HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

PHẠM NGỌC HOÀN

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

HÀ NỘI – 2023

HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG

PHẠM NGỌC HOÀN

NGHIÊN CỨU MÔ HÌNH HỌC SÂU VÀ ỨNG DỤNG BIGDL CHO BÀI TOÁN NHẬN DIỆN VÀ PHÂN LOẠI NÔNG SẢN

CHUYÊN NGÀNH : KHOA HỌC MÁY TÍNH

MÃ SỐ : 8.48.01.01

LUẬN VĂN THẠC SĨ KỸ THUẬT

(Theo định hướng ứng dụng)

NGƯỜI HƯỚNG DẪN KHOA HỌC

PGS. TS. NGUYỄN VĂN THỦY

HÀ NỘI – 2023

LỜI CAM ĐOAN

Tôi xin cam đoan Luận văn thạc sĩ với đề tài: "***Nghiên cứu mô hình học sâu và ứng dụng BIGDL cho bài toán nhận diện và phân loại nông sản***" dưới sự hướng dẫn của thầy PGS. TS. Nguyễn Văn Thủy là công trình nghiên cứu của riêng tôi. Các kết quả nghiên cứu trong luận văn là trung thực, các tài liệu tham khảo được trích dẫn đầy đủ.

Hà Nội, ngày tháng năm 2023

Học viên

Phạm Ngọc Hoàn

LỜI CẢM ƠN

Đầu tiên, tôi xin được gửi lời cảm ơn sâu sắc đến Học viện công nghệ Bưu chính Viễn thông nói chung và các thầy cô đã giảng dạy tôi nói riêng, Thầy/Cô đã truyền đạt những kiến thức và kinh nghiệm quý báu trong suốt quá trình tôi học tập tại Học viện.

Tôi xin được gửi lời tri ân sâu sắc đến thầy giáo PGS. TS. Nguyễn Văn Thủy, người đã dìu dắt và hướng dẫn tôi trong suốt quá trình thực hiện luận văn. Sự chỉ bảo và định hướng của thầy đã giúp tôi nghiên cứu và giải quyết các vấn đề một cách khoa học và đúng đắn hơn.

Tiếp theo, tôi xin được gửi lời cảm ơn tới bố mẹ, vợ và anh chị em đồng nghiệp đã luôn động viên, giúp đỡ tôi vượt qua những khó khăn trong học tập, công việc và cuộc sống.

Trong quá trình thực hiện luận văn, dù đã rất cố gắng nhưng không thể tránh khỏi những thiếu sót, tôi rất mong nhận được sự đóng góp ý kiến từ Thầy/Cô và các bạn để luận văn của tôi được hoàn thiện hơn.

Tôi xin chân thành cảm ơn!

Hà Nội, ngày tháng năm 2023

Học viên

Phạm Ngọc Hoàn

MỤC LỤC

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 – Tinh chỉnh đơn vị tuyến tính
4	GPU	Graphics Processing Unit – Bộ vi xử lý đồ họa

DANH MỤC HÌNH VẼ

DANH MỤC BẢNG BIỂU

MỞ ĐẦU

1. Tính cấp thiết của đề tài

Hiện nay, ở nước ta nói riêng và ở các nước đang phát triển có nền nông nghiệp là một trong các ngành sản xuất chủ yếu, quá trình thu hoạch, phân loại, đánh giá chất lượng các loại sản phẩm nông nghiệp, đặc biệt là các loại nông sản, chủ yếu còn phải thực hiện bằng các phương pháp thủ công. Đây là công việc không quá khó, nhưng tiêu tốn nhiều thời gian, công sức của con người và là rào cản đối với mở rộng phát triển quy mô sản xuất nông nghiệp. Do đó, nhiều phương pháp tự động hóa công việc thu hoạch, nhận diện và đánh giá chất lượng nông sản 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 diện một số lượng lớn các loại nông sản với độ chính xác cao do bị hạn chế bởi các đặc trưng của bài toán nhận diện nông sản, số lượng chủng loại lớn với nhiều nông sản hết sức tương tự nhau, sự biến thiên về hình dạng, màu sắc, chi tiết của từng nông sản cũng rất khó dự đoán trước…

Trong thời gian gần đây, nhờ 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. Nối tiếp sự phát triển của Học máy, 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 có ứng dụng sâu rộng trong các lĩnh vực của đời sống như tìm kiếm sự khác nhau giữa các văn bản, phát hiện gian lận, phát hiện spam, nhận dạng chữ viết, giọng nói, nhận dạng hình ảnh,… góp phần quan trọng trong việc hỗ trợ con người trong nhiều lĩnh vực đời sống. Từ những ứng dụng thực

tế và những lợi ích mà Học sâu đem lại, đề tài nghiên cứu "*Nghiên cứu mô hình học sâu và ứng dụng BIGDL cho bài toán nhận diện và phân loại nông sản* " 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 diện nông sản tự động.

2. Tổng quan về vấn đề nghiên cứu

Nhận diện 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: sự đa dạng trong điểm nhìn, đa dạng trong kích thước, các điều kiện khác nhau của ánh sáng, sự lộn xộn phức tạp của nền,…

Bộ cơ sở dữ liệu ảnh là một trong các thành phần quan trọng hàng đầu trong các phương pháp Học máy bao gồm cả Học sâu, được sử dụng để phục vụ cho quá trình tính toán tham số, huấn luyện và 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.

Là một trường hợp cụ thể của bài toán nhận diện và phân lớp, bài toán nhận diện nông sản kế thừa các khó khăn vốn có của bài toán gốc, và kèm theo là các khó khăn riêng của chính nó, như: yêu cầu một bộ cơ sở dữ liệu lớn về chủng loại nông sản theo mùa, vùng miền, địa hình… với vô số loại nông sản 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… Nhằm tăng cường kích thước cho bộ CSDL, tạo điều kiện cho việc huấn luyện các mô hình, sau khi thu thập đủ số lượng ảnh cho các loại nông sản, một số thuật toán chỉnh sửa ảnh sẽ được áp dụng, như làm nghiêng

ảnh, chèn thêm nhiễu hoặc ghép ảnh với các nền khác.

Bên cạnh sự chuẩn bị về bộ cơ sở dữ liệu, việc lựa chọn mô hình học sâu cũng được xem như quyết định đến toàn bộ quá trình xử lý, nhận diện vật thể. Những năm qua, mô hình học sâu đặc biệt là mạng nơ-ron tích chập CNNs là mô hình được sử dụng phổ biến cho hiệu quả trong các bài toán phân loại hình ảnh, phân loại văn bản,... Ưu điểm của CNNs là tận dụng được tính năng trích chọn đặc trưng của lớp tích chập và bộ phân lớp được huấn luyện đồng thời. Ý tưởng học cùng lúc đặc trưng và bộ phân lớp có thể hỗ trợ với nhau trong quá trình huấn luyện và quá trình phân lớp tìm ra các tham số phù hợp với các véc-tơ đặc trưng tìm được từ lớp tích chập và ngược lại lớp tích chập điều chỉnh các tham số của lớp tích chập để cho các véc-tơ đặc trưng thu được là tuyến tính phù hợp với bộ phân lớp của lớp cuối cùng .

Mã nguồn mở BIGDL được Intel giới thiệu vào năm 2017, cung cấp hỗ trợ thuật toán học sâu toàn diện trên Apache Spark. Được xây dựng trên nền tảng Apache Spark có khả năng mở rộng cao, BIGDL có thể dễ dàng mở rộng lên hàng trăm hoặc hàng nghìn máy chủ. BIGDL có thể cung cấp hỗ trợ cho các mô hình Học Sâu khác nhau (Ví dụ: Phát hiện, phân loại đối tượng,…). Ngoài ra, nó cũng cho phép huấn luyện và tinh chỉnh lại các mô hình được đào tạo trước đây được gắn với các khuôn khổ và nền tảng cụ thể (Caffe, Torch, TensorFlow,...), sang nền tảng phân tích dữ liệu lớn mục đích chung thông qua BIGDL. Do đó, ứng dụng có thể được tối ưu hóa hoàn toàn để mang lại hiệu suất được tăng tốc đáng kể .

Với những ưu điểm trên, Đề tài nghiên cứu lựa chọn mô hình CNNs áp dụng cho bài toán nhận diện và phân loại nông sản thông qua mã nguồn mở BIGDL.

3. Mục đích nghiên cứu

Đề tài tìm hiểu ứng dụng nhận diện và phân loại nông sản cũng như cách

triển khai công cụ tìm kiếm hình ảnh phần mềm tự động để giảm nguồn nhân lực và đảm bảo chất lượng phần hơn với công việc tìm kiếm bằng tay.

Mục tiêu chính của đề tài là nghiên cứu mô hình học sâu và ứng dụng nhận diện và phân loại nông sản để đạt được tốc độ tìm kiếm nhanh và chuẩn xác nhất để cho người dùng không mất nhiều thời gian tìm kiếm sản phẩm.

- Nghiên cứu về các hệ thống nhận diện hình ảnh.
- Thử nghiệm, đánh giá độ hiệu quả của các thuật toán.
- Xây dựng hệ thống nhận diện và phân loại nông sản tự động.

4. Đối tượng và phạm vi nghiên cứu

➢ Đối tượng nghiên cứu

Đối tượng nghiên cứu của đề tài là mô hình học sâu và ứng dụng được mã nguồn mở BIGDL cho bài toán nhận diện và phân loại nông sản.

➢ Phạm vi nghiên cứu

- Số lượng nông sản sẽ nhận diện: 40 loại nông sản phổ biến ở nước ta như nho, táo, chuối, thanh long…
- Số lượng ảnh gốc cho mỗi loại quả: 500 ảnh, bao gồm các ảnh chụp nông sản ở các góc độ khác nhau với nền tùy ý, có thể lấy từ nguồn trên mạng hoặc tự chụp bằng thiết bị camera cá nhân.

5. Phương pháp nghiên cứu

- Phương pháp nghiên cứu lý thuyết

+ Đọc và phân tích tài liệu về các phương pháp, thuật toán đã từng được sử dụng để xây dựng hệ thống nhận diện hình ảnh.

- Phương pháp thực nghiệm

+ Thử nghiệm và đánh giá độ hiệu quả của các thuật toán.

+ Xây dựng hệ thống nhận diện hình ảnh

CHƯƠNG 1. GIỚI THIỆU TỔNG QUAN

1.1 Bài toán nhận diện và phân loại nông sản

1.1.1. Bài toán nhận dạng vật thể

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 [1] phải kể đến:

- Sự đa dạng trong điểm nhìn – 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 ẩn giấu 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

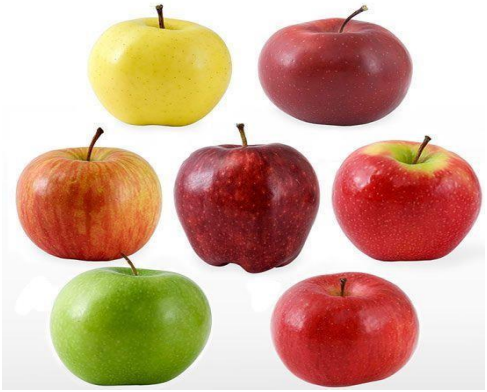
1.1.2. Bài toán nhận diện và phân loại nông sản

Là một trường hợp cụ thể của bài toán nhận dạng và phân lớp, bài toán nhận dạng nông sản kế thừa các khó khăn vốn có của bài toán gốc, và kèm theo là các khó khăn riêng của chính nó, như:

- *Đa dạng về đối tượng:* Trái ngược với bài toán nhận diện hình ảnh thông thường, bài toán nhận diện và phân loại nông sản đòi hỏi phải xử lý một loạt các loại nông sản khác nhau như lúa, hạt điều, cà phê, cà chua, v.v. Mỗi loại nông sản có những đặc trưng và thuộc tính riêng, đòi hỏi mô hình phải có khả năng phân biệt chúng.
- *Quy mô và độ phức tạp:* Bài toán nhận diện nông sản thường đối mặt với quy mô lớn hơn và độ phức tạp cao hơn so với bài toán nhận diện

hình ảnh thông thường. Nó bao gồm nhiều loại nông sản, từ các loại cây trồng đến các sản phẩm nông nghiệp chế biến. Điều này đòi hỏi mô hình phải có khả năng nhận diện và phân loại một loạt các đối tượng phức tạp và đa dạng.

- *Đặc trưng và thuộc tính:* Bài toán nhận diện nông sản thường đòi hỏi phải xem xét những đặc trưng và thuộc tính riêng của từng loại nông sản. Các thuộc tính này có thể bao gồm kích thước, hình dạng, màu sắc, texture, độ chín, độ bị hư hỏng và các thuộc tính khác liên quan đến chất lượng và trạng thái của nông sản.
- *Ứng dụng trong ngành nông nghiệp:* Bài toán nhận diện và phân loại nông sản thường được áp dụng trong ngành nông nghiệp để kiểm tra chất lượng, độ chín, phân loại sản phẩm, quản lý hệ thống sản xuất và tổ chức chuỗi cung ứng nông sản. Do đó, các ứng dụng của bài toán này thường liên quan mật thiết đến nhu cầu thực tế trong lĩnh vực nông nghiệp và thực phẩm.



Hình 1.2: Sự đa dạng về chủng loại của một loại nông sản

Tổng quan, bài toán nhận diện và phân loại nông sản đặt ra những thách thức riêng và yêu cầu kiến thức về nông nghiệp, công nghệ thông tin và ứng dụng trong lĩnh vực nông nghiệp. Dữ liệu đầu vào và đầu ra của bài toán nhận diện và phân loại nông sản có thể khá đa dạng, tùy thuộc vào phạm vi và mục đích cụ thể của bài toán. Dưới đây là các dạng thông tin thường được sử dụng:

- Dữ liệu đầu vào:
  - *Hình ảnh:* Đây là dạng dữ liệu quan trọng trong bài toán nhận diện và phân loại nông sản. Hình ảnh của nông sản được sử dụng để trích xuất các đặc trưng và đưa vào quá trình phân loại. Dữ liệu hình ảnh có thể là ảnh RGB (được biểu diễn bằng các giá trị màu đỏ, xanh lá cây và xanh lam), ảnh grayscale (được biểu diễn bằng các mức xám từ 0 đến 255), hoặc ảnh đa kênh khác.
  - *Thông tin đặc trưng:* Ngoài hình ảnh, các thông tin đặc trưng khác có thể được sử dụng để phân loại và nhận diện nông sản. Điều này có thể bao gồm thông tin về kích thước, hình dạng, màu sắc, texture, đặc điểm về mùi, vị, độ cứng/tơi của nông sản, và các thông số vật lý khác.
- Dữ liệu đầu ra:
  - *Nhãn lớp:* Đầu ra của bài toán phân loại nông sản là một nhãn lớp chỉ ra loại nông sản tương ứng. Ví dụ: "táo", "xoài", "nho", "cà chua",...
  - *Độ tin cậy (confidence score):* Ngoài nhãn lớp, bài toán có thể cung cấp một độ tin cậy cho kết quả dự đoán. Điều này cho biết mức độ tự tin của mô hình trong việc phân loại nông sản. Độ tin cậy thường được biểu thị dưới dạng xác suất, phần trăm hoặc điểm số.

Dữ liệu đầu vào và đầu ra của bài toán nhận diện và phân loại nông sản có thể được tùy chỉnh để đáp ứng các yêu cầu và mục đích cụ thể của ứng dụng. Quá trình phân loại và nhận diện nông sản có thể áp dụng cho nhiều mục đích khác nhau, bao gồm:

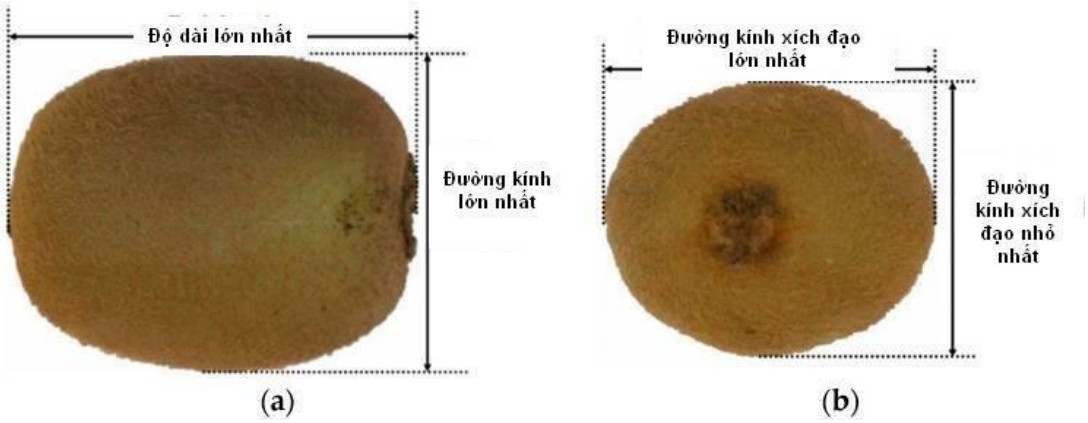
- Đánh giá chất lượng nông sản: Xác định chất lượng, độ chín, độ bị hư hỏng của nông sản để đảm bảo chất lượng và giúp quyết định về việc tiếp thị và phân phối sản phẩm.
- Phân loại tự động: Tự động phân loại các loại nông sản thành các nhóm

khác nhau dựa trên các đặc điểm và thuộc tính của chúng.

- Theo dõi và kiểm soát sản xuất: Giám sát quá trình sản xuất nông sản để đảm bảo tuân thủ các tiêu chuẩn và quy trình sản xuất.
- Nghiên cứu và phát triển: Cung cấp dữ liệu và thông tin để nghiên cứu, phân tích và phát triển các phương pháp mới để nâng cao hiệu suất và chất lượng nông sản.

1.2 Các hướng tiếp cận và giải quyết bài toán

Bài toán tự động nhận dạng nông sản 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 nông sản [2, 3]. 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 nông sản quá phức tạp, kết quả đạt được của các phương pháp này không được cao và phạm vi áp dụng trên số lượng loại nông sản cũng bị hạn chế.



Hình 1.3: 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 bài báo khoa học đề xuất áp dụng phương pháp Học máy - Machine Learning - vào bài toán

nhận dạng nông sản với độ chính xác cao [4], 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 [5-7]. Kết quả thu được tương đối khả quan, khả năng nhận dạng nông sản tự động đã được cải thiện với số lượng loại nông sản được mở rộng và độ chính xác của nhận dạng cao hơn nhiều so với các phương pháp thuần Xử lý ảnh ban đầu. Nối 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 nông sản, trong các thử nghiệm với phạm vi hạn chế về số lượng loại nông sản 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 nông sản 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.

1.2.1 Phương pháp Học máy truyền thống

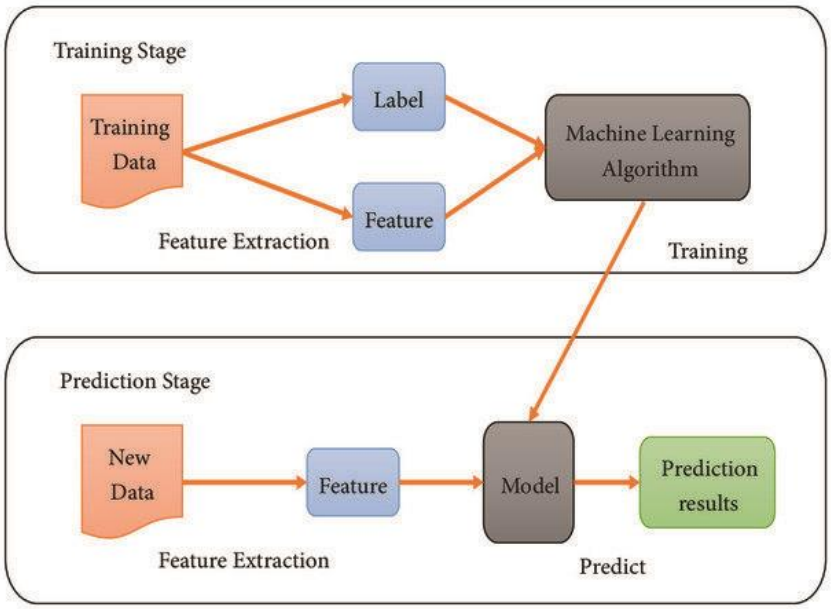
Học máy là một lĩnh vực của trí tuệ nhân tạo, nghiên cứu về việc phát triển các thuật toán và mô hình để giúp máy tính có khả năng học hỏi và cải thiện hiệu suất trong việc giải quyết các vấn đề. Học máy có thể được áp dụng để giải quyết các bài toán trong nhiều lĩnh vực khác nhau như thị giác máy tính, xử lý ngôn ngữ tự nhiên, truy vấn thông tin, điều khiển robot và phân tích dữ liệu.

Trong học máy, các thuật toán và mô hình được phát triển để tìm ra các mẫu và quy luật trong dữ liệu. Điều này được thực hiện bằng cách cung cấp cho máy tính một tập dữ liệu đầu vào và cho phép nó học từ đó. Sau khi được

huấn luyện, mô hình học máy sẽ có khả năng dự đoán kết quả cho các dữ liệu mới dựa trên kiến thức đã học được từ dữ liệu huấn luyện[8].

Các phương pháp học máy truyền thống bao gồm:

- *Học giám sát (Supervised Learning):* Đây là phương pháp học máy phổ biến nhất, trong đó mô hình được huấn luyện với dữ liệu đã được gán nhãn.
- *Học không giám sát (Unsupervised Learning):* Phương pháp này được sử dụng khi không có dữ liệu được gán nhãn.
- *Học bán giám sát (Semi-supervised Learning):* Đây là một phương pháp học máy kết hợp giữa học giám sát và không giám sát. Nó sử dụng các dữ liệu gán nhãn và không gán nhãn để tạo ra một mô hình học máy.
- *Học tăng cường (Reinforcement Learning):* Đây là một phương pháp học máy sử dụng để giải quyết các vấn đề liên quan đến việc tối ưu hóa một hành động để đạt được một mục tiêu nhất định.



Hình 1.4: Mô hình hoạt động chung của các phương pháp Học máy

Trong quá trình học máy (hình 1.4), huấn luyện và thử nghiệm là hai giai đoạn quan trọng để xây dựng một mô hình học máy chất lượng.

Huấn luyện (Training) là quá trình tạo ra một mô hình học máy từ dữ liệu huấn luyện. Trong giai đoạn này, mô hình học máy được huấn luyện thông qua việc tối ưu hóa các tham số của nó. Quá trình này có thể được thực hiện bằng cách sử dụng các thuật toán như Gradient Descent, Stochastic Gradient Descent, hoặc Adam Optimizer.

Sau khi hoàn thành giai đoạn huấn luyện, thử nghiệm (Testing) sẽ được tiến hành để đánh giá hiệu suất của mô hình. Giai đoạn này có thể được thực hiện bằng cách sử dụng tập dữ liệu kiểm tra hoặc tập dữ liệu đánh giá. Quá trình này sẽ đánh giá khả năng dự đoán của mô hình trên dữ liệu mới mà chưa được sử dụng trong quá trình huấn luyện.

Ngoài ra, để đảm bảo hiệu suất của mô hình, có thể sử dụng một số kỹ thuật như cross-validation hoặc holdout validation để chia tập dữ liệu huấn luyện và kiểm tra và đánh giá hiệu suất của mô hình trên các tập dữ liệu khác nhau. Từ kết quả của giai đoạn thử nghiệm, có thể tinh chỉnh và cải tiến mô hình học máy để đạt được hiệu suất tốt hơn.

Đối với mỗi giai đoạn của học máy đều có hai thành phần quan trọng do người xử lý bài toán thiết kế: Trích chọn đặc trưng và Thuật toán phân loại. Cả hai thành phần này đều ảnh hưởng trực tiếp đến kết quả của bài toán. Thiết kế hai thành phần này đòi hỏi người thiết kế phải có kiến thức chuyên môn và hiểu rõ đặc điểm của bài toán. Ngoài ra, việc thiết kế cẩn thận và tốn nhiều thời gian cũng rất quan trọng để đạt được kết quả tốt nhất. Khi thiết kế trích chọn đặc trưng và thuật toán phân loại, người thiết kế phải xem xét các yếu tố như tính hiệu quả, tính khả dụng, độ chính xác, khả năng mở rộng và độ phức tạp tính toán của các thành phần.

1.2.1.1. Trích chọn đặc trưng

Trích chọn đặc trưng (Feature Engineering hoặc Feature Extraction) là quá trình lựa chọn và trích xuất các đặc trưng quan trọng và phù hợp nhất để

mô tả dữ liệu đầu vào trong quá trình học máy. Các đặc trưng này có thể là các thông tin trực tiếp từ dữ liệu như độ dài, chiều rộng, màu sắc, cường độ, hoặc được tạo ra thông qua các kỹ thuật phân tích dữ liệu phức tạp hơn như PCA (Principal Component Analysis) và LDA (Linear Discriminant Analysis).

Việc trích chọn đặc trưng là rất quan trọng trong quá trình học máy vì nó có thể giúp giảm chiều dữ liệu và tăng tính hiệu quả của mô hình học máy. Với dữ liệu có số chiều lớn, việc giảm chiều sẽ giúp giảm thời gian tính toán, tăng độ chính xác của mô hình và giảm khả năng overfitting.

Ngoài ra, việc trích chọn đặc trưng còn giúp cải thiện tính khả dụng của mô hình bởi vì những đặc trưng quan trọng được lựa chọn sẽ giúp mô hình có thể dự đoán kết quả chính xác hơn. Các kỹ thuật trích chọn đặc trưng phổ biến bao gồm PCA, LDA, Feature Scaling, Principal Coordinate Analysis (PCoA), t-SNE (t-Distributed Stochastic Neighbor Embedding), và chiều tương quan (Correspondence Analysis).

Tuy nhiên, việc trích chọn đặc trưng cũng có một số hạn chế. Một trong những hạn chế đó là việc lựa chọn sai đặc trưng có thể dẫn đến kết quả sai hoặc không chính xác trong quá trình học máy. Hơn nữa, việc trích chọn đặc trưng phức tạp có thể dẫn đến tăng thời gian tính toán và khó khăn trong việc hiểu và giải thích kết quả của mô hình học máy.

Do đó, việc thiết kế một quá trình trích chọn đặc trưng tốt là rất quan trọng để đạt được kết quả tốt nhất trong quá trình học máy.

1.2.1.2. Thuật toán

Thuật toán phân loại là một trong những thuật toán cơ bản của học máy, nó được sử dụng để phân loại dữ liệu vào các nhóm khác nhau dựa trên các đặc trưng của chúng.

Các thuật toán phân loại có thể được chia thành hai loại chính:

- Supervised learning algorithms: Đây là các thuật toán phân loại

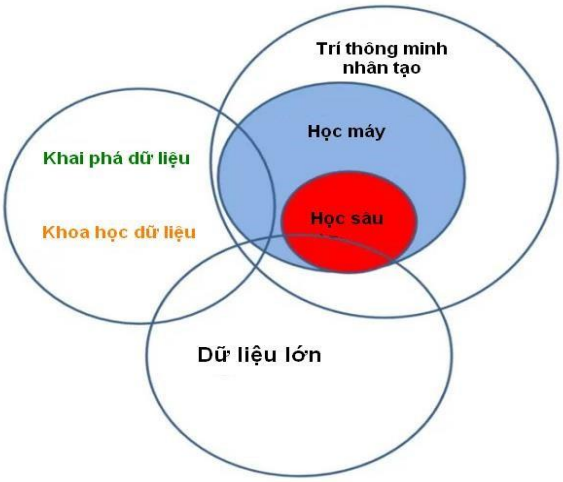
được sử dụng để huấn luyện mô hình từ các tập dữ liệu có nhãn. Các tập dữ liệu này bao gồm các đặc trưng và nhãn tương ứng của chúng. Các thuật toán phân loại được sử dụng trong supervised learning bao gồm: Logistic Regression, Decision Tree, Random Forest, Naive Bayes, Support Vector Machine (SVM), Neural Networks, K-Nearest Neighbor (KNN) và Gradient Boosting.

- Unsupervised learning algorithms: Đây là các thuật toán phân loại được sử dụng để phân nhóm dữ liệu mà không có thông tin về nhãn trước đó.

Các thuật toán phân loại được sử dụng trong unsupervised learning bao gồm: K-means, Hierarchical Clustering và Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Các thuật toán phân loại được sử dụng trong học máy có thể được lựa chọn dựa trên loại dữ liệu và mục tiêu của bài toán. Ví dụ, các thuật toán như Naive Bayes và SVM thường được sử dụng để phân loại văn bản trong các bài toán xử lý ngôn ngữ tự nhiên, trong khi các thuật toán như K-means và Hierarchical Clustering thường được sử dụng để phân nhóm các điểm dữ liệu trong không gian đa chiều.

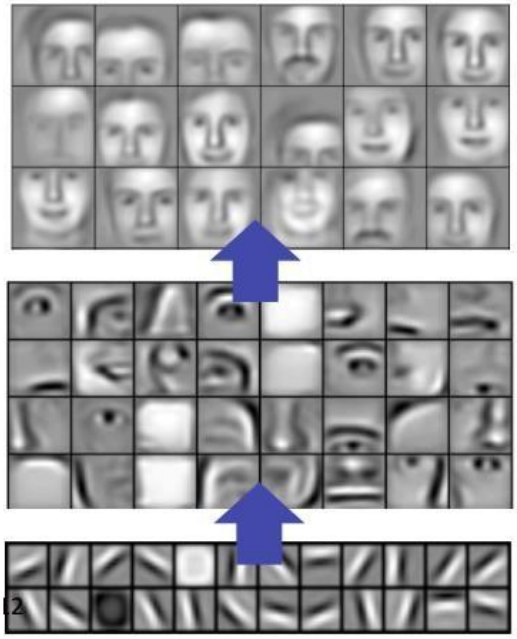
1.2.2 Phương pháp Học sâu

Học sâu (deep learning) là một lĩnh vực của trí tuệ nhân tạo (AI) liên quan đến việc sử dụng một mạng lưới nơ-ron nhân tạo (artificial neural network) để học và trích xuất các đặc trưng từ dữ liệu đầu vào. Phương pháp học sâu đã đạt được nhiều thành công đáng kể trong các lĩnh vực như xử lý ngôn ngữ tự nhiên, thị giác máy tính và nhận dạng giọng nói. Mối quan hệ giữa Học sâu với Học máy cũng như các lĩnh vực liên quan khác qua hình ảnh mô tả bên dưới (Hình 1.5) [10]:



Hình 1.5: Mối quan hệ của Học sâu với các lĩnh vực liên quan

Các phương pháp học sâu bao gồm nhiều lớp nơ-ron được kết nối với nhau để tạo thành một mạng lưới nơ-ron sâu(Deep Neural Network). Mỗi lớp nơ-ron đóng vai trò trích xuất các đặc trưng từ dữ liệu đầu vào, và các lớp này được kết hợp với nhau để tạo ra một mô hình học sâu có khả năng tự động học và cải thiện (xem Hình 1.6).



Hình 1.6: Mức độ trừu tượng tăng dần qua các tầng học của Học sâu [11]

Các phương pháp học sâu thường được huấn luyện thông qua một quá trình tối ưu hóa tham số, ví dụ như sử dụng thuật toán lan truyền ngược

(backpropagation) để điều chỉnh trọng số của các liên kết giữa các nơ-ron trong mạng lưới. Điều này giúp tăng độ chính xác và hiệu suất của mô hình.

Một số kiến trúc mạng lưới nơ-ron phổ biến trong học sâu bao gồm: mạng nơ-ron tích chập (convolutional neural network), mạng nơ-ron hồi quy (recurrent neural network), mạng nơ-ron tự học (autoencoder) và mạng nơ-ron đối nghịch (generative adversarial network).

Học sâu đòi hỏi khối lượng dữ liệu lớn và tính toán mạnh mẽ để huấn luyện mô hình hiệu quả. Vì vậy, nó được sử dụng phổ biến trong các ứng dụng có quy mô lớn như nhận dạng hình ảnh và xử lý ngôn ngữ tự nhiên.

Mặc dù học sâu là một công nghệ tuyệt vời và đang ngày càng được áp dụng rộng rãi trong nhiều lĩnh vực khác nhau, nhưng vẫn tồn tại một số hạn chế, bao gồm:

- Số lượng dữ liệu lớn: Mô hình học sâu yêu cầu một lượng dữ liệu lớn để đào tạo. Nếu không đủ dữ liệu, mô hình có thể bị quá khớp hoặc không đủ khả năng chính xác.
- Khó hiểu và giải thích: Mô hình học sâu có thể đạt được hiệu quả tốt, nhưng cách thức hoạt động của nó thường khó hiểu và khó giải thích, đặc biệt là đối với các mô hình phức tạp.
- Cấu hình phần cứng: Để đào tạo các mô hình học sâu phức tạp, cần sử dụng phần cứng mạnh như GPU hoặc TPU, điều này có thể gây ra chi phí cao và khó tiếp cận đối với các nhà nghiên cứu và doanh nghiệp nhỏ.
- Overfitting: Học sâu có thể bị overfitting nếu mô hình quá phức tạp và không có đủ dữ liệu đào tạo. Overfitting là khi mô hình chỉ học cách phân loại các điểm dữ liệu trong tập đào tạo mà không thể tổng quát hóa cho dữ liệu mới.

- Không thể xử lý dữ liệu bị thiếu sót: Mô hình học sâu không thể xử lý dữ liệu bị thiếu sót hoặc dữ liệu không chính xác. Việc chuẩn bị dữ liệu tốn nhiều thời gian và công sức để đảm bảo tính chính xác của dữ liệu.
- Khả năng chống lại tấn công: Mô hình học sâu có thể bị tấn công bằng các kỹ thuật tấn công mới như là tấn công phá hoại, chèn dữ liệu, lừa đảo...
- Vấn đề đạo đức: Học sâu có thể được sử dụng để sinh ra những sản phẩm hoặc dịch vụ mà có thể gây hại cho con người, ví dụ như tự động sinh ra thông tin giả, xâm phạm quyền riêng tư, tạo ra vũ khí tự động...

Trong ví dụ Hình 1.7, ta có thể nhận thấy sự vô lý trong bức ảnh về quả tạ hai đầu mà mạng Học sâu tạo ra sau khi được huấn luyện với hàng loạt ảnh mẫu. Bức ảnh có chứa các phần ảnh về cánh tay con người, là thành phần không phải thuộc về quả tạ. Việc hình ảnh cánh tay xuất hiện trong phần lớn các ảnh mẫu đã dẫn đến sự nhầm lẫn của mô hình dự đoán này.

Hình 1.7: Bức ảnh quả tạ hai đầu sinh ra bởi mô hình dự đoán Học sâu

Đối với bài toán nhận diện và phân loại hình ảnh, học sâu đã đạt được những thành tựu đáng kể và trở thành một trong những phương pháp phổ biến nhất. Tuy nhiên, nó cũng có những ưu nhược điểm riêng so với các mô hình nhận diện và phân loại hình ảnh khác. Dưới đây là sự so sánh ưu nhược điểm của học sâu với các mô hình khác:

Ưu điểm của học sâu:

- Hiệu suất cao: Học sâu thường đạt được kết quả rất tốt trong việc nhận diện và phân loại hình ảnh, đặc biệt là trong các nhiệm vụ phức tạp và lượng dữ liệu lớn.
- Tự học đặc trưng: Với học sâu, mô hình có khả năng học và trích xuất các đặc trưng tự động từ dữ liệu mà không cần phải định nghĩa các đặc trưng thủ công. Điều này giúp giảm công sức và chi phí của việc chuẩn bị dữ liệu.
- Mở rộng và tái sử dụng: Các mô hình học sâu có thể được mở rộng và tái sử dụng để giải quyết nhiều bài toán khác nhau trong lĩnh vực nhận diện và phân loại hình ảnh.

Nhược điểm của học sâu:

- Yêu cầu lượng dữ liệu lớn: Mô hình học sâu thường cần một lượng dữ liệu huấn luyện lớn để đạt được hiệu suất tốt. Nếu dữ liệu huấn luyện có kích thước nhỏ, mô hình có thể bị overfitting và không hoạt động tốt trên dữ liệu mới.
- Yêu cầu tính toán cao: Để huấn luyện và sử dụng mô hình học sâu, cần có nguồn tài nguyên tính toán mạnh mẽ, đặc biệt là khi mô hình có kích thước lớn hoặc được áp dụng trong thời gian thực.
- Dễ bị ảnh hưởng bởi nhiễu: Mô hình học sâu có thể bị ảnh hưởng bởi nhiễu trong dữ liệu huấn luyện hoặc không chính xác nếu dữ liệu không được chẩn đoán hoặc tiền xử lý đúng cách.

1.3 Thành tựu của phương pháp Học sâu trong các lĩnh vực

Học sâu (deep learning) đã đạt được nhiều thành tựu quan trọng trong các lĩnh vực khác nhau, bao gồm:

Phân tích ngữ nghĩa văn bản

Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP) là một

lĩnh vực của khoa học máy tính và trí tuệ nhân tạo liên quan đến việc giúp máy tính hiểu và xử lý ngôn ngữ tự nhiên của con người. Ngôn ngữ tự nhiên là các ngôn ngữ được sử dụng bởi con người để giao tiếp với nhau, ví dụ như tiếng Anh, tiếng Việt, tiếng Pháp,... Học sâu đã đạt được nhiều thành tựu trong việc xử lý ngôn ngữ tự nhiên (NLP), bao gồm:

- Mô hình ngôn ngữ: Học sâu đã giúp phát triển các mô hình ngôn ngữ như BERT, GPT-3, XLNet, Transformer,... các mô hình này đã giúp nâng cao đáng kể khả năng hiểu và phản hồi lại ngôn ngữ tự nhiên của máy tính.
- Dịch máy: Học sâu đã cải thiện đáng kể khả năng dịch máy, đặc biệt là trong các bài toán dịch máy tự động sử dụng mô hình Neural Machine Translation (NMT). Các mô hình dịch máy này có thể tự động học được từ các bản dịch trước đó để cải thiện khả năng dịch của chúng.
- Phân tích cảm xúc: Học sâu đã giúp cải thiện khả năng phân tích cảm xúc từ văn bản. Ví dụ, các mô hình như LSTM (Long Short-Term Memory) có thể phân tích các đoạn văn bản để xác định cảm xúc của người viết.
- Phân loại văn bản: Học sâu cũng đã được sử dụng để phát triển các mô hình phân loại văn bản, như phân loại thể loại của một cuốn sách hay phân loại các tin tức theo chủ đề.
- Tạo văn bản: Học sâu cũng đã được sử dụng để tạo ra văn bản tự động, bao gồm cả các mô hình sinh văn bản (text generation) và tóm tắt văn bản (text summarization).

Y học

Học sâu đã giúp cải thiện nhiều khía cạnh trong lĩnh vực y học, đồng thời cung cấp nhiều tiềm năng cho việc nghiên cứu và phát triển các phương pháp chẩn đoán và điều trị bệnh. Học sâu đang được sử dụng trong nhiều ứng dụng

trong lĩnh vực y học, bao gồm:

- Điều trị ung thư: Học sâu đã được sử dụng để phát hiện và phân loại ung thư từ các hình ảnh y khoa như CT scan và MRI. Nó cũng đã được sử dụng để phát hiện các dấu hiệu của ung thư qua việc phân tích các hình ảnh và video y khoa.
- Dự đoán bệnh: Học sâu đã được sử dụng để phân tích dữ liệu lâm sàng và dự đoán các bệnh, bao gồm bệnh tim mạch, tiểu đường và bệnh Alzheimer.
- Sinh học phân tử: Học sâu đã được sử dụng để phân tích dữ liệu gen và dữ liệu protein để tìm kiếm các tương tác giữa chúng, đồng thời cũng được sử dụng để phân loại và dự đoán các hợp chất dược phẩm tiềm năng.
- Phát hiện dị tật thai nhi: Học sâu đã được sử dụng để phát hiện dị tật thai nhi qua việc phân tích các hình ảnh siêu âm và cung cấp thông tin hữu ích cho các bác sĩ để đưa ra quyết định chẩn đoán và điều trị.
- Dịch tự động cho y học: Học sâu đã được sử dụng để phát triển các hệ thống dịch tự động cho y học, giúp các bác sĩ và nhà nghiên cứu có thể truy cập thông tin y khoa từ các nguồn trực tuyến và các tài liệu viết bằng các ngôn ngữ khác nhau.

Tài chính

Trong lĩnh vực tài chính, học sâu có thể được sử dụng để giải quyết nhiều vấn đề khác nhau, bao gồm:

- Dự báo giá chứng khoán: Học sâu có thể được sử dụng để dự báo giá chứng khoán và xu hướng thị trường. Nó có thể giúp các nhà đầu tư đưa ra các quyết định đầu tư thông minh và cải thiện hiệu suất đầu tư.
- Phát hiện gian lận tài chính: Học sâu có thể được sử dụng để phát hiện gian lận tài chính bằng cách phân tích các giao dịch và giao dịch không

hợp lệ trong các ngân hàng và tổ chức tài chính.

- Quản lý rủi ro tài chính: Học sâu có thể được sử dụng để phân tích và đánh giá rủi ro tài chính trong các hoạt động kinh doanh. Nó có thể giúp các tổ chức tài chính quản lý rủi ro và giảm thiểu các rủi ro tiềm ẩn.
- Dự báo khách hàng: Học sâu có thể được sử dụng để phân tích dữ liệu khách hàng và dự báo hành vi khách hàng. Nó có thể giúp các tổ chức tài chính đưa ra các chiến lược tiếp thị và quảng cáo hiệu quả hơn.
- Xử lý tài chính tự động: Học sâu có thể được sử dụng để tự động hóa các quy trình tài chính như quản lý danh mục đầu tư và xử lý thanh toán. Nó có thể giúp các tổ chức tài chính tối ưu hóa hiệu suất và giảm thiểu sai sót.

Tự động lái xe

Nhờ vào khả năng xử lý và phân tích dữ liệu phức tạp, học sâu đã giúp cho các hệ thống tự động lái xe ngày càng hoàn thiện và an toàn hơn. Dưới đây là một số thành tựu của học sâu trong lĩnh vực này:

- Nhận diện và phân loại đối tượng: Học sâu đã giúp cho các hệ thống tự động lái xe có thể nhận diện và phân loại đối tượng trên đường như xe hơi, người đi bộ, đèn giao thông, v.v. Từ đó, hệ thống có thể đưa ra quyết định lái xe an toàn hơn.
- Tự động lái xe trên đường cao tốc: Các công nghệ tự động lái xe dựa trên học sâu đã được triển khai trên đường cao tốc. Nhờ vào các cảm biến và mô hình học sâu, hệ thống có thể tự động điều khiển tốc độ, giữ khoảng cách an toàn với các phương tiện khác và duy trì làn đường.
- Hệ thống phát hiện va chạm: Học sâu đã được sử dụng để xây dựng các hệ thống phát hiện va chạm và tránh va chạm trong các tình huống nguy hiểm. Các mô hình học sâu có thể phân tích dữ liệu từ các cảm biến và đưa ra quyết định lái xe phù hợp.

- Tự động lái xe trong thành phố: Học sâu cũng đang được sử dụng để phát triển các hệ thống tự động lái xe trong thành phố. Tuy nhiên, đây là một thách thức lớn do phải đối mặt với nhiều yếu tố khác nhau như đông đúc, đường phố hẹp, đèn giao thông phức tạp, v.v.
- Hệ thống dự báo tình huống giao thông: Học sâu cũng có thể được sử dụng để phát triển các hệ thống dự báo tình huống giao thông trên đường. Các mô hình học sâu có thể phân tích dữ liệu giao thông và đưa ra dự báo về các tình huống khó khăn trên đường, giúp cho hệ thống tự động lái xe có thể đưa ra quyết định phù hợp.

Thị giác máy tính

Dưới đây là một số ứng dụng của học sâu trong thị giác máy tính:

- Nhận diện khuôn mặt: Học sâu đã được sử dụng để phát triển các hệ thống nhận diện khuôn mặt, bao gồm nhận diện và phân loại khuôn mặt trong các bức ảnh hoặc video.
- Nhận diện vật thể: Học sâu được sử dụng để phát triển các hệ thống nhận diện vật thể, bao gồm phân loại các vật thể trong ảnh hoặc video, cũng như xác định vị trí và đường viền của chúng.
- Phát hiện và theo dõi đối tượng: Học sâu được sử dụng để phát triển các hệ thống phát hiện và theo dõi đối tượng trong các hình ảnh hoặc video, cho phép máy tính tự động xác định vị trí và di chuyển của các đối tượng.
- Tăng cường ảnh: Học sâu được sử dụng để phát triển các hệ thống tăng cường ảnh, giúp cải thiện chất lượng và độ phân giải của các hình ảnh.
- Tạo ảnh và video mới: Học sâu được sử dụng để phát triển các hệ thống tạo ra ảnh và video mới, bao gồm tạo ra các ảnh ghép và ảnh động.

Vì những lí do trên, học sâu là phương pháp cần thiết để giải quyết bài toán nhận diện và phân loại nông sản của luận văn và mục tiêu nghiên cứu của chương 2.

1.4 Kết luận chương

Như đã trình bày trong phần mở đầu, mục đích của luận văn là tìm hiểu và ứng dụng một mô hình Học sâu vào bài toán nhận dạng, phân loại nông sản, 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 nông sản. Để chứng minh cho nhận định này, luận văn đã thực hiện phép so sánh độ chính xác của hai mô hình nhận dạng, được huấn luyện lần lượt bởi hai phương pháp trên với cùng bộ dữ liệu đầu vào. Kết quả cụ thể sẽ được trình bày trong Chương 3 – Kết quả thực nghiệm và Đánh giá.

CHƯƠNG 2. PHƯƠNG PHÁP NHẬN DIỆN, PHÂN LOẠI NÔNG SẢN

2.1 Mô hình mạng nơron tích chập

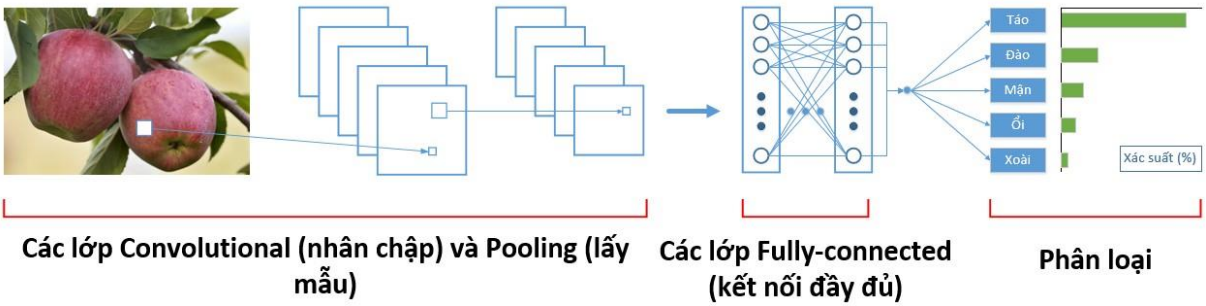
Mạng nơ ron tích chập (Convolutional Neural Network - CNN) là một kiểu mạng nơ ron sử dụng cho việc xử lý ảnh và âm thanh với độ chính xác rất cao, thậm chí còn tốt hơn con người trong nhiều trường hợp. Mô hình này đã và đang được phát triển, ứng dụng vào các hệ thống xử lý ảnh lớn của Facebook, Google 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.

Mạng nơ ron tích chập (CNN) được phát triển từ những năm 1980 và 1990, khi các nhà khoa học như Yann LeCun, Yoshua Bengio và Geoffrey Hinton đã đưa ra các ý tưởng đầu tiên về kiến trúc mạng nơ ron tích chập. Tuy nhiên, vào thời điểm đó, công nghệ chưa đủ tiên tiến để triển khai mô hình này với hiệu suất cao. Sau đó, vào những năm 2010, việc phát triển công nghệ GPU (Graphics Processing Unit) đã giúp cho việc huấn luyện mạng CNN trở nên nhanh hơn đáng kể. Bên cạnh đó, sự ra đời của các bộ dữ liệu lớn như ImageNet cũng đã giúp cho mạng CNN trở nên phổ biến hơn.

Mạng CNN đã đạt được nhiều thành công trong các lĩnh vực như nhận dạng ảnh, nhận dạng giọng nói, xử lý ngôn ngữ tự nhiên và trở thành một trong những mô hình nổi tiếng nhất trong lĩnh vực trí tuệ nhân tạo. Hiện nay, các kiến trúc CNN như VGG, ResNet, Inception và EfficientNet được sử dụng rộng rãi trong các ứng dụng thực tế.

Các lớp cơ bản trong một mạng CNN bao gồm: Lớp tích chập (Convolutional), Lớp kích hoạt phi tuyến ReLU (Rectified Linear Unit), Lớp lấy mẫu (Pooling) và Lớp kết nối đầy đủ (Fully-connected). Trong một số trường hợp, các lớp này có thể được xếp chồng lên nhau để tạo thành một kiến

trúc mạng phức tạp hơn. Ví dụ, một mô hình CNN thông thường có thể bao gồm nhiều lớp tích chập, lớp kích hoạt và lớp tổng hợp, trước khi kết thúc bằng các lớp kết nối đầy đủ và đầu ra.



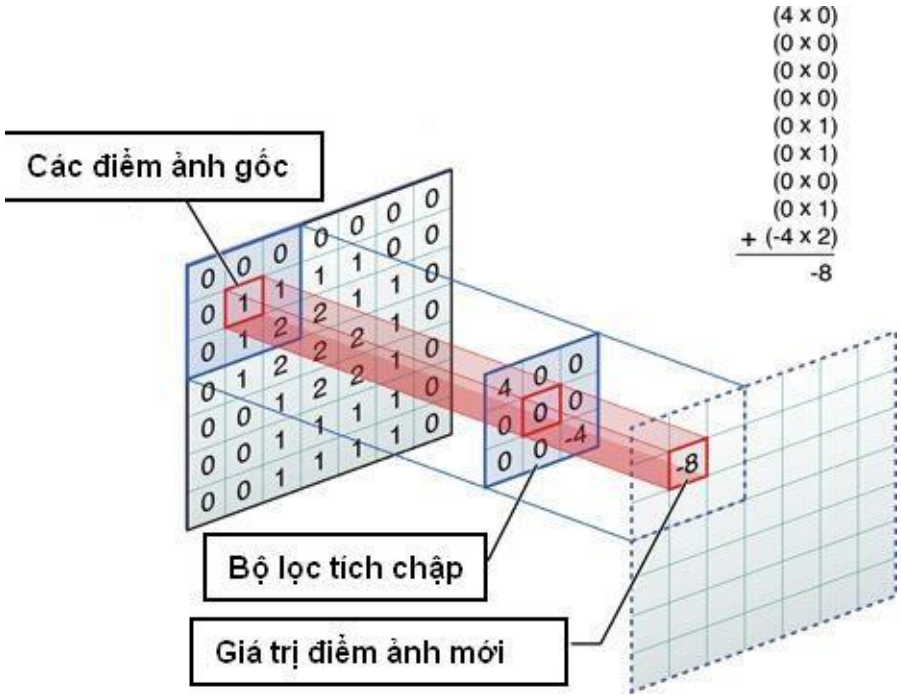
Hình 2.1: Kiến trúc cơ bản của một mạng tích chập

- Lớp tích chập:

Lớp tích chập (convolutional layer) là một lớp quan trọng trong kiến trúc của mạng nơ-ron tích chập (CNN). Lớp tích chập giúp mạng CNN trích xuất các đặc trưng từ dữ liệu đầu vào bằng cách sử dụng các bộ lọc (filters) để quét (convolve) qua các vùng của dữ liệu.

Mỗi bộ lọc là một ma trận số được áp dụng lên các vùng của dữ liệu đầu vào. Quá trình này tạo ra một bản đồ đặc trưng (feature map) mới bằng cách tính toán tích vô hướng giữa các giá trị trong bộ lọc và các giá trị trong vùng của dữ liệu đầu vào được quét. Điều này giúp giảm kích thước của dữ liệu và tăng cường tính đặc trưng của các đặc trưng được trích xuất.

Ngoài ra, các tham số như độ sâu (depth), bước nhảy (stride), và độ lề (padding) cũng ảnh hưởng đến cách mà lớp tích chập hoạt động và cách mà đầu ra được tính toán. Tuy nhiên, các giá trị này thường được thiết lập mặc định cho một số kiến trúc mạng CNN thông dụng như VGG, ResNet và Inception.



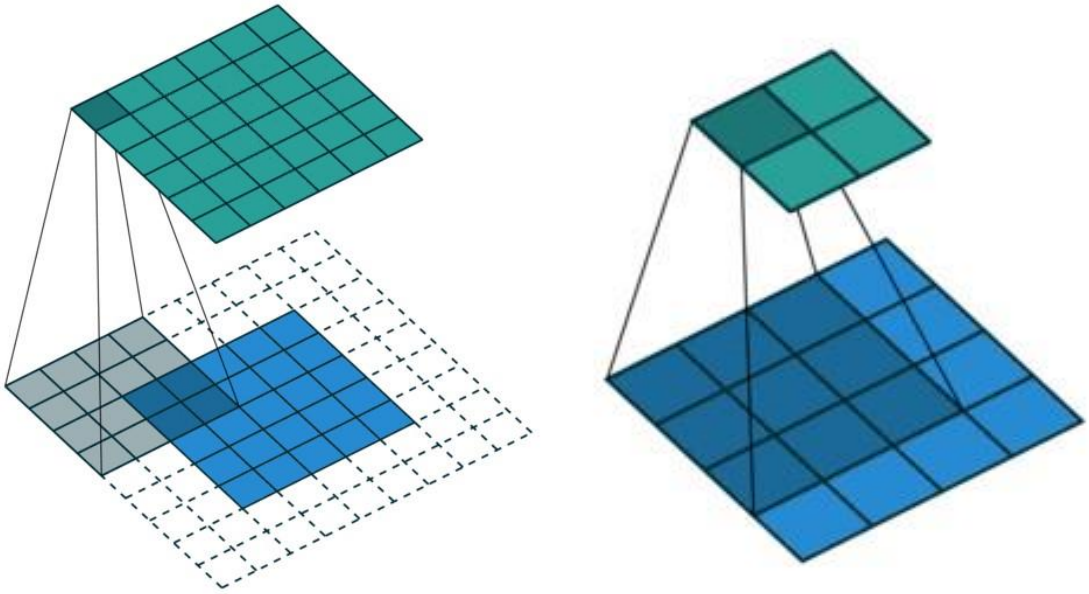
Hình 2.2: Ví dụ bộ lọc tích chập được sử dụng trên ma trận điểm ảnh

Trong ví dụ ở Hình 2.2 [12], 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. 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 (1) [13]:

$$o = \frac{i+2*p-k}{s} + 1 \quad (1)$$

Trong đó:

- o: kích thước ảnh đầu ra
- i: kích thước ảnh đầu vào
- p: kích thước khoảng trắng phía ngoài viền của ảnh gốc
- k: kích thước bộ lọc
- s: bước trượt của bộ lọc



Hình 2.3: Trường hợp thêm/không thêm viền trắng vào ảnh khi tích chập

Như vậy, sau khi đưa một bức ảnh đầu vào cho lớp Tích chập ta nhận được kết quả đầu ra là một loạt ảnh tương ứng với các bộ lọc đã được sử dụng để thực hiện phép tích chập. Các trọng số của các bộ lọc này được khởi tạo ngẫu nhiên trong lần đầu tiên và sẽ được cải thiện dần xuyên suốt quá trình huấn luyện.

- **Lớp kích hoạt phi tuyến ReLU**:

Lớp kích hoạt phi tuyến ReLU (Rectified Linear Unit) là một lớp quan trọng trong các mạng nơ-ron, đặc biệt là trong mạng nơ-ron tích chập (CNN). ReLU được sử dụng để kích hoạt các nơ-ron trong các lớp tích chập và kết nối đầy đủ của mạng. Hàm kích hoạt ReLU được định nghĩa bằng cách đặt đầu ra của các nơ-ron có giá trị nhỏ hơn 0 bằng 0, và giữ nguyên giá trị đầu ra của các nơ-ron có giá trị lớn hơn hoặc bằng 0. Cụ thể, hàm ReLU được định nghĩa như sau:

$$f(x) = max(0, x) \quad (2)$$

Trong đó, x là giá trị đầu vào của một nơ-ron, và f(x) là giá trị đầu ra của hàm ReLU. Hàm ReLU là một hàm phi tuyến tính và khá đơn giản để tính

toán.

Các ưu điểm của hàm ReLU là nó cho phép mạng học được nhanh hơn bằng cách tránh hiện tượng gradient biến mất (vanishing gradient) trong quá trình lan truyền ngược. Ngoài ra, ReLU cũng giúp giảm thiểu các vấn đề liên quan đến overfitting và tăng tốc quá trình hội tụ của mô hình.

Tuy nhiên, hàm ReLU cũng có một số nhược điểm, chẳng hạn như vấn đề của các nơ-ron chết (dead neurons), khi các nơ-ron có giá trị đầu vào nhỏ hơn 0 sẽ không được kích hoạt và không thể học được gì thêm. Để khắc phục vấn đề này, các biến thể của hàm ReLU, chẳng hạn như Leaky ReLU và PReLU, đã được đưa ra.

- **Lớp lấy mẫu**:

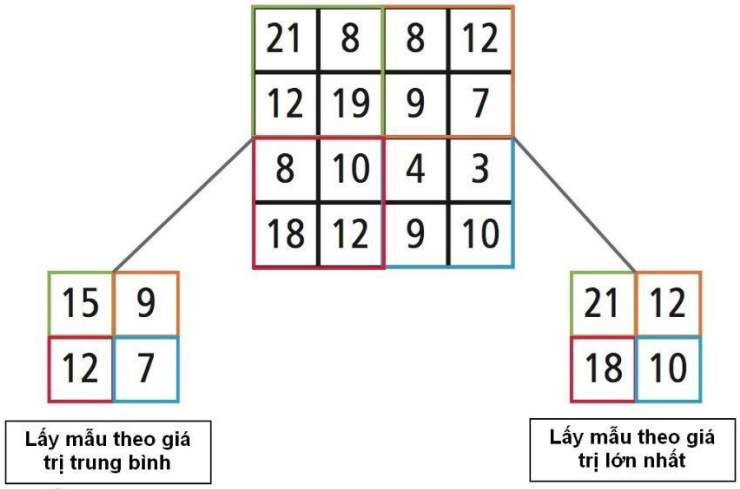
Lớp lấy mẫu (pooling layer) là một lớp quan trọng trong mạng nơ-ron tích chập (CNN). Lớp này thường được sử dụng để giảm kích thước của bản đồ đặc trưng (feature map) và giảm thiểu overfitting. Các loại lớp lấy mẫu thường được sử dụng trong mạng CNN bao gồm:

- Lớp lấy mẫu tối đa (max pooling layer): lớp này sẽ chọn giá trị lớn nhất trong một vùng của bản đồ đặc trưng và giữ lại giá trị đó, bỏ qua các giá trị khác trong vùng đó. Ví dụ, nếu kích thước của vùng là 2x2, lớp lấy mẫu tối đa sẽ chọn giá trị lớn nhất trong mỗi vùng 2x2 và tạo ra một bản đồ đặc trưng mới có kích thước giảm đi một nửa.
- Lớp lấy mẫu trung bình (average pooling layer): lớp này tính trung bình cộng các giá trị trong một vùng của bản đồ đặc trưng và giữ lại giá trị đó. Tương tự như lớp lấy mẫu tối đa, lớp lấy mẫu trung bình cũng giảm kích thước của bản đồ đặc trưng.
- Lớp lấy mẫu thống kê (statistical pooling layer): lớp này tính toán các giá trị thống kê (ví dụ như phương sai) của các giá trị trong một vùng

của bản đồ đặc trưng và giữ lại giá trị đó.

Lớp lấy mẫu thường được sử dụng sau lớp tích chập để giảm kích thước của bản đồ đặc trưng và giảm số lượng tham số của mô hình. Tuy nhiên, việc sử dụng lớp lấy mẫu cũng có thể gây mất thông tin, do đó cần cân nhắc để đạt được sự cân bằng giữa việc giảm kích thước và giữ lại thông tin quan trọng.

Hình 2.4 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 Avarage Pooling (lấy giá trị trung bình của các điểm ảnh trong vùng ảnh cục bộ) [14].



Hình 2.4: Phương thức Avarage Pooling và Max 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.

- 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. Lớp này thường được sử dụng để kết nối các bản đồ đặc trưng (feature maps) từ lớp tích chập và lớp lấy mẫu với nhau để tạo thành một đầu ra.

Lớp kết nối đầy đủ sẽ kết nối mỗi neuron trên lớp này với tất cả các

neuron trên lớp trước nó, tạo ra một ma trận trọng số đầy đủ (fully connected weight matrix) và một vector bias cho mỗi neuron. Các trọng số và bias này sẽ được học thông qua quá trình lan truyền ngược (backpropagation) trong quá trình huấn luyện mô hình. Lớp kết nối đầy đủ thường được sử dụng để phân loại ảnh hoặc nhận dạng đối tượng trong mạng CNN. Sau khi các bản đồ đặc trưng đã được trích xuất từ ảnh đầu vào thông qua các lớp tích chập và lớp lấy mẫu, lớp kết nối đầy đủ sẽ đưa ra các dự đoán cho lớp đầu ra.

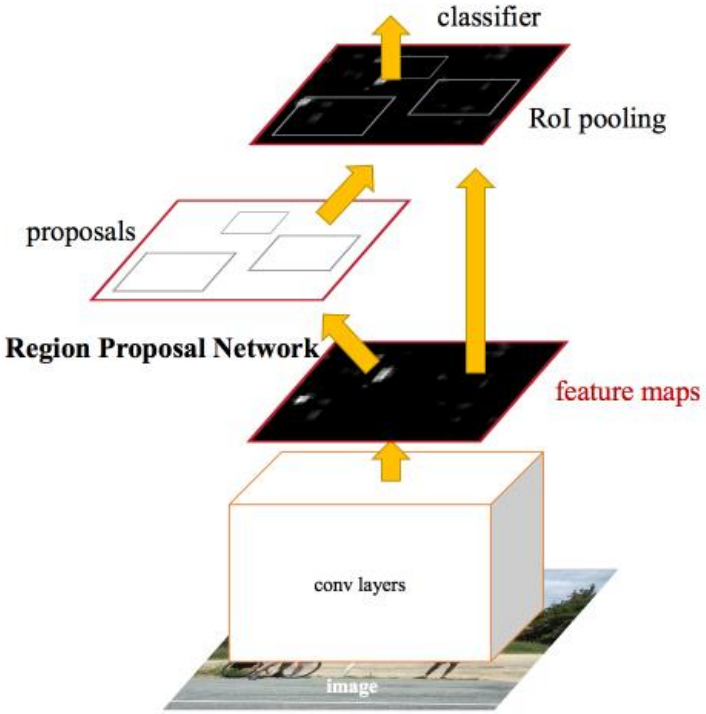
Tuy nhiên, lớp kết nối đầy đủ cũng có một số hạn chế, bao gồm số lượng tham số lớn và dễ gây overfitting. Do đó, trong một số kiến trúc CNN hiện đại, thay vì sử dụng lớp kết nối đầy đủ ở cuối cùng của mô hình bằng việc sử dụng các lớp kết nối thưa (sparse connection layers) hoặc các lớp kết nối đặc trưng (feature pooling layers) để giảm số lượng tham số và cải thiện độ chính xác của mô hình.

2.2 Các mô hình CNN phổ biến

2.2.1. Faster R-CNN (2016)

Faster R-CNN (Faster Region-based Convolutional Neural Network) là một mô hình object detection (phát hiện đối tượng) phổ biến trong deep learning. Nó được giới thiệu bởi Shaoqing Ren, Kaiming He, Ross Girshick và Jian Sun vào năm 2015.

Điểm khác biệt chính của Faster R-CNN so với các mô hình trước đó là việc sử dụng một mạng neural riêng biệt gọi là Region Proposal Network (RPN) để tạo ra các khu vực đề xuất (region proposals) cho việc phát hiện đối tượng. RPN sử dụng một mạng CNN để đưa ra các khu vực có thể chứa đối tượng trong ảnh đầu vào, thay vì việc sử dụng các phương pháp đặt dấu vết (sliding window) và các thuật toán tối ưu hóa để tạo ra các khu vực đề xuất.



Hình 2.5: Kiến trúc mô hình Faster R-CNN [18].

Kiến trúc của Faster R-CNN [18] bao gồm:

- Backbone network (mạng gốc): được sử dụng để rút trích đặc trưng từ hình ảnh đầu vào. Thường là một mạng CNN được huấn luyện trước đó, chẳng hạn như VGG, ResNet, hoặc Inception.
- Region Proposal Network (RPN): một mạng neural riêng biệt được sử dụng để tạo ra các khu vực đề xuất cho việc phát hiện đối tượng. RPN sử dụng các tính năng được trích xuất từ mạng CNN trích xuất đặc trưng để dự đoán các khu vực đề xuất chứa đối tượng.
- Region-based CNN (R-CNN): một mạng CNN đặc biệt được sử dụng để phát hiện và phân loại các đối tượng trong ảnh. Sau khi có được các khu vực đề xuất từ RPN, chúng được đưa vào mạng R-CNN để tạo ra các dự đoán cho từng lớp đối tượng.

2.2.2 Lớp các mô hình họ YOLO

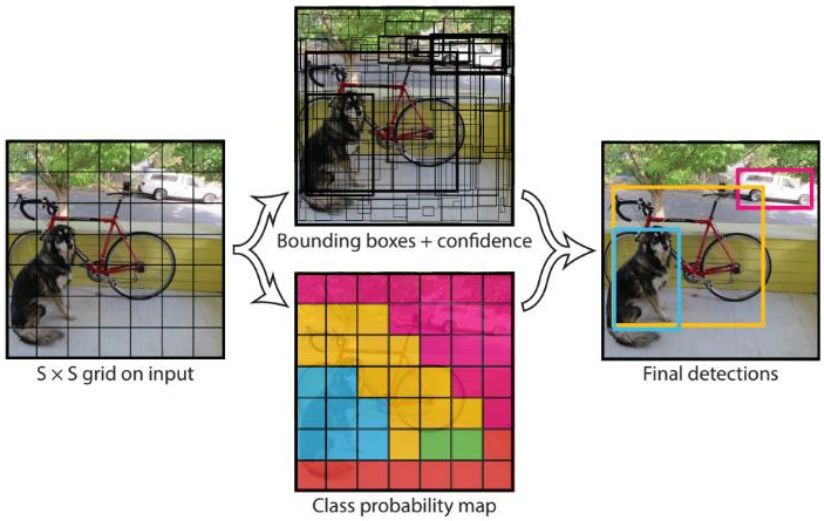
Yolo (You Only Look Once) là một họ các mô hình nhận diện đối tượng

dựa trên deep learning, được phát triển bởi nhóm nghiên cứu của Joseph Redmon. Các mô hình Yolo được sử dụng rộng rãi trong các ứng dụng nhận diện đối tượng, như nhận diện giao thông, nhận diện khuôn mặt, và nhận diện vật thể trong ảnh chụp từ camera an ninh, camera giám sát.

YOLOv1 (2015)

Mô hình YOLO v1 là phiên bản đầu tiên của mô hình YOLO, được giới thiệu vào năm 2015 bởi Joseph Redmon. Đây là một trong những mô hình đầu tiên sử dụng phương pháp "định vị và phân loại đối tượng cùng một lúc" để giải quyết bài toán nhận diện đối tượng.

YOLO v1 hoạt động bằng cách chia ảnh đầu vào thành một lưới (grid) các ô (grid cells), sau đó sử dụng một mạng CNN để tính toán các hộp giới hạn (bounding box) và xác suất lớp cho mỗi ô trong lưới. Kết quả cuối cùng của YOLOv1 là một danh sách các hộp giới hạn cho tất cả các đối tượng được tìm thấy trong ảnh đầu vào, cùng với xác suất cho từng lớp. YOLOv1 có thể hoạt động với tốc độ rất nhanh so với các phương pháp khác nhưng có độ chính xác thấp hơn trên các đối tượng nhỏ và các đối tượng có tỷ lệ khác nhau.



Hình 2.6: Các bước xử lý trong mô hình YOLO [19]

YOLOv2 (2016)

YOLOv2 được phát triển bởi Joseph Redmon và các đồng nghiệp tại đại học Washington và công bố vào năm 2016. YOLOv2 là phiên bản cải tiến của YOLOv1. YOLOv2 có nhiều cải tiến so với YOLOv1, bao gồm sử dụng một backbone mới, batch normalization, kích thước lưới được tinh chỉnh, đào tạo đa tỉ lệ, dự đoán kích thước đối tượng tốt hơn và non-maximum suppression được tinh chỉnh. YOLOv2 cho phép nhận diện các vật thể trên các ảnh có kích thước khác nhau, đồng thời có thể xử lý các vật thể gần nhau hơn và đa dạng hơn. Nó cũng có khả năng xác định vị trí của các vật thể chính xác hơn và giảm thiểu sai sót khi dự đoán các hộp giới hạn.

YOLOv3 (2018) và YOLOv4 (2020)

YOLOv3 được phát hành vào năm 2018 với các tính năng mới như hộp giới hạn (bounding box) có kích thước khác nhau (để xác định các vật thể lớn hơn), kết hợp thông tin từ các lớp trước đó để tăng độ chính xác và sử dụng kỹ thuật Non-maximum suppression (NMS) để loại bỏ các hộp trùng lắp. YOLOv3 cũng sử dụng ResNet, một mạng CNN được phát triển bởi Microsoft, để tăng độ sâu của mô hình và cải thiện độ chính xác.

Trên cơ sở của YOLOv3, YOLOv4 được phát hành vào năm 2020 với nhiều cải tiến mới như: *Mô-đun SPP (Spatial Pyramid Pooling)*, *Mô-đun CSP*, Cải tiến việc kết nối các lớp trong mạng CNN để cải thiện độ chính xác và tốc độ xử lý và các kỹ thuật cải tiến khác như kỹ thuật Mish Activation, Mosaic Data Augmentation, DropBlock regularization,...

2.2.3 SSD Model

SSD (Single Shot MultiBox Detector) là một thuật toán phát hiện đối tượng được sử dụng phổ biến trong lĩnh vực thị giác máy tính. Thuật toán này được giới thiệu bởi nhóm nghiên cứu của Google DeepMind vào năm 2016.

SSD sử dụng một mạng neural sâu để phân tích ảnh đầu vào và tạo ra

các bounding box và xác suất lớp tương ứng cho mỗi đối tượng được phát hiện trong ảnh. Điều này được thực hiện bằng cách sử dụng các lớp tích chập bổ sung được thêm vào trên mạng cơ sở.

Một trong những ưu điểm của SSD so với các thuật toán phát hiện đối tượng khác là việc tạo ra bounding box và xác suất lớp trong một lần chạy (single shot), giúp cho việc phát hiện đối tượng nhanh chóng và tiết kiệm tài nguyên tính toán. Hơn nữa, SSD cũng có khả năng phát hiện đối tượng với các kích thước khác nhau trên một ảnh.

Tuy nhiên, SSD cũng có một số hạn chế, bao gồm khả năng phát hiện đối tượng không chính xác khi chúng có kích thước nhỏ hoặc bị che khuất bởi các đối tượng khác trong ảnh.

Bên dưới là bảng so sánh tốc độ running của các mô hình object detection.

Method	mAP	FPS	batch size	# Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	1	~ 6000	~ 1000 × 600
Fast YOLO	52.7	155	1	98	448 × 448
YOLO (VGG16)	66.4	21	1	98	448 × 448
SSD300	74.3	46	1	8732	300 × 300
SSD512	76.8	19	1	24564	512 × 512
SSD300	74.3	59	8	8732	300 × 300
SSD512	76.8	22	8	24564	512 × 512

Hình 2.7: Bảng so sánh tốc độ xử lý và độ chính xác của các lớp model [20]

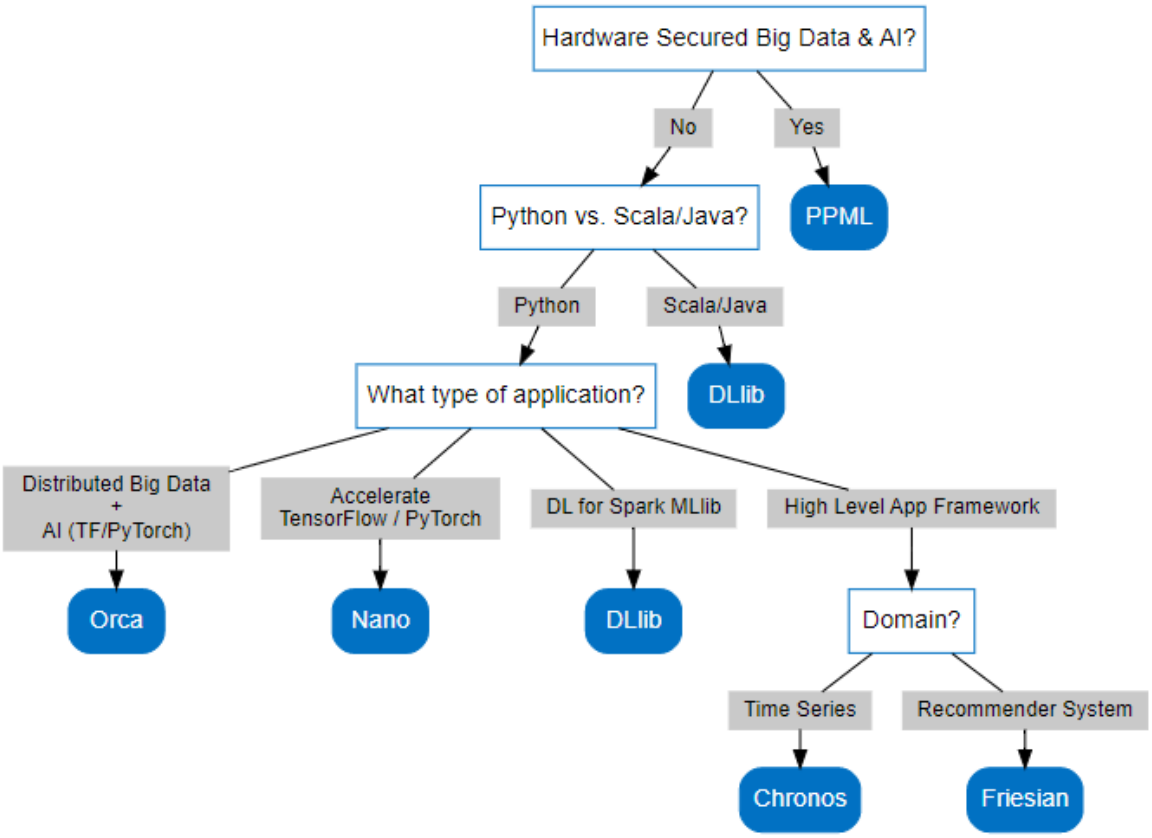
2.3 Mô hình mã nguồn mở BigDL

2.3.1 Tổng quan về BigDL

BigDL là một mô hình mã nguồn mở cho học sâu trên Apache Spark, được phát triển bởi Intel và được phát hành theo giấy phép Apache 2.0. BigDL được thiết kế để tối ưu hóa cho các mô hình học sâu lớn trên các hệ thống phân tán, nhưng vẫn giữ nguyên tính linh hoạt của Spark. Nó cung cấp một loạt các lớp và phương thức xây dựng mô hình học sâu, bao gồm các lớp cho các mô hình mạng nơ-ron thần kinh sâu như CNN, RNN, LSTM, và GRU.. BigDL cho

phép người dùng sử dụng các công cụ và thư viện quen thuộc của Spark để xây dựng và huấn luyện các mô hình Deep Learning trên các dữ liệu lớn. BigDL hỗ trợ nhiều kiến trúc mô hình như Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), và nhiều kiến trúc mạng Deep Learning khác. Các mô hình được huấn luyện trên BigDL có thể được triển khai trên các môi trường phân tán như Apache Spark hay các thiết bị nhúng để triển khai các ứng dụng AI.

- Orca: cung cấp API đơn giản để triển khai và quản lý các ứng dụng AI phân tán trên Apache Spark và Ray.
- Nano: tối ưu hóa cho các ứng dụng TensorFlow và PyTorch để tăng tốc độ huấn luyện và đạt được hiệu quả cao hơn.
- DLlib: cung cấp các công cụ và thư viện để triển khai và quản lý các mô hình Deep Learning trên Apache Spark, tương đương với Spark Mllib cho học sâu.
- Chronos: cung cấp các công cụ thống kê thời gian tính được sử dụng AutoML để giúp người dùng tối ưu hóa thời gian huấn luyện và đạt được hiệu quả cao hơn.
- Friesian: cung cấp các công cụ đầu cuối để xây dựng các ứng dụng AI và học máy trên Apache Spark và Ray.
- PPML (experimental): cung cấp một môi trường an toàn cho xử lý dữ liệu lớn và các ứng dụng trí tuệ nhân tạo, với sự hỗ trợ của SGX Hardware Security.



Hình 2.8: BigDL

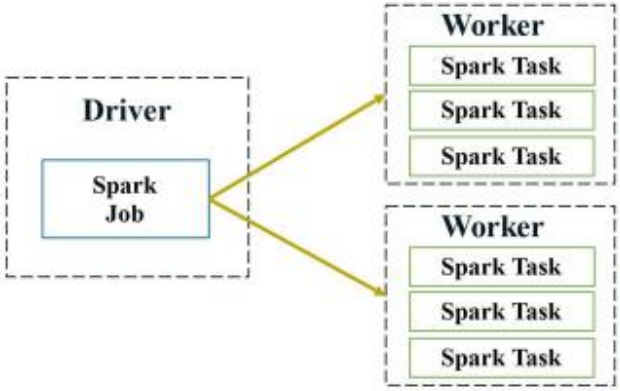
2.3.2 Mô hình thực thi BigDL

Trong khi sử dụng các phương pháp tiêu chuẩn như huấn luyện song song dữ liệu, máy chủ tham số để huấn luyện có khả năng mở rộng, điểm mới của BigDL là cách triển khai hiệu quả các chức năng này trên một mô hình tính toán cấu trúc của Apache Spark.

Trong cộng đồng học máy, truy cập dữ liệu chi tiết và thực hiện sửa đổi dữ liệu tại chỗ được xem là cực kỳ quan trọng để hỗ trợ cho việc huấn luyện phân tán hiệu quả với các máy chủ tham số. Tuy nhiên, trong hệ thống big data như Spark, mô hình tính toán cấu trúc khác được áp dụng, trong đó bộ dữ liệu không thay đổi và chỉ có thể được chuyển đổi thành bộ dữ liệu mới mà không có tác động phụ (tức là sao chép khi cần thiết); ngoài ra, các phép biến đổi đều là các thao tác cấu trúc thô (tức là áp dụng cùng một phép biến đổi cho tất cả

các mục dữ liệu cùng một lúc).

- ***Mô hình tính toán của Spark***



Hình 2.9: Mô hình của Spark: Driver Node có chức năng lập lịch và phân công công việc cho các Worker Node

Mô hình tính toán của Spark dựa trên nguyên tắc chia để trị (divide-and-conquer), trong đó các tác vụ tính toán được chia thành các phần nhỏ hơn và được phân tán trên các worker nodes để tính toán đồng thời. Kết quả tính toán được tổng hợp và trả về cho driver program để xử lý. Quá trình chia tác vụ tính toán thành các phần nhỏ hơn và phân tán chúng được quản lý bởi Spark Core và RDDs. Để tự động phân tán xử lý dữ liệu trên cụm một cách đáng tin cậy, Spark cung cấp một mô hình tính toán hàm số và phân tán dữ liệu. Trong một công việc Spark, dữ liệu được đại diện bằng Resilient Distributed Dataset (RDD), đó là một tập hợp các đối tượng có thể chia sẻ và bất biến được phân tán trên nhiều node trong một cluster, và chỉ có thể được biến đổi để tạo ra các RDD mới (tức là sao chép khi cần) thông qua các toán tử hàm số như map, filter và reduce. Ngoài ra, các phép toán này đều là phân tán dữ liệu (tức là được áp dụng cho các phân vùng dữ liệu riêng lẻ song song bởi các tác vụ Spark khác nhau) và thô sơ (tức là áp dụng cùng một phép toán cho tất cả các mục dữ liệu cùng một lúc).

Một công cụ chính của Spark là Spark Core, đó là một thư viện trung tâm cung cấp các chức năng cơ bản cho việc xử lý dữ liệu, bao gồm các phép

biến đổi, lọc và tính toán. Ngoài ra, Spark còn cung cấp các thư viện và công cụ cho việc xử lý dữ liệu cấu trúc và phi cấu trúc, bao gồm Spark SQL, Spark Streaming, GraphX và MLib.

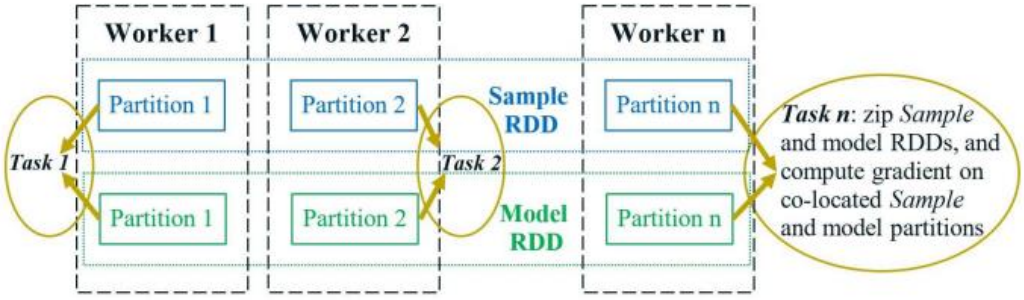
- ***Mô hình tính toán của BigDL***

BigDL được xây dựng trên cơ sở mô hình tính toán song song dữ liệu của Spark, cung cấp việc huấn luyện mô hình mạng neural sử dụng đào tạo dữ liệu song song đồng bộ trên cụm máy tính, đã được chứng minh là đạt được tính mở rộng và hiệu suất tốt hơn (tính bằng thời gian cho đến chất lượng) so với đào tạo không đồng bộ. Để cụ thể hơn, quá trình huấn luyện phân tán trong BigDL được thực hiện dưới dạng một quá trình lặp, như được minh họa dưới đây:

for i = 1 to M **do**
    // Tác vụ “forward-backward”
    for each task in the Spark job **do**
        read the latest **weights**;
        get a random **batch** of data from local Sample partition;
        compute local **gradients** (forward-backward on local model replica);
    end for
    // Tác vụ “đồng bộ hóa tham số”
    aggregate (sum) all the **gradients**;
    update the **weights** per specified optimization method;
end for

Mỗi vòng lặp chạy một vài tác vụ Spark để tính toán gradient sử dụng mini-batch hiện tại (bằng một tác vụ “forward-backward”), và sau đó thực hiện một cập nhật đơn lẻ cho các thông số của mô hình mạng neural (bằng một tác vụ "đồng bộ hóa thông số").

Mô hình tính toán của BigDL sử dụng các RDDs trong Apache Spark để lưu trữ dữ liệu huấn luyện và các tham số của mô hình. BigDL tạo ra một RDD các mô hình, mỗi mô hình là một bản sao của mô hình nơ-ron sâu ban đầu. RDDs này được cache trong bộ nhớ và phân phối trên các máy tính trong cụm., và được phân vùng và đặt cùng một vị trí trên cluster, như được thể hiện trong Hình 2.10. Do đó, trong mỗi vòng lặp của huấn luyện mô hình, một tác vụ "forward-backward" có thể đơn giản áp dụng operator zip function để ghép các phân vùng được đặt cùng vị trí của hai RDD (mà không tốn chi phí phụ), và tính gradient song song cho mỗi bản sao của mô hình (sử dụng một lượng nhỏ dữ liệu trong phân vùng Sample đặt cùng vị trí), như được minh họa trong Hình 2.10.



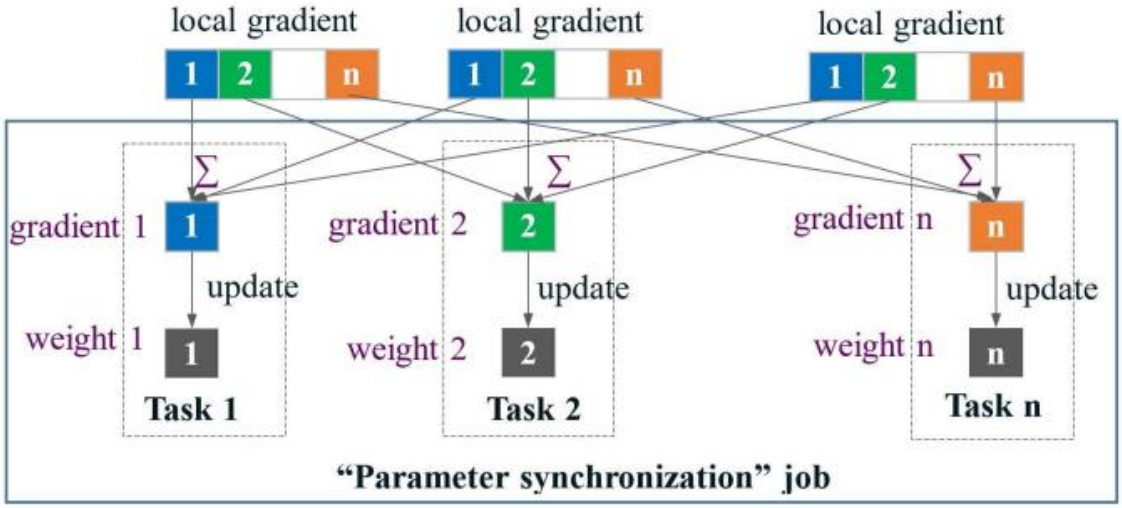
Hình 2.10: Tác vụ "forward-backword" của Spark tính toán gradient cho mỗi bản sao mô hình mạng nơ-ron song song

BigDL không hỗ trợ phân tán mô hình (model parallelism) tức là không có việc phân phối mô hình trên các worker khác nhau. Tuy nhiên, điều này không gây hạn chế trong thực tế, vì BigDL chạy trên các máy chủ Intel Xeon CPU, thường có dung lượng bộ nhớ lớn (100s GB) và có thể dễ dàng chứa các mô hình rất lớn.

- ***Đồng bộ hóa tham số trong BigDL***

Đồng bộ hóa tham số là một phép tính quan trọng đối với huấn luyện mô hình phân tán song song trên dữ liệu (về tốc độ và khả năng mở rộng). Để hỗ trợ đồng bộ hóa tham số hiệu quả, các framework học sâu hiện có thường triển

khai máy chủ tham số hoặc AllReduce bằng cách sử dụng các phép tính như truy cập dữ liệu chi tiết và thay đổi dữ liệu tại chỗ. Thật không may, các phép tính này không được hỗ trợ bởi mô hình tính toán chức năng của hệ thống dữ liệu lớn (như Spark).



Hình 2.11: Đồng bộ hóa tham số trong BigDL

BigDL đã tiếp cận hoàn toàn khác biệt bằng cách trực tiếp triển khai một hoạt động AllReduce hiệu quả sử dụng các nguyên tắc hiện có trong Spark. Sau khi tính toán gradient cho mỗi mô hình con trong tác vụ "model forward-backward", BigDL sử dụng một tác vụ Spark khác, được gọi là tác vụ "Đồng bộ hóa tham số" để cập nhật trọng số cho mô hình chính trên driver, được thể hiện trong hình 2.11.

Cụ thể, BigDL sử dụng thuật toán "AllReduce" để tính trung bình của các gradient của tất cả các mô hình con, sau đó áp dụng trung bình này để cập nhật các tham số trên mô hình chính. Thuật toán "AllReduce" sử dụng các phép tính trên ma trận và được triển khai bằng các hàm gọi tới hệ thống tính toán phân tán của Spark. Quá trình này được lặp lại cho tất cả các vòng lặp của quá trình huấn luyện

for each task n in the "parameter synchronization" job **do**

shuffle the n^{th} partition of all gradients to this task;

aggregate (sum) these **gradients**;

updates the n^{th} partition of the **weights**;

broadcast the n^{th} partition of the updated **weights**;

end for

BigDL đã tuân theo thực tiễn tiêu chuẩn (như huấn luyện song song dữ liệu và các phép toán AllReduce) để huấn luyện có khả năng mở rộng, tuy nhiên cài đặt của nó rất khác so với các framework deep learning hiện có. Bằng cách áp dụng trạng thái của thực tiễn hệ thống big data, BigDL cung cấp một giải pháp thiết kế khả thi khác cho huấn luyện mô hình phân phối. Điều này cho phép các thuật toán học sâu và phân tích dữ liệu lớn được tích hợp một cách liền mạch vào một luồng dữ liệu duy nhất, và hoàn toàn loại bỏ vấn đề không phù hợp về truyền thông. Hơn nữa, điều này cũng làm cho việc xử lý sự cố, thay đổi tài nguyên, ưu tiên tác vụ,…

Các framework deep learning phân tán hiện có (ví dụ như TensorFlow, MXNet, Petuum, ChainerMN,…) đã áp dụng một kiến trúc trong đó nhiều tác vụ dài hạn, có trạng thái tương tác với nhau để tính toán mô hình và đồng bộ hóa tham số, thường theo cách chặn để hỗ trợ huấn luyện phân tán đồng bộ. Mặc dù điều này được tối ưu hóa cho giao tiếp liên tục giữa các tác vụ, nhưng chỉ hỗ trợ khôi phục lỗi có độ mịn thô bằng cách bắt đầu lại từ snapshot trước đó (ví dụ như vài epoch trước đó).

Ngược lại, BigDL chạy một loạt các Spark jobs ngắn hạn (ví dụ như hai jobs trên mỗi mini-batch được mô tả trước đó), và mỗi tác vụ trong job đều không có trạng thái, không chặn và hoàn toàn độc lập với nhau. Do đó, các tác vụ BigDL có thể chạy đơn giản mà không cần lập lịch nhóm. Ngoài ra, nó cũng có thể hỗ trợ khôi phục lỗi có độ mịn bằng cách chỉ chạy lại tác vụ bị lỗi (sau đó tạo lại phần của gradient cục bộ hoặc trọng số được cập nhật trong bộ nhớ đệm của Spark). Điều này cho phép framework tự động và hiệu quả giải quyết

các lỗi (ví dụ như giảm quy mô cluster, preemption của tác vụ, bug ngẫu nhiên trong mã,...) kịp thời.

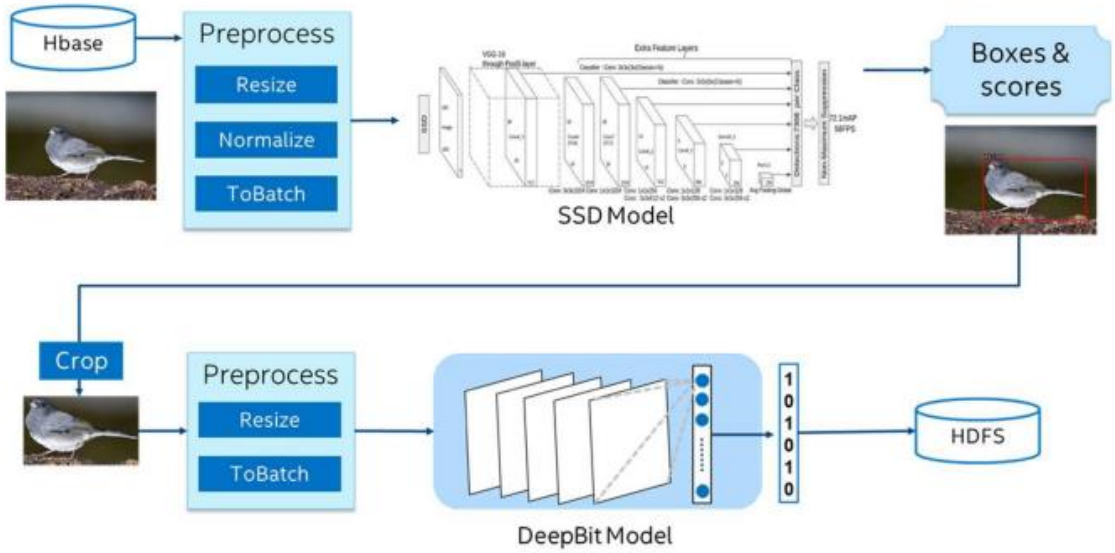
2.3.3 Ứng dụng BigDL với bài toán phân loại và nhận diện hình ảnh

BigDL có thể được sử dụng để giải quyết bài toán phân loại và nhận diện hình ảnh bằng cách sử dụng các mô hình như SSD và DeepBit, được thể hiện trong hình 2.12.

Quá trình phân loại và nhận diện nông sản có thể được thực hiện bằng cách kết hợp các bước sau:

1. Phát hiện vật thể: Sử dụng mô hình SSD trong BigDL để phát hiện các vật thể trong hình ảnh. Mô hình này có thể xác định vị trí và độ tin cậy của các vật thể khác nhau trong hình ảnh.
2. Trích xuất hình ảnh: Dựa trên địa điểm và kích thước của các vật thể đã được phát hiện, bạn có thể cắt ra các hình ảnh đầy đủ chứa các vật thể này hoặc tập trung vào các vùng quan tâm để nhận diện.
3. Trích xuất đặc trưng: Sử dụng mô hình DeepBit trong BigDL để trích xuất các đặc trưng của các hình ảnh đã được cắt ra. Mô hình này có thể chuyển đổi các hình ảnh thành các vectơ đặc trưng có số chiều thấp nhưng biểu diễn thông tin quan trọng về hình dạng và nội dung của đối tượng.
4. Phân loại: Sử dụng các mô hình phân loại nông sản đã được huấn luyện trước để dự đoán loại nông sản dựa trên các đặc trưng đã trích xuất. Kết quả phân loại có thể được lưu trữ trong HDFS dưới dạng RDD (Resilient Distributed Dataset) để sử dụng cho các mục đích khác nhau.

Quá trình này kết hợp việc phát hiện vật thể và trích xuất đặc trưng để nhận diện và phân loại hình ảnh. Các công cụ như BigDL và mô hình SSD, DeepBit trong BigDL có thể hỗ trợ trong việc triển khai quá trình này trên môi trường phân tán và xử lý lượng dữ liệu lớn.



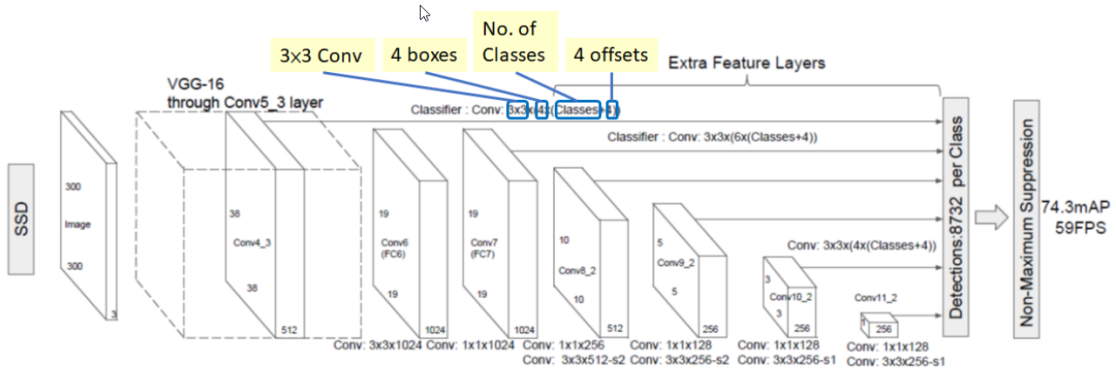
Hình 2.12: Ứng dụng BigDL với bài toán phân loại và nhận diện hình ảnh

❖ SSD Model

Mô hình SSD được xây dựng trên cơ sở của mạng neural tích chập (Convolutional Neural Network - CNN) và các lớp tích chập bổ sung. Kiến trúc của mô hình SSD bao gồm hai phần chính:

- Base Network: Một mạng neural tích chập (CNN) được sử dụng để xử lý ảnh đầu vào và trích xuất các đặc trưng của ảnh. Mạng CNN thường được huấn luyện trước trên một bộ dữ liệu lớn như ImageNet để trích xuất các đặc trưng có giá trị từ ảnh.
- MultiBox Head: Các lớp tích chập bổ sung được thêm vào sau mạng CNN để dự đoán các bounding box và xác suất lớp cho các đối tượng trong ảnh. MultiBox Head bao gồm các lớp tích chập và kết nối đầy đủ (fully connected) để biến đổi đầu vào từ mạng CNN thành các vector đặc trưng dùng để dự đoán vị trí và lớp của đối tượng. Bounding box được dự đoán bằng cách áp dụng một số lượng đặc trưng trên từng vị trí trên ảnh và dự đoán vị trí và kích thước của bounding box. Xác suất lớp cho các đối tượng được dự đoán bằng cách áp dụng một số lượng đặc trưng trên từng vị trí trên ảnh và tính xác suất đối tượng thuộc các lớp đã biết.

Với kiến trúc này, SSD có thể dự đoán bounding box và xác suất lớp tương ứng cho tất cả các đối tượng trong ảnh trong một lần chạy (single shot), giúp cho việc phát hiện đối tượng nhanh chóng và tiết kiệm tài nguyên tính toán.



Hình 2.13: Sơ đồ kiến trúc của mạng SSD [20]

SSD dựa trên việc áp dụng một kiến trúc chuẩn (Ví dụ: VGG16) để thực hiện tiến trình lan truyền thuận và tạo ra một khối feature map 3D ở giai đoạn sớm. Kiến trúc mạng này được gọi là "base network" (từ input Image đến Conv7). Sau đó, chúng ta thêm các kiến trúc phía sau "base network" để tiến hành phát hiện vật thể, được gọi là "Extra Feature Layers" trong sơ đồ. Các lớp này có thể được giải thích một cách đơn giản như sau:

- **Các layer của mô hình SSD**:
    - **Input Layer**: Input Layer của mô hình SSD có kích thước đầu vào là 300x300 pixel và được sử dụng để đưa ảnh đầu vào vào mạng neuron. Để chuẩn bị dữ liệu đầu vào cho Input Layer của mô hình SSD, ảnh đầu vào sẽ được chuyển đổi thành một tensor có kích thước 300x300x3 (chiều rộng, chiều cao và số kênh màu RGB). Sau đó, tensor này sẽ được đưa vào Input Layer để được xử lý bởi các lớp tích chập và lớp pooling trong mô hình SSD.
    - **Conv5_3 Layer**: Conv5_3 là lớp tích chập thứ tư trong mạng VGG-16 được sử dụng trong kiến trúc mô hình SSD nhưng loại

bỏ một số layers fully connected ở cuối cùng. Đầu vào của lớp Conv5_3 là feature map (bản đồ đặc trưng) được tạo ra từ lớp Conv5_2. Conv5_3 sử dụng các bộ lọc (filters) có kích thước 3x3 với số lượng bộ lọc là 512. Conv5_3 sử dụng hàm kích hoạt ReLU để giúp tăng tính phi tuyến của mô hình. Đầu ra của lớp Conv5_3 là một feature map có kích thước 38x38 với 512 kênh đặc trưng.

- **Conv4_3 Layer**: Conv4_3 là một lớp quan trọng trong kiến trúc mô hình SSD, vì nó được sử dụng để tạo ra các feature map tầng thấp hơn so với Conv5_3. Các feature map này được sử dụng để phát hiện các đối tượng nhỏ hơn trên ảnh, vì chúng chứa các đặc trưng của các đối tượng nhỏ hơn và chi tiết hơn so với các feature map tầng cao hơn. Sau khi được tạo ra từ Conv4_3, các feature map này được sử dụng để tạo ra các anchor box để phát hiện đối tượng trên ảnh. Đầu vào của lớp Conv4_3 là feature map (bản đồ đặc trưng) được tạo ra từ lớp Conv4_2. Conv4_3 sử dụng các bộ lọc (filters) có kích thước 3x3 với số lượng bộ lọc là 512. Conv4_3 sử dụng hàm kích hoạt ReLU để giúp tăng tính phi tuyến của mô hình. Đầu ra của lớp Conv4_3 là một feature map có kích thước 38x38 với 512 kênh đặc trưng.
- Sau khi tạo ra các feature map từ các lớp tích chập Conv4_3 và Conv5_3, mô hình SSD sẽ áp dụng các lớp classifier để dự đoán lớp và tọa độ của các đối tượng trên ảnh. Các lớp classifier trong mô hình SSD bao gồm các lớp Convolutional (Conv) và Fully Connected (FC), được sử dụng để tạo ra các dự đoán cho lớp và tọa độ của các đối tượng trên ảnh. Các lớp này sử dụng feature map được tạo ra từ các lớp tích chập trước đó làm đầu vào. Kích thước của các feature map sau đó sẽ phụ thuộc vào stride (độ lớn

bước nhảy) của tích chập và kích thước kernel filter (thường là 3x3) của các lớp tích chập trước đó. Tuy nhiên, để đảm bảo rằng các default bounding boxes được xác định đúng vị trí trên ảnh, các feature map sau đó sẽ được kết hợp với các lớp Convolutional và Fully Connected khác để tạo ra các dự đoán cho lớp và tọa độ của các đối tượng.

Trên mỗi cell của các feature map, mô hình SSD sẽ xác định một số lượng default bounding boxes. Đối với các feature map tầng thấp hơn như Conv4_3 và Conv7, số lượng default bounding boxes sẽ nhiều hơn để phát hiện các đối tượng nhỏ hơn và chi tiết hơn. Các default bounding boxes này sẽ được sử dụng để tính toán offset tọa độ của các đối tượng so với các default bounding boxes. Sau đó, các dự đoán về lớp và tọa độ của các đối tượng sẽ được tạo ra bằng cách kết hợp các dự đoán từ các feature map khác nhau.Do đó, số lượng các default boxes sản sinh ra ở các layers tiếp theo lần lượt như sau:

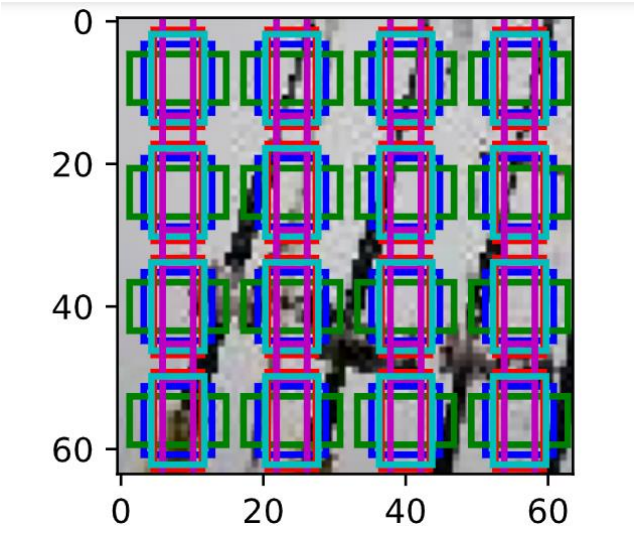
- **Conv7**: 19×19×6 = 2166 boxes (6 boxes/cell)
- **Conv8_2**: 10×10×6 = 600 boxes (6 boxes/cell)
- **Conv9_2**: 5×5×6 = 150 boxes (6 boxes/cell)
- **Conv10_2**: 3×3×4 = 36 boxes (4 boxes/cell)
- **Conv11_2**: 1×1×4 = 4 boxes (4 boxes/cell). Lớp Conv11_2 trong mô hình SSD là một lớp tích chập (Convolutional layer) được sử dụng để tạo ra các dự đoán cho tọa độ của các đối tượng trên ảnh. Kết quả của lớp Conv11_2 là một feature map với kích thước giảm xuống so với feature map đầu vào và được sử dụng để tạo ra các dự đoán về tọa độ của các đối tượng trên ảnh. Các dự đoán này sẽ được tính toán bằng cách áp dụng các lớp Fully Connected

(FC) lên feature map này.

- **Áp dụng các feature**: Sau khi thu được feature map ở base network, các convolutional layers sẽ được thêm vào sau base network để giảm kích thước của feature map và cho phép dự báo và nhận diện vật thể ở nhiều hình dạng kích thước khác nhau. Các feature map có kích thước lớn sẽ phát hiện tốt các vật thể nhỏ và các feature map kích thước nhỏ giúp phát hiện tốt hơn các vật thể lớn. Việc chọn kích thước kernel filters sẽ được thể hiện chi tiết hơn trong sơ đồ kiến trúc trong phần Extra Features Layers. Việc giảm kích thước feature map giúp giảm số lượng khung hình cần dự báo và tăng độ chính xác của phát hiện vật thể. Feature map có kích thước lớn tốt cho việc phát hiện vật thể nhỏ, trong khi feature map có kích thước nhỏ thì phù hợp hơn với việc phát hiện vật thể lớn. Sơ đồ kiến trúc của mô hình SSD sẽ chỉ rõ về kích thước kernel filters trong phần Extra Features Layers.
- **Dự báo thông qua mạng tích chập**: Mỗi một layer feature được thêm vào trong Extra Features Layers của mô hình SSD sẽ tạo ra một tập hợp các dự đoán cố định giúp nhận diện đối tượng trong ảnh thông qua việc áp dụng các bộ lọc tích chập. Kích thước đầu ra của mỗi layer feature (với chiều rộng x chiều cao x số kênh) phụ thuộc vào kích thước của kernel filter và được tính toán hoàn toàn tương tự như trong mạng neural tích chập thông thường.

- **Default box và Aspect ratio**: Mỗi cell trên feature map sẽ được liên kết với một tập hợp các default bounding boxes. Các default boxes sẽ được phân bố trên feature map theo thứ tự từ trên xuống dưới và từ trái qua phải để tính tích chập, do đó vị trí của mỗi default box sẽ tương ứng với cell mà nó được liên kết và ánh xạ với một vùng ảnh trên bức ảnh gốc. Ngoài ra, mỗi default box sẽ có một tỷ lệ cạnh (aspect ratio) cố định để phù hợp với

độ tỉ lệ khung hình của các đối tượng cần phát hiện.



Hình 2.14: Vị trí của các default bounding box trên bức ảnh gốc khi áp dụng trên feature map có kích thước 4 x 4.

Như vậy, mỗi ô lưới trên feature map sẽ có kích thước là 4x4 và sẽ được liên kết với 4 default bounding box khác nhau như được minh họa trên hình vẽ. Tất cả các bounding box này có tâm trùng nhau và chính là tọa độ tâm của ô lưới mà chúng liên kết.

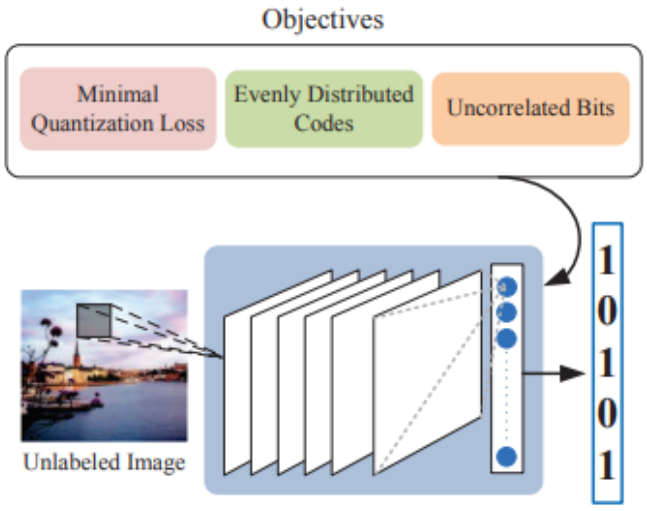
Tại mỗi một default bounding box trên feature map, chúng ta sẽ dự báo 4 offsets tương ứng với tọa độ và kích thước của nó. Các offsets này được biểu diễn bởi một tọa độ gồm 4 tham số (cx, cy, w, h), trong đó (cx, cy) xác định tọa độ tâm và (w, h) xác định kích thước của bounding box. Phần thứ hai trong dự báo là điểm số của bounding box tương ứng với mỗi lớp. Lưu ý rằng chúng ta sẽ có một lớp thứ C+1 để đại diện cho trường hợp mà default bounding box không chứa vật thể (hoặc thuộc lớp background).

Tương tự như anchor boxes trong mạng faster R-CNN, default boxes cũng được sử dụng trên một vài feature maps với các độ phân giải khác nhau. Điều này giúp cho các default bounding box có thể phân biệt hiệu quả kích thước của các vật thể khác nhau.

❖ **DeepBit Model**

Phương pháp phân loại DeepBit Model là một phương pháp học sâu không giám sát mới trong việc nhận diện vật thể. Khác với các phương pháp giải mã nhị phân khác, DeepBit Model sử dụng một mạng neuron để học bộ giải mã một cách không giám sát bằng cách đặt ba mục tiêu cho bộ giải mã. Các mục tiêu đó bao gồm:

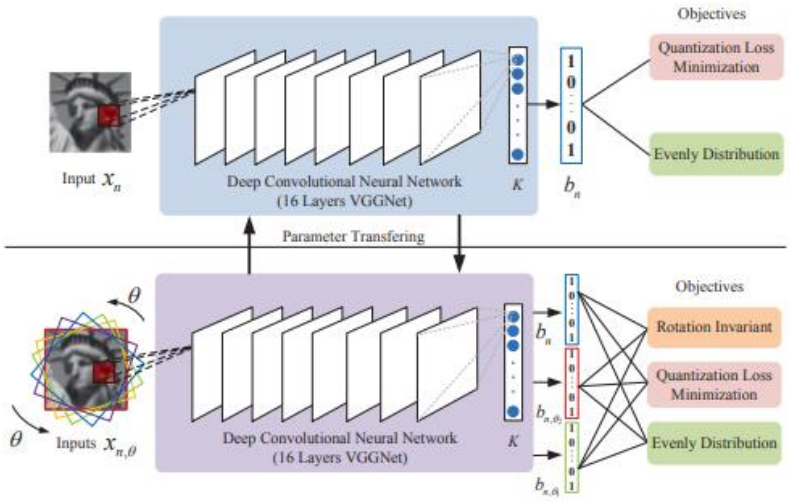
- Phân cụm đối tượng: Bộ giải mã được đào tạo để phân chia các đối tượng vào các cụm khác nhau. Khi đối tượng mới được đưa vào, bộ giải mã sẽ xác định cụm tương ứng cho nó.
- Phát hiện bên trong cụm: Bộ giải mã cũng được đào tạo để phát hiện bên trong các cụm và tạo ra một biểu diễn phù hợp cho chúng.
- Tạo biểu diễn đối tượng: Cuối cùng, bộ giải mã sẽ tạo ra một biểu diễn đối tượng cho mỗi đối tượng bằng cách kết hợp các biểu diễn của cụm và các đặc trưng bên trong cụm đó.



Hình 2.15: DeepBit Model

Các mục tiêu này giúp bộ giải mã học được các đặc trưng quan trọng của đối tượng một cách tự động, từ đó giúp nó có thể phân loại đối tượng hiệu quả hơn. Điều này giúp DeepBit Model tránh được những hạn chế của các phương

pháp giải mã nhị phân truyền thống, giúp cải thiện độ chính xác và hiệu quả của việc nhận diện vật thể.



Hình 2.16: Mô hình chi tiết của DeepBit Model

Mô hình xây dựng bộ giải mã bằng cách đặt cửa sổ chiếu lên ảnh đầu và nhị phân hóa kết quả

$$b = 0.5 \times (\text{sign}(\mathcal{F}(x; \mathcal{W})) + 1), \qquad (1)$$

x đại diện cho ảnh đầu vào, b là bộ giải mã trong dạng vector. Sign(k)=1 nếu k>0 và bằng -1 nếu ngược lại. F(x, W) là 1 tập hợp các chức năng chiếu xuống có thể viết như sau:

$$\mathcal{F}(x; \mathcal{W}) = f_k(\cdots f_2(f_1(x; w_1); w_2) \cdots ; w_k), \qquad (2)$$

f lấy dữ liệu xi và tham số wi là đầu vào và tạo ra kết quả chiếu xuống xi+1

Cách giải quyết này dùng giúp thông số đối chiếu W=(w1,w2,w3,…wn) lượng tử hóa hình ảnh đầu vào x thành 1 vector b nhị phân gọn nhẹ mà không làm mất thông tin từ đầu vào. Để tạo 1 bộ giải mã gọn nhẹ và phân biệt tốt, bộ giải mã phải có sự mất mát lượng tử hóa ít nhất để giữ được cấu trúc dữ liệu từ lớp trước. Thứ hai, bộ giải mã phải phân bố đồng đều để xâu nhị phân phân biệt được nhiều thông điệp khác nhau hơn. Cuối cùng bộ giải mã phải bất biến trước

sự xoay hay nhiễu của vật thể, từ đó bộ giải mã có thể bắt được nhiều thông tin hơn từ mội ảnh.

2.4 Kết luận chương

Tại chương này, luận văn đã trình bày tổng quan các phương pháp nhận diện đặt ra và lý thuyết cho hệ thống.

Đầu tiên, luận văn đã giải thích các khái niệm cơ bản về nhận diện đối tượng và học sâu, cùng với kiến trúc mạng phổ biến như Convolutional Neural Networks (CNNs).

Tiếp theo, luận văn cũng đã giới thiệu về BigDL - một thư viện học sâu mã nguồn mở - và cách sử dụng BigDL cho bài toán nhận diện hình ảnh. Sử dụng BigDL có thể tăng tốc độ đào tạo và chạy mô hình, đồng thời cũng giảm thiểu thời gian xử lý dữ liệu. Từ những kiến thức và kinh nghiệm đã tìm hiểu được trong chương này, luận văn sẽ ứng dụng BigDL cho bài toán nhận diện và phân loại nông sản ở chương 3.

CHƯƠNG 3 . KẾT QUẢ THỰC NGHIỆM VÀ ĐÁNH GIÁ

3.1 Thu thập dữ liệu

Bước thu thập dữ liệu là một giai đoạn quan trọng trong quá trình xây dựng cơ sở dữ liệu nông sản. Quá trình này đòi hỏi sự cẩn thận và chuẩn bị kỹ lưỡng để đảm bảo tính chính xác và đa dạng của dữ liệu thu thập được.

Bước đầu tiên trong quá trình thu thập dữ liệu nông sản là tìm kiếm các nguồn dữ liệu phù hợp trong các dataset chứa ảnh nông sản trên mạng thông qua các nguồn thông tin như Kaggle, Open Images,…; ngoài ra thu thập thêm hình ảnh bằng cách chụp ảnh nông sản bằng thiết bị camera cá nhân.

Danh sách các nông sản được sử dụng bao gồm:

Nông sản	**Số lượng ảnh**	**Nông sản**	**Số lượng ảnh**	**Nông sản**	**Số lượng ảnh**
Táo	510	Bưởi	520	Lê	520
Mơ	530	Ổi	520	Ớt chuông	520
Chuối	540	Su hào	550	Hồng	570
Cải Bắp	550	Quất	560	Dứa	570
Dưa lưới	580	Chanh	510	Thanh long	590
Khế	520	Nhãn	520	Mận	590
Cà rốt	570	Vải	570	Lựu	590
Súp lơ	530	Xoài	540	Khoai tây	520
Dừa	560	Măng cụt	590	Chôm chôm	520
Ngô	530	Dâu tằm	540	Hồng xiêm	510
Cà tím	540	Hành tây	550	Dâu tây	560
Tỏi	580	Cam	540	Cà chua	560

Gừng	590	Đào	560	Dưa hấu	560
Nho	590				

Bước tiếp theo sau khi đã có tập hợp ảnh chụp là thu thập và tổ chức dữ liệu. Trong quá trình này, cần xác định các thông tin liên quan như nhãn, định danh, vị trí chụp và các thông tin khác cần thiết. Thông tin nhãn được sử dụng để chỉ định loại nông sản mà mỗi ảnh đại diện. Ví dụ: "Táo", "Mơ", "Chuối", "Cải Bắp", ... Nhãn giúp xác định và phân loại các loại nông sản trong cơ sở dữ liệu. Định danh được sử dụng để xác định duy nhất mỗi ảnh trong cơ sở dữ liệu. Điều này đảm bảo rằng mỗi hình ảnh được định danh riêng biệt và không bị trùng lặp trong quá trình tổ chức dữ liệu.

Quá trình thu thập và tổ chức dữ liệu nông sản cần đảm bảo tính chính xác và đầy đủ của thông tin. Thông tin nhãn, định danh và vị trí chụp giúp xác định và phân loại các loại nông sản và xây dựng cơ sở dữ liệu một cách cụ thể và có tổ chức, cụ thể với bài toán này cơ sở dữ liệu COCO JSON. COCO JSON cung cấp một cấu trúc dữ liệu đa dạng để lưu trữ thông tin về đối tượng, vùng quan tâm, nhãn và các thông tin liên quan khác trong một tệp JSON. Định dạng này cho phép bạn tổ chức dữ liệu theo các danh sách đối tượng, trong đó mỗi đối tượng được mô tả bằng nhãn và các đặc điểm khác như tọa độ, độ chính xác, …

3.2 Thực nghiệm với các phương pháp

Thực nghiệm với phương pháp Học máy truyền thống:

- **Bước 1:** Xây dựng CSDL ảnh nông sản (bước 3.1)
- **Bước 2**: Tiền xử lý ảnh trong CSDL (đồng bộ kích cỡ, lọc nền) và gán nhãn. Đặc trưng của bộ CSDL ảnh này là các ảnh đều được thu thập bằng cách chụp thủ công, nhằm đảm bảo các ảnh có chất lượng cao, có cùng kích thước và tỉ lệ ảnh, với nền đã bị loại bỏ hoàn toàn.

Hình 3.1: Một số ảnh đã lọc nền trong bộ CSDL

- **Bước 3:** Chọn lọc đặc trưng, cụ thể:
  - *Về màu sắc*:

Sử dụng 16 đặc trưng về số lượng các điểm ảnh với giá trị màu tính theo hệ màu HSI (Hue-Saturation-Intensity). Ta không sử dụng hệ màu thường gặp nhất là RGB bởi sau khi chuyển sang hệ màu HSI, ta đã có thể tách biệt được thông tin màu sắc với những thành phần khác như độ sáng, sự bão hòa…

Cụ thể hơn, ta chia dải màu Hue thành 12 đoạn tương ứng với 12 dải màu chính (đỏ, vàng, xanh lục…) và chia dải giá trị độ thuần khiết màu sắc Saturation thành 4 đoạn, sau đó thống kê số điểm ảnh có giá trị điểm màu nằm trong các dải này để thu được 16 giá trị đặc trưng về màu sắc cho mỗi ảnh đầu vào.

  - *Về hình dạng*:

Sử dụng 4 đặc trưng về hình dạng của nông sản trong ảnh là chu vi, diện tích, độ dài lớn nhất, độ rộng lớn nhất của nông sản trong ảnh.

  - *Về kết cấu*:

Sử dụng 10 đặc trưng về kết cấu, là 10 tham số trong bộ ma trận GLCM (Grey Level Co-occurrence Matrix) – một ma trận tính toán đặc trưng kết cấu phổ biến trong lĩnh vực Xử lý ảnh

Tổng kết lại, với mỗi ảnh đầu vào ta sẽ tính toán được 30 giá trị đại diện cho 30 đặc trưng về màu sắc, hình dạng và kết cấu. Những đặc trưng này được chọn lựa sau quá trình tìm hiểu các bài báo, công trình khoa học về sử dụng

Học máy trong bài toán nhận dạng nông sản và thống kê các đặc trưng được sử dụng nhiều nhất, đạt hiệu quả tốt nhất. [2][3][4]

- **Bước 4:** Huấn luyện mô hình nhận dạng nông sản từ CSDL ảnh đã xây dựng. Bộ CSDL ảnh này chỉ để so sánh tương đối độ chính xác của mô hình truyền thống so với mô hình học sâu tiên tiến bây giờ.
- **Bước 5:** Thống kê độ chính xác của bộ test với tỉ lệ bộ training/test là 75/25.

Thực nghiệm với phương pháp Học sâu (sử dụng BIGDL):

- **Bước 1. Chuẩn bị dữ liệu:** Xây dựng CSDL ảnh nông sản, kèm theo các nhãn cho từng hình ảnh để đánh dấu loại nông sản tương ứng
- **Bước 2**. **Tiền xử lý ảnh:** Ứng dụng mô hình SSD Model chuẩn bị dữ liệu trước khi đưa vào mô hình để huấn luyện hoặc dự đoán. Các bước thực hiện như sau:

- Resize ảnh: Kích thước ảnh đầu vào thường khác nhau, vì vậy cần chuyển về kích thước 512 x 512 pixel để đưa vào mô hình.
- Chuẩn hóa giá trị pixel: Các giá trị pixel của ảnh thường nằm trong khoảng từ 0 đến 255, vì vậy cần chuẩn hóa chúng về khoảng từ -1 đến 1 để giúp cho mô hình hội tụ nhanh hơn.
- Tạo feature map: Từ ảnh đầu vào, ta sử dụng một mạng tích chập để tạo ra feature map, tức là một ma trận các giá trị đại diện cho các đặc trưng của ảnh.
- Tạo anchor boxes: Sử dụng anchor boxes để dự đoán vị trí và kích thước của các đối tượng trong ảnh. Anchor boxes là các hình chữ nhật có kích thước và tỷ lệ khác nhau được đặt trên toàn bộ ảnh.
- Chuẩn bị dữ liệu huấn luyện: Từ các anchor boxes và thông tin về đối tượng trong ảnh, ta tạo ra các dữ liệu huấn luyện gồm đầu ra mong muốn (ground truth) cho mô hình.

- **Bước 3**. **Xây dựng mô hình:** Sử dụng BigDL để xây dựng mô hình phân loại nông sản. Mô hình sử dụng SSD model để phát hiện vật thể trong ảnh, sau đó sử dụng DeepBit model để rút trích đặc trưng và phân loại nông sản. Các bước thực hiện như sau:

- Huấn luyện mô hình phát hiện vật thể sử dụng SSD model
  - Sử dụng mô hình SSD có sẵn trong BigDL
  - Thiết lập các tham số cho mô hình số lớp và kích thước ảnh đầu vào là 512 x 512 pixel.
  - Thực hiện fine-tuning trên tập dữ liệu: Trong quá trình này, mô hình sẽ được cập nhật để phân loại các đối tượng trong tập dữ liệu mới.
  - Lưu trữ trọng số của mô hình phát hiện vật thể.
- Rút trích đặc trưng và phân loại nông sản sử dụng DeepBit model
  - Sử dụng trọng số đã được huấn luyện từ SSD model để thực hiện rút trích đặc trưng từ ảnh nông sản.
  - Sử dụng mô hình DeepBit có sẵn trong BigDL để phân loại nông sản dựa trên đặc trưng đã được rút trích.
  - Sử dụng các lớp Convolutional Neural Network (CNN) để trích xuất các đặc trưng từ đầu vào.
  - Sau khi trích xuất được các đặc trưng từ ảnh, kết nối các lớp CNN với nhau bằng các Fully Connected Layer để tạo ra một mạng Neural Network hoàn chỉnh.
  - Cuối cùng, sử dụng một Softmax Activation Layer để đưa ra dự đoán cho từng lớp của dữ liệu và sử dụng hàm Loss Function (như Cross-Entropy Loss) để tính toán độ chính xác của mô hình.

- **Bước 4**. **Huấn luyện mô hình:** Sử dụng tập dữ liệu đã chuẩn bị để huấn luyện mô hình. Sử dụng thuật toán Stochastic Gradient Descent (SGD) để tối ưu hóa hàm loss function.

Trong mỗi epoch, chia tập huấn luyện thành các mini-batch (tập mẫu nhỏ) và áp dụng SGD để cập nhật trọng số của mô hình. Công thức được áp dụng như sau:

w = w - lr * gradient

Trong đó:

- w là vector trọng số cần cập nhật
- lr là learning rate, tốc độ học của mô hình
- gradient là gradient của hàm mất mát tính trên mini-batch hiện tại

Với SGD, gradient được tính trên từng mini-batch, điều này giúp tăng tốc quá trình huấn luyện mô hình và giảm thiểu chi phí tính toán so với GD truyền thống.

- **Bước 5. Đánh giá mô hình:** Thống kê độ chính xác của bộ test với tỉ lệ bộ training/test là 75/25.

Đánh giá kết quả:

Với kết quả thu được từ hai mô hình huấn luyện sử dụng hai phương pháp khác nhau trên cùng một bộ CSDL ảnh chất lượng tốt và đã được tiền xử lý cũng như gán nhãn cẩn thận, ta có thể rút ra kết luận như sau: Với các bài toán nhận dạng và phân loại đối tượng nói chung, trong đó rất khó có thể chọn được các đặc trưng hiệu quả, thì Học sâu là phương pháp có ưu thế vượt trội so với các phương pháp Học máy truyền thống. Học sâu giúp đơn giản hóa quá trình huấn luyện mô hình nhận dạng khi không yêu cầu sự tham gia của người huấn luyện trong quá trình trích chọn đặc trưng, đồng thời cho phép tái sử dụng các mô hình đã huấn luyện trước để giảm thời gian cài đặt giải pháp cho các bài toán nhận dạng mới.

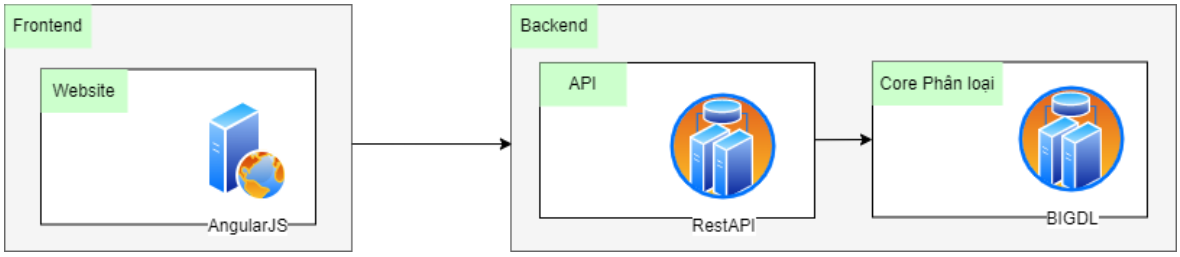
Thông tin tổng quan về bộ CSDL ảnh và quá trình huấn luyện cũng như kết quả đạt được của hai phương pháp cũng được tóm lược trong bảng bên dưới:

Bảng 3.1: So sánh sơ bộ kết quả huấn luyện của 2 phương pháp

	Thời gian huấn luyện	Độ chính xác
Học máy truyền thống	~30 phút	70,82%
Học sâu (sử dụng BIGDL)	~60 phút	~95.34%

3.3 Ứng dụng Nhận diện và phân loại nông sản

Ứng dụng nhận diện và phân loại nông sản được xây dựng theo mô hình Frontend/Backend bao gồm 2 thành phần chính:



Hình 3.2: Mô hình Ứng dụng Nhận dạng nông sản

- **Frontend:** Giao diện website tương tác với người dùng (được trình bày ở phần 3.4 kết quả) cho phép người dùng gửi ảnh lên hệ thống Ứng dụng Nhận diện và phân loại nông sản và nhận được kết quả phân loại nông sản tương ứng.
- **Backend:** Hệ thống xử lý logic và dữ liệu của ứng dụng phân loại và nhận diện nông sản, xử lý các yêu cầu và trả về dữ liệu cho phía frontend. Hệ thống này được xây dựng bao gồm 2 thành phần:
  - Restful API: Hệ thống sử dụng để truyền tải dữ liệu giữa các ứng dụng phía Frontend và phía Core phân loại. Khi người dùng gửi yêu cầu đến server thông qua một URL, server sẽ trả về một response chứa dữ liệu được yêu cầu. Request mà Frontend gửi lên sẽ bao gồm

ảnh chụp của nông sản cần nhận diện. Response được trả về Frontend là danh sách dự đoán nông sản dưới dạng JSON:

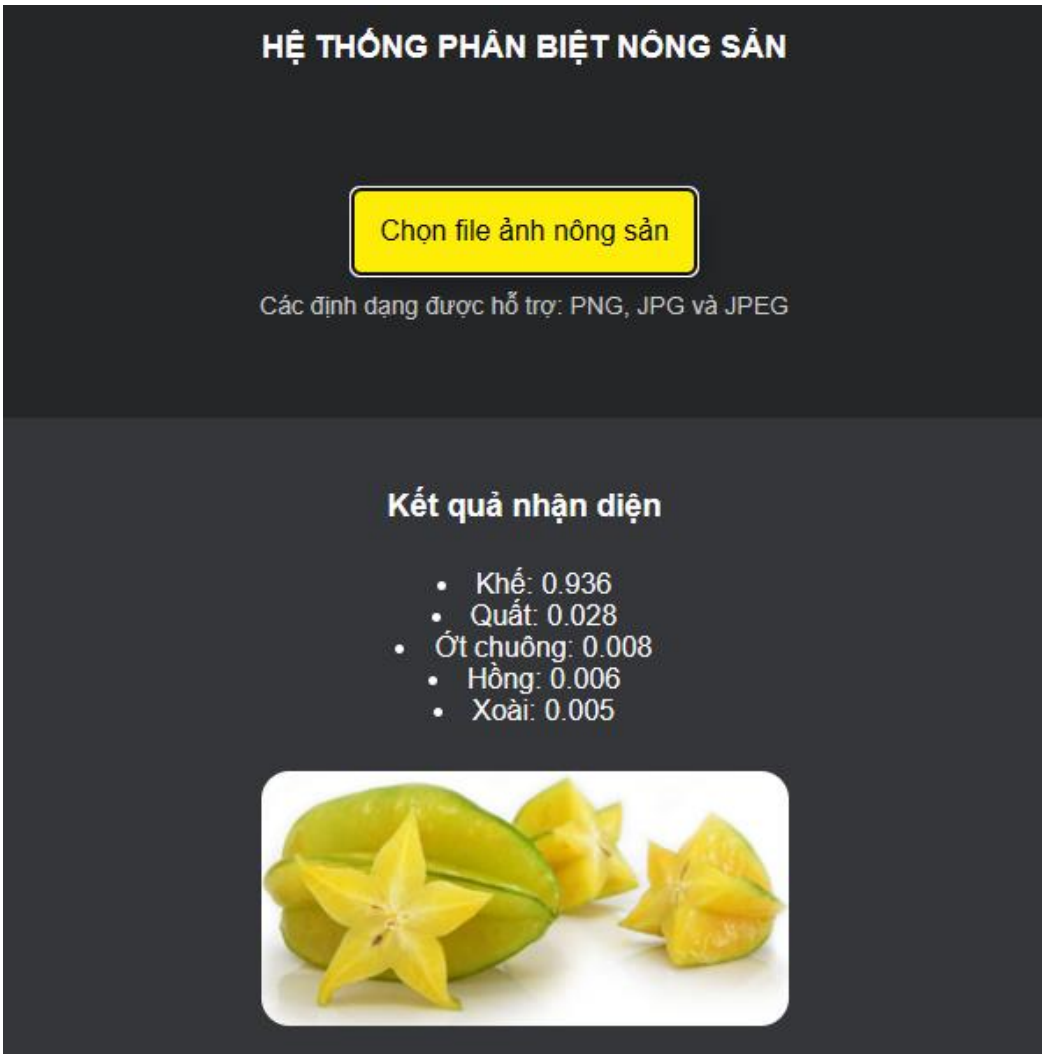

```
{
  "code": "00",
  "desc": "Thành công",
  "data": [
    {
      "id": "1",
      "name": "Khế",
      "ratio": "0.936"
    },
    {
      "id": "2",
      "name": "Quất",
      "ratio": "0.028"
    },
    {
      "id": "3",
      "name": "Ớt Chuông",
      "ratio": "0.008"
    },
    {
      "id": "4",
      "name": "Hồng",
      "ratio": "0.006"
    }
  ]
}
```


- Core Phân loại: Hệ thống phân loại nông sản thông qua framework BIGDL (được trình bày ở phần 3.2).

3.4Kết quả

Ứng dụng Nhận diện và phân loại nông sản đã được thử nghiệm thực tế với nhiều mẫu nông sản khác nhau, được chia thành hai nhóm chính: Nhóm đã được huấn luyện nhận dạng và nhóm chưa được huấn luyện. Kết quả đạt được tương đối tốt, cụ thể như sau:

-Nhóm nông sản nằm trong danh sách nông sản được chọn để xây dựng bộ dữ liệu nhận dạng: Kết quả nhận dạng đạt độ chính xác khá cao, đặc biệt là với những loại nông sản có nét đặc trưng về màu sắc hoặc hình dạng như chuối, thanh long, chôm chôm…



Hình 3.3: Kết quả nhận dạng tốt với loại nông sản có đặc trưng riêng biệt

Đối với những loại nông sản có nhiều nét tương đồng lẫn nhau, kết quả nhận dạng của ứng dụng còn đôi lúc bị nhầm lẫn, đặc biệt trong các trường hợp ảnh được chụp theo góc nhìn chưa tốt dẫn đến ảnh không thể hiện được các đặc trưng riêng của quả. Nguyên nhân dẫn đến nhầm lẫn bao gồm:

- *Thiếu dữ liệu đa dạng:* Nếu tập dữ liệu sử dụng để huấn luyện mô hình không đủ đa dạng, mô hình có thể không nhận diện được các đặc điểm khác biệt giữa các loại nông sản.
- *Sai số trong quá trình thu thập dữ liệu:* Nếu dữ liệu được sử dụng để huấn luyện mô hình không chính xác hoặc không đầy đủ, các đặc trưng quan trọng có thể bị bỏ qua và dẫn đến sự nhầm lẫn.
- *Độ phân giải ảnh không đủ:* Nếu độ phân giải của ảnh được sử dụng để huấn luyện mô hình không đủ, các chi tiết nhỏ hoặc đặc trưng quan trọng có thể bị mất đi, dẫn đến sự nhầm lẫn.
- *Địa hình và điều kiện ánh sáng:* Nếu hình ảnh được chụp trong điều kiện ánh sáng kém hoặc trên địa hình khác nhau, các đặc trưng quan trọng có thể không được nhận diện, dẫn đến sự nhầm lẫn.
- *Kiến trúc mô hình không phù hợp:* Nếu kiến trúc mô hình được sử dụng không phù hợp với bài toán hoặc tập dữ liệu cụ thể, mô hình có thể không đưa ra các kết quả chính xác.

Hình 3.4: Kết quả nhận dạng chưa tốt với loại quả không có đặc trưng riêng biệt

Trong hình trên, quả đào được chụp ở góc độ không tốt và độ phân giải ảnh không đủ, dẫn đến hệ thống nhận dạng nhầm lẫn giữa các nông sản có hình dạng, kết cấu và màu sắc tương đồng nhau. Tuy nhiên, thông số độ chính xác của mô hình cho thấy rằng tỉ lệ nhận dạng đúng của quả mận là 46,8%, không cao hơn nhiều so với quả đào là 31,2%, và quá thấp so với tỉ lệ nhận dạng thông thường (lớn hơn 90%).

- Nhóm nông sản nằm ngoài danh sách nông sản: Hệ thống sẽ tính toán và trả về kết quả nhận dạng là một trong loại nông sản có tỉ lệ giống nhất với loại quả cần nhận dạng. Độ tương đồng giữa hai loại quả này ta có thể nhận thấy rất rõ ràng:

Hình 3.5: Kết quả nhận dạng với loại quả không được huấn luyện

Trong trường hợp như hình trên, khiyêu cầu hệ thống nhận dạng quả bòn bon, do bòn bon không có trong danh sách nông sản được huấn luyện nhận dạng nên kết quả trả về là loại quả có sự tương đồng cao nhất, quả nhãn.

Ngoài ra, kết quả thực nghiệm thu được cho thấy hệ thống nhận dạng đạt được kết quả tương đối chuẩn xác với các trường hợp hình ảnh quả trong ảnh đầu vào bị che khuất một phần, điều kiện ánh sáng không thực sự tốt cũng như các trường hợp ảnh bị biến dạng nhẹ. Đây chính là các khó khăn đối với bài toán nhận dạng vật thể nói chung mà ta đã đề cập tới trong phần mở đầu của luận văn, lý giải cho điều này là do trong quá trình thu thập ảnh ban đầu cũng như sinh ảnh tự động từ các ảnh gốc, mô hình nhận dạng đã được huấn

luyện để nhận ra các trường hợp tương tự. Khả năng dự đoán mạnh mẽ này đã giúp cho các phương pháp Học sâu, đặc biệt là mạng huấn luyện no ron tích chập SSD trở thành giải pháp mạnh mẽ nhất trong lĩnh vực nhận dạng ảnh bây giờ.

3.5 Kết luận chương

Kết thúc chương, luận văn đã nghiên cứu, tìm hiểu bài toán tự động nhận dạng và phân loại nông sản trong ảnh màu, và thực hiện phát triển, cài đặt phương án giải quyết cho bài toán dựa trên sự thống kê các hướng tiếp cận đã được công bố qua rất nhiều bài báo, công trình khoa học trên thế giới. Các kết quả chính mà luận văn đã đạt được, tương ứng với các mục tiêu đề ra.

KẾT LUẬN

Đề tài luận văn đã nghiên cứu về mô hình học sâu như CNN cũng như tiến hành xây dựng cơ sở dữ liệu ảnh nông sản và phát triển một hệ thống nhận diện nông sản sử dụng BigDL. Qua quá trình nghiên cứu và thực hiện, luận văn đã đạt được những kết quả đáng kể và đối mặt với một số khó khăn.

Việc xây dựng cơ sở dữ liệu ảnh nông sản là một bước quan trọng, đảm bảo nguồn dữ liệu phong phú và đa dạng để huấn luyện và đánh giá mô hình nhận diện. Luận văn đã hoàn thiện quá trình thu thập ảnh từ các nguồn dữ liệu khác nhau hoặc tự chụp bằng thiết bị cá nhân. Các ảnh được tổ chức và gán nhãn theo từng loại nông sản. Điều này tạo ra một bộ cơ sở dữ liệu đầy đủ và có tổ chức, cung cấp nguồn dữ liệu phong phú cho việc huấn luyện và kiểm tra mô hình nhận diện nông sản. Ngoài ra, luận văn đã nghiên cứu và thống kê các đặc trưng thường được sử dụng trong việc nhận diện nông sản, bao gồm màu sắc, hình dạng và kết cấu. Việc này giúp xác định những đặc điểm quan trọng của các loại nông sản và tạo ra một cơ sở lý thuyết cho việc phân loại và nhận diện.

Trong quá trình nghiên cứu, luận văn đã tiếp cận và thử nghiệm các phương pháp trí tuệ nhân tạo (bao gồm cả học máy và học sâu) để nhận diện nông sản. Bằng việc sử dụng mô hình BIGDL, hệ thống nhận diện nông sản đã được xây dựng và kiểm thử, và kết quả đạt được đã chứng minh hiệu quả và độ chính xác của phương pháp đề xuất.

Tuy nhiên, luận văn cũng đã đối mặt với một số khó khăn đáng kể trong quá trình nghiên cứu và thực hiện. Một trong những khó khăn chính là việc thu thập dữ liệu ảnh nông sản. Việc này đòi hỏi sự đầu tư về thời gian, công sức và tài nguyên. Không chỉ đơn thuần là việc chụp ảnh, mà còn cần đảm bảo chất lượng ảnh, đa dạng hóa các góc chụp, ánh sáng và nền phù hợp để đảm bảo ảnh chụp có độ rõ nét và chính xác cao. Gán nhãn và xử lý dữ liệu cũng là một thách

thức khó khăn trong quá trình xây dựng cơ sở dữ liệu. Việc gán nhãn đúng và chính xác từng loại nông sản trong ảnh đòi hỏi sự tỉ mỉ và kiên nhẫn. Ngoài ra, việc xử lý dữ liệu, bao gồm tiền xử lý ảnh, cắt bớt phần không cần thiết, điều chỉnh độ sáng và chuyển đổi định dạng ảnh, cũng đòi hỏi sự cẩn thận và sự hiểu biết về kỹ thuật xử lý ảnh.

Tổng hợp lại, đề tài luận văn đã đạt được những kết quả quan trọng trong việc xây dựng cơ sở dữ liệu ảnh nông sản, nghiên cứu các đặc trưng và xây dựng mô hình nhận diện. Các kết quả này mang lại sự tiện ích và ứng dụng trong việc phân loại và nhận diện nông sản, góp phần nâng cao hiệu quả và tự động hóa trong lĩnh vực nông nghiệp.

DANH MỤC CÁC TÀI LIỆU THAM KHẢO

[1] Andrej Karpathy. CS231n Convolutional Neural Networks for Visual Recognition - Image Classification. http://cs231n.github.io/classification/

[2] Sadrnia, H., Rajabipour, A., Jafary, A., Javadi, A., & Mostofi, Y. (2007). Classification and analysis of fruit shapes in long type watermelon using image processing. Int J Agric Biol

[3] Fu, L., Sun, S., Li, R., & Wang, S. (2016). Classification of kiwifruit grades based on fruit shape using a single camera. Sensors (Switzerland)

[4] Seng, W. C., & Mirisaee, S. H. (2009). A new method for fruits recognition system. Proceedings of the 2009 International Conference on Electrical Engineering and Informatics, ICEEI 2009,

[5] Arivazhagan, S., Shebiah, R. N., Nidhyanandhan, S. S., & Ganesan, L. (2010). Fruit Recognition using Color and Texture Features. Information Sciences, 1(2), 90–94.

[6] Zhang, Y., & Wu, L. (2012). Classification of fruits using computer vision and a multiclass support vector machine. Sensors (Switzerland), 12(9), 12489–12505.

[7] Naskar, S. (2015). A Fruit Recognition Technique using Multiple Features and Artificial Neural Network, 116(20), 23–28.

[8] Richard Szeliski. (2010). Computer Vision: Algorithms and Applications.

[9] Jian Yang ,Chongchong Zhao, Big Data Market Optimization Pricing Model Based on Data Quality

[10] GilPress. (2016). Visually Linking AI, Machine Learning, Deep Learning, Big Data and Data Science | What's The Big Data? https://whatsthebigdata.com/2016/10/17/visually-linking-ai-machine-learning-deep-learning-big-data-and-data-science/

[11] Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations.

[12] Alexandros Agapitos, Michael O'Neill, Miguel Nicolau, David Fagan,. (2015). Deep evolution of image representations for handwritten digit recognition.

[13] Dumoulin, V., & Visin, F. (2016). A guide to convolution arithmetic for deep learning.

[14] Samer, C. H., Rishi, K., & Rowen. (2015). Image Recognition Using Convolutional Neural Networks. Cadence Whitepaper,

[15] Jason Dai, Xianyan J., Wang Zhenhua. (2019). Building Large-Scale Image Feature Extraction with BigDL at JD.com.

[16] https://www.deeplearning.ai/resources/natural-language-processing/

[17] Iván Sánchez Fernández, Jurriaan M. Peters (2023) Machine learning and deep learning in medicine and neuroimaging.

[18] S Ren, Kaiming H, Ross G, and Jian S (2016) Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

[19] Joseph R , Santosh D, Ross G , Ali F (2016) You Only Look Once: Unified, Real-Time Object Detection

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg (2015) SSD: Single Shot MultiBox Detector

[21] Jason (Jinquan) Dai, Yiheng Wang, Xin Qiu, Ding Ding, Yao Zhang, Yanzhang Wang, Xianyan Jia, Cherry (Li) Zhang (2019) BigDL: A Distributed Deep Learning Framework for Big Data