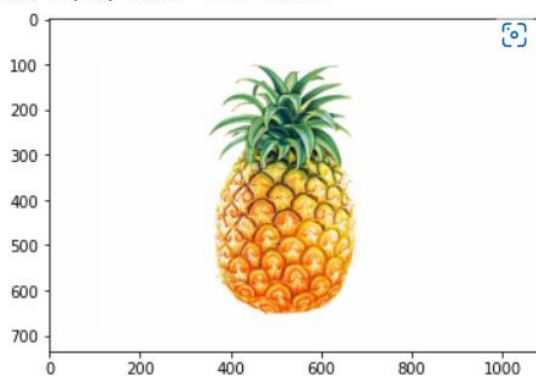
Tiến hành đặt tên cho 70 class ứng với 70 loại trái cây:

```
fruit = ['Apple Braeburn','Grape Pink','Hazelnut','Kaki','Kiwi','Kohlrabi','Kumquats','Lemon Meyer',
'Limes','Lychee','Mandarine','Mango Red','Mangostan','Maracuja','Melon Piel de Sapo','Mulberry',
'Nectarine','Nectarine Flat','Nut Forest','Nut Pecan','Onion Red','Onion Red Peeled','Onion White',
'Orange','Papaya','Passion Fruit','Peach','Peach Flat','Pear','Pear Abate','Pear Forelle','Pear Kaiser',
'Pear Monster','Pear Red','Pear Stone','Pear Williams','Pepino','Pepper Green','Pepper Orange','Pepper Red',
'Pepper Yellow','Physalis','Physalis with Husk','Pineapple','Pineapple Mini','Pitahaya Red','Plum',
'Pomegranate','Pomelo Sweetie','Potato Red','Potato Red Washed','Potato Sweet','Potato White',
'Quince','Rambutan','Raspberry','Redcurrant','Salak','Strawberry','Strawberry Wedge','Tamarillo',
'Tangelo','Tomato 1','Tomato Cherry Red','Tomato Heart','Tomato Maroon','Tomato not Ripened',
'Tomato Yellow','Walnut','Watermelon']
```

Sử dụng load_img và img_to_array từ thư viện tensorflow để thử nghiệm với các kết quả bên ngoài:

```
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
from keras.preprocessing import image
%matplotlib inline
uploaded=files.upload()
for fn in uploaded.keys():
    #predicting images
    path='/content/' + fn
    #In ảnh đọc được
    plt.imshow(mpimg.imread(path))
    img=image.load_img(path,target_size=(256,256))
    x=image.img_to_array(img)
    x=np.expand_dims(x,axis=0)
    images=np.vstack([x])
    y_predict = model.predict(images,batch_size=10)
    print(y_predict)
    print('Giá trị dự đoán: ', fruit[np.argmax(y_predict)])
```

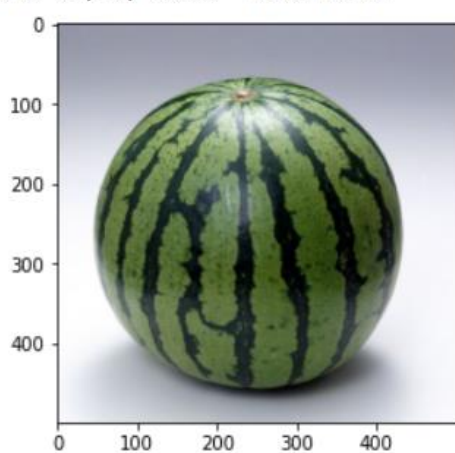
Giá trị dự đoán: Pear Abate



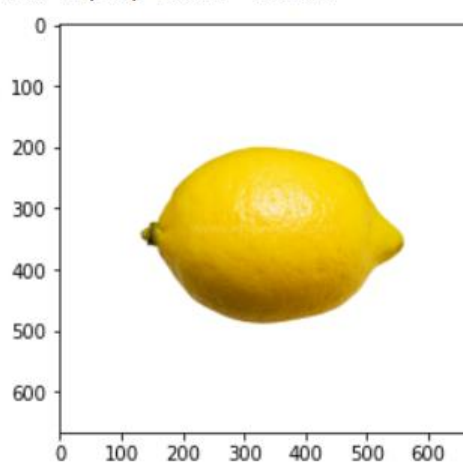
Giá trị dự đoán: Pitahaya Red



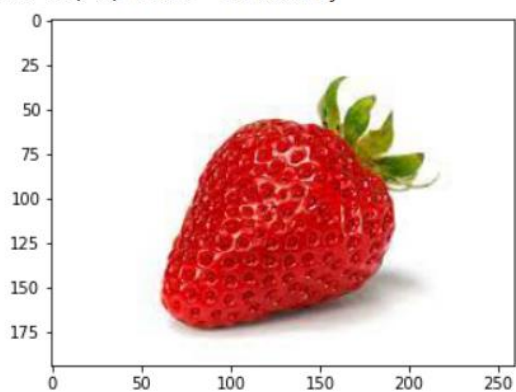
Giá trị dự đoán: Watermelon



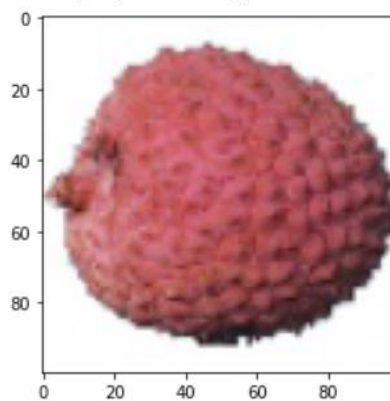
Giá trị dự đoán: Lemon



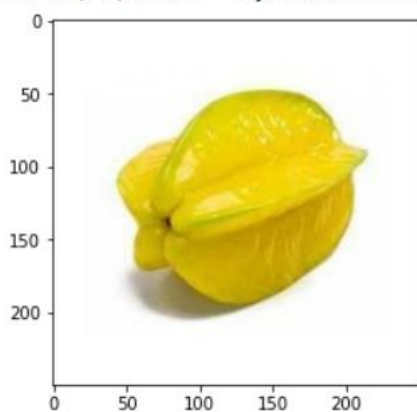
Giá trị dự đoán: Strawberry



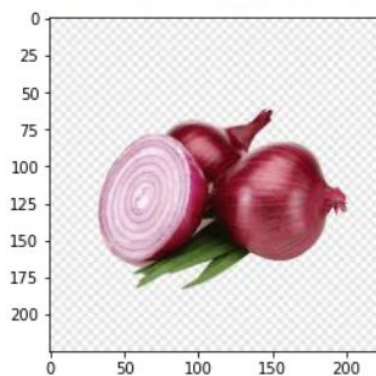
Giá trị dự đoán: Lychee



Giá trị dự đoán: Physalis with Husk

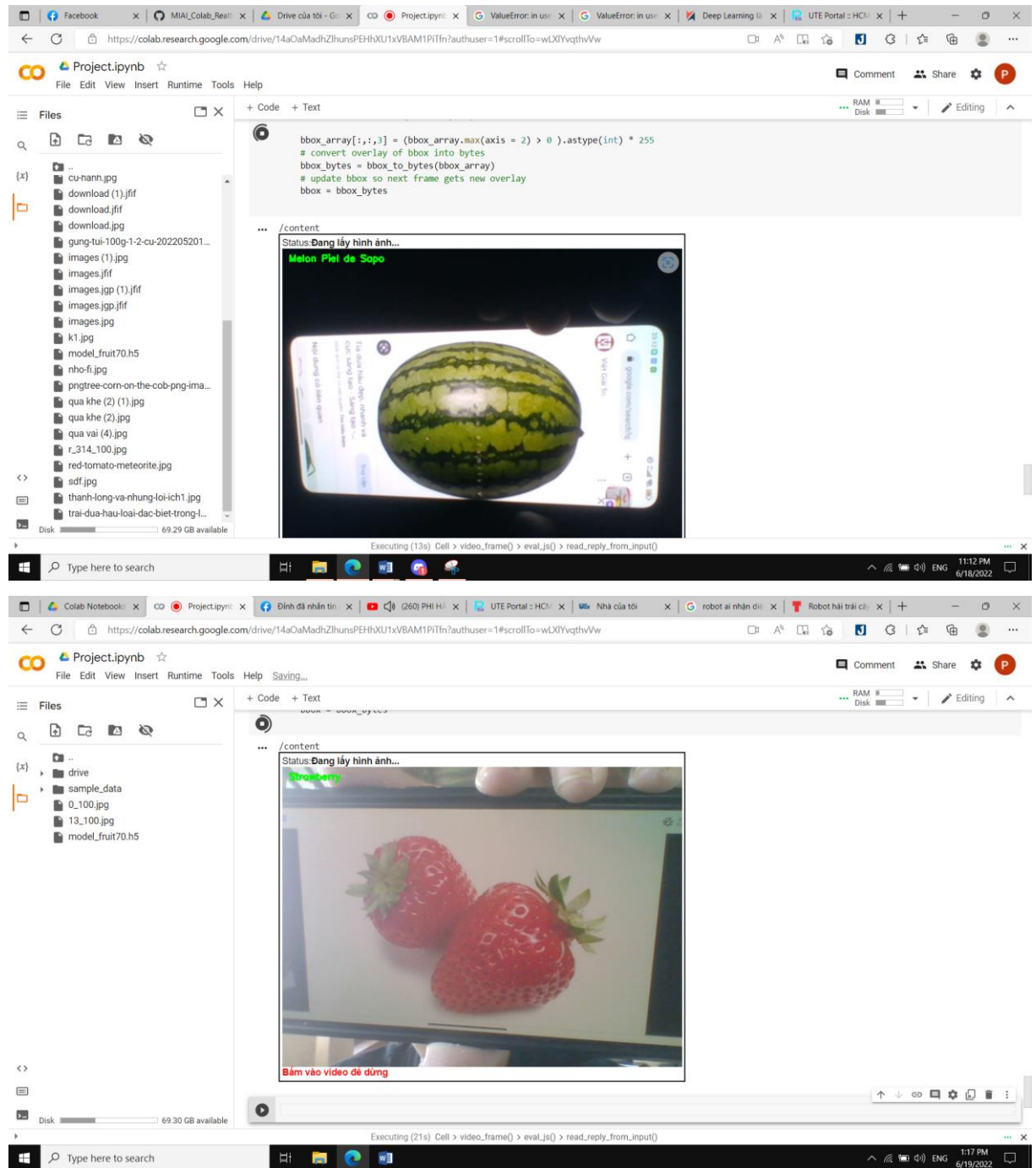


Giá trị dự đoán: Melon Piel de Sapo



Có thể thấy được từ 8 mẫu thử bất kì như trên mô hình đem lại độ chính xác không quá cao nhưng vẫn trong khoảng có thể chấp nhận được.

4.4 Chạy thử mô hình real-time trên môi trường google colab



Chương 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

Sau quá trình thực hiện đồ án nghiên cứu và ứng dụng trí tuệ nhân tạo (Artificial Intelligence) nhận diện trái cây đã hoàn được một số công việc cơ bản như sau:

Nghiên cứu được các cơ sở lí thuyết và chọn lựa được các phương pháp, xây dựng mô hình huấn luyện.

Hoàn thành việc xây dựng và viết code chạy mô hình trên google colab.

Vì thời gian thực hiện đồ án nghiên cứu và ứng dụng trí tuệ nhân tạo (Artificial Intelligence) phân loại trái cây là có giới hạn và còn nhiều thiếu sót về kiến thức mà đồ án vẫn còn nhiều sai sót và còn nhiều lỗi cũng như độ chính xác và hàm lượng khoa học chưa cao.

5.2 Hướng phát triển

Do phụ thuộc vào nhiều yếu tố khách quan lẫn chủ quan, trình độ có giới hạn nên hiện tại đề tài vẫn còn nhiều vấn đề chưa thể thực hiện, cần tiếp tục nghiên cứu và thực nghiệm để hoàn thiện mô hình. Vì vậy hướng phát triển tiếp theo của đề tài cần tập trung vào những cần đề sau để hoàn thiện mô hình đưa vào thực tế:

- + Hoàn thiện mô hình và kiểm nghiệm những tính toán.
- + Thực hiện khảo sát thực tế để biết được nhiều yếu tố khách quan hơn.
- + Thử nghiệm trên các mô hình khác nhau để đưa ra kết quả tốt hơn.
- + Bổ sung tập dữ liệu để dữ liệu phong phú hơn cũng như tăng thêm độ chính xác cho mô hình.

PHỤ LỤC

Link github của toàn bộ code:

<https://github.com/chjf123456789/VoNguyenPhuThinh.git>

TÀI LIỆU THAM KHẢO

[1] Chris Albon (2018), “Machine Learning with Python Cookbook Practical Solutions from Preprocessing to Deep Learning”, Published by O’Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.

[2] Vũ Hữu Tiệp, Machine Learning cơ bản.

[3] Nguyễn Mạnh Hùng, “Nghiên cứu về mạng Neural convolutional, áp dụng vào bài toán nhận dạng đối tượng trong lĩnh vực thị giác máy tính”, trường đại học Công Nghệ, 2019.

[4] Lê Thị Thu Hằng, “Nghiên cứu về mạng neural tích chập và ứng dụng cho bài toán nhận dạng biển số xe”, trường đại học Công Nghệ, 2016.

[5] [Robot hái trái cây tự động – giải pháp mới ngành nông nghiệp - Tạp chí tự động hóa ngày nay | Automation today \(vnautomate.net\)](#)