

BỘ THÔNG TIN VÀ TRUYỀN THÔNG
HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG



BÁO CÁO ĐỒ ÁN
THỰC TẬP TỐT NGHIỆP

Đề tài :

**“ỨNG DỤNG HỌC MÁY PHÂN LOẠI CHẤT
LƯỢNG HẠT CÀ PHÊ ARABICA”**

Giáo viên hướng dẫn : NGUYỄN HOÀNG THÀNH
Sinh viên thực hiện : Hà Nam Thái
Mã số sinh viên : N19DCCN187
Lớp : D19CQCNPM02-N
Khóa : 2019
Hệ : ĐẠI HỌC CHÍNH QUY

TPHCM, tháng 07 năm 2023

LỜI CẢM ƠN

Kính thưa quý thầy cô!

Qua 4 năm học tập và rèn luyện tại trường Học viện Công nghệ Bru chính – Viễn thông, được sự chỉ bảo và giảng dạy nhiệt tình của các quý thầy cô Khoa Công nghệ thông tin 2 đã truyền đạt cho em những kiến thức về lý thuyết và thực hành trong suốt thời gian học ở trường. Và trong thời gian thực tập em đã có cơ hội áp dụng những kiến thức học ở trường vào thực tế ở công ty, đồng thời học hỏi được nhiều kinh nghiệm thực tế tại công ty. Cùng với sự nỗ lực của bản thân và sự trợ giúp từ phía công ty, em đã hoàn thành đồ án thực tập tốt nghiệp có tên là **“ỨNG DỤNG HỌC MÁY PHÂN LOẠI CHẤT LƯỢNG HẠT CÀ PHÊ ARABICA”**.

Từ những kết quả mà em đã đạt được, xin chân thành cảm ơn:

Quý thầy cô Học viện Công nghệ Bru chính – Viễn thông, đã dạy dỗ và truyền đạt kiến thức cho em trong thời gian qua. Đặc biệt, là thầy Nguyễn Hoàng Thành đã tận tình hướng dẫn em hoàn thành tốt đồ án thực tập tốt nghiệp này.

Do kiến thức còn hạn hẹp nên không tránh khỏi những thiếu sót trong cách hiểu vấn đề và lỗi trình bày. Em rất mong nhận được những đóng góp ý kiến của quý thầy cô và Ban lãnh đạo, các anh chị trong công ty để báo cáo tốt nghiệp đạt được kết quả tốt hơn.

Xin cảm ơn!

TP. Hồ Chí Minh, ngày 14 tháng 8 năm 2023

Sinh viên thực hiện

Hà Nam Thái

MỤC LỤC

MỤC LỤC	0
DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT.....	2
DANH MỤC CÁC BẢNG.....	0
DANH MỤC CÁC HÌNH VẼ	0
LỜI MỞ ĐẦU	0
CHƯƠNG 1 : TỔNG QUAN	1
1.1 LÝ DO CHỌN ĐỀ TÀI	1
1.2 MỤC TIÊU XÂY DỰNG ĐỀ TÀI.....	2
CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT	3
2.1 TỔNG QUAN VỀ HỌC SÂU	3
2.1.1 KHÁI NIỆM.....	3
2.1.2 LỊCH SỬ RA ĐỜI.....	3
2.1.3 MỤC ĐÍCH	3
2.1.4 ƯU ĐIỂM	3
2.1.5 KHUYẾT ĐIỂM.....	4
2.1.6 ỨNG DỤNG CỦA HỌC SÂU	4
2.1.7 MẠNG NƠ-RON TÍCH CHẬP (CNN)	5
2.1.8 EPOCH.....	13
2.2 SƠ LƯỢC VỀ BỘ DATASET USK-COFFEE	13
2.2.1 CẤU TRÚC TẬP DỮ LIỆU:.....	13
2.2.2 ĐẶC ĐIỂM TỪNG LỚP TRONG TẬP DỮ LIỆU:.....	13
CHƯƠNG 3 : CÀI ĐẶT, XÂY DỰNG VÀ THỰC NGHIỆM.....	14
3.1 YÊU CẦU	14
3.2 XÂY DỰNG MÔ HÌNH CNN.....	14
3.3 IMPORT	17
3.4 PHÂN TÍCH MODEL SAU KHI HUẤN LUYỆN:.....	17
3.5 DATA VISUALIZATION.....	19
3.5.1 ĐỘ CHÍNH XÁC VÀ HÀM MẤT MẤT.....	19
3.5.2 CONFUSION MATRIX	19
3.6 ĐÁNH GIÁ THỰC NGHIỆM	22
3.7 XÂY DỰNG GIAO DIỆN	22
CHƯƠNG 5: KẾT LUẬN.....	24
5.1 TÓM TẮT KẾT QUẢ ĐẠT ĐƯỢC.....	24
5.2 HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI.....	24

TÀI LIỆU THAM KHẢO.....	24
DANH MỤC CÁC WEBSITE THAM KHẢO	24

DANH MỤC KÝ HIỆU VÀ CHỮ VIẾT TẮT

Từ viết tắt/kí hiệu	Tiếng anh	Giải nghĩa
CNN	Convolutional Neural Network	Mạng nơ-ron tích chập
DL	Deep Learning	Học sâu

DANH MỤC CÁC BẢNG

Bảng 1.1 Bảng chi tiết của một tập train, test, validation.....	13
Bảng 1.2 Bảng chi tiết của tập dữ liệu USK-COFFEE	13

DANH MỤC CÁC HÌNH VẼ

Hình 2.1 Minh học tích chập	6
Hình 2.2 Ảnh mờ sau khi chập	6
Hình 2.2.3 Ảnh được phát hiện biên sau khi chập.....	7
Hình 2.4 Mô hình mạng neuron tích chập	8
Bảng 2.2.5 Mô hình mạng perceptron đa tầng	9
Hình 2.6 Kết nối các điểm ảnh	10
Hình 2.7 Trường tiếp nhận cục bộ.....	11
Hình 2.8 Trượt trường tiếp nhận cục bộ	11
Hình 2.9 Đơn giản hóa thông tin đầu ra từ lớp tích chập	12
Hình 3.1 Các lớp của mô hình.....	14
Hình 3.2 Kiến trúc mô hình	16
Hình 3.3 Bắt đầu huấn luyện tập dữ liệu	17
Hình 3.4 Sau khi huấn luyện tập dữ liệu	18
Hình 3.5 Kết quả sau khi chạy qua tập thử nghiệm.....	18
Hình 3.6 Hình ảnh gây nhiễu cho tập huấn luyện	18
Hình 3.7 Đồ thị về độ chính xác và hàm mất mát trên tập huấn luyện và tập kiểm tra.....	19
Hình 3.8 Kết quả sau khi chạy qua một số bức ảnh của tập kiểm thử.....	22
Hình 3.9 Giao diện chính.....	23
Hình 3.10 Giao diện khi tải lên một bức ảnh.....	23
Bảng 3.11 Giao diện khi sử dụng camera máy tính.....	24

LỜI MỞ ĐẦU

Trong ngành công nghiệp cà phê, việc phân loại chất lượng hạt cà phê là một quá trình quan trọng để đảm bảo chất lượng và giá trị của sản phẩm cuối cùng. Hạt cà phê Arabica được coi là một trong những loại cà phê cao cấp nhất trên thế giới với hương vị tuyệt vời và mùi thơm đặc trưng. Tuy nhiên, quá trình phân loại chất lượng cà phê Arabica vẫn còn tương đối phức tạp và mất thời gian do sự đánh giá thủ công từ phía con người.

Trong lĩnh vực Học máy, các phương pháp và thuật toán đã được phát triển để tự động hóa quá trình phân loại, từ đó giúp cải thiện hiệu suất và độ chính xác trong công việc này. Ứng dụng Học máy trong việc phân loại chất lượng hạt cà phê Arabica là một lĩnh vực nổi bật và tiềm năng trong lĩnh vực nghiên cứu và ứng dụng công nghệ.

Hy vọng rằng thông qua việc nghiên cứu và áp dụng Học máy trong lĩnh vực này, chúng ta có thể tận dụng được tiềm năng của công nghệ để nâng cao chất lượng và hiệu suất trong việc phân loại chất lượng hạt cà phê Arabica.

CHƯƠNG 1 : TỔNG QUAN

1.1 LÝ DO CHỌN ĐỀ TÀI

Trong thế kỷ 21, công nghệ và khoa học dữ liệu đang đóng vai trò ngày càng quan trọng trong hầu hết các lĩnh vực của đời sống hiện đại. Một trong những lĩnh vực mà công nghệ này đã thực sự tạo ra sự thay đổi đáng kể là ngành công nghiệp cà phê. Với hàng tỷ tách cà phê được tiêu thụ mỗi ngày trên toàn cầu, nhu cầu tìm kiếm và phân loại chất lượng hạt cà phê đã trở thành một vấn đề cực kỳ quan trọng và cần thiết.

Trong bối cảnh đó, ứng dụng học máy trong việc phân loại chất lượng hạt cà phê Arabica đã nhanh chóng trở thành một đề tài nổi bật trong nghiên cứu khoa học và công nghệ thực tế. Chất lượng của hạt cà phê Arabica được xem là một yếu tố quyết định trong việc xác định giá trị thương mại và thưởng thức của cà phê. Tuy nhiên, quá trình phân loại chất lượng cà phê Arabica truyền thống vẫn phụ thuộc vào sự kinh nghiệm và khả năng nhận biết của con người, đồng thời đòi hỏi sự đánh giá một cách chủ quan và tiêu tốn nhiều thời gian.

Để giải quyết những hạn chế này, việc áp dụng các thuật toán học máy và phương pháp phân loại tự động đã đem lại những kết quả ấn tượng. Sự phát triển của công nghệ hiện đại đã cung cấp cho chúng ta khả năng khai thác các đặc điểm quan trọng từ dữ liệu và xây dựng các mô hình học máy để phân loại chất lượng cà phê Arabica một cách tự động và chính xác hơn.

Trong bối cảnh đó, đề tài Ứng dụng học máy phân loại chất lượng hạt cà phê Arabica đã thu hút sự quan tâm của nhiều nhà nghiên cứu và nhà sản xuất cà phê trên toàn cầu. Bằng cách kết hợp sức mạnh của học máy và dữ liệu cà phê, chúng ta có thể tạo ra một hệ thống tự động phân loại chất lượng hạt cà phê, giúp nâng cao hiệu quả, đồng thời giảm thiểu sự phụ thuộc vào con người trong quá trình phân loại.

Trong đề án này, chúng ta sẽ tìm hiểu về ứng dụng học máy trong phân loại chất lượng hạt cà phê Arabica, từ việc thu thập và tiền xử lý dữ liệu, xây dựng mô hình học máy cho đến đánh giá kết quả. Đồng thời, chúng ta sẽ xem xét những lợi ích và thách thức của việc áp dụng học máy trong ngành công nghiệp cà phê, mở ra cơ hội cho sự tiến bộ và đổi mới trong việc phân loại chất lượng cà phê Arabica.

1.2 MỤC TIÊU XÂY DỰNG ĐỀ TÀI

- Tổng quan về học máy, học sâu
- Nghiên cứu về bộ dữ liệu USK-COFFEE
- Xây dựng mô hình học máy/học sâu để phân loại chất lượng hạt cà phê dựa vào bộ dataset USK-COFFEE
- Xây dựng WebApp kết hợp với webcam để phân loại

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

2.1 TỔNG QUAN VỀ HỌC SÂU

2.1.1 KHÁI NIỆM

Học sâu là một phương pháp tiên tiến trong lĩnh vực trí tuệ nhân tạo và học máy, tập trung vào xây dựng và huấn luyện các mô hình máy học sâu có khả năng học tự động và trích xuất các đặc trưng phức tạp từ dữ liệu đầu vào. Điều độc đáo của học sâu chính là khả năng tự tạo ra các đặc trưng cần thiết để giải quyết các nhiệm vụ phức tạp, mà trước đây yêu cầu sự can thiệp và thiết kế thủ công từ con người.

Cơ bản thì học sâu dựa trên cấu trúc mạng neuron với nhiều lớp ẩn, mỗi lớp đóng vai trò trích xuất thông tin từ lớp trước và tái tổ hợp chúng để tạo ra những đặc trưng phức tạp hơn. Những mô hình này hoạt động tương tự như não người, mô phỏng quá trình học và tìm hiểu cách các đặc trưng tương tác để thực hiện các tác vụ như phân loại, dự đoán hoặc phân tích.

Học sâu đã thay đổi cách chúng ta tiếp cận và giải quyết các bài toán phức tạp trong nhiều lĩnh vực. Từ thị giác máy tính cho đến xử lý ngôn ngữ tự nhiên và y học, học sâu đã mang lại khả năng nhận diện, hiểu và tạo ra thông tin từ dữ liệu không xử lý trước. Điều này giúp tạo ra những ứng dụng mạnh mẽ, từ xe tự hành đến hệ thống tự động phát hiện dịch vụ gian lận tài chính. Từ việc giúp máy tính "nhìn thấy" hình ảnh như con người đến khả năng "hiểu" ngôn ngữ tự nhiên như một người đang đọc và hiểu nghĩa của văn bản. Học sâu đã mở ra cánh cửa cho một tương lai đầy tiềm năng, nơi máy tính không chỉ là công cụ thực hiện theo hướng dẫn, mà còn là một "người bạn" có khả năng học hỏi và định hình mới.

2.1.2 LỊCH SỬ RA ĐỜI

Thật ra các khái niệm liên quan đến mạng nơ-ron nhân tạo và Deep Learning đã xuất hiện từ khoảng những năm 1960, tuy nhiên nó lại bị giới hạn bởi khả năng tính toán và số lượng dữ liệu lúc bấy giờ. Trong những năm gần đây, những tiến bộ trong phân tích dữ liệu lớn (Big Data) đã cho phép ta tận dụng được tối đa khả năng của mạng nơ-ron nhân tạo.

Mạng nơ-ron nhân tạo chính là động lực chính để phát triển Deep Learning. Các mạng nơ-ron sâu (DNN) bao gồm nhiều lớp nơ-ron khác nhau, có khả năng thực hiện các tính toán có độ phức tạp rất cao. Deep Learning hiện đang phát triển rất nhanh và được xem là một trong những bước đột phá lớn nhất trong Machine Learning.

2.1.3 MỤC ĐÍCH

Mục đích chính của deep learning là khám phá và phân tích cấu trúc, tính chất và ý nghĩa của dữ liệu phức tạp, đặc biệt là dữ liệu phi cấu trúc và phi tuyến tính. Deep learning hướng đến việc xây dựng mô hình máy tính có khả năng học và tự điều chỉnh thông qua việc tự động học các đặc trưng và biểu diễn cấu trúc sâu của dữ liệu.

2.1.4 ƯU ĐIỂM

Khả năng học tự động các đặc trưng:

Deep learning có khả năng tự động học và trích xuất các đặc trưng sâu từ dữ liệu, mà không cần phải xác định trước các đặc trưng cụ thể. Điều này cho phép mô hình học các mức biểu diễn cấu trúc sâu và tìm ra các mối quan hệ phức tạp giữa các

đặc trưng, giúp cải thiện hiệu suất và khả năng tổng quát hóa của mô hình.

Xử lý dữ liệu phức tạp:

Deep learning có khả năng xử lý dữ liệu phi cấu trúc và phi tuyến tính như hình ảnh, âm thanh, văn bản và video. Các mô hình deep learning có thể học và hiểu thông tin từ các dạng dữ liệu phức tạp và trừu tượng, đặc biệt là khi dữ liệu lớn và đa dạng.

Độ chính xác cao:

Deep learning có khả năng đạt được độ chính xác cao trong các nhiệm vụ phân loại, nhận dạng và dự đoán. Khả năng học sâu và khám phá các mẫu phức tạp giúp mô hình deep learning nhận diện và hiểu được những điểm đặc trưng quan trọng trong dữ liệu.

2.1.5 KHUYẾT ĐIỂM

Đòi hỏi lượng dữ liệu lớn:

Deep learning yêu cầu lượng dữ liệu huấn luyện lớn để đạt được hiệu suất tốt. Việc thiếu dữ liệu có thể gây ra hiện tượng quá khớp (overfitting), khi mô hình chỉ học được từ dữ liệu huấn luyện cụ thể mà không tổng quát hóa tốt cho các dữ liệu mới.

Độ phức tạp cao:

Deep learning có cấu trúc mạng nơ-ron sâu và lượng tham số lớn, dẫn đến độ phức tạp tính toán và đòi hỏi tài nguyên tính toán cao. Việc huấn luyện mô hình deep learning có thể mất thời gian và cần sự tinh chỉnh kỹ lưỡng của các siêu tham số.

Giải thích kết quả khó khăn:

Deep learning thường được xem như "hộp đen" do tính phi tuyến tính và độ sâu của mô hình. Việc giải thích rõ ràng về quá trình ra quyết định của mô hình deep learning có thể khó khăn, làm giảm sự tin cậy và sự hiểu biết về quyết định của mô hình.

2.1.6 ỨNG DỤNG CỦA HỌC SÂU

Mô hình học sâu có một số trường hợp sử dụng trong lĩnh vực ô tô, hàng không vũ trụ, sản xuất, điện tử, nghiên cứu y học và nhiều lĩnh vực khác

- Thị giác máy tính:

Thị giác máy tính là khả năng của máy tính thực hiện trích xuất dữ liệu cũng như thông tin chuyên sâu từ hình ảnh và video. Máy tính có thể sử dụng các kỹ thuật học sâu để hiểu hình ảnh theo cách giống như con người. Thị giác máy tính được ứng dụng trong nhiều trường hợp, chẳng hạn như:

- Kiểm duyệt nội dung để tự động loại bỏ nội dung không an toàn hoặc không phù hợp khỏi kho lưu trữ hình ảnh và video
- Nhận diện khuôn mặt để xác định khuôn mặt cũng như các đặc điểm như mở mắt, đeo kính và để râu
- Phân loại hình ảnh để xác định logo thương hiệu, quần áo, đồ bảo hộ và các chi tiết hình ảnh khác

- Nhận dạng giọng nói

Các mô hình học sâu có thể phân tích giọng nói con người, bắt kể mẫu giọng, cao độ, tông, ngôn ngữ và giọng vùng miền khác nhau. Trợ lý ảo như Amazon Alexa

và phần mềm phiên âm tự động sử dụng nhận dạng giọng nói để thực hiện các tác vụ sau:

- Hỗ trợ các nhân viên trực tổng đài và tự động phân loại cuộc gọi.
- Chuyển đổi các cuộc trò chuyện về y khoa thành văn bản trong thời gian thực.
- Tạo phụ đề chính xác cho video và bản ghi âm cuộc họp để mở rộng phạm vi tiếp cận nội dung.
- Kỹ thuật xử lý ngôn ngữ tự nhiên

Máy tính sử dụng các thuật toán học sâu để thu thập thông tin chuyên sâu và ý nghĩa từ dữ liệu văn bản và tài liệu. Khả năng xử lý văn bản tự nhiên, do con người tạo ra này có một số trường hợp sử dụng, kể cả trong các chức năng sau:

- Tổng đài viên ảo và chatbot tự động
- Tự động tóm tắt tài liệu hoặc bài viết tin tức
- Phân tích nghiệp vụ thông minh các tài liệu dài, chẳng hạn như các email và biểu mẫu
- Lập chỉ mục các cụm từ quan trọng thể hiện cảm xúc, chẳng hạn như những bình luận tích cực và tiêu cực trên mạng xã hội
- Công cụ đề xuất

Ứng dụng có thể sử dụng các phương pháp học sâu để theo dõi hoạt động của người dùng và phát triển các đề xuất cá nhân hóa. Các phương pháp này có thể phân tích hành vi của nhiều người dùng khác nhau và giúp họ khám phá các sản phẩm hoặc dịch vụ mới. Ví dụ: nhiều công ty truyền thông và giải trí, chẳng hạn như Netflix, Fox và Peacock, sử dụng mô hình học sâu để đưa ra các video đề xuất cá nhân hóa.

2.1.7 MẠNG NƠ-RON TÍCH CHẬP (CNN)

2.1.7.1 ĐỊNH NGHĨA MẠNG NƠ-RON TÍCH CHẬP

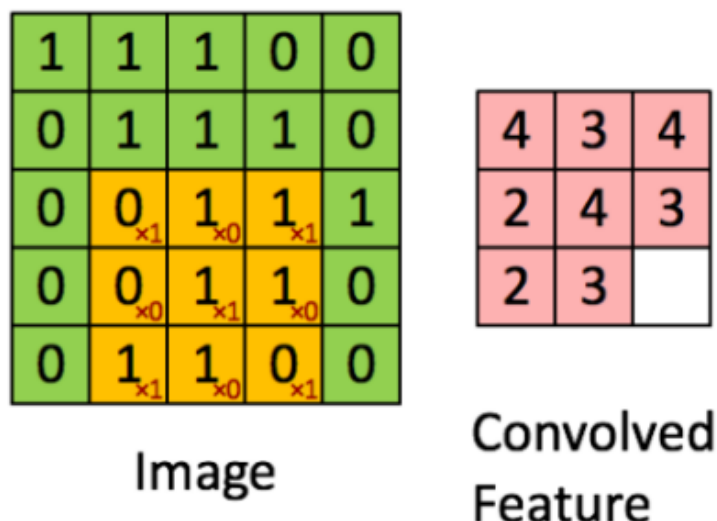
Những năm gần đây, ta đã chứng kiến được nhiều thành tựu vượt bậc trong ngành Thị giác máy tính (Computer Vision). Các hệ thống xử lý ảnh lớn như Facebook, Google hay Amazon đã đưa vào sản phẩm của mình những chức năng thông minh như nhận diện khuôn mặt người dùng, phát triển xe hơi tự lái hay drone giao hàng tự động

Convolutional Neural Network (CNNs – Mạng nơ-ron tích chập) là một trong những mô hình Deep Learning tiên tiến giúp cho chúng ta xây dựng được những hệ thống thông minh với độ chính xác cao như hiện nay. Trong đề án thực tập lần này, chúng ta sẽ trình bày về Convolution (tích chập) cũng như ý tưởng của mô hình CNNs trong bài toán phân loại hạt cà phê (Image Classification).

2.1.7.2 CONVOLUTION (TÍCH CHẬP)

Tích chập được sử dụng đầu tiên trong xử lý tín hiệu số (Signal processing). Nhờ vào nguyên lý biến đổi thông tin, các nhà khoa học đã áp dụng kỹ thuật này vào xử lý ảnh và video số.

Để dễ hình dung, ta có thể xem tích chập như một cửa sổ trượt (sliding window) áp đặt lên một ma trận. Bạn có thể theo dõi cơ chế của tích chập qua hình minh họa bên dưới.

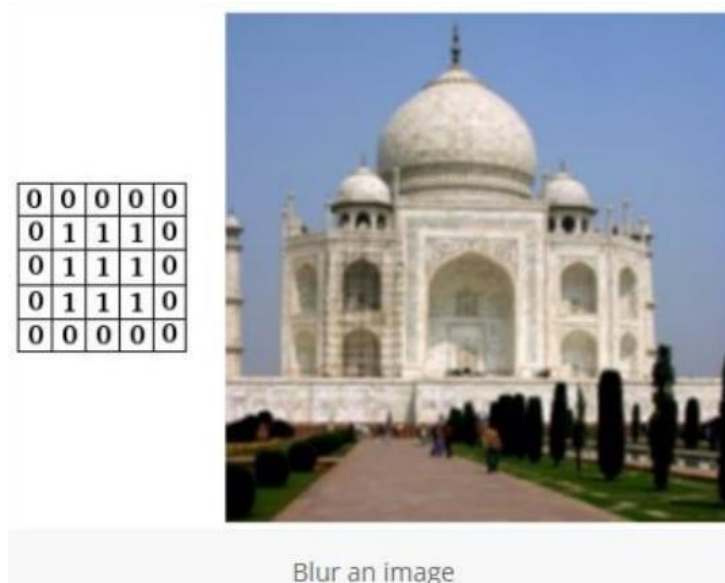


Hình 2.1 Minh họa tích chập

Ma trận bên trái là một bức ảnh đen trắng. Mỗi giá trị của ma trận tương đương với một điểm ảnh (pixel), 0 là màu đen, 1 là màu trắng (nếu là ảnh grayscale thì giá trị biến thiên từ 0 đến 255).

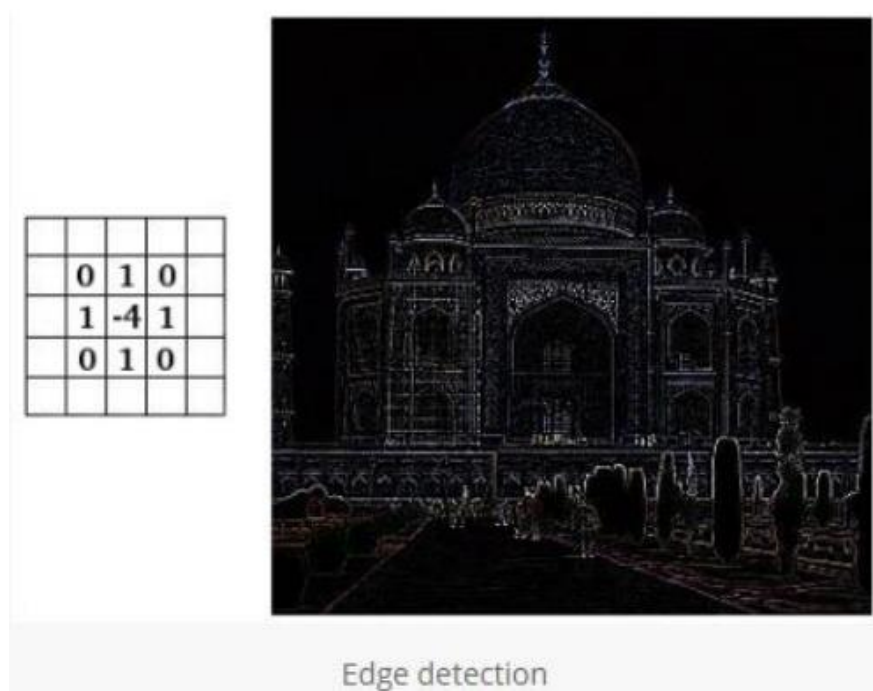
Sliding window còn có tên gọi là kernel, filter hay feature detector. Ở đây, ta dùng một ma trận filter 3×3 nhân từng thành phần tương ứng (element-wise) với ma trận ảnh bên trái. Giá trị đầu ra do tích của các thành phần này cộng lại. Kết quả của tích chập là một ma trận (convolved feature) sinh ra từ việc trượt ma trận filter và thực hiện tích chập cùng lúc lên toàn bộ ma trận ảnh bên trái. Dưới đây là một vài ví dụ của phép toán tích chập.

Ta có thể làm mờ bức ảnh ban đầu bằng cách lấy giá trị trung bình của các điểm ảnh xung quanh cho vị trí điểm ảnh trung tâm.



Hình 2.2 Ảnh mờ sau khi chập

Ngoài ra, ta có thể phát hiện biên cạnh bằng cách tính vi phân (độ dị biệt) giữa các điểm ảnh lân cận.



Hình 2 2.3 Ảnh được phát hiện biên sau khi chụp

2.1.7.3 MÔ HÌNH MẠNG NƠ-RON TÍCH CHẬP

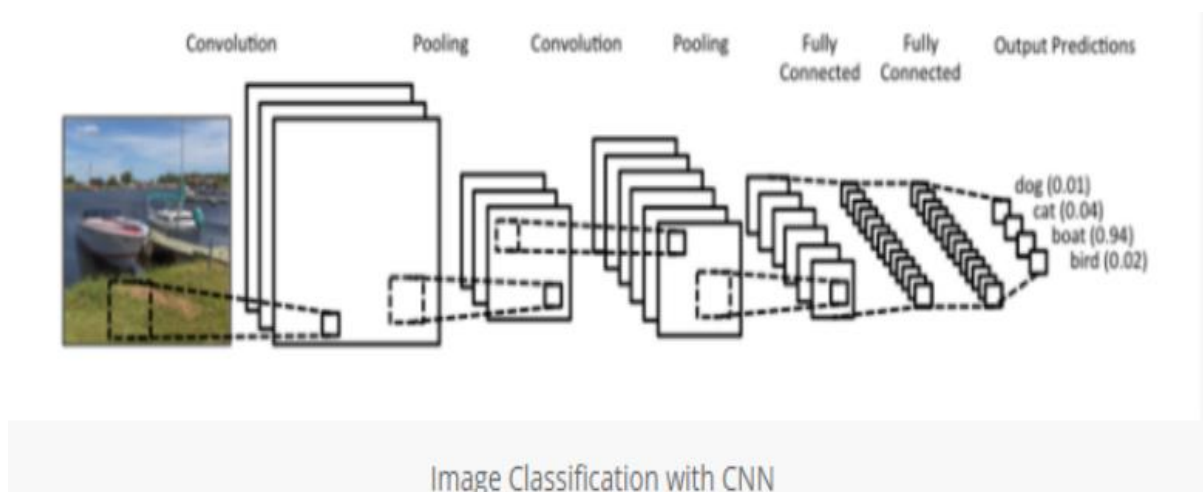
Bây giờ, Chúng ta đã biết thế nào là convolution. Vậy CNNs là gì? CNNs chỉ đơn giản gồm một vài layer của convolution kết hợp với các hàm kích hoạt phi tuyến (nonlinear activation function) như ReLU hay tanh để tạo ra thông tin trừu tượng hơn (abstract/higher-level) cho các layer tiếp theo.

Trong mô hình Feedforward Neural Network (mạng nơ-ron truyền thẳng), các layer kết nối trực tiếp với nhau thông qua một trọng số w (weighted vector). Các layer này còn được gọi là có kết nối đầy đủ (fully connected layer) hay affine layer.

Trong mô hình CNNs thì ngược lại. Các layer liên kết được với nhau thông qua cơ chế convolution. Layer tiếp theo là kết quả convolution từ layer trước đó, nhờ vậy mà ta có được các kết nối cục bộ. Nghĩa là mỗi nơ-ron ở layer tiếp theo sinh ra từ filter áp đặt lên một vùng ảnh cục bộ của nơ-ron layer trước đó.

Mỗi layer nhờ vậy được áp đặt các filter khác nhau, thông thường có vài trăm đến vài nghìn filter như vậy. Một số layer khác như pooling/subsampling layer dùng để chắt lọc lại các thông tin hữu ích hơn (loại bỏ các thông tin nhiễu). Tuy nhiên, ta sẽ không đi sâu vào khái niệm của các layer này.

Trong suốt quá trình huấn luyện, CNNs sẽ tự động học được các thông số cho các filter. Ví dụ trong tác vụ phân lớp ảnh, CNNs sẽ cố gắng tìm ra thông số tối ưu cho các filter tương ứng theo thứ tự raw **pixel** > **edges** > **shapes** > **facial** > **high-level features**. Layer cuối cùng được dùng để phân lớp ảnh.



Hình 2.4 Mô hình mạng neuron tích chập

CNNs có tính bất biến và tính kết hợp cục bộ (Location Invariance and Compositionality). Với cùng một đối tượng, nếu đối tượng này được chiếu theo các góc độ khác nhau (translation, rotation, scaling) thì độ chính xác của thuật toán sẽ bị ảnh hưởng đáng kể. Pooling layer sẽ cho bạn tính bất biến đối với phép dịch chuyển (translation), phép quay (rotation) và phép co giãn (scaling).

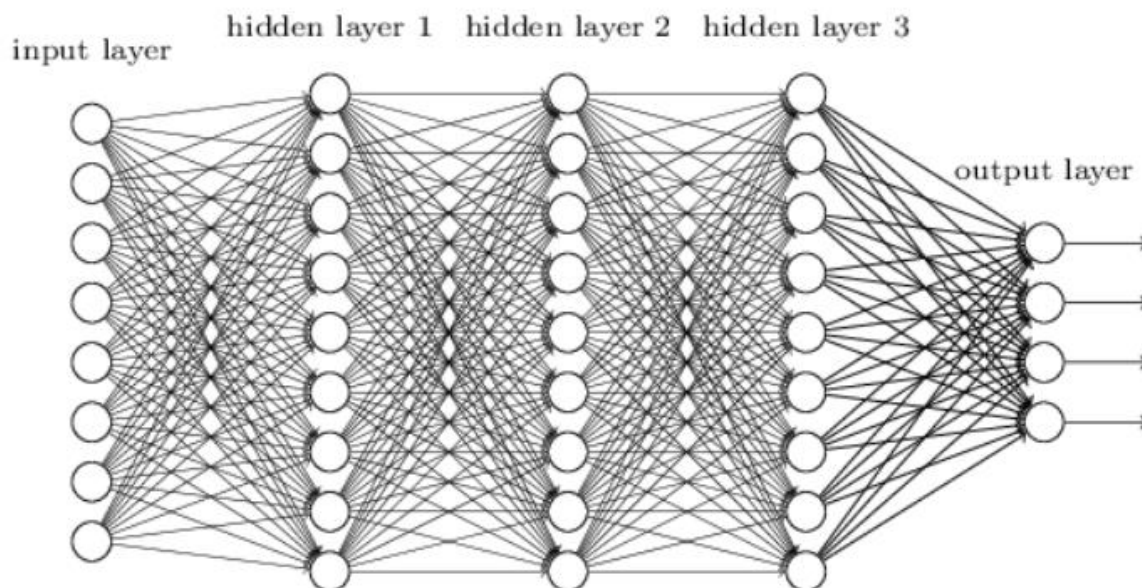
Tính kết hợp cục bộ cho ta các cấp độ biểu diễn thông tin từ mức độ thấp đến mức độ cao và trừu tượng hơn thông qua convolution từ các filter. Đó là lý do tại sao CNNs cho ra mô hình với độ chính xác rất cao. Cũng giống như cách con người nhận biết các vật thể trong tự nhiên. Ta phân biệt được một con chó với một con mèo nhờ vào các đặc trưng từ mức độ thấp (có 4 chân, có đuôi) đến mức độ cao (dáng đi, hình thể, màu lông).

2.1.7.4 XÂY DỰNG MẠNG NƠ-RON TÍCH CHẬP

Phần này sẽ giới thiệu một trong những mạng sâu được sử dụng rộng rãi đó là mạng tích chập sâu (deep convolutional networks). Chúng ta sẽ khám phá ra rất nhiều kỹ thuật hiệu quả: Tích chập (convolution), giảm số chiều (pooling), sử dụng GPUs để huấn luyện được nhiều dữ liệu hơn chúng ta đã thực hiện trên mạng cũ, mở rộng giải thuật huấn luyện dữ liệu (để giảm quá khớp – overfitting), sử dụng kỹ thuật dropout để giảm overfitting, việc sử dụng tổng hợp các mạng và các kỹ thuật khác. Kết quả là hệ thống làm việc gần như con người.

Phần còn lại của chương sẽ thảo luận về học sâu dưới góc độ tổng quan và chi tiết. Chúng ta sẽ tìm hiểu làm thế nào để các mô hình mạng nơron tích chập có thể ứng dụng để giải quyết các bài toán nhận dạng tiếng nói, xử lý ngôn ngữ tự nhiên và các lĩnh vực khác. Và chúng ta sẽ nghiên cứu về mạng nơron trong tương lai và học sâu (deep learning), từ các ý tưởng như giao diện người sử dụng hướng đích đến vai trò của học sâu trong trí tuệ nhân tạo. Phần này xây dựng dựa trên các phần trước sử dụng các ý tưởng như: lan truyền ngược (backpropagation), regularization, hàm softmax...

Chúng ta đã sử dụng mạng nơron mà trong đó các tầng liên kết đầy đủ với nhau. Tức là mỗi nơron trong mạng liên kết với tất cả các nơron trong tầng liền kề.



Bảng 2 2.5 Mô hình mạng perceptron đa tầng

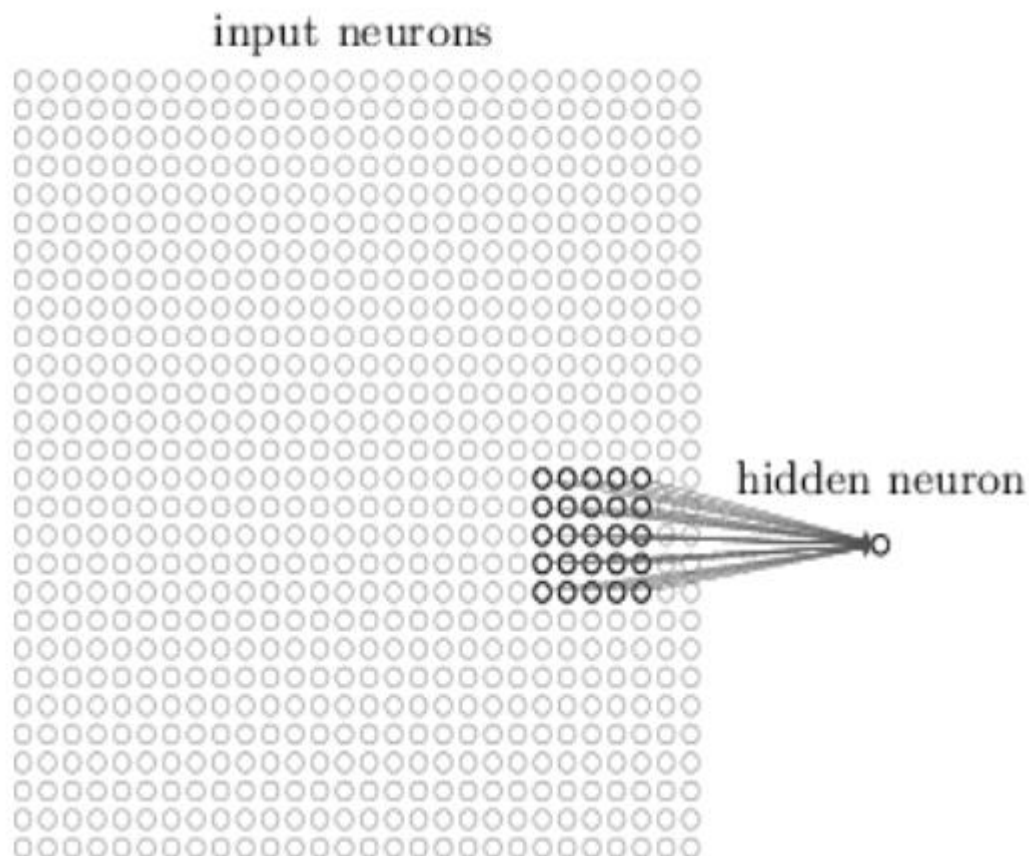
Những mạng này sử dụng một kiến trúc đặc biệt phù hợp cho bài toán phân loại ảnh. Sử dụng kiến trúc này làm cho mạng tích chập huấn luyện nhanh hơn. Kết quả là giúp chúng ta huấn luyện sâu, mạng nhiều tầng, rất phù hợp cho phân loại ảnh. Ngày nay, mạng tích chập sâu hoặc một số biến thể của nó được sử dụng trong các mạng nơron để nhận dạng ảnh.

Mạng tích chập sử dụng 3 ý tưởng cơ bản: các trường tiếp nhận cục bộ (local receptive field), trọng số chia sẻ (shared weights) và tổng hợp (pooling). Chúng ta hãy xem xét lần lượt từng ý tưởng.

Trường tiếp nhận cục bộ (Local receptive fields): Trong các tầng kết nối đầy đủ được chỉ ra trước đây, đầu vào đã được mô tả là một đường thẳng đứng chứa các nơron. Trong mạng tích chập, ta sẽ thay thế các đầu vào là 28×28 nơron, giá trị tương ứng với 28×28 cường độ điểm ảnh mà chúng ta sử dụng:

Như thường lệ chúng ta sẽ kết nối các điểm ảnh đầu vào cho các nơron ở tầng ẩn. Nhưng chúng ta sẽ không kết nối mỗi điểm ảnh đầu vào cho mỗi neuron ẩn. Thay vào đó, chúng ta chỉ kết nối trong phạm vi nhỏ, các vùng cục bộ của bức ảnh.

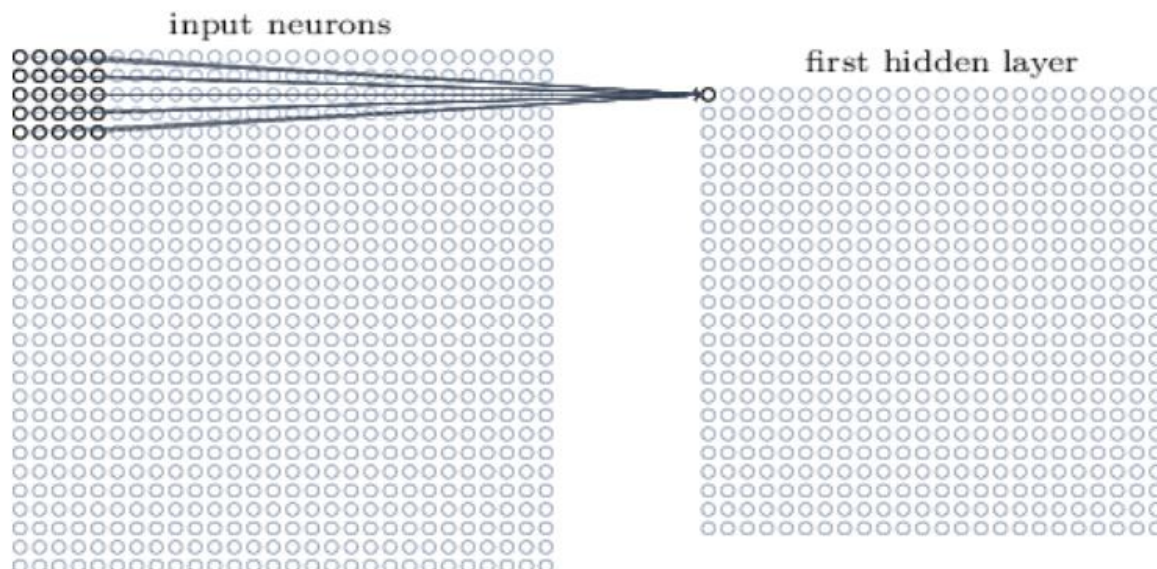
Để được chính xác hơn, mỗi nơron trong lớp ẩn đầu tiên sẽ được kết nối với một vùng nhỏ của các nơron đầu vào, ví dụ, một vùng 5×5 , tương ứng với 25 điểm ảnh đầu vào. Vì vậy, đối với một nơron ẩn cụ thể, chúng ta có thể có các kết nối như sau:



Hình 2.6 Kết nối các điểm ẩn

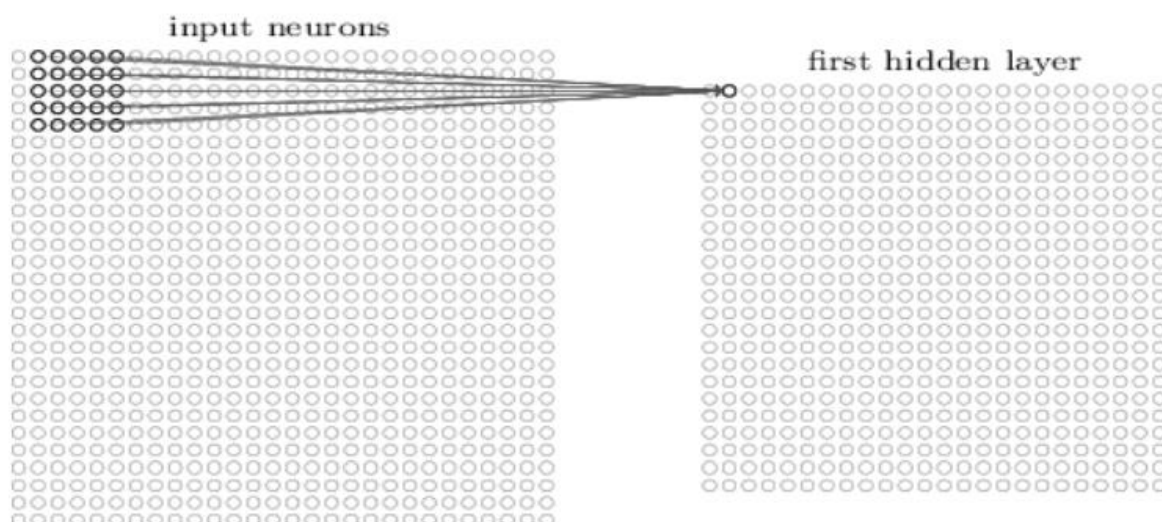
Vùng đó trong bức ảnh đầu vào được gọi là vùng tiếp nhận cục bộ cho neuron ẩn. Đó là một cửa sổ nhỏ trên các điểm ảnh đầu vào. Mỗi kết nối sẽ học một trọng số. Và neuron ẩn cũng sẽ học một độ lệch (overall bias). Bạn có thể hiểu rằng neuron lớp ẩn cụ thể là học để phân tích trường tiếp nhận cục bộ cụ thể của nó.

Sau đó chúng ta trượt trường tiếp nhận cục bộ trên toàn bộ bức ảnh. Đối với mỗi trường tiếp nhận cục bộ, có một neuron ẩn khác trong tầng ẩn đầu tiên. Để minh họa điều này một cách cụ thể, chúng ta hãy bắt đầu với một trường tiếp nhận cục bộ ở góc trên bên trái:



Hình 2.7 Trường tiếp nhận cục bộ

Sau đó, chúng ta trượt trường tiếp nhận cục bộ trên bởi một điểm ảnh bên phải (tức là bằng một neuron), để kết nối với một neuron ẩn thứ hai:



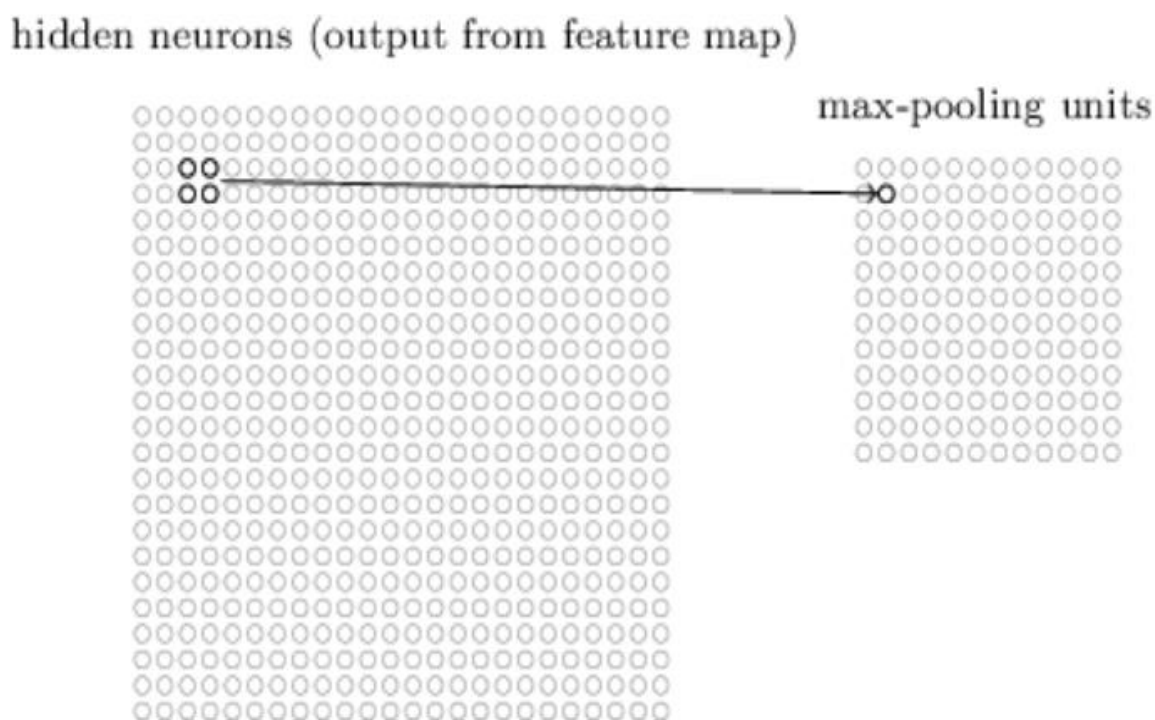
Hình 2.8 Trượt trường tiếp nhận cục bộ

Chúng ta chưa xác định chính xác khái niệm về đặc trưng. Ta có thể nghĩ rằng của đặc trưng là loại mẫu đầu vào mà làm cho neuron hoạt động: ví dụ, nó có thể là biên của ảnh hoặc có thể là một dạng hình khối khác, ngay tại các vị trí khác nhau của ảnh đầu vào. Tại sao điều này lại có lí, giả sử rằng các trọng số và độ lệch mà các neuron ẩn chọn ra, một biên thẳng đứng (vertical edge) trong trường tiếp nhận cục bộ. Khả năng đó rất hữu ích ở các vị trí khác nhau trong bức ảnh. Và do đó, nó là hữu ích để áp dụng phát hiện các đặc trưng giống nhau trong ảnh. Để đặt nó trong thuật ngữ trừu tượng hơn một chút, mạng chập được thích nghi với bất biến dịch (translation invariance) của các ảnh: di chuyển ảnh của một con mèo một ít, và nó vẫn là một hình ảnh của một con mèo.

Cái tên “convolutional” xuất phát là các hoạt động trong phương trình đôi khi được biết đến như convolution. Chính xác hơn một chút, người ta đôi khi viết phương trình như $a_1 = \sigma(b + w * a_0)$, trong đó a_1 là tập kích hoạt đầu ra từ một bản đồ đặc trưng, a_0 là tập hợp các kích hoạt đầu vào, và $*$ được gọi là phép toán chập.

Lớp chứa hay lớp tổng hợp (Pooling layer): Ngoài các lớp tích chập vừa mô tả, mạng nơron tích chập cũng chứa các lớp pooling. Lớp pooling thường được sử dụng ngay sau lớp tích chập. Những gì các lớp pooling làm là đơn giản hóa các thông tin ở đầu ra từ các lớp tích chập.

Ví dụ, mỗi đơn vị trong lớp pooling có thể thu gọn một vùng 2×2 nơron trong lớp trước. Một thủ tục pooling phổ biến là max-pooling. Trong maxpooling, một đơn vị pooling chỉ đơn giản là kết quả đầu ra kích hoạt giá trị lớn nhất trong vùng đầu vào 2×2 , như minh họa trong sơ đồ sau:



Hình 2.9 Đơn giản hóa thông tin đầu ra từ lớp tích chập

Chúng ta có thể hiểu max-pooling như là một cách cho mạng để hỏi xem một đặc trưng nhất được tìm thấy ở bất cứ đâu trong một khu vực của ảnh. Sau đó nó bỏ đi những thông tin định vị chính xác. Trực giác là một khi một đặc trưng đã được tìm thấy, vị trí chính xác của nó là không quan trọng như vị trí thô của nó so với các đặc trưng khác. Một lợi ích lớn là có rất nhiều tính năng gộp ít hơn (fewer pooled features), và vì vậy điều này sẽ giúp giảm số lượng các tham số cần thiết trong các lớp sau.

Max-pooling không phải là kỹ thuật duy nhất được sử dụng để pooling. Một phương pháp phổ biến khác được gọi là L2 pooling. Ở đây, thay vì lấy giá trị kích hoạt tối đa (maximum activation) của một vùng 2×2 nơron, chúng ta lấy căn bậc hai của tổng các bình phương của kích hoạt trong vùng 2×2 . Trong khi các chi tiết thì khác nhau, nhưng về trực giác thì tương tự như max-pooling: L2 pooling là một cách để cô đọng thông tin từ các lớp

tích chập. Trong thực tế, cả hai kỹ thuật đã được sử dụng rộng rãi. Và đôi khi người ta sử dụng các loại pooling khác.

2.1.8 EPOCH

Epoch (vòng lặp) là một khái niệm quan trọng để đo lường mức độ hoàn thiện của quá trình huấn luyện mô hình học máy hoặc học sâu. Mỗi epoch tương ứng với việc đưa toàn bộ dữ liệu huấn luyện qua mô hình một lần duy nhất. Trong quá trình mỗi epoch, mô hình được cập nhật dựa trên sự sai khác giữa dự đoán và giá trị thực tế (ground truth) của dữ liệu.

Khi bạn huấn luyện một mô hình, bạn không thể đưa toàn bộ dữ liệu vào mô hình một lần duy nhất vì dung lượng bộ nhớ và tốc độ tính toán. Thay vào đó, bạn chia dữ liệu thành các "batch" (phần), và trong mỗi epoch, mô hình được huấn luyện trên từng batch một. Sau khi mọi batch đã được sử dụng một lần, một epoch được coi là đã hoàn thành.

2.2 SƠ LƯỢC VỀ BỘ DATASET USK-COFFEE

2.2.1 CẤU TRÚC TẬP DỮ LIỆU:

Bảng 1.1 Bảng chi tiết của một tập train, test, validation

Phân loại	Tổng cộng	Số chiều
Peaberry	2,000	256 x 256 pixels
Longberry	2,000	256 x 256 pixels
Premium	2,000	256 x 256 pixels
Defect	2,000	256 x 256 pixels

Bảng 1.2 Bảng chi tiết của tập dữ liệu USK-COFFEE

Phân loại	Tập huấn luyện	Tập kiểm chứng	Tập kiểm thử
Peaberry	1200	400	400
Longberry	1200	400	400
Premium	1200	400	400
Defect	1200	400	400
Tổng cộng	4800	1600	1600
Tổng số dữ liệu	8000		

2.2.2 ĐẶC ĐIỂM TỪNG LỚP TRONG TẬP DỮ LIỆU:

- Hạt cà phê Peaberry có hình dạng giống như hạt đậu, nhỏ và tròn.
- Hạt cà phê Longberry dài, rộng, và có hạt kép.
- Cà phê hảo hạng (Premium) có hình dạng lớn và tròn hơn so với 2 loại Peaberry và Longberry.
- Cà phê bị lỗi (defect) bị biến dạng và thường có màu đen. [1]

CHƯƠNG 3 : CÀI ĐẶT, XÂY DỰNG VÀ THỰC NGHIỆM

3.1 YÊU CẦU

- Phân tích và xây dựng model của dataset USK-Coffee sử dụng **GOOGLE COLAB**.
- Môi trường phát triển tích hợp (IDEs) sử dụng **Visual Studio Code**.
- Ngôn ngữ lập trình, framework sử dụng là **Python và Flask**.
- Nơi lưu trữ source code và tài liệu tham khảo là **Github**

3.2 XÂY DỰNG MÔ HÌNH CNN

```
model = models.Sequential([
    resize_and_rescale,
    data_augmentation,
    layers.Conv2D(32, (3,3), activation='relu', input_shape = input_shape),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, kernel_size = (3,3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Conv2D(64, (3, 3), activation='relu'),
    layers.MaxPooling2D((2, 2)),
    layers.Flatten(),
    layers.Dense(64, activation='relu'),
    layers.Dense(n_classes, activation='softmax'),
])
```

Hình 3.1 Các lớp của mô hình

- `resize_and_rescale`: Đây là một lớp tiền xử lý dữ liệu đầu vào, thực hiện việc điều chỉnh kích thước và tỉ lệ pixel của ảnh.
- `data_augmentation`: Đây là một lớp tiền xử lý khác cho việc tăng cường dữ liệu. Tăng cường dữ liệu (data augmentation) là kỹ thuật tạo ra các biến thể của dữ liệu huấn luyện bằng cách thực hiện các phép biến đổi đơn giản như xoay, cắt, lật, v.v. Nhờ việc này, mô hình sẽ được huấn luyện trên các phiên bản biến đổi của ảnh, giúp nâng cao khả năng tổng quát hóa và tránh tình trạng overfitting.
- `layers.Conv2D(32, (3,3), activation='relu', input_shape = input_shape)`: Đây là một lớp tích chập với 32 bộ lọc (filter) có kích thước 3x3, và hàm kích hoạt là hàm ReLU (Rectified Linear Activation). Lớp này được sử dụng để trích xuất các đặc trưng từ ảnh đầu vào. `input_shape` là kích thước của ảnh đầu vào.
- `layers.MaxPooling2D((2, 2))`: Đây là lớp giảm kích thước của ảnh bằng cách lấy giá trị lớn nhất trong các vùng 2x2 không gồm chồng chéo. Lớp này giúp giảm số lượng tham số và tính toán trong mạng, giúp mô hình học được các đặc trưng chung và bền

vững hơn.

-
- Các bước 3 và 4 lặp lại 6 lần, tạo thành 6 lớp tích chập và lớp giảm kích thước tương ứng. Các lớp tích chập có số lượng bộ lọc tăng dần từ 32 đến 64, nhằm cải thiện khả năng trích xuất đặc trưng của mạng.
- `layers.Flatten()`: Đây là lớp để biến đổi các đặc trưng thành một vector 1 chiều để sử dụng cho các lớp kết nối đầy đủ (fully connected layers).
- `layers.Dense(64, activation='relu')`: Đây là một lớp kết nối đầy đủ với 64 đơn vị ẩn và hàm kích hoạt là ReLU. Lớp này giúp học các mối quan hệ phức tạp giữa các đặc trưng đã trích xuất và thực hiện việc biểu diễn dạng số để tiến hành phân loại.
- `layers.Dense(n_classes, activation='softmax')`: Đây là lớp đầu ra với số đơn vị đầu ra bằng với số lớp cần phân loại (`n_classes`). Hàm kích hoạt ở đây là softmax, giúp đưa ra xác suất phân loại của dữ liệu vào từng lớp.

Layer (type)	Output Shape	Param #
sequential (Sequential)	(32, 256, 256, 3)	0
sequential_1 (Sequential)	(32, 256, 256, 3)	0
conv2d (Conv2D)	(32, 254, 254, 32)	896
max_pooling2d (MaxPooling2D)	(32, 127, 127, 32)	0
conv2d_1 (Conv2D)	(32, 125, 125, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(32, 62, 62, 64)	0
conv2d_2 (Conv2D)	(32, 60, 60, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(32, 30, 30, 64)	0
conv2d_3 (Conv2D)	(32, 28, 28, 64)	36928
max_pooling2d_3 (MaxPooling2D)	(32, 14, 14, 64)	0
conv2d_4 (Conv2D)	(32, 12, 12, 64)	36928
max_pooling2d_4 (MaxPooling2D)	(32, 6, 6, 64)	0
conv2d_5 (Conv2D)	(32, 4, 4, 64)	36928
max_pooling2d_5 (MaxPooling2D)	(32, 2, 2, 64)	0
flatten (Flatten)	(32, 256)	0
dense (Dense)	(32, 64)	16448
dense_1 (Dense)	(32, 4)	260
=====		
Total params: 183,812		
Trainable params: 183,812		
Non-trainable params: 0		

Hình 3.2 Kiến trúc mô hình

3.3 IMPORT

- import tensorflow as tf
 - TensorFlow là một thư viện mã nguồn mở chuyên về tính toán số học và học sâu. Nó cung cấp các công cụ và khung làm việc mạnh mẽ để xây dựng, huấn luyện và triển khai các mô hình học máy và học sâu.
- from tensorflow.keras import models, layers
 - tensorflow.keras: Keras là một API cao cấp để xây dựng và huấn luyện mô hình học máy. Keras được tích hợp sâu vào TensorFlow và được gọi là tensorflow.keras. Nó cung cấp một cách dễ dàng và trực quan để định nghĩa, xây dựng và huấn luyện mô hình học sâu bằng cách sử dụng các lớp và phương thức cung cấp sẵn.
- import matplotlib.pyplot as plt
 - matplotlib.pyplot: Thư viện matplotlib là một thư viện trực quan hóa dữ liệu. matplotlib.pyplot được sử dụng để vẽ đồ thị và biểu đồ để hiển thị các thông số hoặc dữ liệu liên quan đến huấn luyện và đánh giá mô hình.
- import numpy as np
 - numpy: NumPy là một thư viện cung cấp hỗ trợ cho các phép toán số học và xử lý mảng nhiều chiều.
- import os
 - os: Thư viện os cung cấp các chức năng để tương tác với hệ điều hành, ví dụ như thao tác với các tệp và thư mục.

3.4 PHÂN TÍCH MODEL SAU KHI HUẤN LUYỆN:

- Khi mới bắt đầu huấn luyện:

```
Epoch 1/100
151/151 [=====] - 39s 98ms/step - loss: 1.3875 - accuracy: 0.2391 - val_loss: 1.3854 - val_accuracy: 0.3006
Epoch 2/100
151/151 [=====] - 10s 65ms/step - loss: 1.3793 - accuracy: 0.2649 - val_loss: 1.3866 - val_accuracy: 0.2500
Epoch 3/100
151/151 [=====] - 10s 65ms/step - loss: 1.3867 - accuracy: 0.2445 - val_loss: 1.3864 - val_accuracy: 0.2500
Epoch 4/100
151/151 [=====] - 10s 65ms/step - loss: 1.3865 - accuracy: 0.2566 - val_loss: 1.3866 - val_accuracy: 0.2519
Epoch 5/100
151/151 [=====] - 10s 65ms/step - loss: 1.3605 - accuracy: 0.3070 - val_loss: 1.3740 - val_accuracy: 0.2656
Epoch 6/100
151/151 [=====] - 10s 65ms/step - loss: 1.3187 - accuracy: 0.3474 - val_loss: 1.5122 - val_accuracy: 0.2819
Epoch 7/100
151/151 [=====] - 10s 67ms/step - loss: 1.2434 - accuracy: 0.4214 - val_loss: 1.2188 - val_accuracy: 0.4406
Epoch 8/100
151/151 [=====] - 10s 65ms/step - loss: 1.2268 - accuracy: 0.4572 - val_loss: 1.2194 - val_accuracy: 0.4444
Epoch 9/100
151/151 [=====] - 10s 65ms/step - loss: 1.1209 - accuracy: 0.5122 - val_loss: 1.1704 - val_accuracy: 0.4600
Epoch 10/100
151/151 [=====] - 10s 65ms/step - loss: 1.0845 - accuracy: 0.5411 - val_loss: 1.1179 - val_accuracy: 0.4881
```

Hình 3.3 Bắt đầu huấn luyện tập dữ liệu

- Khi vừa bắt đầu huấn luyện thì độ chính xác của model ở những epoch đầu tiên khá là thấp, độ chính xác chỉ khoảng 23%.
- Sau khi huấn luyện xong:

```
Epoch 90/100
151/151 [=====] - 10s 64ms/step - loss: 0.2932 - accuracy: 0.8925 - val_loss: 0.5277 - val_accuracy: 0.8181
Epoch 91/100
151/151 [=====] - 10s 65ms/step - loss: 0.2847 - accuracy: 0.8959 - val_loss: 0.5497 - val_accuracy: 0.8056
Epoch 92/100
151/151 [=====] - 10s 65ms/step - loss: 0.2717 - accuracy: 0.9023 - val_loss: 0.4449 - val_accuracy: 0.8438
Epoch 93/100
151/151 [=====] - 10s 65ms/step - loss: 0.2897 - accuracy: 0.8973 - val_loss: 0.5549 - val_accuracy: 0.7975
Epoch 94/100
151/151 [=====] - 10s 65ms/step - loss: 0.2833 - accuracy: 0.8998 - val_loss: 0.5478 - val_accuracy: 0.8037
Epoch 95/100
151/151 [=====] - 10s 65ms/step - loss: 0.2624 - accuracy: 0.9054 - val_loss: 0.3994 - val_accuracy: 0.8612
Epoch 96/100
151/151 [=====] - 10s 65ms/step - loss: 0.2814 - accuracy: 0.8998 - val_loss: 0.4571 - val_accuracy: 0.8494
Epoch 97/100
151/151 [=====] - 10s 65ms/step - loss: 0.2703 - accuracy: 0.9015 - val_loss: 0.5725 - val_accuracy: 0.8094
Epoch 98/100
151/151 [=====] - 10s 65ms/step - loss: 0.2704 - accuracy: 0.8979 - val_loss: 0.4838 - val_accuracy: 0.8381
Epoch 99/100
151/151 [=====] - 10s 65ms/step - loss: 0.2625 - accuracy: 0.9069 - val_loss: 0.5045 - val_accuracy: 0.8263
Epoch 100/100
151/151 [=====] - 10s 65ms/step - loss: 0.2563 - accuracy: 0.9038 - val_loss: 0.5801 - val_accuracy: 0.8037
```

Hình 3.4 Sau khi huấn luyện tập dữ liệu

- Sau mỗi epoch thì hệ thống sẽ tự động điều chỉnh tham số của bộ lọc để tối ưu hóa model và sau khi kết thúc 100 epoch thì độ chính xác đã lên đến hơn 90%.
- Sau khi chạy qua tập kiểm thử:

```
50/50 [=====] - 4s 22ms/step - loss: 0.4738 - accuracy: 0.8250
```

Hình 3.5 Kết quả sau khi chạy qua tập thử nghiệm

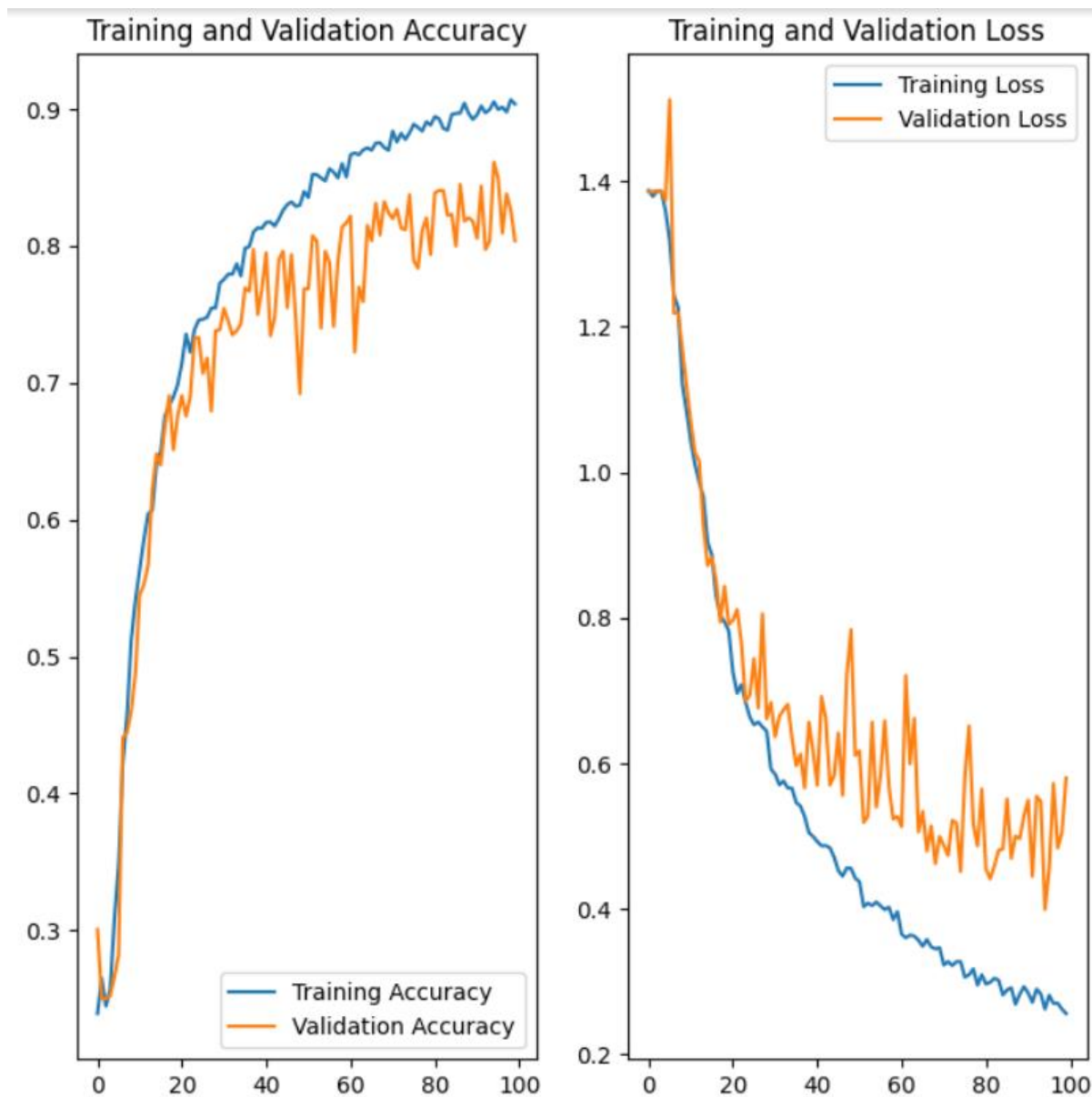
- Sau khi chạy qua tập kiểm thử để kiểm tra độ chính xác của model thì độ chính xác chỉ đạt hơn 82%, sai số khá là lớn, lý do là do đâu? Đó là vì dữ liệu huấn luyện không đạt chuẩn do có quá nhiều dữ liệu bị gán nhãn sai. Ví dụ:
 - Ở nhãn premium của dữ liệu huấn luyện thì có nhiều bức hình của những hạt hồng nên sẽ gây giảm hiệu suất của model. Điều này cũng xảy ra tương tự ở cả nhãn longberry, peaberry và defect.



Hình 3.6 Hình ảnh gây nhiễu cho tập huấn luyện

3.5 DATA VISUALIZATION

3.5.1 ĐỘ CHÍNH XÁC VÀ HÀM MẤT MÁT



Hình 3.7 Đồ thị về độ chính xác và hàm mất mát trên tập huấn luyện và tập kiểm tra

- Dựa vào đồ thị trên ta thấy được hiệu suất của model tăng lên sau mỗi lần lặp.
- Độ chính xác của tập huấn luyện với độ chính xác của tập kiểm thử có sự chênh lệch nhẹ khi độ chính xác của mô hình tăng lên, một trong số nguyên nhân đó có thể là do tập huấn luyện chưa được chuẩn. (giải thích đã được nêu rõ ở mục 4.2)

3.5.2 CONFUSION MATRIX

Cách tính sử dụng accuracy như ở trên chỉ cho chúng ta biết được bao nhiêu phần trăm lượng dữ liệu được phân loại đúng mà không chỉ ra được cụ thể mỗi loại được phân loại như thế nào, lớp nào được phân loại đúng nhiều nhất, và dữ liệu thuộc lớp nào thường bị phân loại

nhầm vào lớp khác. Để có thể đánh giá được các giá trị này, chúng ta sử dụng một ma trận được gọi là confusion matrix.

Về cơ bản, confusion matrix thể hiện có bao nhiêu điểm dữ liệu thực sự thuộc vào một class, và được dự đoán là rơi vào một class.

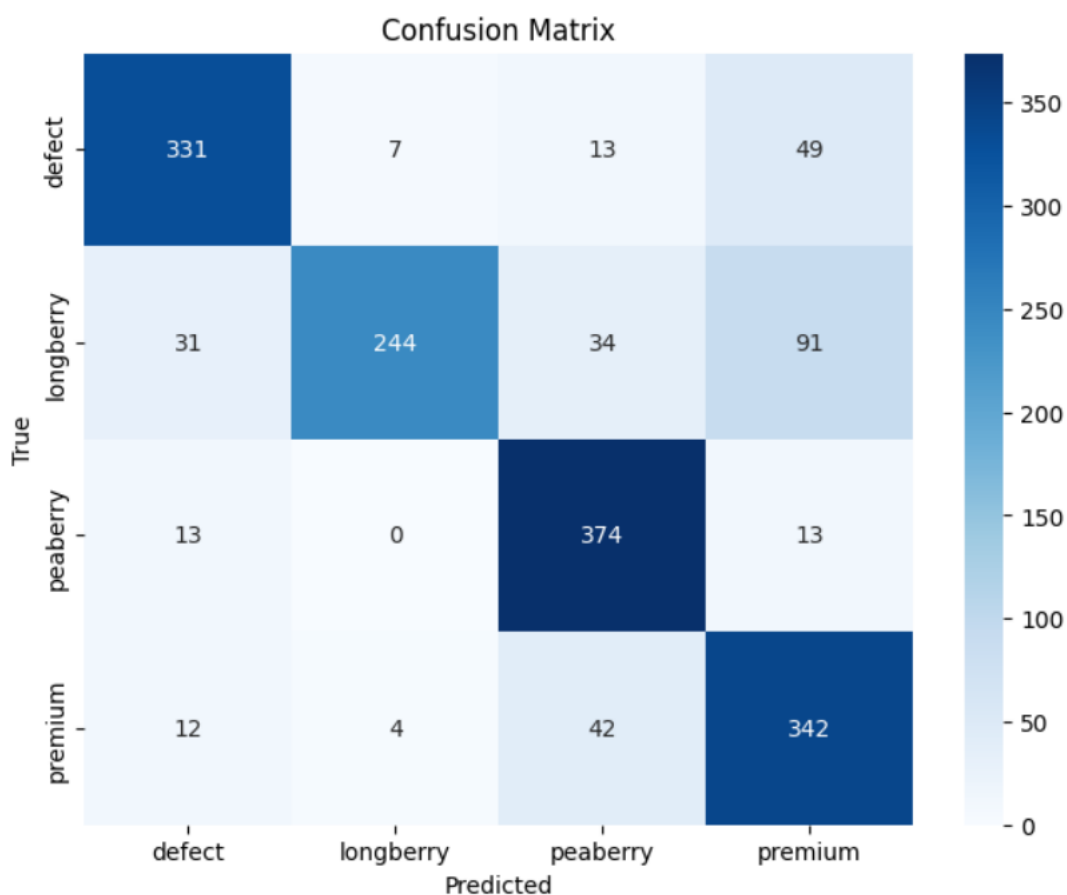
Một confusion matrix gồm 4 chỉ số sau đối với mỗi lớp phân loại:

- TP (True Positive): Số lượng dự đoán chính xác.
- TN (True Negative): Số lượng dự đoán chính xác một cách gián tiếp.
- FP (False Positive - Type 1 Error): Số lượng các dự đoán sai lệch.
- FN (False Negative - Type 2 Error): Số lượng các dự đoán sai lệch một cách gián tiếp.
- Precision: Trong tất cả các dự đoán Positive được đưa ra, bao nhiêu dự đoán là chính xác? Chỉ số này được tính theo công thức:

$$\text{Precision} = \frac{TP}{TP+FP}$$

- Recall: Trong tất cả các trường hợp Positive, bao nhiêu trường hợp đã được dự đoán chính xác? Chỉ số này được tính theo công thức:

$$\text{Recall} = \frac{TP}{TP+FN}$$



Hình 4.6 Ma trận nhầm lẫn của mô hình

Các ô màu đậm thể hiện các giá trị cao. Một mô hình tốt sẽ cho một confusion matrix có các phần tử trên đường chéo chính có giá trị lớn, các phần tử còn lại có giá trị nhỏ. Nói cách khác, khi biểu diễn bằng màu sắc, đường chéo có màu càng đậm so với phần còn lại sẽ càng tốt. Từ hình trên ta thấy rằng confusion matrix đã chuẩn hoá mang nhiều thông tin hơn.

- Accuracy (Độ chính xác):
 - o Tổng số lượng mẫu: $331 + 7 + 13 + 49 + 31 + 244 + 34 + 91 + 13 + 0 + 374 + 13 + 12 + 4 + 42 + 342 = 1600$
 - o Số lượng dự đoán đúng: $331 + 244 + 374 + 342 = 1291$
 - o Accuracy (Độ chính xác): $(\text{Số lượng dự đoán đúng}) / (\text{Tổng số lượng mẫu}) = 1291 / 1600 \approx 80.69\%$
- Precision (Độ chính xác trong dự đoán positive):
 - o Precision của "defect": $331 / (331 + 31 + 13 + 12) \approx 87.63\%$
 - o Precision của "longberry": $244 / (7 + 244 + 0 + 4) \approx 95.36\%$
 - o Precision của "peaberry": $374 / (13 + 34 + 374 + 42) \approx 84.74\%$
 - o Precision của "premium": $342 / (49 + 91 + 13 + 342) \approx 66.67\%$
- Recall (Tỷ lệ phát hiện - Sensitivity - True Positive Rate):
 - o Recall của "defect": $331 / (331 + 7 + 13 + 49) \approx 80.93\%$
 - o Recall của "longberry": $244 / (31 + 244 + 34 + 91) \approx 57.95\%$
 - o Recall của "peaberry": $374 / (13 + 0 + 374 + 13) \approx 95.42\%$
 - o Recall của "premium": $342 / (12 + 4 + 42 + 342) \approx 86.23\%$
- F1-score:
 - o F1-score của "defect": $2 * (0.8763 * 0.8093) / (0.8763 + 0.8093) \approx 84.04\%$
 - o F1-score của "longberry": $2 * (0.9536 * 0.5795) / (0.9536 + 0.5795) \approx 71.72\%$
 - o F1-score của "peaberry": $2 * (0.8474 * 0.9542) / (0.8474 + 0.9542) \approx 89.67\%$
 - o F1-score của "premium": $2 * (0.6667 * 0.8623) / (0.6667 + 0.8623) \approx 75.71\%$
- Kết luận:
 - o Lớp "longberry": Precision và F1-score đối với lớp này khá cao, nhưng recall lại thấp. Điều này cho thấy mô hình không phát hiện được một phần lớn các mẫu thực tế của lớp "longberry".
 - o Lớp "premium": Precision thấp hơn so với các lớp khác, điều này cho thấy có một số lượng lớn các dự đoán tích cực trong lớp này là sai.

3.6 ĐÁNH GIÁ THỰC NGHIỆM

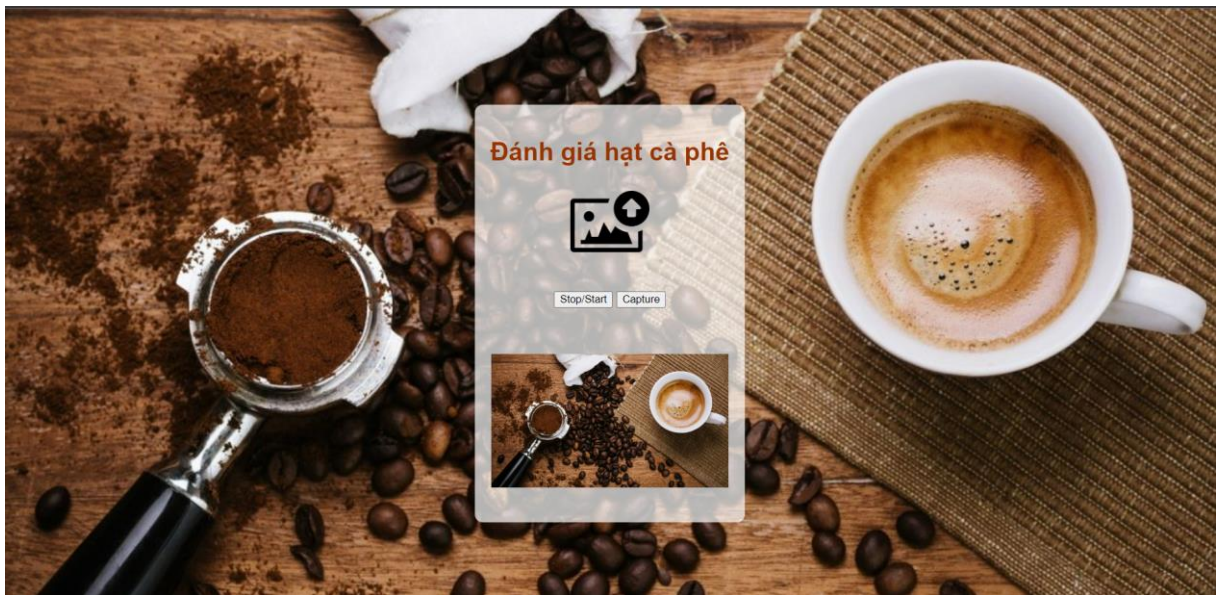


Hình 3.8 Kết quả sau khi chạy qua một số bức ảnh của tập kiểm thử

- Sau khi chạy qua một số bức ảnh, ta nhận định được rằng model khá tốt và đã tự phân loại được hạt cà phê với tỷ lệ khá cao.

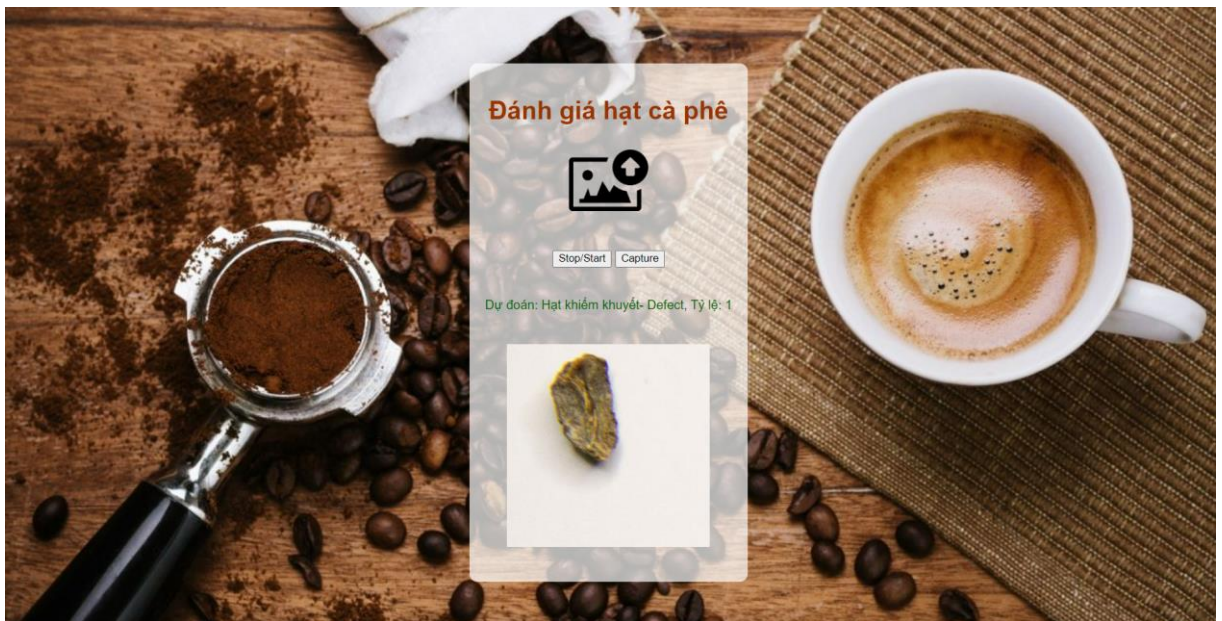
3.7 XÂY DỰNG GIAO DIỆN

- Giao diện chính:



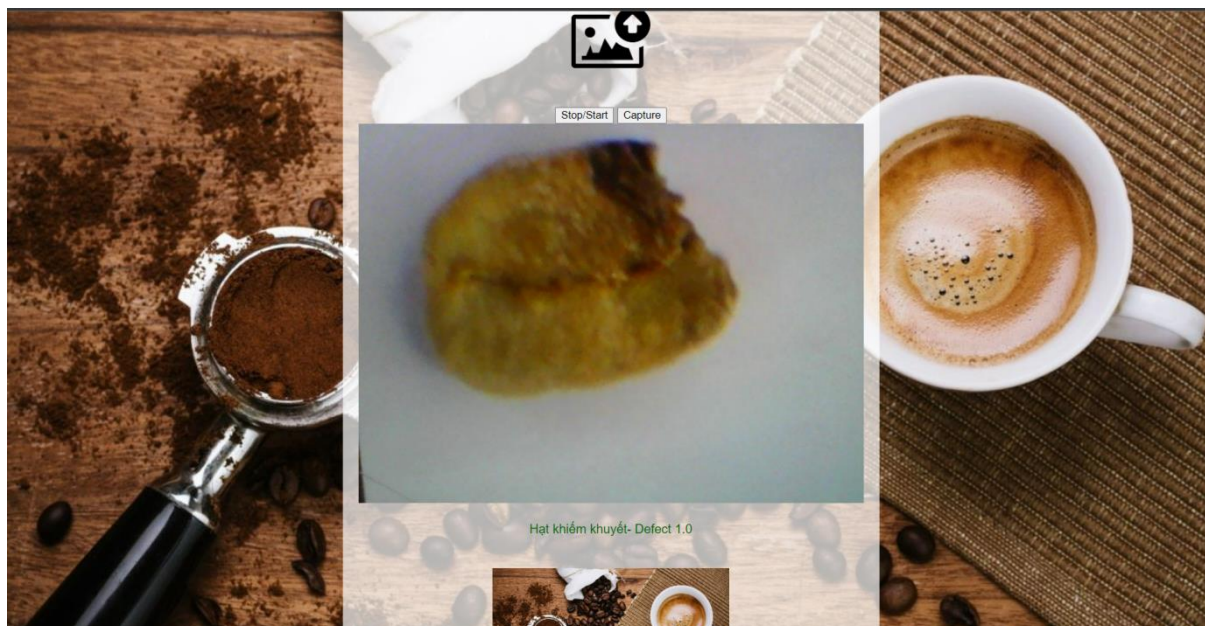
Hình 3.9 Giao diện chính

- Chức năng tải lên một bức ảnh để phân loại hạt cà phê:



Hình 3.10 Giao diện khi tải lên một bức ảnh

- Chức năng sử dụng camera máy tính để phân loại hạt cà phê:



Bảng 3.11 Giao diện khi sử dụng camera máy tính

CHƯƠNG 4: KẾT LUẬN

4.1 TÓM TẮT KẾT QUẢ ĐẠT ĐƯỢC

- Xây dựng được Model phân loại bốn loại hạt cà phê (peaberry, longberry, premium và defect). Xây dựng bằng mô hình CNN với độ chính xác hơn 90%.
- Xây dựng được giao diện trang web với hai chức năng.
 - o Tải ảnh có sẵn trên máy lên và hệ thống sẽ trả về dự đoán.
 - o Sử dụng camera máy tính để chụp lại bức ảnh và tải bức ảnh vừa chụp được lên hệ thống để trả về dự đoán.

4.2 HƯỚNG PHÁT TRIỂN TRONG TƯƠNG LAI

- Huấn luyện tập dữ liệu: Loại bỏ bớt những dữ liệu gây nhiễu làm ảnh hưởng đến hiệu suất của mô hình, để gia tăng tỷ lệ dự đoán đúng.
- App web:
 - o Phát triển thêm các chức năng khác như có thể xử lý nhiều file ảnh truyền lên cùng.
 - o Có thể truyền lên dạng file nén như zip hoặc rar.
 - o Khoanh vùng hạt cà phê khi sử dụng camera.

TÀI LIỆU THAM KHẢO

[1] Alifya Febriana, Kahlil Muchtar, Rahmad Dawood, Chih-Yang Lin, *USK-COFFEE Dataset: A Multi-Class Green Arabica Coffee Bean Dataset for Deep Learning*

DANH MỤC CÁC WEBSITE THAM KHẢO

<https://comvis.unsyiah.ac.id/usk-coffee/>

https://www.tensorflow.org/api_docs/python/tf/keras/utils/image_dataset_from_directory