

BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM

--- 📖 ---



ĐỒ ÁN TỐT NGHIỆP

CNN TRONG PHÂN LOẠI RÁC

SINH VIÊN THỰC HIỆN

: TRẦN VŨ TRỌNG HAI

MÃ SINH VIÊN

: 1451020071

KHOA

: CÔNG NGHỆ THÔNG TIN

HÀ NỘI - 2024

**BỘ GIÁO DỤC ĐÀO TẠO
TRƯỜNG ĐẠI HỌC ĐẠI NAM**



TRẦN VŨ TRỌNG HAI

CNN TRONG PHÂN LOẠI RÁC

CHUYÊN NGÀNH : CÔNG NGHỆ THÔNG TIN
MÃ SINH VIÊN : 1451020071

NGƯỜI HƯỚNG DẪN: ThS. TRẦN THU TRANG

HÀ NỘI - 2024

LỜI CAM ĐOAN

Em xin cam đoan rằng đồ án tốt nghiệp với đề tài "CNN TRONG PHÂN LOẠI RÁC" là kết quả nghiên cứu và làm việc nghiêm túc của bản thân em dưới sự hướng dẫn của Cô Trần Thu Trang.

Toàn bộ nội dung trong đồ án này là do em tự nghiên cứu, tổng hợp từ các nguồn tài liệu có thẩm quyền và được trích dẫn rõ ràng.

Các kết quả, dữ liệu và phương pháp nghiên cứu trong đồ án này là hoàn toàn trung thực, không sao chép từ bất kỳ nguồn nào mà không được trích dẫn hợp lệ.

Nếu phát hiện bất kỳ sự gian lận nào trong nội dung đồ án, em hoàn toàn chịu trách nhiệm trước Hội đồng chấm đồ án và trước quy định của nhà trường.

Em xin chân thành cảm ơn sự giúp đỡ tận tình của Cô Trần Thu Trang, cũng như sự hỗ trợ của các bạn bè, đồng nghiệp và gia đình trong quá trình thực hiện đồ án này.

Hà Nội, ngày 24 tháng 05 năm 2024

Sinh viên thực hiện

LỜI CẢM ƠN

Để hoàn thành đồ án tốt nghiệp này, em đã nhận được rất nhiều sự giúp đỡ, động viên và hướng dẫn quý báu từ nhiều cá nhân và tổ chức. Em xin bày tỏ lòng biết ơn chân thành đến tất cả những ai đã hỗ trợ và đồng hành cùng em trong suốt quá trình nghiên cứu và thực hiện đề tài này.

Trước hết, em xin gửi lời cảm ơn sâu sắc đến Cô Trần Thu Trang, người đã tận tình hướng dẫn, góp ý và truyền đạt kiến thức quý báu cho tôi trong suốt quá trình thực hiện đồ án. Sự chỉ dẫn và hỗ trợ của Cô đã giúp tôi vượt qua nhiều khó khăn và hoàn thành tốt đồ án này.

Em cũng xin chân thành cảm ơn các Thầy Cô trong khoa Công Nghệ Thông Tin, Trường Đại học Đại Nam đã cung cấp cho em nền tảng kiến thức vững chắc trong suốt những năm học qua. Em xin gửi lời cảm ơn đến bạn bè và đồng nghiệp đã luôn bên cạnh, động viên và chia sẻ những kinh nghiệm quý báu trong suốt quá trình nghiên cứu và thực hiện đồ án.

Cuối cùng, tôi xin gửi lời cảm ơn đến gia đình, những người luôn là nguồn động viên, hỗ trợ tinh thần to lớn, giúp tôi vượt qua mọi khó khăn và áp lực trong suốt thời gian qua.

Em xin chân thành cảm ơn!

Trân trọng,

Sinh viên

MỞ ĐẦU

Trong bối cảnh hiện nay, vấn đề ô nhiễm môi trường và quản lý rác thải đang trở thành một thách thức lớn đối với nhiều quốc gia trên thế giới. Việc phân loại rác thải một cách hiệu quả không chỉ giúp giảm tải cho các bãi rác mà còn đóng vai trò quan trọng trong việc tái chế và bảo vệ môi trường. Tuy nhiên, quá trình phân loại rác thải thường đòi hỏi nhiều công sức và thời gian nếu thực hiện thủ công. Chính vì vậy, việc áp dụng công nghệ vào quá trình này là một hướng đi đầy triển vọng và cần thiết.

Trong bối cảnh đó, các phương pháp học sâu (deep learning) và đặc biệt là các mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) đã chứng tỏ được hiệu quả vượt trội trong việc nhận dạng và phân loại hình ảnh. CNN không chỉ giúp tăng độ chính xác trong việc nhận diện các loại rác thải khác nhau mà còn giảm thiểu thời gian xử lý và sự can thiệp của con người.

Dự án "CNN TRONG PHÂN LOẠI RÁC" được thực hiện với mục tiêu nghiên cứu và phát triển một hệ thống tự động phân loại rác thải dựa trên hình ảnh. Hệ thống này có khả năng nhận diện và phân loại các loại rác thải khác nhau như nhựa, kim loại, giấy, và chất hữu cơ dựa trên các đặc trưng hình ảnh của chúng. Qua đó, góp phần nâng cao hiệu quả trong công tác quản lý và xử lý rác thải, hướng tới một môi trường sống xanh, sạch và bền vững hơn.

Trong báo cáo này, chúng tôi sẽ trình bày chi tiết các bước từ việc thu thập và tiền xử lý dữ liệu, thiết kế và huấn luyện mô hình CNN, cho đến việc đánh giá và triển khai hệ thống. Hy vọng rằng những kết quả đạt được từ dự án này sẽ không chỉ đóng góp vào lĩnh vực nghiên cứu khoa học mà còn mang lại những ứng dụng thực tiễn có giá trị cao trong cuộc sống hàng ngày.

NHẬN XÉT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ký và ghi họ tên

DANH MỤC HÌNH ẢNH

Hình 2.1. Khái quát về cách hoạt động của CNN	26
Hình 2.2. Công thức Softmax Activation.....	30
Hình 2.3. Công thức hàm Categori Cross-Entropy	30
Hình 3.1. Dữ liệu ảnh đã thu thập.....	34
Hình 3.2. Kết quả sau khi xử lý dữ liệu.....	35
Hình 3.3. Kết quả thu được sau khi tải dữ liệu.....	35
Hình 4.1. Sơ đồ model train	43
Hình 4.2. Biểu đồ accuracy của train và validation sau khi huấn luyện	46
Hình 4.3. Biểu đồ loss của train và validation sau khi huấn luyện	47
Hình 4.4. Đoạn mã phân chia dữ liệu để lấy dữ liệu test	48
Hình 4.5. Tải lại dữ liệu.....	48
Hình 4.6. Kết quả ma trận nhầm lẫn trên console	50
Hình 4.7. Kết quả ma trận nhầm lẫn trên trực quan giao diện người dùng	50
Hình 4.8. Khởi tạo biến chạy để chụp từng frame trong video	54
Hình 4.9. Định nghĩa các lớp rác.....	54
Hình 4.10. Tải lại model đã train và chạy lại dữ liệu có rate cao nhất.....	55
Hình 4.11. Mã code quá trình làm việc quay chính	55
Hình 4.12. Kết quả thu được	56

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN VỀ LÝ THUYẾT.....	1
1.1. Giới thiệu về đề tài	1
1.1.1. Tại Sao Phân Loại Rác Là Quan Trọng?	1
1.1.2. Vai Trò Của CNN Trong Phân Loại Rác	1
1.1.3. Mục Tiêu Của Đề Tài.....	2
1.1.4. Phạm Vi Nghiên Cứu	2
1.2. Mục tiêu của đề tài	2
1.2.1. Phát Triển Hệ Thống Phân Loại Rác Tự Động.....	3
1.2.2. Nâng Cao Hiệu Quả Phân Loại Rác Thái	3
1.2.3. Đóng Góp Vào Bảo Vệ Môi Trường	3
1.2.4. Thu Thập và Tiền Xử Lý Dữ Liệu	3
1.2.5. Thiết Kế và Huấn Luyện Mô Hình CNN	4
1.2.6. Đánh Giá và Cải Tiến Mô Hình	4
1.2.7. Ứng Dụng Thực Tế	4
1.3. Đối tượng và phạm vi nghiên cứu	5
1.3.1. Đối Tượng Nghiên Cứu	5
1.3.2. Phạm Vi Nghiên Cứu	5
1.4. Tầm quan trọng của đề tài	6
1.4.1. Giải Quyết Vấn Đề Ô Nhiễm Môi Trường	6
1.4.2. Thúc Đẩy Tái Chế và Tái Sử Dụng.....	7
1.4.3. Ứng Dụng Công Nghệ Hiện Đại trong Quản Lý Rác Thái.....	7
1.4.4. Kinh Tế và Xã Hội	7
1.4.5. Nghiên Cứu và Phát Triển Khoa Học	7
1.4.6. Đáp Ứng Các Mục Tiêu Phát Triển Bền Vững.....	7
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	9
2.1. Khái niệm	9
2.1.1. Khái niệm về học máy.....	9
2.1.2. Khái niệm về CNN.....	12
2.2. Ứng dụng	13
2.2.1. Nhận dạng hình ảnh và video.....	13
2.2.2. Hệ thống gợi ý.....	14
2.2.3. Phân loại hình ảnh.....	15

2.2.4.	Phân đoạn hình ảnh	16
2.2.5.	Phân tích hình ảnh y tế	18
2.2.6.	Xử lý ngôn ngữ tự nhiên	20
2.2.7.	Giao diện não-máy tính.....	21
2.2.8.	Chuỗi thời gian tài chính.....	23
2.3.	Mô hình.....	25
2.3.1.	Convolutional layers	26
2.3.2.	Lớp kích hoạt (Activation Layer).....	27
2.3.3.	Pooling layers.....	27
2.3.4.	Fully connected layers.....	27
2.3.5.	Receptive field	28
2.3.6.	Weights.....	28
2.4.	Quy trình thực hiện thuật toán.....	29
2.4.1.	Tiền xử lý ảnh đầu vào	29
2.4.2.	Convolutional Layer (Lớp tích chập).....	29
2.4.3.	Pooling Layer (Lớp gộp).....	29
2.4.4.	Repeat Convolutional and Pooling Layers.....	29
2.4.5.	Flattening (Làm phẳng).....	29
2.4.6.	Fully Connected Layers (Lớp kết nối đầy đủ)	30
2.4.7.	Output Layer (Lớp đầu ra)	30
2.4.8.	Loss Function (Hàm mất mát).....	30
2.4.9.	Training (Huấn luyện).....	30
2.4.10.	Evaluation (Đánh giá)	30
2.4.11.	Inference (Dự đoán)	31
CHƯƠNG 3:	XỬ LÝ DỮ LIỆU	32
3.1.	Thu thập dữ liệu	32
3.1.1.	Xác định yêu cầu dữ liệu	32
3.1.2.	Nguồn dữ liệu	32
3.1.3.	Tổ chức và gắn nhãn dữ liệu.....	32
3.1.4.	Làm sạch và kiểm tra dữ liệu.....	33
3.1.5.	Lưu trữ và quản lý dữ liệu	33
3.1.6.	Đảm bảo đa dạng dữ liệu	33
3.1.7.	Ghi chép và tài liệu hóa	33
3.2.	Tiền xử lý dữ liệu	33

3.2.1. Xem xét dữ liệu.....	33
3.2.2. Tải dữ liệu.....	35
3.3. Phân chia dữ liệu	35
3.3.1. Lý do phải phân chia dữ liệu	36
3.3.2. Thực hiện phân chia dữ liệu	36
3.4. Định dạng dữ liệu đầu vào	37
3.4.1. Lý do phải định dạng dữ liệu đầu vào	37
3.4.2. Thực hiện định dạng dữ liệu đầu vào	37
CHƯƠNG 4: ỨNG DỤNG CNN TRONG DỰ ĐOÁN RÁC.....	40
4.1. Giới thiệu.....	40
4.2. Xây dựng mô hình CNN	40
4.3. Huấn luyện mô hình	44
4.4. Đánh giá mô hình	46
4.5. Dự đoán ảnh mới.....	47
4.5.1. Chạy trên một ảnh.....	47
4.5.2. Ma trận nhầm lẫn	48
4.6. Thử nghiệm thực tế	54
KẾT LUẬN	57
DANH MỤC TÀI LIỆU THAM KHẢO	61

CHƯƠNG 1: TỔNG QUAN VỀ LÝ THUYẾT

1.1. Giới thiệu về đề tài

Trong những năm gần đây, ô nhiễm môi trường và quản lý rác thải đã trở thành những thách thức nghiêm trọng đối với các thành phố và quốc gia trên toàn thế giới. Khối lượng rác thải sinh hoạt và công nghiệp tăng cao đòi hỏi phải có những phương pháp quản lý hiệu quả và bền vững hơn. Một trong những giải pháp tiên tiến hiện nay là ứng dụng trí tuệ nhân tạo (AI), đặc biệt là các thuật toán học sâu (deep learning), vào quá trình phân loại và xử lý rác thải.

1.1.1. Tại Sao Phân Loại Rác Là Quan Trọng?

- Phân loại rác thải chính xác là bước đầu tiên và quan trọng nhất trong quá trình tái chế và xử lý rác thải. Việc phân loại hiệu quả giúp:
 - Tăng cường tái chế: Tách biệt các vật liệu có thể tái chế (như nhựa, kim loại, giấy) từ rác thải tổng hợp.
 - Giảm thiểu ô nhiễm: Ngăn ngừa các vật liệu nguy hiểm như pin, hóa chất không được xử lý đúng cách gây ô nhiễm môi trường.
 - Tiết kiệm tài nguyên: Giảm thiểu việc khai thác nguyên liệu mới bằng cách tái sử dụng các vật liệu đã qua sử dụng.

1.1.2. Vai Trò Của CNN Trong Phân Loại Rác

- Mạng nơ-ron tích chập (Convolutional Neural Networks - CNN) đã chứng minh được hiệu quả vượt trội trong việc xử lý và phân loại hình ảnh, làm cho chúng trở thành công cụ lý tưởng cho việc phân loại rác thải dựa trên hình ảnh. Một số lý do chính bao gồm:
 - Khả năng tự động trích xuất đặc trưng: CNN có khả năng tự động học và trích xuất các đặc trưng quan trọng từ hình ảnh mà không cần phải thiết kế thủ công.
 - Độ chính xác cao: CNN có khả năng nhận diện và phân loại hình ảnh với độ chính xác cao, ngay cả khi hình ảnh có độ phức tạp cao.
 - Tính khả chuyển: Mô hình CNN được huấn luyện cho một nhiệm vụ cụ thể có thể được điều chỉnh và áp dụng cho các nhiệm vụ liên quan khác với ít thay đổi.

1.1.3. Mục Tiêu Của Đề Tài

- Đề tài "CNN TRONG PHÂN LOẠI RÁC" được thực hiện nhằm đạt các mục tiêu sau:
 - Phát triển hệ thống phân loại rác tự động: Sử dụng CNN để xây dựng một hệ thống có khả năng nhận diện và phân loại các loại rác thải khác nhau từ hình ảnh.
 - Nâng cao hiệu quả phân loại rác thải: So sánh hiệu suất của hệ thống tự động với các phương pháp phân loại thủ công và các hệ thống khác để xác định tính ưu việt của CNN.
 - Đóng góp vào bảo vệ môi trường: Giảm thiểu ô nhiễm và thúc đẩy tái chế thông qua việc phân loại rác thải hiệu quả và chính xác.

1.1.4. Phạm Vi Nghiên Cứu

- Dự án tập trung vào việc:
 - Thu thập và tiền xử lý dữ liệu: Xây dựng bộ dữ liệu hình ảnh các loại rác thải khác nhau để huấn luyện và kiểm thử mô hình.
 - Thiết kế và huấn luyện mô hình CNN: Xây dựng các mô hình CNN và huấn luyện chúng trên tập dữ liệu đã thu thập.
 - Đánh giá và cải tiến mô hình: Đánh giá hiệu suất của mô hình trên tập dữ liệu kiểm thử và thực hiện các cải tiến cần thiết để nâng cao độ chính xác.
 - Ứng dụng thực tế: Triển khai mô hình vào hệ thống thực tế để kiểm thử hiệu suất và tính khả thi.

Đề tài "CNN TRONG PHÂN LOẠI RÁC" không chỉ là một nghiên cứu khoa học có giá trị mà còn mang lại những ứng dụng thực tiễn to lớn. Việc ứng dụng công nghệ CNN trong phân loại rác thải có thể cải thiện đáng kể hiệu quả và độ chính xác của quá trình phân loại, góp phần bảo vệ môi trường và phát triển bền vững. Thông qua dự án này, chúng tôi hy vọng sẽ đóng góp vào việc xây dựng một hệ thống quản lý rác thải thông minh và hiệu quả hơn, mang lại lợi ích thiết thực cho cộng đồng và môi trường.

1.2. Mục tiêu của đề tài

Đề tài "CNN TRONG PHÂN LOẠI RÁC" được xây dựng nhằm giải quyết các vấn đề liên quan đến quản lý và xử lý rác thải thông qua việc áp dụng công nghệ học sâu (deep learning), cụ thể là mạng nơ-ron tích chập (Convolutional Neural Networks - CNN). Các mục tiêu cụ thể của đề tài bao gồm:

1.2.1. Phát Triển Hệ Thống Phân Loại Rác Tự Động

- Mục tiêu chính: Xây dựng một hệ thống tự động có khả năng nhận diện và phân loại các loại rác thải khác nhau từ hình ảnh.
- Mô tả: Sử dụng mạng nơ-ron tích chập (CNN) để phát triển một mô hình phân loại rác thải dựa trên hình ảnh, giúp tự động xác định các loại rác như nhựa, kim loại, giấy và chất hữu cơ.
- Kết quả mong đợi: Hệ thống có khả năng hoạt động trong thời gian thực, nhận diện chính xác các loại rác thải và phân loại chúng một cách tự động.

1.2.2. Nâng Cao Hiệu Quả Phân Loại Rác Thải

- Mục tiêu chính: Cải thiện hiệu suất phân loại rác thải so với các phương pháp truyền thống và các hệ thống khác.
- Mô tả: So sánh hiệu quả của hệ thống phân loại rác tự động dựa trên CNN với các phương pháp phân loại thủ công và các mô hình khác, nhằm xác định mức độ chính xác và hiệu quả của CNN.
- Kết quả mong đợi: Hệ thống tự động phân loại rác thải có độ chính xác cao hơn, thời gian xử lý ngắn hơn và yêu cầu ít sự can thiệp của con người hơn so với các phương pháp truyền thống.

1.2.3. Đóng Góp Vào Bảo Vệ Môi Trường

- Mục tiêu chính: Góp phần giảm thiểu ô nhiễm môi trường và thúc đẩy tái chế thông qua việc phân loại rác thải hiệu quả và chính xác.
- Mô tả: Bằng cách phân loại chính xác các loại rác thải, hệ thống giúp tối ưu hóa quy trình tái chế, giảm thiểu lượng rác thải đổ vào các bãi rác và hạn chế việc xả thải không đúng quy cách.
- Kết quả mong đợi: Tăng tỷ lệ tái chế, giảm thiểu ô nhiễm môi trường và bảo vệ tài nguyên thiên nhiên.

1.2.4. Thu Thập và Tiền Xử Lý Dữ Liệu

- Mục tiêu chính: Xây dựng bộ dữ liệu phong phú và đa dạng về các loại rác thải để huấn luyện và kiểm thử mô hình.
- Mô tả: Thu thập và tiền xử lý hình ảnh rác thải từ nhiều nguồn khác nhau, bao gồm các quy trình xử lý dữ liệu như cắt, xoay, và tăng cường dữ liệu để cải thiện chất lượng và độ đa dạng của tập dữ liệu.

- Kết quả mong đợi: Bộ dữ liệu chất lượng cao với đủ số lượng và độ đa dạng để huấn luyện và kiểm thử mô hình CNN một cách hiệu quả.

1.2.5. Thiết Kế và Huấn Luyện Mô Hình CNN

- Mục tiêu chính: Xây dựng và huấn luyện các mô hình CNN có khả năng nhận diện và phân loại rác thải với độ chính xác cao.
- Mô tả: Thiết kế cấu trúc mạng CNN phù hợp, thực hiện quá trình huấn luyện và tối ưu hóa mô hình trên tập dữ liệu đã thu thập và tiền xử lý.
- Kết quả mong đợi: Mô hình CNN có khả năng trích xuất đặc trưng quan trọng từ hình ảnh rác thải và phân loại chúng một cách chính xác.

1.2.6. Đánh Giá và Cải Tiến Mô Hình

- Mục tiêu chính: Đánh giá hiệu suất của mô hình và thực hiện các cải tiến cần thiết để nâng cao độ chính xác và khả năng phân loại.
- Mô tả: Sử dụng các kỹ thuật đánh giá hiệu suất như ma trận nhầm lẫn, độ chính xác, độ nhạy và độ đặc hiệu để kiểm tra mô hình. Thực hiện các điều chỉnh và cải tiến dựa trên kết quả đánh giá.
- Kết quả mong đợi: Mô hình được tối ưu hóa với hiệu suất cao, độ chính xác cải thiện và khả năng phân loại tốt hơn.

1.2.7. Ứng Dụng Thực Tế

- Mục tiêu chính: Triển khai mô hình vào hệ thống thực tế để kiểm thử hiệu suất và tính khả thi.
- Mô tả: Xây dựng và triển khai hệ thống phân loại rác tự động trong các môi trường thực tế như trung tâm xử lý rác thải, khu dân cư và cơ sở công nghiệp.
- Kết quả mong đợi: Hệ thống hoạt động ổn định trong môi trường thực tế, cung cấp giải pháp phân loại rác thải tự động và hiệu quả.

Các mục tiêu của đề tài "CNN TRONG PHÂN LOẠI RÁC" không chỉ nhằm phát triển một hệ thống tự động và hiệu quả trong việc phân loại rác thải mà còn hướng tới việc đóng góp tích cực vào bảo vệ môi trường và phát triển bền vững. Bằng cách ứng dụng công nghệ học sâu vào thực tiễn, chúng tôi hy vọng sẽ mang lại những giải pháp đột phá và hiệu quả trong quản lý và xử lý rác thải.

1.3. Đối tượng và phạm vi nghiên cứu

1.3.1. Đối Tượng Nghiên Cứu

- Đối tượng nghiên cứu của đề tài "CNN TRONG PHÂN LOẠI RÁC" bao gồm các yếu tố chính sau:
 - Rác Thải:
 - Phân loại rác thải: Nghiên cứu tập trung vào các loại rác thải phổ biến như nhựa, kim loại, giấy, chất hữu cơ và các loại rác khác.
 - Đặc trưng hình ảnh: Khai thác các đặc trưng hình ảnh của rác thải để mô hình CNN có thể phân biệt và nhận diện chính xác.
 - Thuật Toán Mạng Nơ-ron Tích Chập (CNN):
 - Cấu trúc mạng CNN: Nghiên cứu các kiến trúc CNN khác nhau để xác định mô hình phù hợp nhất cho bài toán phân loại rác thải.
 - Kỹ thuật huấn luyện: Sử dụng các kỹ thuật huấn luyện và tối ưu hóa mạng CNN để đạt được hiệu suất phân loại cao.
 - Dữ Liệu Hình Ảnh:
 - Thu thập dữ liệu: Thu thập một tập dữ liệu hình ảnh đa dạng và phong phú về các loại rác thải từ nhiều nguồn khác nhau.
 - Tiền xử lý dữ liệu: Thực hiện các bước tiền xử lý dữ liệu như cắt, xoay, tăng cường dữ liệu để chuẩn bị cho quá trình huấn luyện mô hình.

1.3.2. Phạm Vi Nghiên Cứu

Phạm vi nghiên cứu của đề tài bao gồm các hoạt động chính từ việc thu thập dữ liệu đến triển khai hệ thống phân loại rác thải tự động:

- Thu Thập và Tiền Xử Lý Dữ Liệu:
 - Nguồn dữ liệu: Sưu tầm hình ảnh rác thải từ các nguồn khác nhau như cơ sở dữ liệu công khai, tự chụp ảnh thực tế, và dữ liệu từ các dự án nghiên cứu khác.
 - Phương pháp tiền xử lý: Áp dụng các kỹ thuật tiền xử lý dữ liệu để nâng cao chất lượng ảnh và chuẩn bị cho quá trình huấn luyện mô hình.
- Thiết Kế và Huấn Luyện Mô Hình CNN:
 - Kiến trúc mạng CNN: Khảo sát và lựa chọn kiến trúc mạng CNN phù hợp như LeNet, AlexNet, VGG, ResNet, hoặc các mô hình hiện đại khác.

- Quá trình huấn luyện: Huấn luyện mô hình CNN trên tập dữ liệu rác thải đã qua tiền xử lý, sử dụng các kỹ thuật tối ưu hóa như Adam, SGD, và các hàm loss phù hợp.
- **Đánh Giá Hiệu Suất Mô Hình:**
 - Kỹ thuật đánh giá: Sử dụng các kỹ thuật đánh giá như ma trận nhầm lẫn, độ chính xác (accuracy), độ nhạy (recall), độ đặc hiệu (specificity) và điểm F1 (F1 score) để đo lường hiệu suất của mô hình.
 - Kiểm thử mô hình: Thực hiện kiểm thử mô hình trên tập dữ liệu kiểm thử độc lập để đánh giá khả năng tổng quát hóa của mô hình.
- **Triển Khai và Ứng Dụng Thực Tế:**
 - Phát triển hệ thống: Xây dựng hệ thống phân loại rác thải tự động dựa trên mô hình CNN đã huấn luyện.
 - Kiểm thử thực tế: Triển khai và kiểm thử hệ thống trong các môi trường thực tế như trung tâm xử lý rác thải, khu dân cư, và các cơ sở công nghiệp để đánh giá tính khả thi và hiệu quả của hệ thống.

Đề tài "CNN TRONG PHÂN LOẠI RÁC" nhằm nghiên cứu và phát triển một hệ thống phân loại rác thải tự động bằng cách sử dụng mạng nơ-ron tích chập (CNN). Đối tượng nghiên cứu bao gồm các loại rác thải và các đặc trưng hình ảnh của chúng, cùng với các mô hình và kỹ thuật học sâu. Phạm vi nghiên cứu bao gồm từ việc thu thập và tiền xử lý dữ liệu, thiết kế và huấn luyện mô hình, đến việc đánh giá hiệu suất và triển khai hệ thống trong thực tế. Bằng cách này, đề tài hướng tới mục tiêu tạo ra một giải pháp công nghệ tiên tiến giúp nâng cao hiệu quả quản lý và xử lý rác thải, góp phần bảo vệ môi trường và phát triển bền vững.

1.4. Tầm quan trọng của đề tài

Đề tài "CNN TRONG PHÂN LOẠI RÁC" mang lại nhiều ý nghĩa và tầm quan trọng trong cả lĩnh vực nghiên cứu khoa học và ứng dụng thực tiễn. Những lý do sau đây nhấn mạnh tầm quan trọng của đề tài này:

1.4.1. Giải Quyết Vấn Đề Ô Nhiễm Môi Trường

- **Giảm Thiểu Rác Thải:** Việc phân loại rác hiệu quả giúp giảm thiểu lượng rác thải không phân hủy được đổ vào các bãi rác, từ đó giảm ô nhiễm đất và nước.
- **Bảo Vệ Hệ Sinh Thái:** Phân loại đúng các loại rác thải nguy hại và chất độc hại giúp ngăn ngừa ô nhiễm môi trường và bảo vệ hệ sinh thái.

1.4.2. Thúc Đẩy Tái Chế và Tái Sử Dụng

- Tăng Cường Tái Chế: Phân loại chính xác các vật liệu như nhựa, kim loại, và giấy giúp tối ưu hóa quá trình tái chế, từ đó giảm nhu cầu khai thác tài nguyên thiên nhiên mới.
- Tiết Kiệm Tài Nguyên: Tái sử dụng các vật liệu đã qua sử dụng giúp tiết kiệm tài nguyên và năng lượng, đồng thời giảm chi phí sản xuất và tiêu thụ.

1.4.3. Ứng Dụng Công Nghệ Hiện Đại trong Quản Lý Rác Thải

- Công Nghệ Học Sâu (Deep Learning): Áp dụng các thuật toán học sâu như CNN trong nhận diện và phân loại rác thải chứng minh khả năng của công nghệ AI trong việc giải quyết các vấn đề thực tiễn.
- Tự Động Hóa Quy Trình: Hệ thống phân loại rác tự động giúp giảm thiểu sự can thiệp của con người, tăng cường hiệu quả và độ chính xác của quá trình phân loại.

1.4.4. Kinh Tế và Xã Hội

- Tiết Kiệm Chi Phí: Hệ thống phân loại tự động giúp giảm chi phí lao động và tăng hiệu suất hoạt động của các cơ sở xử lý rác thải.
- Tạo Công Ăn Việc Làm: Phát triển và triển khai các hệ thống công nghệ mới có thể tạo ra các cơ hội việc làm trong lĩnh vực công nghệ và quản lý môi trường.
- Nâng Cao Nhận Thức Cộng Đồng: Thúc đẩy việc phân loại và tái chế rác thải góp phần nâng cao ý thức của cộng đồng về bảo vệ môi trường và phát triển bền vững.

1.4.5. Nghiên Cứu và Phát Triển Khoa Học

- Đóng Góp Vào Lĩnh Vực AI và Học Sâu: Nghiên cứu và triển khai mô hình CNN trong phân loại rác thải góp phần phát triển và hoàn thiện các thuật toán học sâu.
- Mở Rộng Ứng Dụng AI: Thành công của dự án có thể mở ra nhiều ứng dụng mới của công nghệ AI trong các lĩnh vực khác như nông nghiệp, y tế, và quản lý đô thị.

1.4.6. Đáp Ứng Các Mục Tiêu Phát Triển Bền Vững

- Hỗ Trợ Mục Tiêu Phát Triển Bền Vững của Liên Hợp Quốc: Đề tài góp phần vào việc thực hiện các mục tiêu phát triển bền vững (SDGs) của Liên Hợp Quốc, đặc biệt là các mục tiêu liên quan đến bảo vệ môi trường và sử dụng bền vững tài nguyên thiên nhiên.

Tầm quan trọng của đề tài "CNN TRONG PHÂN LOẠI RÁC" không chỉ nằm ở việc ứng dụng công nghệ hiện đại vào giải quyết vấn đề môi trường mà còn ở các lợi ích kinh tế, xã hội và khoa học mà nó mang lại. Việc phát triển và triển khai hệ thống phân loại rác

tự động dựa trên CNN không chỉ giúp bảo vệ môi trường mà còn đóng góp vào sự phát triển bền vững và nâng cao chất lượng cuộc sống của cộng đồng.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Khái niệm

2.1.1. Khái niệm về học máy

Học máy là một lĩnh vực của trí tuệ nhân tạo mà mục tiêu chính là phát triển các phương pháp và thuật toán để cho phép máy tính học từ dữ liệu mà không cần phải được lập trình một cách cụ thể. Thay vào đó, máy tính sẽ sử dụng các thuật toán và mô hình học từ các mẫu dữ liệu đầu vào để nhận biết các mẫu và tạo ra dự đoán hoặc quyết định dựa trên dữ liệu mới mà nó chưa từng gặp trước đó.

- Dữ liệu (Data)
 - Dữ liệu Huấn luyện (Training Data): Tập dữ liệu được sử dụng để huấn luyện mô hình, bao gồm các đặc trưng (features) và nhãn (labels) tương ứng.
 - Dữ liệu Thử nghiệm (Test Data): Tập dữ liệu độc lập được sử dụng để đánh giá hiệu suất của mô hình sau khi huấn luyện.
 - Dữ liệu không Giám sát (Unlabeled Data): Dữ liệu không chứa nhãn, thường được sử dụng trong bài toán học không giám sát để phát hiện cấu trúc ẩn.
- Mô hình (Model)
 - Mô hình Học máy (Machine Learning Model): Một biểu diễn toán học của mối quan hệ giữa các đặc trưng và nhãn trong dữ liệu huấn luyện.
 - Tham số (Parameters): Các tham số của mô hình là các giá trị có thể thay đổi trong quá trình huấn luyện để tối ưu hóa hiệu suất của mô hình.
 - Siêu tham số (Hyperparameters): Các tham số được cấu hình bên ngoài mô hình và không được học từ dữ liệu, ví dụ như tỷ lệ học (learning rate) và số lượng lớp ẩn.
- Thuật toán Học (Learning Algorithm)
 - Học Giám sát (Supervised Learning): Thuật toán học từ cặp dữ liệu đầu vào và nhãn tương ứng.
 - Học Không giám sát (Unsupervised Learning): Thuật toán học từ dữ liệu không có nhãn, thường được sử dụng để phát hiện cấu trúc ẩn.

- Học Tăng cường (Reinforcement Learning): Một loại học máy mà mô hình học thông qua tương tác với một môi trường và nhận phần thưởng dựa trên hành vi của nó.
- Quá trình Huấn luyện (Training Process)
 - Hàm Mất mát (Loss Function): Một phép đo đánh giá sự sai lệch giữa giá trị dự đoán và giá trị thực tế của mô hình.
 - Tối ưu hóa (Optimization): Quá trình điều chỉnh các tham số của mô hình để giảm thiểu hàm mất mát.
 - Hàm Mất mát và Hàm Tối ưu hóa (Loss and Optimization Functions): Hàm mất mát được tối ưu hóa thông qua các thuật toán tối ưu hóa như Gradient Descent và Adam Optimizer.
- Đánh giá Mô hình (Model Evaluation)
 - Chính Xác (Accuracy): Tỷ lệ dự đoán đúng trên tổng số dự đoán.
 - Chính Xác Trên Mỗi Lớp (Precision): Tỷ lệ dự đoán đúng của một lớp nhất định so với tổng số các dự đoán của mô hình cho lớp đó.
 - Độ Phủ (Recall): Tỷ lệ các mẫu dự đoán đúng của một lớp nhất định so với tổng số mẫu thực sự thuộc lớp đó.
 - F1-score: Một sự kết hợp của precision và recall.
 - Ma trận Lỗi (Confusion Matrix): Một bảng được sử dụng để đánh giá hiệu suất của mô hình phân loại.
- Quá khớp và Quá mức (Overfitting and Underfitting)
 - Quá khớp (Overfitting): Mô hình quá mức học các đặc điểm cụ thể của dữ liệu huấn luyện và không thể tổng quát hóa cho dữ liệu mới.
 - Quá mức (Underfitting): Mô hình quá đơn giản để học mối quan hệ giữa các đặc trưng và nhãn trong dữ liệu.
- Xác Định (Inference)
 - Dự đoán (Prediction): Quá trình sử dụng mô hình đã được huấn luyện để tạo ra dự đoán cho dữ liệu mới nằm ngoài tập huấn luyện.
 - Khám phá (Exploration): Trong học không giám sát, quá trình khám phá cấu trúc và mẫu trong dữ liệu không có nhãn.
- Tăng cường (Ensemble)

- Tăng cường (Ensemble): Kỹ thuật kết hợp nhiều mô hình khác nhau để cải thiện hiệu suất dự đoán.
- Học Sâu (Deep Learning)
 - Học Sâu (Deep Learning): Một nhánh của học máy mà mô hình được cấu trúc thành nhiều lớp xếp chồng lên nhau, có khả năng học các đặc trưng phức tạp từ dữ liệu. Các mạng nơ-ron sâu (deep neural networks) là một ví dụ phổ biến của học sâu.
- Mạng Nơ-ron (Neural Networks)
 - Nơ-ron (Neuron): Một đơn vị tính toán cơ bản trong mạng nơ-ron, nhận đầu vào, tính toán trọng số và đầu ra.
 - Mạng Nơ-ron Nới (Feedforward Neural Network): Một loại mạng nơ-ron trong đó thông tin chỉ di chuyển theo một hướng từ lớp đầu vào đến lớp đầu ra.
 - Mạng Nơ-ron Hồi Quy (Recurrent Neural Network): Một loại mạng nơ-ron có khả năng xử lý dữ liệu tuần tự hoặc dữ liệu có mối quan hệ thời gian.
 - Mạng Nơ-ron Tích chập (Convolutional Neural Network): Một loại mạng nơ-ron thường được sử dụng cho các tác vụ liên quan đến xử lý ảnh và video, có khả năng nhận diện các đặc trưng không gian.
- Tính Diễn Giải (Interpretability)
 - Tính Diễn Giải (Interpretability): Khả năng hiểu được cách mà một mô hình đưa ra dự đoán hoặc quyết định của nó. Đối với nhiều ứng dụng thực tế, tính diễn giải là một yếu tố quan trọng.
- Tính Khả Dụng (Scalability)
 - Tính Khả Dụng (Scalability): Khả năng mở rộng mô hình để xử lý lượng dữ liệu lớn hoặc tăng cường hiệu suất mô hình mà không cần phải thay đổi quá nhiều cấu trúc của nó.
- Học Máy Áp Dụng (Applied Machine Learning)
 - Học Máy Áp Dụng (Applied Machine Learning): Lĩnh vực sử dụng các phương pháp và kỹ thuật học máy để giải quyết các vấn đề thực tế và tạo ra các ứng dụng hữu ích.
- Học Máy Tùy Biến (Custom Machine Learning)

- Học Máy Tùy Biến (Custom Machine Learning): Xây dựng các mô hình học máy tùy chỉnh cho nhu cầu cụ thể của một tổ chức hoặc người dùng cá nhân.
- Học Máy Hỏi và Đáp (Question Answering)
 - Học Máy Hỏi và Đáp (Question Answering): Một lĩnh vực của học máy nghiên cứu về việc phát triển các hệ thống tự động trả lời câu hỏi của con người.
- Học máy là một lĩnh vực rộng lớn và đa dạng, có ứng dụng trong nhiều lĩnh vực khác nhau và đang phát triển mạnh mẽ theo thời gian. Để hiểu rõ hơn về học máy, việc nắm vững các khái niệm và phương pháp cơ bản là rất quan trọng.

2.1.2. Khái niệm về CNN

- Convolutional neural network (CNN) là một loại mạng thần kinh chuyển tiếp nguồn cấp dữ liệu chính quy có khả năng tự học kỹ thuật tính năng thông qua tối ưu hóa bộ lọc (hoặc hạt nhân). Độ dốc biến mất và độ dốc bùng nổ, được thấy trong quá trình truyền ngược trong các mạng thần kinh trước đó, được ngăn chặn bằng cách sử dụng trọng số chuẩn hóa trên ít kết nối hơn.
- Ví dụ: đối với mỗi nơ-ron trong lớp được kết nối đầy đủ, sẽ cần 10.000 trọng số để xử lý hình ảnh có kích thước 100×100 pixel. Tuy nhiên, khi áp dụng các hạt nhân tích chập (hoặc tương quan chéo), chỉ cần 25 nơ-ron để xử lý các ô có kích thước 5×5 . Các tính năng ở lớp cao hơn được trích xuất từ các cửa sổ ngữ cảnh rộng hơn so với các tính năng ở lớp thấp hơn.
- CNN còn được gọi là Mạng thần kinh nhân tạo bất biến dịch chuyển hoặc bất biến không gian (SIANN), dựa trên kiến trúc trọng lượng chung của các hạt tích chập hoặc các bộ lọc trượt dọc theo các tính năng đầu vào và cung cấp các phản hồi tương đương với dịch thuật được gọi là bản đồ tính năng. Ngược lại với trực giác, hầu hết các mạng nơ-ron tích chập không bất biến đối với dịch mã, do hoạt động lấy mẫu xuống mà chúng áp dụng cho đầu vào.
- Mạng nơ-ron chuyển tiếp nguồn cấp dữ liệu thường là mạng được kết nối đầy đủ, nghĩa là mỗi nơ-ron trong một lớp được kết nối với tất cả các nơ-ron ở lớp tiếp theo. 'Khả năng kết nối đầy đủ' của các mạng này khiến chúng có xu hướng trang bị quá nhiều dữ liệu. Các cách chính quy hóa hoặc ngăn ngừa trang bị quá mức điển hình bao gồm: xử phạt các tham số trong quá trình đào tạo (chẳng hạn như giảm trọng

lượng) hoặc cắt bớt kết nối (bỏ qua kết nối, bỏ kết nối, v.v.) Các bộ dữ liệu mạnh mẽ cũng làm tăng khả năng CNN sẽ học các nguyên tắc tổng quát đặc trưng cho một tập dữ liệu chứ không phải là thành kiến của một tập hợp dân số kém.

- Mạng tích chập được lấy cảm hứng từ các quá trình sinh học trong đó mô hình kết nối giữa các tế bào thần kinh giống với tổ chức của vỏ não thị giác của động vật. Các tế bào thần kinh vỏ não riêng lẻ chỉ phản ứng với các kích thích trong một vùng hạn chế của trường thị giác được gọi là trường tiếp nhận. Các trường tiếp nhận của các tế bào thần kinh khác nhau chồng lên nhau một phần sao cho chúng bao phủ toàn bộ trường thị giác.
- CNN sử dụng tương đối ít tiền xử lý so với các thuật toán phân loại hình ảnh khác. Điều này có nghĩa là mạng học cách tối ưu hóa các bộ lọc (hoặc hạt nhân) thông qua việc học tự động, trong khi ở các thuật toán truyền thống, các bộ lọc này được thiết kế thủ công. Sự độc lập này khỏi kiến thức trước đó và sự can thiệp của con người vào việc trích xuất đặc điểm là một lợi thế lớn.

2.2. Ứng dụng

2.2.1. Nhận dạng hình ảnh và video

- Nhận diện đối tượng (Object Recognition): CNN được sử dụng để xác định và phân loại các đối tượng trong hình ảnh. Các hệ thống như Google Photos, Facebook, và các ứng dụng nhận diện hình ảnh khác sử dụng CNN để tự động gắn thẻ và phân loại ảnh.
- Phát hiện đối tượng (Object Detection): Không chỉ nhận diện mà còn xác định vị trí chính xác của các đối tượng trong hình ảnh. Các mô hình như YOLO (You Only Look Once) và SSD (Single Shot MultiBox Detector) là những ví dụ điển hình sử dụng CNN cho nhiệm vụ này.
- Nhận diện khuôn mặt (Face Recognition): CNN được sử dụng rộng rãi trong các hệ thống nhận diện khuôn mặt, chẳng hạn như trong an ninh và giám sát, mở khóa điện thoại thông minh, và trong các ứng dụng mạng xã hội.
- Phân đoạn hình ảnh (Image Segmentation): Các mô hình CNN như U-Net và Mask R-CNN phân đoạn hình ảnh thành các vùng khác nhau, giúp xác định các vùng của một đối tượng cụ thể trong hình ảnh. Điều này rất hữu ích trong y tế (phân đoạn các vùng bệnh trong hình ảnh y khoa) và trong công nghệ xe tự lái.

- Chuyển đổi phong cách (Style Transfer): CNN có thể áp dụng phong cách của một bức tranh hoặc hình ảnh nghệ thuật lên một bức ảnh khác, tạo ra các hình ảnh mới mang phong cách nghệ thuật của bức tranh ban đầu.
- Siêu phân giải hình ảnh (Image Super-Resolution): CNN có thể được sử dụng để nâng cao độ phân giải của hình ảnh, tạo ra các hình ảnh chi tiết hơn từ các hình ảnh có độ phân giải thấp. Các mô hình như SRCNN (Super-Resolution Convolutional Neural Network) là một ví dụ điển hình.
- Video Analysis: CNN không chỉ áp dụng cho hình ảnh mà còn cho video. Chúng có thể được sử dụng để phân loại video, nhận diện hành động trong video, và phát hiện sự kiện quan trọng. Điều này rất hữu ích trong giám sát an ninh, phân tích thể thao và các ứng dụng giải trí.
- Tạo hình ảnh mới (Image Generation): CNN, đặc biệt khi kết hợp với Generative Adversarial Networks (GANs), có thể tạo ra các hình ảnh mới dựa trên các dữ liệu huấn luyện ban đầu. Điều này được sử dụng trong việc tạo ra các hình ảnh giả lập, thiết kế đồ họa, và nghệ thuật số.
- Ứng dụng trong y tế (Medical Imaging): CNN được sử dụng để phân tích các hình ảnh y khoa như X-quang, MRI, và CT scan. Chúng giúp phát hiện sớm các bệnh lý như ung thư, bệnh tim, và các vấn đề về thần kinh bằng cách nhận diện các dấu hiệu bất thường trong hình ảnh y khoa.

2.2.2. Hệ thống gợi ý

- Gợi ý nội dung dựa trên hình ảnh (Image-based Content Recommendation): CNN có thể phân tích nội dung hình ảnh để gợi ý các sản phẩm tương tự. Ví dụ, trong các trang thương mại điện tử như Amazon hay eBay, CNN được sử dụng để phân tích hình ảnh sản phẩm và đề xuất các sản phẩm tương tự hoặc liên quan dựa trên đặc điểm hình ảnh.
- Gợi ý phim và chương trình truyền hình (Movie and TV Show Recommendations): Các dịch vụ như Netflix có thể sử dụng CNN để phân tích poster phim và các hình ảnh liên quan để gợi ý các bộ phim và chương trình truyền hình có nội dung hoặc phong cách tương tự.
- Gợi ý âm nhạc (Music Recommendation): CNN có thể phân tích bìa album và hình ảnh nghệ sĩ để gợi ý các bản nhạc hoặc nghệ sĩ tương tự. Ngoài ra, CNN cũng có

thể được sử dụng để phân tích sóng âm và đặc điểm âm thanh trong các bản nhạc để đưa ra gợi ý chính xác hơn.

- Gợi ý thời trang (Fashion Recommendation): Trong ngành thời trang, CNN được sử dụng để phân tích hình ảnh quần áo, phụ kiện để gợi ý các sản phẩm phù hợp với phong cách của người dùng. Các ứng dụng như Polyvore hay các trang web mua sắm thời trang sử dụng CNN để tạo ra các bộ trang phục gợi ý dựa trên những gì người dùng đã xem hoặc mua.
- Gợi ý nội dung trên mạng xã hội (Social Media Content Recommendation): Trên các nền tảng như Instagram, Pinterest, CNN có thể phân tích hình ảnh và video mà người dùng tương tác để gợi ý nội dung tương tự. Điều này giúp cải thiện trải nghiệm người dùng bằng cách cung cấp nội dung phù hợp với sở thích cá nhân của họ.
- Gợi ý sản phẩm trong bán lẻ (Retail Product Recommendations): Trong bán lẻ, CNN có thể được sử dụng để phân tích hình ảnh sản phẩm và hành vi mua sắm của khách hàng để gợi ý các sản phẩm mà họ có thể quan tâm. Điều này được ứng dụng trong các hệ thống đề xuất của các cửa hàng trực tuyến.
- Gợi ý sách (Book Recommendation): Các hệ thống gợi ý sách có thể sử dụng CNN để phân tích bìa sách và các hình ảnh liên quan để gợi ý các sách tương tự hoặc thuộc cùng thể loại. Điều này có thể thấy trong các dịch vụ như Goodreads.
- Gợi ý nội dung quảng cáo (Ad Recommendation): CNN có thể phân tích hình ảnh và video trong các quảng cáo để gợi ý những quảng cáo tương tự mà người dùng có thể quan tâm. Điều này giúp tăng hiệu quả của các chiến dịch quảng cáo bằng cách cung cấp nội dung phù hợp với sở thích của người dùng.

2.2.3. Phân loại hình ảnh

- Phân loại vật thể trong hình ảnh (Object Classification): CNNs được sử dụng để phân loại các đối tượng trong hình ảnh. Ví dụ, các ứng dụng di động như Google Lens có thể nhận diện và phân loại các đối tượng như động vật, đồ vật, địa điểm, và thực phẩm từ hình ảnh chụp.
- Phân loại bệnh trong hình ảnh y khoa (Medical Image Classification): CNNs được ứng dụng rộng rãi trong y tế để phân loại các hình ảnh y khoa như X-quang, MRI, và CT scan. Chúng có thể giúp phát hiện các bệnh như ung thư, bệnh tim, và các vấn đề về thần kinh bằng cách phân loại hình ảnh theo các loại bệnh lý cụ thể.

- Phân loại các loại thực phẩm (Food Classification): CNNs được sử dụng trong các ứng dụng nhận diện thực phẩm để phân loại các loại thực phẩm từ hình ảnh, giúp đếm calo và theo dõi chế độ ăn uống. Các ứng dụng như Calorie Mama hoặc FoodAI sử dụng công nghệ này để nhận diện và phân loại thực phẩm từ ảnh chụp bữa ăn.
- Phân loại động vật và thực vật (Animal and Plant Classification): Trong lĩnh vực sinh học, CNNs được sử dụng để phân loại các loài động vật và thực vật từ hình ảnh. Các ứng dụng như PlantSnap hay iNaturalist giúp người dùng nhận diện và phân loại các loài cây cỏ và động vật dựa trên hình ảnh chụp từ tự nhiên.
- Phân loại ký tự viết tay (Handwritten Character Recognition): CNNs được sử dụng trong các hệ thống nhận diện chữ viết tay như OCR (Optical Character Recognition). Chúng có thể phân loại và nhận diện các ký tự viết tay, giúp chuyển đổi văn bản viết tay thành văn bản số hóa.
- Phân loại các loại trang phục (Fashion Classification): CNNs được sử dụng trong các ứng dụng thương mại điện tử và thời trang để phân loại các loại trang phục và phụ kiện từ hình ảnh. Ví dụ, các ứng dụng như StyleSnap của Amazon giúp người dùng tìm kiếm các sản phẩm thời trang dựa trên hình ảnh.
- Phân loại giống chó (Dog Breed Classification): CNNs được sử dụng để phân loại giống chó từ hình ảnh. Các ứng dụng như Dog Scanner hoặc Fetch! sử dụng công nghệ này để nhận diện và phân loại giống chó từ ảnh chụp.
- Phân loại các biển báo giao thông (Traffic Sign Classification): Trong các hệ thống hỗ trợ lái xe tự động, CNNs được sử dụng để phân loại các biển báo giao thông từ hình ảnh, giúp hệ thống xe tự lái nhận diện và tuân thủ các biển báo giao thông một cách chính xác.
- Phân loại ảnh vệ tinh (Satellite Image Classification): CNNs được ứng dụng trong việc phân loại ảnh vệ tinh để giám sát môi trường, quy hoạch đô thị, và phát hiện thay đổi trên bề mặt Trái Đất. Chúng có thể phân loại các vùng đất, rừng, nước, và các đối tượng khác từ ảnh vệ tinh.
- Phân loại tác phẩm nghệ thuật (Art Classification): CNNs có thể được sử dụng để phân loại các tác phẩm nghệ thuật dựa trên phong cách, họa sĩ, hoặc thời kỳ nghệ thuật. Các ứng dụng này giúp trong việc quản lý bảo tàng và nghiên cứu nghệ thuật.

2.2.4. Phân đoạn hình ảnh

- Phân đoạn y tế (Medical Image Segmentation):

- Phát hiện khối u và tổn thương: CNNs được sử dụng để phân đoạn các khối u trong ảnh MRI, CT và X-quang, giúp phát hiện sớm các bệnh như ung thư.
- Phân đoạn cơ quan: Các công cụ phân đoạn dựa trên CNN như U-Net giúp xác định và phân tách các cơ quan nội tạng trong hình ảnh y khoa, hỗ trợ trong phẫu thuật và chẩn đoán.
- Phân đoạn giao thông (Traffic and Autonomous Vehicles):
 - Phân đoạn đường và làn đường: CNNs giúp phân biệt các làn đường, đường đi và vỉa hè từ hình ảnh và video, hỗ trợ hệ thống lái xe tự động.
 - Phân đoạn phương tiện và người đi bộ: Phân đoạn các đối tượng như xe cộ, người đi bộ giúp các hệ thống xe tự lái nhận diện và tránh chướng ngại vật.
- Phân đoạn hình ảnh vệ tinh (Satellite Image Segmentation):
 - Phân đoạn đất và nước: CNNs được sử dụng để phân tích và phân đoạn các loại đất, rừng, nước từ ảnh vệ tinh, hỗ trợ trong quản lý tài nguyên và theo dõi môi trường.
 - Phân đoạn các khu vực đô thị: Giúp xác định các khu vực xây dựng, đường phố và công trình hạ tầng từ ảnh vệ tinh để quy hoạch đô thị và quản lý đất đai.
- Phân đoạn đối tượng trong ảnh (Object Segmentation in Images):
 - Phân đoạn người và vật nuôi: Các ứng dụng như Photoshop sử dụng CNNs để phân đoạn người hoặc vật nuôi trong ảnh, giúp trong việc chỉnh sửa và tạo hiệu ứng ảnh.
 - Phân đoạn thực phẩm: Giúp tách riêng các thành phần thực phẩm trong hình ảnh, ứng dụng trong các ứng dụng đếm calo và theo dõi chế độ ăn uống.
- Phân đoạn ảnh trong lĩnh vực thời trang (Fashion Segmentation):
 - Phân đoạn trang phục: CNNs giúp phân đoạn và nhận diện các phần của trang phục như áo, quần, giày từ hình ảnh thời trang, hỗ trợ trong việc tìm kiếm và gợi ý sản phẩm.
- Phân đoạn trong nông nghiệp (Agricultural Segmentation):
 - Phân đoạn cây trồng: CNNs được sử dụng để phân đoạn các vùng cây trồng từ ảnh chụp trên không hoặc ảnh vệ tinh, giúp theo dõi sức khỏe cây trồng và quản lý nông nghiệp.

- Phân đoạn bệnh trên lá cây: Phát hiện và phân đoạn các vùng bị bệnh trên lá cây, giúp nông dân phát hiện sớm và điều trị bệnh cho cây trồng.
- Phân đoạn trong thực tế tăng cường và thực tế ảo (AR/VR Segmentation):
 - Phân đoạn môi trường: Giúp tách biệt các đối tượng và bề mặt trong môi trường thực tế tăng cường và thực tế ảo, tạo ra trải nghiệm tương tác chân thực hơn.
- Phân đoạn trong nghệ thuật số (Digital Art Segmentation):
 - Phân đoạn các phần của hình ảnh nghệ thuật: Hỗ trợ các nghệ sĩ trong việc chỉnh sửa, tô màu và tạo hiệu ứng cho các phần cụ thể của tác phẩm nghệ thuật số.
- Phân đoạn trong giám sát an ninh (Security Surveillance Segmentation):
 - Phân đoạn và nhận diện các đối tượng đáng ngờ: Giúp phân tích và phân đoạn các đối tượng như người, xe cộ trong video giám sát, hỗ trợ trong việc phát hiện hành vi đáng ngờ và đảm bảo an ninh.
- Phân đoạn trong công nghiệp (Industrial Segmentation):
 - Phân đoạn các thành phần sản phẩm: Trong sản xuất, CNNs được sử dụng để phân đoạn và kiểm tra các thành phần của sản phẩm, đảm bảo chất lượng và phát hiện lỗi.

2.2.5. *Phân tích hình ảnh y tế*

- Phát hiện và phân loại khối u (Tumor Detection and Classification):
 - Ung thư phổi: CNNs được sử dụng để phát hiện và phân loại khối u trong ảnh CT phổi, giúp chẩn đoán sớm ung thư phổi.
 - Ung thư vú: Phân tích ảnh chụp X-quang vú (mammograms) để phát hiện các khối u và phân loại chúng thành lành tính hoặc ác tính.
- Phân đoạn y khoa (Medical Image Segmentation):
 - Phân đoạn não: CNNs phân đoạn các vùng khác nhau của não trong ảnh MRI, giúp chẩn đoán các bệnh như u não, đột quỵ và bệnh Alzheimer.
 - Phân đoạn cơ quan nội tạng: Phân đoạn các cơ quan như gan, thận, tim từ ảnh CT hoặc MRI, hỗ trợ trong phẫu thuật và điều trị.
- Phân tích ảnh siêu âm (Ultrasound Image Analysis):

- Phân loại và phát hiện bệnh tim: CNNs được sử dụng để phân tích ảnh siêu âm tim (echocardiograms), giúp phát hiện các vấn đề như hẹp van tim, suy tim.
- Phân tích siêu âm thai nhi: Giúp theo dõi sự phát triển của thai nhi và phát hiện các dị tật bẩm sinh.
- Phát hiện và phân tích tổn thương da (Skin Lesion Detection and Analysis):
 - Chẩn đoán ung thư da: CNNs phân tích hình ảnh tổn thương da để phát hiện và phân loại các loại ung thư da như melanoma.
 - Phân loại bệnh da liễu: Giúp phân loại các bệnh da khác nhau như chàm, vẩy nến.
- Phân tích ảnh X-quang (X-ray Image Analysis):
 - Chẩn đoán bệnh phổi: CNNs giúp phát hiện các bệnh như viêm phổi, lao phổi, COVID-19 từ ảnh X-quang ngực.
 - Chẩn đoán gãy xương: Phát hiện các vết nứt và gãy xương trong ảnh X-quang.
- Phân tích ảnh mắt (Ophthalmology Image Analysis):
 - Phát hiện bệnh võng mạc tiểu đường: CNNs phân tích ảnh chụp đáy mắt để phát hiện các dấu hiệu của bệnh võng mạc tiểu đường.
 - Phát hiện bệnh thoái hóa điểm vàng: Giúp phát hiện sớm bệnh thoái hóa điểm vàng (AMD) thông qua phân tích ảnh OCT (Optical Coherence Tomography).
- Phân tích ảnh dạ dày-ruột (Gastrointestinal Image Analysis):
 - Phát hiện polyp đại tràng: CNNs phân tích ảnh chụp từ nội soi đại tràng để phát hiện các polyp có khả năng dẫn đến ung thư.
 - Chẩn đoán các bệnh về dạ dày: Giúp phát hiện viêm loét dạ dày, ung thư dạ dày từ ảnh chụp nội soi.
- Phân tích hình ảnh gan (Liver Image Analysis):
 - Phát hiện và phân loại khối u gan: CNNs phân tích ảnh CT hoặc MRI gan để phát hiện và phân loại các khối u gan.
 - Đánh giá bệnh gan nhiễm mỡ: Giúp phát hiện và đánh giá mức độ gan nhiễm mỡ (NAFLD).
- Phân tích ảnh răng hàm mặt (Dental Image Analysis):

- Chẩn đoán bệnh nha chu: CNNs phân tích ảnh X-quang nha khoa để phát hiện bệnh nha chu, sâu răng, viêm tủy.
- Lập kế hoạch chỉnh nha: Giúp phân tích và lập kế hoạch điều trị trong chỉnh nha.
- Phân tích ảnh MRI tim (Cardiac MRI Analysis):
 - Chẩn đoán bệnh cơ tim: CNNs giúp phát hiện và phân loại các bệnh cơ tim như bệnh cơ tim phì đại, viêm cơ tim.
 - Phân tích chức năng tim: Giúp đo lường và phân tích chức năng của các buồng tim, hỗ trợ trong việc đánh giá và điều trị bệnh tim.

2.2.6. Xử lý ngôn ngữ tự nhiên

- Phân loại văn bản (Text Classification):
 - Phân loại tin tức: CNNs có thể được sử dụng để phân loại các bài báo vào các chủ đề khác nhau như chính trị, thể thao, giải trí, kinh doanh.
 - Phân loại email: Giúp phân loại email thành các thư mục như hộp thư đến, thư rác, quảng cáo.
- Phân tích cảm xúc (Sentiment Analysis):
 - Đánh giá sản phẩm: CNNs phân tích các đánh giá sản phẩm để xác định xem cảm xúc của khách hàng là tích cực, tiêu cực hay trung tính.
 - Phân tích mạng xã hội: Giúp xác định cảm xúc chung của người dùng trên các nền tảng mạng xã hội như Twitter, Facebook về một sự kiện hoặc sản phẩm cụ thể.
- Nhận diện thực thể có tên (Named Entity Recognition - NER):
 - Phân tích văn bản tin tức: CNNs giúp nhận diện các thực thể như tên người, địa điểm, tổ chức trong các bài báo.
 - Phân tích tài liệu pháp lý: Giúp xác định và trích xuất các thực thể quan trọng từ các tài liệu pháp lý, hợp đồng.
- Tóm tắt văn bản (Text Summarization):
 - Tóm tắt tin tức: CNNs có thể tóm tắt các bài báo dài thành các đoạn văn ngắn hơn, chứa đựng các thông tin chính.
 - Tóm tắt tài liệu kỹ thuật: Giúp tóm tắt các báo cáo kỹ thuật, nghiên cứu khoa học để dễ dàng nắm bắt thông tin quan trọng.
- Dịch máy (Machine Translation):

- Dịch ngôn ngữ: CNNs được sử dụng trong các hệ thống dịch máy để dịch văn bản từ ngôn ngữ này sang ngôn ngữ khác, cải thiện độ chính xác và tốc độ dịch.
- Trả lời câu hỏi (Question Answering):
 - Trả lời tự động: CNNs có thể được sử dụng để xây dựng các hệ thống trả lời câu hỏi tự động, tìm kiếm và cung cấp câu trả lời từ một tập dữ liệu lớn.
 - Hỗ trợ khách hàng: Giúp xây dựng các chatbot và trợ lý ảo trả lời các câu hỏi thường gặp của khách hàng.
- Phân tích cú pháp (Syntax Analysis):
 - Phân tích cấu trúc câu: CNNs giúp phân tích và hiểu cấu trúc ngữ pháp của câu, hỗ trợ trong việc xây dựng các ứng dụng ngôn ngữ như kiểm tra ngữ pháp.
 - Phân tích mối quan hệ ngữ nghĩa: Giúp xác định mối quan hệ giữa các từ và cụm từ trong câu, cải thiện khả năng hiểu ngữ cảnh.
- Mô hình hóa ngữ cảnh (Contextual Modeling):
 - Hiểu ngữ cảnh văn bản: CNNs giúp mô hình hóa ngữ cảnh của từ và cụm từ trong văn bản, cải thiện độ chính xác của các tác vụ NLP khác như NER, phân loại văn bản.
 - Xây dựng từ điển ngữ nghĩa: Giúp xây dựng các mô hình từ điển ngữ nghĩa (word embeddings) như Word2Vec, GloVe.
- Tạo văn bản (Text Generation):
 - Viết nội dung tự động: CNNs có thể được sử dụng để tạo ra các đoạn văn bản mới dựa trên dữ liệu đầu vào, ứng dụng trong việc viết báo cáo, nội dung marketing.
 - Sáng tác văn học: Giúp tạo ra các đoạn văn bản sáng tạo, thơ ca dựa trên phong cách và ngữ cảnh cụ thể.
- Nhận diện ngôn ngữ (Language Detection):
 - Phát hiện ngôn ngữ: CNNs giúp xác định ngôn ngữ của một đoạn văn bản, hỗ trợ trong các ứng dụng dịch thuật và xử lý ngôn ngữ đa ngôn ngữ.

2.2.7. Giao diện não-máy tính

- Giải mã tín hiệu điện não đồ (EEG Signal Decoding):

- Phân loại trạng thái não: CNNs có thể phân loại các trạng thái não khác nhau như sự tập trung, thư giãn, hoặc buồn ngủ từ tín hiệu EEG. Điều này có thể được ứng dụng trong các hệ thống giám sát sức khỏe và hiệu suất làm việc.
- Nhận diện cảm xúc: Phân tích tín hiệu EEG để nhận diện cảm xúc của người dùng, hỗ trợ trong các ứng dụng tương tác người-máy và trò chơi điện tử.
- Điều khiển thiết bị và robot:
 - Điều khiển xe lăn: CNNs giúp giải mã tín hiệu EEG để điều khiển xe lăn điện, hỗ trợ người khuyết tật di chuyển một cách độc lập.
 - Điều khiển thiết bị thông minh: Giải mã tín hiệu não để điều khiển các thiết bị thông minh như máy tính, điện thoại, hoặc hệ thống nhà thông minh.
- Phục hồi chức năng và trị liệu:
 - Phục hồi chức năng sau đột quỵ: CNNs có thể phân tích tín hiệu EEG để hỗ trợ phục hồi chức năng cho bệnh nhân sau đột quỵ thông qua các giao diện điều khiển các thiết bị phục hồi chức năng.
 - Điều trị chứng rối loạn tâm thần: Sử dụng CNNs để phân tích và nhận diện các mẫu tín hiệu não liên quan đến các rối loạn tâm thần như trầm cảm, lo âu, hỗ trợ trong việc theo dõi và điều trị.
- Giao diện thần kinh ngoại vi (Peripheral Neural Interfaces):
 - Điều khiển chi giả: CNNs giúp giải mã tín hiệu thần kinh để điều khiển các chi giả, mang lại khả năng điều khiển chính xác và tự nhiên cho người dùng.
 - Phản hồi cảm giác: Phân tích tín hiệu thần kinh để cung cấp phản hồi cảm giác cho các thiết bị chi giả, giúp người dùng có cảm giác về môi trường xung quanh.
- Phân tích và chẩn đoán bệnh lý:
 - Phát hiện động kinh: CNNs phân tích tín hiệu EEG để phát hiện các cơn động kinh, giúp trong việc chẩn đoán và theo dõi bệnh nhân động kinh.
 - Chẩn đoán rối loạn giấc ngủ: Giúp phân tích tín hiệu EEG trong quá trình ngủ để phát hiện các rối loạn giấc ngủ như ngưng thở khi ngủ hoặc chứng mất ngủ.
- Nghiên cứu khoa học thần kinh (Neuroscience Research):

- Phân tích hoạt động não bộ: CNNs hỗ trợ trong việc phân tích các mẫu hoạt động não bộ từ dữ liệu EEG, fMRI, hoặc MEG, giúp nghiên cứu cấu trúc và chức năng của não.
- Mô hình hóa kết nối não: Giúp mô hình hóa và hiểu rõ hơn về kết nối chức năng và cấu trúc trong não bộ.
- Trò chơi và giải trí:
 - Trò chơi điều khiển bằng ý nghĩ: CNNs giúp phát triển các trò chơi mà người dùng có thể điều khiển bằng suy nghĩ, tạo ra trải nghiệm chơi game mới lạ và hấp dẫn.
 - Tương tác thực tế ảo (VR): Sử dụng tín hiệu não để tương tác với môi trường thực tế ảo, nâng cao trải nghiệm người dùng trong VR.
- Giáo dục và đào tạo:
 - Hỗ trợ học tập: CNNs phân tích tín hiệu EEG để hiểu rõ hơn về trạng thái học tập của học sinh, từ đó điều chỉnh phương pháp giảng dạy phù hợp.
 - Huấn luyện kỹ năng: Sử dụng tín hiệu não để theo dõi tiến bộ và điều chỉnh quá trình huấn luyện kỹ năng thể thao hoặc kỹ năng chuyên môn.
- Giao tiếp và tương tác:
 - Giao tiếp không lời: Giúp người dùng giao tiếp thông qua suy nghĩ mà không cần sử dụng giọng nói hoặc cử chỉ, hỗ trợ đặc biệt cho người bị liệt hoặc mắc các chứng bệnh ảnh hưởng đến khả năng giao tiếp.
 - Tương tác với máy tính: Phát triển các hệ thống cho phép người dùng điều khiển máy tính hoặc nhập liệu thông qua tín hiệu não, cải thiện sự tiện lợi và hiệu quả.

2.2.8. Chuỗi thời gian tài chính

- Dự báo giá cổ phiếu (Stock Price Prediction):
 - Dự báo ngắn hạn: CNNs có thể được sử dụng để dự báo giá cổ phiếu trong ngắn hạn dựa trên dữ liệu giá quá khứ và các chỉ số kỹ thuật.
 - Phân tích xu hướng: Giúp nhận diện các xu hướng giá cổ phiếu và đưa ra dự báo về sự thay đổi giá trong tương lai.
- Phân loại và phát hiện tín hiệu giao dịch (Trading Signal Classification and Detection):

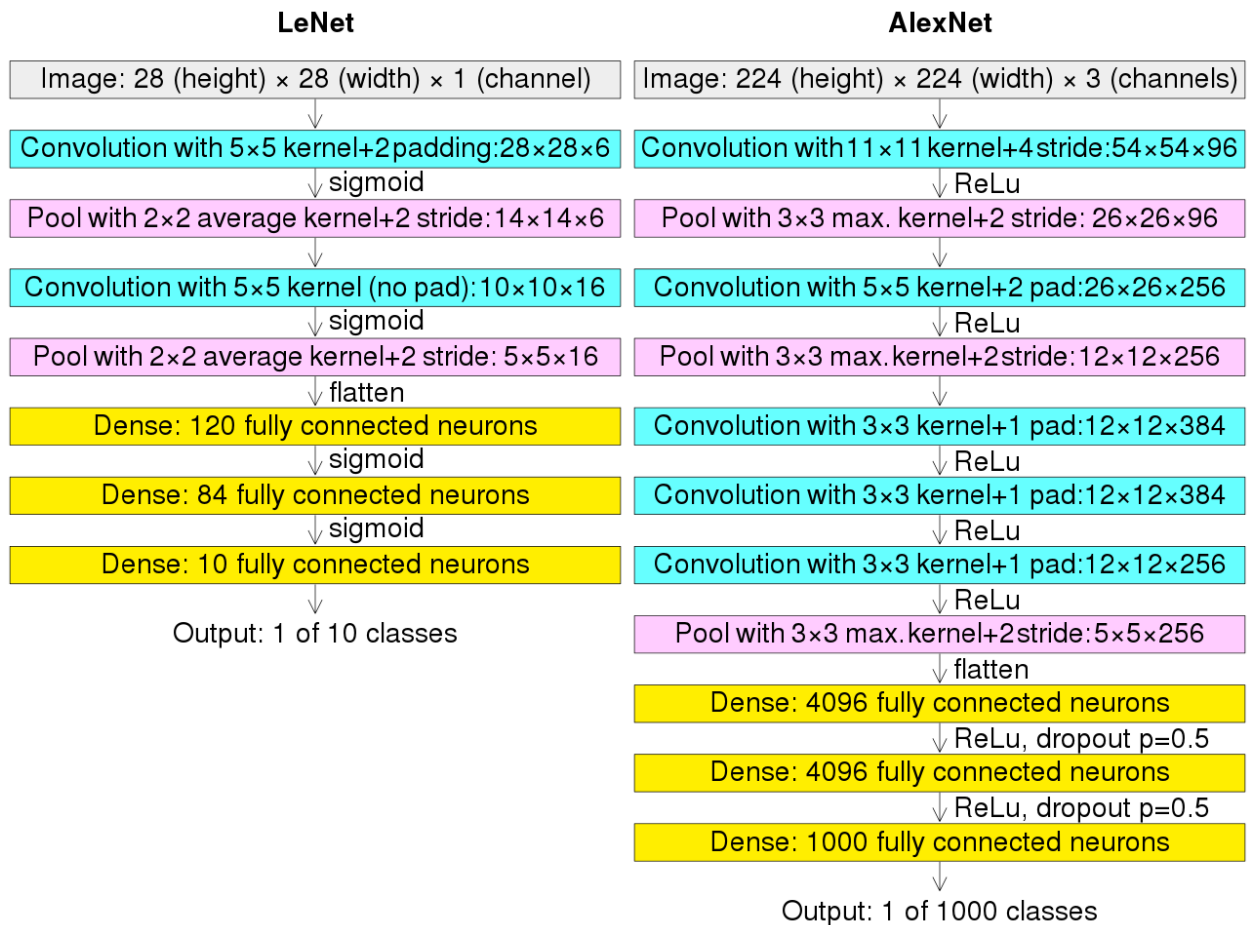
- Xác định điểm mua/bán: CNNs có thể phân tích các chỉ số kỹ thuật và dữ liệu giá để xác định các điểm mua và bán tối ưu.
- Phát hiện các mẫu hình giá: Giúp nhận diện các mẫu hình giá như đầu và vai, cốc và tay cầm, hỗ trợ trong việc đưa ra quyết định giao dịch.
- Quản lý rủi ro (Risk Management):
 - Phân tích rủi ro danh mục đầu tư: Sử dụng CNNs để phân tích và dự báo rủi ro liên quan đến danh mục đầu tư, giúp đưa ra các chiến lược phòng ngừa rủi ro.
 - Phát hiện biến động thị trường: Giúp dự báo và phát hiện các giai đoạn biến động mạnh của thị trường, hỗ trợ trong việc điều chỉnh chiến lược đầu tư.
- Phân tích chuỗi thời gian đa chiều (Multivariate Time Series Analysis):
 - Phân tích mối quan hệ giữa các tài sản: CNNs có thể phân tích mối quan hệ giữa nhiều tài sản tài chính khác nhau, giúp tối ưu hóa danh mục đầu tư.
 - Dự báo chỉ số kinh tế: Giúp dự báo các chỉ số kinh tế như GDP, lạm phát, dựa trên dữ liệu đa chiều từ nhiều nguồn khác nhau.
- Phát hiện gian lận (Fraud Detection):
 - Phát hiện giao dịch gian lận: CNNs có thể được sử dụng để phân tích và phát hiện các giao dịch gian lận trong thời gian thực, hỗ trợ các hệ thống bảo mật tài chính.
 - Phân tích hành vi người dùng: Giúp nhận diện các mẫu hành vi bất thường của người dùng, từ đó phát hiện sớm các hoạt động gian lận.
- Tự động hóa giao dịch (Automated Trading):
 - Chiến lược giao dịch thuật toán: CNNs giúp phát triển các chiến lược giao dịch thuật toán tự động, dựa trên phân tích dữ liệu lịch sử và chỉ số thị trường.
 - Điều chỉnh động (Dynamic Adjustment): Các hệ thống giao dịch tự động có thể sử dụng CNNs để điều chỉnh chiến lược giao dịch theo thời gian thực dựa trên các điều kiện thị trường thay đổi.
- Phân tích tình cảm thị trường (Market Sentiment Analysis):
 - Phân tích tin tức tài chính: CNNs có thể phân tích tin tức tài chính và mạng xã hội để đánh giá tình cảm thị trường, từ đó dự báo xu hướng giá.

- Phân tích báo cáo tài chính: Giúp phân tích và trích xuất thông tin từ các báo cáo tài chính, thông cáo báo chí để đánh giá tình hình tài chính của các công ty.
- Dự báo tỷ giá hối đoái (Forex Prediction):
 - Dự báo biến động tỷ giá: CNNs giúp dự báo sự biến động của tỷ giá hối đoái giữa các cặp tiền tệ, hỗ trợ trong giao dịch ngoại hối.
 - Phân tích yếu tố ảnh hưởng: Phân tích các yếu tố kinh tế và chính trị ảnh hưởng đến tỷ giá hối đoái, giúp đưa ra dự báo chính xác hơn.
- Phân tích và tối ưu hóa chiến lược đầu tư (Investment Strategy Analysis and Optimization):
 - Đánh giá hiệu suất đầu tư: CNNs giúp phân tích và đánh giá hiệu suất của các chiến lược đầu tư khác nhau, từ đó tối ưu hóa chiến lược.
 - Điều chỉnh danh mục đầu tư: Giúp đưa ra các khuyến nghị về điều chỉnh danh mục đầu tư dựa trên phân tích dữ liệu thị trường và dự báo.
- Phân tích chuỗi cung ứng tài chính (Financial Supply Chain Analysis):
 - Dự báo nhu cầu vốn: CNNs có thể dự báo nhu cầu vốn của các doanh nghiệp trong chuỗi cung ứng, hỗ trợ trong quản lý tài chính.
 - Phân tích dòng tiền: Giúp phân tích và tối ưu hóa dòng tiền trong chuỗi cung ứng tài chính, đảm bảo tính thanh khoản và hiệu quả sử dụng vốn.

2.3. Mô hình

Một convolutional neural network bao gồm lớp đầu vào, lớp ẩn và lớp đầu ra. Trong một convolutional neural network, các lớp ẩn bao gồm một hoặc nhiều lớp thực hiện convolutions. Thông thường, điều này bao gồm một lớp thực hiện tích số chấm của hạt convolution với ma trận đầu vào của lớp. Sản phẩm này thường là sản phẩm bên trong Frobenius và chức năng kích hoạt của nó thường là ReLU. Khi nhân convolution trượt dọc theo ma trận đầu vào của lớp, thao tác tích chập sẽ tạo ra một bản đồ đặc trưng, từ đó đóng góp vào đầu vào của lớp tiếp theo. Tiếp theo là các lớp khác như lớp gộp, lớp được kết nối đầy đủ và lớp chuẩn hóa. Ở đây cần lưu ý mức độ gần gũi của một convolutional

neural network với bộ lọc phù hợp.



Hình 2.1. Khái quát về cách hoạt động của CNN

2.3.1. Convolutional layers

- Trong CNN, đầu vào là một tensor có hình dạng:
 - (số lượng đầu vào) × (chiều cao đầu vào) × (chiều rộng đầu vào) × (kênh đầu vào)
- Sau khi đi qua lớp chập, hình ảnh sẽ được trừu tượng hóa thành bản đồ đặc trưng, còn được gọi là bản đồ kích hoạt, có hình dạng:
 - (số lượng đầu vào) × (chiều cao bản đồ đối tượng) × (chiều rộng bản đồ đối tượng) × (kênh bản đồ đối tượng).
- Các lớp tích chập sẽ kết hợp đầu vào và chuyển kết quả của nó sang lớp tiếp theo. Điều này tương tự như phản ứng của tế bào thần kinh trong vỏ não thị giác với một kích thích cụ thể. Mỗi nơ-ron tích chập chỉ xử lý dữ liệu cho trường tiếp nhận của nó.
- Ví dụ chuyển tiếp nguồn cấp dữ liệu mạng nơ-ron tích chập 1D

- Mặc dù các mạng nơ-ron chuyển tiếp được kết nối đầy đủ có thể được sử dụng để tìm hiểu các tính năng và phân loại dữ liệu, nhưng kiến trúc này thường không thực tế đối với các đầu vào lớn hơn (ví dụ: hình ảnh có độ phân giải cao), đòi hỏi số lượng lớn nơ-ron vì mỗi pixel là một tính năng đầu vào có liên quan. Một lớp được kết nối đầy đủ cho hình ảnh có kích thước 100×100 có 10.000 trọng số cho mỗi nơ-ron ở lớp thứ hai. Tích chập làm giảm số lượng tham số miễn phí, cho phép mạng sâu hơn. Ví dụ: sử dụng vùng xếp kề 5×5 , mỗi vùng có cùng trọng số chung, chỉ cần 25 nơ-ron. Việc sử dụng các trọng số được chuẩn hóa trên ít tham số hơn sẽ tránh được các vấn đề về độ dốc biến mất và sự bùng nổ của độ dốc được thấy trong quá trình truyền ngược trong các mạng thần kinh trước đó.
- Để tăng tốc độ xử lý, các lớp chập tiêu chuẩn có thể được thay thế bằng các lớp chập có thể phân tách theo chiều sâu, dựa trên phép tích chập theo chiều sâu, sau đó là tích chập theo điểm. Tích chập theo chiều sâu là tích chập không gian được áp dụng độc lập trên mỗi kênh của tensor đầu vào, trong khi tích chập theo điểm là tích chập tiêu chuẩn được giới hạn trong việc sử dụng 1×1 hạt nhân.

2.3.2. *Lớp kích hoạt (Activation Layer)*

- Chức năng: Áp dụng hàm kích hoạt phi tuyến tính cho các giá trị đầu ra của lớp tích chập. Hàm kích hoạt phổ biến nhất là ReLU (Rectified Linear Unit), giúp mô hình học các đặc trưng phi tuyến tính.
- Thành phần: ReLU: $f(x) = \max(0, x)$

2.3.3. *Pooling layers*

Mạng tích chập có thể bao gồm các lớp tổng hợp cục bộ và/hoặc toàn cầu cùng với các lớp tích chập truyền thống. Các lớp gộp làm giảm kích thước của dữ liệu bằng cách kết hợp đầu ra của các cụm nơ-ron ở một lớp thành một nơ-ron duy nhất ở lớp tiếp theo. Nhóm cục bộ kết hợp các cụm nhỏ, kích thước ổ lát như 2×2 thường được sử dụng. Việc gộp chung tác động lên tất cả các nơ-ron của bản đồ đặc trưng. Có hai loại gộp phổ biến được sử dụng phổ biến: tối đa và trung bình. Nhóm tối đa sử dụng giá trị tối đa của từng cụm nơ-ron cục bộ trong bản đồ đặc trưng, trong khi nhóm trung bình lấy giá trị trung bình.

2.3.4. *Fully connected layers*

Các lớp được kết nối đầy đủ kết nối mọi nơ-ron trong một lớp với mọi nơ-ron trong lớp khác. Nó giống như mạng nơ-ron perceptron đa lớp truyền thống (MLP). Ma trận phẳng đi qua một lớp được kết nối đầy đủ để phân loại hình ảnh.

2.3.5. *Receptive field*

- Trong mạng nơ-ron, mỗi nơ-ron nhận đầu vào từ một số vị trí ở lớp trước. Trong lớp chập, mỗi nơ-ron chỉ nhận đầu vào từ một vùng hạn chế của lớp trước đó gọi là trường tiếp nhận của nơ-ron. Thông thường diện tích là một hình vuông (ví dụ: 5 x 5 nơ-ron). Trong khi đó, ở lớp được kết nối đầy đủ, trường tiếp nhận là toàn bộ lớp trước đó. Do đó, trong mỗi lớp chập, mỗi nơ-ron lấy đầu vào từ một vùng lớn hơn trong đầu vào so với các lớp trước. Điều này là do việc áp dụng lặp đi lặp lại phép tích chập, tính đến giá trị của một pixel cũng như các pixel xung quanh nó. Khi sử dụng các lớp được giãn nở, số lượng pixel trong trường tiếp nhận vẫn không đổi, nhưng trường này sẽ thưa thớt hơn khi kích thước của nó tăng lên khi kết hợp hiệu ứng của một số lớp.
- Để điều khiển kích thước trường tiếp nhận như mong muốn, có một số lựa chọn thay thế cho lớp chập tiêu chuẩn. Ví dụ, tích chập nhĩ hoặc giãn nở mở rộng kích thước trường tiếp nhận mà không tăng số lượng tham số bằng cách xen kẽ các vùng nhìn thấy và vùng mù. Hơn nữa, một lớp chập giãn đơn có thể bao gồm các bộ lọc có nhiều tỷ lệ giãn nở, do đó có kích thước trường tiếp nhận thay đổi.

2.3.6. *Weights*

- Mỗi nơ-ron trong mạng nơ-ron tính toán một giá trị đầu ra bằng cách áp dụng một hàm cụ thể cho các giá trị đầu vào nhận được từ trường tiếp nhận ở lớp trước. Hàm được áp dụng cho các giá trị đầu vào được xác định bởi vector trọng số và độ lệch (thường là số thực). Việc học bao gồm việc điều chỉnh lặp đi lặp lại những thành phần và trọng số này.
- Các vector trọng số và độ lệch được gọi là bộ lọc và biểu thị các đặc điểm cụ thể của đầu vào (ví dụ: một hình dạng cụ thể). Một đặc điểm khác biệt của CNN là nhiều nơ-ron có thể chia sẻ cùng một bộ lọc. Điều này làm giảm đáng kể bộ nhớ vì một độ lệch duy nhất và một vector trọng số duy nhất được sử dụng trên tất cả các trường tiếp nhận chia sẻ bộ lọc đó, trái ngược với mỗi trường tiếp nhận có độ lệch và trọng số vector riêng.

2.4. Quy trình thực hiện thuật toán

Convolutional Neural Network (CNN) là một mô hình học sâu (deep learning) rất phổ biến trong xử lý và phân loại ảnh. Dưới đây là các bước hoạt động chi tiết của CNN trong phân loại ảnh có nhiều categories:

2.4.1. Tiền xử lý ảnh đầu vào

Resize: Điều chỉnh kích thước ảnh về một kích thước cố định (ví dụ: 224x224).

Normalization: Chuẩn hóa các giá trị pixel của ảnh (ví dụ: giá trị pixel từ [0, 255] chuyển sang [0, 1] hoặc [-1, 1]).

2.4.2. Convolutional Layer (Lớp tích chập)

- Convolution Operation: Áp dụng một hoặc nhiều kernel (filter) lên ảnh đầu vào để trích xuất các đặc trưng (features).
- Activation Function: Thường sử dụng hàm ReLU (Rectified Linear Unit) để thêm tính phi tuyến cho mô hình: $\text{ReLU}(x) = \max(0, x)$
- Feature Maps: Kết quả của quá trình tích chập là các feature maps thể hiện các đặc trưng đã được trích xuất.

2.4.3. Pooling Layer (Lớp gộp)

- Max Pooling hoặc Average Pooling: Giảm kích thước của feature maps, giữ lại các thông tin quan trọng và giảm tính toán. Ví dụ, Max Pooling lấy giá trị lớn nhất trong một vùng cửa sổ.
- Downsampling: Giảm độ phân giải của ảnh bằng cách giảm số lượng các tham số và tính toán trong mạng.

2.4.4. Repeat Convolutional and Pooling Layers

- Lặp lại nhiều lần: Thường các mạng CNN có nhiều tầng tích chập và gộp để trích xuất các đặc trưng ngày càng phức tạp từ ảnh.
- Feature Extraction: Mỗi lớp sẽ trích xuất các đặc trưng ở mức độ trừu tượng cao hơn từ ảnh.

2.4.5. Flattening (Làm phẳng)

- Flatten: Chuyển đổi các feature maps cuối cùng thành một vector 1 chiều để làm đầu vào cho các lớp fully connected phía sau.

2.4.6. Fully Connected Layers (Lớp kết nối đầy đủ)

- Dense Layers: Các lớp kết nối đầy đủ (dense layers) có chức năng học các tổ hợp phức tạp của các đặc trưng đã trích xuất.
- Activation Function: Thường sử dụng hàm ReLU hoặc các hàm kích hoạt khác.

2.4.7. Output Layer (Lớp đầu ra)

- Softmax Activation: Trong bài toán phân loại nhiều categories, sử dụng hàm softmax ở lớp đầu ra để chuyển đổi các giá trị thành xác suất:

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_j e^{z_j}}$$

Hình 2.2. Công thức Softmax Activation

- Class Probabilities: Lớp này cung cấp xác suất cho mỗi category, và category có xác suất cao nhất sẽ được chọn làm kết quả phân loại.

2.4.8. Loss Function (Hàm mất mát)

- Categorical Cross-Entropy: Thường sử dụng hàm mất mát cross-entropy cho bài toán phân loại nhiều categories:

$$L = - \sum_i y_i \log(\hat{y}_i)$$

Hình 2.3. Công thức hàm Categorical Cross-Entropy

- Optimization: Sử dụng các thuật toán tối ưu hóa như SGD, Adam để tối ưu hóa hàm mất mát.

2.4.9. Training (Huấn luyện)

- Forward Pass: Truyền dữ liệu từ đầu vào qua các lớp của mạng để tính toán đầu ra.
- Backward Pass (Backpropagation): Tính toán gradient của hàm mất mát với từng tham số của mô hình và cập nhật các tham số này dựa trên gradient.

2.4.10. Evaluation (Đánh giá)

- Accuracy, Precision, Recall, F1-Score: Sử dụng các metrics như độ chính xác, precision, recall, F1-score để đánh giá hiệu suất của mô hình.
- Confusion Matrix: Ma trận nhầm lẫn để đánh giá chi tiết hơn về kết quả phân loại.

2.4.11. Inference (Dự đoán)

- Predicting New Data: Sử dụng mô hình đã huấn luyện để dự đoán category của các ảnh mới.

CNN hoạt động qua nhiều lớp tích chập và gộp để trích xuất các đặc trưng quan trọng từ ảnh. Sau đó, các lớp fully connected và hàm softmax sẽ phân loại các đặc trưng này thành các categories cụ thể. Quá trình huấn luyện sử dụng backpropagation và tối ưu hóa để cải thiện hiệu suất của mô hình.

CHƯƠNG 3: XỬ LÝ DỮ LIỆU

3.1. Thu thập dữ liệu

Thu thập dữ liệu là bước đầu tiên và rất quan trọng trong việc chuẩn bị dữ liệu cho dự án phân loại rác dùng thuật toán CNN. Dưới đây là hướng dẫn chi tiết về các bước thu thập dữ liệu:

3.1.1. Xác định yêu cầu dữ liệu

- Xác định các loại rác cần phân loại: Xác định rõ các danh mục rác cần phân loại (ví dụ: nhựa, kim loại, giấy, hữu cơ, thủy tinh, điện tử, và các loại rác khác).
- Số lượng hình ảnh cần thiết: Xác định số lượng hình ảnh cần thiết cho mỗi loại rác để đảm bảo đủ dữ liệu cho quá trình huấn luyện, kiểm tra, và kiểm thử mô hình.

3.1.2. Nguồn dữ liệu

- Dữ liệu công khai: Tìm kiếm và tải xuống các bộ dữ liệu công khai đã có sẵn trên mạng. Một số nguồn phổ biến bao gồm:
 - o TrashNet: Một bộ dữ liệu phổ biến cho việc phân loại rác.
 - o Kaggle: Nền tảng cung cấp nhiều bộ dữ liệu cho các dự án học máy.
 - o Data.gov: Nền tảng cung cấp nhiều bộ dữ liệu công khai của chính phủ.
- Tự tạo bộ dữ liệu: Nếu không tìm thấy bộ dữ liệu phù hợp, bạn có thể tự tạo bộ dữ liệu bằng cách:
 - Chụp ảnh: Sử dụng máy ảnh hoặc điện thoại để chụp ảnh các loại rác khác nhau. Đảm bảo chụp ảnh từ nhiều góc độ và trong các điều kiện ánh sáng khác nhau.
 - Thu thập từ internet: Tìm kiếm và tải xuống hình ảnh từ các trang web như Google Images, Flickr, hoặc các trang web chia sẻ ảnh khác. Chú ý bản quyền và quyền sử dụng hình ảnh.

3.1.3. Tổ chức và gắn nhãn dữ liệu

- Tổ chức dữ liệu: Sắp xếp hình ảnh vào các thư mục con tương ứng với từng loại rác. Ví dụ, bạn có thể có các thư mục con như /plastic, /metal, /paper, /organic, /glass, /electronic.
- Gắn nhãn dữ liệu: Nếu bạn tự chụp hoặc thu thập hình ảnh từ nhiều nguồn, bạn cần gắn nhãn cho từng hình ảnh bằng cách sử dụng các công cụ như:
 - o LabelImg: Một công cụ mã nguồn mở cho việc gắn nhãn ảnh.
 - o VGG Image Annotator (VIA): Một công cụ trực tuyến để sử dụng để gắn nhãn hình ảnh.

3.1.4. Làm sạch và kiểm tra dữ liệu

- Loại bỏ dữ liệu không hợp lệ: Kiểm tra và loại bỏ các hình ảnh không phù hợp hoặc không thuộc loại rác đã xác định.
- Kiểm tra chất lượng hình ảnh: Đảm bảo các hình ảnh có chất lượng tốt, không bị mờ hoặc quá tối/sáng.

3.1.5. Lưu trữ và quản lý dữ liệu

- Lưu trữ dữ liệu: Lưu trữ hình ảnh và các tệp gắn nhãn trong một cấu trúc thư mục rõ ràng và có hệ thống. Đảm bảo rằng bạn có các bản sao lưu dữ liệu để tránh mất mát dữ liệu.
- Quản lý phiên bản dữ liệu: Sử dụng các công cụ quản lý phiên bản như Git để theo dõi các thay đổi trong dữ liệu, đặc biệt khi bạn làm việc trong một nhóm.

3.1.6. Đảm bảo đa dạng dữ liệu

- Đảm bảo sự đa dạng: Đảm bảo rằng dữ liệu thu thập bao gồm các hình ảnh từ nhiều điều kiện khác nhau (ánh sáng, góc chụp, môi trường, v.v.) để mô hình có thể học tốt hơn và tổng quát hóa tốt hơn khi gặp dữ liệu mới.

3.1.7. Ghi chép và tài liệu hóa

- Ghi chép quá trình: Ghi chép lại quá trình thu thập dữ liệu, bao gồm các nguồn dữ liệu, cách gắn nhãn, và bất kỳ vấn đề nào gặp phải trong quá trình.
- Tài liệu hóa dữ liệu: Tạo tài liệu mô tả chi tiết về dữ liệu, bao gồm các loại rác, số lượng hình ảnh, phương pháp gắn nhãn, và bất kỳ thông tin quan trọng nào khác.

Sau khi hoàn tất các bước thu thập dữ liệu này, bạn sẽ có một bộ dữ liệu chất lượng cao, sẵn sàng cho các bước tiếp theo trong quy trình huấn luyện mô hình CNN để phân loại rác.

3.2. Tiền xử lý dữ liệu

3.2.1. Xem xét dữ liệu

- Trước hết, chúng ta cần xem xét một số hình ảnh mẫu từ tập dữ liệu 'train' để hiểu cấu trúc và nội dung của dữ liệu.
- Tải lên dữ liệu đã lưu để làm việc với dữ liệu. Ta có:

```
def save_data(rf=raw_folder):  
    print("Bắt đầu xử lý ảnh...")  
    pixels = []  
    labels = []  
    for folder in listdir(rf):  
        if folder != '.DS_Store':
```

```

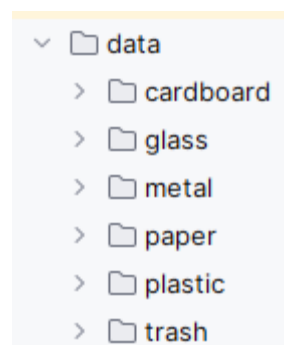
print("Folder=", folder)
for file in listdir(rf + folder):
    if file != '.DS_Store':
        print("File=", file)
        pixels.append(cv2.resize(cv2.imread(rf +
folder + "/" + file), dsize=(128, 128)))
        labels.append(folder)
pixels = np.array(pixels)
labels = np.array(labels)  # .reshape(-1,1)

from sklearn.preprocessing import LabelBinarizer
encoder = LabelBinarizer()
labels = encoder.fit_transform(labels)
print(labels)

file = open('pix/pix.data', 'wb')
# dump information to that file
pickle.dump((pixels, labels), file)
# close the file
file.close()
return

```

- Đọc tất cả các tệp ảnh trong một thư mục và thay đổi kích thước chúng về 128x128.
- Chuyển đổi tên thư mục chứa ảnh thành nhãn.
- Chuyển đổi nhãn thành định dạng nhị phân.
- Lưu trữ dữ liệu ảnh và nhãn đã được xử lý vào một tệp nhị phân để sử dụng sau này trong việc huấn luyện mô hình.
- Ở đây chúng ta tạo ra labels tương ứng với các tập hình ảnh trong folder data. Chúng ta có 6 tập



Hình 3.1. Dữ liệu ảnh đã thu thập

- Sử dụng LabelBinarizer để chuyển hoá các labels đó

```

o from sklearn.preprocessing import LabelBinarizer
  encoder = LabelBinarizer()
  labels = encoder.fit_transform(labels)

```

- Kết quả thu được

```

Pixels (7457, 128, 128, 3)
Labels [[1 0 0 0 0 0]
        [1 0 0 0 0 0]
        [1 0 0 0 0 0]
        ...
        [0 0 0 0 0 1]
        [0 0 0 0 0 1]
        [0 0 0 0 0 1]]

```

Hình 3.2. Kết quả sau khi xử lý dữ liệu

3.2.2. Tải dữ liệu

- Sau khi chúng ta lưu dữ liệu vào đã gắn nhãn vào file pix.data thì bước tiếp theo là cần tải lên dữ liệu đó để làm việc

```

def load_data():
    file = open('pix/pix.data', 'rb')

    # dump information to that file
    (pixels, labels) = pickle.load(file)

    # close the file
    file.close()

    print(pixels.shape)
    print(labels.shape)

    return pixels, labels

X, y = load_data()

```

- Kết quả thu được sau quá trình này sẽ là

```

(7457, 128, 128, 3)
(7457, 6)

```

Hình 3.3. Kết quả thu được sau khi tải dữ liệu

3.3. Phân chia dữ liệu

Sau bước tiền xử lý dữ liệu ta sẽ phải phân chia dữ liệu

3.3.1. Lý do phải phân chia dữ liệu

Phân chia dữ liệu thành tập huấn luyện và tập validation là một phần quan trọng trong quá trình huấn luyện mô hình máy học, bao gồm cả mô hình CNN. Dưới đây là một số lý do chính:

- **Đánh giá hiệu suất:** Phân chia dữ liệu giúp đảm bảo rằng chúng ta có thể đánh giá hiệu suất của mô hình trên dữ liệu mà nó chưa từng thấy trước đó. Việc đánh giá trên dữ liệu mới giúp đảm bảo tính tổng quát của mô hình, và không phải là hiệu suất chỉ đơn thuần trên dữ liệu huấn luyện.
- **Tránh overfitting:** Overfitting xảy ra khi mô hình quá tinh chỉnh để phù hợp chính xác với dữ liệu huấn luyện mà không thể tổng quát hóa được cho dữ liệu mới. Phân chia dữ liệu giúp chúng ta đánh giá và kiểm soát overfitting bằng cách đo hiệu suất của mô hình trên tập validation.
- **Tinh chỉnh tham số:** Phân chia dữ liệu cho phép chúng ta tinh chỉnh các tham số của mô hình, chẳng hạn như tỷ lệ học (learning rate) hoặc số lượng epochs, dựa trên hiệu suất trên tập validation. Điều này giúp cải thiện hiệu suất tổng quát của mô hình.
- **Đánh giá tổng quát:** Bằng cách đánh giá mô hình trên dữ liệu validation, chúng ta có thể đo lường hiệu suất của mô hình trên dữ liệu mà nó chưa từng thấy, giúp đưa ra các quyết định đúng đắn về việc triển khai mô hình trong các tình huống thực tế.

3.3.2. Thực hiện phân chia dữ liệu

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    test_size=0.3, random_state=42)
print("X:")
print(X_train)
print("y:")
print(y_train)
```

- **X:** Mảng hoặc DataFrame chứa các đặc trưng (features) của dữ liệu đầu vào.
- **y:** Mảng hoặc Series chứa các nhãn (labels) tương ứng với dữ liệu đầu vào X.
- **test_size=0.3:** Tham số này chỉ định tỷ lệ phần trăm của dữ liệu được sử dụng làm tập kiểm tra. Ở đây, 0.3 có nghĩa là 30% dữ liệu sẽ được tách ra để làm tập kiểm tra, và 70% còn lại sẽ làm tập huấn luyện.

- `random_state=42`: Tham số này đặt hạt giống (seed) cho bộ tạo số ngẫu nhiên, giúp đảm bảo rằng kết quả phân chia dữ liệu là có thể tái lập (reproducible). Sử dụng giá trị 42 là một thực tiễn phổ biến nhưng giá trị này có thể được thay đổi tùy ý.
- Tập huấn luyện (`X_train, y_train`): Được sử dụng để huấn luyện mô hình học máy. Gồm 70% dữ liệu ban đầu.
- Tập kiểm tra (`X_test, y_test`): Được sử dụng để đánh giá hiệu suất của mô hình sau khi đã huấn luyện. Gồm 30% dữ liệu ban đầu.
- Việc chia tách dữ liệu thành hai tập riêng biệt là một bước quan trọng trong quy trình học máy, giúp đảm bảo rằng mô hình có thể tổng quát hóa tốt trên dữ liệu chưa thấy trước đó và tránh hiện tượng quá khớp (overfitting).

3.4. Định dạng dữ liệu đầu vào

3.4.1. Lý do phải định dạng dữ liệu đầu vào

- Phù hợp với kiến trúc mạng CNN: Mạng CNN yêu cầu đầu vào là các tensor hình ảnh với các kích thước cố định. Định dạng dữ liệu đảm bảo rằng dữ liệu hình ảnh được biến đổi thành đúng định dạng này để có thể được đưa vào mô hình CNN một cách chính xác.
- Tối ưu hóa hiệu suất: Định dạng dữ liệu thích hợp giúp tối ưu hóa hiệu suất của mô hình bằng cách loại bỏ hoặc giảm thiểu các bước tiền xử lý phức tạp trong quá trình huấn luyện. Ví dụ, việc chuẩn hóa giá trị pixel hoặc thay đổi kích thước hình ảnh có thể được thực hiện trước đó, giúp tăng tốc quá trình huấn luyện.
- Đảm bảo đồng nhất: Định dạng dữ liệu giúp đảm bảo rằng tất cả các hình ảnh đầu vào đều có cùng một kích thước và định dạng, giúp cho mô hình CNN hoạt động một cách đồng nhất trên toàn bộ dữ liệu.
- Tăng cường dữ liệu: Một số phương pháp định dạng dữ liệu có thể được sử dụng để tăng cường dữ liệu, bao gồm việc áp dụng các phép biến đổi như xoay, lật, cắt, v.v. Trong khi tiền xử lý, chúng ta có thể thêm các phép biến đổi này để tăng khả năng tổng quát hóa của mô hình và giảm nguy cơ overfitting.
- Hiệu suất tính toán: Việc chuẩn bị dữ liệu trước đó giúp tối ưu hóa việc sử dụng tài nguyên tính toán, giảm thiểu thời gian đào tạo và tăng tốc độ dự đoán của mô hình.

3.4.2. Thực hiện định dạng dữ liệu đầu vào

- Dữ liệu đầu vào được định dạng và chuẩn bị cho quá trình huấn luyện mô hình CNN thông qua các bước tiền xử lý bổ sung như thay đổi kích thước, chuẩn hóa giá trị

pixel, và thêm các phép biến đổi để tăng cường dữ liệu. Điều này được thực hiện thông qua việc sử dụng ImageDataGenerator từ thư viện Keras để tạo ra các đối tượng dữ liệu cho tập huấn luyện và tập validation.

- Định dạng dữ liệu train

```
o aug = ImageDataGenerator(rotation_range=20, zoom_range=0.1,
                           rescale=1. / 255,
                           width_shift_range=0.1,
                           height_shift_range=0.1,
                           horizontal_flip=True,
                           brightness_range=[0.2, 1.5],
                           fill_mode="nearest")
```

- o rotation_range=20: Xoay ngẫu nhiên các hình ảnh trong khoảng ± 20 độ.
- o zoom_range=0.1: Phóng to hoặc thu nhỏ ngẫu nhiên các hình ảnh trong khoảng $\pm 10\%$ so với kích thước ban đầu.
- o rescale=1. / 255: Chia giá trị pixel của hình ảnh cho 255 để chuẩn hóa dữ liệu, đưa giá trị pixel từ khoảng [0, 255] về khoảng [0, 1].
- o width_shift_range=0.1: Dịch chuyển ngẫu nhiên hình ảnh theo chiều ngang trong khoảng $\pm 10\%$ chiều rộng của hình ảnh.
- o height_shift_range=0.1: Dịch chuyển ngẫu nhiên hình ảnh theo chiều dọc trong khoảng $\pm 10\%$ chiều cao của hình ảnh.
- o horizontal_flip=True: Lật ngang ngẫu nhiên hình ảnh.
- o brightness_range=[0.2, 1.5]: Điều chỉnh độ sáng của hình ảnh trong khoảng từ 20% đến 150% độ sáng ban đầu.
- o fill_mode="nearest": Xác định cách điền các pixel trống khi hình ảnh được xoay, dịch chuyển hoặc biến đổi. Ở đây, nearest có nghĩa là sử dụng giá trị pixel gần nhất để điền vào.
- o Tạo ra một bộ biến đổi dữ liệu hình ảnh với nhiều phép biến đổi khác nhau nhằm cải thiện chất lượng dữ liệu huấn luyện và giúp mô hình học tốt hơn từ dữ liệu có sự đa dạng cao hơn.

- Định dạng dữ liệu validation

```
o aug_val = ImageDataGenerator(rescale=1. / 255)
```

- o rescale=1. / 255: Tương tự như trong aug (đối tượng dùng cho tập huấn luyện), tham số này chuẩn hóa giá trị các pixel của hình ảnh bằng cách chia chúng cho 255. Điều này chuyển đổi các giá trị pixel từ khoảng [0, 255] về

khoảng $[0, 1]$. Chuẩn hóa này thường cần thiết để cải thiện hiệu suất của các mô hình học sâu, bởi vì nó giúp tốc độ hội tụ nhanh hơn và làm cho quá trình huấn luyện ổn định hơn.

- Khác với aug dùng cho tập huấn luyện, đối tượng `aug_val` chỉ sử dụng một phép biến đổi duy nhất là `rescale`. Điều này là hợp lý vì mục tiêu của tập kiểm tra hoặc tập kiểm thử là đánh giá chính xác hiệu suất của mô hình trên dữ liệu chưa thấy trước đó, mà không thêm bất kỳ biến đổi ngẫu nhiên nào có thể làm sai lệch kết quả đánh giá. Việc chuẩn hóa (`rescale`) là cần thiết để dữ liệu kiểm tra có cùng định dạng với dữ liệu huấn luyện, đảm bảo tính nhất quán trong quá trình đánh giá mô hình.

CHƯƠNG 4: ỨNG DỤNG CNN TRONG DỰ ĐOÁN RÁC

4.1. Giới thiệu

Trong chương này, chúng ta sẽ tìm hiểu cách sử dụng mạng neural tích chập (CNN) để dự đoán loại của các hình ảnh. CNN là một công cụ mạnh mẽ trong lĩnh vực thị giác máy tính và thường được sử dụng trong các ứng dụng như nhận diện đối tượng và phân loại ảnh.

4.2. Xây dựng mô hình CNN

- Trước tiên, chúng ta cần xây dựng một mô hình CNN bằng cách sử dụng thư viện Keras. Mô hình sẽ bao gồm một loạt các lớp tích chập và lớp kích hoạt để học các đặc trưng từ hình ảnh đầu vào. Sau đó, chúng ta sử dụng lớp max pooling để giảm kích thước của đặc trưng và lớp kết nối đầy đủ để dự đoán lớp của hình ảnh.

```
def get_model():
    model_vgg16_conv = VGG16(weights='imagenet', include_top=False)
    # Dong bang cac layer
    for layer in model_vgg16_conv.layers:
        layer.trainable = False
    # Tao model
    input = Input(shape=(128, 128, 3), name='image_input')
    output_vgg16_conv = model_vgg16_conv(input)
    # Them cac layer FC va Dropout
    x = Flatten(name='flatten')(output_vgg16_conv)
    x = Dense(4096, activation='relu', name='fc1')(x)
    x = Dropout(0.5)(x)
    x = Dense(4096, activation='relu', name='fc2')(x)
    x = Dropout(0.5)(x)
    x = Dense(y.shape[-1], activation='softmax', name='predictions')(x)

    # Compile
    my_model = Model(inputs=input, outputs=x)
    my_model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])

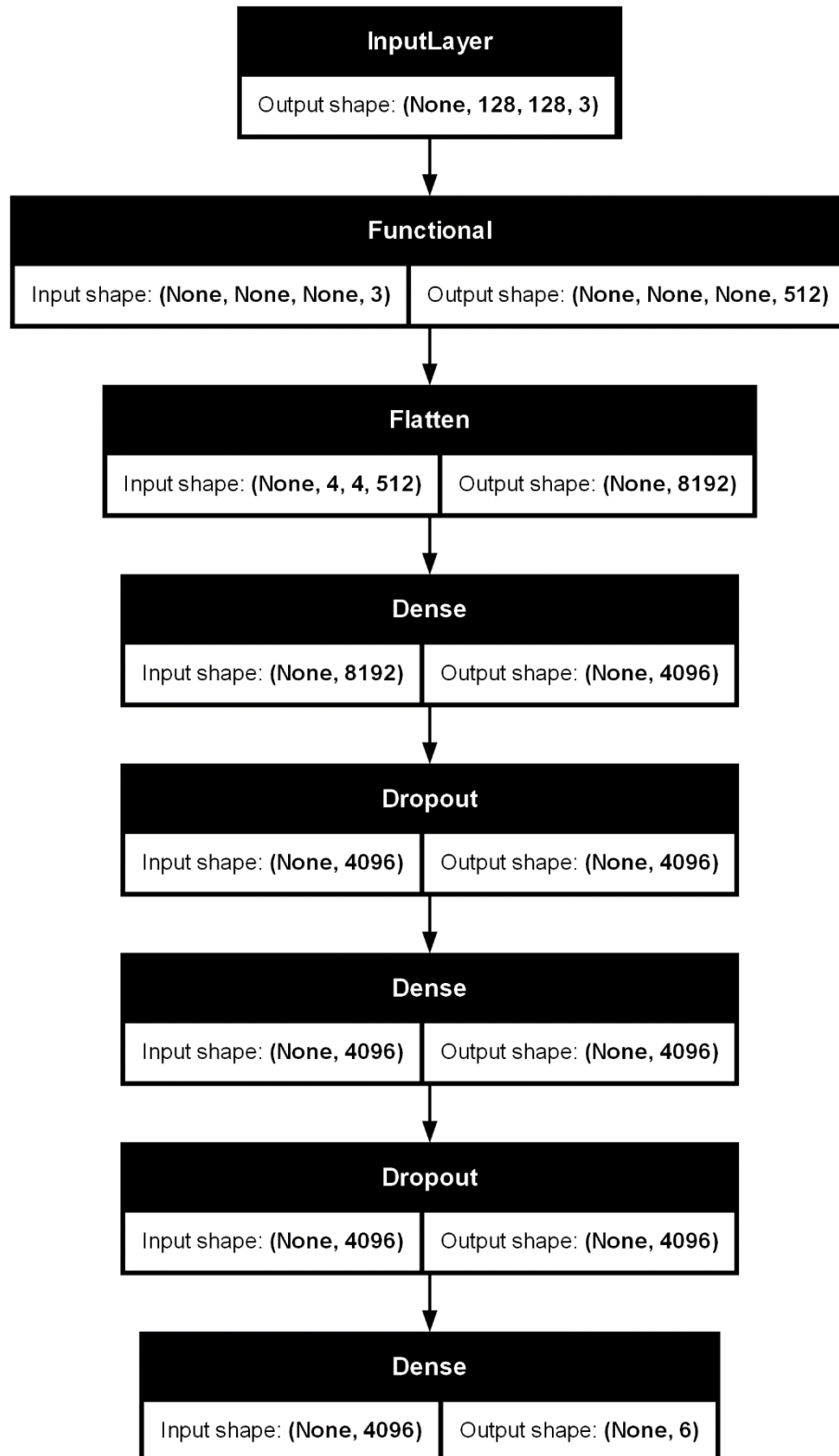
    return my_model
```

- Xây dựng và biên dịch một mô hình học sâu sử dụng mạng VGG16 làm phần mạng cơ sở (base model) và thêm các lớp Fully Connected (FC) cùng với Dropout cho việc phân loại hình ảnh. Chi tiết từng bước như sau:

- Khởi tạo Mô hình VGG16 không bao gồm các lớp Fully Connected (include_top=False)
 - `model_vgg16_conv = VGG16(weights='imagenet', include_top=False)`
 - `weights='imagenet'`: Sử dụng trọng số đã được huấn luyện trên tập dữ liệu ImageNet.
 - `include_top=False`: Loại bỏ các lớp Fully Connected ở phần đỉnh của mô hình VGG16, chỉ giữ lại các lớp convolutional.
- Đóng băng các lớp của mô hình VGG16
 - `for layer in model_vgg16_conv.layers:`
 `layer.trainable = False`
 - Đặt thuộc tính trainable của tất cả các lớp trong mô hình VGG16 thành False, nghĩa là các lớp này sẽ không được cập nhật trọng số trong quá trình huấn luyện. Điều này giúp giữ nguyên các đặc trưng đã học được từ ImageNet.
- Tạo mô hình với đầu vào mới
 - `input = Input(shape=(128, 128, 3), name='image_input')`
 `output_vgg16_conv = model_vgg16_conv(input)`
 - `Input(shape=(128, 128, 3))`: Định nghĩa đầu vào của mô hình là các hình ảnh kích thước 128x128 với 3 kênh màu (RGB).
 - `output_vgg16_conv`: Đầu ra từ phần mạng VGG16 với đầu vào mới.
- Thêm các lớp Fully Connected và Dropout
 - `x = Flatten(name='flatten')(output_vgg16_conv)`
 `x = Dense(4096, activation='relu', name='fc1')(x)`
 `x = Dropout(0.5)(x)`
 `x = Dense(4096, activation='relu', name='fc2')(x)`
 `x = Dropout(0.5)(x)`
 `x = Dense(y.shape[-1], activation='softmax', name='predictions')(x)`
 - `Flatten(name='flatten')`: Làm phẳng đầu ra của VGG16 để chuyển đổi từ tensor nhiều chiều thành vector.
 - `Dense(4096, activation='relu', name='fc1')`: Thêm lớp Fully Connected với 4096 đơn vị và hàm kích hoạt ReLU.
 - `Dropout(0.5)`: Thêm lớp Dropout với tỉ lệ 0.5 để giảm thiểu hiện tượng overfitting.

- `Dense(4096, activation='relu', name='fc2')`: Thêm một lớp Fully Connected thứ hai với 4096 đơn vị và hàm kích hoạt ReLU.
- `Dropout(0.5)`: Thêm một lớp Dropout thứ hai với tỉ lệ 0.5.
- `Dense(y.shape[-1], activation='softmax', name='predictions')`: Lớp đầu ra với số đơn vị bằng số lượng lớp (labels) trong y và hàm kích hoạt softmax cho bài toán phân loại đa lớp.
- Biên dịch mô hình
 - `my_model = Model(inputs=input, outputs=x)`
`my_model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])`
 - `Model(inputs=input, outputs=x)`: Tạo một mô hình mới với đầu vào input và đầu ra x.
 - `compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])`: Biên dịch mô hình với:
 - `loss='categorical_crossentropy'`: Sử dụng hàm mất mát `categorical_crossentropy` cho bài toán phân loại đa lớp.
 - `optimizer='adam'`: Sử dụng trình tối ưu hóa Adam.
 - `metrics=['accuracy']`: Đo lường hiệu suất mô hình bằng độ chính xác (accuracy).

- Sơ đồ model



Hình 4.1. Sơ đồ model train

4.3. Huấn luyện mô hình

- Tiếp theo, chúng ta sẽ huấn luyện mô hình bằng cách sử dụng dữ liệu từ tập huấn luyện. Quá trình huấn luyện này sẽ cập nhật trọng số của mô hình để tối ưu hóa hiệu suất dự đoán.

```
vggmodel = get_model()

filepath = "weights-{epoch:02d}-{val_accuracy:.2f}.keras"
checkpoint = ModelCheckpoint(filepath, monitor='val_accuracy',
                             verbose=1, save_best_only=True, mode='max')
callbacks_list = [checkpoint]
vggHist = vggmodel.fit(aug.flow(X_train, y_train, batch_size=64),
                      epochs=50, # steps_per_epoch=len(X_train)//64,
                      validation_data=aug_val.flow(X_test, y_test,
                                                    batch_size=64),
                      callbacks=callbacks_list)
vggmodel.save("vggmodel.h5")
```

- Mô hình VGG16:
 - o Các lớp tích chập (Convolutional layers):
 - 13 lớp tích chập sử dụng bộ lọc (kernel) 3x3 với stride và padding là 1.
 - Sau mỗi lớp tích chập là một lớp kích hoạt phi tuyến (ReLU).
 - o Các lớp gộp (Pooling layers):
 - 5 lớp gộp cực đại (max-pooling) 2x2 với stride là 2.
 - Lớp gộp giúp giảm kích thước không gian của đặc trưng mà vẫn giữ lại thông tin quan trọng.
 - o Các lớp kết nối đầy đủ (Fully connected layers): 2 lớp kết nối đầy đủ với 4096 nơ-ron mỗi lớp và một lớp kết nối đầy đủ với 1000 nơ-ron (tương ứng với số lớp trong tập dữ liệu ImageNet).
 - o Lớp Softmax: Lớp đầu ra sử dụng hàm softmax để phân loại đầu ra thành 1000 lớp.
- Sử dụng mô hình VGG16 đã được tùy chỉnh để huấn luyện với dữ liệu hình ảnh, đồng thời lưu lại trọng số của mô hình khi hiệu suất trên tập kiểm tra (validation set) đạt tốt nhất
- Lấy mô hình VGG16 tùy chỉnh

- `vggmodel = get_model()`
- `get_model()`: Hàm này giả định trả về mô hình VGG16 đã được tùy chỉnh, tương tự như mô hình đã được tạo ra trong đoạn mã trước đó.
- Thiết lập `ModelCheckpoint`
 - `filepath = "weights-{epoch:02d}-{val_accuracy:.2f}.keras"`
`checkpoint = ModelCheckpoint(filepath, monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')`
`callbacks_list = [checkpoint]`
 - `filepath`: Chuỗi định dạng để lưu trữ tên tệp trọng số mô hình. `{epoch:02d}` và `{val_accuracy:.2f}` là các định dạng chuỗi để thêm số epoch và độ chính xác trên tập kiểm tra vào tên tệp.
 - `ModelCheckpoint`: Callback để lưu lại trọng số mô hình khi giá trị `val_accuracy` đạt tốt nhất.
 - `monitor='val_accuracy'`: Giám sát độ chính xác trên tập kiểm tra.
 - `verbose=1`: In ra các thông tin trong quá trình lưu mô hình.
 - `save_best_only=True`: Chỉ lưu mô hình khi giá trị `val_accuracy` tốt nhất.
 - `mode='max'`: Lưu mô hình khi `val_accuracy` đạt giá trị lớn nhất.
- Huấn luyện mô hình
 - `vggHist = vggmodel.fit(aug.flow(X_train, y_train, batch_size=64), epochs=50, # steps_per_epoch=len(X_train)//64, validation_data=aug_val.flow(X_test, y_test, batch_size=64), callbacks=callbacks_list)`
 - `aug.flow(X_train, y_train, batch_size=64)`: Sử dụng `ImageDataGenerator` để sinh dữ liệu huấn luyện với batch size là 64.
 - `epochs=50`: Số lượng epoch để huấn luyện mô hình là 50.
 - `validation_data=aug_val.flow(X_test, y_test, batch_size=64)`: Sử dụng `ImageDataGenerator` để sinh dữ liệu kiểm tra với batch size là 64.
 - `callbacks=callbacks_list`: Sử dụng danh sách callbacks bao gồm `ModelCheckpoint` để lưu lại mô hình khi hiệu suất tốt nhất.

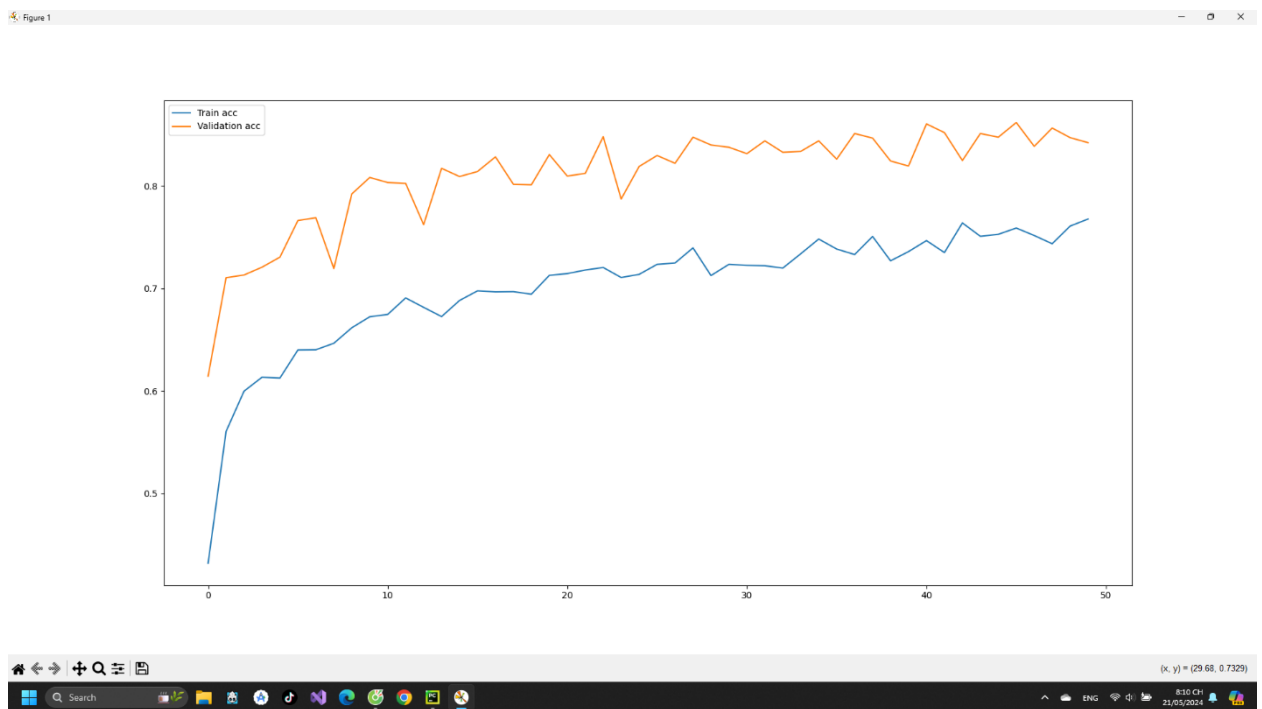
- Lưu mô hình: `save("vggmodel.h5")`: Lưu toàn bộ kiến trúc mô hình và trọng số vào tệp `vggmodel.h5`.

4.4. Đánh giá mô hình

- Sau khi mô hình đã được huấn luyện, chúng ta sẽ đánh giá hiệu suất của nó bằng cách sử dụng dữ liệu từ tập validation. Quá trình này sẽ đánh giá khả năng tổng quát hóa của mô hình trên dữ liệu mà nó chưa từng thấy.

```
plt.figure(figsize=(10, 60))
plt.plot(vggHist.history["accuracy"], label="Train acc")
plt.plot(vggHist.history["val_accuracy"], label="Validation acc")
plt.legend()
plt.show()
```

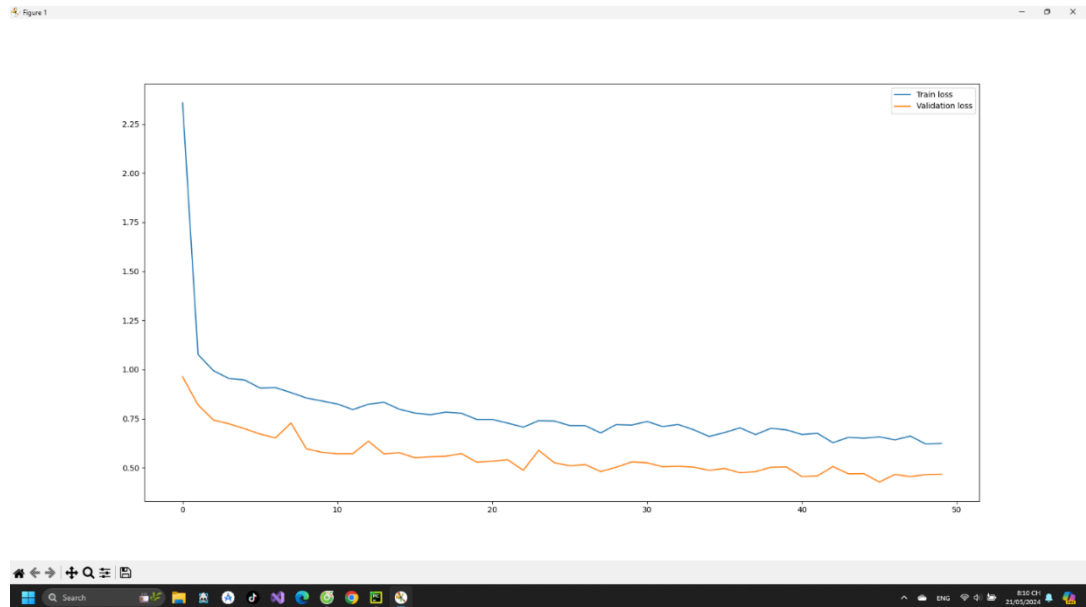
```
plt.figure(figsize=(10, 6))
plt.plot(vggHist.history['loss'], label="Train loss")
plt.plot(vggHist.history['val_loss'], label="Validation loss")
plt.legend()
plt.show()
```



Hình 4.2. Biểu đồ accuracy của train và validation sau khi huấn luyện

- Qua biểu đồ accuracy ta thấy
 - o Cả tỉ lệ train và tỉ lệ validation đều tăng
 - o Tỉ lệ train tăng từ 0% đến 75%
 - o Tỉ lệ validation tăng từ 0 % đến 86%

- Tỷ lệ tăng trưởng qua từng giai đoạn của train ổn định hơn so với tỷ lệ tăng trưởng của validation
- Tỷ lệ của train trong cả quá trình mặc dù biến động lớn nhưng cao nhất chỉ có 75% còn tỷ lệ validation mặc dù biến động nhưng kết quả cuối cùng vẫn hơn tỷ lệ train 11%



Hình 4.3. Biểu đồ loss của train và validation sau khi huấn luyện

- Qua biểu đồ loss ta thấy
 - Cả tỷ lệ train và tỷ lệ validation đều giảm
 - Tỷ lệ train giảm từ 250% đến 52%
 - Tỷ lệ validation giảm từ 100% đến 46%
 - Tỷ lệ giảm đi qua từng giai đoạn của train ổn định hơn so với tỷ lệ giảm đi của validation
 - Tỷ lệ của train trong cả quá trình mặc dù biến động lớn nhưng thấp nhất là 52% còn tỷ lệ validation mặc dù biến động nhưng kết quả cuối cùng vẫn nhỏ hơn tỷ lệ train 8%

4.5. Dự đoán ảnh mới

4.5.1. Chạy trên một ảnh

- Cuối cùng, chúng ta sẽ sử dụng mô hình đã được huấn luyện để dự đoán loại của các hình ảnh mới. Đầu tiên, chúng ta cần chuẩn bị dữ liệu của các hình ảnh mới và đưa chúng qua quá trình tiền xử lý. Sau đó, chúng ta sử dụng mô hình để dự đoán loại của các hình ảnh này.

```

class_name = [
    'Cardboard',
    'Glass',
    'Metal',
    'Paper',
    'Plastic',
    'Trash',
]

def predict_func(img):
    plt.figure(figsize=(6, 4))
    plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
    plt.tight_layout()
    img = cv2.resize(img, (128, 128))
    img = np.reshape(img, [-1, 128, 128, 3])
    result = np.argmax(vggmodel.predict(img))
    print("\033[94m" + "This image -> " + class_name[result] + " " +
        "\033[0m")
    return

test_img = cv2.imread("data/plastic/plastic2.jpg")
predict_func(test_img)

```

- Trong quá trình này, chúng ta đã sử dụng mô hình CNN đã huấn luyện để dự đoán loại của các hình ảnh mới và trực quan hóa kết quả để dễ dàng hiểu. Điều này cho phép chúng ta áp dụng mô hình đã xây dựng vào thực tế và đưa ra dự đoán về dữ liệu mới một cách tự động.

4.5.2. Ma trận nhầm lẫn

- Ta chạy thử nghiệm test trên nhiều ảnh dùng ma trận nhầm lẫn để trực quan hoá dữ liệu
- Tải lại dữ liệu và phân chia dữ liệu test

```

X, y = load_data()
X_train, X_test, y_train, y_test = train_test_split(*arrays: X, y, test_size=0.1, random_state=42)

```

Hình 4.4. Đoạn mã phân chia dữ liệu để lấy dữ liệu test

- Ở đây ta lấy 10% của dữ liệu gốc để thử nghiệm mô hình
- Tải lại model và tải lại dữ liệu train cao nhất

```

my_model = load_model("vggmodel.h5")
my_model.load_weights("weights-46-0.86.keras")

```

Hình 4.5. Tải lại dữ liệu

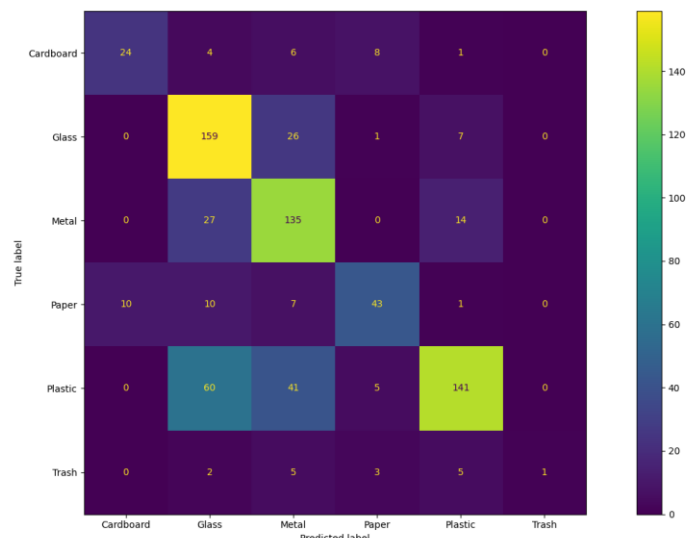
- `np.argmax(y_test, axis=1)` chuyển đổi mảng one-hot encoded `y_test` thành mảng 1 chiều chứa các nhãn lớp thực tế bằng cách tìm chỉ số của giá trị lớn nhất trong mỗi hàng của `y_test`.
- `Y_pred = np.argmax(my_model.predict(X_test), axis=1)` tìm chỉ số của lớp có xác suất cao nhất cho mỗi mẫu, tạo ra một mảng 1 chiều chứa các dự đoán lớp cuối cùng của mô hình.
- `confusion_matrix(y_test_arg, Y_pred)` tính toán ma trận nhầm lẫn giữa các nhãn thực tế và nhãn dự đoán. Ma trận nhầm lẫn giúp đánh giá chi tiết hiệu suất của mô hình phân loại bằng cách cho thấy số lần mỗi nhãn được dự đoán cho mỗi nhãn thực tế.
- `ConfusionMatrixDisplay` giúp hiển thị ma trận nhầm lẫn một cách trực quan với các nhãn lớp cụ thể. Ma trận nhầm lẫn giúp đánh giá chi tiết hiệu suất của mô hình phân loại, cho phép bạn thấy rõ số lượng mẫu được phân loại chính xác hoặc nhầm lẫn giữa các lớp.

- Kết quả khi in ra

```
x_test (128, 128, 3)
y_test [0 0 0 0 1 0]
24/24 ————— 45s 2s/step
Confusion Matrix
[[ 24   4   6   8   1   0]
 [  0 159  26   1   7   0]
 [  0  27 135   0  14   0]
 [ 10  10   7  43   1   0]
 [  0  60  41   5 141   0]
 [  0   2   5   3   5   1]]
```

Hình 4.6. Kết quả ma trận nhầm lẫn trên console

- Kết quả trực quan



Hình 4.7. Kết quả ma trận nhầm lẫn trên trực quan giao diện người dùng

- Ma trận nhầm lẫn này cho chúng ta biết mô hình dự đoán như thế nào cho từng lớp so với nhãn thực tế. Hãy xem xét từng hàng một để hiểu rõ hiệu suất của mô hình:
 - Cardboard (Bìa cứng):
 - Được dự đoán chính xác: 24
 - Sai lệch sang Glass: 4

- Sai lệch sang Metal: 6
- Sai lệch sang Paper: 8
- Sai lệch sang Plastic: 1
- Sai lệch sang Trash: 0
- Glass (Thủy tinh):
 - Được dự đoán chính xác: 159
 - Sai lệch sang Cardboard: 0
 - Sai lệch sang Metal: 26
 - Sai lệch sang Paper: 1
 - Sai lệch sang Plastic: 7
 - Sai lệch sang Trash: 0
- Metal (Kim loại):
 - Được dự đoán chính xác: 135
 - Sai lệch sang Cardboard: 0
 - Sai lệch sang Glass: 27
 - Sai lệch sang Paper: 0
 - Sai lệch sang Plastic: 14
 - Sai lệch sang Trash: 0
- Paper (Giấy):
 - Được dự đoán chính xác: 43
 - Sai lệch sang Cardboard: 10
 - Sai lệch sang Glass: 10
 - Sai lệch sang Metal: 7
 - Sai lệch sang Plastic: 1
 - Sai lệch sang Trash: 0
- Plastic (Nhựa):

- Được dự đoán chính xác: 141
- Sai lệch sang Cardboard: 0
- Sai lệch sang Glass: 60
- Sai lệch sang Metal: 41
- Sai lệch sang Paper: 5
- Sai lệch sang Trash: 0
- Trash (Rác):
 - Được dự đoán chính xác: 1
 - Sai lệch sang Cardboard: 0
 - Sai lệch sang Glass: 2
 - Sai lệch sang Metal: 5
 - Sai lệch sang Paper: 3
 - Sai lệch sang Plastic: 5
- Tổng hợp nhận xét:
 - Mô hình phân loại Plastic có hiệu suất khá tốt với 141 dự đoán đúng, nhưng vẫn có một số lượng lớn nhầm lẫn với Glass (60) và Metal (41).
 - Đối với các nhãn khác, mô hình có vẻ khó khăn hơn trong việc phân biệt, ví dụ như Glass có 27 dự đoán nhầm là Metal.
 - Hiệu suất tốt nhất có thể thấy ở các lớp Cardboard, Glass, và Metal với số lượng dự đoán đúng khá cao.
 - Nhãn Trash có hiệu suất thấp nhất với chỉ 1 dự đoán đúng và nhiều nhầm lẫn sang các lớp khác.
- Dưới đây là các chỉ số đánh giá (Sensitivity, Accuracy, Specificity, PPV) cho từng lớp trong ma trận nhầm lẫn:
 - Cardboard (Bìa cứng):
 - Sensitivity: 0.558
 - Accuracy: 0.961

- Specificity: 0.986
- PPV: 0.706
- Glass (Thủy tinh):
 - Sensitivity: 0.824
 - Accuracy: 0.816
 - Specificity: 0.814
 - PPV: 0.607
- Metal (Kim loại):
 - Sensitivity: 0.767
 - Accuracy: 0.831
 - Specificity: 0.851
 - PPV: 0.614
- Paper (Giấy):
 - Sensitivity: 0.606
 - Accuracy: 0.940
 - Specificity: 0.975
 - PPV: 0.717
- Plastic (Nhựa):
 - Sensitivity: 0.571
 - Accuracy: 0.820
 - Specificity: 0.944
 - PPV: 0.834
- Trash (Rác):
 - Sensitivity: 0.063
 - Accuracy: 0.980
 - Specificity: 1.000

- PPV: 1.000
- Tóm tắt nhận xét:
 - Lớp "Trash" có độ nhạy rất thấp (0.063) nhưng có giá trị dự đoán dương và độ đặc hiệu là 1.000.
 - Lớp "Cardboard" và "Paper" có độ nhạy dưới 0.6, cho thấy mô hình gặp khó khăn trong việc nhận diện chính xác các lớp này.
 - Các lớp "Glass" và "Metal" có độ nhạy và độ chính xác khá tốt, mặc dù độ chính xác của lớp "Glass" thấp hơn một chút so với lớp "Metal".
 - Lớp "Plastic" có giá trị dự đoán dương khá cao (0.834) nhưng độ nhạy chỉ ở mức 0.571.
 - Những chỉ số này cho thấy mô hình phân loại có hiệu suất tốt cho một số lớp, nhưng vẫn cần cải thiện cho các lớp khác như "Trash" và "Cardboard"

4.6. Thử nghiệm thực tế

- Chúng ta thử nghiệm thực tế với việc quan sát rác qua camera
- Khởi tạo biến

```
cap = cv2.VideoCapture(0)
```

Hình 4.8. Khởi tạo biến chạy để chụp từng frame trong video

- Định nghĩa các lớp

```
# Định nghĩa class
class_name = [
    'Cardboard',
    'Glass',
    'Metal',
    'Paper',
    'Plastic',
    'Trash',
]
```

Hình 4.9. Định nghĩa các lớp rác

- Tải lại model và huấn luyện qua file đã lưu

```

46
47
48 my_model = load_model("vggmodel.h5")
49 my_model.load_weights("weights-46-0.86.keras")
50
51

```

Hình 4.10. Tải lại model đã train và chạy lại dữ liệu có rate cao nhất

- Bắt đầu quay video và chuẩn đoán

```

def run_camera():
    while True:
        ret, image_org = cap.read()
        if not ret:
            continue
        image_org = cv2.resize(image_org, dsize=None, fx=0.5, fy=0.5)
        # Resize
        image = image_org.copy()
        image = cv2.resize(image, dsize=(128, 128))
        # image = np.reshape(image, [-1, 128, 128, 3])
        image = image.astype('float') * 1. / 255
        # Convert to tensor
        image = np.expand_dims(image, axis=0)

        # Predict
        predict = my_model.predict(image)
        print("This picture is: ", class_name[np.argmax(predict[0])], (predict[0]))
        print(np.max(predict[0], axis=0))
        if (np.max(predict[0]) >= 0.8) and (np.argmax(predict[0]) != 0):
            # Show image
            font = cv2.FONT_HERSHEY_SIMPLEX
            org = (50, 50)
            fontScale = 1.5
            color = (0, 255, 0)
            thickness = 2
            cv2.putText(image_org, class_name[np.argmax(predict)], org, font,
                        fontScale, color, thickness, cv2.LINE_AA)
            cv2.imshow( winname: "Picture", image_org)
            if cv2.waitKey(1) & 0xFF == ord('q'):
                break
    # When everything done, release the capture
    cap.release()
    cv2.destroyAllWindows()
    return

```

Hình 4.11. Mã code quá trình làm việc quay chính

- Kết quả



Hình 4.12. Kết quả thu được

KẾT LUẬN

Ưu điểm

Dự án "CNN TRONG PHÂN LOẠI RÁC" mang lại nhiều ưu điểm nổi bật, từ khía cạnh công nghệ đến ứng dụng thực tiễn và tác động môi trường. Dưới đây là những ưu điểm chính của dự án:

- **Độ Chính Xác Cao**
 - **Khả Năng Nhận Diện Tốt:** Mạng nơ-ron tích chập (CNN) có khả năng học và trích xuất các đặc trưng phức tạp từ hình ảnh, giúp nâng cao độ chính xác trong việc phân loại các loại rác thải khác nhau.
 - **Giảm Thiểu Lỗi Nhầm Lẫn:** Hệ thống tự động phân loại rác thải dựa trên CNN giảm thiểu các lỗi nhầm lẫn so với phương pháp thủ công, đảm bảo việc phân loại chính xác và nhất quán.
- **Tự Động Hóa Quy Trình**
 - **Giảm Thiểu Can Thiệp Con Người:** Hệ thống phân loại rác tự động giúp giảm thiểu sự can thiệp của con người trong quá trình phân loại, từ đó tăng cường hiệu quả và giảm chi phí lao động.
 - **Hoạt Động Liên Tục:** Hệ thống có thể hoạt động liên tục 24/7 mà không cần nghỉ ngơi, giúp xử lý khối lượng rác thải lớn trong thời gian ngắn.
- **Tối Ưu Hóa Quy Trình Tái Chế**
 - **Tăng Tỷ Lệ Tái Chế:** Phân loại chính xác các vật liệu có thể tái chế như nhựa, kim loại, và giấy giúp tối ưu hóa quy trình tái chế, giảm thiểu lượng rác thải đổ vào các bãi rác.
 - **Bảo Vệ Tài Nguyên Thiên Nhiên:** Việc tái sử dụng và tái chế các vật liệu giúp tiết kiệm tài nguyên thiên nhiên và năng lượng, đóng góp vào phát triển bền vững.
- **Khả Năng Mở Rộng và Linh Hoạt**
 - **Dễ Dàng Mở Rộng:** Hệ thống có thể được mở rộng để nhận diện và phân loại nhiều loại rác thải mới bằng cách huấn luyện thêm dữ liệu và điều chỉnh mô hình CNN.
 - **Ứng Dụng Đa Dạng:** Công nghệ CNN có thể được áp dụng trong nhiều lĩnh vực khác nhau như nông nghiệp, y tế, và quản lý đô thị, mở ra nhiều cơ hội ứng dụng mới.

- **Bảo Vệ Môi Trường**
 - Giảm Thiểu Ô Nhiễm: Phân loại rác thải chính xác giúp giảm thiểu lượng rác thải không phân hủy được đổ vào các bãi rác, từ đó giảm ô nhiễm môi trường.
 - Ngăn Ngừa Rác Thải Nguy Hại: Nhận diện và xử lý đúng cách các loại rác thải nguy hại như pin, hóa chất giúp ngăn ngừa ô nhiễm đất và nước.
- **Hiệu Quả Kinh Tế**
 - Giảm Chi Phí Xử Lý: Hệ thống tự động giúp giảm chi phí lao động và nâng cao hiệu suất xử lý rác thải.
 - Tiết Kiệm Tài Nguyên: Tái chế các vật liệu đã qua sử dụng giúp tiết kiệm chi phí sản xuất và giảm nhu cầu khai thác tài nguyên mới.
- **Phát Triển Công Nghệ và Nghiên Cứu**
 - Đóng Góp Vào Lĩnh Vực AI: Dự án góp phần vào sự phát triển và hoàn thiện các thuật toán học sâu, đặc biệt là mạng nơ-ron tích chập.
 - Thúc Đẩy Nghiên Cứu Liên Ngành: Thành công của dự án mở ra nhiều cơ hội nghiên cứu liên ngành trong các lĩnh vực khác như môi trường, công nghệ thông tin, và quản lý đô thị.

Dự án "CNN TRONG PHÂN LOẠI RÁC" mang lại nhiều ưu điểm vượt trội từ độ chính xác cao, tự động hóa quy trình, tối ưu hóa tái chế, đến bảo vệ môi trường và hiệu quả kinh tế. Những ưu điểm này không chỉ giúp cải thiện hiệu quả quản lý và xử lý rác thải mà còn góp phần vào bảo vệ môi trường và phát triển bền vững. Đồng thời, dự án còn thúc đẩy sự phát triển của công nghệ AI và mở ra nhiều ứng dụng tiềm năng trong các lĩnh vực khác.

Nhược điểm

Mặc dù dự án "CNN TRONG PHÂN LOẠI RÁC" có nhiều ưu điểm, nhưng cũng tồn tại một số nhược điểm và thách thức cần được xem xét. Dưới đây là các nhược điểm chính của dự án:

- **Yêu Cầu Về Dữ Liệu Lớn**
 - Dữ Liệu Đa Dạng: Để huấn luyện mô hình CNN đạt hiệu quả cao, cần một lượng lớn dữ liệu hình ảnh đa dạng về các loại rác thải. Việc thu thập và gán nhãn dữ liệu này có thể tốn nhiều thời gian và công sức.

- **Chất Lượng Dữ Liệu:** Chất lượng dữ liệu không đồng đều hoặc không đủ đa dạng có thể ảnh hưởng đến hiệu suất của mô hình, dẫn đến việc mô hình khó tổng quát hóa khi áp dụng vào các tình huống thực tế.
- **Tính Toán Phức Tạp và Yêu Cầu Tài Nguyên Cao**
 - **Yêu Cầu Về Phần Cứng:** Huấn luyện các mô hình CNN đòi hỏi phần cứng mạnh mẽ, như GPU hoặc TPU, để xử lý khối lượng tính toán lớn. Điều này có thể đòi hỏi chi phí đầu tư ban đầu cao.
 - **Thời Gian Huấn Luyện:** Quá trình huấn luyện mô hình CNN có thể mất nhiều thời gian, đặc biệt là khi làm việc với các tập dữ liệu lớn.
- **Khả Năng Thích Ứng và Tổng Quát Hóa**
 - **Tính Tổng Quát:** Mô hình huấn luyện trên một tập dữ liệu cụ thể có thể không hoạt động tốt trên các dữ liệu mới hoặc các loại rác thải chưa được học. Việc chuyển giao và áp dụng mô hình sang các môi trường khác nhau có thể gặp khó khăn.
 - **Cập Nhật Mô Hình:** Khi xuất hiện các loại rác thải mới hoặc các đặc trưng mới, mô hình cần được cập nhật thường xuyên, điều này đòi hỏi quá trình thu thập dữ liệu và huấn luyện lại mô hình.
- **Độ Tin Cậy Trong Điều Kiện Thực Tế**
 - **Môi Trường Thực Tế:** Các yếu tố như ánh sáng, góc chụp, và điều kiện môi trường khác có thể ảnh hưởng đến độ chính xác của mô hình. Mô hình cần được kiểm thử và điều chỉnh liên tục để đảm bảo hoạt động hiệu quả trong điều kiện thực tế.
 - **Nhiều Hình Ảnh:** Hình ảnh có thể bị nhiễu hoặc bị che khuất, dẫn đến việc mô hình khó nhận diện và phân loại chính xác.
- **Chi Phí Triển Khai và Bảo Trì**
 - **Chi Phí Đầu Tư Ban Đầu:** Triển khai hệ thống phân loại rác tự động yêu cầu đầu tư ban đầu lớn vào phần cứng, phần mềm và cơ sở hạ tầng.
 - **Bảo Trì và Nâng Cấp:** Hệ thống cần được bảo trì và nâng cấp thường xuyên để đảm bảo hoạt động ổn định và hiệu quả. Điều này có thể tốn kém và đòi hỏi kỹ năng chuyên môn cao.
- **Hạn Chế Trong Việc Xử Lý Rác Thải Đặc Biệt**

- Rác Thải Nguy Hại: Việc nhận diện và phân loại các loại rác thải nguy hại hoặc chứa chất độc hại đòi hỏi các biện pháp an toàn và quy trình xử lý đặc biệt, có thể không được mô hình CNN xử lý đầy đủ.
- Rác Thải Có Kích Thước Lớn: Các loại rác thải có kích thước lớn hoặc hình dạng phức tạp có thể gặp khó khăn trong việc nhận diện và phân loại bằng hình ảnh.

Mặc dù dự án "CNN TRONG PHÂN LOẠI RÁC" mang lại nhiều lợi ích và tiềm năng ứng dụng cao, nhưng cũng tồn tại nhiều nhược điểm và thách thức. Các yêu cầu về dữ liệu lớn, tài nguyên tính toán cao, khả năng tổng quát hóa, độ tin cậy trong điều kiện thực tế, chi phí triển khai và bảo trì, cùng với các hạn chế trong việc xử lý rác thải đặc biệt đều là những vấn đề cần được xem xét và giải quyết. Việc nhận diện và khắc phục các nhược điểm này sẽ góp phần nâng cao hiệu quả và ứng dụng thực tiễn của hệ thống phân loại rác tự động dựa trên CNN.

Hướng phát triển

- Mở rộng tập dữ liệu: Ở thời điểm hiện tại dự án này mới có 6 tập ảnh cho nên tôi muốn phát triển nó lên nhiều hơn nữa với nhiều loại rác hơn và đa dạng hình dạng hơn
- Triển khai trên một hệ thống và ứng dụng vào đời sống: Với dự án hiện tại, tôi muốn mở rộng ra để ứng dụng cho cuộc sống thực tế. Có thể là kết hợp với IoT hoặc trên website.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1]. <https://www.kaggle.com/datasets/mostafaabla/garbage-classification>
- [2]. <https://medium.com/@khwabkalra1/convolutional-neural-networks-for-image-classification-f0754f7b94aa>
- [3]. <https://viblo.asia/p/ung-dung-convolutional-neural-network-trong-bai-toan-phan-loai-anh-4dbZNg8ylYM>