

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



ĐOÀN MẠNH CƯỜNG

**ỨNG DỤNG XỬ LÝ ẢNH TRONG
PHÂN LOẠI HOA QUẢ**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ kỹ thuật Điện tử Truyền Thông

HÀ NỘI – 2021

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

ĐOÀN MẠNH CƯỜNG

**ỨNG DỤNG XỬ LÝ ẢNH TRONG
PHÂN LOẠI HOA QUẢ**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ kỹ thuật Điện tử Truyền Thông

Cán bộ hướng dẫn: ThS. Phạm Đình Tuấn

HÀ NỘI - 2021

TÓM TẮT

Tóm tắt: Trong những năm gần đây sự tiến bộ của công nghệ đã cho thấy tính hiệu quả trong việc triển khai các giải pháp tự động trong nhiều lĩnh vực của đời sống. Học máy (Machine Learning) là một công cụ mạnh mẽ đã được ứng dụng vào nhiều lĩnh vực nhằm mục đích tự động hóa các hoạt động cơ bản và tối ưu hóa kết quả của các hoạt động này. Sự phát triển của học máy cùng vai trò và đóng góp của nó là không thể phủ nhận: xác định đối tượng, phân loại, ... mà không cần sự can thiệp của con người, xử lý và phân tích dữ liệu. Tuy nhiên, một số ứng dụng ít được nhìn thấy hơn nhưng có ý nghĩa và tầm quan trọng lớn trong đời sống. Một trong những ứng dụng ít được nhìn thấy này là phân loại trái cây và hoa quả.

Khoá luận *“Ứng dụng xử lý ảnh trong phân loại hoa quả”* xây dựng một cách cơ bản việc kết hợp xử lý ảnh sử dụng mạng nơ-ron tích chập (CNN) trong một hệ thống phân loại hoa quả, hệ thống này sẽ giúp con người giảm thời gian, công sức cần thiết cho việc phân loại hoa quả và loại bỏ nhu cầu tiếp xúc trực tiếp với nhiều nông sản cùng với khả năng giám sát trực quan quá trình phân loại qua màn hình. Với các chủng vi rút và vi khuẩn mới gây ra các vấn đề sức khỏe trên toàn cầu hiện nay, việc có thể hạn chế tiếp xúc nông sản từ những người không cần thiết và không chuyên thông qua tự động hóa có thể giúp giải quyết các vấn đề an toàn vệ sinh thực phẩm.

Từ khoá: *Machine Learning, CNN, xử lý ảnh, phân loại sản phẩm.*

LỜI CAM ĐOAN

Tôi xin cam đoan khoá luận tốt nghiệp: “Ứng dụng xử lý ảnh trong phân loại hoa quả” là công trình nghiên cứu của bản thân tôi với sự hướng dẫn của Ths Phạm Đình Tuấn. Các số liệu, kết quả thực nghiệm trong khoá luận là những gì tôi đã nghiên cứu, và chưa từng được công bố trong bất kỳ công trình nào. Việc sử dụng các tài liệu tham khảo khác đều được trích dẫn đầy đủ.

Nếu phát hiện bất kỳ sự gian lận nào, tôi xin chịu trách nhiệm theo quy định của trường Đại học Công Nghệ - Đại học Quốc Gia Hà Nội.

Sinh viên thực hiện

Đoàn Mạnh Cường

LỜI CẢM ƠN

Lời đầu tiên, tôi xin gửi lời cảm ơn chân thành nhất tới ThS. Phạm Đình Tuấn về những lời khuyên vô giá, cùng sự tận tình hướng dẫn và giúp đỡ tôi nghiên cứu, hoàn thành khóa luận này.

Tôi cũng xin chân thành cảm ơn các quý thầy cô giảng viên trong trường Đại học Công nghệ - Đại học Quốc gia Hà Nội đã tạo điều kiện thuận lợi cho tôi trong quá trình học tập tại trường.

Bên cạnh đó, tôi cũng xin cảm ơn gia đình bạn bè những người thân yêu đã động viên, hỗ trợ và giúp đỡ tôi hoàn thành tốt khóa luận này.

Hà Nội, ngày ... tháng 05 năm 2021

Sinh viên

Đoàn Mạnh Cường

MỤC LỤC

TÓM TẮT	1
LỜI CAM ĐOAN	2
LỜI CẢM ƠN	3
DANH MỤC HÌNH VẼ.....	6
DANH MỤC BẢNG BIỂU	7
DANH MỤC CỤM TỪ VIẾT TẮT	8
MỞ ĐẦU	9
CHƯƠNG 1. TỔNG QUAN	12
1.1. Bài toán phân loại hoa quả	12
1.2. Hướng tiếp cận và giải quyết.....	13
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	14
2.1. Mạng nơ-ron Tích Chập (Convolution Neural Network).....	14
2.1.1. Mạng nơ-ron	14
2.1.2. Học Sâu (Deep Learning)	14
2.1.3. Mạng nơ-ron Tích Chập.....	15
2.1.4. Kiến trúc mạng CNN	16
2.2. Mạng huấn luyện VGG	21
2.2.1. Kiến trúc mạng VGG16.....	22
2.2.2. Ứng dụng mạng VGG cho bài toán phân loại hoa quả.....	23
2.3. Thiết bị phần cứng.....	24
2.3.1. Raspberry Pi 4.....	24
2.3.2. Camera Pi.....	25
2.3.3. Arduino Uno	26
2.4. Phần Mềm.....	27
2.4.1. Google Colab	27
2.4.2. Raspberry Pi OS.....	28
2.4.3. Open CV [13].....	28
2.4.4. Flask Python framework	29
2.4.5. Các thư viện liên quan	29
2.4.5.1. Tensorflow	29
2.4.5.2. Keras	29

2.4.5.3. Tensorflow Lite.....	30
2.4.5.4. Numpy	30
2.4.5.5. Scikit-learn.....	31
2.5. Website	31
CHƯƠNG 3: XÂY DỰNG, THIẾT KẾ HỆ THỐNG VÀ THỰC NGHIỆM.....	33
3.1. Xây dựng và thiết kế hệ thống.....	33
3.1.1 Cấu trúc hệ thống.....	33
3.1.2. Mô hình phần cứng	33
3.2. Thực nghiệm.....	35
3.2.1. Thiết lập đào tạo và thử nghiệm	35
3.2.2. Đánh giá mô hình.....	40
3.2.3. Thiết lập dự đoán cho Raspberry Pi.....	41
3.2.4. Phân loại trên băng tải.....	43
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	46
TÀI LIỆU THAM KHẢO.....	47

DANH MỤC HÌNH VẼ

Hình 2.1 Tế bào thần kinh sinh học.....	14
Hình 2.2 Mối quan hệ giữa học sâu với học máy.....	15
Hình 2.3 Cấu trúc CNN	16
Hình 2.4 Tích chập	17
Hình 2.5 Stride.....	17
Hình 2.6 Pooling Layer	18
Hình 2.7 Max Pooling	18
Hình 2.8 Hàm kích hoạt.....	21
Hình 2.9 Kiến trúc VGG16	22
Hình 2.10 Raspberry Pi 4	24
Hình 2.11 Camera Pi v2	25
Hình 2.12 Arduino Uno	26
Hình 2.13 Raspberry Pi OS	28
Hình 2.14 Giao diện website	32
Hình 3.1 Cấu trúc hệ thống.....	33
Hình 3.2 Hệ thống băng tải.....	34
Hình 3.3 Lưu đồ thuật toán đào tạo	35
Hình 3.4 Tăng dữ liệu.....	36
Hình 3.5 Tăng dữ liệu (thay đổi độ sáng).....	36
Hình 3.6 Thống kê dữ liệu đào tạo	37
Hình 3.7 Thống kê dữ liệu xác thực	37
Hình 3.8 Hàm kích hoạt Relu	37
Hình 3.9 Dropout.....	38
Hình 3.10 Quá trình đào tạo	39
Hình 3.11 Đồ thị Accuracy and Loss	39
Hình 3.12 Kết quả đánh giá mô hình.....	40
Hình 3.13 Mô hình chuyển đổi Tensorflow Lite.....	41
Hình 3.14 Kiểm tra sau khi đào tạo	42
Hình 3.15 Quá trình phân loại	43
Hình 3.16 Kết quả kiểm tra hình ảnh	44
Hình 3.17 Kết quả phân loại thực tế.....	45

DANH MỤC BẢNG BIỂU

Bảng 2.1 Thông số Arduino Uno	26
Bảng 2.2 Cấu hình Google Colab.....	27

DANH MỤC CỤM TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
CNN	Convolution Neural Network - Mạng nơ-ron tích chập
ReLU	Rectified Linear Units - Đơn vị tuyến tính chỉnh lưu
CPU	Central Processing Unit - Vi xử lý trung tâm
GPU	Graphics Processing Unit - Bộ xử lý đồ họa
TPU	Tensor Processing Unit - Bộ xử lý Tensor
RGB	Red Green Blue - Mô hình màu RGB

MỞ ĐẦU

Trong những năm gần đây sự tiến bộ của công nghệ đã cho thấy tính hiệu quả trong việc triển khai các giải pháp tự động trong nhiều lĩnh vực của đời sống. Sự phát triển của học máy cùng vai trò và đóng góp của nó là không thể phủ nhận: xác định đối tượng, phân loại mà không cần sự can thiệp của con người, xử lý và phân tích dữ liệu. Tuy nhiên, một số ứng dụng ít được nhìn thấy hơn nhưng có ý nghĩa và tầm quan trọng lớn trong đời sống. Một trong những ứng dụng ít được nhìn thấy này là phân loại trái cây và rau quả. Cùng với sự phát triển của canh tác thông minh đã làm nổi bật việc sử dụng học máy cho một số khía cạnh chuyên sâu trong nông nghiệp như thu thập dữ liệu trên thực địa, phân tích dữ liệu, giám sát, dự đoán, v.v. Tất cả các ứng dụng và nghiên cứu này nhằm mục đích nâng cao năng suất, chất lượng sản phẩm và giúp quản lý các nguồn tài nguyên hạn chế và thích ứng với biến đổi khí hậu.

Học máy đã mang lại khả năng tự động hóa và tính toán mạnh mẽ cho canh tác quy mô lớn theo nhiều cách, bao gồm các hệ thống phân loại trái cây cũng như phát hiện bệnh tật và khiếm khuyết. Các hệ thống này cũng đã được sử dụng cho mục đích nghiên cứu để giúp chúng ta hiểu và giải quyết các vấn đề đang gây ra cho ngành nông nghiệp. Ngoài ra, đối với việc bảo quản, vận chuyển và kiểm tra nông sản, học máy đã đóng một vai trò trong việc giúp các chuyên gia đưa ra các quyết định sáng suốt và hiệu quả hơn.

1. Đặt vấn đề

Hình thức của một loại trái cây hoặc rau quả là cơ sở đầu tiên để đưa ra quyết định mua hàng. Người bình thường sẽ không thể nhìn hình thức của một loại trái cây để chắc chắn rằng trái cây hay rau củ đó có ngon hay không. Tuy nhiên, ở một mức độ nào đó, những kết luận rút ra từ phân tích cơ bản về trái cây hoặc rau quả của khách hàng có thể thực sự cho thấy chất lượng của trái cây đó, điều đó có nghĩa là nhà cung cấp hoặc nhà bán lẻ cần đảm bảo rằng hoa quả họ bán phù hợp với mong muốn người tiêu dùng, để thúc đẩy họ mua hàng. Vì lý do này, cần rất nhiều thời gian và công sức dành cho việc kiểm tra và phân loại trái cây và rau quả tại các siêu thị và cửa hàng.

Hơn nữa, việc xác định chất lượng hoa quả cũng giúp giảm thời gian và công sức cần thiết cho việc phân loại trái cây trước khi bày bán các siêu thị và loại bỏ nhu cầu tiếp xúc trực tiếp với nhiều nông sản trong chuỗi cung ứng. Cùng với việc các chủng vi rút và vi khuẩn mới gây ra các vấn đề sức khỏe trên toàn cầu, việc có thể hạn chế tiếp

xúc nông sản của những người không cần thiết và không chuyên thông qua tự động hóa có thể giúp giải quyết các vấn đề an toàn vệ sinh thực phẩm.

Phân loại trái cây là một thách thức vì rất khó để đưa ra định nghĩa về một loại trái cây. Tuy nhiên, việc phân loại trái cây có thể giúp ích trong việc đóng gói và vận chuyển trái cây tự động tại nhà máy [10], xác định giá ở siêu thị [8]. Hiện nay, có hai cách phân loại quả: một là xác định loại quả cụ thể, hai là phân loại nhiều loại quả. Trước đây, nhiều nhà nghiên cứu có xu hướng sử dụng hình ảnh hồng ngoại gần [15], cảm biến khí [16] để quét quả. Tuy nhiên, các phương pháp đó cần các thiết bị đắt tiền (các loại cảm biến khác nhau) và người vận hành chuyên nghiệp, và độ chính xác tổng thể của chúng thường thấp hơn 85% [11]. Phân loại trái cây dựa trên xử lý ảnh đã thu hút sự chú ý trong thời gian qua vì chỉ cần chuẩn bị một số thiết bị với giá thành vừa phải và có hiệu suất tuyệt vời. [20]

2. Mục tiêu

Mục đích của khoá luận này là phát triển một hệ thống có thể phân loại các lớp đối tượng hoa quả. Do thời gian hạn chế trong thời gian thực hiện, khoá luận trước hết xây dựng mô hình xử lý ảnh kết hợp với máy tính nhúng để phân loại. Bên cạnh đó hình ảnh của đối tượng cần xử lý sẽ được giám sát trực tiếp trên màn hình hiển thị cùng từng nhãn đối tượng đã qua.

Tôi sẽ đào tạo một mạng nơ-ron tích chập (CNN) được sử dụng để giải quyết vấn đề xác định, phân loại các loại quả. Với mục đích này, khoảng 3.000 hình ảnh trái cây được sử dụng để đào tạo mô hình và khoảng 1.000 hình ảnh được sử dụng để thử nghiệm độ chính xác của mô hình. Lý do chọn phương pháp này là vì dấu hiệu chính của trái cây hoặc rau bị thối rữa hoặc hư hỏng là sự thay đổi về màu sắc và màu sắc có thể được sử dụng làm tiêu chí để phân loại trái cây. [12] [14]

Bằng cách này, mô hình được xây dựng để phát hiện các loại trái cây trông khỏe mạnh hoặc thối rữa dựa trên sự thay đổi màu sắc. Khoá luận này hy vọng sẽ cung cấp một phương pháp giúp tiết kiệm thời gian và công sức trong quá trình phân loại tại nhà vườn, trang trại và cũng như giúp ích ở một mức độ nhất định trong việc kiểm tra và phân loại hoa quả.

3. Mặt hạn chế

Việc làm cho mô hình phân loại có thể phát hiện đối tượng trái cây hồng là một khía cạnh quan trọng của khoá luận này. Tuy nhiên, vì thiếu dữ liệu đáng kể hình ảnh về trái cây hồng cho tất cả các ví dụ về trái cây của tôi, do đó để thay thế, tôi đã xử lý trước dữ liệu của mình bằng cách tăng dữ liệu [7]. Việc không có nguồn dữ liệu lớn là một hạn chế đối với việc đào tạo mô hình.

4. Nội dung khoá luận

Đề tài: “**Ứng dụng xử lý ảnh trong phân loại hoa quả**” gồm những nội dung:

Chương 1: Tổng quan: Tổng quan một số phương pháp phân loại hoa quả, học máy, kiến trúc mạng VGG và ứng dụng của mạng VGG cho bài toán phân loại hoa quả.

Chương 2: Cơ sở lý thuyết.

Chương 3: Xây dựng, thiết kế hệ thống: Cấu trúc hệ thống và thiết bị. Thực nghiệm và triển khai.

CHƯƠNG 1. TỔNG QUAN

1.1. Bài toán phân loại hoa quả

Phân loại và nhận dạng đối tượng trong ảnh được coi là bài toán cơ bản trong lĩnh vực Thị giác máy tính (Computer Vision), là nền tảng của nhiều bài toán khác như phân lớp, định vị, tách biệt vật thể... Nhưng có một số vấn đề mà con người chưa giải quyết được, do tồn tại một số khó khăn để máy tính có thể hiểu các thông tin từ ảnh như: [1]

- Kích thước đa dạng: Kích thước của vật thể trong đời thực rất khó để thể hiện thông qua các bức ảnh. Bằng cách tính số lượng các điểm ảnh của vật đó chiếm trong ảnh mà máy tính cũng chỉ tính toán tương đối tỉ lệ của vật thể thật so với vật trong ảnh.
- Điểm nhìn đa dạng: Có nhiều góc nhìn và vị trí khác nhau trong cùng một vật thể, dẫn đến các hình ảnh thu được về vật thể đó sẽ không giống nhau. Việc đào tạo các mô hình để máy tính hiểu được vấn đề này là một thách thức.
- Điều kiện chiếu sáng: thông tin thể hiện trong một bức ảnh sẽ bị ảnh hưởng mạnh mẽ do điều kiện chiếu sáng khác nhau, đặc biệt là ở mức độ thấp như mức độ điểm ảnh.
- Sự phức tạp của nền: Nhiều trường hợp, nền của bức ảnh với vật thể cần nhận dạng bị lẫn hoàn toàn vào nhau. Hòa tiết giữa các vật thể, có màu sắc lẫn lộn gây khó khăn trong việc nhận dạng, kể cả với thị giác con người.
- Đa dạng về chủng loại: Sự khác nhau giữa kết cấu, hình dạng, màu sắc của nhiều chủng loại khác nhau. Đây là một thử thách lớn với bài toán nhận dạng và phân lớp. Làm cách nào để nhận biết các biến thể về chủng loại của vật thể thông qua các mô hình nhận dạng của máy tính. Ví dụ hoa quả khác nhau, các vật thể khác loại thì vẫn tách biệt được đâu là đối tượng cam, xoài.

Bài toán phân loại hoa quả là một trường hợp cụ thể của bài toán nhận dạng và phân lớp do đó kế thừa các khó khăn từ bài toán gốc và kèm theo các khó khăn riêng như: chủng loại hoa quả theo mùa, địa lý, vùng miền...; số lượng nhiều cùng với vô số loại hoa quả có màu sắc, hình dáng, kết cấu giống nhau; sự đa dạng về điều kiện thổ nhưỡng hay sự biến thiên màu sắc của quả trong quá trình sinh trưởng.

Chính vì thế khoá luận này chỉ đi vào phân loại các nhãn từ đối tượng xoài trong vô số loại hoa quả khác nhau.

1.2. Hướng tiếp cận và giải quyết

Bài toán phân loại hoa quả đã xuất hiện từ lâu và có nhiều rất nhiều phương pháp đưa ra nhằm đề xuất hoặc cải tiến thuật toán nhận dạng, phân loại. Trong số đó phải kể đến như phương pháp xử lý ảnh (Image Processing), phương pháp này tập trung vào phát triển các thuật toán để trích xuất thông tin, ví dụ các tham số về hình dạng, màu sắc, kích thước..., từ bức ảnh đầu vào để nhận dạng hoa quả. Một số hạn chế của các phương pháp này là kết quả đạt được chưa được cao, cùng phạm vi áp dụng trên số lượng loại hoa quả cũng hạn chế do chỉ đơn thuần xử lý trên một vài ảnh đầu vào trong khi sự khác nhau về màu sắc, hình dạng, kích thước... của hoa quả rất phức tạp.

Trong những năm gần đây nổi lên như một bằng chứng của cách mạng khoa học lần thứ tư Học máy - Machine Learning đã được áp dụng rất thành công vào bài toán nhận dạng hoa quả, trong các thử nghiệm với phạm vi hạn chế về số lượng loại hoa quả cần nhận dạng, phương pháp này đã đạt được kết quả cao [18], hướng giải quyết bài toán đã tập trung vào ứng dụng và cải tiến các thuật toán học máy, cụ thể thử nghiệm trích chọn các đặc trưng phù hợp nhất để đưa vào huấn luyện bộ nhận dạng tự động. Theo sau sự phát triển của học máy Học máy có một nhánh đặc biệt đó là Học sâu – Deep Learning đã có nhiều bước tiến lớn cùng nhiều thành tựu, đặc biệt là trong lĩnh vực Xử lý ảnh và xử lý ngôn ngữ tự nhiên. Một trong số những ứng dụng của Học sâu được đã được áp dụng rất thành công vào bài toán phân loại hoa quả. Do đó kết quả thu được rất khả quan, tỉ lệ nhận dạng hoa quả được cải thiện khi số lượng loại hoa quả tăng lên, cùng với độ chính xác đạt được cao hơn nhiều so với các phương pháp thuần Xử lý ảnh ban đầu.

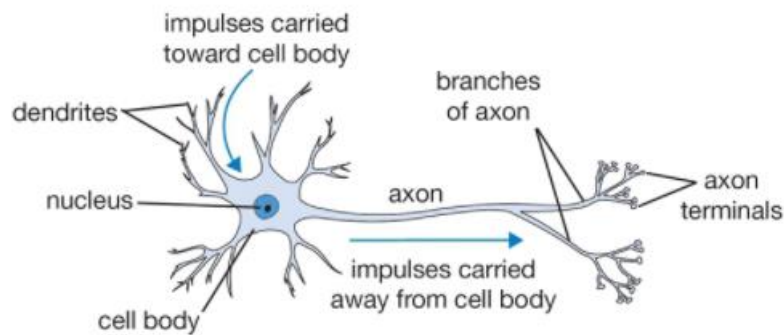
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Mạng nơ-ron Tích Chập (Convolution Neural Network)

2.1.1. Mạng nơ-ron

Mạng nơ-ron được xây dựng bằng cách mô phỏng các tế bào thần kinh hoặc mạng lưới tế bào thần kinh trong não. Tế bào thần kinh có các sợi đầu vào được gọi là đuôi gai. Axon được sử dụng để gửi tín hiệu đến các tế bào thần kinh khác, nghĩa là gửi thông tin đến các tế bào thần kinh khác. Ở mức độ đơn giản nhất, tế bào thần kinh là một đơn vị tính toán lấy giá trị đầu vào thông qua các sợi nhánh và thực hiện tính toán. Sau đó, nó gửi đầu ra qua sợi trục đến các nút khác hoặc các tế bào thần kinh khác trong não.

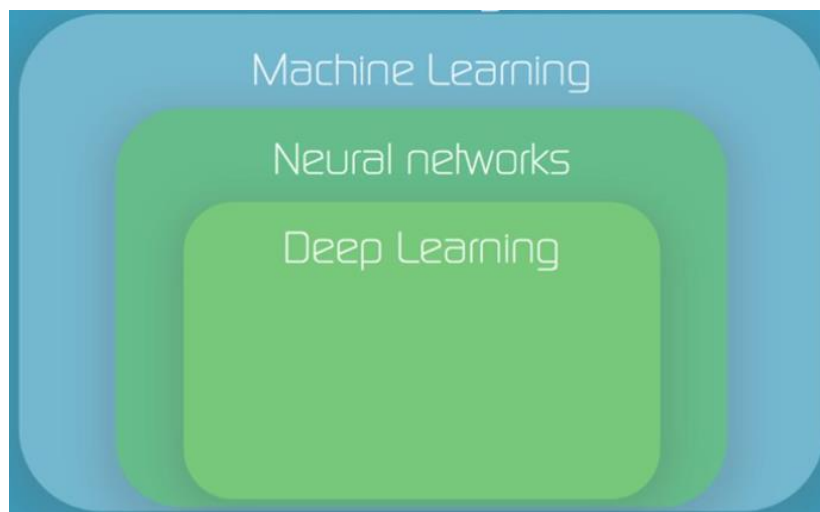
Trong mạng nơ-ron hoặc trong mạng nơ-ron nhân tạo mà chúng ta phát triển trên máy tính, chúng ta sử dụng một mô hình rất đơn giản của tế bào thần kinh [3]



Hình 2.1 Tế bào thần kinh sinh học

2.1.2. Học Sâu (Deep Learning)

Deep learning được bắt nguồn từ thuật toán Neural Network vốn xuất phát chỉ là một nhánh của Machine Learning. Deep learning là một lĩnh vực con của học máy liên quan đến các thuật toán lấy cảm hứng từ cấu trúc và chức năng của não được gọi là mạng thần kinh nhân tạo



Hình 2.2 Mối quan hệ giữa học sâu với học máy

Từ cách thức hoạt động của chúng, mạng huấn luyện theo phương pháp Học sâu còn được gọi là mạng nơ-ron sâu (Deep Neural Network). Về cơ bản, các mạng này bao gồm rất nhiều lớp khác nhau, mỗi lớp sẽ phân tích dữ liệu đầu vào theo các khía cạnh khác nhau và theo mức độ trừu tượng nâng cao dần [1]. Một phương pháp được sử dụng khá phổ biến và được cải tiến nhiều góp phần chẩn đoán chính xác và nhanh chóng là Mạng nơ-ron tích chập (CNN). Nó tự động phát hiện các tính năng quan trọng mà không cần bất kỳ sự giám sát nào của con người thông qua các lớp tích chập, lớp lấy mẫu, lớp kết nối đầy đủ.

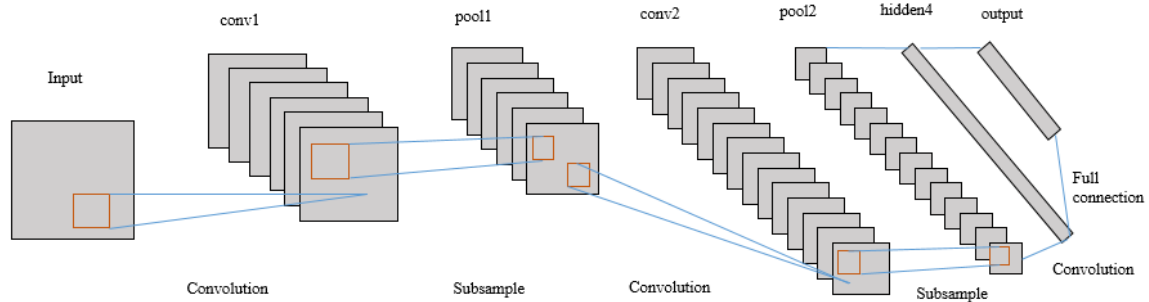
Trong khoá luận này CNN là phương pháp được lựa chọn bởi những lợi ích và sự tiện lợi của nó so với các phương pháp khác để giải quyết vấn đề tiết kiệm thời gian kiểm tra và phân loại. Phương pháp được đề xuất là tích hợp camera trên băng tải để thu thập hình ảnh và đưa về máy tính nhúng để phân tích, sau đó tiến hành phân loại dựa vào dữ liệu trả về từ máy tính nhúng.

2.1.3. Mạng nơ-ron Tích Chập

Mạng nơ-ron tích chập (Convolutional Neural Network) viết tắt là CNN hay ConvNet là một mô hình mạng học sâu phổ biến hiện nay, có khả năng nhận dạng và phân loại hình ảnh với độ chính xác cao [4]. Mô hình này được sử dụng trong các hệ thống xử lý ảnh lớn của Google, Amazon... cho các tính năng gợi ý sản phẩm hoặc tìm kiếm hình ảnh.

2.1.4. Kiến trúc mạng CNN

Mạng CNN được tạo thành từ một chuỗi các lớp chính gồm: Lớp tích chập (Convolutional), Lớp lấy mẫu (Pooling) và Lớp kết nối đầy đủ (Fully-connected), được thay đổi về số lượng và cách sắp xếp để tạo thành một kiến trúc CNN đầy đủ phù hợp với các bài toán khác nhau.



Hình 2.3 Cấu trúc CNN

Lớp Tích Chập

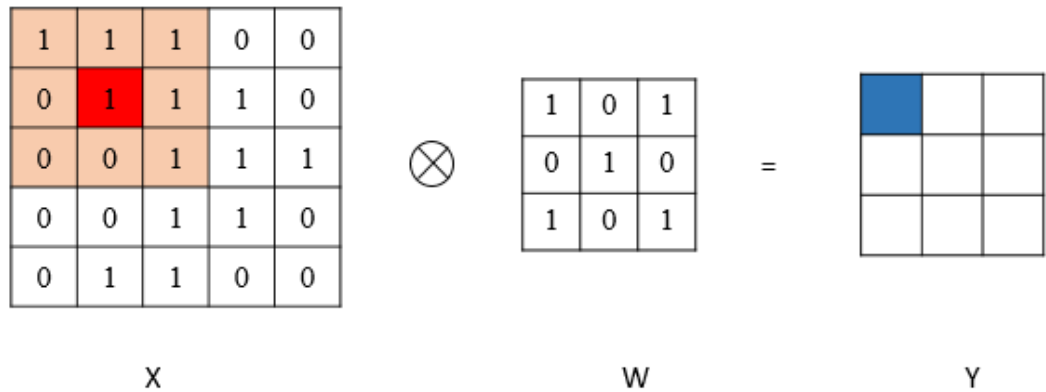
Lớp tích chập là thành phần quan trọng nhất trong mạng CNN. Trong các ứng dụng phân loại ảnh, chúng được tạo thành từ ma trận các pixel ảnh được gọi là kernel.

Phép chập là tạo ra một kernel, nó là một ma trận vuông có kích thước $k * k$ trong đó k là số lẻ, k có thể bằng 1,3,5,7,9, ... Phép tích chập được thực hiện bằng cách di chuyển kernel qua hình ảnh, thường bắt đầu ở góc trên cùng bên trái, để di chuyển kernel qua tất cả các vị trí mà kernel hoàn toàn nằm trong ranh giới của hình ảnh.

Ví dụ, một kernel $3 * 3$:

$$W = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad (1)$$

X là ma trận dữ liệu đầu vào, với mỗi phần tử x_{ij} lấy một ma trận có kích thước bằng W và x_{ij} là tâm. Sau khi tích chập, chúng ta nhận được một điểm mới trong một ma trận mới nhỏ hơn X .

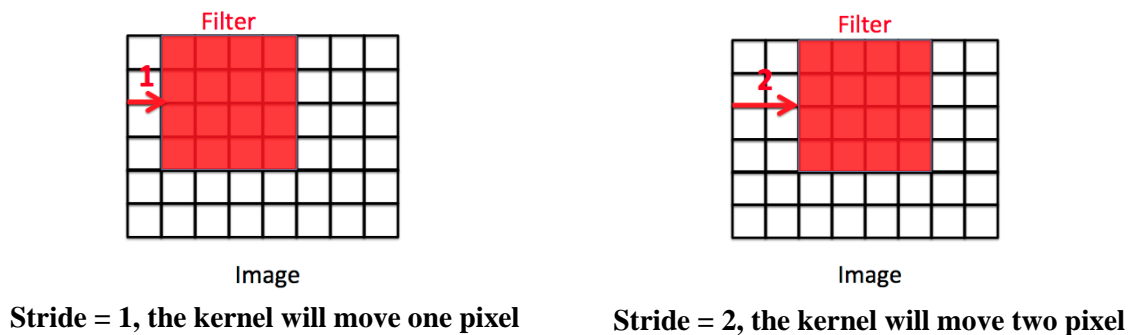


Hình 2.4 Tích chập

Với x_{22} , Tính toán $y_{11} = x_{11} * w_{11} + x_{12} * w_{12} + x_{13} * w_{13} + x_{21} * w_{21} + x_{22} * w_{22} + x_{23} * w_{23} + x_{31} * w_{31} + x_{32} * w_{32} + x_{33} * w_{33}$.

Padding là giúp giữ cho kích thước Y bằng với kích thước X ban đầu, padding = k tức là thêm k vector 0 theo mỗi hướng, thường là $k = 1$ thì các phần tử trong đường viền của X sẽ là tích chập.

Stride là một thành phần của mạng nơ-ron tích tụ hoặc mạng nơ-ron được điều chỉnh để nén hình ảnh và dữ liệu video. Stride là một tham số của bộ lọc mạng nơ-ron điều chỉnh lượng chuyển động trên hình ảnh hoặc video.

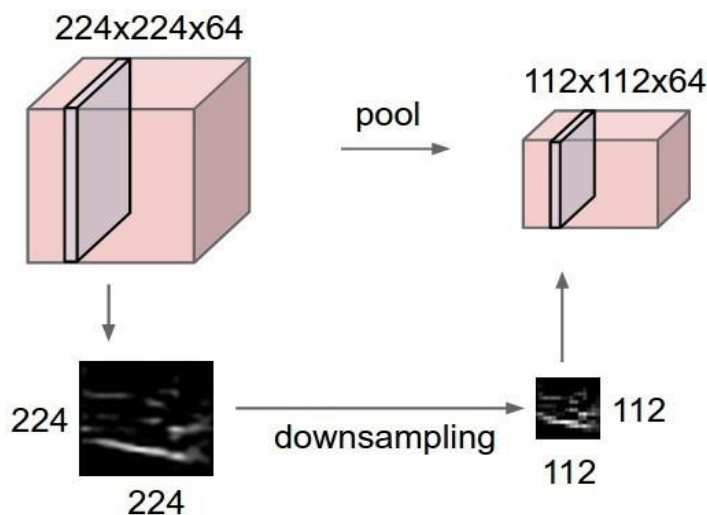


Hình 2.5 Stride

Mục đích của phép tính tích chập trên một hình ảnh là để mở và làm sắc nét hình ảnh, xác định các dòng, ... Mỗi kernel khác nhau thì phép tính tích chập sẽ có ý nghĩa khác nhau.

Lớp lấy mẫu (Pooling Layer)

Thông thường, định kỳ chèn một lớp lấy mẫu (Pooling) vào giữa các lớp chuyển đổi liên tiếp trong kiến trúc CNN. Chức năng của nó là giảm dần kích thước không gian của biểu diễn để giảm số lượng tham số, tính toán trong mạng và do đó cũng kiểm soát việc trang bị quá mức (overfitting). Trong khi giảm kích thước ảnh đầu ra nhưng lớp lấy mẫu vẫn giữ được các thông tin quan trọng của ảnh đầu vào.



Hình 2.6 Pooling Layer

Đầu vào của lớp Pooling có kích thước $H * W * D$, được tách thành D ma trận kích thước $H * W$. Đối với mỗi ma trận trên vùng kích thước $K * K$ trên ma trận ta tìm giá trị lớn nhất hoặc giá trị trung bình của dữ liệu rồi viết vào ma trận kết quả. Quy tắc về stride và padding áp dụng như phép tính tích chập.

3.0	3.0	3.0
3.0	3.0	3.0
3.0	2.0	3.0

3	3	2	1	0
0	0	1	3	1
3	1	2	2	3
2	0	0	2	2
2	0	0	0	1

Hình 2.7 Max Pooling

Sau khi hình ảnh được chuyển qua nhiều lớp chập và các lớp lấy mẫu, mô hình đã học được các đặc điểm tương đối của hình ảnh, tensor đầu ra của lớp cuối cùng, kích

thước $H * W * D$, sẽ được chuyển đổi thành một vector có kích thước $(H * W * D)$. Sau đó ta sử dụng các lớp được kết nối đầy đủ để kết hợp các đặc điểm hình ảnh để tạo đầu ra của mô hình.

Lớp kết nối đầy đủ (Fully-Connected Layer)

Lớp kết nối đầy đủ (Fully Connected) nhận đầu vào là các dữ liệu đã được làm phẳng, mà mỗi đầu vào đó được kết nối đến tất cả nơ-ron. Trong mô hình mạng CNN, các tầng kết nối đầy đủ thường được tìm thấy ở cuối mạng và được dùng để tối ưu hóa mục tiêu của mạng ví dụ như độ chính xác của lớp. So với mạng nơ-ron truyền thống, các ảnh đầu vào của lớp kết nối đầy đủ có kích thước giảm bớt rất nhiều, đồng thời vẫn giữ được các thông tin quan trọng.

Trong quá trình xây dựng mô hình có thể gặp phải một số vấn đề khiến mô hình có độ chính xác kém là Overfitting và Underfitting. Nhưng hiện nay, vấn đề phổ biến nhất xuất hiện là Overfitting [2].

Overfitting là hiện tượng mô hình tìm được quá khớp với dữ liệu đào tạo, việc này khiến cho mô hình không thể tổng quát hóa các quy luật để hoạt động với dữ liệu chưa từng được chứng kiến. Việc quá khớp này có thể dẫn đến việc chất lượng mô hình không còn tốt trên dữ liệu kiểm tra và chất lượng mô hình không còn tốt trên dữ liệu kiểm tra. Dữ liệu kiểm tra được giả sử là không biết trước, và không được dùng để xây dựng các mô hình học máy.

Mặt khác overfitting còn xảy ra khi mô hình quá phức tạp để mô phỏng dữ liệu đào tạo. Khi lượng dữ liệu đào tạo quá nhỏ trong khi độ phức tạp của mô hình quá cao. Một mô hình được coi là tốt (fit) nếu cả train error và test error đều thấp. Nếu train error thấp nhưng test error cao, ta nói mô hình bị overfitting. Nếu train error cao và test error cao, ta nói mô hình bị underfitting.

Overfitting thực sự là một vấn đề quan trọng bởi vì việc đánh giá mô hình học máy trên bộ dữ liệu huấn luyện sẽ khác biệt với việc đánh giá độ chính xác của tổng thể (những dữ liệu mà mô hình chưa gặp bao giờ).

Để tránh Overfitting ta có thể sử dụng các kỹ thuật để tìm ra điểm dừng tốt nhất trong quá trình huấn luyện, đó là kỹ thuật lấy lại mẫu (resampling methods) và kỹ thuật validation.

- Sử dụng kỹ thuật lấy lại mẫu để ước lượng độ chính xác của mô hình
- Sử dụng tập Validation test

Lấy lại mẫu (resampling methods) là kỹ thuật phổ biến hơn. Khi đó, ta sẽ chia tập dữ liệu thành k tập con. Cách này được gọi là k -fold cross validation. Điều này cho ta thực hiện huấn luyện trên các tập dữ liệu khác nhau k lần, và từ đó, xây dựng ước lượng độ chính xác của mô hình học máy với dữ liệu mới.

Sử dụng Cross-validation là một tiêu chuẩn tốt trong học máy để ước lượng độ chính xác của mô hình với bộ dữ liệu mới. Với trường hợp có nhiều dữ liệu, việc sử dụng tập Validation sẽ là một phương pháp tốt.

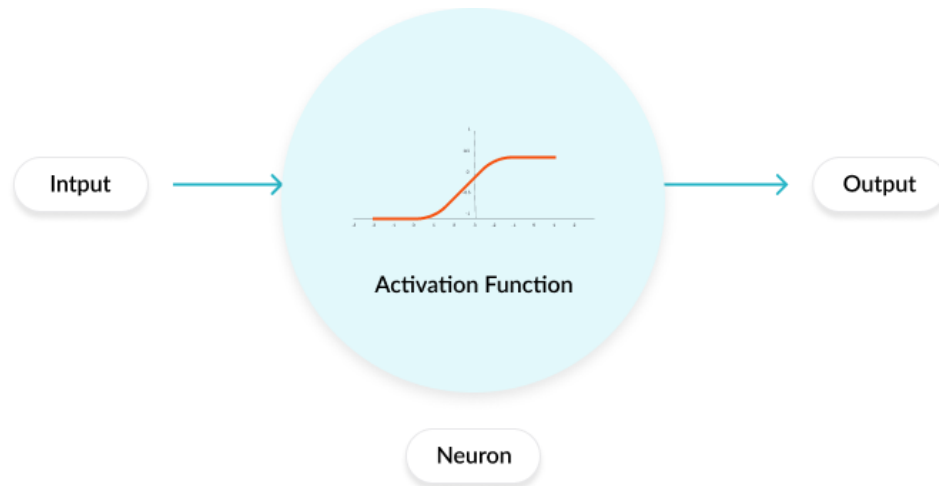
Hàm kích hoạt

Đầu ra của lớp tích chập này trước khi trở thành đầu vào của lớp tích chập của lớp tiếp theo sẽ thông qua hàm kích hoạt.

Hàm kích hoạt (Activation function) là các phương trình toán học xác định đầu ra của mạng nơ-ron. Hàm được gắn với từng nơ-ron trong mạng và xác định xem nó có nên được kích hoạt hay không, dựa trên việc liệu đầu vào của từng nơ-ron có liên quan đến dự đoán của mô hình hay không. Các hàm kích hoạt cũng giúp chuẩn hóa đầu ra của mỗi nơ-ron thành phạm vi từ 1 đến 0 hoặc từ -1 đến 1. Một khía cạnh bổ sung của các hàm kích hoạt là chúng phải hiệu quả về mặt tính toán vì chúng được tính toán trên hàng nghìn hoặc thậm chí hàng triệu nơ-ron cho mỗi mẫu dữ liệu.

Mục đích của một hàm kích hoạt là thêm một số loại thuộc tính phi tuyến tính vào hàm, đó là một mạng nơ-ron. Nếu không có các chức năng kích hoạt, mạng nơ-ron chỉ có thể thực hiện các ánh xạ tuyến tính từ đầu vào x đến đầu ra y . Nếu không có các hàm kích hoạt, phép toán duy nhất trong quá trình truyền về phía trước sẽ là các tích số giữa một vector đầu vào và một ma trận trọng số. Một mạng nơ-ron không có bất kỳ chức năng kích hoạt nào sẽ không thể nhận ra các ánh xạ phức tạp như vậy về mặt toán học và sẽ không thể giải quyết các nhiệm vụ mà chúng ta muốn mạng giải quyết.

Hàm kích hoạt là một “công” toán học ở giữa đầu vào cung cấp nơ-ron hiện tại và đầu ra của nó đi đến lớp tiếp theo. Nó có thể đơn giản như một hàm để bật và tắt đầu ra nơ-ron, tùy thuộc vào quy tắc hoặc ngưỡng. Hoặc nó có thể là một phép biến đổi ánh xạ các tín hiệu đầu vào thành các tín hiệu đầu ra cần thiết cho mạng nơ-ron hoạt động.



Hình 2.8 Hàm kích hoạt

2.2. Mạng huấn luyện VGG

Dựa trên nền tảng của mạng thần kinh cụ thể là mạng nơ-ron tích chập. Để thực hiện mục tiêu phân loại nông sản, khoá luận sử dụng phương pháp CNN. Vì là một trong những phương pháp phổ biến được sử dụng để phân loại, có nhiều mô hình cải tiến dựa trên CNN với ít tham số hơn, thời gian chạy nhanh hơn và độ chính xác cao. Nó giúp tự động trích xuất các tính năng hữu ích.

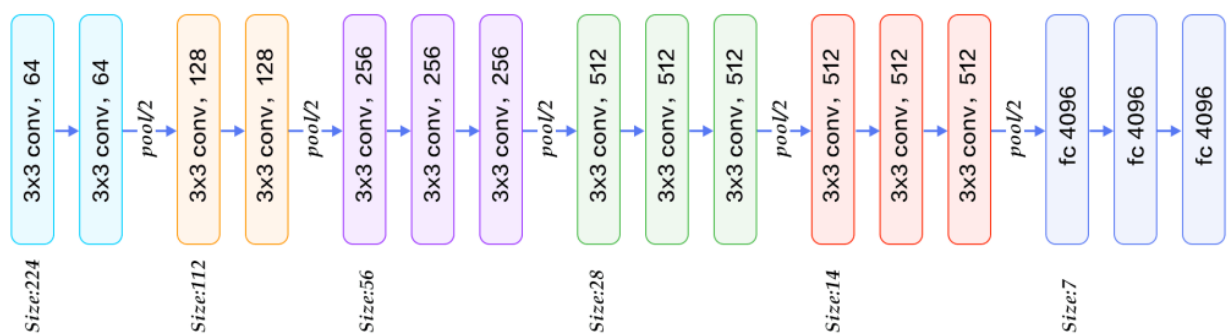
Việc phân loại sử dụng CNN ngày càng được cải thiện bởi các mô hình như AlexNet, VGG, GoogLeNet, ResNet, ... Trong khoá luận này, mô hình được sử dụng là VGG16.

VGG16 được giới thiệu vào năm 2014 bởi Karen Simonyan và Andrew Zisserman trong bài báo có tiêu đề Mạng lưới kết nối rất sâu để nhận dạng hình ảnh quy mô lớn. Mô hình đạt được độ chính xác 92,7% trong bài kiểm tra top 5 trong ImageNet trong ILSVRC-2014.

Trong phần tiếp theo ta sẽ tìm hiểu kỹ hơn về kiến trúc tổng thể của mạng VGG16 cũng như cách thức ứng dụng nó vào bài toán nhận dạng hoa quả.

2.2.1. Kiến trúc mạng VGG16

Khối cơ bản của mạng tích chập cổ điển là một chuỗi các lớp sau đây: một lớp tích chập (với phần padding để duy trì độ phân giải), một hàm kích hoạt phi tuyến như ReLU, và một lớp lấy mẫu như lớp lấy mẫu cực đại (max pooling). Một khối VGG gồm một chuỗi các lớp tích chập, tiếp nối bởi một lớp lấy mẫu cực đại để giảm chiều không gian. Trong bài báo gốc của VGG (Simonyan & Zisserman, 2014), tác giả sử dụng tích chập với các kernel 3×3 và lớp lấy mẫu cực đại 2×2 với stride bằng 2 (giảm một nửa độ phân giải sau mỗi khối) [5].



Hình 2.9 Kiến trúc VGG16

Ý tưởng cơ bản của VGG16 được tóm tắt như sau:

- Kích thước bộ lọc tối đa là 3×3 và kích thước tối thiểu là 1×1 . Điều này có nghĩa là kích thước bộ lọc nhỏ hơn với số lượng lớn hơn được sử dụng, so với kích thước bộ lọc lớn hơn và số lượng nhỏ hơn cho AlexNet; điều này dẫn đến ít tham số hơn so với AlexNet.
- Stride chuyển đổi là 1 và là 1 cho lớp tích chập 3×3 . Tổng hợp tối đa được thực hiện trên cửa sổ 2×2 với bước đi là 2.
- Ba hàm kích hoạt ReLU phi tuyến được sử dụng thay vì một hàm duy nhất trong mỗi lớp, điều này làm cho hàm quyết định trở nên phân biệt hơn bằng cách giảm vấn đề gradient biến mất và cho phép mạng tìm hiểu sâu. Học sâu ở đây có nghĩa là học các hình phức tạp, chẳng hạn như cạnh, đặc điểm, ranh giới, v.v.
- Tổng số tham số là 13
- 8 triệu.

2.2.2. Ứng dụng mạng VGG cho bài toán phân loại hoa quả

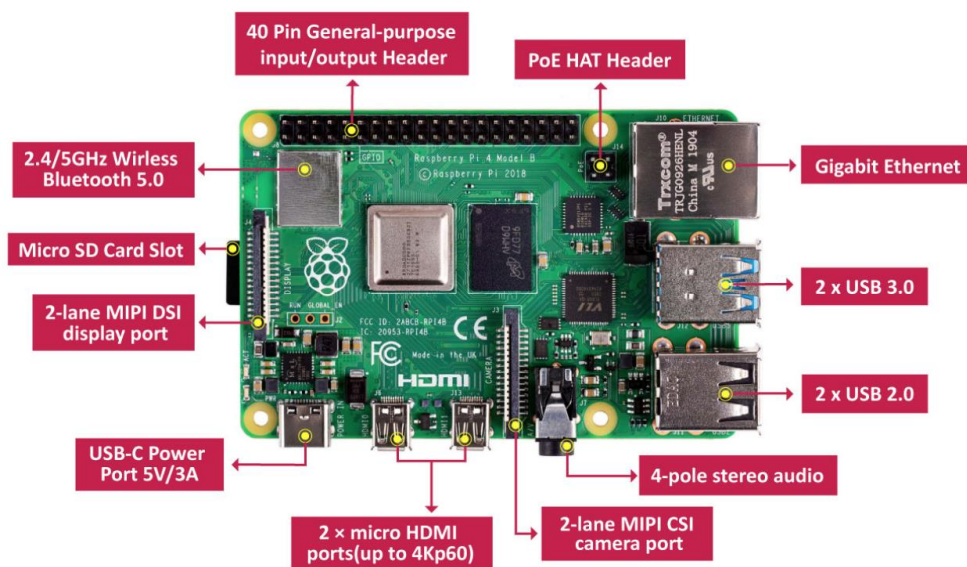
Thuật toán được đào tạo trước với VGG16 trên Google Colab cùng bộ dữ liệu hình ảnh đầu vào cho quá trình đào tạo, sau đó lưu lại mô hình rồi cài đặt trên máy tính nhúng Raspberry Pi 4. Dữ liệu từ camera trên bảng tải được gửi đến máy tính nhúng dựa trên mô hình đào tạo đã có, tiến hành phân loại cho từng bức ảnh. Kết quả sẽ gửi đến Arduino để điều khiển Servo tiến hành phân loại, đồng thời hiển thị quá trình phân loại lên màn hình giám sát.

Từ kết luận rút ra trong phần 2.1.3 về lợi ích của việc học sâu với mô hình CNN cụ thể là mạng VGG16 với thời gian chạy nhanh hơn và độ chính xác cao và tự động trích xuất các tính năng hữu ích, khoá luận đề xuất phương hướng giải quyết bài toán nhận dạng hoa quả như sau:

1. Cài đặt mạng VGG16.
2. Xây dựng cơ sở dữ liệu ảnh đào tạo cho 3 lớp phân loại đối tượng với ảnh được chọn lựa theo tiêu chuẩn về kích thước, màu sắc cũng như độ rõ nét, đồng thời được gán nhãn cẩn thận.
3. Tăng dữ liệu cho bộ ảnh huấn luyện tránh tình trạng overfitting và cải thiện độ mạnh mẽ của mô hình.
4. Thử nghiệm với mô hình dự đoán để giải quyết và cải thiện lại mô hình.

2.3. Thiết bị phần cứng

2.3.1. Raspberry Pi 4



Hình 2.10 Raspberry Pi 4

Raspberry Pi 4 là một máy tính nhúng kích thước chỉ bằng một cái thẻ ATM. Nó cung cấp sự gia tăng đột phá về tốc độ bộ xử lý, đa phương tiện hiệu suất, bộ nhớ và kết nối so với các thế hệ trước trong khi vẫn giữ được khả năng tương thích ngược và tương tự sự tiêu thụ năng lượng. Đối với người dùng cuối, Raspberry Pi 4 Model B cung cấp hiệu suất tương đương có thể so sánh với các hệ thống PC x86.

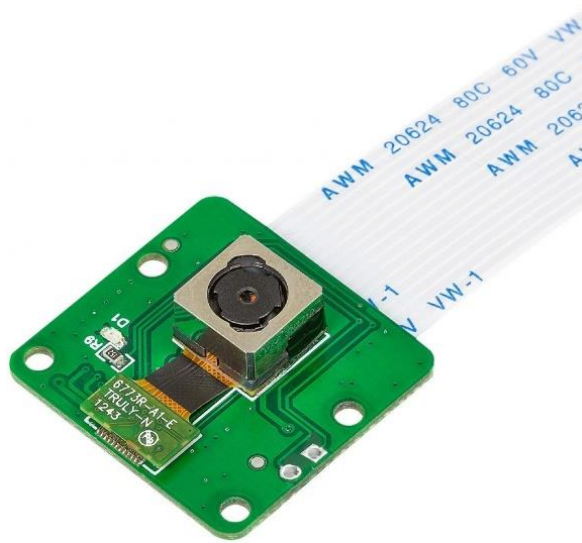
Thông số cấu hình:

- *Vi xử lý:* Broadcom BCM2711 - ARM Cortex A72 (SoC 64-bit 1.5GHz).
- *Ram:* 1GB, 2GB, 4GB hoặc 8GB LPDDR4 (tùy thuộc).
- *Hỗ trợ kết nối:* x2 USB 3.0, x2 USB 2.0, Wireless Lan 802.11ac - (2.4 / 5.0 GHz), LAN, Bluetooth 5.0, GPIO 40 chân tiêu chuẩn.
- *Xuất Video:* 2 x cổng micro HDMI (hỗ trợ 4Kp60)
- *Camera Port:* 2-lane MIPI CSI.
- *Bộ nhớ:* 1x Micro SD Card.
- *Nguồn đầu vào:* 5V DC qua cổng USB-C hoặc 5V DC qua đầu cắm GPIO (3A)

Ưu điểm: Giá rẻ, nhỏ gọn, phục vụ cho nhiều mục đích, Khả năng hoạt động liên tục, giá thành rẻ hơn so với máy tính thông thường, ...

Nhược điểm: Máy tính nhúng hiệu năng chưa cao, chưa thể so với PC thông thường. Phiên bản được dùng trong khoá luận này là Raspberry Pi 4 Model B với 1Gb Ram.

2.3.2. Camera Pi



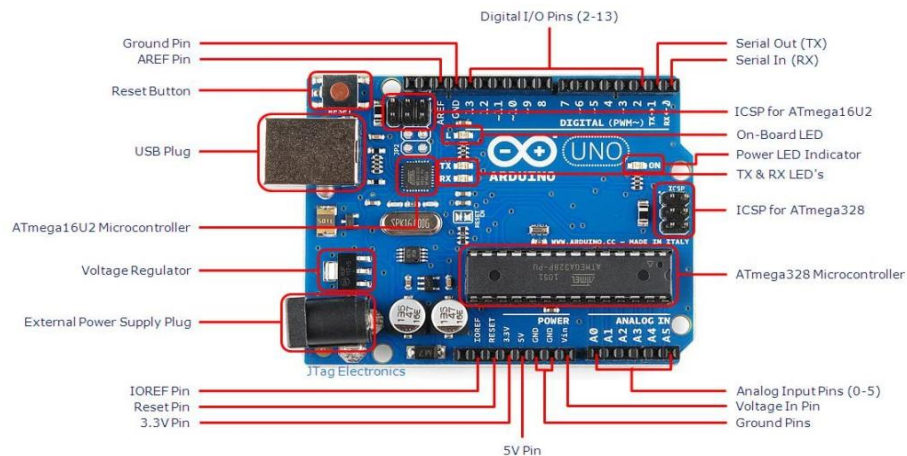
Hình 2.11 Camera Pi v2

Raspberry Pi Camera Module V2 có cảm biến Sony IMX219 8 megapixel. Camera Module có thể được sử dụng để quay video độ nét cao cũng như chụp ảnh tĩnh. Nó khá dễ dàng để sử dụng cho người mới bắt đầu, nhưng cũng có rất nhiều giải pháp mở rộng để cung cấp cho người dùng yêu cầu cao. Kết nối với Raspberry Pi thông qua cáp ribbon đi kèm dài 15 cm. Một số công dụng của Camera Module như chụp Time-Lapse, Slow-Motion và rất nhiều ứng dụng khác. [6]

Thông số kỹ thuật:

- Ống kính tiêu cự cố định.
- Cảm biến độ phân giải 8 megapixel cho khả năng chụp ảnh kích thước 3280 x 2464.
- Hỗ trợ video 1080p30, 720p60 và 640x480p90.
- Kích thước 25mm x 23mm x 9mm.
- Trọng lượng 3g.
- Kết nối với Raspberry Pi thông qua cáp ribbon đi kèm dài 15 cm.
- Camera Module được hỗ trợ với phiên bản mới nhất của Raspbian.

2.3.3. Arduino Uno



Hình 2.12 Arduino Uno

Thông số:

Vi điều khiển	Atmega328 8Bit
Điện áp hoạt động	5V DC (qua cổng USB)
Tần số hoạt động	16MHz
Dòng tiêu thụ	30mA
Điện áp khuyến dùng	7-12V DC
Điện áp vào giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân PWM)
Số chân Analog	6 (độ phân giải 10bit)
Dòng ra tối đa(5V)	500mA
Dòng ra tối đa(3.3V)	50mA
Bộ nhớ flash	32KB (Atmega328) với 0.5KB dùng bởi bootloader
SRAM	2KB(Atmrga328)
EEPROM	1KB(Atmega328)

Bảng 2.1 Thông số Arduini Uno

Arduino Uno là một board mạch vi điều khiển dựa trên ATmega328P. Với bộ vi điều khiển này, chúng ta có tổng cộng 14 chân (đầu vào) vào / ra được đánh số từ 0 đến 13 (trong đó có 6 chân PWM, được đánh dấu ~ trước mã pin). Song song đó, chúng ta có thêm 6 pin để nhận tín hiệu analog có ký hiệu là A0 - A5, 6 pin này cũng có thể sử dụng như các pin vào / ra bình thường (như pin 0 - 13). Trên bo mạch còn có nút reset, đầu kết nối với máy tính qua cổng USB và cổng cấp nguồn sử dụng jack 2.1mm để lấy điện trực tiếp từ bộ chuyển AC-DC hoặc qua nguồn pin. Ngoài ra board Arduino còn cung cấp các dòng ra khác nhau như 3.3V, 5V, nối đất GND ...

2.4. Phần Mềm

2.4.1. Google Colab

Với A.I. DeepLearning, Google không chỉ có những sản phẩm dự án đã và đang phát triển. Google còn có công cụ giúp cho các developer phát triển, train model, một cách dễ dàng nhất. Đó là Google Colab.

Colaboratory hay còn gọi là 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, bạn có thể sử dụng tài nguyên máy tính từ CPU tốc độ cao và cả GPUs và cả TPUs đều được cung cấp cho bạn.

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. Vì là dịch vụ miễn phí, nên Colab sẽ có những thứ tự ưu tiên trong việc sử dụng tài nguyên hệ thống, cũng như giới hạn thời gian sử dụng, thời gian sử dụng tối đa lên tới 12 giờ.

Cấu hình phần cứng Google Colab cung cấp:

CPU	GPU	TPU
Intel Xeon Processor with two Cores @2.3 Ghz and 13GB Ram	Up to Tesla T4 with 16GB Of GDDR6 VRAM, Intel Xeon Processor with two Cores @2.2 Ghz and 13GB Ram	Cloud TPU with 180 Teraflops of Computation, Intel Xeon Processor with two Cores @2.3 Ghz and 13GB Ram

Bảng 2.2 Cấu hình Google Colab

Một số ưu nhược điểm:

- Google Colab miễn phí.
- Cung cấp phần cứng mạnh cho việc chạy thử code từ CPU, GPU đến TPU.
- Dễ dàng sử dụng và truy cập, kết nối lưu trữ trực tiếp trên Google Drive, Github.
- Sử dụng được nhiều thư viện phổ biến dành cho DeepLearning như Keras, TensorFlow, PyTorch, và OpenCV.
- Thời gian sử dụng có giới hạn, sau 12 giờ VM của colab bị reset.
- Không thể tự lựa chọn cấu hình phần cứng, với tài khoản miễn phí.

2.4.2. Raspberry Pi OS

Raspberry Pi OS là hệ điều hành được khuyến nghị để sử dụng bình thường trên Raspberry Pi. Raspberry Pi OS là hệ điều hành miễn phí dựa trên Debian, được tối ưu hóa cho phần cứng Raspberry Pi. Hệ điều hành Raspberry Pi đi kèm với hơn 35.000 gói: phần mềm được biên dịch sẵn đi kèm với định dạng để dễ dàng cài đặt trên Raspberry Pi [17]



Hình 2.13 Raspberry Pi OS

Raspbian được hướng đến người dùng có mục đích:

- Phát triển các thiết bị điều khiển tự động.
- Sử dụng như một máy chủ cung cấp các dịch vụ như web, file server, printer server, ...
- Sử dụng Raspberry Pi như máy tính văn phòng để lướt web, soạn văn bản, check mail và thi thoảng nghe nhạc/xem phim.

2.4.3. Open CV [13]

OpenCV (Open Source Computer Vision) là một thư viện mã nguồn mở. Thư viện được lập trình trên các ngôn ngữ: C++, C, Python, hay Java và hỗ trợ trên các nền tảng Window, Linux, Mac OS, iOS và Android. OpenCV có cấu trúc mô-đun, có nghĩa là gói bao gồm một số thư viện được chia sẻ hoặc thư viện tĩnh.

Trong khoá luận này OpenCV chạy trên Python và được sử dụng để quay video và codec video lên màn hình giám sát.

2.4.4. Flask Python framework

Flask là một micro web frameworks được xây dựng bằng ngôn ngữ lập trình Python. Flask có kiến trúc nhỏ, gọn nên thuận tiện khi cấu hình hay tổ chức ứng dụng. Mặt khác, Flask còn có ưu điểm: dễ tìm hiểu, sử dụng linh hoạt, định tuyến dễ dàng, dễ mở rộng. Flask có thể xây dựng các ứng dụng web từ đơn giản tới phức tạp [9]. Nó có thể xây dựng các ứng dụng web, các API hay một website dựa theo thời gian thực, blog...

Flask được sử dụng trong khoá luận để xây dựng một giao diện web và xử lý HTTP request, response dữ liệu theo thời gian thực cho mục đích giám sát phân loại.

2.4.5. Các thư viện liên quan

2.4.5.1. Tensorflow

TensorFlow là một nền tảng mã nguồn mở dành cho học máy. Nó có một hệ sinh thái toàn diện, linh hoạt gồm các công cụ, thư viện và tài nguyên cộng đồng cho phép các nhà nghiên cứu thúc đẩy các tính năng tiên tiến trong học máy và các nhà phát triển dễ dàng xây dựng và triển khai các ứng dụng hỗ trợ học máy. Tensorflow giúp dễ dàng xây dựng và đào tạo các mô hình học máy bằng cách sử dụng các API cấp cao trực quan như Keras với khả năng thực thi nhanh chóng, giúp lặp lại mô hình ngay lập tức và gỡ lỗi dễ dàng.

Kiến trúc Tensorflow hoạt động chia làm 3 phần:

- Tiền xử lý dữ liệu
- Xây dựng hình
- Đào tạo và mô hình hoá.

2.4.5.2. Keras

Keras là một thư viện mã nguồn mở cung cấp giao diện Python cho các mạng nơ-ron nhân tạo. Keras hoạt động như một giao diện cho thư viện TensorFlow. Nó là một API bậc cao có thể sử dụng chung với các thư viện Deep Learning nổi tiếng như Tensorflow, CNTK, Theano. Từ phiên bản Keras 2.4 chỉ Tensorflow được hỗ trợ.

Keras có một số ưu điểm như:

- Dễ sử dụng, xây dựng model nhanh.
- Có thể chạy trên cả CPU và GPU

Keras cho phép người dùng xuất các mô hình trên các thiết bị IOT, Mobile hoặc Website. Nó cũng cho phép sử dụng đào tạo phân tán các mô hình học sâu trên các cụm đơn vị xử lý đồ họa (CPU) và đơn vị xử lý tensor (TPU).

2.4.5.3. Tensorflow Lite

TensorFlow Lite là một phiên bản thu nhỏ của Tensorflow cho phát triển khai các mô hình học máy trên các thiết bị di động, nhúng và IoT.

Các tính năng chính:

- Được tối ưu hóa cho việc học máy trên thiết bị, bằng cách giải quyết 5 hạn chế chính: độ trễ, riêng tư (không có dữ liệu cá nhân nào rời khỏi thiết bị), kết nối (không cần kết nối internet), kích thước (giảm kích thước mô hình) và tiêu thụ năng lượng (suy luận hiệu quả và không cần kết nối mạng).
- Hỗ trợ nhiều nền tảng, bao gồm các thiết bị Linux nhúng, vi điều khiển, mobile (Android, IOS)
- Hỗ trợ đa dạng ngôn ngữ bao gồm Java, Swift, Objective-C, C ++ và Python.
- Hiệu suất cao, với khả năng tăng tốc phần cứng và tối ưu hoá mô hình.
- Hỗ trợ cho các tác vụ học máy phổ biến như phân loại hình ảnh, phát hiện đối tượng, phân loại văn bản, ... trên nhiều nền tảng.

2.4.5.4. Numpy

Numpy (viết tắt của Numeric Python), là một thư viện toán học rất mạnh mẽ và là một trong những thư viện về toán học phổ biến nhất cộng đồng sử dụng Python. Numpy được trang bị các phép toán, hàm số toán học đã được tối ưu, hỗ trợ làm việc đạt hiệu quả cao nhất với ma trận hay mảng, đặc biệt với sự tối ưu cho tính toán đối với ma trận, điều đó giúp tính toán trên Numpy cho ma trận hiệu quả hơn khi chỉ dùng Python cơ bản.

Numpy có thể thực hiện những thao tác sau ở mức tốt nhất:

- Các phép toán, phép logic trên ma trận nhiều chiều hay mảng.
- Các quá trình để thao tác với shape, các phép biến đổi – chuẩn hóa Fourier.
- Các phép toán liên quan đến các hàm tạo số ngẫu nhiên, các phép toán đại số tuyến tính.

2.4.5.5. Scikit-learn

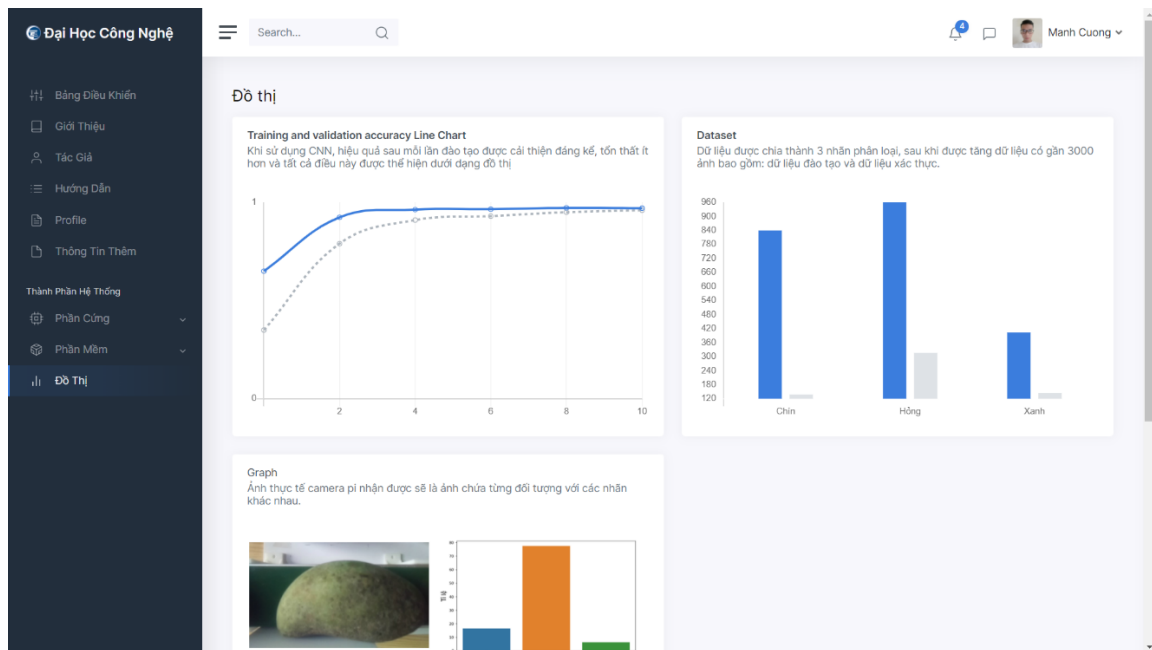
Scikit-learn là một trong những thư viện học máy mạnh mẽ nhất, bao gồm tất cả các thuật toán học máy từ cổ điển đến hiện đại. Scikitlearn tích hợp tốt với nhiều thư viện Python khác liên quan đến học máy, tính toán, thống kê như Matplotlib hỗ trợ xây dựng biểu đồ, cùng với Numpy nhằm hỗ trợ vector hóa các mảng hay ma trận, hỗ trợ Pandas dataframe, các tính toán cùng Scipy, ...

Thư viện cung cấp đầy đủ một tập các công cụ xử lý cho các bài toán Học máy và Thống kê, bao gồm: Phân loại (classification), Hồi quy (regression), Phân cụm (clustering) và Giảm chiều dữ liệu (dimensionality reduction).

Trong khoá luận này Scikit-learn được sử dụng để tích hợp với Numpy nhằm hỗ trợ vector hoá các mảng hay ma trận và Matplotlib hỗ trợ xây dựng biểu đồ trực quan hoá dữ liệu.

2.5. Website

Sau khi đào tạo mô hình, máy tính nhúng có thể phân biệt được các nhãn phân loại của đối tượng. Trên băng tải, một camera được chỉ định để giúp chụp ảnh đối tượng khi di chuyển tới khu vực chứa camera. Tất cả các hình ảnh chụp được sẽ được lưu vào một thư mục và hình ảnh phân loại đối tượng được gửi tới màn hình phân loại. Dữ liệu được đồng bộ hóa trên tất cả các ứng dụng khách trong thời gian thực. Ngoài ra, người dùng có thể tương tác với cơ sở dữ liệu thời gian thực.



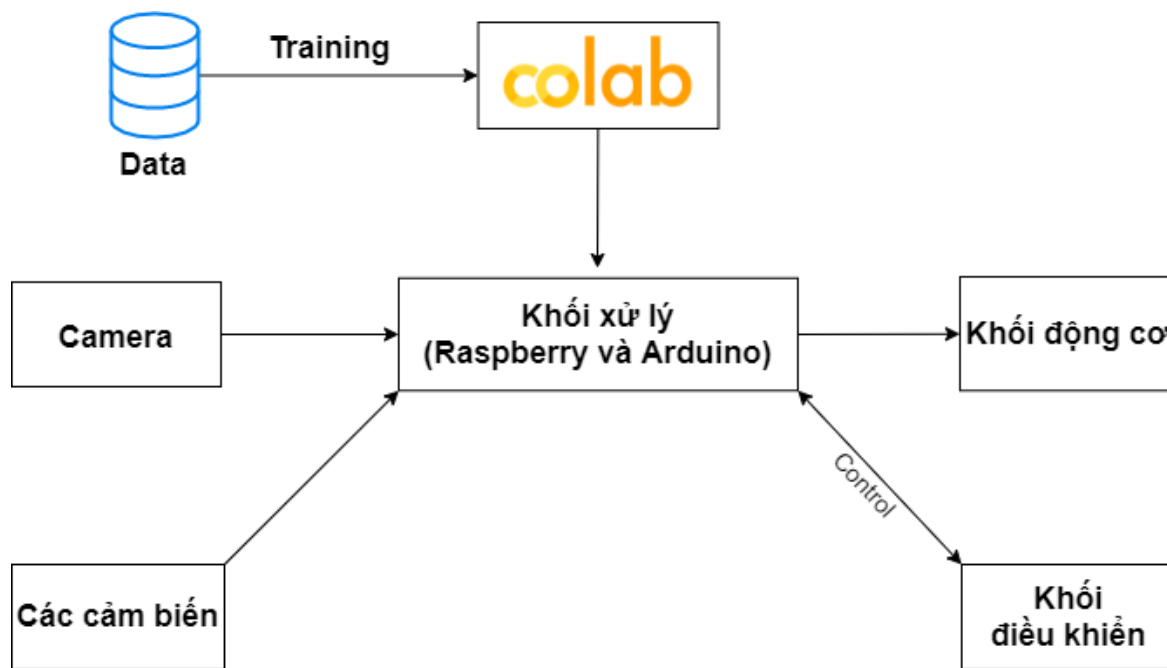
Hình 2.14 Giao diện website

Để có thể hiển thị trực quan tất cả dữ liệu sau khi đã được xử lý cần một nơi để quản lý và giám sát dữ liệu được gửi từ Raspberry Pi đó là sử dụng website, một cách phổ biến và dễ tiếp cận hiện nay. Tất cả đều hướng tới sự tiện lợi và thông minh, để có thể biến phương pháp của mình thành hiện thực thì việc phát triển website là điều cần thiết. Website nhằm hiển thị trực quan quá trình phân loại và giúp việc quản lý dễ dàng hơn. Ngoài ra, có thể điều khiển bằng tải ở một mức độ theo thời gian thực từ website, khi đó website sẽ gửi yêu cầu tới Raspberry Pi sau đó được truyền lệnh điều khiển tới motor DC. Website cũng được phát triển với các tính năng phù hợp, tiện lợi để người dùng tiết kiệm thời gian và đạt hiệu quả cao. Ngôn ngữ để phát triển website là JavaScript, một trong những ngôn ngữ phát triển web phổ biến và mạnh mẽ nhất hiện nay.

CHƯƠNG 3: XÂY DỰNG, THIẾT KẾ HỆ THỐNG VÀ THỰC NGHIỆM

3.1. Xây dựng và thiết kế hệ thống

3.1.1 Cấu trúc hệ thống



Hình 3.1 Cấu trúc hệ thống

Đây là cấu trúc của hệ thống được đề xuất trong khoá luận. Bộ dữ liệu được gán nhãn kết hợp với CNN để đào tạo trên Google Colab với mục đích để phân loại các đặc điểm nông sản khác nhau. Một camera Pi được sử dụng kết hợp với máy tính nhúng Raspberry Pi truyền hình ảnh của từng đối tượng đi qua trên băng tải và dùng mô hình đã được đào tạo trước đó để phân loại, dữ liệu sau đó được gửi về cho arduino điều khiển servo chuyển hướng phân loại ứng với từng nhãn khác nhau. Đi cùng đó là việc hiển thị quá trình phân loại với số lượng đối tượng đã phân loại, cũng như điều khiển hệ thống trên khối điều khiển.

3.1.2. Mô hình phần cứng

Dựa trên phần cứng đã được thiết kế sẵn. Việc phân loại hoa quả được thực hiện thông qua servo điều khiển qua Arduino nhận lệnh từ máy tính nhúng. Dây chuyền phân loại được thiết kế phù hợp với các tính năng đã đề ra và các thành phần được lựa chọn phải phù hợp với kích thước, khả năng tải của đối tượng phân loại và hoạt động ổn định.

Vì vậy việc lựa chọn băng tải phù hợp là điều tất yếu.

Động cơ hoạt động với trục quay của băng truyền cùng với khả năng tải được trọng lượng của vật thể. Tốc độ phù hợp để băng truyền tải có thể dừng được băng tải vào vị trí và góc chụp Camera.

Với chiều dài của băng tải là 50cm thì cần tốc độ băng tải chậm để đảm bảo được Camera có thể lấy được hình ảnh thì tốc độ phù hợp là 15 vòng/phút. Với tỉ lệ trục của động cơ và trục quay băng tải là 1:3 thì tốc độ phù hợp để có thể tải được vật thể chạy trên băng tải là 45vòng/phút.

Sau khi thông tin xác định đối tượng được máy tính nhúng gửi về Arduino. Vì điều khiển này sẽ gửi tín hiệu điều khiển tới Servo. Tín hiệu điều khiển đại diện cho vị trí đầu ra mong muốn của đối tượng. Khi đối tượng đi qua cảm biến vật cản, Servo tiến hành quay tới vị trí tương ứng với nhãn phân loại của đối tượng.

Servo được điều khiển bằng cách gửi một xung điện thay đổi, hoặc điều chế độ rộng xung (PWM), thông qua dây điều khiển. Có một xung tối thiểu, một xung tối đa và tốc độ lặp lại. Một động cơ servo có thể quay 90° theo một trong hai hướng với tổng chuyển động là 180° . Nhưng động cơ servo không quay liên tục. Vòng quay của chúng bị giới hạn ở giữa các góc cố định.

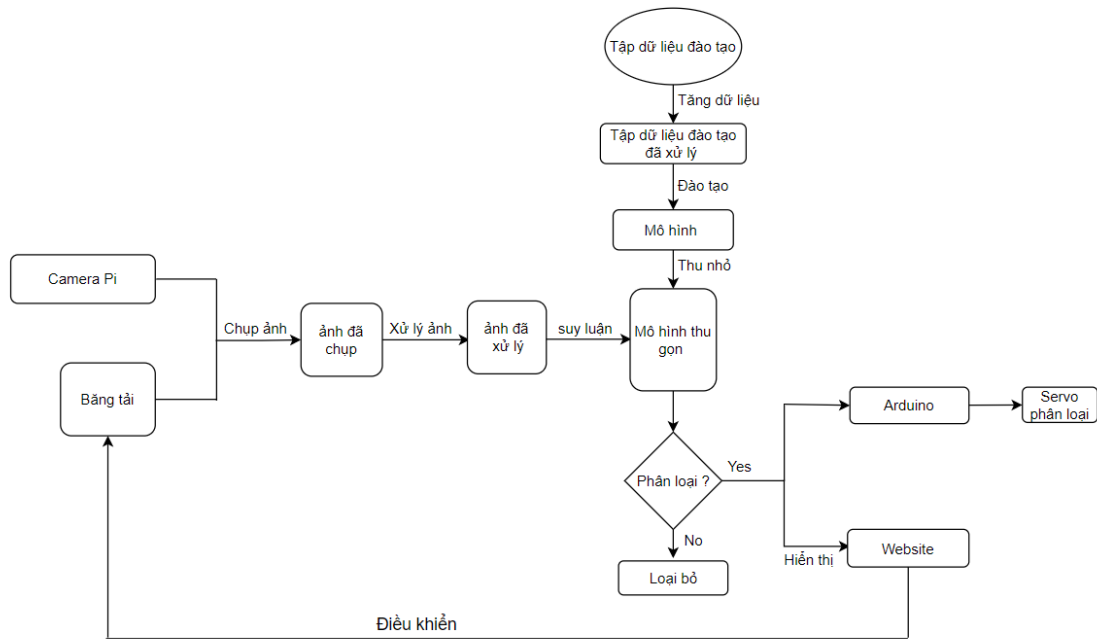
Khi được lệnh di chuyển, vòng quay của servo sẽ di chuyển đến vị trí và giữ vị trí đó. Nếu một lực bên ngoài đẩy vào servo trong khi servo đang giữ một vị trí, thì servo sẽ chống lại việc di chuyển ra khỏi vị trí đó. Lượng lực tối đa mà servo có thể tác động được gọi là định mức mô-men xoắn của servo. Servo sẽ không giữ vị trí của nó mãi mãi, xung vị trí phải được lặp lại để hướng dẫn servo giữ nguyên vị trí.



Hình 3.2 Hệ thống băng tải

3.2. Thực nghiệm

3.2.1. Thiết lập đào tạo và thử nghiệm



Hình 3.3 Lưu đồ thuật toán đào tạo

Trước khi đào tạo với dữ liệu thô, tất cả hình ảnh đều có nền và độ sáng giống nhau. Độ chính xác của phương pháp có thể nhận được giá trị cao nhưng ảnh kiểm tra cho giá trị thấp vì độ sáng của ảnh kiểm tra khác với độ sáng của dữ liệu đào tạo. Đồng thời phương pháp đề xuất để đào tạo mô hình yêu cầu một bộ dữ liệu được chia thành các lớp đào tạo để máy tính nhúng có thể phân biệt được các hình thái khác nhau của quả.

Để đáp ứng yêu cầu cho mỗi lớp phân loại cần khoảng 1000 bức ảnh để đào tạo, và tránh lặp quá nhiều trong quá trình đào tạo cần phải tăng lượng dữ liệu đầu vào. Phương pháp được sử dụng là tăng dữ liệu [7]. Tăng dữ liệu là sự gia tăng quy mô và đa dạng của dữ liệu hiện có mà không yêu cầu thu thập thủ công bất kỳ dữ liệu mới nào. Dữ liệu gia tăng này có được bằng cách thực hiện một loạt các phép biến đổi tiền xử lý thành dữ liệu hiện có, các phép biến đổi có thể bao gồm lật ngang, dọc, xiên, cắt, xoay, thay đổi độ sáng và nhiều hơn thế trong dữ liệu ảnh. Dữ liệu nâng cao này có thể mô phỏng nhiều điểm dữ liệu khác nhau, trái ngược với việc chỉ sao chép cùng một dữ liệu. Sự khác biệt tinh tế của những hình ảnh bổ sung này sẽ đủ để giúp đào tạo một mô hình mạnh mẽ hơn.

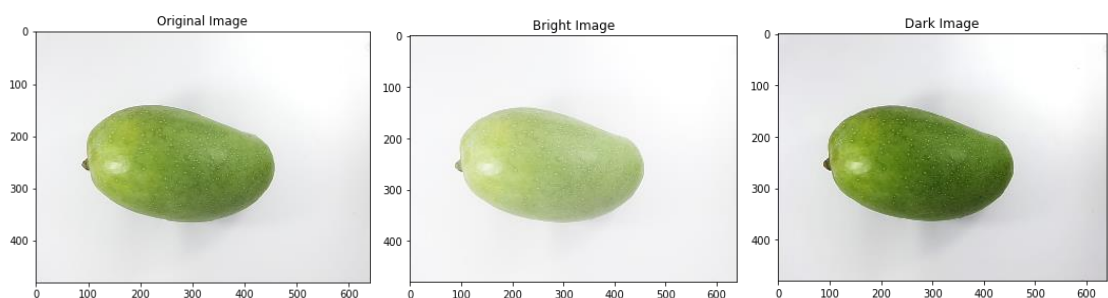
Phương pháp này có thể sử dụng để thay đổi độ sáng của hình ảnh kiểm tra cũng như tạo ra một lượng dữ liệu đào tạo mới với những thay đổi về độ sáng. Với hình ảnh

có độ sáng khác nhau có thể được sử dụng để làm cho mô hình xử lý ảnh hoạt động tốt hơn trước những thay đổi trong điều kiện ánh sáng.



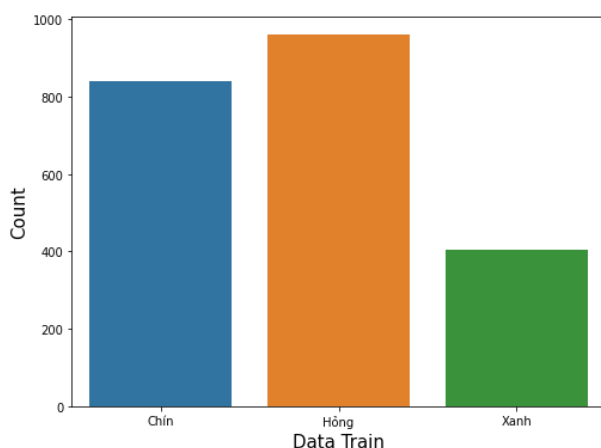
Hình 3.4 Tăng dữ liệu

Với lượng dữ liệu hình ảnh đầu vào cho đào tạo được chụp với cùng độ sáng. Nếu ảnh chụp trực tiếp để đào tạo thì việc kiểm tra bằng máy tính nhúng sẽ cho kết quả không mong muốn. Nguyên nhân của kết quả sai lệch này là do ảnh chụp bằng máy ảnh có góc độ, điều kiện ánh sáng khác với ảnh dùng để đào tạo. Các bức ảnh không được xử lý trước thì máy tính nhúng sẽ căn cứ vào mô hình đã được đào tạo để phân tích các đặc điểm của ảnh đó. Giải pháp là thay đổi độ sáng của ảnh trước khi đưa vào xử lý. Phương pháp dựa trên việc tăng dữ liệu có thể thay đổi độ sáng của ảnh.

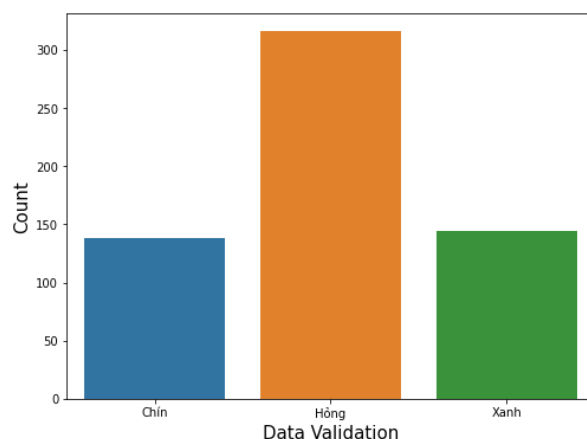


Hình 3.5 Tăng dữ liệu (thay đổi độ sáng)

Dữ liệu được chia thành 3 nhãn phân loại, sau khi được tăng dữ liệu có gần 3000 ảnh bao gồm: dữ liệu đào tạo và dữ liệu xác thực.



Hình 3.6 Thống kê dữ liệu đào tạo

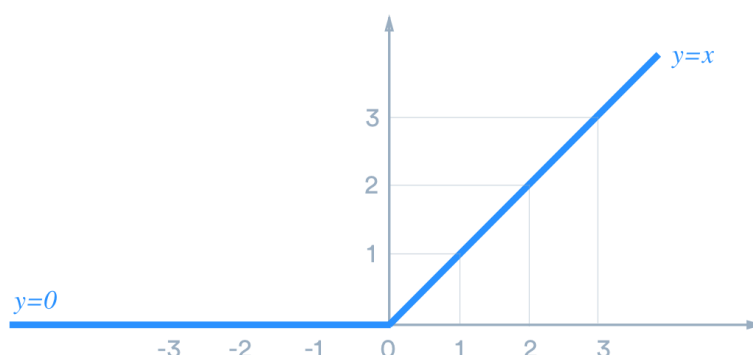


Hình 3.7 Thống kê dữ liệu xác thực

Sau khi có được tập dữ liệu mới đủ lớn, kích thước của những bức ảnh này đã được điều chỉnh để giúp việc đào tạo diễn ra nhanh chóng và tiết kiệm thời gian.

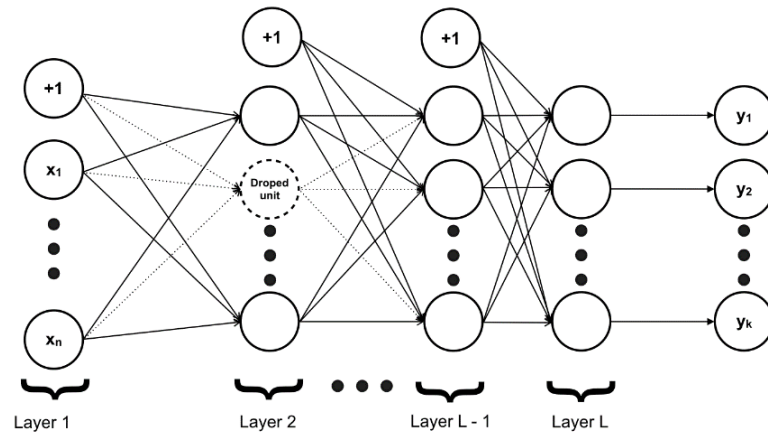
Kết quả của lớp trước khi trở thành đầu vào của lớp sau được chuyển qua một hàm kích hoạt và hàm được sử dụng trong phương thức này là ReLu.

Mục đích của một hàm kích hoạt là thêm một số loại thuộc tính phi tuyến tính vào hàm, đó là một mạng nơ-ron. Nếu không có các chức năng kích hoạt, mạng nơ-ron chỉ có thể thực hiện các ánh xạ tuyến tính từ đầu vào x đến đầu ra y . Nếu không có các hàm kích hoạt, phép toán duy nhất trong quá trình truyền về phía trước sẽ là các tích số giữa một vectơ đầu vào và một ma trận trọng số. Và các phép toán tuyến tính liên tiếp có thể được coi là một phép toán học duy nhất. Một mạng nơ-ron không có bất kỳ chức năng kích hoạt nào sẽ không thể nhận ra các ánh xạ phức tạp như vậy về mặt toán học và sẽ không thể giải quyết các nhiệm vụ mà chúng ta muốn mạng giải quyết.



Hình 3.8 Hàm kích hoạt Relu

Trong lớp kết nối đầy đủ, là sự kết hợp của nhiều nơ-ron, các nơ-ron sẽ phụ thuộc "mạnh mẽ" vào nhau trong quá trình đào tạo, điều này dẫn đến việc đào tạo mô hình quá khớp (overfitting). Để tránh điều này, một dropout được sử dụng để "phớt lờ" một số đơn vị sẽ không tham gia việc đào tạo. Việc dropout buộc các mạng nơ-ron phải tìm ra các tính năng mạnh mẽ hơn, với các đặc điểm là chúng phải hữu ích hơn, tốt hơn khi kết hợp với nhiều nơ-ron khác. Dropout đòi hỏi sự đào tạo gấp đôi để đạt được sự hội tụ. Tuy nhiên, thời gian đào tạo cho mỗi epoch sẽ ít hơn.



Hình 3.9 Dropout

Trong lớp đầu ra, hàm kích hoạt được sử dụng là hàm softmax. Softmax là một cách liên kết các đầu ra của mạng nơ-ron với tổng bằng 1. Qua đó, các giá trị đầu ra của hàm softmax có thể được coi là phân phối xác suất của các biến đầu ra. Nó rất hữu ích trong vấn đề phân loại nhiều lớp. Softmax là một dạng hàm kích hoạt với điều đặc biệt là đầu ra của nó sẽ bằng 1. Để làm được điều này, hàm softmax sẽ quy đổi giá trị, đầu ra của mạng nơ-ron chia cho tổng giá trị. Đầu ra bây giờ có thể được coi là một véc tơ xác suất dự đoán của các lớp.

$$\text{softmax}_i(a) = \frac{e^{a_i}}{\sum e^{a_i}} \quad (2)$$


```

46/46 [=====] - 34s 730ms/step - loss: 0.3097 - accuracy: 0.8815 - val_loss: 0.2092 - val_accuracy: 0.9343
Epoch 00003: loss improved from 0.59030 to 0.30929, saving model to model_best_weights.h5
Epoch 4/50
46/46 [=====] - 34s 731ms/step - loss: 0.2209 - accuracy: 0.9206 - val_loss: 0.2981 - val_accuracy: 0.8801

Epoch 00004: loss improved from 0.30929 to 0.22066, saving model to model_best_weights.h5
Epoch 5/50
46/46 [=====] - 34s 729ms/step - loss: 0.2078 - accuracy: 0.9196 - val_loss: 0.0850 - val_accuracy: 0.9646

Epoch 00005: loss improved from 0.22066 to 0.20827, saving model to model_best_weights.h5
Epoch 6/50
46/46 [=====] - 33s 726ms/step - loss: 0.1331 - accuracy: 0.9523 - val_loss: 0.0623 - val_accuracy: 0.9775

Epoch 00006: loss improved from 0.20827 to 0.13328, saving model to model_best_weights.h5
Epoch 7/50
46/46 [=====] - 33s 725ms/step - loss: 0.1116 - accuracy: 0.9574 - val_loss: 0.0364 - val_accuracy: 0.9877

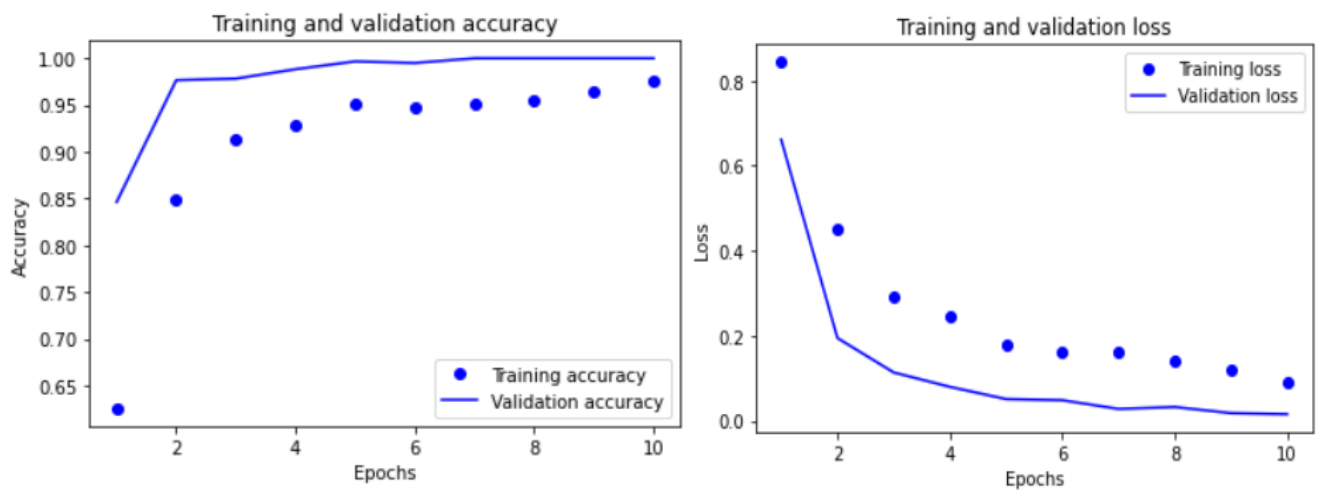
Epoch 00007: loss improved from 0.13328 to 0.11160, saving model to model_best_weights.h5
Epoch 8/50
46/46 [=====] - 34s 729ms/step - loss: 0.0667 - accuracy: 0.9765 - val_loss: 0.0451 - val_accuracy: 0.9833

Epoch 00008: loss improved from 0.11160 to 0.06667, saving model to model_best_weights.h5
Epoch 9/50
46/46 [=====] - 33s 727ms/step - loss: 0.0605 - accuracy: 0.9806 - val_loss: 0.1505 - val_accuracy: 0.9343

```

Hình 3.10 Quá trình đào tạo

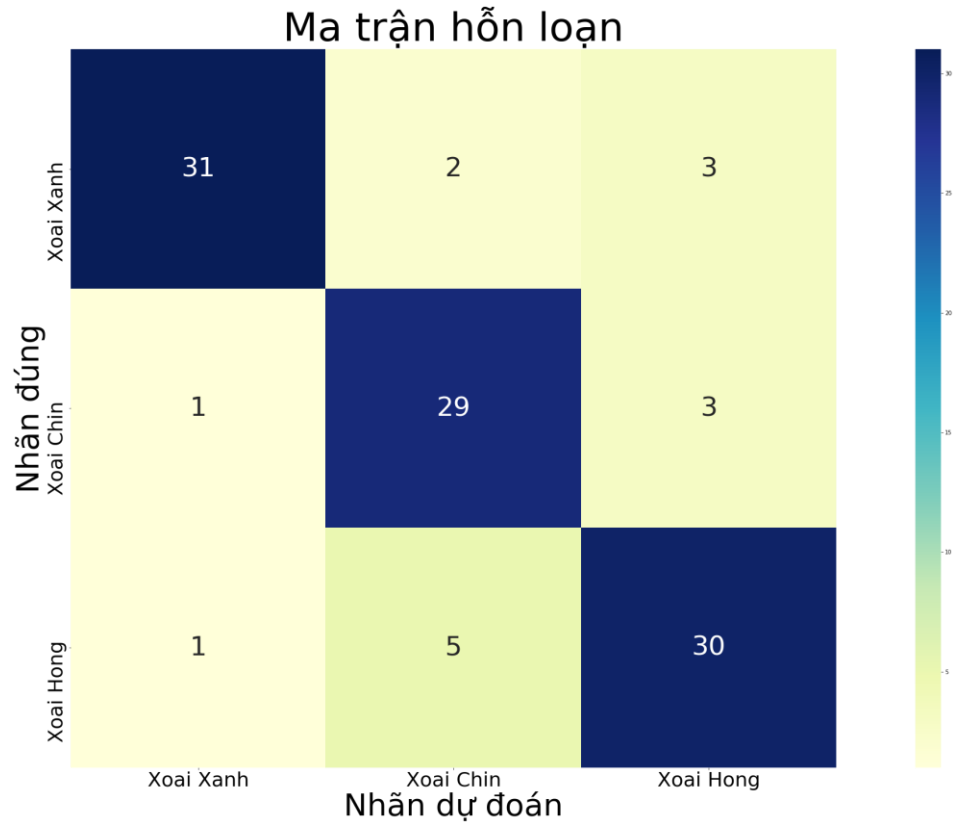
Khi sử dụng CNN, hiệu quả sau mỗi lần đào tạo được cải thiện đáng kể, tổn thất ít hơn và tất cả điều này được thể hiện dưới dạng đồ thị.



Hình 3.11 Đồ thị Accuracy and Loss

3.2.2. Đánh giá mô hình

Sau khi triển khai mô hình đào tạo, thông qua bước đánh giá/ thực nghiệm được kết quả như sau:



Hình 3.12 Kết quả đánh giá mô hình

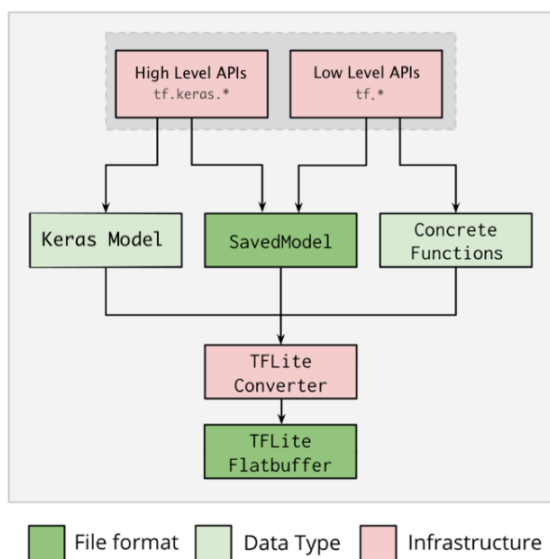
Dựa vào ma trận hỗn loạn thu được từ việc chạy mô hình với tập kiểm thử, có thể thấy kết quả thu được tương đối tốt. Quá trình đào tạo trên Colab có tốc độ xử lý nhanh và cho kết quả chính xác cao dao động trên 86%.

Ma trận hỗn loạn là một biểu diễn trực quan về hiệu quả của một thuật toán trong học máy hoặc phân loại thống kê. Các hàng đại diện cho các trường hợp được dự đoán và các cột đại diện cho các trường hợp thực tế.

3.2.3. Thiết lập dự đoán cho Raspberry Pi

Các thiết bị IOT và mobile có phần cứng hạn chế để triển khai các mô hình vì nó quá lớn. Ta cần một kỹ thuật giúp giảm nhẹ kích thước các mô hình nhiều lần, đồng thời giảm độ trễ (latency) và tăng tốc độ (inference) khi triển khai.

Sau khi hoàn thành đào tạo, để sử dụng mô hình trên máy tính nhúng Raspberry Pi ta cần phải chuyển đổi từ định dạng .h5 thành .tflite nhằm mục đích giảm kích thước mô hình. Tflite là một định dạng file của Tensorflow Lite [19]



Hình 3.13 Mô hình chuyển đổi Tensorflow Lite

TensorFlow Lite là một giải pháp nhẹ của TensorFlow dành cho thiết bị di động và thiết bị nhúng. Nó cho phép chạy các mô hình TensorFlow với độ trễ thấp và nhanh chóng.

Trước khi nạp tệp dữ liệu .tflite vào Raspberry Pi ta cần kiểm tra độ chính xác của mô hình.

```

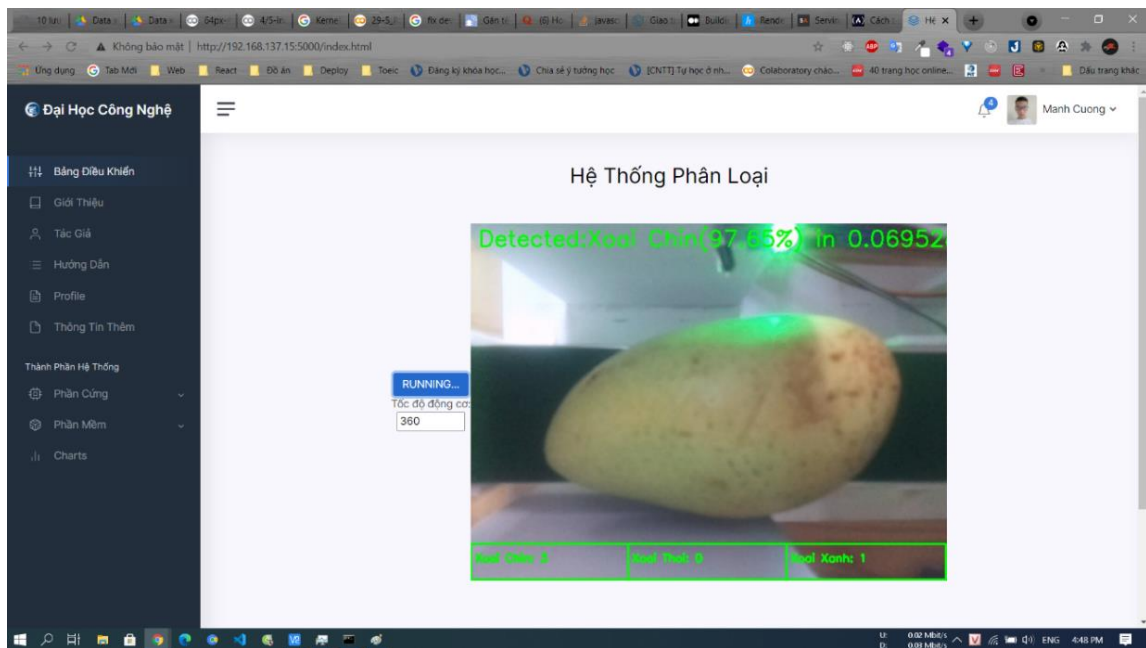
input image: /content/drive/MyDrive/Khoá luận/Data xoai full)/Train/Xoai Chin/c (433).jpg
[1.9920699e-04 9.9979001e-01 1.0849164e-05]
--- 6.963621616363525 seconds ---
(0.0, 1.0, 0.0)
input image: /content/drive/MyDrive/Khoá luận/Data xoai full)/Train/Xoai Chin/c_1000.jpg
[0.9236015 0.01077149 0.06562699]
--- 0.008504629135131836 seconds ---
(0.92, 0.01, 0.07)
input image: /content/drive/MyDrive/Khoá luận/Data xoai full)/Train/Xoai Chin/c_999.jpg
[0.7670429 0.02396664 0.2089905 ]
--- 0.0076596736907958984 seconds ---
(0.77, 0.02, 0.21)
input image: /content/drive/MyDrive/Khoá luận/Data xoai full)/Train/Xoai Thoi/t_0.jpg
[0.01309558 0.77803195 0.2088725 ]
--- 0.0074920654296875 seconds ---
(0.01, 0.78, 0.21)
input image: /content/drive/MyDrive/Khoá luận/Data xoai full)/Train/Xoai Xanh/x_1000.jpg
[6.3897547e-04 3.7252116e-03 9.9563581e-01]
--- 0.008002758026123047 seconds ---
(0.0, 0.0, 1.0)
input image: /content/drive/MyDrive/Khoá luận/Data xoai full)/Train/Xoai Xanh/x_999.jpg
[2.2781765e-04 6.4466791e-03 9.9332547e-01]
--- 0.0076446533203125 seconds ---
(0.0, 0.01, 0.99)
input image: /content/drive/MyDrive/Khoá luận/Data xoai full)/Train/Xoai Xanh/x_998.jpg
[3.6427434e-04 3.6937373e-03 9.9594194e-01]
--- 0.007811784744262695 seconds ---
(0.0, 0.0, 1.0)

```

Hình 3.14 Kiểm tra sau khi đào tạo

Sau khi thực hiện chuyển đổi chúng ta luôn phải kiểm tra lại độ chính xác trên một tập validation độc lập. Quá trình này đảm bảo mô hình sau khi chuyển đổi có chất lượng không quá giảm so với mô hình gốc.

Trên thân của băng tải được gắn Raspberry Pi 4 cùng với camera nhằm mục đích chụp ảnh đồng thời truyền video lên màn hình điều khiển khi vật đi trên băng tải. Khi đối tượng đến khu vực của camera thì băng tải sẽ tự dừng lại và camera sẽ chụp ảnh lặp lại cho đến hết. Tất cả các hình đã chụp sẽ được máy tính nhúng thực hiện công việc kiểm tra của mình, ứng với mỗi nhãn Raspberry Pi sẽ gửi dữ liệu xuống arduino để thực hiện việc phân loại qua servo. Với kết quả đào tạo tốt sẽ hỗ trợ dự đoán chính xác, độ tin cậy cao. Đồng thời kết quả phân loại và hình ảnh sẽ được giám sát trực tiếp tới màn hình.



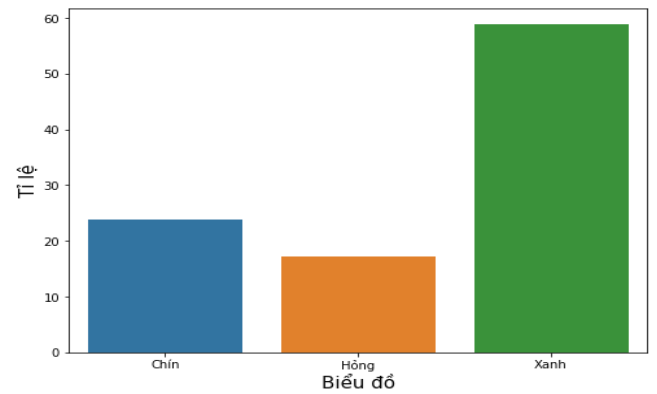
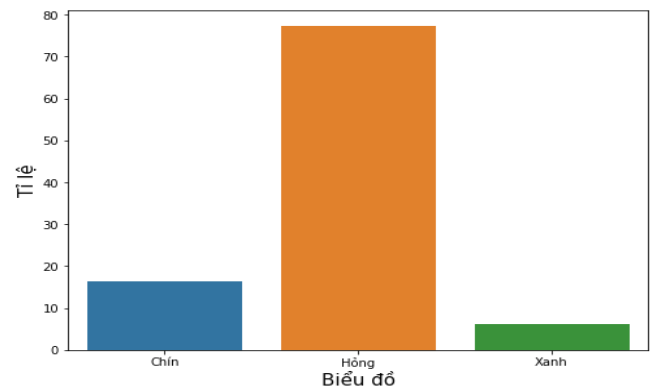
Hình 3.15 Quá trình phân loại

3.2.4. Phân loại trên băng tải

Khi khởi động dây chuyền, từng đối tượng sẽ lần lượt di chuyển trên băng tải. Ở phía đầu băng tải được gắn cảm biến vật cản (IR sensor), khi có vật đi trên băng tải tới vị trí cảm biến vật cản, băng tải sẽ dừng lại đúng với vị trí đặt camera để ảnh đầu vào Raspberry trong vùng rõ nét. Raspberry sẽ xử lý ảnh đồng thời gửi dữ liệu tới Arduino để điều khiển phân loại đối tượng, tiếp tục lặp lại như vậy cho đến khi kết thúc. Nếu muốn dừng băng tải thì có thể tắt thông qua màn hình giám sát được hiển thị khi kết nối tới raspberry pi.

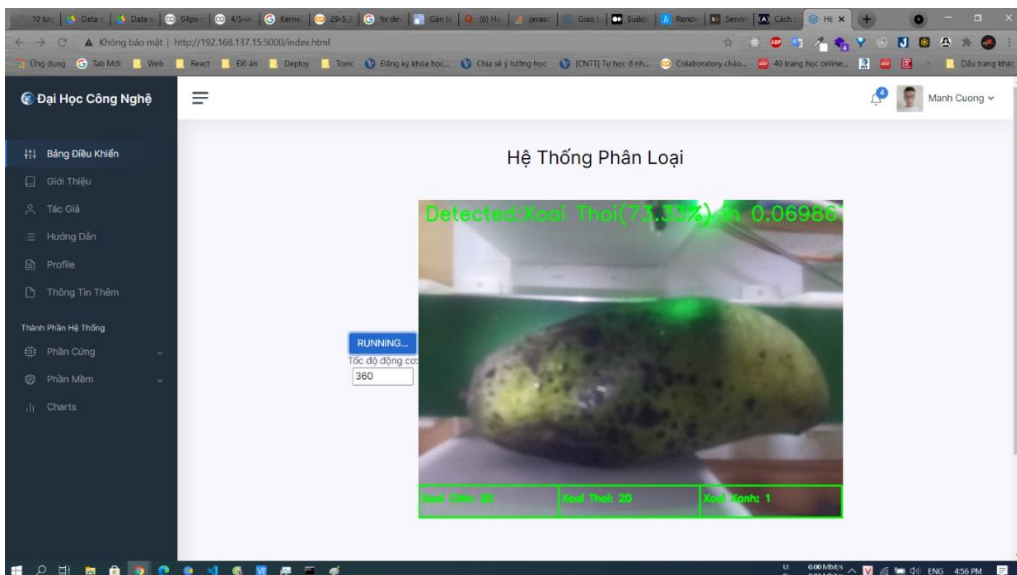
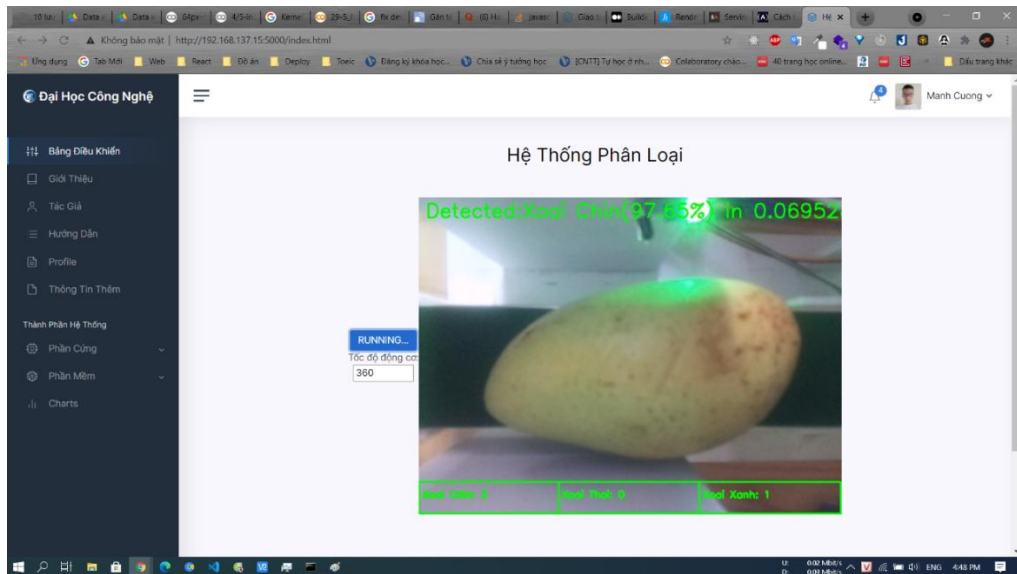
Camera pi được gắn trên băng tải với mục đích chụp ảnh và quay quá trình băng tải hoạt động khi đi qua camera. Hình ảnh được camera ghi lại sẽ được máy tính nhúng raspberry pi chuẩn đoán hầu như ngay lập tức với độ chính xác 80% - 92%. Do mô hình dự đoán trong Raspberry pi được chuyển đổi giảm kích thước so với mô hình gốc có thể độ chính xác bị suy giảm, camera dùng mô phỏng không có độ nhạy sáng thấp nên hình ảnh chưa thể hiện rõ các đặc điểm của đối tượng cần phân loại. Để khắc phục cần cải thiện hình ảnh đầu vào từ camera pi hoặc cải thiện mô hình dự đoán bằng cách kiểm tra độ chính xác của mô hình nhiều lần rồi xác định tệp dự đoán tối ưu, trước khi nạp tệp dự đoán vào Pi.

Ảnh thực tế camera pi nhận được sẽ là ảnh chứa từng đối tượng với các nhãn khác nhau.



Hình 3.16 Kết quả kiểm tra hình ảnh

Trong màn hình quan sát, kết quả chuẩn đoán được truyền đồng thời với từng ảnh của quả đi qua và số lượng đã phân loại tương ứng với từng nhãn trường hợp. Lệnh kiểm soát băng tải thông qua màn hình được gửi tới arduino để tiến hành điều khiển động cơ phân loại.



Hình 3.17 Kết quả phân loại thực tế

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận

Trong khoá luận này, tôi đã phát triển một phương pháp phân loại hoa quả dựa trên mạng nơ-ron tích chập (CNN). Kết quả mô hình đạt độ chính xác tương đối cao. Bên cạnh đó, việc tăng dữ liệu đào tạo cho thấy hiệu quả khi tránh được overfitting. Mô hình phần cứng hoạt động tốt, thiết bị nhỏ gọn, lắp đặt dễ dàng, có tính kinh tế, thuận tiện cho việc học tập và cũng có thể phát triển thành dây chuyền sản xuất.

Những thiếu sót trong phương pháp được sử dụng trong khoá luận là tập dữ liệu không đủ lớn và do đó nó hoạt động không tốt trên các hình ảnh không hoàn hảo. Ngoài ra thì hệ thống vẫn còn những hạn chế sau: Tốc độ xử lý còn chậm do camera yếu tốc độ khung hình thấp, chất lượng chụp ảnh thấp trong điều kiện ánh sáng yếu.

Hướng phát triển

Trong tương lai, tôi sẽ cố gắng sử dụng hình ảnh thực tế thu được trong các nhà vườn, siêu thị và nhà máy. Bên cạnh đó cần thử nghiệm các ý tưởng phân loại nâng cao khác, chẳng hạn như học chuyển giao và sử dụng FGPA để đẩy nhanh việc triển khai thuật toán. Phương pháp trên có thể áp dụng để phát hiện thêm nhiều đối tượng cho các loại nông sản khác và cải tiến dây chuyền phân loại với camera tốt đi cùng cấu hình phần cứng mạnh hơn. Đồng thời tăng khả năng giám sát và kiểm soát hoạt động dây chuyền từ xa.

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] Nguyễn Đắc Thành (2017). Nhận dạng và phân loại hoa quả trong ảnh màu.
- [2] Vũ Hữu Tiệp. Machine Learning cơ bản – Overfitting (2017). <https://machinelearningcoban.com/2017/03/04/overfitting/>

Tiếng Anh

- [3] Alan Lapedes, Robert Farber, “How Neural Nets Work”, *Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545*.
- [4] Andrej Karpathy. CS231n Convolutional Neural Networks for Visual Recognition - Image Classification. <https://cs231n.github.io/classification/>
- [5] Aston Zhang, Zach C. Lipton (2019). Convolutional Neural Networks - Dive into Deep Learning.
- [6] Camera module v2. <https://www.raspberrypi.org/products/camera-module-v2>
- [7] Data augmentation (2020), https://en.wikipedia.org/wiki/Data_augmentation
- [8.] Deliens, T., B. Deforche, L. Annemans, et al., Effectiveness of Pricing Strategies on French Fries and Fruit Purchases among University Students: Results from an On-Campus Restaurant Experiment. Plos One, 2016.
- [9] Flask (2020), [https://en.wikipedia.org/wiki/Flask_\(web_framework\)](https://en.wikipedia.org/wiki/Flask_(web_framework))
- [10] Getahun, S., A. Ambaw, M. Delele, et al., Analysis of airflow and heat transfer inside fruit packed refrigerated shipping container: Part I - Model development and validation. Journal of Food Engineering, 2017. 203: pp. 58-68.
- [11] Liu, F., L. Snetkov, and D. Lima, Summary on fruit identification methods: A literature review. Advances in Social Science, Education and Humanities Research, 2017.
- [12] Naik, S., & Patel, B. (2017). Machine Vision based Fruit Classification and Grading – A Review. International Journal of Computer Applications, 170(9), 22-34.
- [13] Open CV (2021), <https://docs.opencv.org/master/d1/dfb/intro.html>

- [14] Nishi, T., Kurogi, S., & Matsuo, K. (2017). Grading fruits and vegetables using RGB-D images and convolutional neural network. 2017 IEEE Symposium Series on Computational Intelligence (SSCI)
- [15] Shao, W.H., Y.J. Li, S.F. Diao, et al., Rapid classification of Chinese quince (*Chaenomeles speciosa* Nakai) fruit provenance by near-infrared spectroscopy and multivariate calibration. *Analytical and Bioanalytical Chemistry*.
- [16] Radi, S. Ciptohadijoyo, W.S. Litananda, et al., *Electronic nose based on partition column integrated with gassensor for fruit identification and classification*. *Computers and Electronics in Agriculture*, 2016.
- [17] Raspberry Pi (2021), <https://www.raspberrypi.org/documentation/raspbian>
- [18] Seng, W. C., & Mirisae, S. H. (2009). *A new method for fruits recognition system*. *Proceedings of the 2009 International Conference on Electrical Engineering and Informatics, ICEEI 2009*, 1, 130–134.
- [19] Tensorflow Lite (2019). <https://www.tensorflow.org/lite/guide>
- [20] Wu, L., Classification of fruits using computer vision and a multiclass support vector machine. *Sensors*, 2012.12(9)