

TRƯỜNG ĐẠI HỌC TRÀ VINH
KHOA KỸ THUẬT VÀ CÔNG NGHỆ



ISO 9001:2015

NGUYỄN XUÂN VINH

**PHÂN LOẠI ẢNH DỰA TRÊN KỸ THUẬT
HỌC SÂU**

ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

TRÀ VINH, NĂM 2024

TRƯỜNG ĐẠI HỌC TRÀ VINH
KHOA KỸ THUẬT VÀ CÔNG NGHỆ

PHÂN LOẠI ẢNH DỰA TRÊN KỸ THUẬT
HỌC SÂU

ĐỒ ÁN TỐT NGHIỆP
NGÀNH CÔNG NGHỆ THÔNG TIN

Sinh viên: Nguyễn Xuân Vinh

Lớp: DA20TTB

MSSV: 110120085

GVHD: Nguyễn Mộng Hiền

TRÀ VINH, NĂM 2024

LỜI MỞ ĐẦU

Hiện nay, sự phát triển của công nghệ thông tin và trí tuệ nhân tạo đã giúp cho đời sống con người được phát triển và ngày càng được cải thiện. Trí tuệ nhân tạo ứng dụng vào trong đời sống như xe tự lái, thiết bị không người lái, trợ lý ảo, nhận dạng giọng nói, nhận diện hình ảnh, phân loại hình ảnh,... Trong các lĩnh vực ứng dụng trên thì phân loại ảnh có các ứng dụng vào các lĩnh vực trong đời sống như an ninh giúp nhận diện khuôn mặt, giao thông có hệ thống lái xe tự động giúp phát hiện và phân loại các vật thể xung quanh xe như các ô tô khác, người đi bộ, các biển báo và đèn giao thông, sản xuất thì đảm bảo chất lượng và phát hiện lỗi giúp phát hiện các vấn đề của sản phẩm và đảm bảo chất lượng, thực hiện sắp xếp và phân loại dựa trên đặc điểm của sản phẩm, trong y tế thì các hình ảnh y tế như chụp X-quang, MRI và CT có thể được phân tích tự động bằng cách sử dụng phân loại hình ảnh. Điều này giúp nhân viên y tế chẩn đoán và điều trị bệnh nhân chính xác và hiệu quả hơn. Việc ứng dụng ứng dụng trí tuệ nhân tạo giúp cho công việc và đời sống con người được cải thiện và phát triển.

Trong khóa luận tốt nghiệp này tôi sẽ tập trung nghiên cứu và ứng dụng mô hình học sâu để phân loại ảnh. Nội dung các chương của khóa luận này bao gồm: Chương 1 là trình bày lý do chọn đề tài, mục tiêu, nội dung, đối tượng và phạm vi nghiên cứu, phương pháp nghiên cứu. Chương 2 là trình bày các lý thuyết liên quan đến đề tài thực hiện. Chương 3 là mô tả việc hiện thực hóa nghiên cứu. Chương 4 là trình bày các kết quả đạt được của đề tài. Chương 5 là trình bày kết luận và hướng phát triển của đề tài.

Hy vọng rằng thông qua khóa luận này, chúng ta sẽ hiểu rõ hơn về các kiến thức và ứng dụng của học sâu trong phân loại ảnh mà còn khám phá ra những hướng đi mới và sáng tạo trong lĩnh vực này, góp phần thúc đẩy sự phát triển của trí tuệ nhân tạo và ứng dụng công nghệ vào cuộc sống một cách thiết thực và hiệu quả.

LỜI CẢM ƠN

Trong cuộc sống, để có được sự thành công thì không chỉ có những thử thách, khó khăn, chông gai, sự thất bại mà bên cạnh đó còn có sự giúp đỡ của những người xung quanh. Đó là những điều rất đáng trân trọng vì nhờ có các sự giúp đỡ đó mà chúng ta mới có được sự thành công. Trong quá trình tôi học tập tại Trường Đại học Trà Vinh, tôi đã nhận được sự giúp đỡ của thầy cô, bạn bè và gia đình.

Cám ơn các thầy, cô của Khoa Kỹ thuật và Công Nghệ và các thầy, cô tại Trường Đại học Trà Vinh đã truyền đạt các kiến thức quý báu của mình cho tôi trong quá trình học tập tại trường. Và trong học kì này, tôi được thực hiện đề tài “phân loại ảnh dựa trên kỹ thuật học sâu” thật sự hữu ích và giúp tôi có được các kinh nghiệm sau này.

Xin chân thành cảm ơn thầy Nguyễn Mộng Hiền đã tận tình hướng dẫn tôi trong quá trình tôi thực hiện đề tài “phân loại ảnh dựa trên kỹ thuật học sâu”.

Xin chân thành cảm ơn!

Trà Vinh, ngày tháng năm 2024

Sinh viên thực hiện

Nguyễn Xuân Vinh

[illegible]

iii

BẢN NHẬN XÉT ĐỒ ÁN, KHÓA LUẬN TỐT NGHIỆP
(*Của giảng viên hướng dẫn*)

Họ và tên sinh viên: Nguyễn Xuân Vinh

MSSV: 110120085

Ngành: Công nghệ thông tin

Khóa: 2020

Tên đề tài: Phân loại ảnh dựa trên kỹ thuật học sâu

Họ và tên Giáo viên hướng dẫn: Nguyễn Mộng Hiền

Chức danh: Giảng viên

Học vị: Thạc sĩ

NHẬN XÉT

1. Nội dung đề tài:

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

2. Ưu điểm:

.....

.....

.....

.....

3. Khuyết điểm:

.....

.....

.....

.....

.....

4. Điểm mới đề tài:

.....

.....

.....

.....

.....

5. Giá trị thực trên đề tài:

.....

.....

.....

.....

.....

.....

.....

7. Đề nghị sửa chữa bổ sung:

.....

.....

.....

.....

.....

.....

.....

.....

8. Đánh giá:

.....

.....

.....

.....

Trà Vinh, ngày tháng năm 2024
Giảng viên hướng dẫn
(Ký & ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1. ĐẶT VẤN ĐỀ	1
1.1. Lý do chọn đề tài.....	1
1.2. Mục tiêu	1
1.3. Nội dung.....	1
1.4. Đối tượng và phạm vi nghiên cứu.....	1
1.4.1. Đối tượng nghiên cứu	1
1.4.2. Phạm vi nghiên cứu.....	2
1.5. Phương pháp nghiên cứu.....	2
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT.....	3
2.1. Học sâu.....	3
2.1.1. Định nghĩa.....	3
2.1.2. Ưu điểm và nhược điểm của học sâu	3
2.1.3. Ứng dụng.....	3
2.2. CNN	3
2.2.1. Định nghĩa.....	3
2.2.2. Kiến trúc của mô hình	4
2.2.3. Các loại kiến trúc của mô hình CNN	5
2.2.4. Ưu điểm và nhược điểm	5
2.3. Các vấn đề trong CNN	6
2.3.1. Underfitting	6
2.3.2. Overfitting	6
2.4. Google Colab	7
2.4.1. Định nghĩa	7
2.4.2. Tính năng.....	8
2.4.3. Ứng dụng.....	8
2.5. Python	8
2.5.1. Định nghĩa	8
2.5.2. Tính năng.....	8
2.5.3. Ưu điểm và nhược điểm	8
2.5.4. Ứng dụng.....	9
2.6. Các công cụ sử dụng trong đề tài	9
2.6.1. Keras.....	9
2.6.2. Tensorflow	10
2.6.3. Scikit-learn	10
2.6.4. Numpy.....	11

2.6.5. Matplotlib	11
2.7. Các công trình nghiên cứu liên quan.....	12
CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU	13
3.1. Mô tả bài toán.....	13
3.2. Các bước thực hiện.....	13
3.2.1. Chuẩn bị dữ liệu	13
3.2.2. Chia tập dữ liệu	14
3.2.3. Tiền xử lý dữ liệu	14
3.2.4. Xây dựng mô hình.....	15
3.2.5. Biên dịch mô hình	16
3.2.6. Huấn luyện mô hình	17
3.2.7. Đánh giá mô hình	17
3.2.8. Biểu đồ hiển thị độ chính xác.....	17
3.2.9. Biểu đồ hiển thị giá trị mất mát.....	17
3.2.10. Hiển thị hình ảnh và kết quả dự đoán	18
3.3. Hướng dẫn sử dụng	18
3.4. Mục tiêu cải tiến.....	18
CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU	19
4.1. Kết quả thực hiện	19
4.1.1. Kết quả hiển thị thông tin dữ liệu	19
4.1.2. Kết quả xây dựng kiến trúc mô hình.....	19
4.1.3. Kết quả huấn luyện mô hình	21
4.1.4. Đánh giá mô hình	21
4.1.5. Biểu đồ hiển thị độ chính xác.....	22
4.1.6. Biểu đồ hiển thị mất mát	23
4.1.7. Kết quả hiển thị hình ảnh và kết quả dự đoán.....	24
4.2. So sánh kết quả trước và sau khi cải thiện mô hình.....	34
CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	38
5.1. Kết quả đạt được	38
5.2. Hướng phát triển	38
DANH MỤC TÀI LIỆU THAM KHẢO.....	39

DANH MỤC HÌNH ẢNH

Hình 2.1 Kiến trúc của mô hình CNN	4
Hình 3.1 Hình ảnh các lớp trong tập dữ liệu	14
Hình 3.2 Sơ đồ mô hình	16
Hình 4.1 Kết quả hiển thị thông tin dữ liệu.....	19
Hình 4.2 Kết quả xây dựng kiến trúc mô hình.....	20
Hình 4.3 Kết quả huấn luyện mô hình	21
Hình 4.4 Kết quả đánh giá mô hình	21
Hình 4.5 Biểu đồ hiển thị độ chính xác.....	22
Hình 4.6 Biểu đồ hiển thị giá trị mất mát.....	23
Hình 4.7 Kết quả hiển thị hình ảnh và dự đoán xe tải.....	24
Hình 4.8 Kết quả hiển thị hình ảnh và dự đoán nhãn tàu.....	25
Hình 4.9 Kết quả hiển thị hình ảnh và dự đoán nhãn con ngựa.....	26
Hình 4.10 Kết quả hiển thị hình ảnh và dự đoán nhãn con chó	27
Hình 4.11 Kết quả hiển thị hình ảnh và dự đoán nhãn con ếch	28
Hình 4.12 Kết quả hiển thị hình ảnh và dự đoán nhãn máy bay	29
Hình 4.13 Kết quả hiển thị hình ảnh và dự đoán nhãn con chim.....	30
Hình 4.14 Kết quả hiển thị hình ảnh và dự đoán nhãn con nai	31
Hình 4.15 Kết quả hình ảnh và dự đoán nhãn xe ô tô.....	32
Hình 4.16 Kết quả hiển thị hình ảnh và dự đoán nhãn con mèo	33
Hình 4.17 Kết quả hình ảnh trước và sau khi cải thiện với hình ảnh con mèo	34
Hình 4.18 Kết quả hình ảnh trước và sau khi cải thiện với hình ảnh con chó	35
Hình 4.20 Kết quả cả hai mô hình dự đoán sai nhãn con mèo.....	36
Hình 4.21 Kết quả cả hai mô hình dự đoán sai nhãn con chó.....	37

DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa
AI	Artificial Intelligence
CNN	Convolutional Neural Network
ML	Machine Learning

CHƯƠNG 1. ĐẶT VẤN ĐỀ

1.1. Lý do chọn đề tài

Hiện nay, sự phát triển của trí tuệ nhân tạo và học sâu ngày càng phát triển giúp cho đời sống con người ngày càng nâng cao và cải thiện, phân loại ảnh đang trở thành một xu hướng phổ biến và mang lại hiệu quả cao trong nhiều lĩnh vực khác nhau như y tế, an ninh, giao thông, nông nghiệp, và thương mại điện tử. Nhu cầu phân loại và xử lý hình ảnh ngày càng tăng cao trong các ngành công nghiệp và dịch vụ. Từ việc chẩn đoán bệnh qua hình ảnh y tế, nhận diện khuôn mặt trong an ninh, đến tự động hóa quy trình sản xuất, việc nghiên cứu và phát triển các phương pháp phân loại ảnh có độ chính xác cao là rất cần thiết.

Từ các lý do trên tôi thực hiện đề tài “phân loại ảnh dựa trên kỹ thuật học sâu” để giúp cho chúng ta hiểu được các kiến thức về học sâu và mô hình học sâu trong phân loại, áp dụng phân loại ảnh vào thực tiễn đời sống.

1.2. Mục tiêu

- Cải thiện độ chính xác, tính tự động hóa, tính tổng quát hóa và hiệu suất của quá trình phân loại, từ đó áp dụng vào nhiều lĩnh vực và giải quyết các vấn đề thực tiễn.
- Giúp cho sinh viên nắm vững kiến thức hiểu và vận dụng được mô hình học sâu để phân loại ảnh.

1.3. Nội dung

- Chuẩn bị dữ liệu.
- Tiền xử lý dữ liệu.
- Xây dựng mạng neural.
- Huấn luyện mô hình.
- Kiểm tra và đánh giá mô hình.

1.4. Đối tượng và phạm vi nghiên cứu

1.4.1. Đối tượng nghiên cứu

Đối tượng nghiên cứu trong đề tài là xây dựng mô hình CNN để thực hiện phân loại ảnh.

1.4.2. Phạm vi nghiên cứu

Phạm vi nghiên cứu là các phương pháp và mô hình sử dụng trong đề tài. Mô hình sẽ được huấn luyện và tiến hành thử nghiệm trên tập dữ liệu để phân loại hình ảnh chính xác.

1.5. Phương pháp nghiên cứu

- Phương pháp nghiên cứu lý thuyết: Là tìm hiểu các tài liệu, sách và thông tin liên quan đến phân loại ảnh dựa trên kỹ thuật học sâu. Tìm hiểu mô hình CNN trong phân loại ảnh. Xây dựng mô hình CNN.
- Phương pháp nghiên cứu thực nghiệm: Tải tập dữ liệu cifar-10. Thực hiện chia dữ liệu. Thực hiện tiền xử lý dữ liệu như chuẩn hóa dữ liệu, chuyển nhãn sang one-hot encoding, tăng cường dữ liệu. Xây dựng mô hình. Biên dịch mô hình. Huấn luyện mô hình. Đánh giá mô hình. Vẽ biểu đồ hiển thị độ chính xác. Vẽ biểu đồ hiển thị độ mất mát. Hiển thị hình ảnh và kết quả dự đoán.

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. Học sâu

2.1.1. Định nghĩa

Học sâu (deep learning) là một nhánh của học máy (machine learning) dựa trên kiến trúc mạng nơ-ron nhân tạo [1].

2.1.2. Ưu điểm và nhược điểm của học sâu

2.1.2.1. Ưu điểm

Sau đây sẽ là các ưu điểm của học sâu sẽ bao gồm các ưu điểm sau đây [1]:

- Độ chính xác cao.
- Tính năng tự động.
- Khả năng mở rộng.
- Cải tiến liên tục.
- Tính linh hoạt.

2.1.2.2. Nhược điểm

Sau đây là các nhược điểm của học sâu bao gồm các nhược điểm sau [1]:

- Yêu cầu tính toán cao.
- Yêu cầu lượng dữ liệu lớn được gán nhãn.

2.1.3. Ứng dụng

Sau đây sẽ là các ứng dụng của học sâu sẽ bao gồm các ứng dụng sau [1]:

- Phát hiện và nhận dạng đối tượng.
- Phân loại ảnh.
- Phân đoạn ảnh.
- Tạo văn bản tự động.
- Nhận dạng giọng nói.

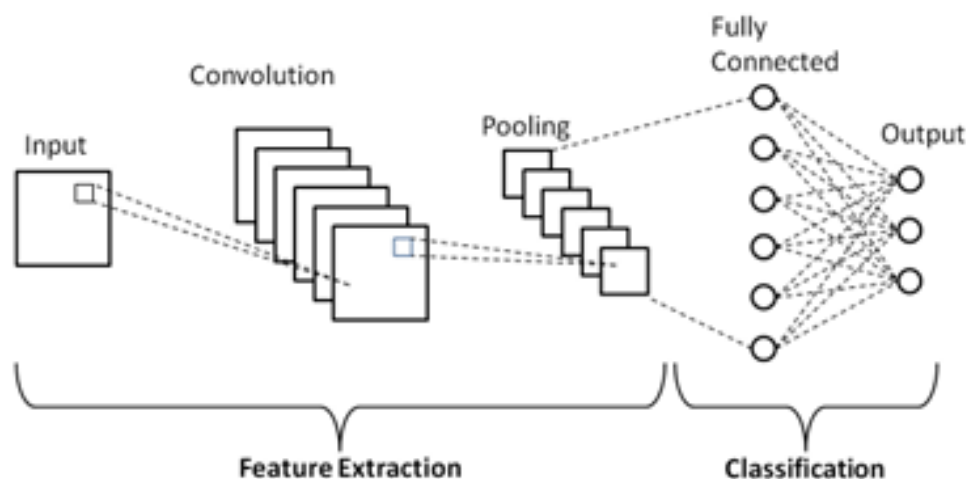
2.2. CNN

2.2.1. Định nghĩa

CNN là viết tắt của Convolutional Neural Network hay còn được gọi là convnets. Là một mô hình học máy, cụ thể hơn là một thuật toán học sâu. CNN thường được sử dụng cho xử lý hình ảnh, nhận dạng và phân loại hình ảnh [2].

2.2.2. Kiến trúc của mô hình

Kiến trúc của mô hình CNN bao gồm 5 lớp: Lớp đầu vào (Input layer), lớp tích chập (convolutional layer), lớp gộp (pooling layer), lớp kết nối đầy đủ (fully connected layer) và lớp đầu ra (output layer).



Hình 2.1 Kiến trúc của mô hình CNN

- Lớp đầu vào (input layer): Lớp đầu vào (input layer) là lớp chỉ định kích thước cho các hình ảnh đầu vào và có thể thay đổi kích thước nếu cần [3].
- Lớp tích chập (convolutional layer): Lớp tích chập (convolutional layer) được sử dụng để trích xuất, lấy các đặc trưng, đặc điểm của đối tượng trong hình ảnh. Lớp tích chập (convolutional layer) sử dụng các filter có kích thước bằng nhau và sử dụng các filter để nhận dạng các đối tượng trong hình ảnh [2].
- Lớp gộp (pooling layer): Lớp gộp (pooling layer) là lớp sau lớp tích chập (convolutional layer) trong mô hình CNN. Lớp gộp (pooling layer) được sử dụng nhằm mục đích giảm chiều dữ liệu đầu vào nhưng vẫn giữ các thông tin quan trọng và tăng khả năng khái quát hóa cho mô hình để giảm tình trạng overfitting. Các kỹ thuật thường sử dụng trong lớp gộp (pooling layer) là max pooling và average pooling. Max pooling là giữ lại giá trị tối đa còn average pooling thì giữ giá trị trung bình [2].
- Lớp kết nối đầy đủ (fully connected layer): Lớp kết nối đầy đủ (fully connected layer) được sử dụng để kết hợp các đặc trưng của các lớp trước đó là lớp tích chập (convolutional layer) và lớp gộp (pooling layer) [3].

- Lớp đầu ra (output layer): Lớp đầu ra (output layer) là lớp cuối cùng. Lớp đầu ra (ouput layer) thực hiện đưa ra kết quả cuối cùng [3].

2.3.3. Các loại kiến trúc của mô hình CNN

Sau đây là một số loại kiến trúc khác nhau của mô hình CNN:

- AlexNet: Kiến trúc AlexNet được phát triển bởi Alex Krizhevsky, Ilya Sutskever và Geoff Hinton. AlexNet là mô hình mạng CNN có quy mô lớn đầu tiên. Kiến trúc AlexNet sử dụng các tập dữ liệu có kích thước lớn và đạt kết quả cao tại thời điểm công bố. Tổng số tham số trong kiến trúc AlexNet là khoảng 60 triệu [5].
- GoogleNet: Kiến trúc GoogleNet được phát triển bởi Jeff Dean, Christian Szegedy, Alexandro Szegedy và cộng sự. Kiến trúc GoogleNet được ứng dụng để thực hiện các công việc như nhận dạng chữ số và được sử dụng làm proxy để phát hiện các vật thể ven đường [5].
- ResNet: Kiến trúc ResNet là kiến trúc CNN được phát triển bởi Kaiming He. ResNet có 152 lớp và hơn một triệu tham số. Kiến trúc ResNet có hiệu quả về mặt tính toán và có thể mở rộng hoặc thu hẹp để phù hợp với tính toán của GPU [5].
- MobileNets: Kiến trúc MobileNet là có thể triển khai trên các thiết bị di động để phân loại hình ảnh hoặc phát hiện đối tượng. Kiến trúc MobileNet được phát triển bởi Andrew G Trillion và các cộng sự [5].

2.2.4. Ưu điểm và nhược điểm

2.2.4.1. Ưu điểm

Sau đây là các ưu điểm của mô hình CNN bao gồm các ưu điểm sau đây [4]:

- Nhận dạng hình ảnh với độ chính xác cao.
- Tự động phát hiện các tính năng quan trọng mà không cần sự giám sát của con người.
- Chia sẻ trọng số.

2.2.4.2. Nhược điểm

Sau đây là nhược điểm của mô hình CNN bao gồm các nhược điểm đây [4]:

- Cần lượng lớn dữ liệu.
- Thiếu tính bất biến.

2.3. Các vấn đề trong CNN

2.3.1. Underfitting

2.3.1.1. Định nghĩa

Một mô hình thống kê hoặc thuật toán học máy được cho là không phù hợp khi một mô hình quá đơn giản để nắm bắt được độ phức tạp của dữ liệu. Thể hiện sự bất lực của mô hình trong việc học dữ liệu huấn luyện một cách hiệu quả, dẫn đến hiệu suất kém cả trên dữ liệu huấn luyện và kiểm tra. Nói một cách đơn giản, mô hình không phù hợp là không chính xác, đặc biệt khi áp dụng cho các ví dụ mới, chưa được nhìn thấy. Chủ yếu xảy ra khi sử dụng mô hình rất đơn giản với các giả định quá đơn giản. Để giải quyết vấn đề không phù hợp của mô hình, cần sử dụng các mô hình phức tạp hơn, với tính năng biểu diễn nâng cao và ít chính quy hơn [6].

Lưu ý: Mô hình underfitting có độ lệch cao và phương sai thấp.

2.3.1.2. Nguyên nhân

- Mô hình quá đơn giản nên có thể không thể hiện được sự phức tạp của dữ liệu.
- Các đặc điểm đầu vào được sử dụng để huấn luyện mô hình không phải là sự thể hiện đầy đủ các yếu tố cơ bản ảnh hưởng đến biến mục tiêu.
- Kích thước của tập dữ liệu huấn luyện được sử dụng là không đủ.
- Việc chính quy hóa quá mức được sử dụng để ngăn chặn việc trang bị quá mức, điều này hạn chế mô hình thu thập dữ liệu tốt.
- Các tính năng không được thu nhỏ.

2.3.1.3. Giải pháp

- Tăng độ phức tạp của mô hình.
- Tăng số lượng tính năng, thực hiện kỹ thuật tính năng.
- Loại bỏ nhiễu khỏi dữ liệu.
- Tăng số lượng kỷ nguyên hoặc tăng thời gian đào tạo để có kết quả tốt hơn.

2.3.2. Overfitting

2.3.2.1. Định nghĩa

Là quá phù hợp khi mô hình đó không đưa ra dự đoán chính xác về dữ liệu thử nghiệm. Khi một mô hình được đào tạo với quá nhiều dữ liệu, sẽ bắt đầu học hỏi từ các mục nhập dữ liệu nhiễu và không chính xác trong tập dữ liệu. Và khi thử nghiệm với dữ liệu thử nghiệm cho kết quả có phương sai cao. Khi đó mô hình không phân loại dữ liệu một cách chính xác do có quá nhiều chi tiết và nhiễu. Nguyên nhân của

việc trang bị quá mức là do các phương pháp phi tham số và phi tuyến tính vì các loại thuật toán học máy này có nhiều quyền tự do hơn trong việc xây dựng mô hình dựa trên tập dữ liệu và do đó chúng thực sự có thể xây dựng các mô hình phi thực tế. Một giải pháp để tránh trang bị quá mức là sử dụng thuật toán tuyến tính nếu chúng ta có dữ liệu tuyến tính hoặc sử dụng các tham số như độ sâu tối đa nếu chúng ta đang sử dụng cây quyết định [6].

2.3.2.2. Nguyên nhân

- Phương sai cao và độ lệch thấp.
- Mô hình quá phức tạp.
- Kích thước của dữ liệu đào tạo.

2.3.2.3. Giải pháp

- Cải thiện chất lượng dữ liệu huấn luyện giúp giảm việc trang bị quá mức bằng cách tập trung vào các mẫu có ý nghĩa, giảm thiểu rủi ro khớp các tính năng nhiều hoặc không liên quan.
- Việc tăng dữ liệu huấn luyện có thể cải thiện khả năng khái quát hóa của mô hình đối với dữ liệu chưa nhìn thấy và giảm khả năng khớp quá mức.
- Giảm độ phức tạp của mô hình.
- Early stopping trong giai đoạn huấn luyện.
- Ridge Regularization và Lasso Regularization.
- Sử dụng dropout cho mạng lưới thần kinh để giải quyết vấn đề trang bị quá mức.

2.4. Google Colab

2.4.1. Định nghĩa

Google Colab, viết tắt của Google Colaboratory, là một dịch vụ cung cấp môi trường Jupyter Notebook hoàn toàn trực tuyến. Cho phép người dùng tạo, chia sẻ và chỉnh sửa các tệp notebook một cách dễ dàng mà không cần cài đặt bất kỳ phần mềm nào.

Google Colab cung cấp một môi trường làm việc tích hợp với Google Drive, cho phép truy cập và xử lý dữ liệu trực tiếp từ trong notebook.

Mục đích chính của Google Colab là hỗ trợ người dùng trong việc phát triển và chia sẻ các dự án liên quan đến khoa học dữ liệu, học máy (machine learning), và nghiên cứu khoa học thông qua môi trường lập trình Python để sử dụng [7].

2.4.2. Tính năng

Sau đây sẽ là các tính năng của Google Colab sẽ bao gồm các tính năng sau [7]:

- Sử dụng Jupyter Notebooks trực tuyến.
- Khả năng chia sẻ và cộng tác.
- Dùng GPU và TPU miễn phí.
- Lưu trữ dữ liệu trên Google Drive và tích hợp Google Cloud.

2.4.3. Ứng dụng

Sau đây sẽ là các ứng dụng của Google Colab bao gồm các ứng dụng sau [7]:

- Học máy và khoa học dữ liệu
- Phát triển ứng dụng AI và ML
- Nghiên cứu và phân tích dữ liệu
- Giáo dục và đào tạo.

2.5. Python

2.5.1. Định nghĩa

Python là một ngôn ngữ lập trình bậc cao. Python thường được sử dụng để phát triển các phần mềm, trang web, phân tích dữ liệu và tự động hóa dịch vụ [8].

2.5.2. Tính năng

Sau đây sẽ là các tính năng của ngôn ngữ Python bao gồm các tính năng sau [8]:

- Thông dịch.
- Linh hoạt.
- Dễ sử dụng.
- Hướng đối tượng.

2.5.3. Ưu điểm và nhược điểm

2.5.3.1. Ưu điểm

Sau đây sẽ là các ưu điểm của ngôn ngữ Python bao gồm các ưu điểm sau [8]:

- Dễ đọc và dễ sử dụng.
- Tính ứng dụng rộng rãi.
- Tốc độ xử lý nhanh.
- Sửa lỗi mã nguồn.
- Bảo trì đơn giản.
- Cơ sở dữ liệu phong phú.
- Hỗ trợ GUI programming.

2.5.3.2. Nhược điểm

Sau đây là các nhược điểm của ngôn ngữ Python bao gồm nhược điểm sau [8]:

- Tốc độ thực thi chậm.
- Cần nhiều bộ nhớ.
- Không phù hợp cho phát triển trò chơi và ứng dụng trên thiết bị di động.
- Khó phát hiện lỗi.
- Quyền truy cập cơ sở dữ liệu.
- Hạn chế trong thiết kế.
- Khó kiểm thử.

2.5.4. Ứng dụng

Sau đây là các ứng dụng của ngôn ngữ Python bao gồm các ứng dụng sau [8]:

- Phát triển web.
- Phát triển game.
- Phân tích số liệu và nghiên cứu khoa học.
- Công cụ tự động hóa.
- Khoa học máy tính và machine learning.

2.6. Các công cụ sử dụng trong đề tài

2.6.1. Keras

2.6.1.1. Định nghĩa

Keras là một API cấp cao, thân thiện với người dùng. Keras cho phép bạn xây dựng, đào tạo và triển khai các mô hình học sâu. Keras cung cấp một API cấp cao trực quan và dễ sử dụng, lý tưởng cho cả người mới bắt đầu và chuyên gia [9].

2.6.1.2. Lịch sử

Keras được phát triển bởi kỹ sư của Google tên là François Chollet. Keras được phát hành vào tháng 3 năm 2015 [9].

2.6.1.3. Tính năng

Sau đây là các tính năng của Keras bao gồm các tính năng sau đây [9]:

- Đơn giản
- Mở rộng và tùy chỉnh.
- Tương thích với đa nền tảng.
- Thử nghiệm nhanh.

2.6.1.4. Ứng dụng

Sau đây là các ứng dụng của keras bao gồm các ứng dụng sau đây [9]:

- Xử lý hình ảnh và video.
- Xử lý ngôn ngữ tự nhiên.

2.6.2. Tensorflow

2.6.2.1. Định nghĩa

TensorFlow là thư viện mã nguồn mở cho học máy (machine learning), được phát triển bởi các nhà nghiên cứu của Google. Việc hỗ trợ mạnh mẽ các phép toán học để tính toán trong học máy (machine learning) và học sâu (deep learning) đã giúp việc tiếp cận các bài toán trở nên đơn giản, nhanh chóng và tiện lợi hơn. [10]

2.6.2.2. Lịch sử

TensorFlow được ra mắt lần đầu vào cuối năm 2015, phiên bản TensorFlow cuối cùng xuất hiện vào năm 2017. Tensorflow là mã nguồn mở dưới sự cho phép của Apache Open Source [10].

2.6.3. Scikit-learn

2.6.3.1. Định nghĩa

Scikit-learn, thường được gọi là sklearn, là một thư viện mã nguồn mở của Python cung cấp một bộ thuật toán và công cụ học máy toàn diện để thực hiện tiên xử lý dữ liệu, phân loại, lựa chọn mô hình,... Scikit-learn được xây dựng dựa trên các thư viện khoa học cơ bản như NumPy, SciPy và matplotlib [11].

2.6.3.2. Tính năng

Sau đây là các tính năng của scikit-learn bao gồm các tính năng sau đây [11]:

- Là một API dễ sử dụng.
- Cung cấp nhiều thuật toán.
- Tiên xử lý và trích xuất các tính năng.
- Đánh giá mô hình và xác thực.

2.6.3.3. Ứng dụng

Sau đây là các ứng dụng của scikit-learn bao gồm các ứng dụng sau đây [11]:

- Phân loại
- Phân cụm
- Hồi quy.

2.6.4. Numpy

2.6.4.1. Định nghĩa

NumPy là một thư viện bao gồm các đối tượng mảng đa chiều và một tập hợp các hàm để thực hiện các thao tác. Đây là một trong những gói của Python được sử dụng nhiều nhất cho tính toán khoa học vì numpy cho phép thực hiện các phép toán và logic trên các mảng. NumPy là một ngôn ngữ lập trình Python [12].

2.6.4.2. Lịch sử

Travis Oliphant đã xây dựng NumPy vào năm 2005. Numeric là tiền thân của NumPy, được thành lập vào năm 1995 bởi Jim Hugunin với sự giúp đỡ của các cộng sự [12].

2.6.4.3. Tính năng

Sau đây là các tính năng của numpy sẽ bao gồm các tính năng sau đây [12]:

- Hiệu suất cao
- Tích hợp mã từ C/C++, Fortran:
- Làm việc với cơ sở dữ liệu đa dạng.

2.6.5. Matplotlib

2.6.5.1. Định nghĩa

Matplotlib là thư viện dùng để vẽ đồ thị trong Python được sử dụng để tạo hình ảnh tĩnh, động và tương tác. Mục đích chính của Matplotlib là cung cấp cho người dùng các công cụ và chức năng để biểu diễn dữ liệu dưới dạng đồ họa, giúp phân tích và hiểu dữ liệu dễ dàng hơn. Matplotlib được John D. Hunter phát triển vào năm 2003 và hiện được duy trì bởi một cộng đồng lớn các nhà phát triển [13].

2.6.5.2. Tính năng

Sau đây là các tính năng của matplotlib sẽ bao gồm các tính năng sau đây [13]:

- Linh hoạt
- Tùy chỉnh.
- Tích hợp với numpy.
- Đa nền tảng.
- Biểu đồ tương tác.

2.6.5.3. Ưu điểm

Sau đây là các ưu điểm của numpy sẽ bao gồm các ưu điểm sau đây [13]:

- Linh hoạt

- Tùy chỉnh.
- Tích hợp với numpy.
- Đa nền tảng.
- Biểu đồ tương tác.
- Tích hợp với Jupyter
- Tài liệu phong phú và hỗ trợ cộng đồng

2.6.5.4 . Nhược điểm

Sau đây là các nhược điểm của matplotlib sẽ bao gồm các nhược điểm sau đây:

- Cú pháp chi tiết
- Tương tác hạn chế.
- Khả năng vẽ biểu đồ 3D hạn chế.
- Hiệu suất với tập dữ liệu lớn.

2.7. Các công trình nghiên cứu liên quan

Học sâu là một nhánh của công nghệ học máy trong lĩnh vực trí tuệ nhân tạo [14, 15]. Công nghệ học sâu có thể tự động trích xuất, khai thác các đặc trưng, đặc điểm, phân loại bằng tính toán. Công nghệ học sâu được ứng dụng nhiều trong các lĩnh vực đời sống như an ninh, y tế, giao thông,...

Hiện nay, kĩ thuật học sâu được ứng dụng để thực hiện các công việc trong cuộc sống. Công trình nghiên cứu số [16] áp dụng mô hình CNN trong học sâu để thực hiện tìm kiếm hình ảnh theo nội dung. Công trình nghiên cứu số [17] áp dụng CNN để thực hiện để phân loại và so sánh độ tương đồng để tìm kiếm các ảnh tương tự trên tập các ảnh sản phẩm gồm 11 nhãn. Ngoài ra còn có công trình nghiên cứu số [18] áp dụng các mô hình như: CNN, ResNet, InceptionNet, DenseNet và VGG16 để thử nghiệm bệnh viêm phổi thông qua hình ảnh X-quang lồng ngực và thông qua đó có thể đánh giá được mô hình có độ chính xác cao nhất.

CHƯƠNG 3. HIỆN THỰC HÓA NGHIÊN CỨU

3.1. Mô tả bài toán

Trong đời sống thực tế, việc phân loại ảnh thường được thực hiện bằng mắt thường, bằng phương pháp thủ công thì sẽ có thể dẫn đến sai sót, tốn thời gian. Từ vấn đề trên, thì việc tạo ra hệ thống phân loại ảnh sẽ giúp việc phân loại ảnh chính xác hơn, tiết kiệm thời gian và phân loại hình ảnh đúng với nhóm của hình ảnh đó. Để thực hiện phân loại ảnh trong đề này thì sử dụng mô hình CNN trong học sâu. Mô hình CNN phân loại ảnh rất hiệu quả.

Đầu vào của bài toán phân loại ảnh dựa trên kỹ thuật học sâu là tập dữ liệu hình ảnh cifar-10 gồm 60,000 hình ảnh màu thuộc 10 lớp khác nhau, mỗi lớp chứa 6,000 hình ảnh. Các lớp này bao gồm các đối tượng hằng ngày như máy bay, ô tô, chim, mèo, hươu, chó, ếch, ngựa, thuyền và xe tải.

Quá trình xử lý dữ liệu bao gồm chuẩn hóa dữ liệu, chuyển nhãn sang one-hot encoding và tăng cường dữ liệu. Sau đó, thực hiện xây dựng mô hình, biên dịch mô hình để chuẩn bị cho quá trình huấn luyện mô hình.

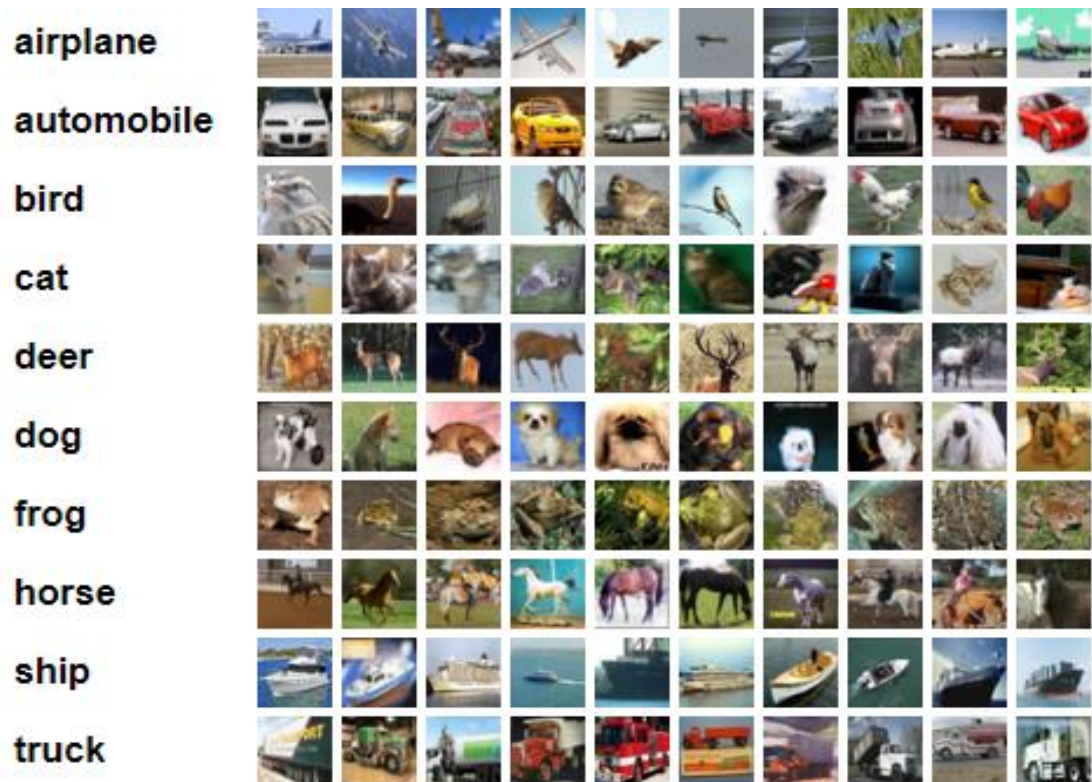
Đầu ra của bài toán phân loại dựa trên kỹ thuật học sâu là hình ảnh và kết quả dự đoán của hình ảnh.

Bài toán phân loại ảnh dựa trên kỹ thuật học sâu giúp nhận dạng và phân loại chính xác các đối tượng trong hình ảnh. Từ đó, ta có thể áp dụng vào trong đời sống như nhận dạng khuôn mặt, robot tự động có thể dựa vào phân loại để nhận diện và xử lý môi trường xung quanh,...

3.2. Các bước thực hiện

3.2.1. Chuẩn bị dữ liệu

Trong đề tài “phân loại ảnh dựa trên kỹ thuật học sâu” sử dụng bộ dữ liệu cifar-10. Bộ dữ liệu CIFAR-10 bao gồm 60000 hình ảnh màu 32x32 trong 10 lớp, với 6000 hình ảnh mỗi lớp. Có 50000 hình ảnh đào tạo và 10000 hình ảnh thử nghiệm. Bộ dữ liệu được chia thành năm đợt huấn luyện và một đợt thử nghiệm, mỗi đợt có 10000 hình ảnh. Lô thử nghiệm chứa chính xác 1000 hình ảnh được chọn ngẫu nhiên từ mỗi lớp. Các đợt huấn luyện chứa các hình ảnh còn lại theo thứ tự ngẫu nhiên, nhưng một số đợt huấn luyện có thể chứa nhiều hình ảnh từ lớp này hơn lớp khác. Giữa chúng, các đợt huấn luyện chứa chính xác 5000 hình ảnh từ mỗi lớp [19].



Hình 3.1 Hình ảnh các lớp trong tập dữ liệu

3.2.2. Chia tập dữ liệu

Sau khi thực hiện tải dữ liệu từ tập dữ liệu cifar-10 thì thực hiện chia tập dữ liệu. Tập dữ liệu được chia thành tập huấn luyện và tập xác thực. Trong đó, tham số `test_size` dùng để xác định tỷ lệ của tập huấn luyện ban đầu được sử dụng để xác thực là 20%

3.2.3. Tiền xử lý dữ liệu

Trong tiền xử lý dữ liệu thì thực hiện chuẩn hóa dữ liệu, chuyển đổi nhãn các lớp sang dạng one-hot encoding và tăng cường dữ liệu.

3.2.3.1. Chuẩn hóa dữ liệu

Chuẩn hóa dữ liệu sẽ lấy dữ liệu huấn luyện, dữ liệu thẩm định và dữ liệu kiểm tra chia cho 255. Việc chuẩn hóa dữ liệu sẽ giúp việc học tốt hơn và cải thiện hiệu suất trong quá trình huấn luyện.

3.2.3.2. Chuyển đổi nhãn sang one-hot encoding

Chuyển đổi nhãn các lớp sang dạng one-hot encoding thì sẽ lấy nhãn của tập dữ liệu huấn luyện, dữ liệu thẩm định và kiểm tra chuyển sang one-hot encoding.

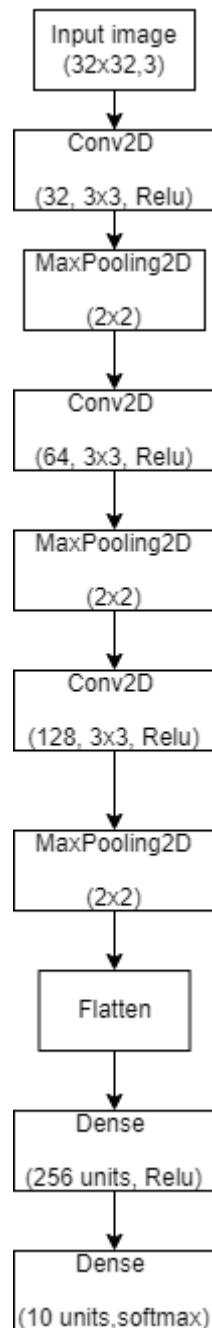
3.2.3.3. Tăng cường dữ liệu

Tăng cường dữ liệu thì sẽ sử dụng các kỹ thuật để tăng cường dữ liệu như xoay, dịch chuyển, lật ngang, lật dọc, phóng to hoặc thu nhỏ, cắt và thay đổi độ sáng để tạo ra các biến thể của dữ liệu hình ảnh trong quá trình huấn luyện mô hình. Các biến thể này giúp cải thiện khả năng tổng quát hóa của mô hình và giảm thiểu hiện tượng overfitting bằng cách mở rộng tập dữ liệu huấn luyện.

3.2.4. Xây dựng mô hình

Mô hình được lựa chọn để phân loại ảnh là mô hình CNN. Mô hình CNN phân loại ảnh rất hiệu quả, có độ chính xác cao, chia sẻ trọng lượng, tự động phát hiện các tính năng quan trọng mà không cần sự giám sát của con người.

Để xây dựng mô hình thì trước tiên là khởi tạo mô hình. Tiếp theo, thêm lớp tích chập đầu tiên với bộ lọc là 32, kích thước ma trận bộ lọc là 3x3, padding là same, hàm kích hoạt là relu, hiển thị ảnh đầu vào là 32x32 và 3 là kênh màu của hình ảnh RGB. Tiếp theo, thêm lớp gộp (pooling) với giá trị tối đa trên cửa sổ đầu vào là 2x2. Thêm lớp tích chập với bộ lọc là 64, kích thước ma trận bộ lọc là 3x3, padding là same, hàm kích hoạt là relu. Tiếp theo, thêm lớp gộp (pooling) với giá trị tối đa trên cửa sổ đầu vào là 2x2. Thêm lớp tích chập với bộ lọc là 128, kích thước ma trận bộ lọc là 3x3, padding là same, hàm kích hoạt là relu. Tiếp theo, thêm lớp gộp (pooling) với giá trị tối đa trên cửa sổ đầu vào là 2x2. Thêm lớp Flatten để làm phẳng đầu vào. Lớp fully connected (hoàn toàn kết nối) với 256 units và hàm kích hoạt là ReLU. Lớp fully connected cuối cùng với 10 units và hàm kích hoạt softmax.



Hình 3.2 Sơ đồ mô hình

3.2.5. Biên dịch mô hình

Biên dịch mô hình sẽ sử dụng phương thức compile trong keras để chuẩn bị cho huấn luyện mô hình. Trong phương thức này, thực hiện định nghĩa trình tối ưu hóa, hàm mất mát và các chỉ số đánh giá (metrics). Sử dụng hàm mất mát 'categorical_crossentropy' để đo lường mức độ "sai lệch" giữa các dự đoán của mô hình và nhãn thực tế. Sử dụng thuật toán Adam cho quá trình tối ưu hóa, và accuracy yêu cầu keras theo dõi độ chính xác của mô hình trong quá trình huấn luyện.

3.2.6. Huấn luyện mô hình

Khởi tạo `Early_Stopping`. Trong đó, `monitor` là `val_accuracy` để dừng huấn luyện khi độ chính xác trên tập `validation` không cải thiện. `Patiencel` là 5 là sau 5 epoch mà không có cải thiện, huấn luyện sẽ dừng lại. `Restore_best_weights` là `True`: Dừng để khôi phục lại trọng số của mô hình tại thời điểm có `val_accuracy` tốt nhất khi dừng huấn luyện.

Quá trình huấn luyện mô hình tôi sẽ tạo biến `hist` để lưu quá trình quá trình huấn luyện. Sử dụng phương thức `fit` để huấn luyện mô hình. Trong đó, đưa vào phương thức là `x_train` là dữ liệu đầu vào, `y_train_epoch` là các nhãn đã được chuyển sang one-hot encoding, `batch_size` là 256 là số lượng mẫu được sử dụng trong mỗi lần cập nhật, `epochs` là 100 là số lần huấn luyện, `validation_data = (x_val, y_val_epoch)` là dữ liệu `validation` và `callbacks` là `early_stopping` để áp dụng kỹ thuật dừng sớm.

3.2.7. Đánh giá mô hình

Hàm `evaluate` trong `keras` được sử dụng để trả về giá trị mất mát và độ chính xác của mô hình trên tập dữ liệu kiểm tra. Trong hàm `evaluate` là `x-test` là dữ liệu đầu vào, `y_test_epoch` là nhãn đã được chuyển one-hot encoding, `verbose=1` dùng để hiển thị thanh tiến trình và số liệu đánh giá. Cuối cùng, thực hiện in ra giá trị mất mát và độ chính xác.

3.2.8. Biểu đồ hiển thị độ chính xác

Để vẽ biểu đồ hiển thị độ chính xác thì tôi sử dụng thư viện `matplotlib` để vẽ biểu đồ. Đầu tiên, sử dụng `hist.history['accuracy']` để chứa danh sách các giá trị độ chính xác của mô hình trên tập huấn luyện và đặt màu của đường biểu diễn là màu xanh lá cây (`green`), sử dụng `hist.history['val_accuracy']` chứa danh sách các giá trị độ chính xác của mô hình trên tập kiểm tra và đặt màu của đường biểu diễn là màu đỏ (`red`). Đặt tiêu đề là `Model Accuracy`. Trục y đặt là `Accuracy`. Trục x đặt là `Epoch`, tạo 1 chú thích đặt ở phía trái bên biểu đồ và cuối cùng là in ra biểu đồ.

3.2.9. Biểu đồ hiển thị giá trị mất mát

Để vẽ biểu đồ hiển thị độ chính xác thì tôi sử dụng thư viện `matplotlib` để vẽ biểu đồ. Đầu tiên, sử dụng `hist.history['accuracy']` để chứa danh sách các giá trị mất mát của mô hình trên tập kiểm tra và đặt màu của đường biểu diễn là màu xanh lá cây (`green`), sử dụng `hist.history['val_accuracy']` chứa danh sách các giá trị mất mát của mô hình trên tập huấn luyện và đặt màu của đường biểu diễn là màu đỏ (`red`). Đặt tiêu

đề là Model Loss. Trục y đặt là Accuracy. Trục x đặt là Epoch, tạo 1 chú thích đặt ở phía bên phải biểu đồ và cuối cùng là in ra biểu đồ.

3.2.10. Hiển thị hình ảnh và kết quả dự đoán

Đầu tiên, sử dụng hàm `imread` để đọc hình ảnh từ tệp `'sample_data/bird.jpg'`. Sau đó hiển thị ảnh gốc với tiêu đề `'Original Image'`. Tiếp theo, thực hiện `resize` hình ảnh để phù hợp với dữ liệu ban đầu và hiển thị hình ảnh sau khi `resize` với tiêu đề `'Resized Image'`. Cuối cùng, sẽ sử dụng hàm `predict` để trả về mảng chứa xác suất của các lớp. Sau đó, sử dụng `argmax` để tìm ra giá trị lớn nhất của các phần tử có trong mảng và sẽ in ra kết quả dự đoán.

3.3. Hướng dẫn sử dụng

Để thực hiện phân loại ảnh cần thực hiện các bước sau:

- Bước 1: Đầu tiên là thực thi thư viện.
- Bước 2: Thực thi phân tải dữ liệu từ tập dữ liệu `cifar-10`.
- Bước 3: Thực thi phân chia dữ liệu.
- Bước 4: Thực thi các tiền xử lý dữ liệu như chuẩn hóa dữ liệu, chuyển nhãn `one-hot encoding` và tăng cường dữ liệu.
- Bước 5: Thực thi xây dựng mô hình.
- Bước 6: Tải mô hình đã huấn luyện.
- Bước 7: Click vào dấu ba chấm của tệp `sample_data` để chọn hình ảnh, `resize` kích thước hình ảnh để phù hợp kích thước dữ liệu ban đầu và thực hiện dự đoán và hiển thị hình ảnh.

3.4. Mục tiêu cải tiến

Trong quá trình thực đề tài “phân loại ảnh dựa trên kỹ thuật học sâu” thì sau khi huấn luyện mô hình và đánh giá mô hình thì độ chính xác chỉ 65.12% và vẽ biểu đồ độ chính xác và biểu đồ giá trị mất mát thì xảy ra `overfitting`. Sau đó, thực hiện tăng cường dữ liệu, sử dụng `earlystopping` để dừng huấn luyện khi độ chính xác trên tập `validation` không còn cải thiện, giúp tránh `overfitting` và tiết kiệm thời gian huấn luyện thì độ chính xác cải thiện là 73.34%.

CHƯƠNG 4. KẾT QUẢ NGHIÊN CỨU

4.1. Kết quả thực hiện

4.1.1. Kết quả hiển thị thông tin dữ liệu

Sau đây là kết quả hiển thị thông tin sau khi tải dữ liệu và chia dữ liệu. Trong đó, tập train có xTrain shape có 40000 hình ảnh đào tạo với kích thước là 32x32 và 3 kênh màu và yTrain shape có 40000 hình ảnh đào tạo và 1 là số lượng nhãn đại diện cho mỗi phần tử trong tập dữ liệu huấn luyện. Trong tập test có yTrain shape 10000 hình ảnh kiểm tra với kích thước là 32x32 và 3 kênh màu và yTest shape có 10000 hình ảnh kiểm tra và 1 là số lượng nhãn đại diện cho mỗi phần tử trong tập dữ liệu huấn luyện. Trong tập validation có xVal có 40000 hình ảnh đào tạo với kích thước là 32x32 và 3 kênh màu và yVal shape có 40000 hình ảnh đào tạo và 1 là số lượng nhãn đại diện cho mỗi phần tử trong tập dữ liệu huấn luyện

```
xTrain shape : (40000, 32, 32, 3)
yTrain shape : (40000, 1)
xTest shape : (10000, 32, 32, 3)
yTest shape : (10000, 1)
xVal shape : (10000, 32, 32, 3)
yVal shape : (10000, 1)
```

Hình 4.1 Kết quả hiển thị thông tin dữ liệu

4.1.2. Kết quả xây dựng kiến trúc mô hình

Sau đây là kết quả xây dựng kiến trúc mô hình. Output Shape của conv2d (Conv2D) là (None, 32,32, 32) None là batch size, là số lượng mẫu đầu vào không xác định trước, 32 số lượng bộ lọc (filter), kích thước hình ảnh là 32, 32 và Param là số lượng tham số 896. Output Shape của max_pooling 2d (MaxPooling2D) là (None, 16,16, 32) là batch size, tức là số lượng mẫu đầu vào không xác định trước, có 16, 16 do giảm kích thước sau khi thực hiện pooling, 32 là số lượng các kênh đặc trưng của ma trận đầu ra. Output Shape của conv2d_1 (Conv2D) là (None, 16, 16, 64) None là batch size, là số lượng mẫu đầu vào không xác định trước, 64 số lượng bộ lọc (filter), 16, 16 là đầu ra sau khi áp dụng MaxPooling2D và Param là số lượng tham số 18496. Output Shape của max_pooling 2d_1 (MaxPooling2D) là (None, 16, 16, 64) None là batch size, là số lượng mẫu đầu vào không xác định trước, có 8, 8 do giảm kích thước sau khi thực hiện pooling, 64 là số lượng các kênh đặc trưng của ma trận đầu ra.

Output Shape của conv2d_2 (Conv2D) là (None, 8, 8, 128) None là batch size, tức là số lượng mẫu đầu vào không xác định trước, 128 số lượng bộ lọc (filter), 8, 8 là đầu do sau khi áp dụng MaxPooling2D và Param là số lượng tham số 73856. Output Shape của max_pooling_2d_1 (MaxPooling2D) là (None, 4, 4, 128) None là batch size, là số lượng mẫu đầu vào không xác định trước, có 8, 8 do giảm kích thước sau khi thực hiện pooling, 128 là số lượng các kênh đặc trưng của ma trận đầu ra. Output shape của flatten (Flatten) là (None, 2048) là batch size, tức là số lượng mẫu đầu vào không xác định trước, 2048 là số chiều của vector đầu ra, tương ứng với số lượng phần tử trong ma trận đầu vào đã được làm phẳng (flatten) từ ma trận 2D thành vector 1D. Output shape của dense (Dense) là (None, 256) là batch size, tức là số lượng mẫu đầu vào không xác định trước, 256 là số lượng unit và Param là số lượng tham số 524544. Output Shape của dense_1 (Dense) là (None,10) là batch size, tức là số lượng mẫu đầu vào không xác định trước, 10 số lượng lớp đầu ra, tương ứng với 10 nhãn trong tập dữ liệu CIFAR-10 và Param là số lượng tham số 2570. Total params là tổng tham số trong mô hình là 620362. Trainable params là tham số có thể huấn luyện là 620362. Non-trainable params là tham số không thể huấn luyện là 0.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 32, 32, 32)	896
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_1 (Conv2D)	(None, 16, 16, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d_2 (Conv2D)	(None, 8, 8, 128)	73,856
max_pooling2d_2 (MaxPooling2D)	(None, 4, 4, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 256)	524,544
dense_1 (Dense)	(None, 10)	2,570

Total params: 620,362 (2.37 MB)
 Trainable params: 620,362 (2.37 MB)
 Non-trainable params: 0 (0.00 B)

Hình 4.2 Kết quả xây dựng kiến trúc mô hình

4.1.3. Kết quả huấn luyện mô hình

Sau đây là kết quả huấn luyện mô hình CNN. Trong đó, tổng số epoch = 100, 157 là tổng số batch cần thực hiện trong mỗi epoch, loss là mất mát trung bình trên tập huấn luyện trong epoch này cho thấy mức độ mất mát trên mỗi mẫu, accuracy là độ chính xác trung bình trên tập huấn luyện trong epoch này cho thấy tỉ lệ phần trăm các mẫu được phân loại đúng, val_loss là mất mát trung bình trên tập kiểm tra trong epoch này. Cho thấy mức độ mất mát trên mỗi mẫu trong tập kiểm tra, val_accuracy là độ chính xác trung bình trên tập kiểm tra trong epoch này, cho thấy tỉ lệ phần trăm các mẫu được phân loại đúng trên tập kiểm tra.

```
Epoch 1/100
157/157 [=====] - 4s 16ms/step - loss: 1.6873 - accuracy: 0.3909 - val_loss: 1.3976 - val_accuracy: 0.4910
Epoch 2/100
157/157 [=====] - 2s 14ms/step - loss: 1.2873 - accuracy: 0.5435 - val_loss: 1.1731 - val_accuracy: 0.5862
Epoch 3/100
157/157 [=====] - 2s 12ms/step - loss: 1.1090 - accuracy: 0.6115 - val_loss: 1.0641 - val_accuracy: 0.6248
Epoch 4/100
157/157 [=====] - 2s 12ms/step - loss: 0.9867 - accuracy: 0.6526 - val_loss: 1.0481 - val_accuracy: 0.6352
Epoch 5/100
157/157 [=====] - 2s 11ms/step - loss: 0.8989 - accuracy: 0.6894 - val_loss: 0.8975 - val_accuracy: 0.6910
Epoch 6/100
157/157 [=====] - 2s 12ms/step - loss: 0.8278 - accuracy: 0.7142 - val_loss: 0.9232 - val_accuracy: 0.6782
Epoch 7/100
157/157 [=====] - 2s 11ms/step - loss: 0.7654 - accuracy: 0.7347 - val_loss: 0.8744 - val_accuracy: 0.6983
Epoch 8/100
157/157 [=====] - 2s 12ms/step - loss: 0.7109 - accuracy: 0.7557 - val_loss: 0.8224 - val_accuracy: 0.7233
Epoch 9/100
157/157 [=====] - 2s 14ms/step - loss: 0.6649 - accuracy: 0.7690 - val_loss: 0.8655 - val_accuracy: 0.7009
Epoch 10/100
157/157 [=====] - 2s 12ms/step - loss: 0.6069 - accuracy: 0.7892 - val_loss: 0.8005 - val_accuracy: 0.7256
Epoch 11/100
157/157 [=====] - 2s 12ms/step - loss: 0.5576 - accuracy: 0.8077 - val_loss: 0.8435 - val_accuracy: 0.7172
Epoch 12/100
157/157 [=====] - 2s 12ms/step - loss: 0.5084 - accuracy: 0.8235 - val_loss: 0.8331 - val_accuracy: 0.7257
Epoch 13/100
157/157 [=====] - 2s 14ms/step - loss: 0.4672 - accuracy: 0.8383 - val_loss: 0.8011 - val_accuracy: 0.7393
Epoch 14/100
157/157 [=====] - 2s 13ms/step - loss: 0.4130 - accuracy: 0.8587 - val_loss: 0.8787 - val_accuracy: 0.7254
Epoch 15/100
157/157 [=====] - 2s 13ms/step - loss: 0.3789 - accuracy: 0.8704 - val_loss: 0.8241 - val_accuracy: 0.7408
```

Hình 4.3 Kết quả huấn luyện mô hình

4.1.4. Đánh giá mô hình

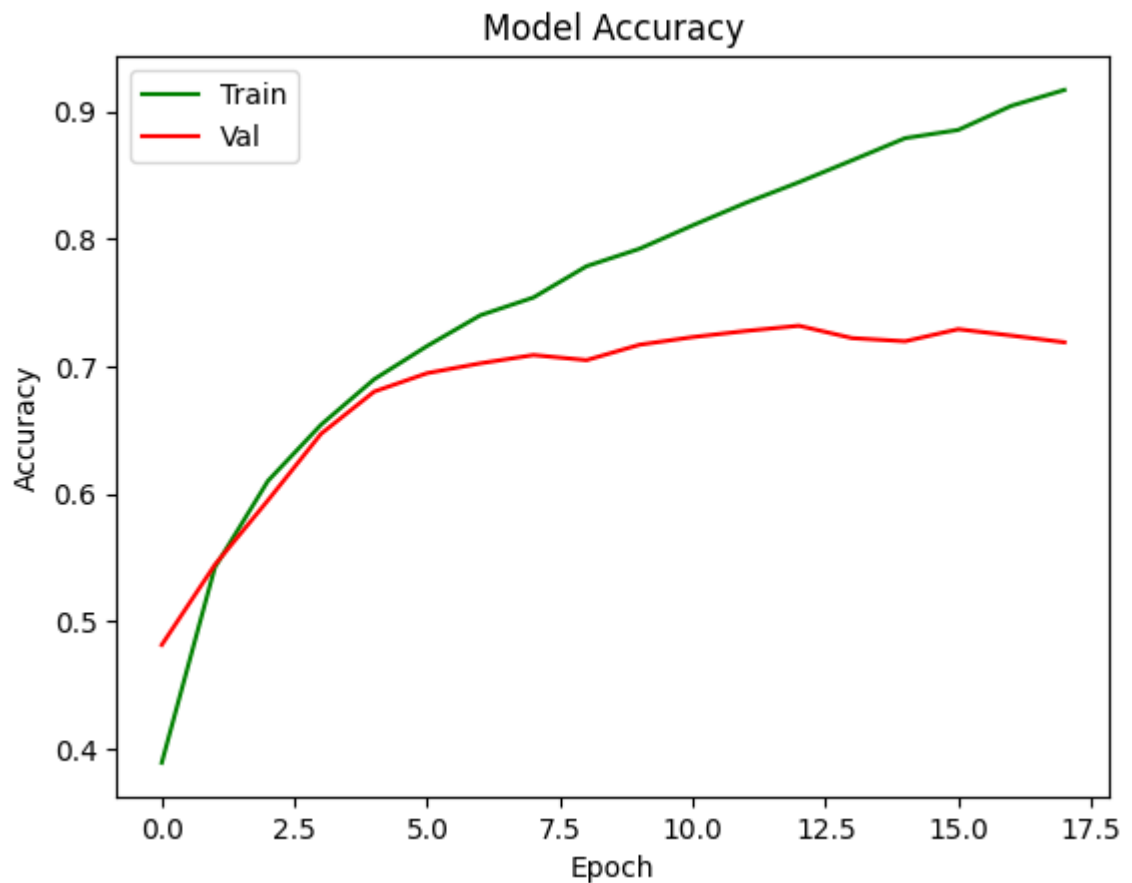
Sau đây là kết quả đánh giá mô hình. Độ chính xác là 72.69%.

```
313/313 ————— 1s 2ms/step - accuracy: 0.7280 - loss: 0.8507
Test Accuracy: 72.69%
```

Hình 4.4 Kết quả đánh giá mô hình

4.1.5. Biểu đồ hiển thị độ chính xác

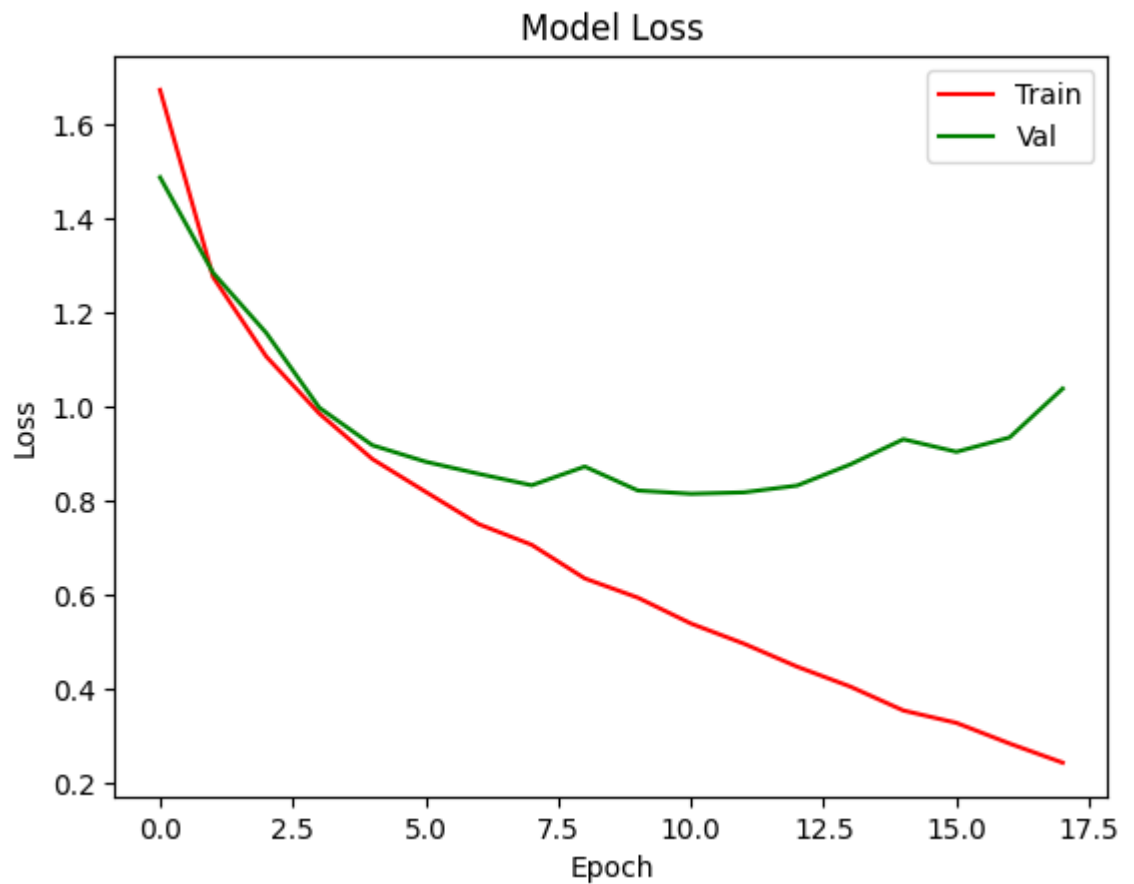
Biểu đồ sau đây dùng để hiển thị độ chính xác của mô hình. Trong đó, đường màu xanh lá là Train thể hiện độ chính xác trên tập huấn luyện. Đường màu xanh lá càng cao thì cho thấy việc học của mô hình càng tốt. Đường màu đỏ là Val thể hiện độ chính xác trên tập kiểm tra. Đường màu đỏ này tăng dần và tiến gần đến độ chính xác của tập huấn luyện, thì mô hình không overfitting và đang học được các đặc trưng tổng quát.



Hình 4.5 Biểu đồ hiển thị độ chính xác

4.1.6. Biểu đồ hiển thị mất mát

Biểu đồ sau đây dùng để hiển thị sự biến động của mất mát trên cả tập huấn luyện và tập kiểm tra qua mỗi epoch trong quá trình huấn luyện mô hình. Trong đó, đường màu đỏ là Train thể hiện mất mát trên tập huấn luyện. Đường này giảm dần và tiến gần đến 0 thì cho thấy mô hình đang học tốt trên tập huấn luyện. Đường màu xanh lá là Val thể hiện mất mát trên tập kiểm tra. Đường này giảm dần và tiến gần đến độ mất mát của tập huấn luyện, điều này có nghĩa là mô hình không bị quá mức (overfitting) và đang học được các đặc trưng tổng quát.



Hình 4.6 Biểu đồ hiển thị giá trị mất mát

4.1.7. Kết quả hiển thị hình ảnh và kết quả dự đoán

4.1.7.1. Kết quả hiển thị hình ảnh và kết quả dự đoán nhãn xe tải

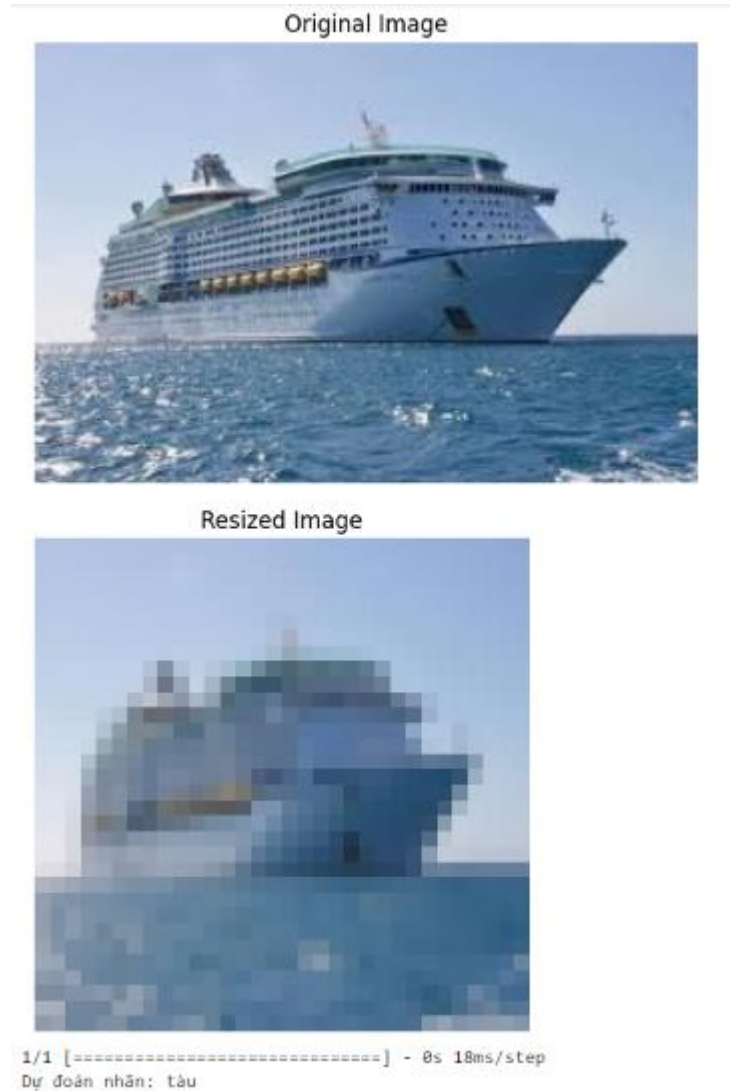
Sau đây là kết quả hiển thị hình ảnh và dự đoán nhãn xe tải. Original Image là hình ảnh xe tải đã tải từ internet về thiết bị. Resized Image là hình ảnh đã thực hiện resize về kích thước 32x32 và cuối cùng là kết quả.



Hình 4.7 Kết quả hiển thị hình ảnh và dự đoán xe tải

4.1.7.2. Kết quả hiển thị hình ảnh và dự đoán nhãn tàu

Sau đây là kết quả hiển thị hình ảnh và dự đoán nhãn tàu. Original Image là hình ảnh tàu tải từ internet về thiết bị. Resized Image là hình ảnh mà đã thực hiện resize về kích thước 32x32 và cuối cùng là kết quả.



Hình 4.8 Kết quả hiển thị hình ảnh và dự đoán nhãn tàu

4.1.7.3. Kết quả hiển thị hình ảnh và dự đoán nhãn con ngựa

Sau đây là kết quả hiển thị hình ảnh và dự đoán nhãn con ngựa. Original Image là hình ảnh con ngựa tải từ internet về thiết bị. Resized Image là hình ảnh đã thực hiện resize về kích thước 32x32 và cuối cùng là kết quả.



Hình 4.9 Kết quả hiển thị hình ảnh và dự đoán nhãn con ngựa

4.1.7.4. Kết quả hiển thị hình ảnh và dự đoán nhãn con chó

Sau đây là kết quả hiển thị hình ảnh và dự đoán nhãn con chó. Original Image là hình ảnh con chó tải từ internet về thiết bị. Resized Image là hình ảnh đã thực hiện resize về kích thước 32x32 và cuối cùng là kết quả.



Hình 4.10 Kết quả hiển thị hình ảnh và dự đoán nhãn con chó

4.1.7.5. Kết quả hiển thị hình ảnh và dự đoán nhãn con ếch

Sau đây là kết quả hiển thị hình ảnh và dự đoán nhãn con ếch. Original Image là hình ảnh con ếch tải từ internet về thiết bị. Resized Image là hình ảnh đã thực hiện resize về kích thước 32x32 và cuối cùng là kết quả.



Hình 4.11 Kết quả hiển thị hình ảnh và dự đoán nhãn con ếch

4.1.7.6. Kết quả hiển thị hình ảnh và dự đoán máy bay

Sau đây là kết quả hiển thị hình ảnh và dự đoán nhãn máy bay. Original Image là hình ảnh máy bay tải từ internet về thiết bị. Resized Image là hình ảnh đã thực hiện resize về kích thước 32x32 và cuối cùng là kết quả.



Hình 4.12 Kết quả hiển thị hình ảnh và dự đoán nhãn máy bay

4.1.7.7. Kết quả hiển thị hình ảnh và dự đoán nhãn con chim

Sau đây là kết quả hiển thị hình ảnh và dự đoán nhãn con chim. Original Image là hình ảnh con chim tải từ internet về thiết bị. Resized Image là hình ảnh đã thực hiện resize về kích thước 32x32 và cuối cùng là kết quả.



Hình 4.13 Kết quả hiển thị hình ảnh và dự đoán nhãn con chim

4.1.7.8. Kết quả hiển thị hình ảnh và dự đoán nhãn con nai

Sau đây là kết quả hiển thị hình ảnh và dự đoán nhãn con nai. Original Image là hình ảnh con nai tải từ internet về thiết bị. Resized Image là hình ảnh đã thực hiện resize về kích thước 32x32 và cuối cùng là kết quả.



Hình 4.14 Kết quả hiển thị hình ảnh và dự đoán nhãn con nai

4.1.7.9. Kết quả hiển thị hình ảnh và dự đoán nhãn xe ô tô

Sau đây là kết quả hiển thị hình ảnh và dự đoán nhãn xe ô tô. Original Image là hình ảnh xe ô tô tải từ internet về thiết bị. Resized Image là hình ảnh đã thực hiện resize về kích thước 32x32 và cuối cùng là kết quả.

Original Image



Resized Image



1/1 [=====] - 0s 74ms/step
Dự đoán nhãn: ô tô

Hình 4.15 Kết quả hình ảnh và dự đoán nhãn xe ô tô

4.1.7.10. Kết quả hiển thị hình ảnh và dự đoán nhãn mèo

Sau đây là kết quả hiển thị hình ảnh và dự đoán nhãn xe con mèo. Original Image là hình ảnh con mèo tải từ internet về thiết bị. Resized Image là hình ảnh đã thực hiện resize về kích thước 32x32 và cuối cùng là kết quả.



Hình 4.16 Kết quả hiển thị hình ảnh và dự đoán nhãn con mèo

4.2.

4.3. So sánh kết quả trước và sau khi cải thiện mô hình

Sau đây là kết quả dự đoán trước khi cải thiện và sau khi cải thiện. Bên trái là kết quả dự đoán trước khi cải thiện với độ chính xác là 65.12% và bên phải là kết quả dự đoán sau khi cải thiện với độ chính xác là 73.34%. Để cải thiện thì cần tăng cường dữ liệu. Sử dụng earlystopping dừng huấn luyện khi độ chính xác trên tập validation không còn cải thiện, giúp tránh overfitting và tiết kiệm thời gian huấn luyện.

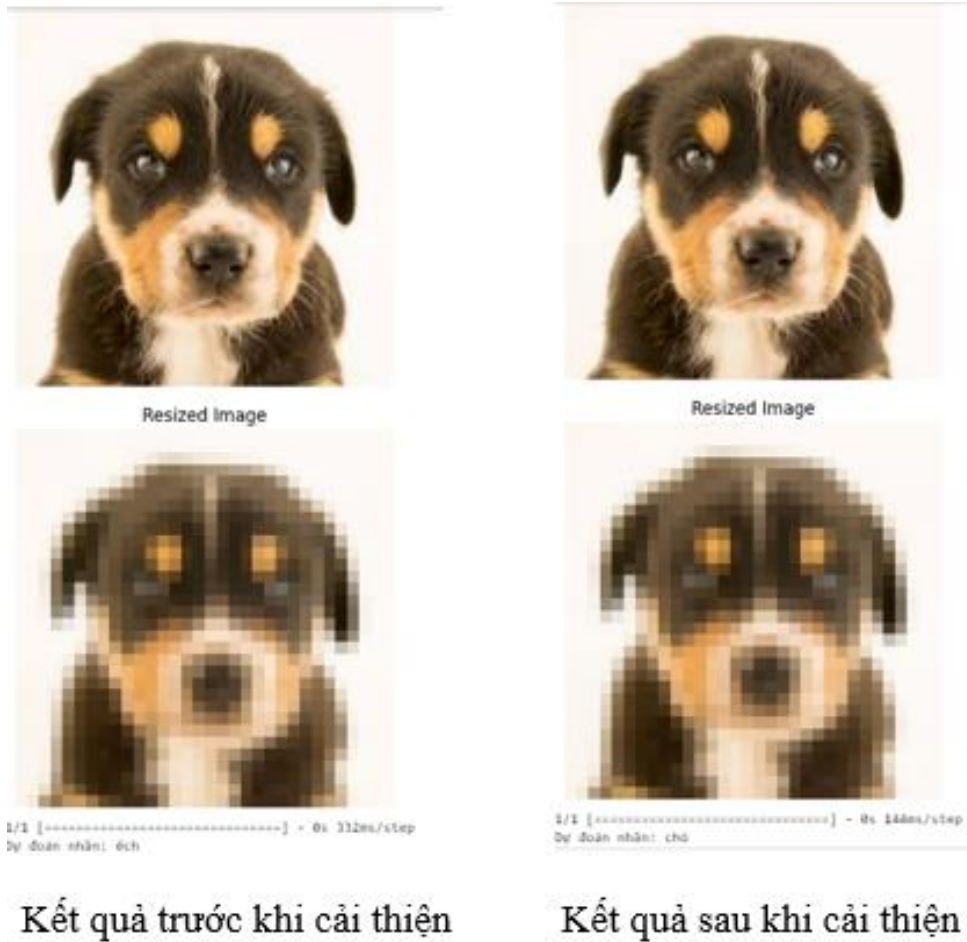
Sau đây là một vài kết quả so sánh kết quả trước và sau khi cải thiện:

- Đây là kết quả dự đoán. Hình ảnh bên trái hình ảnh trước khi cải thiện còn hình phải là hình ảnh sau khi cải thiện. Hình ảnh bên trái hình ảnh trước khi cải thiện với kết quả dự đoán là con nai và hình ảnh bên phải hình ảnh sau khi cải thiện với kết quả dự đoán là con mèo.



Hình 4.17 Kết quả hình ảnh trước và sau khi cải thiện với hình ảnh con mèo

- Đây là kết quả dự đoán. Hình ảnh bên trái hình ảnh trước khi cải thiện còn hình phải là hình ảnh sau khi cải thiện. Hình ảnh bên trái hình ảnh trước khi cải thiện với kết quả dự đoán là con ếch và hình ảnh bên phải hình ảnh sau khi cải thiện với kết quả dự đoán là con chó.

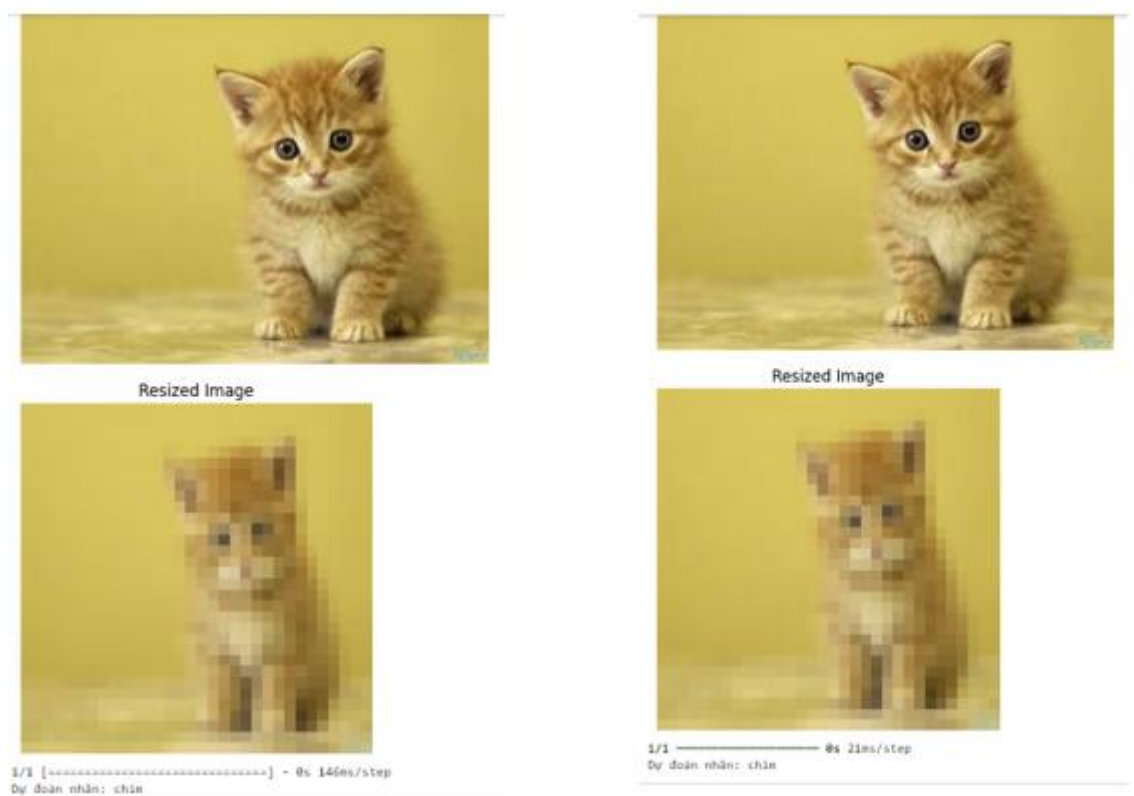


Hình 4.18 Kết quả hình ảnh trước và sau khi cải thiện với hình ảnh con chó

Ngoài ra, có một số trường hợp là cả hai mô hình đều dự đoán sai. Vì mô hình thứ hai sau khi cải thiện là 73,34% mà không phải là 100% nên có một số trường hợp là cả hai mô hình đều dự đoán sai.

Sau đây là các kết quả cho thấy cả hai mô hình đưa ra kết quả dự đoán sai.

- Đây là kết quả dự đoán. Hình ảnh bên trái hình ảnh trước khi cải thiện còn hình phải là hình ảnh sau khi cải thiện. Hình ảnh bên trái hình ảnh trước khi cải thiện với kết quả dự đoán là con chim và hình ảnh bên phải hình ảnh sau khi cải thiện với kết quả dự đoán là con chim. Kết quả là cả hai mô hình đều dự đoán sai.



Hình 4.19 Kết quả cả hai mô hình dự đoán sai nhãn con mèo

- Đây là kết quả dự đoán. Hình ảnh bên trái hình ảnh trước khi cải thiện còn hình phải là hình ảnh sau khi cải thiện. Hình ảnh bên trái hình ảnh trước khi cải thiện với kết quả dự đoán là con nai và hình ảnh bên phải hình ảnh sau khi cải thiện với kết quả dự đoán là con ngựa. Kết quả là cả hai mô hình đều dự đoán sai.



Kết quả trước khi cải thiện

Kết quả sau khi cải thiện

Hình 4.20 Kết quả cả hai mô hình dự đoán sai nhãn con chó

CHƯƠNG 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

Kết quả đạt được trong quá trình thực hiện đề tài “phân loại ảnh dựa trên kỹ thuật học sâu” là xây dựng thành công mô hình CNN và qua đó biết được kiến trúc mô hình, thực hiện huấn luyện mô hình CNN, đánh giá mô hình để biết độ chính xác và giá trị mất mát, vẽ biểu đồ hiển thị độ chính xác và biểu đồ hiển thị giá trị mất mát, đưa ra dự đoán về hình ảnh để biết được hình ảnh đó thuộc nhóm nào.

5.2. Hướng phát triển

Hướng phát triển của đề tài “phân loại ảnh dựa trên kỹ thuật học sâu” là tải thêm các tập dữ liệu với các hình ảnh khác nhau để nhận dạng và phân loại nhiều hình ảnh khác nhau. Hiển thị phần trăm dự đoán hình ảnh để có thể kết quả dự đoán hình ảnh được bao nhiêu phần trăm. Tăng tốc độ huấn luyện để quá trình huấn luyện mô hình huấn luyện nhanh và không tốn thời gian.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] <https://www.geeksforgeeks.org/introduction-deep-learning/?ref=shm>. (truy cập ngày 24 tháng 5 năm 2024)
- [2] <https://www.techtarget.com/searchenterpriseai/definition/convolutional-neural-network>. (truy cập ngày 20 tháng 5 năm 2024)
- [3] Phung, Rhee, "A High-Accuracy Model Average Ensemble of Convolutional Neural Networks for Classification of Cloud Image Patches on Small Datasets", Applied Sciences, tập 9, số 21, 2019.
- [4] <https://www.kaggle.com/discussions/getting-started/338611>. (truy cập ngày 25 tháng 5 năm 2024)
- [5] <https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/> (truy cập ngày 28 tháng 5 năm 2024)
- [6] <https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/>. (truy cập ngày 26 tháng 4 năm 2024)
- [7] <https://www.matbao.net/tin-tuc/google-colab-la-gi-kham-pha-nen-tang-so-ghi-chep-tinh-toan-truc-tuyen-134477.html>. (truy cập ngày 1 tháng 5 năm 2024)
- [8] <https://websitechuyennghiep.vn/python-la-gi.html>. (truy cập ngày 25 tháng 5 năm 2024)
- [9] <https://www.geeksforgeeks.org/what-is-keras/> (truy cập ngày 28 tháng 5 năm 2024)
- [10] <https://topdev.vn/blog/tensorflow-la-gi/#5-natural-language-processing-with-tensorflow-teach-language-to-machines-using-pythons-deep-learning-library> (truy cập ngày 28 tháng 5 năm 2024)
- [11] <https://ivyproschoool.com/blog/what-is-scikit-learn/> (truy cập ngày 28 tháng 5 năm 2024)
- [12] <https://www.edureka.co/blog/what-is-numpy-in-python/> (truy cập ngày 28 tháng 5 năm 2024)

- [13] <https://www.geeksforgeeks.org/python-introduction-matplotlib/> (truy cập ngày 28 tháng 5 năm 2024)
- [14] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [15] J. Schmidhuber, “Deep learning in neural networks: an overview,” *Neural Networks*, vol. 61, no. 1, pp. 85–117, 2015
- [16] Bùi Thanh Hùng, Phạm Hoàng Phương, “tìm kiếm hình ảnh bằng phương pháp học sâu”, *Kỷ yếu Hội nghị KHCN Quốc gia lần thứ XIV về Nghiên cứu cơ bản và ứng dụng Công nghệ thông tin (FAIR)*, 2021.
- [17] Đặng Vũ Kim Ký - Nguyen Huỳnh Thông - Âu Thùy Ân - Phạm Tấn Hưng, “nghiên cứu ứng dụng học sâu vào phân loại rác thải”, *tạp chí công thương*, số 3, tr. 218 – 224, 2022.
- [18] Võ Đức Quang, Nguyễn Hải Yến, Mai Hồng Mận, Nguyễn Thị Nhã, “ứng dụng kỹ thuật học sâu trong hỗ trợ chẩn đoán bệnh viêm phổi thông qua ảnh chụp x-quang”, *Tạp chí khoa học*, tập 50, số 4, tr. 29-39, 2021.
- [19] <https://www.cs.toronto.edu/~kriz/cifar.html> (truy cập ngày 29 tháng 4 năm 2024)