

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN TỬ - VIỄN THÔNG



BÁO CÁO PBL4: CHUYÊN ĐỀ HỆ THỐNG MÁY TÍNH

ĐỀ TÀI:
NHẬN DẠNG CÁC LOẠI QUẢ KHÔ TRONG THUỐC ĐÔNG Y
BẰNG PHƯƠNG PHÁP CNN

Giảng viên hướng dẫn: **TS. Hoàng Lê Uyên Thực**

Sinh viên thực hiện: **Đinh Ngọc Tiên**

Nguyễn Quốc Việt

Mã số sinh viên: **106190085 & 106190092**

Lớp: **19DTCLC2**

Đà Nẵng, 28 tháng 12 năm 2022

MỤC LỤC

NHIỆM VỤ ĐỒ ÁN	iv
LỜI CAM ĐOAN	vi
DANH MỤC HÌNH ẢNH	vii
DANH MỤC TỪ VIẾT TẮT	viii
DANH MỤC BẢNG BIỂU	viii
MỞ ĐẦU	1
CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN	3
1.1. Mở đầu chương	3
1.2. Bài toán nhận dạng và phân loại các loại quả khô thuốc Đông Y	3
1.3. Các hướng tiếp cận và giải quyết bài toán.....	4
1.3.1. Phương pháp học máy truyền thống	5
1.3.2. Phương pháp học sâu	8
1.4. Kết luận chương	10
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	11
2.1. Mở đầu chương	11
2.2. Kiến trúc mạng nơ-ron tích chập	11
2.2.1. Lớp tích chập (Convolution Layer)	12
2.2.2. Hàm kích hoạt phi tuyến ReLU	14
2.2.3. Lớp lấy mẫu (Pooling Layer)	15
2.2.4. Lớp kết nối đầy đủ	15
2.2.5. Thuật toán tối ưu sử dụng cho bài toán	16
2.2. Mạng AlexNet.....	16
2.2.1. Kiến trúc mạng AlexNet	17
2.2.2. Ứng dụng mạng AlexNet vào bài toán nhận dạng các quả khô trong thuốc Đông Y	19
2.3. Kết luận chương	19
CHƯƠNG 3: THIẾT KẾ MÔ HÌNH, THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ	20
3.1. Mở đầu chương	20
3.2. Quy trình thực hiện hệ thống.....	20

3.3. Thu thập và xây dựng cơ sở dữ liệu	20
3.3.1. Cơ sở dữ liệu ảnh các loại quả khô trong thuốc Đông Y	20
3.3.2. Sử dụng thuật toán tăng cường dữ liệu.....	20
3.4. Xây dựng mô hình huấn luyện.....	22
3.5. Đánh giá các mô hình huấn luyện	25
3.6. Kết luận chương	29
KẾT LUẬN HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI.....	30
PHỤ LỤC	31
TÀI LIỆU THAM KHẢO.....	41

NHIỆM VỤ ĐỒ ÁN

1. Tên đề tài: “**Nhận dạng các loại quả khô trong thuốc Đông Y bằng phương pháp CNN**”
2. Các số liệu và dữ liệu ban đầu:
 - Sử dụng bộ dữ liệu thu thập từ Internet (các trang web, Google, mạng xã hội)
3. Nội dung các phần thuyết minh và tính toán

Phần chung:

STT	Họ và tên	Nội dung
1	Đinh Ngọc Tiên	<ul style="list-style-type: none">- Tìm hiểu, chọn lựa phương pháp phù hợp với đề tài.- Thu thập, chọn lọc cơ sở dữ liệu.- Tìm hiểu các lớp có trong mô hình CNN, xây dựng mạng CNN.
2	Nguyễn Quốc Việt	<ul style="list-style-type: none">- Thử nghiệm, điều chỉnh mô hình phù hợp với bộ cơ sở dữ liệu- Tìm hiểu các phương pháp huấn luyện và đánh giá mô hình.- Viết báo cáo cho đề tài.

Phần riêng:

STT	Họ và tên	Nội dung
1	Đinh Ngọc Tiên	<ul style="list-style-type: none">- Tìm hiểu thuật toán tăng cường dữ liệu.- Tìm hiểu mạng AlexNet.- Tìm hiểu công thức, tính toán các thông số trong mạng.
2	Nguyễn Quốc Việt	<ul style="list-style-type: none">- Thử nghiệm nhiều mô hình, mô tả các mô hình bằng đồ thị

		<ul style="list-style-type: none">- Tìm hiểu phương pháp đánh giá mô hình dựa trên các tiêu chí Accuracy, Precision, Recall, F1 score.
--	--	--

4. Các bản vẽ, đồ thị:

Phần chung:

STT	Họ và tên	Nội dung
1	Đinh Ngọc Tiên	<ul style="list-style-type: none">- Đồ thị Accuracy – Loss
2	Nguyễn Quốc Việt	

Phần riêng:

STT	Họ và tên	Nội dung
1	Đinh Ngọc Tiên	<ul style="list-style-type: none">- Kiến trúc mạng AlexNet- Kiến trúc mạng CNN của mô hình 9
2	Nguyễn Quốc Việt	<ul style="list-style-type: none">- Quy trình xây dựng mô hình huấn luyện- Sơ đồ khối của hệ thống nhận diện trái cây khô trong thuốc Đông Y- Ma trận nhầm lẫn

5. Họ và tên người hướng dẫn: TS. Hoàng Lê Uyên Thực

6. Ngày giao nhiệm vụ đồ án: .../.../2022

7. Ngày hoàn thành: .../.../2022

Đà Nẵng, ngày 28 tháng 12 năm 2022

Sinh viên

Người hướng dẫn

LỜI CAM ĐOAN

Chúng tôi xin cam đoan nội dung của đề án “Nhận dạng các loại quả khô trong thuốc Đông Y bằng phương pháp CNN” là công trình nghiên cứu của chúng tôi. Những nội dung tham khảo trong đề án đã được nêu rõ trong phần Tài liệu tham khảo. Các số liệu, kết quả trình bày trong đề án là hoàn toàn trung thực. Nếu vi phạm, chúng tôi xin chịu hoàn toàn trách nhiệm và chịu mọi kỷ luật của giảng viên bộ môn và Khoa đã đề ra.

Đà Nẵng, ngày 28 tháng 12 năm 2022

Sinh viên thực hiện

DANH MỤC HÌNH ẢNH

Hình 1.1. Các khó khăn trong bài toán nhận dạng vật thể trong ảnh	4
Hình 1.2. Các thông tin về hình học được tính toán bởi các thuật toán Xử lý ảnh	5
Hình 1.3. Mô hình hoạt động chung của các phương pháp Học máy	6
Hình 1.4. Mối quan hệ của Học sâu với các lĩnh vực liên quan.....	8
Hình 1.5. Mức độ trừu tượng tăng dần qua các tầng học của Học sâu	9
Hình 2.1. Kiến trúc cơ bản của một mạng tích chập	12
Hình 2.2. Biểu diễn ảnh đầu vào dưới dạng ma trận	12
Hình 2.3. Ví dụ bộ lọc tích chập được sử dụng trên ma trận điểm ảnh.....	13
Hình 2.4. Trường hợp thêm/không thêm viền trắng vào ảnh khi thực hiện tích chập	14
Hình 2.5. Đồ thị hàm phi tuyến ReLU	14
Hình 2.6. Phương thức Max Pooling và Average Pooling	15
Hình 2.7. Kiến trúc mạng AlexNet.....	17
Hình 3.1. Tăng cường dữ liệu với thao quả.....	21
Hình 3.2. Sắp xếp thư mục dữ liệu cho việc gán nhãn	21
Hình 3.3. Quy trình xây dựng mô hình huấn luyện.....	22
Hình 3.4. Kiến trúc mạng CNN của mô hình 9	23
Hình 3.5. Sơ đồ khối của hệ thống nhận diện trái cây khô trong thuốc Đông Y	24
Hình 3.6. Biểu đồ thể hiện Accuracy và Loss của mô hình 9	26
Hình 3.7. Biểu đồ thể hiện Accuracy và Loss của mô hình 9	27
Hình 3.8. Kết quả tính toán các chỉ số đánh giá của mô hình 9	28
Hình 3.9. Ma trận nhầm lẫn	28
Hình 3.10. Kiểm tra nhận diện bằng hình ảnh.....	29

DANH MỤC TỪ VIẾT TẮT

STT	Từ viết tắt	Ý nghĩa
1	CSDL	Cơ sở dữ liệu
2	CNN	Convolutional Neural Network – Mạng nơ-ron tích chập
3	ReLU	Rectified Linear Unit – Tính chỉnh đơn vị tuyến tính
4	GPU	Graphic Processing Unit – Bộ vi xử lý đồ họa
5	CNN	Convolutional Neural Network

DANH MỤC BẢNG BIỂU

Bảng 2.1. Bảng tổng kết của mô hình AlexNet.....	18
Bảng 3.1. Phương pháp gán nhãn One-hot-encoding.....	22
Bảng 3.2. Các mô hình huấn luyện thử nghiệm	23
Bảng 3.3. Kết quả đánh giá các mô hình thử nghiệm theo độ chính xác và thời gian	25

MỞ ĐẦU

Tính cấp thiết của đề tài

Y học cổ truyền đã phát triển từ thời kỳ dựng nước và đến nay vẫn không ngừng phát triển. Trong Đông Y, nguyên liệu thường dùng làm thuốc là thảo mộc, khoáng vật, động vật, trong đó thảo mộc được dùng nhiều hơn cả, bao gồm lá, hoa, quả, hạt, thân cây, rễ cây. Nguyên liệu thu hái về phơi hoặc sấy khô gọi là dược liệu. Khi các thầy thuốc mua dược liệu về bào chế thành phẩm gọi là thuốc Đông Y. Hiện nay việc phân loại và đánh giá chất lượng các nguyên liệu thảo mộc này (cụ thể là quả khô) 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 ngành Đông Y. Do đó, nhiều phương pháp tự động hóa công việc thu hoạch, nhận dạng và đánh giá chất lượng hoa quả đã được nghiên cứu và đưa vào ứng dụng thực tế, trong đó sử dụng chủ yếu các phương pháp Xử lý ảnh đơn thuần. Tuy nhiên, các phương pháp này vẫn chưa thực sự thỏa mãn yêu cầu về khả năng nhận dạng một số lượng lớn các loại quả với độ chính xác cao do bị hạn chế bởi các đặc trưng của bài toán nhận dạng quả: số lượng chủng loại lớn với nhiều loại quả hết sức tương tự nhau, sự biến thiên về hình dạng, màu sắc, chi tiết trong từng loại quả cũng rất khó dự đoán trước...

Trong thời gian gần đây, nhờ có sự phát triển mạnh mẽ về khả năng tính toán của các thế 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 quả. Đề tài nghiên cứu “Nhận dạng các loại quả khô trong thuốc Đông Y” 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 các loại quả tự động, đặc biệt là đối với các loại quả phổ biến tại nước ta.

Mục tiêu của đề tài

Do thời gian hạn chế trong thời gian thực hiện nghiên cứu, đề tài trước hết tập trung nghiên cứu, tìm hiểu và so sánh các phương pháp Học máy truyền thống với phương pháp Học sâu, đồng thời thực hiện cài đặt một mô hình huấn luyện về nhận dạng ảnh trong Học sâu với số lượng quả được hạn chế.

Cấu trúc của đề tài

Dựa trên mục tiêu cụ thể đã trình bày trong phần trước, đề tài được tổ chức thành năm chương với các nội dung cụ thể như sau:

Chương 1: Trong chương tổng quan này, ta sẽ có cái nhìn tổng quan về các hướng tiếp cận và giải pháp đã được ứng dụng trong bài toán nhận dạng các loại quả, từ các

phương pháp thuần tính toán xử lý ảnh tương đối thô sơ cho tới các phương pháp Học máy truyền thống và cuối cùng là các phương pháp Học sâu - một nhánh đặc biệt trong Học máy.

Chương 2: Chương này sẽ đi sâu hơn vào một mạng huấn luyện trong Học sâu thường được sử dụng trong lĩnh vực Nhận dạng ảnh - mạng nơ-ron tích chập, và tìm hiểu chìa khóa giải quyết bài toán nhận dạng ảnh với mạng nơ-ron tích chập CNN

Chương 3: Chương 3 tập trung trình bày về kết quả thực nghiệm, bao gồm kết quả so sánh độ chính xác giữa các mô hình huấn luyện, cùng với các đánh giá về độ hiệu quả của bộ tạo dữ liệu ảnh nhiễu cũng như các ảnh chụp thực tế khi được sử dụng trong thực tế. Dựa trên các kết quả thực nghiệm này, ta sẽ đưa ra một số phân tích và kết luận về điểm mạnh và điểm hạn chế của mô hình huấn luyện Học sâu đã chọn.

Cuối cùng là tổng kết các nội dung đã trình bày trong đồ án, từ đó đề xuất các phương hướng nghiên cứu tiếp theo để tiếp tục cải thiện chất lượng nhận dạng của hệ thống

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

1.1. Mở đầu chương

Trong chương này, nhóm sẽ trình bày bài toán cần giải quyết cũng như các khó khăn trong bài toán để có cái nhìn tổng quan về đề tài. Sau đó, nhóm sẽ đi vào giới thiệu các hướng tiếp cận và giải quyết bài toán nhận dạng.

1.2. Bài toán nhận dạng và phân loại các loại quả khô thuốc Đông Y

Nhận dạng vật thể trong ảnh được coi là bài toán cơ bản nhất trong lĩnh vực Thị giác máy tính, là nền tảng cho rất nhiều bài toán mở rộng khác như bài toán phân lớp, định vị, tách biệt vật thể.... Tuy bài toán cơ bản này đã tồn tại hàng thế kỷ nhưng con người vẫn chưa thể giải quyết nó một cách triệt để, do tồn tại rất nhiều khó khăn để máy tính có thể hiểu được các thông tin trong một bức ảnh. Trong đó, những khó khăn tiêu biểu [3] phải kể đến:

- Sự đa dạng trong góc chụp – Viewpoint: Cùng một vật thể nhưng có thể có rất nhiều vị trí và góc nhìn khác nhau, dẫn đến các hình ảnh thu được về vật thể đó sẽ không giống nhau. Việc huấn luyện để máy tính có thể hiểu được điều này thực sự là một thách thức khó khăn.
- Sự đa dạng trong kích thước: Các bức ảnh không có cách nào thể hiện trường thông tin về kích thước của vật thể trong đời thực, và máy tính cũng chỉ có thể tính toán được tỉ lệ tương đối của vật thể so với bức ảnh bằng cách đếm theo số lượng các điểm ảnh vật thể đó chiếm trong ảnh.
- Các điều kiện khác nhau của chiếu sáng: Ánh sáng có ảnh hưởng mạnh mẽ đến thông tin thể hiện trong một bức ảnh, đặc biệt là ở mức độ thấp như mức độ điểm ảnh.
- Sự che khuất một phần của vật thể sau các đối tượng khác trong ảnh: Trong các bức ảnh, vật thể không nhất định phải xuất hiện với đầy đủ hình dạng mà có thể bị che lấp một phần nào đó bởi nền hoặc các vật thể xung quanh. Sự không đầy đủ về hình dạng của vật thể sẽ dẫn đến việc thiếu thông tin, đặc trưng và càng làm bài toán nhận dạng khó khăn hơn.
- Sự lộn xộn phức tạp của nền: Trong nhiều trường hợp, vật thể cần nhận dạng bị lẫn gần như hoàn toàn vào nền của bức ảnh, sự lẫn lộn về màu sắc, họa tiết giữa vật thể và nền khiến cho việc nhận dạng trở nên vô cùng khó khăn, kể cả với thị giác con người.
- Sự đa dạng về chủng loại vật thể: Vật thể cần nhận dạng có thể bao gồm nhiều chủng loại khác nhau, với hình dạng, màu sắc, kết cấu vô cùng khác biệt. Đây chính là một thách thức nữa với bài toán nhận dạng, đó là làm thế nào để các mô hình nhận dạng của máy tính có thể nhận biết được các biến thể về chủng loại của vật thể, ví dụ các loại ghế khác nhau, trong khi vẫn tách biệt được đâu là các vật thể khác loại, ví dụ phân biệt bàn với ghế...



Hình 1.1. Các khó khăn trong bài toán nhận dạng vật thể trong ảnh

Là một trường hợp cụ thể của bài toán nhận dạng và phân lớp, bài toán nhận dạng các loại quả kế thừa các khó khăn vốn có của bài toán gốc, và kèm theo là các khó khăn riêng của chính nó, như: số lượng khổng lồ về chủng loại quả theo mùa, vùng miền, địa hình... với vô số loại hoa quả có hình dáng, màu sắc, kết cấu 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 dạng của cùng một loại quả do ảnh hưởng của thời tiết, điều kiện thổ nhưỡng và chế độ dinh dưỡng...

1.3. Các hướng tiếp cận và giải quyết bài toán

Bài toán tự động nhận dạng quả đã xuất hiện từ lâu và đã có rất nhiều bài báo, công trình khoa học được đưa ra nhằm đề xuất hoặc cải tiến các thuật toán nhận dạng. Trong đó, xuất hiện sớm nhất là các phương pháp Xử lý ảnh – Image Processing, các phương pháp này tập trung vào phát triển các thuật toán nhằm trích xuất thông tin, ví dụ các tham số về màu sắc, hình dạng, kết cấu, kích thước..., từ bức ảnh đầu vào để nhận dạng hoa quả [5]. Do chỉ đơn thuần xử lý trên một vài ảnh đầu vào trong khi sự biến thiên về màu sắc, hình dạng, kích thước... của quả quá phức tạp, kết quả đạt được của các phương pháp này không được cao và phạm vi áp dụng trên số lượng loại quả cũng bị hạn chế.

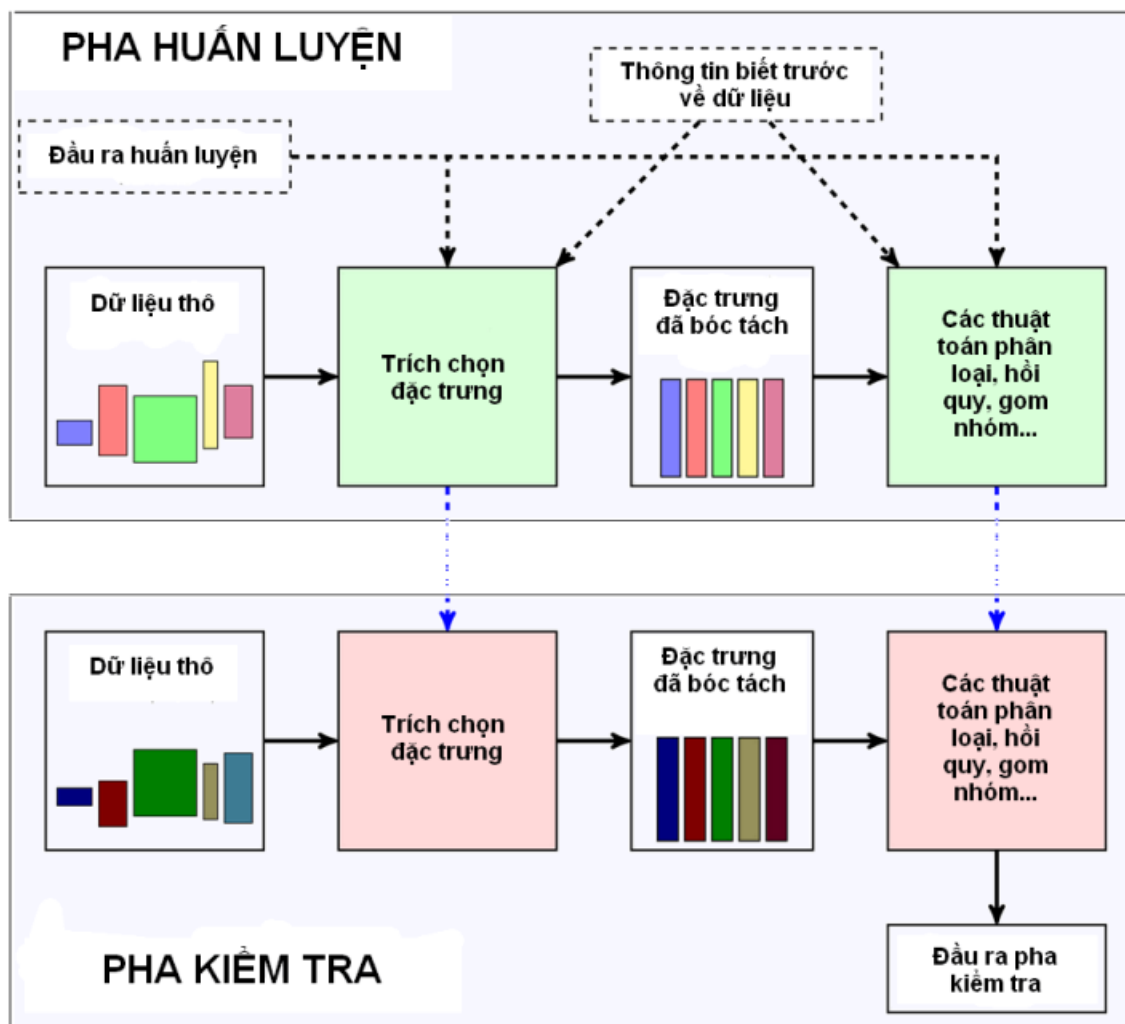


Hình 1.2. Các thông tin về hình học được tính toán bởi các thuật toán Xử lý ảnh

Bắt đầu từ những năm 2000s, sau khi xuất hiện một bài báo khoa học đề xuất áp dụng phương pháp Học máy - Machine Learning - vào bài toán nhận dạng hoa quả với độ chính xác cao [6], hướng giải quyết bài toán đã tập trung vào ứng dụng và cải tiến các thuật toán Học máy, cụ thể là nghiên cứu, thử nghiệm trích chọn các đặc trưng phù hợp nhất để đưa vào huấn luyện bộ nhận dạng tự động. Kết quả thu được tương đối khả quan, khả năng nhận dạng quả tự động đã được cải thiện với số lượng loại quả được mở rộng và độ chính xác của nhận dạng cao hơn nhiều so với các phương pháp thuần Xử lý ảnh ban đầu [7]. Nói tiếp sự phát triển của Học máy, trong những năm gần đây, nhờ sự phát triển vượt bậc về sức mạnh tính toán của các máy tính cũng như sự bùng nổ dữ liệu trên Internet, một nhánh đặc biệt trong Học máy là Học sâu – Deep Learning đã đạt được nhiều thành tựu đáng kể, đặc biệt là trong lĩnh vực Xử lý ảnh và ngôn ngữ tự nhiên. Học sâu cũng đã được áp dụng rất thành công vào bài toán nhận dạng quả, trong các thử nghiệm với phạm vi hạn chế về số lượng loại quả cần nhận dạng, phương pháp này đã đạt được kết quả rất cao. Sau đây ta sẽ tìm hiểu sâu hơn về hai tiếp cận chính hiện nay để giải quyết bài toán nhận dạng quả nói riêng và nhận dạng vật thể trong ảnh nói chung: phương pháp Học sâu và các phương pháp Học máy truyền thống không sử dụng Học sâu.

1.3.1. Phương pháp học máy truyền thống

Mô hình hoạt động chung của các phương pháp Học máy truyền thống được thể hiện trong Hình 1.3 dưới đây [2]:



Hình 1.3. Mô hình hoạt động chung của các phương pháp Học máy

Từ hình ta có thể thấy Học máy gồm hai giai đoạn chính là Huấn luyện – Training và Thử nghiệm – Testing, trong mỗi giai đoạn đều sử dụng hai thành phần quan trọng nhất do người xử lý bài toán thiết kế, đó là Trích chọn đặc trưng – Feature Engineering (hay còn gọi là Feature Extraction) và Thuật toán phân loại, nhận dạng... - Algorithms. Hai thành phần này có ảnh hưởng trực tiếp đến kết quả bài toán, vì thế được thiết kế rất cẩn thận, tốn nhiều thời gian, đòi hỏi người thiết kế phải có kiến thức chuyên môn và nắm rõ đặc điểm của bài toán cần xử lý.

Trích chọn đặc trưng

Trong các bài toán thực tế, ta chỉ có được những dữ liệu thô chưa qua chọn lọc xử lý, và để có thể đưa các dữ liệu này vào huấn luyện ta cần có những phép biến đổi để biến các dữ liệu thô thành dữ liệu chuẩn, với khả năng biểu diễn dữ liệu tốt hơn. Các phép biến đổi bao gồm loại bỏ dữ liệu nhiễu và tính toán để lưu lại các thông tin đặc trưng, có ý nghĩa

từ dữ liệu thô ban đầu. Các thông tin đặc trưng này là khác nhau với từng loại dữ liệu và bài toán cụ thể, vì thế trong từng trường hợp phép biến đổi này cần phải được tùy biến một cách thích hợp để cải thiện độ chính xác của mô hình dự đoán. Quá trình này được gọi là *Trích chọn đặc trưng* – Feature Engineering, là một thành phần rất quan trọng trong các phương pháp Học máy truyền thống.

- **Đầu vào:** Toàn bộ thông tin của dữ liệu, không có quy chuẩn về dạng thông tin (véc tơ, ma trận...) hay kích thước các chiều thông tin. Đồng thời, do chứa toàn bộ thông tin, gồm cả thông tin nhiễu và không có giá trị nên kích thước lưu trữ thường lớn và không có lợi cho tính toán sau này.
- **Đầu ra:** Các thông tin hữu ích đã được tính toán, rút ra từ dữ liệu đầu vào, trong đó không còn các thành phần nhiễu hay vô nghĩa. Kích thước dữ liệu đầu ra đã được rút gọn rất nhiều so với kích thước dữ liệu đầu vào, giúp cho việc tính toán về sau trở nên nhanh gọn, thuận tiện hơn rất nhiều.
- **Thông tin biết trước về dữ liệu:** Đây là thành phần tùy chọn, không bắt buộc với mọi bài toán, mà chỉ xuất hiện trong một số trường hợp cụ thể với những thông tin rõ ràng về đặc trưng hữu ích với mô hình dự đoán. Các thông tin biết trước này giúp người thiết kế có thể lựa chọn được những đặc trưng tốt nhất và các phương pháp tính toán phù hợp nhất để ra được mô hình dự đoán với độ chính xác cao.

Thuật toán

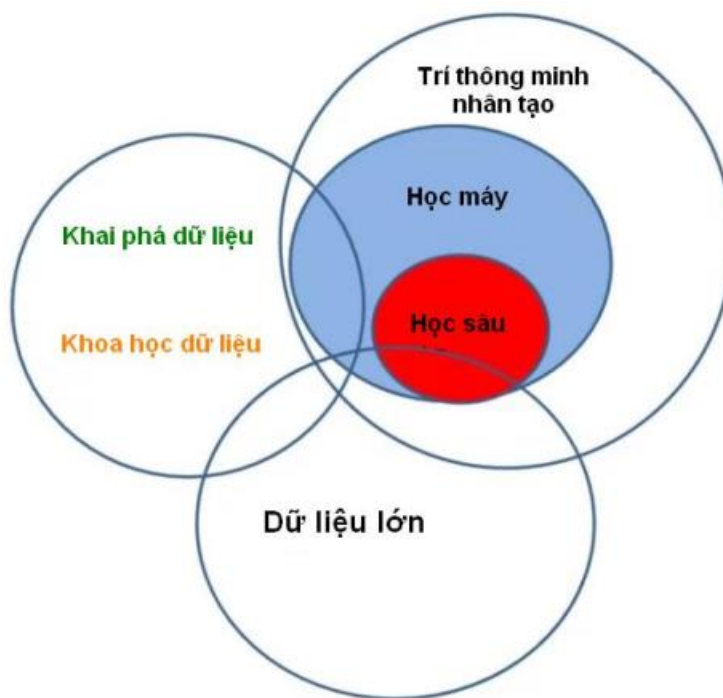
Sau quá trình trích chọn đặc trưng ở bước trước, ta có được các đặc trưng, được lưu trữ ở định dạng chuẩn về kiểu dữ liệu, kích thước dữ liệu..., và các thông tin đặc trưng này có thể được sử dụng cùng với các thông tin biết trước về dữ liệu (nếu có) để xây dựng ra các mô hình dự đoán phù hợp bằng các thuật toán khác nhau. Các thuật toán trong Học máy thường được phân loại theo hai cách phổ biến là theo phương thức học hoặc theo chức năng của thuật toán, ví dụ như:

- Phân nhóm theo phương thức học: Học giám sát (Supervised Learning) và Học không giám sát (Unsupervised Learning)
- Phân nhóm theo chức năng: Các thuật toán hồi quy, phân loại, gom nhóm... Một đặc điểm nổi bật của các phương pháp Học máy truyền thống là độ chính xác của mô hình dự đoán phụ thuộc rất nhiều vào chất lượng các đặc trưng được lựa chọn, các đặc trưng này càng phù hợp với bài toán đưa ra thì kết quả thu được càng tốt. Đây là điểm mạnh, và cũng là điểm yếu của các phương pháp này, bởi việc trích chọn đặc trưng chính là sự đóng góp của bản tay con người trong việc cải tiến các mô hình, nó yêu cầu sự hiểu biết thấu đáo về bài toán cần giải quyết, các thuật toán sử dụng và các thông số trong mô hình huấn luyện. Các đặc trưng được thiết kế riêng cho từng bài toán khác biệt, do vậy hiếm khi chúng có thể được tái sử dụng với các bài toán mới mà cần phải được cải thiện hay thay thế bởi các đặc trưng khác.

Một đặc điểm nổi bật của các phương pháp Học máy truyền thống là độ chính xác của mô hình dự đoán phụ thuộc rất nhiều vào chất lượng các đặc trưng được lựa chọn, các đặc trưng này càng phù hợp với bài toán đưa ra thì kết quả thu được càng tốt. Đây là điểm mạnh, và cũng là điểm yếu của các phương pháp này, bởi việc trích chọn đặc trưng chính là sự đóng góp của bản tay con người trong việc cải tiến các mô hình, nó yêu cầu sự hiểu biết thấu đáo về bài toán cần giải quyết, các thuật toán sử dụng và các thông số trong mô hình huấn luyện. Các đặc trưng được thiết kế riêng cho từng bài toán khác biệt, do vậy hiếm khi chúng có thể được tái sử dụng với các bài toán mới mà cần phải được cải thiện hay thay thế bởi các đặc trưng khác.

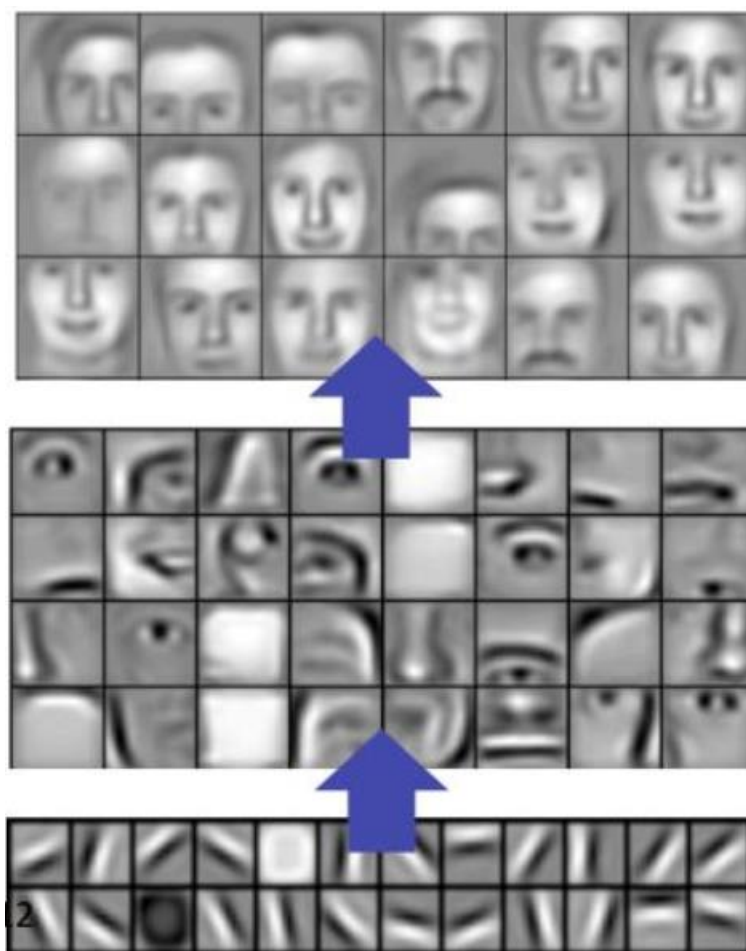
1.3.2. Phương pháp học sâu

Học sâu là một nhánh đặc biệt của ngành Học máy, và bắt đầu trở nên phổ biến trong thập kỷ gần đây do các nhà khoa học đã có thể tận dụng khả năng tính toán mạnh mẽ của các máy tính hiện đại cũng như khối lượng dữ liệu khổng lồ (hình ảnh, âm thanh, văn bản,...) trên Internet. Ta có thể thấy rõ mối quan hệ giữa Học sâu với Học máy cũng như các lĩnh vực liên quan khác qua hình 1.4:



Hình 1.4. Mối quan hệ của Học sâu với các lĩnh vực liên quan

Các mạng huấn luyện theo phương pháp Học sâu còn được gọi với cái tên khác là mạng nơ-ron sâu (Deep Neural Network) do cách thức hoạt động của chúng. 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 (như mô tả trong hình 1.5).



Hình 1.5. Mức độ trừu tượng tăng dần qua các tầng học của Học sâu

Cụ thể, với một mạng Học sâu cho nhận dạng ảnh, các lớp đầu tiên trong mạng chỉ làm nhiệm vụ rất đơn giản là tìm kiếm các đường thẳng, đường cong, hoặc đốm màu trong ảnh đầu vào. Các thông tin này sẽ được sử dụng làm đầu vào cho các lớp tiếp theo, với nhiệm vụ khó hơn là từ các đường, các cạnh đó tìm ra các thành phần của vật thể trong ảnh. Cuối cùng, các lớp cao nhất trong mạng huấn luyện sẽ nhận nhiệm vụ phát hiện ra vật thể trong ảnh.

Với cách thức học thông tin từ ảnh lần lượt qua rất nhiều lớp, nhiều tầng khác nhau như vậy, các phương pháp này có thể giúp cho máy tính hiểu được những dữ liệu phức tạp bằng nhiều lớp thông tin đơn giản qua từng bước phân tích. Đó cũng là lý do chúng được gọi là các phương pháp Học sâu.

Như đã trình bày trong phần mở đầu, mục đích của đề tài là tìm hiểu và ứng dụng một mô hình Học sâu vào bài toán nhận dạng các loại quả khô trong thuốc Đông Y, nguyên nhân chính khiến Học sâu được chọn làm giải pháp là bởi khả năng mạnh mẽ vượt trội của nó đối với các phương pháp Học máy truyền thống khi áp dụng vào các bài toán nhận dạng

vật thể, trong đó vật thể là các đối tượng rất khó chọn lọc đặc trưng phù hợp, cụ thể với trường hợp này là các loại quả thuốc Đông Y.

1.4. Kết luận chương

Qua chương này, nhóm đã trình bày được tổng quan đề tài, các hướng tiếp cận và giải quyết bài toán nhận dạng.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

Mạng nơ-ron tích chập (CNN - Convolutional Neural Network) là một trong những mô hình mạng Học sâu phổ biến nhất hiện nay, có khả năng nhận dạng và phân loại hình ảnh với độ chính xác rất cao, thậm chí còn tốt hơn con người trong nhiều trường hợp. Mô hình này đã và đang được phát triển, ứng dụng vào các hệ thống xử lý ảnh lớn của Facebook, Google hay Amazon... cho các mục đích khác nhau như các thuật toán tagging tự động, tìm kiếm ảnh hoặc gợi ý sản phẩm cho người tiêu dùng.

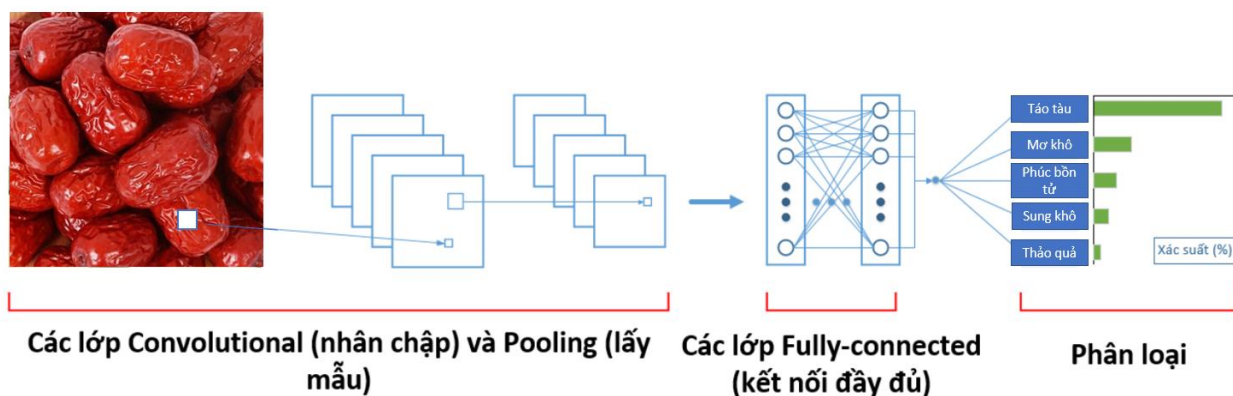
Sự ra đời của mạng CNN là dựa trên ý tưởng cải tiến cách thức các mạng nơ-ron nhân tạo truyền thống học thông tin trong ảnh. Do sử dụng các liên kết đầy đủ giữa các điểm ảnh vào node, các mạng nơ-ron nhân tạo truyền thẳng (Feedforward Neural Network) bị hạn chế rất nhiều bởi kích thước của ảnh, ảnh càng lớn thì số lượng liên kết càng tăng nhanh và kéo theo sự bùng nổ khối lượng tính toán. Ngoài ra sự liên kết đầy đủ này cũng là sự dư thừa khi với mỗi bức ảnh, các thông tin chủ yếu thể hiện qua sự phụ thuộc giữa các điểm ảnh với những điểm xung quanh nó mà không quan tâm nhiều đến các điểm ảnh ở cách xa nhau. Mạng CNN ra đời với kiến trúc thay đổi, có khả năng xây dựng liên kết chỉ sử dụng một phần cục bộ trong ảnh kết nối đến node trong lớp tiếp theo thay vì toàn bộ ảnh như trong mạng nơ-ron truyền thẳng.

2.1. Mở đầu chương

Trong chương này, nhóm sẽ trình bày cơ sở lý thuyết về mạng nơ-ron tích chập và ứng dụng mạng nơ-ron kinh điển AlexNet vào bài toán nhận dạng.

2.2. Kiến trúc mạng nơ-ron tích chập

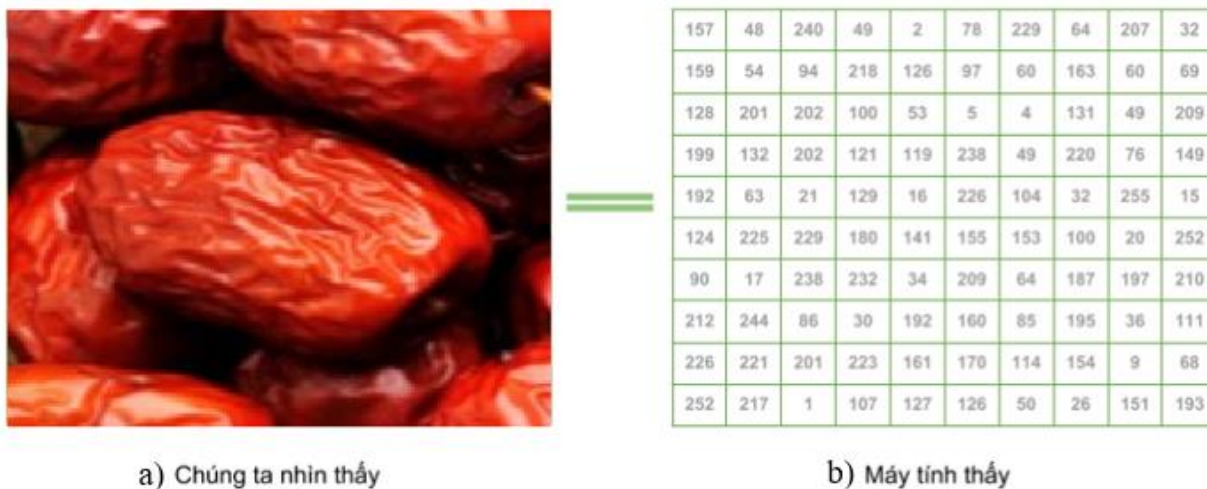
CNN thường có các lớp cơ bản: Lớp tích chập + lớp phi tuyến, lớp lấy mẫu, lớp kết nối đầy đủ. Các lớp này liên kết với nhau theo một thứ tự nhất định. Thông thường, một ảnh sẽ được lan truyền qua tầng gồm lớp tích chập + lớp phi tuyến đầu tiên, sau đó các giá trị tính toán được sẽ lan truyền qua lớp lấy mẫu, bộ ba lớp tích chập + lớp phi tuyến + lớp lấy mẫu có thể được lặp lại nhiều lần trong mạng và sau đó được lan truyền qua tầng gồm lớp kết nối đầy đủ và hàm trung bình mũ để tính xác suất ảnh đó chứa vật thể gì.



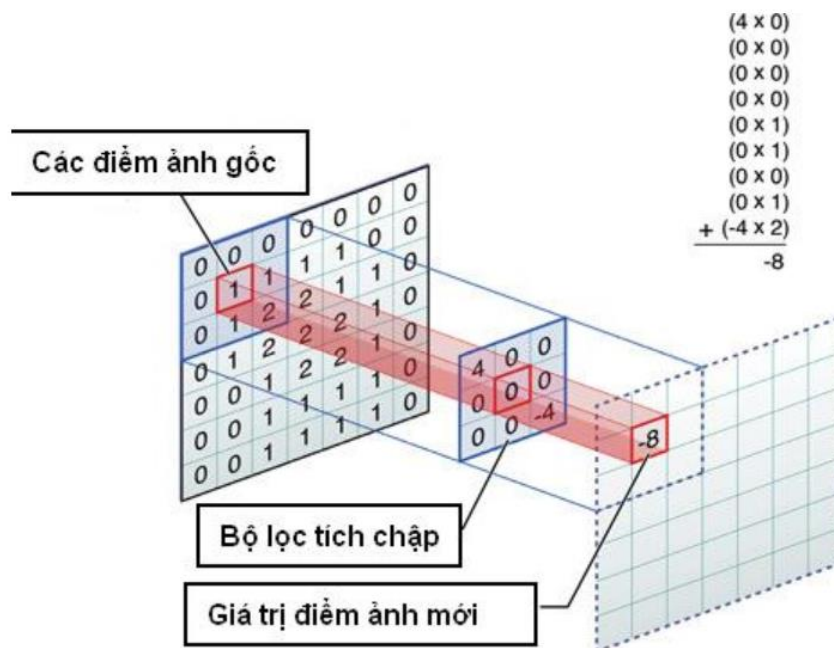
Hình 2.1. Kiến trúc cơ bản của một mạng tích chập

2.1.1. Lớp tích chập (Convolution Layer)

Convolution layer là lớp quan trọng nhất và cũng là lớp đầu tiên của mô hình CNN. Lớp này nhận đầu vào là một ma trận 3 chiều và lớp này có chức năng chính là phát hiện các đặc trưng cụ thể của bức ảnh [1]. Những đặc trưng này bao gồm đặc trưng cơ bản như góc, cạnh, màu sắc hoặc đặc trưng phức tạp hơn như kết cấu của ảnh. Dữ liệu đầu vào của bài toán là một bức ảnh. Một ảnh được biểu diễn bằng ma trận các giá trị (hình 2.2 (b)).



Hình 2.2. Biểu diễn ảnh đầu vào dưới dạng ma trận



Hình 2.3. Ví dụ bộ lọc tích chập được sử dụng trên ma trận điểm ảnh

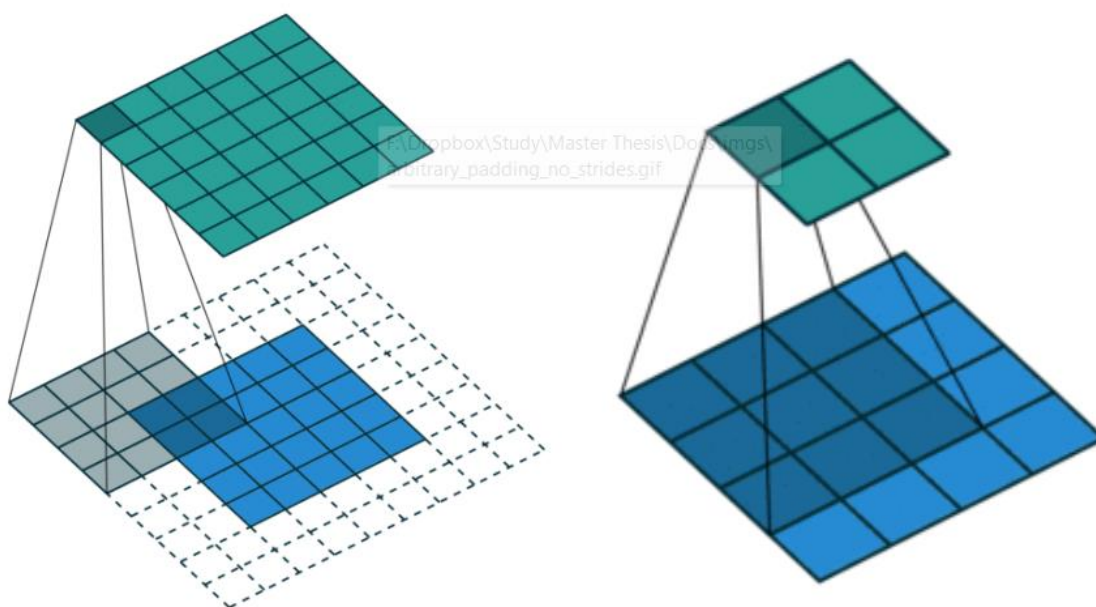
Kích thước bộ lọc (filter) của tầng convolution hầu hết đều là số lẻ, ví dụ như 3x3 hay 5x5. Với kích thước bộ lọc là số lẻ, các giá trị của feature map sẽ xác định một tâm điểm ở tầng phía trước. Nếu chọn bộ lọc có kích thước 2x2, 4x4 thì chúng ta sẽ gặp khó khăn khi muốn tìm vị trí tương ứng của các giá trị feature map trên không gian ảnh.

Trong ví dụ ở Hình 2.3, ta thấy bộ lọc được sử dụng là một ma trận có kích thước 3x3. Bộ lọc này được dịch chuyển lần lượt qua từng vùng ảnh đến khi hoàn thành quét toàn bộ bức ảnh, tạo ra một bức ảnh mới có kích thước nhỏ hơn hoặc bằng với kích thước ảnh đầu vào được gọi là feature map [1, 4]. Kích thước này được quyết định tùy theo kích thước các khoảng trắng được thêm ở viền bức ảnh gốc và được tính theo công thức:

$$W_{out} = \frac{W - F + 2P}{S} + 1 \quad (1)$$

Trong đó:

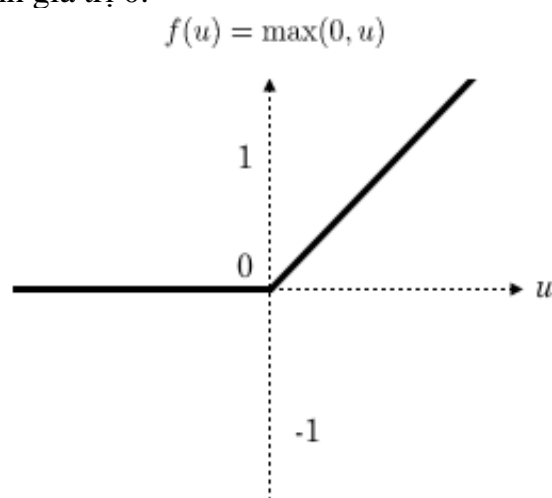
- W_{out} là kích thước ảnh đầu ra
- W là kích thước ảnh đầu vào
- F là kích thước bộ lọc
- P là kích thước khoảng trắng phía ngoài viền của ảnh gốc
- S là bước trượt của bộ lọc



Hình 2.4. Trường hợp thêm/không thêm viền trắng vào ảnh khi thực hiện tích chập

2.1.2. Hàm kích hoạt phi tuyến ReLU

Lớp này được xây dựng với ý nghĩa đảm bảo tính phi tuyến của mô hình huấn luyện sau khi đã thực hiện một loạt các phép tính toán tuyến tính qua các lớp Tích chập. Lớp Kích hoạt phi tuyến nói chung sử dụng các hàm kích hoạt phi tuyến như ReLU hoặc sigmoid, tanh... để giới hạn phạm vi biên độ cho phép của giá trị đầu ra. Trong số các hàm kích hoạt này, hàm ReLU được chọn do cài đặt đơn giản, tốc độ xử lý nhanh mà vẫn đảm bảo được tính toán hiệu quả. Cụ thể, phép tính toán của hàm ReLU chỉ đơn giản là chuyển tất cả các giá trị âm thành giá trị 0.



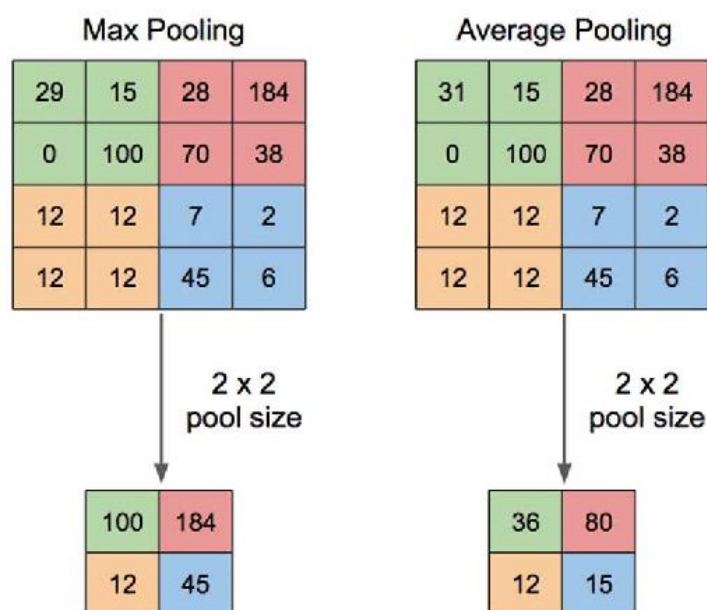
Hình 2.5. Đồ thị hàm phi tuyến ReLU

Thông thường, lớp ReLU được áp dụng ngay phía sau lớp Tích chập, với đầu ra là một ảnh mới có kích thước giống với ảnh đầu vào, các giá trị điểm ảnh cũng hoàn toàn tương tự trừ các giá trị âm đã bị loại bỏ.

2.1.3. Lớp lấy mẫu (Pooling Layer)

Một thành phần tính toán chính khác trong mạng CNN là lấy mẫu (Pooling), thường được đặt sau lớp Tích chập và lớp ReLU để làm giảm kích thước ảnh đầu ra trong khi vẫn giữ được các thông tin quan trọng của ảnh đầu vào. Việc giảm kích thước dữ liệu có tác dụng làm giảm được số lượng tham số cũng như tăng hiệu quả tính toán. Lớp lấy mẫu cũng sử dụng một cửa sổ trượt để quét toàn bộ các vùng trong ảnh tương tự như lớp Tích chập, và thực hiện phép lấy mẫu thay vì phép tích chập – tức là ta sẽ chọn lưu lại một giá trị duy nhất đại diện cho toàn bộ thông tin của vùng ảnh đó.

Hình 2.6 thể hiện các phương thức lấy mẫu thường được sử dụng nhất hiện nay, đó là Max Pooling (lấy giá trị điểm ảnh lớn nhất) và Average Pooling (lấy giá trị trung bình của các điểm ảnh trong vùng ảnh cục bộ) [12].



Hình 2.6. Phương thức Max Pooling và Average Pooling

Như vậy, với mỗi ảnh đầu vào được đưa qua lấy mẫu ta thu được một ảnh đầu ra tương ứng, có kích thước giảm xuống đáng kể nhưng vẫn giữ được các đặc trưng cần thiết cho quá trình tính toán sau này.

2.1.4. Lớp kết nối đầy đủ

Lớp kết nối đầy đủ này được thiết kế hoàn toàn tương tự như trong mạng nơ-ron truyền thống, tức là tất cả các điểm ảnh được kết nối đầy đủ với node trong lớp tiếp theo.

So với mạng nơ-ron truyền thống, các ảnh đầu vào của lớp này đã có kích thước được giảm bớt rất nhiều, đồng thời vẫn đảm bảo các thông tin quan trọng cho việc nhận dạng. Do vậy, việc tính toán nhận dạng sử dụng mô hình truyền thẳng đã không còn phức tạp và tốn nhiều thời gian như trong mạng nơ-ron truyền thống.

2.1.5. Thuật toán tối ưu sử dụng cho bài toán

Trong thuật toán học máy nói chung và kỹ thuật học sâu nói riêng, thuật toán tối ưu hóa là một khâu quan trọng không thể thiếu. Quá trình tối ưu hóa thực hiện xác định hàm mất mát (loss function) và sau đó tối thiểu hóa hàm trên bằng cách sử dụng hàm tối ưu. Cụ thể, thông qua việc cập nhật các tham số của mô hình (w , b) và đánh giá lại hàm mất mát với một tỉ lệ học (learning rate) xác định [8], quá trình tối ưu giúp mô hình tương thích tốt hơn với tập dữ liệu được huấn luyện.

2.2. Mạng AlexNet

Mạng AlexNet là công trình đầu tiên phổ biến mạng CNN trong lĩnh vực Thị giác máy tính, cũng là một trong những mạng huấn luyện CNN nổi tiếng nhất nhờ thành tích ấn tượng mà nó đạt được trong cuộc thi nhận dạng ảnh quy mô lớn tổ chức vào năm 2012 [11]. Mục đích của cuộc thi là nhằm thử nghiệm các công nghệ mới giúp cho máy tính có thể hiểu, phân tích, phát hiện và nhận dạng các vật thể trong một bức ảnh.

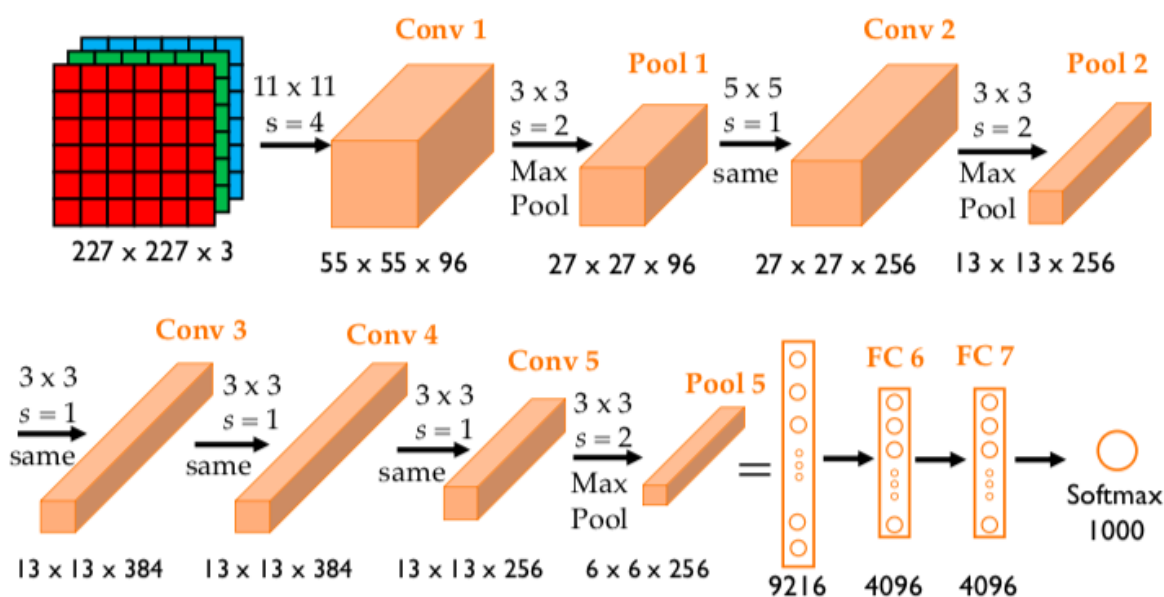
Cụ thể hơn, nhiệm vụ chính của cuộc thi năm 2012 đặt ra mà các đội tham gia phải giải quyết là bài toán nhận dạng, với bộ dữ liệu huấn luyện lên đến 1,2 triệu ảnh được gán nhãn cho 1.000 hạng mục khác nhau. Nhóm SuperVision, gồm các thành viên Alex Krizhevsky, Ilya Sutskever và Geoff Hinton, cùng với mạng AlexNet của họ đã đạt được kết quả đáng kinh ngạc là chiến thắng áp đảo nhóm đứng thứ hai với độ chính xác chênh lệch đến hơn 10% (15,31% và 26,17%) [9]. Điều đặc biệt là mạng huấn luyện này chỉ nhận dữ liệu đầu vào là các giá trị điểm ảnh thô và không hề áp dụng bất kỳ phương pháp trích chọn đặc trưng nào, trong khi mọi hệ thống nhận dạng thị giác truyền thống đều phải gồm nhiều giai đoạn trích chọn đặc trưng hết sức tỉ mỉ, cẩn thận, thậm chí phải áp dụng nhiều mẹo để cải thiện chất lượng nhận dạng. Thiết kế kiến trúc mạng huấn luyện gần như một hộp đen, cộng với khả năng tự học các đặc trưng thông qua các lớp ẩn, đã khiến CNN nổi riêng và Học sâu nói chung trở thành giải pháp mạnh mẽ nhất cho bài toán nhận dạng và phân loại vật thể cho tới bây giờ.

Từ năm 2012, mạng CNN trở thành cái tên gắn liền với cuộc thi và đã có rất nhiều mạng CNN nổi bật khác xuất hiện trong những năm sau đó VGG, GoogleNet hay Microsoft ResNet... Các mạng CNN càng ngày càng đạt độ chính xác cao hơn, tuy nhiên chúng có độ phức tạp và độ sâu lớn hơn rất nhiều, ví dụ mạng CNN có thể coi là tốt nhất hiện nay – ResNet – đã sử dụng đến 152 lớp tính toán. Sự phức tạp này yêu cầu khả năng tính toán lớn, thời gian huấn luyện lâu, và gây nhiều khó khăn trong việc cài đặt triển khai hệ thống, do đó mạng AlexNet đã được chọn làm cơ sở phát triển phiên bản thử nghiệm ban đầu và

việc cài đặt các mạng huấn luyện khác nhằm nâng cao chất lượng nhận dạng của hệ thống sẽ được thử nghiệm trong tương lai.

Phần 2.2.1 sẽ trình bày kỹ hơn về kiến trúc tổng thể của mạng AlexNet cũng như cách thức ứng dụng nó vào bài toán nhận dạng quả khô trong thuốc Đông Y.

2.2.1. Kiến trúc mạng AlexNet



Hình 2.7. Kiến trúc mạng AlexNet

Hình 2.7 thể hiện rõ kiến trúc tổng thể của mạng AlexNet, trong đó:

- **Lớp 1** (Tích chập):
 - Đầu vào: Ảnh với kích thước 227 x 227 x 3 (3 là con số tương ứng với 3 màu đỏ, xanh lục, xanh lam trong hệ màu RGB thông thường)
 - Số bộ lọc: 96
 - Kích thước bộ lọc: 11 x 11 x 3
 - Bước trượt (Stride): 4
 - Đầu ra: $(\frac{227-11}{4} + 1) \times (\frac{227-11}{4} + 1) \times 96 = 55 \times 55 \times 96$
- **Lớp 2** (lớp MaxPooling)
 - Đầu vào: 55 x 55 x 96
 - Đầu ra: $(55/2) \times (55/2) \times 96 = 27 \times 27 \times 96$
- **Lớp 3** (Tích chập):

- Đầu vào: $27 \times 27 \times 96$
 - Số bộ lọc: 256
 - Kích thước bộ lọc: $5 \times 5 \times 48$
 - Đầu ra: $27 \times 27 \times 256$
- **Lớp 4, 5, 6:** Tương tự như với lớp 1 và lớp 2 với các kích thước bộ lọc lần lượt là $3 \times 3 \times 256$, $3 \times 3 \times 384$ và $3 \times 3 \times 384$. Toàn bộ các lớp tính toán này đều được chia đều cho hai GPU để tăng tốc độ xử lý. Đầu ra cuối cùng qua lớp Tích chập thứ 5 là dữ liệu với kích thước $13 \times 13 \times 256$, dữ liệu này sau khi đi qua một lớp Lấy mẫu tối đa cuối cùng sẽ được dùng làm đầu vào cho các lớp sau đó là các lớp Kết nối đầy đủ.
- **Lớp 6 (Kết nối đầy đủ):**
- Đầu vào: $6 \times 6 \times 256$
 - Số nơ-ron: 4096
- **Lớp 7 (Kết nối đầy đủ):** Tương tự lớp 6.
- **Lớp 8 (Kết nối đầy đủ):** Lớp cuối cùng trong mạng AlexNet này có 1000 nơ-ron, tương ứng với 1000 lớp khác nhau mà bộ huấn luyện cần nhận dạng. Ta có thể nhìn rõ hơn kiến trúc mạng AlexNet ở dạng bảng như trong Bảng 2.8:

Bảng 2.1. Bảng tổng kết của mô hình AlexNet

Layer		Feature Map	Size	Kernel Size	Stride	Activation
Input	Image	1	$227 \times 227 \times 3$	-	-	-
1	Convolution	96	$55 \times 55 \times 96$	11×11	4	relu
	Max Pooling	96	$27 \times 27 \times 96$	3×3	2	relu
2	Convolution	256	$27 \times 27 \times 256$	5×5	1	relu
	Max Pooling	256	$13 \times 13 \times 256$	3×3	2	relu
3	Convolution	384	$13 \times 13 \times 384$	3×3	1	relu
4	Convolution	384	$13 \times 13 \times 384$	3×3	1	relu
5	Convolution	256	$13 \times 13 \times 256$	3×3	1	relu
	Max Pooling	256	$6 \times 6 \times 256$	3×3	2	relu
6	FC	-	9216	-	-	relu
7	FC	-	4096	-	-	relu
8	FC	-	4096	-	-	relu
Output	FC	-	1000	-	-	Softmax

2.2.2. Ứng dụng mạng AlexNet vào bài toán nhận dạng các quả khô trong thuốc Đông Y

Nhóm em áp dụng mô hình AlexNet cho phương hướng giải quyết bài toán nhận dạng quả khô thuốc Đông Y như sau:

- 1) Cài đặt mạng AlexNet với bộ dữ liệu gồm 5000 hình train và 600 hình test
- 2) Xây dựng bộ CSDL ảnh huấn luyện cho 5 loại quả khô (táo tàu, sung khô, mơ khô, phúc bồn tử, thảo quả) với ảnh được chọn lựa theo tiêu chuẩn về kích thước, màu sắc đồng thời được gán nhãn bằng phương pháp One-hot-encoding.
- 3) Quan sát kết quả đánh giá mô hình và điều chỉnh thông số để lựa chọn ra mô hình phù hợp cho đề tài.

2.3. Kết luận chương

Qua chương này, nhóm đã trình bày cơ sở lý thuyết về mạng nơ-ron tích chập CNN. Đồng thời tìm hiểu được kiến trúc của mạng AlexNet và ứng dụng mạng AlexNet vào bài toán nhận dạng các loại quả khô trong thuốc Đông Y. Từ đó nhóm sẽ đưa ra các phương pháp, cách thức thực hiện hệ thống ở chương tiếp theo.

CHƯƠNG 3: THIẾT KẾ MÔ HÌNH, THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. Mở đầu chương

Trong chương này nhóm sẽ trình bày về quá trình thực hiện xây dựng cơ sở dữ liệu, thiết kế và đánh giá mô hình huấn luyện giữa mô hình sẵn có như Alexnet và các mô hình tự xây dựng.

3.2. Quy trình thực hiện hệ thống

Hệ thống nhận dạng quả khô trong thuốc Đông Y sẽ được xây dựng thành một hệ thống hoàn chỉnh thông qua các bước sau:

- Bước 1: Thu thập và xây dựng cơ sở dữ liệu
- Bước 2: Xây dựng mô hình cho hệ thống
- Bước 3: Kiểm tra và đánh giá mô hình huấn luyện
- Bước 4: Triển khai tích hợp hệ thống AI vào các hệ thống khác

3.3. Thu thập và xây dựng cơ sở dữ liệu

3.3.1. Cơ sở dữ liệu ảnh các loại quả khô trong thuốc Đông Y

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. Do CSDL quả khô không có sẵn nên nhóm phải tự thu thập và xây dựng bộ CSDL mới nên kích thước CSDL sẽ được hạn chế, cả về số lượng loại quả nhận dạng cũng như số lượng ảnh chụp cho mỗi loại hoa quả đó. Cụ thể: Số lượng hoa quả sẽ nhận dạng: 5 loại quả phổ biến như táo đỏ, sung khô, mơ khô, phúc bồn tử, thảo quả.

Tổng kích thước của bộ CSDL gốc là 600 ảnh, chia đều cho mỗi loại quả: 120 ảnh, bao gồm các ảnh chụp quả khô ở các góc độ khác nhau với nền trơn hoặc không có nền, lấy từ nguồn trên mạng. Sau đó nhóm sẽ thực hiện điều chỉnh kích thước ảnh thành ảnh vuông, cắt ảnh và lọc dữ liệu để tạo thêm ảnh mới nhằm tăng cường kích thước cơ sở dữ liệu.

3.3.2. Sử dụng thuật toán tăng cường dữ liệu

Do CSDL thu thập được có kích thước nhỏ nên nhóm đã sử dụng phương pháp tăng cường dữ liệu bằng hàm có sẵn trong thư viện Keras là ImageDataGenerator và sử dụng phương thức `flow_from_directory()` để đọc các ảnh trong thư mục chứa ảnh. Ảnh sẽ được tăng cường bằng các phương pháp như:

- Thực hiện phóng to ngẫu nhiên
- Thay đổi cường độ sáng
- Xoay ảnh theo các góc độ khác nhau
- Lật ảnh ngẫu nhiên theo chiều ngang, dọc
- Dịch ảnh theo chiều cao, chiều rộng



Hình 3.1. Tăng cường dữ liệu với thảo quả

Dữ liệu sau khi được tăng cường sẽ có tổng cộng là 5600 ảnh chia đều cho 5 lớp. Mỗi lớp sẽ có 1120 ảnh.

Để thực hiện việc phân chia dữ liệu nhóm đã dùng phương pháp Hold out để phân chia bộ dữ liệu ban đầu. Dữ liệu sẽ được chia thành 2 phần: Huấn luyện và Kiểm tra. Số lượng ảnh trong tập huấn luyện là 5000 bức, ảnh trong tập kiểm tra là 600 bức.

Phương pháp gán nhãn mà nhóm đã sử dụng là One-hot-encoding. Việc gán nhãn dựa vào TÊN CỦA THUỐC chứa ảnh của mỗi loại quả.

Name	Date modified	Type	Size
📁 Mơ Khô	10/7/2022 4:53 PM	File folder	
📁 Phúc Bồn Tử	10/20/2022 3:50 PM	File folder	
📁 Sung Khô	10/7/2022 4:52 PM	File folder	
📁 Táo Tàu	10/21/2022 11:40 PM	File folder	
📁 Thảo Quả	10/7/2022 4:53 PM	File folder	

Hình 3.2. Sắp xếp thư mục dữ liệu cho việc gán nhãn

5 giá trị sẽ được mã hóa thành 5 mã One-hot có chiều dài 5 bit nhị phân

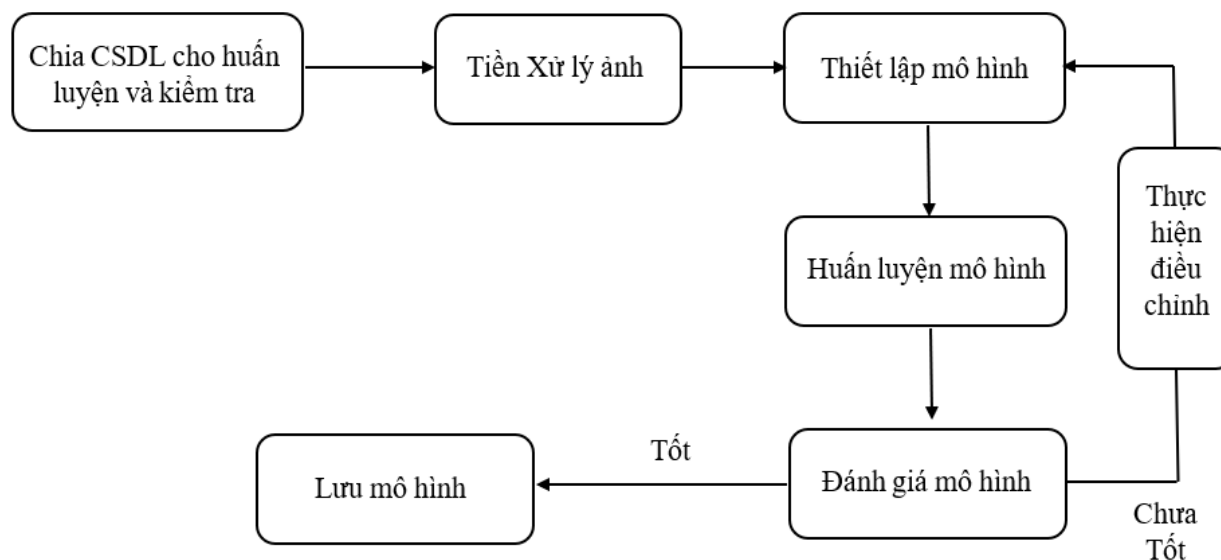
Bảng 3.1. Phương pháp gán nhãn One-hot-encoding

Giá trị	Mã One-hot
‘Mơ khô’	1 0 0 0
‘Phúc bồn tử’	0 1 0 0
‘Sung khô’	0 0 1 0
‘Táo tàu’	0 0 0 1
‘Thảo quả’	0 0 0 1

3.4. Xây dựng mô hình huấn luyện

Để đưa ra đánh giá tổng quát và so sánh độ chính xác tương đối, đồ án thực hiện cài đặt một mạng huấn luyện nơ-ron nhân tạo kinh điển Alexnet và các mạng huấn luyện nơ-ron tích chập đã được điều chỉnh. Sau khi thực hiện huấn luyện các mô hình trên cùng một bộ cơ sở dữ liệu sẽ so sánh kết quả đạt được để chọn ra mô hình tối ưu.

Quá trình huấn luyện dữ liệu thông thường sẽ yêu cầu cân chỉnh một số tham số như: Filter, Epoch, Batch size, ... Số lượng tham số, giá trị tham số thay đổi sẽ cho kết quả huấn luyện khác nhau. Việc lựa chọn thông số phù hợp sẽ tốn thời gian thực nghiệm nhưng cũng chính là điều phải làm để đạt được kết quả tối ưu.

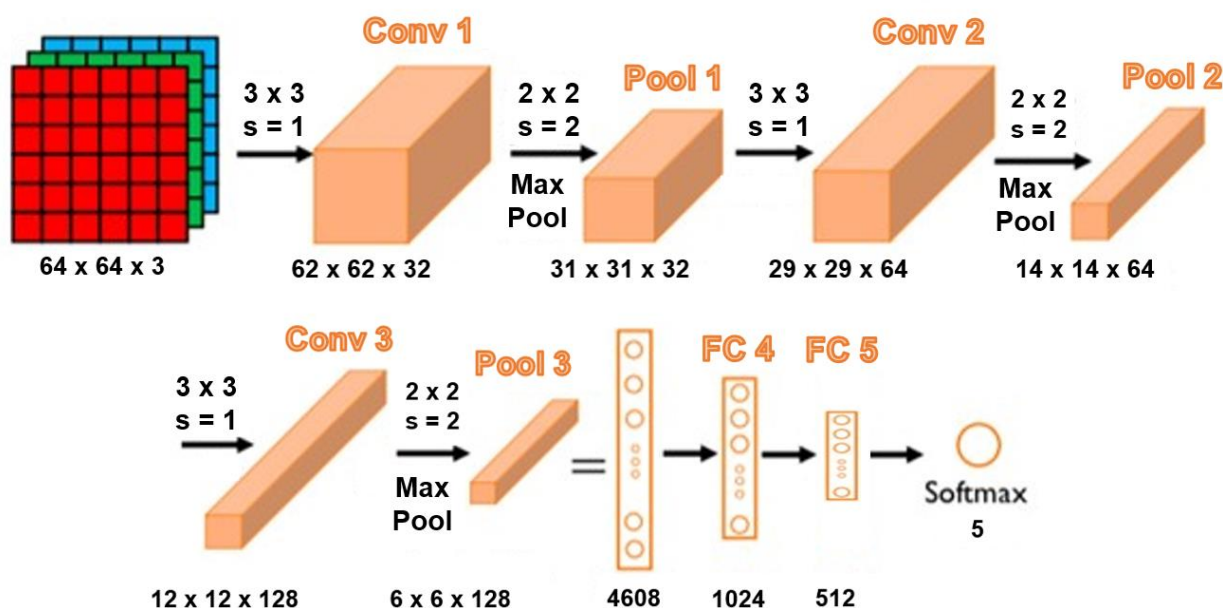


Hình 3.3. Quy trình xây dựng mô hình huấn luyện

Bảng 3.2. Các mô hình huấn luyện thử nghiệm

Mô hình 1: Mô hình Alexnet nguyên bản
Mô hình 2: Mô hình Alexnet giảm số bộ lọc các lớp tích chập
Mô hình 3: Mô hình 2 giảm chỉ số pool_size
Mô hình 4: Mô hình 3 giảm kích thước ảnh đầu vào
Mô hình 5: Mô hình 4 bỏ lớp BatchNormalization
Mô hình 6: Mô hình 5 bỏ lớp convolution số 4
Mô hình 7: Mô hình 6 bỏ lớp convolution số 3
Mô hình 8: Mô hình 7 giảm số lớp Fully Connected
Mô hình 9: Mô hình 7 thay đổi số node Fully Connected 4096 thành 1024 và 512
Mô hình 10: Mô hình 9 Thêm lớp BatchNormalization vào giữa lớp tích chập và pool của mỗi lớp

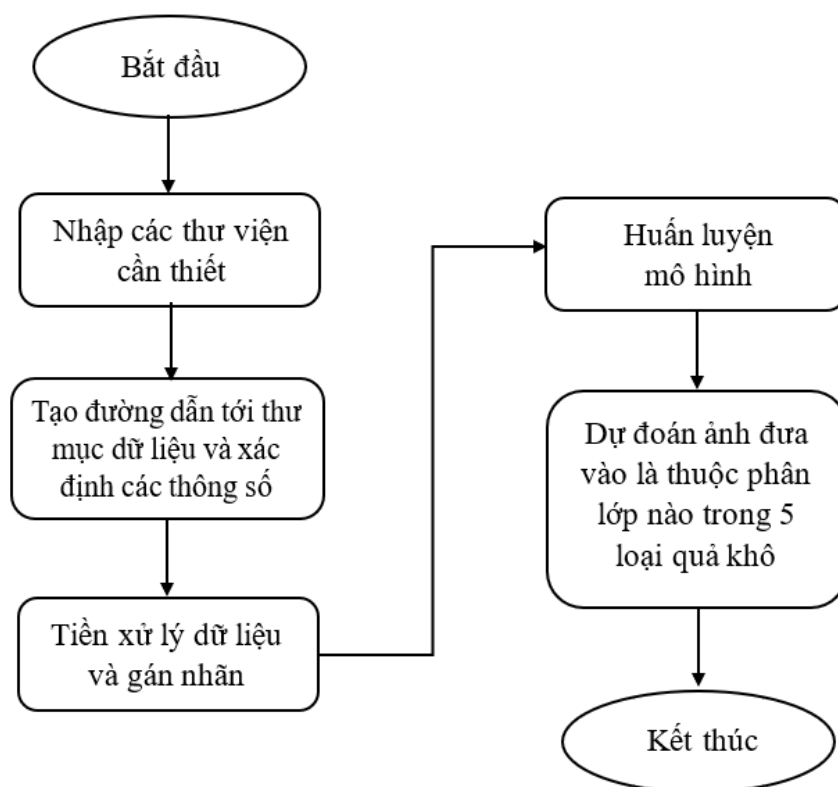
Sau nhiều lần thực nghiệm và đánh giá nhóm đã nhận thấy mô hình 9 là mô hình cho kết quả huấn luyện tối ưu nhất về độ chính xác (Accuracy) và thời gian huấn luyện.



Hình 3.4. Kiến trúc mạng CNN của mô hình 9

Hình 3.4 mô tả mô hình CNN cho bài toán phân lớp trái cây khô. Bài toán phân lớp trái cây khô bao gồm 5 khối thực thi chính: Khối đầu tiên sử dụng đầu vào là các hình ảnh thuộc tập dữ liệu huấn luyện để rút trích các đặc trưng cơ bản với bộ lọc có kích thước 3×3 và tiến hành max-pooling để giảm mẫu biểu đồ đặc trưng. 2 khối tiếp theo cũng rút trích các đặc trưng với số lượng bộ lọc tăng dần và tiến hành max-pooling. Mô hình sử dụng bộ lọc có kích thước nhỏ, số lượng bộ lọc ở từng tầng tích chập tăng theo bội số của 2 (32, 64, 128) giúp mô hình học ngày càng sâu hơn và để kích hoạt biểu đồ đặc trưng sau mỗi lần tích chập thì nhóm sử dụng hàm kích hoạt ReLu. Bên cạnh đó, hàm max-pooling cũng được sử dụng sau mỗi tầng tích chập để vừa giảm kích thước vừa tăng cường biểu đồ đặc trưng. Khối cuối cùng là các tầng FC, được giảm nhỏ dần để tiến tới vector giá trị cuối cùng có độ dài bằng với số lượng phân lớp ban đầu, do sử dụng nhiều bộ lọc nên sẽ thực hiện Dropout giữa các tầng FC để tránh xảy ra tình trạng Overfitting.

Mô hình CNN đề xuất sẽ được hiện thực hóa trên Google Colab với ngôn ngữ Python. Nhóm sẽ sử dụng một số bộ thư viện hỗ trợ trong việc xây dựng mô hình và hoàn thiện hệ thống phân loại ảnh sử dụng trong ngôn ngữ Python như Tensorflow, Keras, Sklearn.



Hình 3.5. Sơ đồ khối của hệ thống nhận diện trái cây khô trong thuốc Đông Y

3.5. Đánh giá các mô hình huấn luyện

Đánh giá hiệu quả của một mô hình máy học là phải đo được mức độ chính xác mà mô hình trả về sau quá trình huấn luyện với bộ dữ liệu. Thường người ta sẽ tập trung vào khả năng dự đoán của mô hình hơn là tốc độ phân loại hay xây dựng mô hình.

Để đánh giá hiệu suất huấn luyện của mô hình CNN, nhóm đã dựa trên các tiêu chí đánh giá mô hình như:

- Thời gian huấn luyện cho mô hình
- Biểu đồ thể hiện độ chính xác (Accuracy) và lỗi (Loss)
- Các chỉ số Accuracy, Precision, Recall, F1 score, ROC AUC
- Ma trận nhầm lẫn

Bảng 3.3. Kết quả đánh giá các mô hình thử nghiệm theo độ chính xác và thời gian

Các mẫu thử nghiệm (Model)	Accuracy	Time
Mô hình 1: Mô hình Alexnet nguyên bản	85%	42m 42s
Mô hình 2: Mô hình Alexnet giảm số bộ lọc các lớp tích chập	68%	9m 06s
Mô hình 3: Mô hình 2 giảm chỉ số pool_size	76%	8m 57s
Mô hình 4: Mô hình 3 giảm kích thước ảnh đầu vào	79%	4m 37s
Mô hình 5: Mô hình 4 bỏ lớp BatchNormalization	84%	4m 47s
Mô hình 6: Mô hình 5 bỏ lớp convolution số 4	84%	5m 29s
Mô hình 7: Mô hình 6 bỏ lớp convolution số 3	85%	5m 23s
Mô hình 8: Mô hình 7 giảm số lớp Fully Connected	79%	3m 43s
Mô hình 9: Mô hình 6 thay đổi số node Fully Connected 4096 thành 1024	89%	3m 22s
Mô hình 10: Mô hình 9 Thêm lớp BatchNormalization vào giữa lớp tích chập và pool của mỗi lớp	67%	3m 52s

Dựa vào kết quả đánh giá ở bảng 3.3, mô hình 9 cho thấy kết quả đánh theo hai tiêu chí về độ chính xác và thời gian là tối ưu nhất trong tất cả các mô hình được đánh giá. Vì vậy nhóm đã chọn mô hình 9 làm mô hình chính thức của hệ thống nhận diện quả khô và tiếp tục sử dụng mô hình này để đánh giá các tiêu chí tiếp theo.

Để tính toán các chỉ số về độ chính xác, Precision, Recall, F1 score, ROC AUC ta dựa trên 4 giá trị kết quả sau:

- TP (true positive): Số lượng ảnh trái cây khô được phân loại chính xác
- FP (false positive): Số lượng ảnh bị phân loại nhầm sang trái cây khô khác
- TN (true negative): Số lượng mẫu không thuộc phân lớp và được dự đoán đúng
- FN (false negative): Số lượng mẫu thuộc phân lớp nhưng lại dự đoán là không

Độ chính xác (Accuracy) được hiểu là những dự đoán đúng của mô hình trên tổng số kết quả dự đoán được. Công thức để tính độ chính xác (Accuracy) của mô hình:

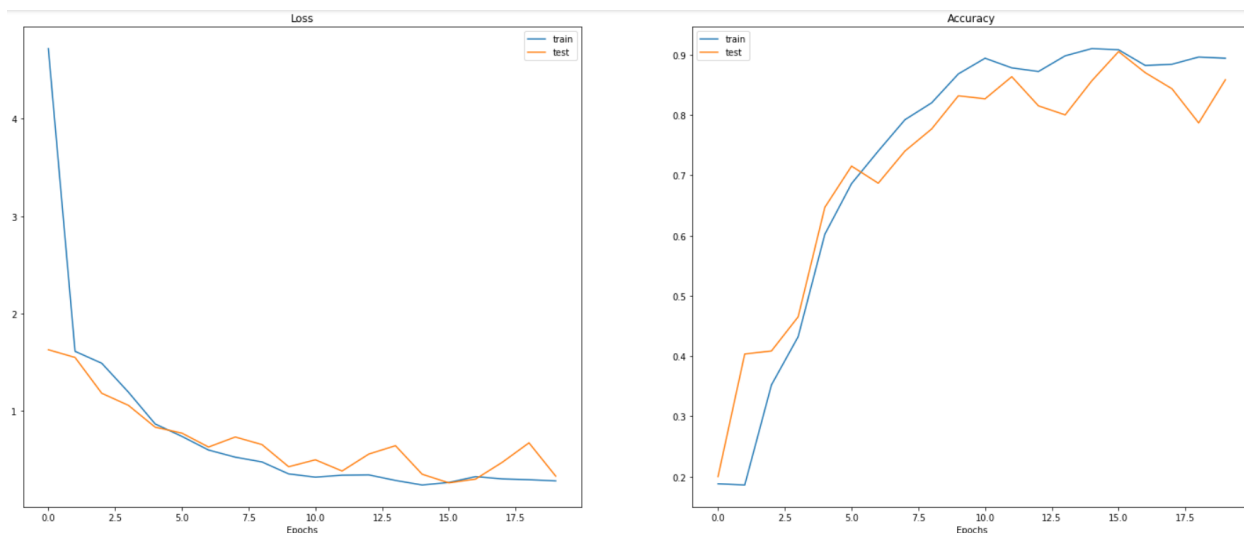
$$\text{Độ chính xác} = \frac{TP + TN}{TP + TN + FP + FN} \quad (2)$$

Dựa vào độ chính xác ta sẽ có thêm khái niệm độ lỗi (Error_rate) của mô hình sẽ được tính bằng công thức: $\text{Độ lỗi} = 1 - \text{Độ chính xác}$ (3)

```
Epoch 18/20
5/5 [=====] - 8s 2s/step - loss: 0.3048 - accuracy: 0.8840 - val_loss: 0.4753 - val_accuracy: 0.8433
Epoch 19/20
5/5 [=====] - 10s 2s/step - loss: 0.2962 - accuracy: 0.8960 - val_loss: 0.6747 - val_accuracy: 0.7867
Epoch 20/20
5/5 [=====] - 8s 2s/step - loss: 0.2844 - accuracy: 0.8940 - val_loss: 0.3332 - val_accuracy: 0.8583
```

Hình 3.6. Biểu đồ thể hiện Accuracy và Loss của mô hình 9

Dựa vào hình 3.6 ta có thể thấy độ chính xác của tập huấn luyện là 89% và của tập kiểm tra là 85%. Tuy nhiên, lỗi vẫn còn khá lớn khoảng 30%. Để có cái nhìn tổng quát hơn thì ta sử dụng biểu đồ biểu diễn độ chính xác và độ lỗi.



Hình 3.7. Biểu đồ thể hiện Accuracy và Loss của mô hình 9

Đường màu xanh là huấn luyện (train), màu cam là kiểm tra (test). Dựa vào biểu đồ hình 3.7, ta có thể đánh giá được kết quả huấn luyện của mô hình khá tốt. Biểu đồ lỗi (loss) có xu hướng giảm dần về 0 còn biểu đồ độ chính xác (Accuracy) có xu hướng tăng dần lên 1. Ở cả hai biểu đồ, đường huấn luyện và kiểm tra nằm sát gần nhau cho thấy mô hình không bị hiện tượng overfitting hay underfitting.

Precision là tỷ lệ kết quả dự đoán đúng phân lớp trên tất cả dự đoán đúng của toàn mô hình. Giá trị ta quan tâm ở đây, với một phân lớp bất kỳ thì kết quả mà mô hình có thể dự đoán đúng là bao nhiêu, còn việc mô hình đánh giá dữ liệu này thuộc một phân lớp nào đó khác thì không quan trọng. Giá trị này được xác định bằng công thức:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (4)$$

Recall là tỷ lệ kết quả dự đoán đúng trên tổng số dữ liệu thuộc về phân lớp đã được biết. Tỷ lệ này rất quan trọng, nếu với một bộ dữ liệu đầu vào ta đã phân lớp sẵn, thì việc mô hình bỏ mất nhiều dữ liệu phân lớp này ảnh hưởng nhiều đến hiệu quả của mô hình.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (5)$$

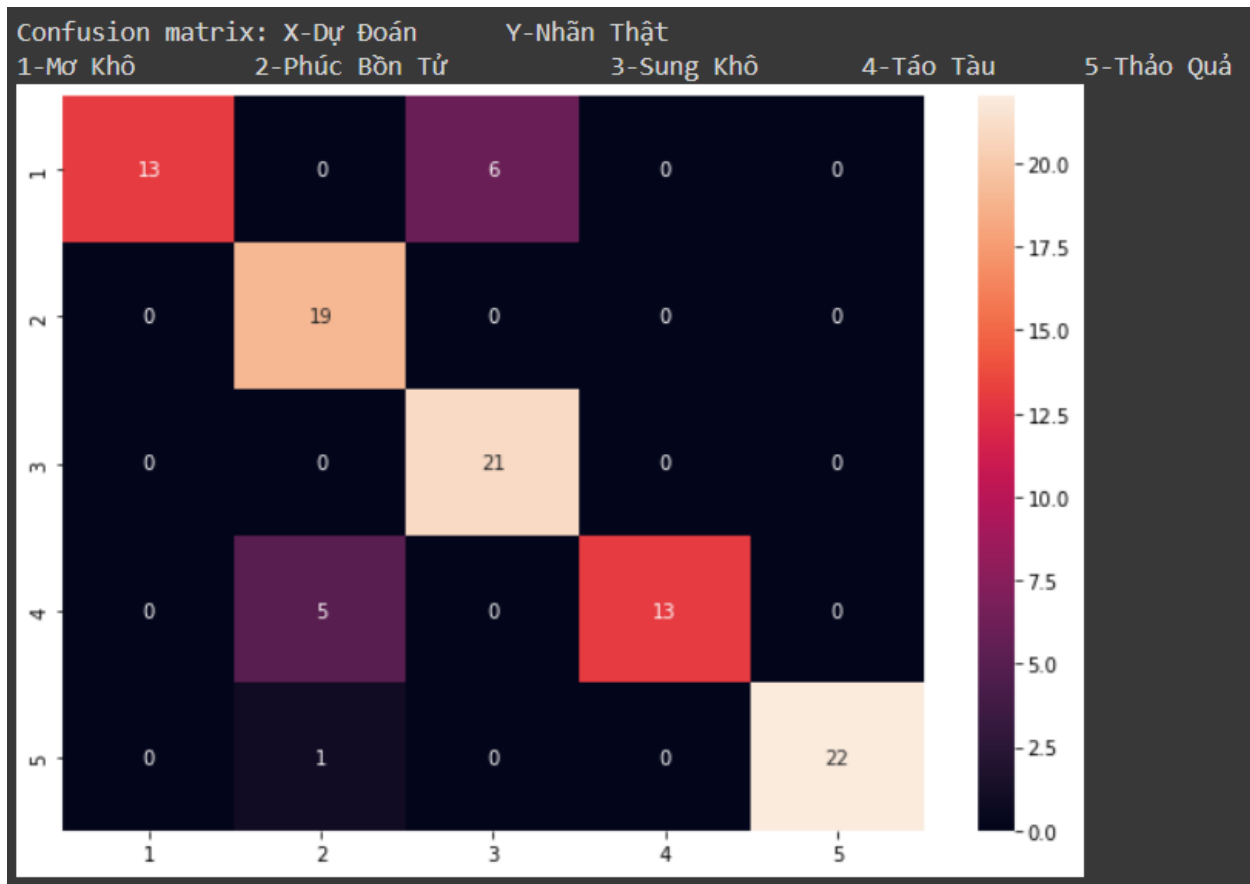
F1 score là giá trị dùng để cân bằng 2 giá trị Precision và Recall. Giá trị F1 càng cao tức là mô hình càng tốt, tỷ lệ dự đoán đúng cao và tỷ lệ không bị bỏ mất dữ liệu cao.

$$\text{F1 score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (6)$$

```
4/4 [=====] - 0s 56ms/step
Accuracy: 0.880000
Precision: 0.907556
Recall: 0.872591
F1 score: 0.873525
ROC AUC: 0.991596
```

Hình 3.8. Kết quả tính toán các chỉ số đánh giá của mô hình 9

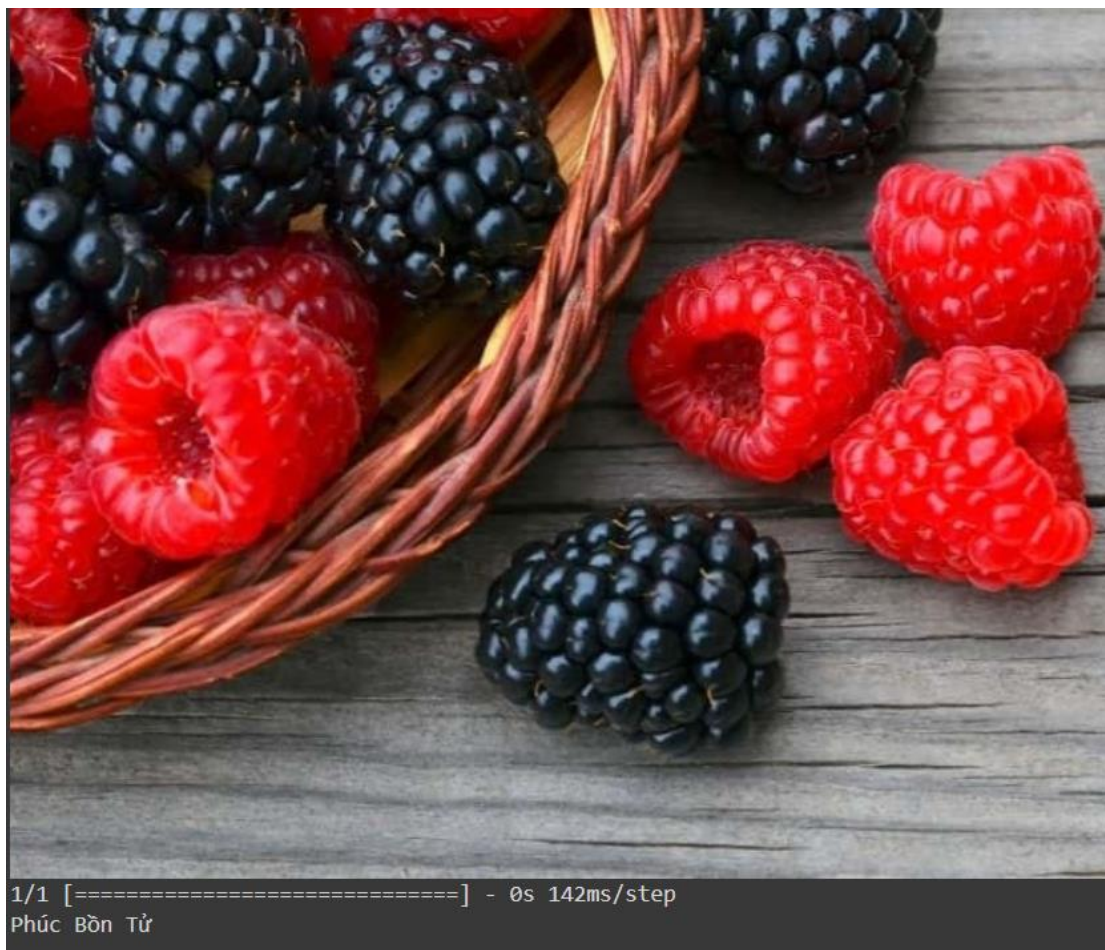
Ma trận nhầm lẫn là một loại biểu đồ thể hiện sự tương quan giữa giá trị trả về của phân lớp dự đoán so với phân lớp thực tế ban đầu. Trên thực tế người ta biểu diễn ma trận nhầm lẫn tương quan giữa tất cả phân lớp của toàn tập dữ liệu để quan sát bao nhiêu dữ liệu được phân lớp đúng và nếu sai thì nó sẽ đưa vào phân lớp nào từ đó điều chỉnh các thông số của mô hình, tập dữ liệu cho phù hợp.[10]



Hình 3.9. Ma trận nhầm lẫn

Dựa vào hình 3.9 có thể thấy mô hình nhận diện tốt nhất đối với 2 loại quả là Phúc bồn tử và Sung khô. Với 19 hình Phúc bồn tử được đưa vào để kiểm tra thì nhận diện đúng cả 19 hình, tương tự với Sung khô. Loại quả cho kết quả nhận diện kém là Mơ khô và Táo

tàu với 19 hình Mơ khô đưa vào kiểm tra thì chỉ nhận diện đúng được 13 hình, 6 hình còn lại bị nhận diện nhầm thành Sung khô.



Hình 3.10. Kiểm tra nhận diện bằng hình ảnh

3.6. Kết luận chương

Qua chương này, nhóm đã trình bày được quá trình thực hiện xây dựng cơ sở dữ liệu, thiết kế và đánh giá các mô hình huấn luyện dựa trên nhiều tiêu chí khác nhau. So sánh được độ hiệu quả của từng mô hình một đối với bộ cơ sở dữ liệu mà nhóm đã xây dựng.

KẾT LUẬN HƯỚNG PHÁT TRIỂN CỦA ĐỀ TÀI

Sau khi đánh giá hệ thống dựa trên các tiêu chí đã trình bày ở chương 3 nhóm đưa ra kết luận như sau:

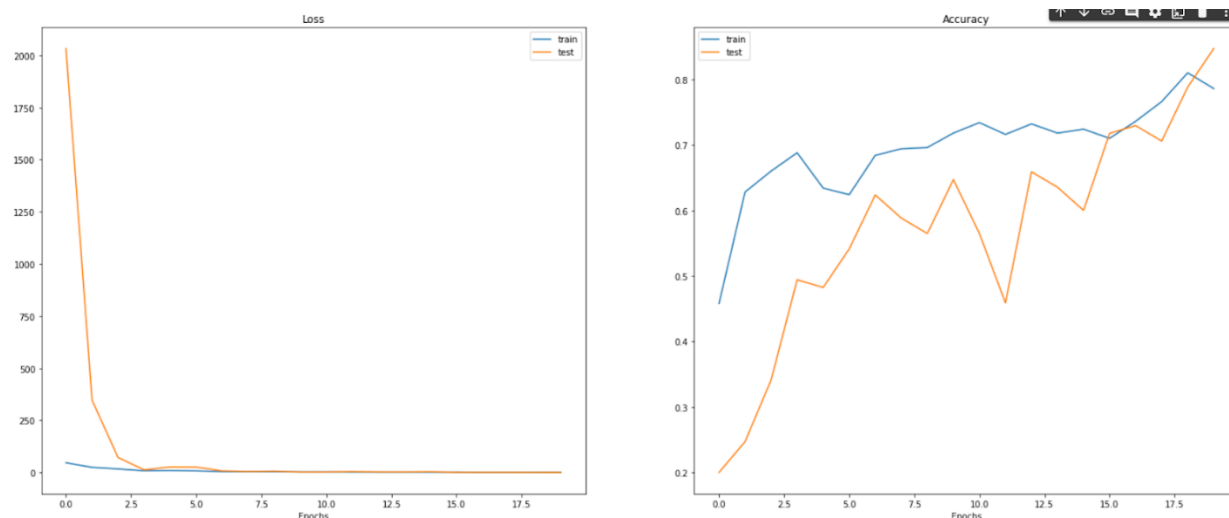
Mô hình hệ thống mà nhóm đã xây dựng cho kết quả huấn luyện tương đối tốt với độ chính xác lên đến 87%, các chỉ số Precision, Recall, F1 score, ROC AUC cho kết quả khá cao đều trên 80%. Mô hình không bị các hiện tượng overfitting hay underfitting. Từ đó có thể thấy mô hình mà nhóm đã xây dựng là phù hợp với đề tài.

Bên cạnh đó còn có một vài điểm mà nhóm cần cải thiện cho mô hình như lỗi (loss) còn khá cao 28% và vẫn còn một số loại quả khô cho kết quả nhận diện chưa tốt. Nguyên nhân là do tập dữ liệu mà nhóm xây dựng có nhiều hình có độ khác biệt quá lớn, dữ liệu dùng để huấn luyện chưa phong phú và đa dạng, phương pháp chia dữ liệu chưa phù hợp.

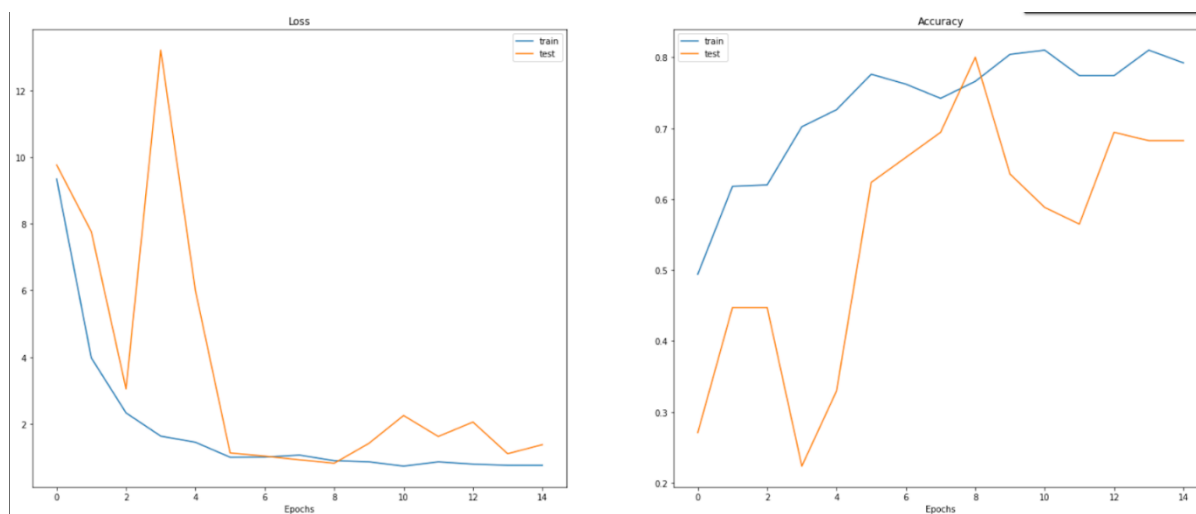
Trong tương lai thì nhóm có dự định đầu tiên là sẽ cải thiện hệ thống bằng cách tăng thêm dữ liệu huấn luyện, làm sạch dữ liệu, sử dụng phương pháp chia dữ liệu khác phù hợp hơn khi có ít dữ liệu huấn luyện như K-fold cross validation. Sau đó sẽ tích hợp hệ thống lên Website và ứng dụng di động. Chức năng của hệ thống sẽ là nhận diện được các loại quả khô và cho biết công dụng của từng loại quả. Ứng dụng nâng cao hơn, có thể đưa ra các bài thuốc trị bệnh dựa vào loại quả vừa nhận diện được. Dữ liệu đưa vào nhận diện có thể là dạng hình ảnh hoặc video.

PHỤ LỤC

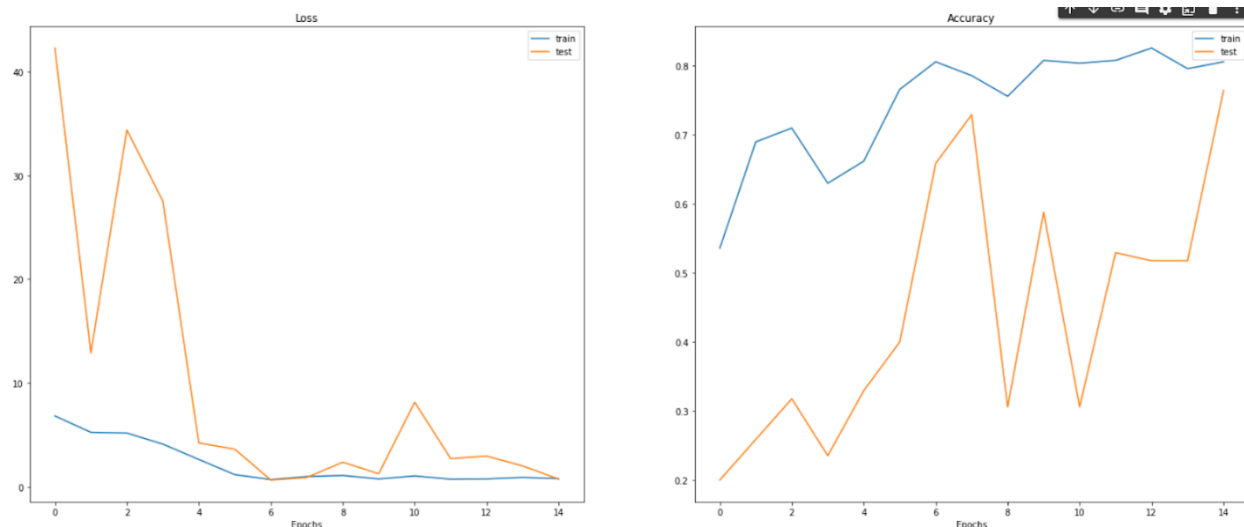
Các biểu đồ sau thể hiện Accuracy và Loss của 10 mô hình mà nhóm đã thử nghiệm



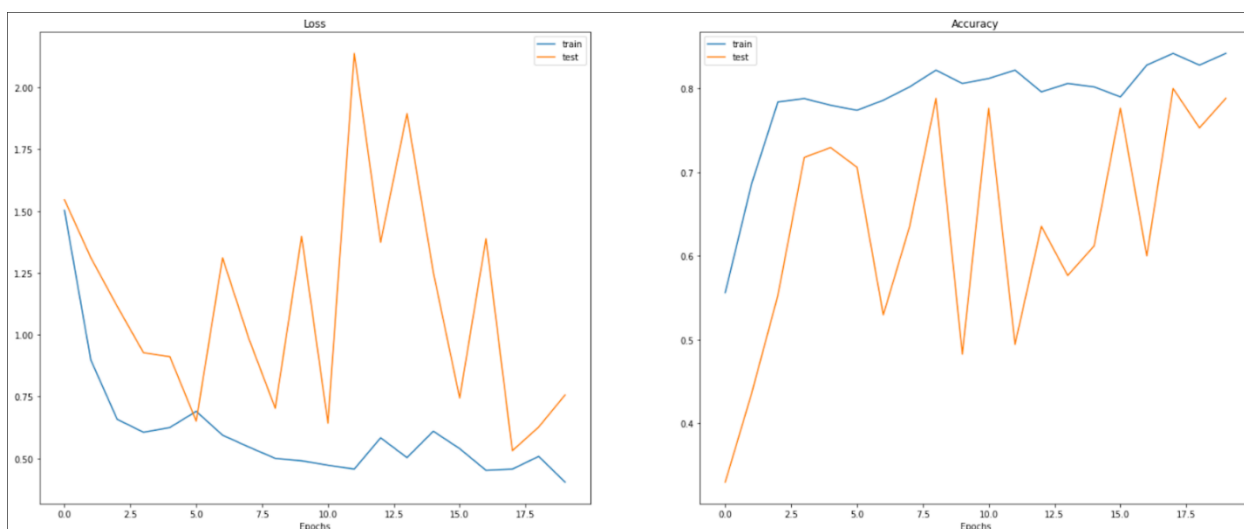
Mô hình 1: Mô hình Alexnet



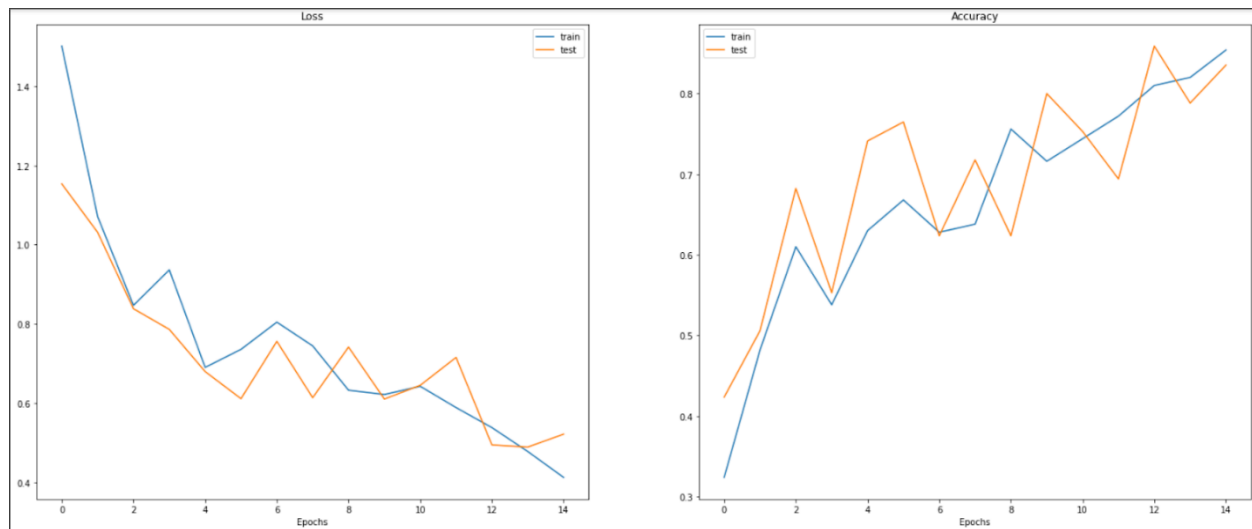
Mô hình 2: Mô hình Alexnet - Giảm số bộ lọc các lớp convolution



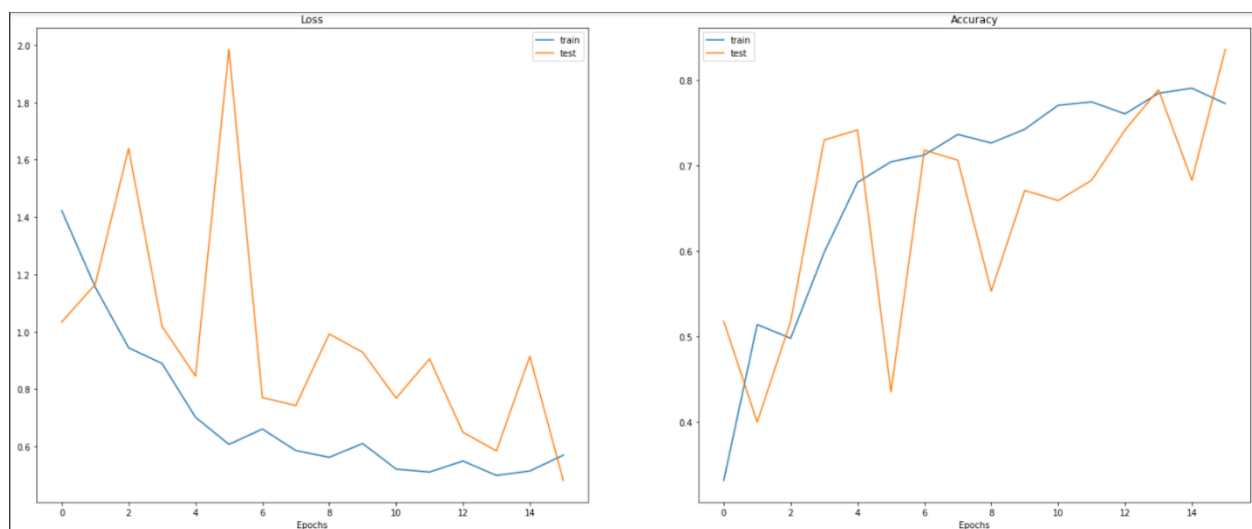
Mô hình 3: Mô hình 2 thay đổi Pool size từ 3 về 2



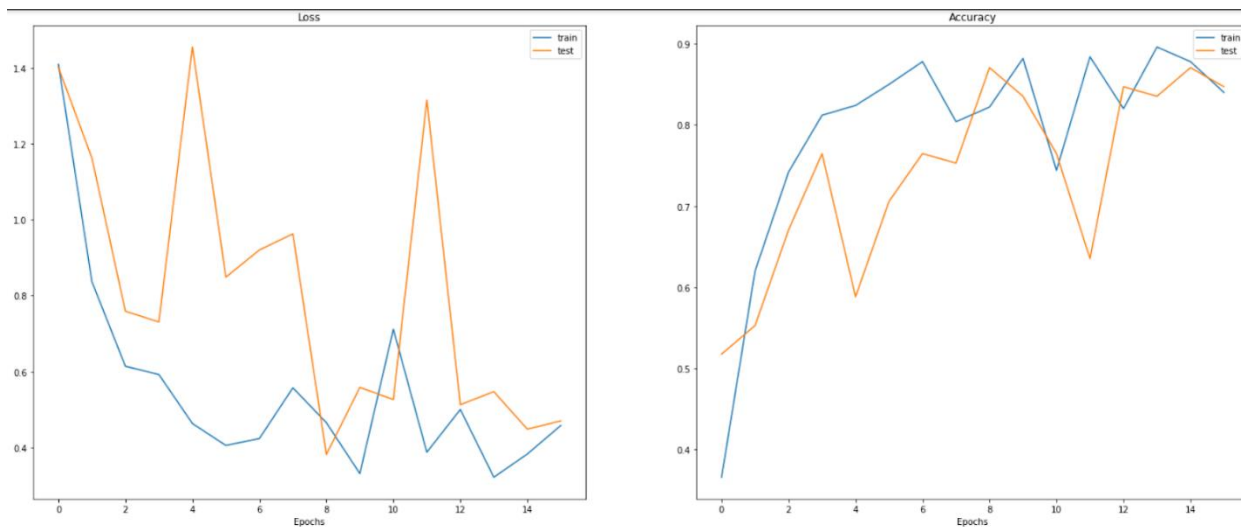
Mô hình 4: Mô hình 3 giảm kích thước ảnh đầu vào



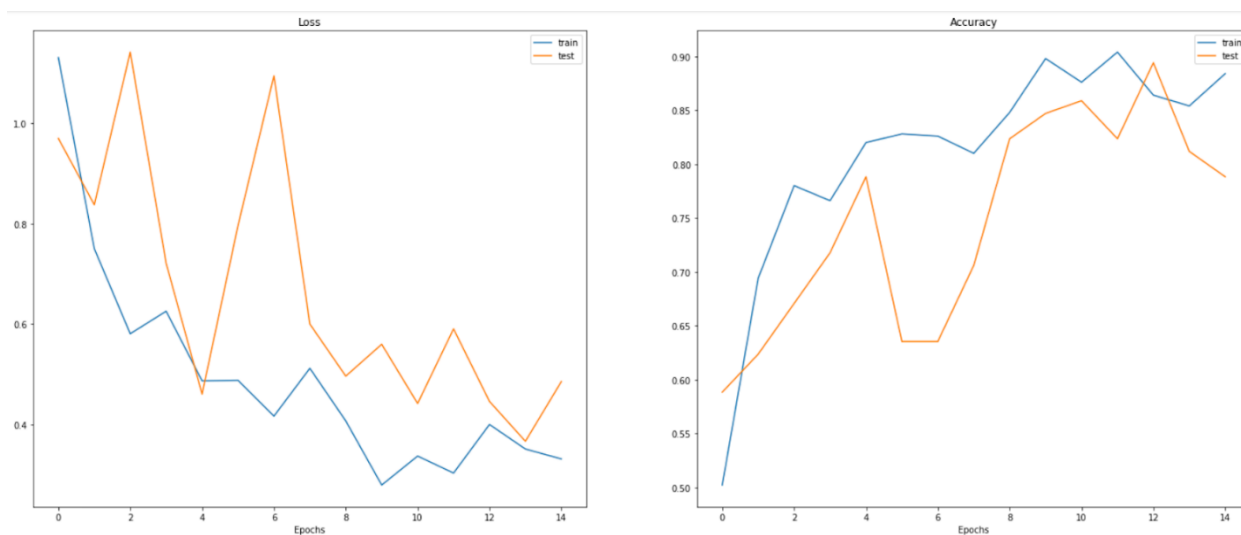
Mô hình 5: Mô hình 4 bổ Batch Normalization



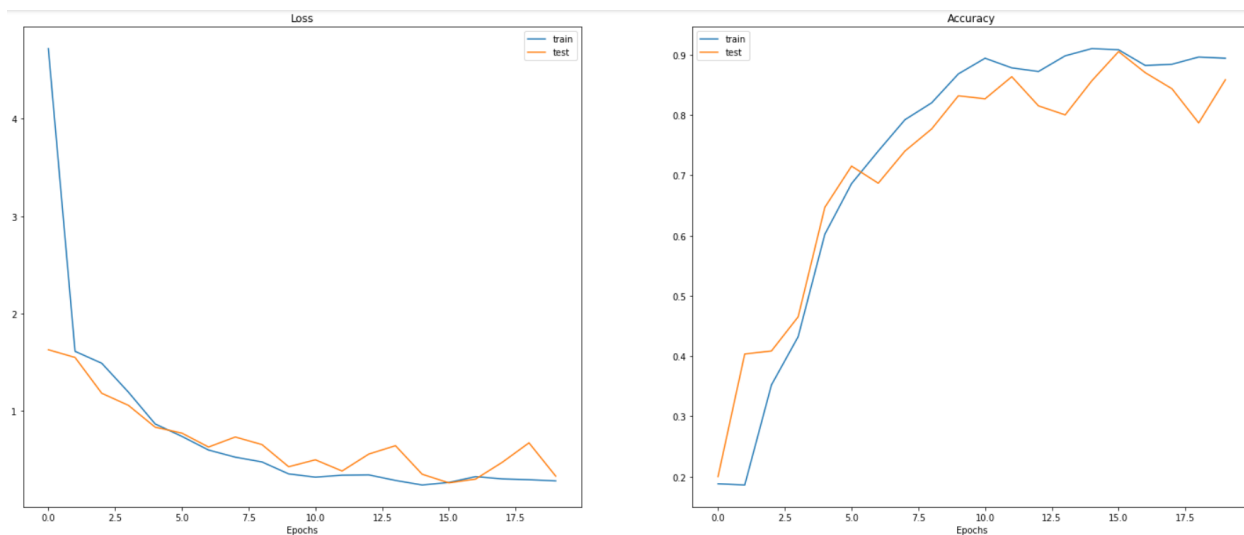
Mô hình 6: Mô hình 5 giảm lớp Convolution số 4



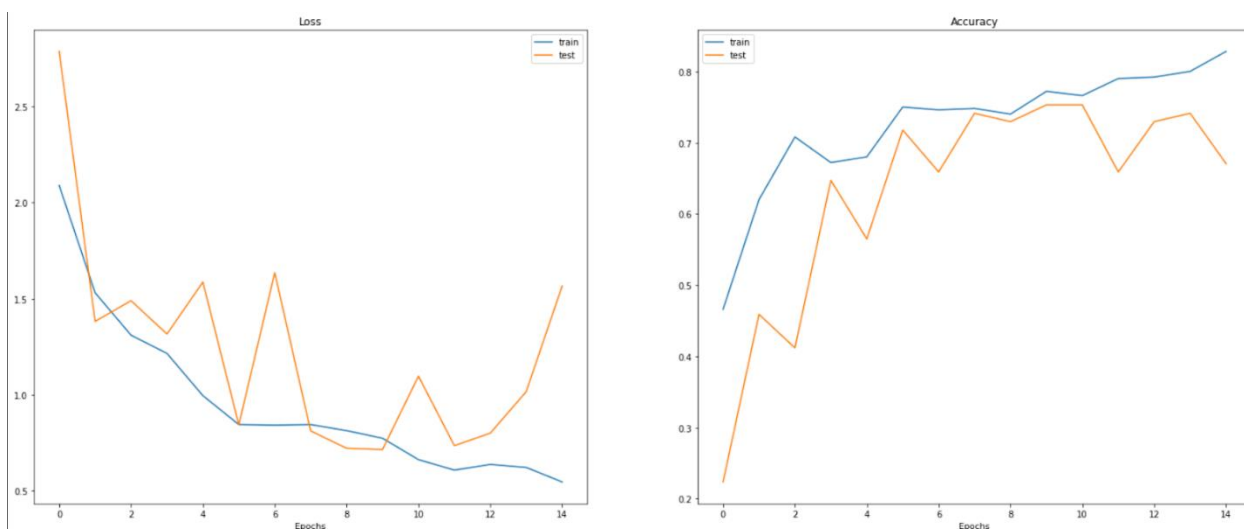
Mô hình 7: Mô hình 6 giảm số lớp Convolution số 3



Mô hình 8: Mô hình 7 giảm số lớp Fully Connected



Mô hình 9: Mô hình 7 giảm số node Fully Connected 4096 thành 1024 và 512



Mô hình 10: Mô hình 9 Thêm lớp BatchNormalization vào giữa conv và pool của mỗi lớp

Link chứa code và dữ liệu:

https://drive.google.com/drive/folders/17ab9Qyv5yXg5cFhLVF4Qknro-ITcDiep?usp=share_link

Code:

```
import os
import cv2
import time
import numpy as np
from cv2 import rotate
import tensorflow as tf
from sklearn import metrics
from sklearn.metrics import accuracy_score
from keras_preprocessing import image
from keras_preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator( rescale = 1./255,      #Tien xu ly du lieu
                                   horizontal_flip = True,
                                   vertical_flip = True,
                                   height_shift_range=0.1,
                                   width_shift_range=0.1,
                                   rotation_range = 90,
                                   brightness_range=[0.5,1.5])

training_set = train_datagen.flow_from_directory('/content/drive/MyDrive/Code
AI/Train',
                                                target_size = (64, 64),
                                                batch_size = 100,
                                                class_mode = 'categorical')

test_datagen = ImageDataGenerator(rescale = 1./255)
test_set = test_datagen.flow_from_directory('/content/drive/MyDrive/Code
AI/Test',
                                            target_size = (64, 64),
                                            batch_size = 100,
                                            class_mode = 'categorical')

classes = ['Mơ Khô', 'Phúc Bồn Tử', 'Sung Khô', 'Táo Tàu', 'Thảo Quả']
print("Image Processing.....Completed")
```

```
#Build Neural Network
model = tf.keras.models.Sequential()
print("Building Neural Network.....")

# Layer Conv 1
model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3,
activation='relu', input_shape=[64, 64, 3], name='Conv_1'))
model.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2, name='MaxPool_1'))

# Layer Conv 2
model.add(tf.keras.layers.Conv2D(filters=64, kernel_size=3,
activation='relu', name='Conv_2'))
model.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2, name='MaxPool_2'))

# Layer Conv 3
model.add(tf.keras.layers.Conv2D(filters=128, kernel_size=3,
activation='relu', name='Conv_3'))
model.add(tf.keras.layers.MaxPool2D(pool_size=2, strides=2, name='MaxPool_3'))

# Flatten
model.add(tf.keras.layers.Flatten(name='Flatten'))

# Fully Connected
model.add(tf.keras.layers.Dense(units=1024, activation='relu', name='Layer1'))
model.add(tf.keras.layers.Dropout(0.5))
model.add(tf.keras.layers.Dense(units=512, activation='relu', name='Layer2'))
model.add(tf.keras.layers.Dropout(0.5))

# Softmax
model.add(tf.keras.layers.Dense(units=5, activation='softmax', name='Output'))

#Function Optimizer Adam
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy',
metrics = ['accuracy'])
model.optimizer.learning_rate = 0.005    # Defaults to 0.001

#In cấu trúc network ra màn hình
print('Detail of network: ')
print(model.summary())
```

```
# Training Model
print("Training model")
history=model.fit(x = training_set, validation_data = test_set, epochs = 20)
model.save("model.h5")

# VẼ ĐỒ THỊ LOSS VÀ ACCURACY
import matplotlib.pyplot as pyplot

# plot loss
pyplot.figure(figsize=(25,10))
pyplot.subplot(121)
pyplot.title('Loss')
pyplot.xlabel('Epochs')
pyplot.plot(history.history['loss'], label='train')
pyplot.plot(history.history['val_loss'], label='test')
pyplot.legend()
# plot accuracy
pyplot.subplot(122)
pyplot.title('Accuracy')
pyplot.xlabel('Epochs')
pyplot.plot(history.history['accuracy'], label='train')
pyplot.plot(history.history['val_accuracy'], label='test')
pyplot.legend()
pyplot.show()

#Tinh toan CONFUSION MATIRIX, PRECISION, RECALL VÀ F1-SCORE
from sklearn.metrics import accuracy_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from sklearn.metrics import f1_score
from sklearn.metrics import roc_auc_score
from sklearn.metrics import confusion_matrix

#Phan tach du lieu test du lieu test
X_test, y_test = next(test_set)
#Tra ve ket qua du doan
y_hat = model.predict(X_test)
y_pred = np.argmax(y_hat, axis=1)
y_test_label = np.argmax(y_test, axis=1)
```

```
# Tính accuracy: (tp + tn) / (p + n)
accuracy = accuracy_score(y_test_label, y_pred)
print('Accuracy: %f' % accuracy)
# Tính precision tp / (tp + fp)
precision = precision_score(y_test_label, y_pred, average='macro')
print('Precision: %f' % precision)
# Tính recall: tp / (tp + fn)
recall = recall_score(y_test_label, y_pred, average='macro')
print('Recall: %f' % recall)
# Tính f1: 2 tp / (2 tp + fp + fn)
f1 = f1_score(y_test_label, y_pred, average='macro')
print('F1 score: %f' % f1)
# Tính Area under ROC
auc = roc_auc_score(y_test, y_hat, multi_class='ovr')
print('ROC AUC: %f' % auc)
# Tính confusion matrix
matrix = confusion_matrix(y_test_label, y_pred)
#print(matrix)

# Ma tran nham lan (Confusion matrix)
import pandas as pd
import seaborn as sn
df_cm = pd.DataFrame(matrix, index = [i for i in "12345"],
                      columns = [i for i in "12345"])
pyplot.figure(figsize = (10,7))
sn.heatmap(df_cm, annot=True)
print("\nConfusion matrix: X-Dự Đoán \t Y-Nhãn Thật\n1-Mơ Khô\t2-Phúc Bồn
Tử\t\t3-Sung Khô \t 4-Táo Tàu\t5-Thảo Quả")

# Kiem tra ket qua
from matplotlib import figure
import numpy as np
from keras_preprocessing import image
import cv2
from tensorflow.keras.models import load_model
from google.colab.patches import cv2_imshow

new_model = load_model('model.h5')
```



```
path = r'/content/drive/MyDrive/Code AI/test/Mơ Khô/
BLEUyi_simg_de2fe0_500x500_maxb.jpg'

img = cv2.imread(path)
cv2.imshow(img)
test_image = image.load_img(path, target_size=(64,64))
test_image = image.img_to_array(test_image)
test_image = np.expand_dims(test_image, axis=0)
result = new_model.predict(test_image)
result1 = result[0]
for y in range(5):
    if result1[y] == 1.:
        break
prediction = classes[y]
print(prediction)
cv2.waitKey()
cv2.destroyAllWindows()
```

TÀI LIỆU THAM KHẢO

- [1] Tìm hiểu CNN: <https://pbcquoc.github.io/cnn/>
- [2] Vũ Hữu Tiệp (2017). Machine Learning cơ bản: <https://machinelearningcoban.com/general/2017/02/06/featureengineering/>
- [3] Andrej Karpathy. CS231n Convolutional Neural Networks for Visual Recognition - Image Classification. <http://cs231n.github.io/classification/>
- [4] Tìm hiểu lớp tích chập: <https://mlnotebook.github.io/post/CNN1/>
- [5] Fu, L., Sun, S., Li, R., & Wang, S. (2016). Classification of kiwifruit grades based on fruit shape using a single camera. Sensors (Switzerland).
- [6] 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.
- [7] Arivazhagan, S., Shebiah, R. N., Nidhyandhan, S. S., & Ganesan, L. (2010). Fruit Recognition using Color and Texture Features. Information Sciences
- [8] Đánh giá các thuật toán tối ưu đối với mô hình mạng nơ-ron tích chập trong tác vụ nhận diện hình ảnh, Vương Quang Phước, Nguyễn Đức Nhật Quang, khoa Điện, Điện tử và Công nghệ vật liệu, trường Đại học Khoa học, Đại học Huế (2021).
- [9] <https://image-net.org/challenges/LSVRC/2012/results.html>
- [10] Đánh giá model AI: <https://www.miai.vn/2020/06/16/oanh-gia-model-ai-theo-cach-mi-an-lien-chuong-2-precision-recall-va-f-score/>
- [11] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision.
- [12] Samer, C. H., Rishi, K., & Rowen. (2015). Image Recognition Using Convolutional Neural Networks. Cadence Whitepaper.