Ảnh có chứa quảng trường

Mô tả được tạo tự động**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

=====\*\*\*=====

A picture containing logo, symbol, font, graphics

Description automatically generated

**BÁO CÁO CUỐI KỲ**

**IOT VÀ ỨNG DỤNG**

**ĐỀ TÀI**: **NGHIÊN CỨU MACHINE LEARNING**

**MODEL PIPELINES**

**Giảng viên hướng dẫn:** Th.S Đàm Minh Lịnh

**NHÓM SINH VIÊN THỰC HIỆN Nhóm 16**

**HỌ VÀ TÊN MSSV**

Nguyễn Thị Nhung N20DCCN052

Nguyễn Thanh Trúc N20DCCN158

**TP. Hồ Chí Minh, 5 tháng 1 năm 2024**

**LỜI CẢM ƠN**

Lời đầu tiên chúng em xin gửi lời cảm ơn chân thành đến Thầy Đàm Minh Lịnh đã đồng hành cùng chúng em trong suốt học kỳ vừa qua và trang bị cho chúng em thêm nhiều kiến thức bổ ích thông qua môn IOT Và Ứng Dụng.

Trong quá trình học tập và tìm hiểu chúng em đã nỗ lực rất nhiều với mong muốn hoàn thành một cách tốt nhất, nhưng vì kiến thức của nhóm còn hạn chế nên không thể không có những sai sót. Chúng em mong Thầy thông cảm, dành thời gian xem qua báo cáo và đưa ra những lời góp ý để chúng em có thể tích lũy thêm kinh nghiệm.

Sau cùng, chúng em xin chân thành cảm ơn Thầy, chúc Thầy sức khoẻ, luôn hạnh phúc và thành công hơn nữa trong công việc cũng như trong cuộc sống.

**Sinh viên thực hiện**

Nguyễn Thị Nhung

Nguyễn Thanh Trúc

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN** ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ........................................................................................................................................... ...........................................................................................................................................

TPHCM, Ngày …. Tháng …. Năm 2024

**Giảng viên hướng dẫn**

**MỤC LỤC**

[I. TỔNG QUAN 6](#_Toc155191440)

[1.1 Giới thiệu về đề tài 6](#_Toc155191441)

[1.2 Mục tiêu và tầm quan trọng 6](#_Toc155191442)

[II. MÔ HÌNH ĐỀ XUẤT 7](#_Toc155191443)

[2.1 Ngôn ngữ và Thư viện 7](#_Toc155191444)

[2.1.1 Ngôn ngữ Python 7](#_Toc155191445)

[2.1.1.1 Khái niệm 7](#_Toc155191446)

[2.1.1.2 Lịch sử Python 7](#_Toc155191447)

[2.1.1.3 Đặc điểm Python 7](#_Toc155191448)

[2.1.1.4 Ứng dụng của Python 8](#_Toc155191449)

[2.1.2 Thư viện Python 9](#_Toc155191450)

[2.1.2.1 Thư viện TensorFlow 9](#_Toc155191451)

[2.1.2.2 Thư viện Matplotlib 9](#_Toc155191452)

[2.1.2.3 Thư viện Pandas 9](#_Toc155191453)

[2.1.2.4 Thư viện Numpy 9](#_Toc155191454)

[2.1.2.5 Thư viện Tkinter 9](#_Toc155191455)

[2.2 Tổng quan về Deep Learning 11](#_Toc155191456)

[2.3 Tìm hiểu Machine Learning Pipeline 13](#_Toc155191457)

[2.3.1 Khái Niệm 13](#_Toc155191458)

[2.3.2 Lợi ích của Machine Learning Pipeline 14](#_Toc155191459)

[2.4 Tìm hiểu về Data Preprocessing 15](#_Toc155191460)

[2.5 Tìm hiểu về Data Validation 16](#_Toc155191461)

[2.6 Tìm hiểu về Feature Engineering 17](#_Toc155191462)

[2.6.1 Giới thiệu về Feature Engineering 17](#_Toc155191463)

[2.6.2 Sự cần thiết của Feature Engineering trong học máy 18](#_Toc155191464)

[2.6.3 Các quy trình liên quan đến Feature Engineering 18](#_Toc155191465)

[2.7 Tìm hiểu về Mô hình CNN 19](#_Toc155191466)

[2.8 Tìm hiểu về Model Evaluation 22](#_Toc155191467)

[III. TRIỂN KHAI 23](#_Toc155191468)

[3.1 Thu Thập Dữ Liệu 23](#_Toc155191469)

[3.2 Xử Lý Dữ Liệu 24](#_Toc155191470)

[3.3 Xây dựng model CNN 26](#_Toc155191471)

[3.4 Huấn luyện model CNN 26](#_Toc155191472)

[3.5 Kiểm thử model 29](#_Toc155191473)

[3.6 Demo 30](#_Toc155191474)

[IV. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 32](#_Toc155191475)

[4.1 Kết quả và hạn chế 32](#_Toc155191476)

[4.1.1 Kết quả 32](#_Toc155191477)

[4.1.2 Hạn chế 32](#_Toc155191478)

[4.2 Hướng phát triển 33](#_Toc155191479)

**DANH MỤC HÌNH ẢNH**

[Hình 2.1 Ứng dụng Python 8](#_Toc155191496)

[Hình 2.2 Thư viện Python 11](#_Toc155191497)

[Hình 2.3 Deep Learning 12](#_Toc155191498)

[Hình 2.4 Machine Learning Pipeline 13](#_Toc155191499)

[Hình 2.5 Data Processing 16](#_Toc155191500)

[Hình 2.6 Data Validation 17](#_Toc155191501)

[Hình 2.7 Kiến trúc CNN đơn giản 19](#_Toc155191502)

[Hình 2.8 Pooling layer 21](#_Toc155191503)

[Hình 2.9 Lớp Fully Connected Layers 21](#_Toc155191504)

[Hình 3.1 Bộ dữ liệu GTSRB 23](#_Toc155191506)

[Hình 3.2 Kết quả huấn luyện model 28](#_Toc155191507)

[Hình 3.3 Đồ thị hiển thị giá trị chính xác của model 28](#_Toc155191508)

[Hình 3.4 Đồ thị hiển thị giá trị mất mát của model 29](#_Toc155191509)

[Hình 3.5 Giao diện để load hình ảnh 30](#_Toc155191510)

[Hình 3.6 Giao diện đã load hình ảnh 30](#_Toc155191511)

[Hình 3.7 Giao diện kết quả 31](#_Toc155191512)

1. **TỔNG QUAN**
   1. **Giới thiệu về đề tài**

Đề tài "Nghiên cứu về Machine Learning Model Pipeline" là quá trình tập trung vào xây dựng một chuỗi các bước hành động cụ thể để đạt được hiệu suất và linh hoạt cao trong quá trình làm việc với máy học. Trong ngữ cảnh này, "pipeline" đề cập đến một dãy các công đoạn liên kết chặt chẽ, bao gồm thu thập và tiền xử lý dữ liệu, xây dựng và tinh chỉnh mô hình, đánh giá hiệu suất, và cuối cùng là triển khai mô hình vào môi trường thực tế.

* 1. **Mục tiêu và tầm quan trọng**

Đề tài này đặt ra mục tiêu là tối ưu hóa mỗi bước trong chuỗi làm việc với máy học, nhằm đảm bảo rằng mô hình phát triển không chỉ đạt được hiệu suất cao mà còn có thể tích hợp linh hoạt vào các hệ thống thực tế. Qua quá trình nghiên cứu, có thể tìm ra các chiến lược tự động hóa, công cụ hỗ trợ, và phương pháp tối ưu hóa để gia tăng khả năng triển khai của mô hình máy học.

Tầm quan trọng của đề tài nằm ở việc giải quyết những thách thức thực tế khi triển khai mô hình máy học và tối ưu hóa chuỗi công việc từ khi có dữ liệu đến khi ứng dụng thực sự đi vào sử dụng. Điều này làm cho đề tài trở thành một phần quan trọng trong lĩnh vực trí tuệ nhân tạo và khoa học dữ liệu, mang lại sự hiểu biết sâu sắc về quá trình thực hiện máy học trong các dự án thực tế.

1. **MÔ HÌNH ĐỀ XUẤT**
   1. **Ngôn ngữ và Thư viện**
      1. **Ngôn ngữ Python**
         1. **Khái niệm**

Python là ngôn ngữ lập trình cấp cao, đa năng và thông dịch được sử dụng trong nhiều lĩnh vực khác nhau bao gồm học máy, trí tuệ nhân tạo, phân tích dữ liệu, phát triển web, v.v. Python được biết đến vì tính dễ sử dụng*,* thư viện tiêu chuẩn mạnh mẽ và ngữ nghĩa động.Nó cũng có một cộng đồng lớn các nhà phát triển luôn đóng góp cho sự phát triển của nó. Trọng tâm chính đằng sau việc tạo ra nó là giúp các nhà phát triển đọc và hiểu dễ dàng hơn, đồng thời giảm bớt các dòng mã [1].

* + - 1. **Lịch sử Python**

Python, được tạo ra lần đầu tiên vào những năm 1980 bởi Guido van Rossum**,** là một trong những ngôn ngữ lập trình phổ biến nhất. Trong quá trình nghiên cứu tại Viện Nghiên cứu Toán học và Khoa học Máy tính Quốc gia Hà Lan**,** ông đã tạo ra Python – một ngôn ngữ lập trình siêu dễ đọc và sử dụng. Phiên bản đầu tiên được phát hành vào năm 1991 chỉ có một số loại dữ liệu tích hợp và chức năng cơ bản [1].

Sau đó, khi nó trở nên phổ biến trong giới khoa học về tính toán số và phân tích dữ liệu, vào năm 1994, Python 1.0 đã được phát hành với các tính năng bổ sung như bản đồ, lambda và chức năng lọc. Tiếp theo đó là việc bổ sung các chức năng mới và đưa các phiên bản Python mới hơn trở thành mốt [1].

* + - 1. **Đặc điểm Python**

Python có rất nhiều tính năng khiến nó trở nên khắt khe nhất và phổ biến hơn [1].

* Dễ đọc và dễ hiểu
* Ngôn ngữ thông dịch
* Ngôn ngữ lập trình hướng đối tượng
* Nguồn mở và miễn phí
* Đa năng và có thể mở rộng
* Đa nền tảng
* Hàng trăm thư viện và framework
* Linh hoạt, hỗ trợ GUI
* Được gõ động
* Cộng đồng lớn và năng động
  + - 1. **Ứng dụng của Python**

Python là ngôn ngữ lập trình có yêu cầu cao và phổ biến được sử dụng trong nhiều ngành khác nhau. Một số ứng dụng của Python là:

* Các nhà phát triển có thể làm việc về phát triển web và tạo ra các trang web tuyệt vời và giàu tính năng bằng cách sử dụng Python và các framework của nó như Django. Python đi kèm với các thư viện và công cụ sẵn có giúp quá trình phát triển ứng dụng web trở nên dễ dàng hơn [1].
* Các nhà khoa học dữ liệu có thể xây dựng các mô hình AI mạnh mẽ bằng cách sử dụng đoạn mã Python. Do tính năng dễ hiểu, nó cho phép các nhà phát triển viết các thuật toán phức tạp [1].
* Bạn cũng có thể sử dụng Python để làm việc trên các thiết kế CAD (thiết kế có sự hỗ trợ của máy tính), để tạo các mô hình 2D và 3D bằng kỹ thuật số [1].
* Sử dụng các thư viện như Pandas và TensorFlow, các chuyên gia có thể làm việc trên các ứng dụng phân tích dữ liệu và học máy để phân tích thống kê, thao tác dữ liệu, v.v [1].
* Python cũng có thể được các nhà phát triển sử dụng để xây dựng trò chơi bằng Pygame nhằm phát triển trò chơi 2D và 3D. Một số trò chơi phổ biến được xây dựng bằng Python là Cướp biển vùng Caribbean, Battlefield 2, v.v [1].



Hình 2.1 Ứng dụng Python

* + 1. **Thư viện Python**

Một số thư viện phổ biến của Python

* + - 1. **Thư viện TensorFlow**

Là một thư viện mã nguồn mở được sử dụng để tính toán cấp cao. Nó cũng được sử dụng trong các thuật toán học máy và học sâu. Nó chứa một số lượng lớn các phép toán tensor [2].

* + - 1. **Thư viện Matplotlib**

Thư viện này chịu trách nhiệm vẽ đồ thị dữ liệu số. Và đó là lý do tại sao nó được sử dụng trong phân tích dữ liệu. Nó cũng là một thư viện mã nguồn mở và vẽ các số liệu có độ xác định cao như biểu đồ hình tròn, biểu đồ, biểu đồ phân tán, đồ thị, v.v [2].

* + - 1. **Thư viện Pandas**

Pandas là một thư viện quan trọng đối với các nhà khoa học dữ liệu. Nó là một thư viện máy học mã nguồn mở cung cấp các cấu trúc dữ liệu cấp cao linh hoạt và nhiều công cụ phân tích khác nhau. Nó giúp giảm bớt việc phân tích dữ liệu, thao tác dữ liệu và làm sạch dữ liệu. Pandas hỗ trợ các hoạt động như Sắp xếp, Lập chỉ mục lại, Lặp lại, Ghép nối, Chuyển đổi dữ liệu, Trực quan hóa, Tổng hợp, v.v [2].

* + - 1. **Thư viện Numpy**

Đây là thư viện máy học phổ biến hỗ trợ các ma trận lớn và dữ liệu đa chiều. Nó bao gồm các hàm toán học được xây dựng sẵn để tính toán dễ dàng. Ngay cả các thư viện như TensorFlow cũng sử dụng Numpy nội bộ để thực hiện một số thao tác trên tensor. Giao diện mảng là một trong những tính năng chính của thư viện này [2].

* + - 1. **Thư viện Tkinter**

**Giao diện đồ hoạ người dùng(GUI)** là một dạng giao diện người dùng cho phép người dùng tương tác với máy tính thông qua các chỉ báo trực quan bằng cách sử dụng các mục như biểu tượng, menu, cửa sổ, v.v. Nó có ưu điểm hơn Giao diện dòng lệnh (CLI) nơi người dùng tương tác với máy tính bằng cách chỉ viết lệnh bằng bàn phím và việc sử dụng nó khó hơn GUI.

Tkinter là mô-đun python sẵn có được sử dụng để tạo các ứng dụng GUI. Đây là một trong những mô-đun được sử dụng phổ biến nhất để tạo các ứng dụng GUI trong Python vì nó đơn giản và dễ làm việc. Nó cung cấp giao diện hướng đối tượng cho bộ công cụ GUI Tk.   
Một số Thư viện Python khác có sẵn để tạo các ứng dụng GUI của riêng chúng ta là

* Kivy
* Python Qt
* WxPython

Một số trường hợp sử dụng phổ biến cho Tkinter:

* Tạo cửa sổ và hộp thoại: Tkinter có thể được sử dụng để tạo cửa sổ và hộp thoại cho phép người dùng tương tác với chương trình của bạn. Chúng có thể được sử dụng để hiển thị thông tin, thu thập đầu vào hoặc đưa ra các tùy chọn cho người dùng.
* Xây dựng GUI cho ứng dụng máy tính để bàn: Tkinter có thể được sử dụng để tạo giao diện cho ứng dụng máy tính để bàn, bao gồm các nút, menu và các thành phần tương tác khác.
* Thêm GUI vào chương trình dòng lệnh: Tkinter có thể được sử dụng để thêm GUI vào chương trình dòng lệnh, giúp người dùng tương tác với chương trình và các đối số đầu vào dễ dàng hơn.
* Tạo các tiện ích tùy chỉnh: Tkinter bao gồm nhiều tiện ích tích hợp sẵn, chẳng hạn như nút, nhãn và hộp văn bản, nhưng nó cũng cho phép bạn tạo các tiện ích tùy chỉnh của riêng mình.
* Tạo nguyên mẫu GUI: Tkinter có thể được sử dụng để nhanh chóng tạo nguyên mẫu GUI, cho phép bạn thử nghiệm và lặp lại các ý tưởng thiết kế khác nhau trước khi thực hiện lần triển khai cuối cùng.

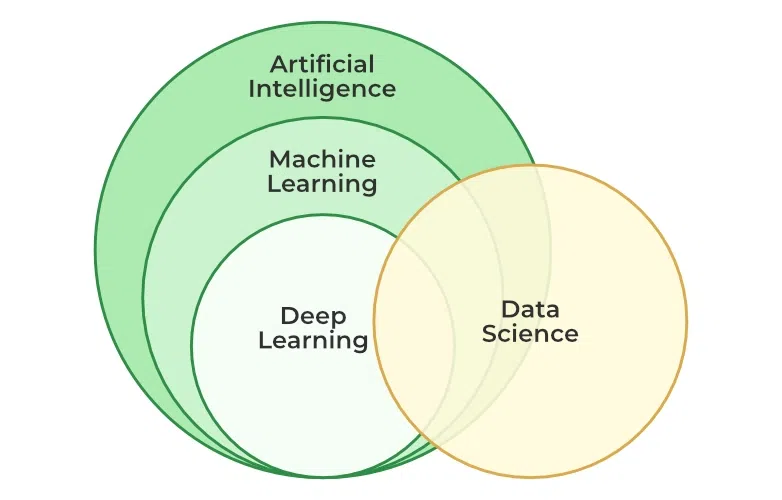


Hình 2.2 Thư viện Python

* 1. **Tổng quan về Deep Learning**

Deep Learning là một nhánh của Machine Learning dựa trên kiến ​​trúc mạng lưới thần kinh nhân tạo. Mạng nơ-ron nhân tạo hay ANN sử dụng các lớp nút được kết nối với nhau gọi là nơ-ron hoạt động cùng nhau để xử lý và học hỏi từ dữ liệu đầu vào [3].

Trong mạng nơ-ron sâu được kết nối đầy đủ, có một lớp đầu vào và một hoặc nhiều lớp ẩn được kết nối lần lượt với nhau. Mỗi nơ-ron nhận đầu vào từ các nơ-ron lớp trước hoặc lớp đầu vào. Đầu ra của một nơ-ron trở thành đầu vào của các nơ-ron khác trong lớp tiếp theo của mạng và quá trình này tiếp tục cho đến khi lớp cuối cùng tạo ra đầu ra của mạng. Các lớp của mạng nơ-ron biến đổi dữ liệu đầu vào thông qua một loạt các phép biến đổi phi tuyến, cho phép mạng tìm hiểu các cách biểu diễn phức tạp của dữ liệu đầu vào [3].



Hình 2.3 Deep Learning

Ngày nay Deep Learning đã trở thành một trong những lĩnh vực phổ biến và dễ thấy nhất của học máy nhờ thành công của nó trong nhiều ứng dụng, chẳng hạn như thị giác máy tính, xử lý ngôn ngữ tự nhiên và Học tăng cường [3].

Deep Learning có thể được sử dụng cho học máy có giám sát, không giám sát cũng như học máy tăng cường. nó sử dụng nhiều cách khác nhau để xử lý những thứ này [3].

**Supervised Machine Learning (Học máy được giám sát):** Supervised Machine Learning là kỹ thuật học máy trong đó mạng lưới thần kinh học cách đưa ra dự đoán hoặc phân loại dữ liệu dựa trên các bộ dữ liệu được gắn nhãn. Ở đây chúng ta nhập cả hai tính năng đầu vào cùng với các biến mục tiêu. mạng lưới thần kinh học cách đưa ra dự đoán dựa trên chi phí hoặc lỗi xuất phát từ sự khác biệt giữa mục tiêu được dự đoán và mục tiêu thực tế, quá trình này được gọi là lan truyền ngược. Các thuật toán học sâu như Convolutional neural networks, Recurrent neural networks được sử dụng cho nhiều tác vụ được giám sát như phân loại và nhận dạng hình ảnh, phân tích tình cảm, dịch ngôn ngữ, v.v [3].

**Unsupervised Machine Learning** (**Học máy không giám sát)**: Unsupervised Machine Learning là kỹ thuật học máy trong đó mạng lưới thần kinh học cách khám phá các mẫu hoặc phân cụm tập dữ liệu dựa trên các tập dữ liệu không được gắn nhãn. Ở đây không có biến mục tiêu. trong khi máy phải tự xác định các mẫu hoặc mối quan hệ ẩn trong bộ dữ liệu. Các thuật toán học sâu như bộ mã hóa tự động và mô hình tổng quát được sử dụng cho các tác vụ không được giám sát như phân cụm, giảm kích thước và phát hiện bất thường [3].

**Reinforcement Machine Learning (**Học máy tăng cường): Reinforcement Machine Learning là kỹ thuật học máy trong đó tác nhân học cách đưa ra quyết định trong môi trường để tối đa hóa tín hiệu khen thưởng. Tác nhân tương tác với môi trường bằng cách thực hiện hành động và quan sát phần thưởng thu được. Học sâu có thể được sử dụng để tìm hiểu các chính sách hoặc một tập hợp hành động nhằm tối đa hóa phần thưởng tích lũy theo thời gian. Các thuật toán học máy tăng cường như Deep Q networks và Deep Deterministic Policy Gradient (DDPG) được sử dụng để củng cố các nhiệm vụ như robot và chơi trò chơi, v.v [3].

* 1. **Tìm hiểu Machine Learning Pipeline**
     1. **Khái Niệm**

Machine Learning Pipeline là một cách để mã hóa và tự động hóa quy trình làm việc cần thiết để tạo ra mô hình học máy. Quy trình máy học bao gồm nhiều bước tuần tự thực hiện mọi việc từ trích xuất và xử lý trước dữ liệu cho đến đào tạo và triển khai mô hình [4].



Hình 2.4 Machine Learning Pipeline

* + 1. **Lợi ích của Machine Learning Pipeline**

**Giảm thời gian tạo ra giá trị**

Đối với các tổ chức bắt đầu hành trình học máy, hiệu quả là điều tối quan trọng. Việc thiết lập các mô hình học máy theo cách thủ công tốn nhiều thời gian, với các quy trình như chuẩn bị dữ liệu, kỹ thuật tính năng, đào tạo mô hình và đánh giá kéo dài nhiều ngày hoặc thậm chí nhiều tuần [5].

Việc tận dụng quy trình học máy có thể hợp lý hóa đáng kể các nhiệm vụ này. Điều này không chỉ tăng tốc độ triển khai mô hình mà còn giảm đáng kể thời gian tạo ra giá trị. Với các công cụ như Akkio, có thể vận hành mô hình của mình chỉ trong vài phút hoặc vài giờ [5].

**Sự nhất quán giữa các thành viên trong nhóm**

Một lý do khác để sử dụng quy trình học máy là nó giúp đảm bảo tính nhất quán giữa các thành viên trong nhóm. Vì quy trình này được tự động hóa nên mọi người sẽ làm việc với cùng một dữ liệu theo cùng một cách bất kể trình độ kỹ năng hoặc kiến ​​thức của họ về quy trình tổng thể [5].

Điều này quan trọng vì hai lý do: Thứ nhất, nó đảm bảo rằng mọi người đều đang làm việc với dữ liệu cập nhật nhất. Thứ hai, nó giảm thiểu khả năng xảy ra lỗi của con người [5].

**Tận dụng sức mạnh của dữ liệu lớn**

Với quy trình máy học, có thể dễ dàng khai thác các tập dữ liệu lớn để đào tạo và đánh giá mô hình của mình [5].

Điều này rất quan trọng vì các mô hình học máy chỉ hoạt động tốt khi có dữ liệu mà chúng được đào tạo. Nói chung, càng có nhiều dữ liệu chất lượng thì kết quả mô hình sẽ càng tốt [5].

**Tận dụng sức mạnh của AI**

Sử dụng quy trình học máy là điều cho phép tận dụng sức mạnh của trí tuệ nhân tạo mà không cần phải thuê các chuyên gia tận tâm hoặc mất hàng tháng trời cho nghiên cứu và phát triển [5].

Với machine learning pipeline, có thể dễ dàng tận dụng các công cụ và kĩ thuật ML hiện có [5].

**Giảm khả năng xảy ra lỗi của con người**

Giảm khả năng xảy ra lỗi của con người là một trong những lợi thế của quy trình học máy vì quy trình này được tự động hóa và có các bước xác định mà mọi người cần phải tuân theo. Lỗi của con người có thể tốn kém và tốn thời gian để sửa chữa [5].

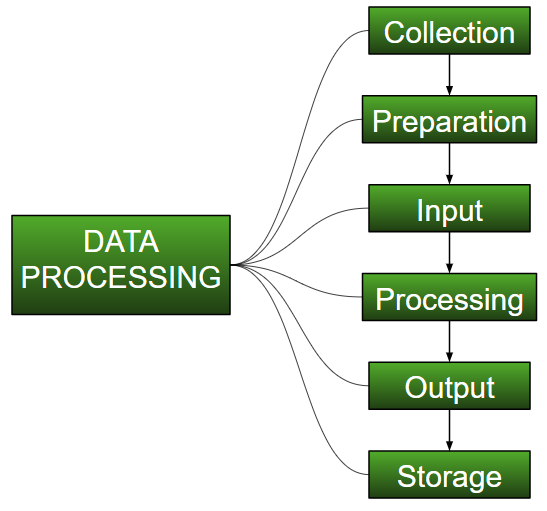
* 1. **Tìm hiểu về Data Preprocessing**

Data Processing là nhiệm vụ chuyển đổi dữ liệu từ một biểu mẫu nhất định sang dạng mong muốn và hữu dụng hơn nhiều, tức là dạng dữ liệu mà học máy có thể hiểu được. Đầu ra của quá trình hoàn chỉnh này có thể ở bất kỳ dạng nào mong muốn như biểu đồ, video, biểu đồ, bảng, hình ảnh, v.v., tùy thuộc vào nhiệm vụ chúng tôi đang thực hiện và yêu cầu của máy [6].

Xử lý dữ liệu là một bước quan trọng trong machine learning (ML) pipeline, vì nó chuẩn bị dữ liệu để sử dụng trong việc xây dựng và đào tạo các mô hình ML. Mục tiêu của việc xử lý dữ liệu là làm sạch, chuyển đổi và chuẩn bị dữ liệu theo định dạng phù hợp cho mô hình hóa [6].

**Các bước chính liên quan đến data processing thường bao gồm:**

1. **Data collection (Thu thập dữ liệu):** Đây là quá trình thu thập dữ liệu từ nhiều nguồn khác nhau, chẳng hạn như cảm biến, cơ sở dữ liệu hoặc các hệ thống khác. Dữ liệu có thể có cấu trúc hoặc không có cấu trúc và có thể ở nhiều định dạng khác nhau như văn bản, hình ảnh hoặc âm thanh [6].
2. **Data preprocessing (Tiền xử lý dữ liệu):** Bước này bao gồm việc làm sạch, lọc và chuyển đổi dữ liệu để phù hợp cho việc phân tích thêm. Điều này có thể bao gồm việc loại bỏ các giá trị bị thiếu, chia tỷ lệ hoặc chuẩn hóa dữ liệu hoặc chuyển đổi dữ liệu sang định dạng khác [6].
3. **Data analysis (Phân tích dữ liệu):** Trong bước này, dữ liệu được phân tích bằng nhiều kỹ thuật khác nhau như phân tích thống kê, thuật toán học máy hoặc trực quan hóa dữ liệu. Mục tiêu của bước này là rút ra những hiểu biết sâu sắc hoặc kiến ​​thức từ dữ liệu [6].
4. **Data interpretation (Giải thích dữ liệu):** Bước này bao gồm việc diễn giải kết quả phân tích dữ liệu và đưa ra kết luận dựa trên những hiểu biết sâu sắc thu được. Nó cũng có thể liên quan đến việc trình bày các phát hiện một cách rõ ràng và ngắn gọn, chẳng hạn như thông qua các báo cáo, bảng điều khiển hoặc các hình ảnh trực quan khác [6].
5. **Data storage and management (Lưu trữ và quản lý dữ liệu):** Sau khi dữ liệu đã được xử lý và phân tích, dữ liệu đó phải được lưu trữ và quản lý theo cách an toàn và dễ dàng truy cập. Điều này có thể liên quan đến việc lưu trữ dữ liệu trong cơ sở dữ liệu, lưu trữ đám mây hoặc các hệ thống khác và thực hiện các chiến lược sao lưu và phục hồi để bảo vệ khỏi mất dữ liệu [6].



Hình 2.5 Data Processing

* 1. **Tìm hiểu về Data Validation**

Data Validation là một quá trình quan trọng trong Machine Learning và Phân tích dữ liệu. Nó liên quan đến việc kiểm tra tính nhất quán, đầy đủ và chính xác của dữ liệu trước khi sử dụng để đào tạo và thử nghiệm các mô hình Machine Learning. Một số bước liên quan đến xác thực dữ liệu bao gồm xử lý các giá trị bị thiếu, loại bỏ hoặc tổng hợp các mục trùng lặp, đảm bảo dữ liệu được định dạng chính xác và tuân theo các ràng buộc miền và quy tắc đã chỉ định. Quá trình này đảm bảo chất lượng dữ liệu, điều này tác động đáng kể đến hiệu suất của các mô hình Machine Learning [7].

Các loại xác thực hình ảnh với Images

1. **Xác định tiêu chí chất lượng dữ liệu**

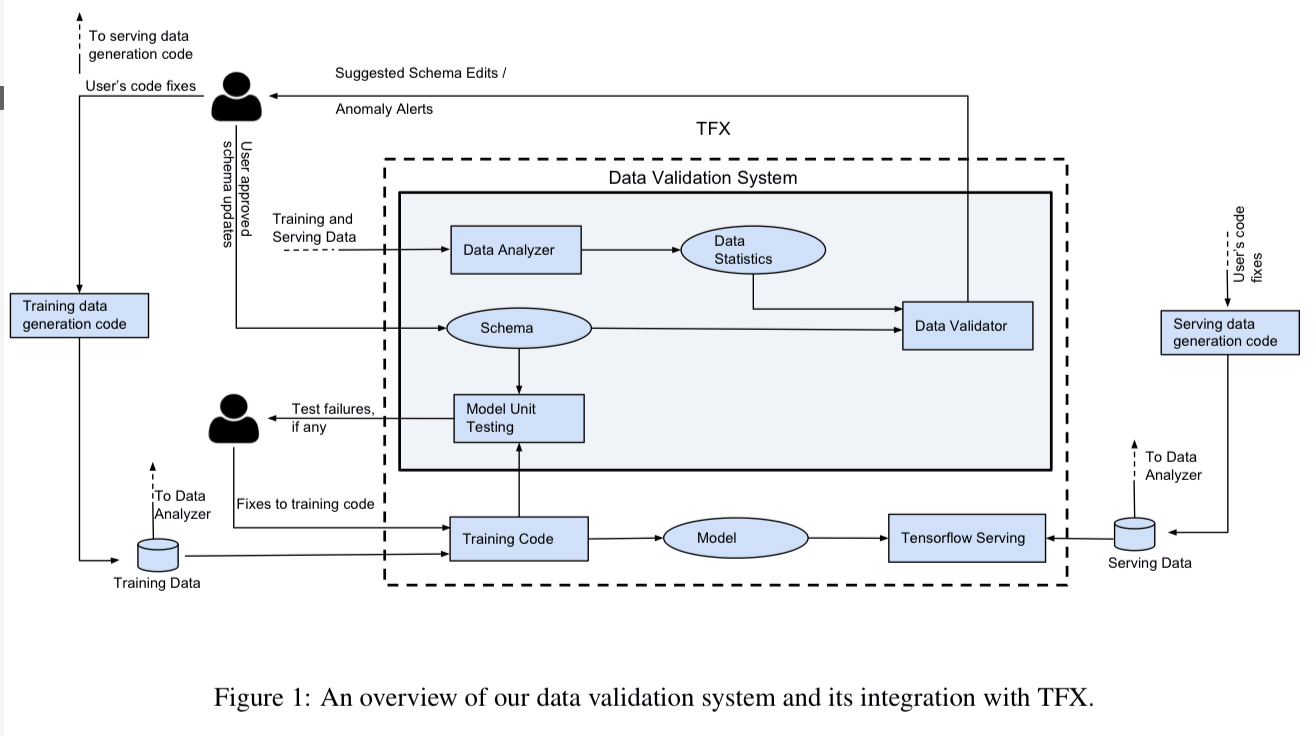
Xác thực dữ liệu trong khai thác hình ảnh là xác định tiêu chí chất lượng dữ liệu phù hợp với mục tiêu. Tiêu chí chất lượng dữ liệu là các tiêu chuẩn hoặc quy tắc đo lường tính hợp lệ, tính đầy đủ, tính nhất quán và mức độ liên quan của dữ liệu. Có thể kiểm tra xem hình ảnh có đúng định dạng, độ phân giải, kích thước, màu sắc và hướng hay không hoặc liệu chúng có chứa bất kỳ nhiễu, biến dạng hoặc tạo tác nào không [8].

1. **Thực hiện làm sạch và tiền xử lý dữ liệu**

Thực hiện làm sạch và tiền xử lý trước dữ liệu để loại bỏ hoặc sửa bất kỳ lỗi, ngoại lệ hoặc sự không nhất quán nào trong dữ liệu. Làm sạch và tiền xử lý dữ liệu là các quá trình chuyển đổi và nâng cao dữ liệu để làm cho dữ liệu phù hợp và đáng tin cậy hơn cho việc khai thác hình ảnh. Ví dụ như cắt xén, thay đổi kích thước, xoay, lọc, làm mịn, làm sắc nét hoặc nâng cao hình ảnh để cải thiện chất lượng và độ rõ nét của chúng [8].

1. **Áp dụng tăng cường và chuyển đổi dữ liệu**

Tăng cường và chuyển đổi dữ liệu là các quá trình tạo hình ảnh mới hoặc sửa đổi từ hình ảnh hiện có bằng cách áp dụng một số thao tác như lật, xoay, chia tỷ lệ, dịch chuyển, cắt, cắt hoặc thêm nhiễu. Các hoạt động này có thể giúp tạo ra hình ảnh thực tế và đa dạng hơn, có thể cải thiện hiệu suất và tính tổng quát của các phương pháp khai thác hình ảnh [8].



Hình 2.6 Data Validation

* 1. **Tìm hiểu về Feature Engineering**
     1. **Giới thiệu về Feature Engineering**

Feature Engineering là quá trình chuyển đổi dữ liệu thô thành các tính năng phù hợp với các mô hình học máy. Nói cách khác, đó là quá trình lựa chọn, trích xuất và chuyển đổi các tính năng phù hợp nhất từ dữ liệu có sẵn để xây dựng mô hình học máy chính xác và hiệu quả [9].

* + 1. **Sự cần thiết của Feature Engineering trong học máy**

Feature Engineering được thiết kế để:

* **Nâng cao trải nghiệm người dùng của sản phẩm:** Bằng cách thêm các tính năng mới có thể làm cho sản phẩm trực quan hơn, hiệu quả hơn và thân thiện với người dùng hơn [9].
* **Tăng lợi thế cạnh tranh trên thị trường:** Bằng cách cung cấp các tính năng độc đáo và sáng tạo để có thể phân biệt sản phẩm của mình với các đối thủ cạnh tranh và thu hút nhiều khách hàng hơn [9].
* **Đáp ứng nhu cầu phát triển của khách hàng:** bằng cách phân tích phản hồi của người dùng, xu hướng thị trường và hành vi của khách hàng [9].
  + 1. **Các quy trình liên quan đến Feature Engineering**

**Feature Creation**

Là quá trình tạo ra các tính năng mới dựa trên:

* Kiến thức về miền [9].
* Quan sát các mẫu trong dữ liệu [9].
* Kết hợp các tính năng hiện có hoặc tổng hợp các điểm dữ liệu mới [9].

**Feature Transformation**

Là quá trình chuyển đổi các tính năng thành một đại diện phù hợp cho mô hình học máy. Quá trình chuyển đổi này bao gồm:

* Chuẩn hóa bằng cách thay đổi tỷ lệ các tính năng để có phạm vị tương tự [9].
* Chia tỷ lệ được sử dụng để chuyển đổi các biến số để có tỷ lệ tương tự để chúng có thể được so sánh dễ dàng hơn [9].
* Mã hóa:Chuyển đổi các tính năng phân loại thành biểu diễn số [9].
* Chuyển đổi**:** Chuyểnđổi các tính năng bằng cách sử dụng các phép toán để thay đổi sự phân bố hoặc quy mô của các tính năng [9].

**Feature Extraction**

Là quá trình tạo các tính năng mới từ các tính năng hiện có để cung cấp thông tin phù hợp hơn cho mô hình học máy [9].

**Feature Selection**

Là quá trình chọn một tập hợp con các tính năng có liên quan từ tập dữ liệu sẽ được sử dụng trong mô hình học máy [9].

**Feature Scaling**

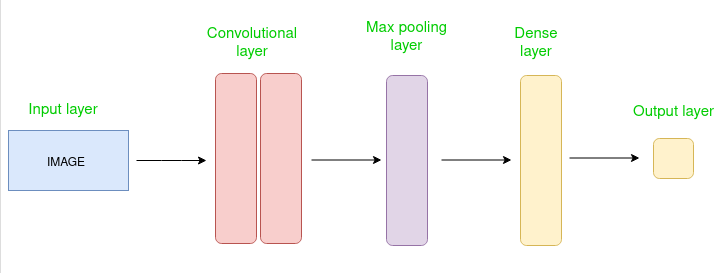
Là quá trình chuyển đổi các tính năng để chúng có quy mô tương tự [9].

* 1. **Tìm hiểu về Mô hình CNN**

Convolutional Neural Network (CNN) là một loại mạng thần kinh học sâu rất phù hợp để phân tích hình ảnh và video. CNN sử dụng một loạt các lớp tích chập và gộp để trích xuất các đặc điểm từ hình ảnh và video, sau đó sử dụng các đặc điểm này để phân loại hoặc phát hiện các đối tượng hoặc cảnh [10].

**Kiến trúc của CNN**

Mạng thần kinh chuyển đổi bao gồm nhiều lớp như lớp đầu vào, lớp tích chập, lớp gộp và các lớp được kết nối đầy đủ [10].



Hình 2.7 Kiến trúc CNN đơn giản

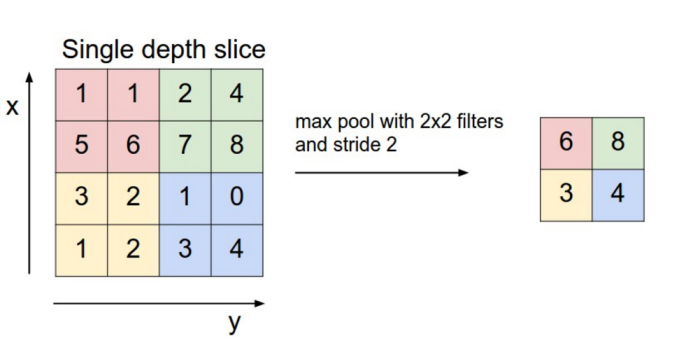
Lớp Convolutional áp dụng các bộ lọc cho hình ảnh đầu vào để trích xuất các đặc điểm, lớp Pooling lấy mẫu hình ảnh xuống để giảm tính toán và lớp được kết nối đầy đủ sẽ đưa ra dự đoán cuối cùng. Mạng tìm hiểu các bộ lọc tối ưu thông qua lan truyền ngược và giảm độ dốc [10].

**Cách thức hoạt động của Lớp tích chập**

* Các lớp tích chập bao gồm một tập hợp các bộ lọc (hoặc hạt nhân) có thể học được có chiều rộng và chiều cao nhỏ và có cùng độ sâu với khối lượng đầu vào (3 nếu lớp đầu vào là đầu vào hình ảnh) [10].
* Trong quá trình chuyển tiếp, chúng ta trượt từng bộ lọc trên toàn bộ âm lượng đầu vào trong đó mỗi bước được gọi là bước tiến (có thể có giá trị 2, 3 hoặc thậm chí 4 đối với hình ảnh có chiều cao) và tính tích số chấm giữa trọng lượng hạt nhân và bản vá từ khối lượng đầu vào [10].
* Khi trượt các bộ lọc, chúng ta sẽ nhận được đầu ra 2-D cho mỗi bộ lọc và kết quả là chúng tôi sẽ xếp chúng lại với nhau, chúng ta sẽ nhận được âm lượng đầu ra có độ sâu bằng số lượng bộ lọc. Mạng sẽ tìm hiểu tất cả các bộ lọc [10].

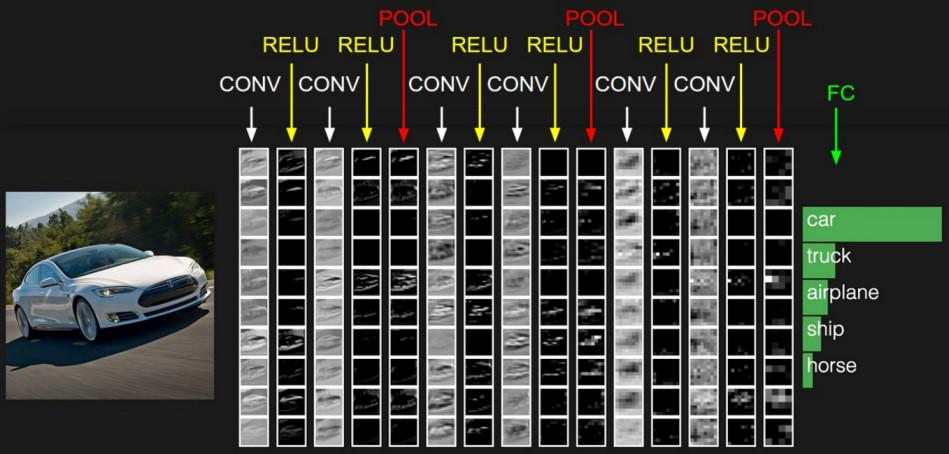
**Các lớp được sử dụng để xây dựng CNN**

* **Input Layers (Lớp đầu vào)**: Đây là lớp mà chúng ta cung cấp đầu vào cho mô hình của mình. Trong CNN, Thông thường, đầu vào sẽ là một hình ảnh hoặc một chuỗi hình ảnh. Lớp này chứa đầu vào thô của hình ảnh với chiều rộng 32, chiều cao 32 và chiều sâu 3 [10].
* **Convolutional Layers (Lớp tích chập**): Đây là lớp được sử dụng để trích xuất tính năng từ tập dữ liệu đầu vào. Nó áp dụng một tập hợp các bộ lọc có thể học được gọi là hạt nhân cho hình ảnh đầu vào. Các bộ lọc/hạt nhân là các ma trận nhỏ hơn thường có hình dạng 2×2, 3×3 hoặc 5×5. nó trượt qua dữ liệu hình ảnh đầu vào và tính toán tích số chấm giữa trọng lượng hạt nhân và bản vá hình ảnh đầu vào tương ứng. Đầu ra của lớp này là bản đồ tính năng quảng cáo được giới thiệu. Giả sử chúng ta sử dụng tổng cộng 12 bộ lọc cho lớp này, chúng ta sẽ nhận được khối lượng đầu ra có kích thước 32 x 32 x 12 [10].
* [**Activation Layer**](https://www.geeksforgeeks.org/activation-functions-neural-networks/) ([**Lớp kích hoạt**):](https://www.geeksforgeeks.org/activation-functions-neural-networks/) Bằng cách thêm chức năng kích hoạt vào đầu ra của lớp trước, các lớp kích hoạt sẽ thêm tính phi tuyến vào mạng. nó sẽ áp dụng hàm kích hoạt theo phần tử cho đầu ra của lớp chập. Một số hàm kích hoạt phổ biến là RELU : max(0, x), Tanh , Leaky RELU , v.v. Âm lượng không thay đổi nên âm lượng đầu ra sẽ có kích thước 32 x 32 x 12 [10].
* [**Pooling layer (Lớp gộp**):](https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/)Lớp này được chèn định kỳ vào các covnets và chức năng chính của nó là giảm kích thước ổ đĩa, khiến quá trình tính toán nhanh chóng làm giảm bộ nhớ và cũng ngăn ngừa việc trang bị quá mức. Hai loại lớp gộp chung phổ biến là gộp chung tối đa và gộp chung trung bình. Nếu chúng tôi sử dụng nhóm tối đa với bộ lọc 2 x 2 và bước 2, âm lượng kết quả sẽ có kích thước 16x16x12 [10].



Hình 2.8 Pooling layer

* **Flattening**: Các bản đồ đặc trưng thu được được làm phẳng thành một vectơ một chiều sau các lớp tích chập và gộp để chúng có thể được chuyển vào một lớp được liên kết hoàn toàn để phân loại hoặc hồi quy [10].
* **Fully Connected Layers**: Nó lấy đầu vào từ lớp trước và tính toán nhiệm vụ phân loại hoặc hồi quy cuối cùng [10].



Hình 2.9 Lớp Fully Connected Layers

* **Output Layer**: Sau đó, đầu ra từ các lớp được kết nối đầy đủ sẽ được đưa vào hàm logistic cho các tác vụ phân loại như sigmoid hoặc softmax để chuyển đổi đầu ra của mỗi lớp thành điểm xác suất của mỗi lớp [10].
  1. **Tìm hiểu về Model Evaluation**

Đánh giá mô hình là quá trình sử dụng các số liệu đánh giá khác nhau để hiểu hiệu suất của mô hình học máy cũng như điểm mạnh và điểm yếu của nó. Đánh giá mô hình rất quan trọng để đánh giá hiệu quả của mô hình trong các giai đoạn nghiên cứu ban đầu và nó cũng đóng vai trò trong việc giám sát mô hình.

Các số liệu phổ biến nhất để đo hiệu suất phân loại bao gồm accuracy, precision, confusion matrix, log-loss, and AUC(diện tích dưới đường cong ROC).

* **Accuracy** đo lường tần suất bộ phân loại đưa ra các dự đoán chính xác, vì đó là tỷ lệ giữa số lượng dự đoán đúng và tổng số lượng dự đoán.
* **Precision** đo lường tỷ lệ Positives được dự đoán thực sự Positive. **Precision** là một lựa chọn tốt về số liệu đánh giá khi bạn muốn chắc chắn về dự đoán của mình.

1. **TRIỂN KHAI**
   1. **Thu Thập Dữ Liệu**

Tập dữ liệu mà chúng tôi sẽ sử dụng để huấn luyện bộ phân loại biển báo giao thông tùy chỉnh của riêng mình là [German Traffic Sign Recognition Benchmark (GTSRB)](https://www.kaggle.com/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign).

Bộ dữ liệu GTSRB bao gồm 43 lớp biển báo giao thông và gần 50.000 hình ảnh.

Bộ dữ liệu với những hình ảnh trong những điều kiện khác nhau, điều này giúp cho mô hình có khả năng nhận diện tốt hơn trong nhiều hoàn cảnh khác nhau.

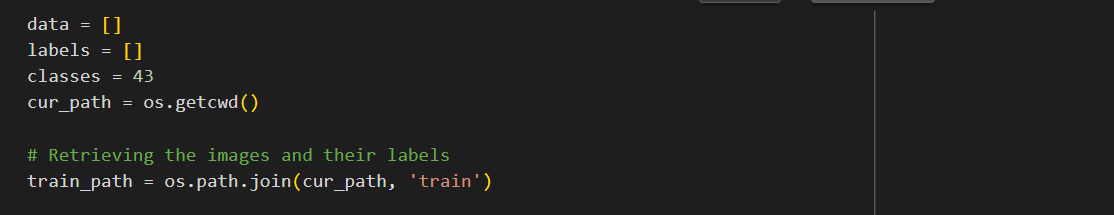
****

Hình 3.1 Bộ dữ liệu GTSRB

**Triển khai code:**

**train\_path** là đường dẫn đến thư mục chứa dữ liệu huấn luyện.

Sử dụng vòng lặp để duyệt qua từng loại biển báo (43 loại) để thu thập dữ liệu.

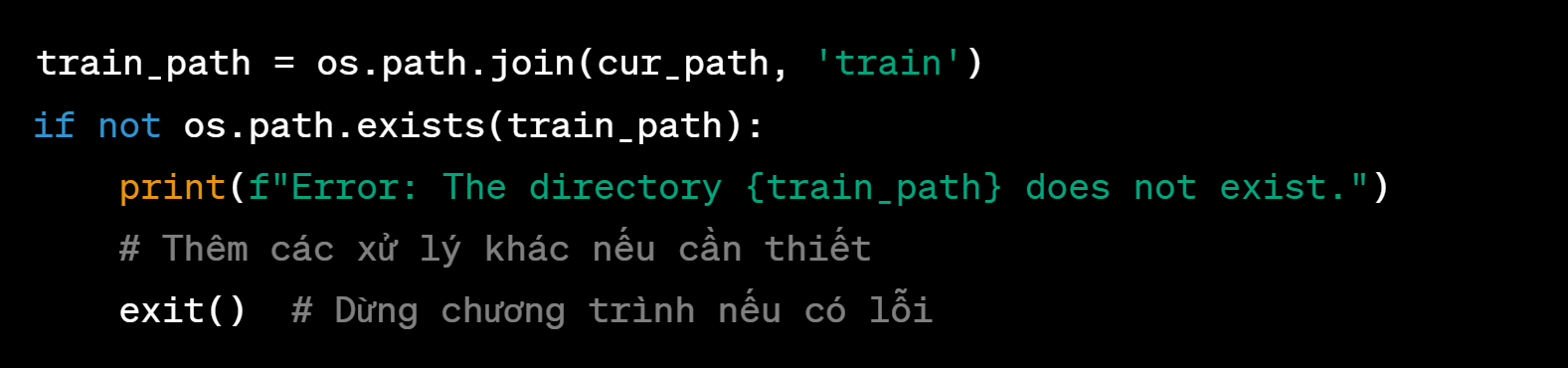


* 1. **Xử Lý Dữ Liệu**

Trong giai đoạn này, nhóm em thực hiện một loạt các kiểm tra để đảm bảo tính toàn vẹn và sẵn có của dữ liệu. Dưới đây là mô tả chi tiết về quá trình kiểm tra:

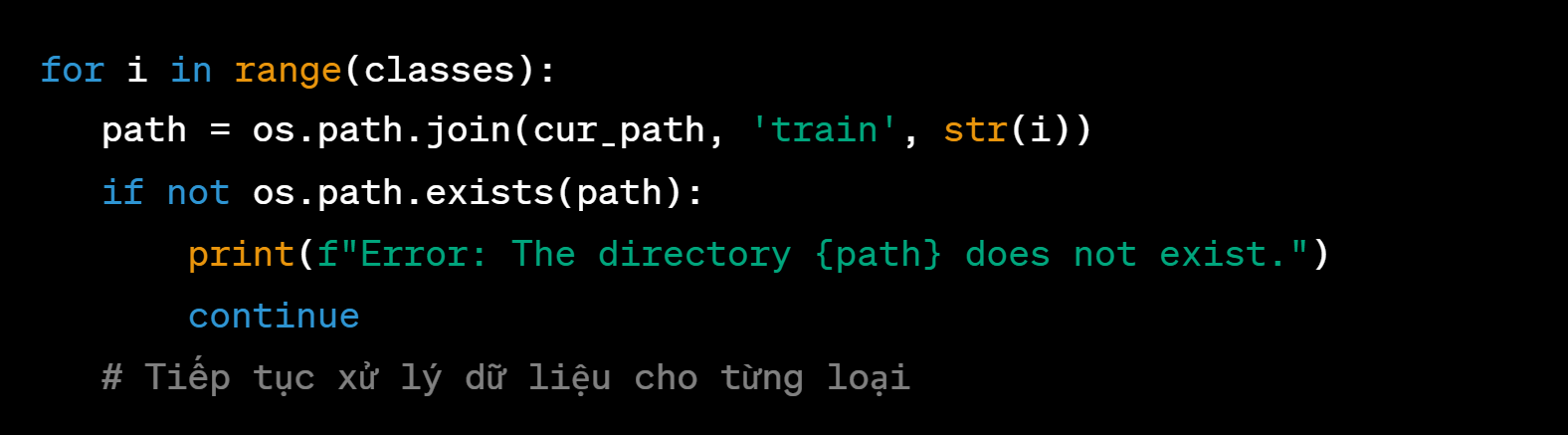
**Kiểm Tra Tổn Tại Thư Mục Chứa Dữ Liệu:**

* Sử dụng hàm **os.path.exists()** để kiểm tra sự tồn tại của thư mục chứa dữ liệu.
* Nếu thư mục không tồn tại, in thông báo lỗi và dừng chương trình để tránh xử lý dữ liệu không tồn tại.



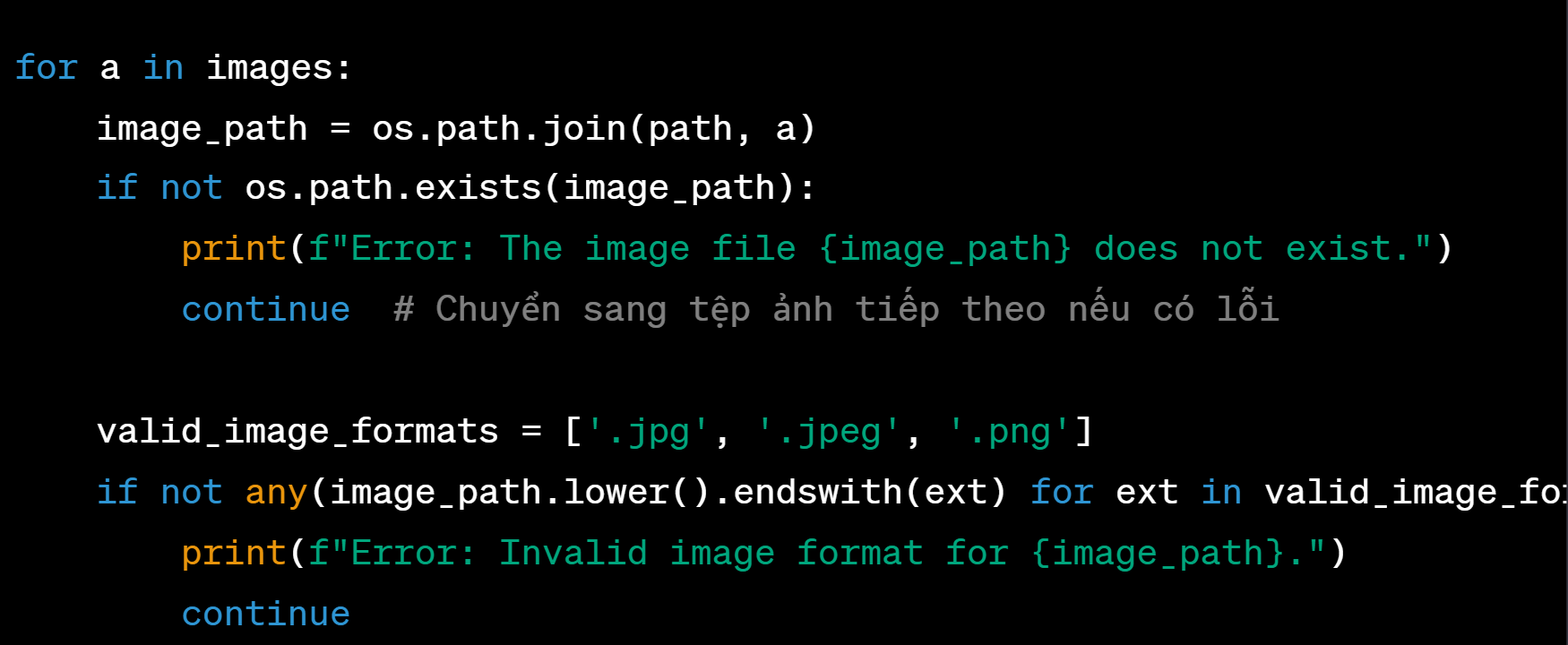
**Kiểm Tra Tổn Tại Các Thư Mục Con Cho Mỗi Loại:**

* Sử dụng một vòng lặp để kiểm tra sự tồn tại của các thư mục con cho từng loại biển báo.
* In thông báo lỗi nếu bất kỳ thư mục con nào không tồn tại và tiếp tục với các loại khác.



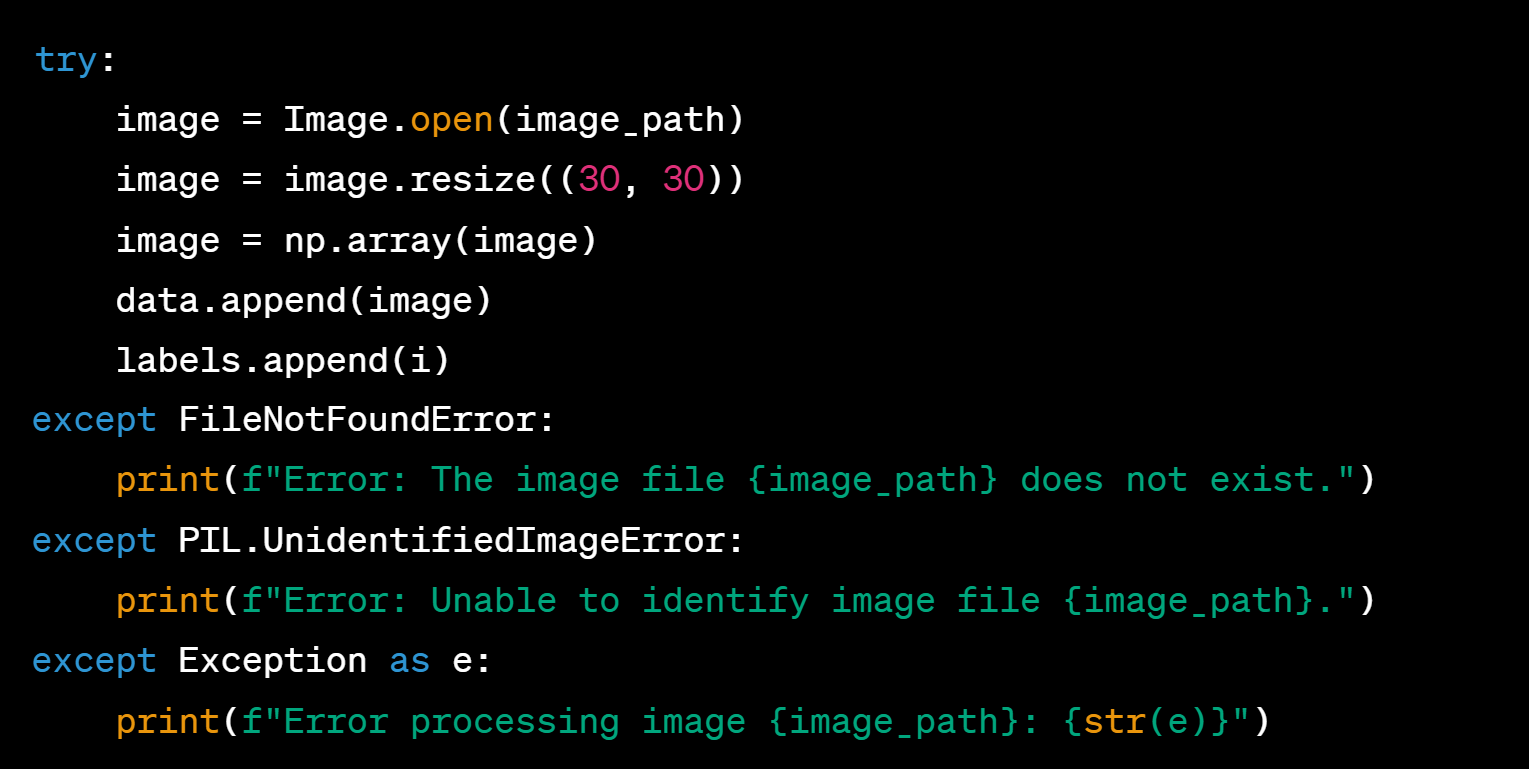
**Kiểm Tra Tổn Tại và Định Dạng Của Ảnh:**

* Sử dụng một vòng lặp để kiểm tra tồn tại và định dạng của từng tệp ảnh trong các thư mục con.
* Kiểm tra định dạng ảnh và in thông báo lỗi nếu định dạng không hợp lệ.

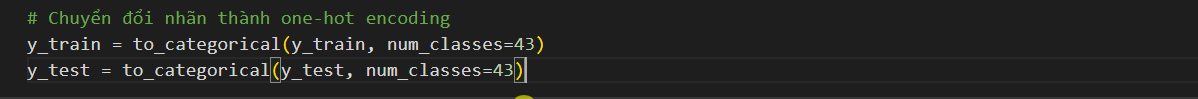


**Xử Lý Ngoại Lệ:**

* Bao gồm các xử lý ngoại lệ để xử lý trường hợp mà không thể mở hoặc nhận diện ảnh.
* In thông báo lỗi cụ thể nếu có vấn đề và tiếp tục với các ảnh khác.



**Chuyển đổi nhãn thành one-hot encoding**



**Kết Luận:** Quá trình kiểm tra dữ liệu này giúp đảm bảo tính toàn vẹn của tập dữ liệu và xử lý một cách an toàn các vấn đề có thể xuất hiện trong quá trình tiền xử lý.

* 1. **Xây dựng model CNN**

Model CNN được xây dựng bởi các lớp: Lớp Convolutional 1 và 2, Lớp MaxPooling2D, Lớp Dropout, Lớp Flatten, Lớp Fully Connected 1 và 2.

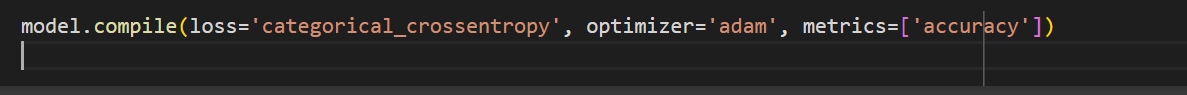
* **Lớp Convolutional 1 và 2:** Thực hiện việc học các đặc trưng từ ảnh thông qua các bộ lọc.
* **Lớp MaxPooling2D:** Giảm kích thước của ma trận đặc trưng bằng cách giữ lại giá trị lớn nhất từ mỗi vùng.
* **Lớp Dropout:** Ngăn chặn overfitting bằng cách tắt một số neuron ngẫu nhiên trong quá trình huấn luyện.
* **Lớp Flatten:** Chuyển từ ma trận 2D sang vector 1D để làm đầu vào cho các lớp Fully Connected.
* **Lớp Fully Connected 1 và 2:** Các lớp này giúp mô hình học được mối quan hệ phức tạp giữa đặc trưng và nhãn.

****

* 1. **Huấn luyện model CNN**

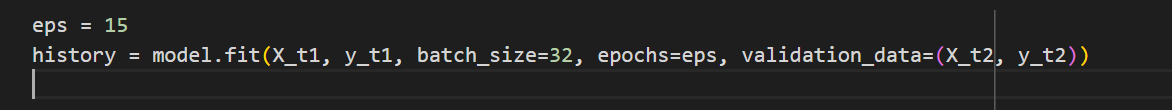
**Biên Dịch Mô Hình:**

* Sử dụng phương thức **compile** của mô hình để thiết lập các tham số như hàm mất mát, trình tối ưu hóa và các độ đo hiệu suất.
* Trong trường hợp này, chúng tôi sử dụng hàm mất mát 'categorical\_crossentropy' (phù hợp với bài toán phân loại nhiều lớp), tối ưu hóa 'adam', và độ đo hiệu suất là 'accuracy'.



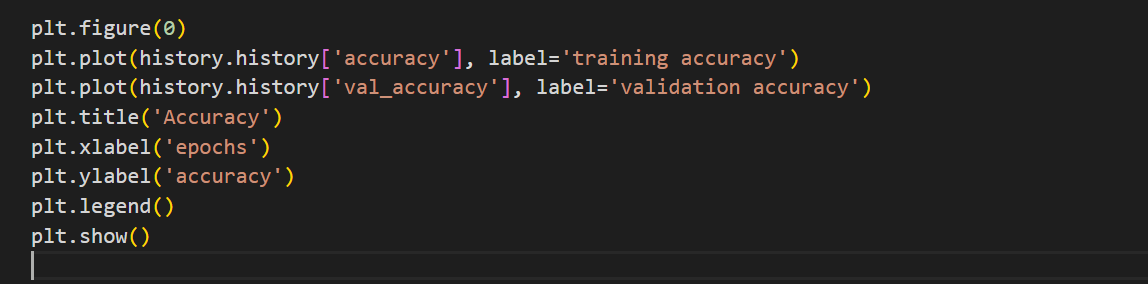
**Huấn Luyện Mô Hình:**

* Sử dụng phương thức fit để bắt đầu quá trình huấn luyện mô hình trên tập dữ liệu huấn luyện.
* Số epoch và batch size được xác định trước để quyết định số lần mô hình sẽ được huấn luyện trên toàn bộ tập dữ liệu.
* Hiệu suất của mô hình trên cả tập huấn luyện và tập kiểm thử được theo dõi để đánh giá và so sánh.



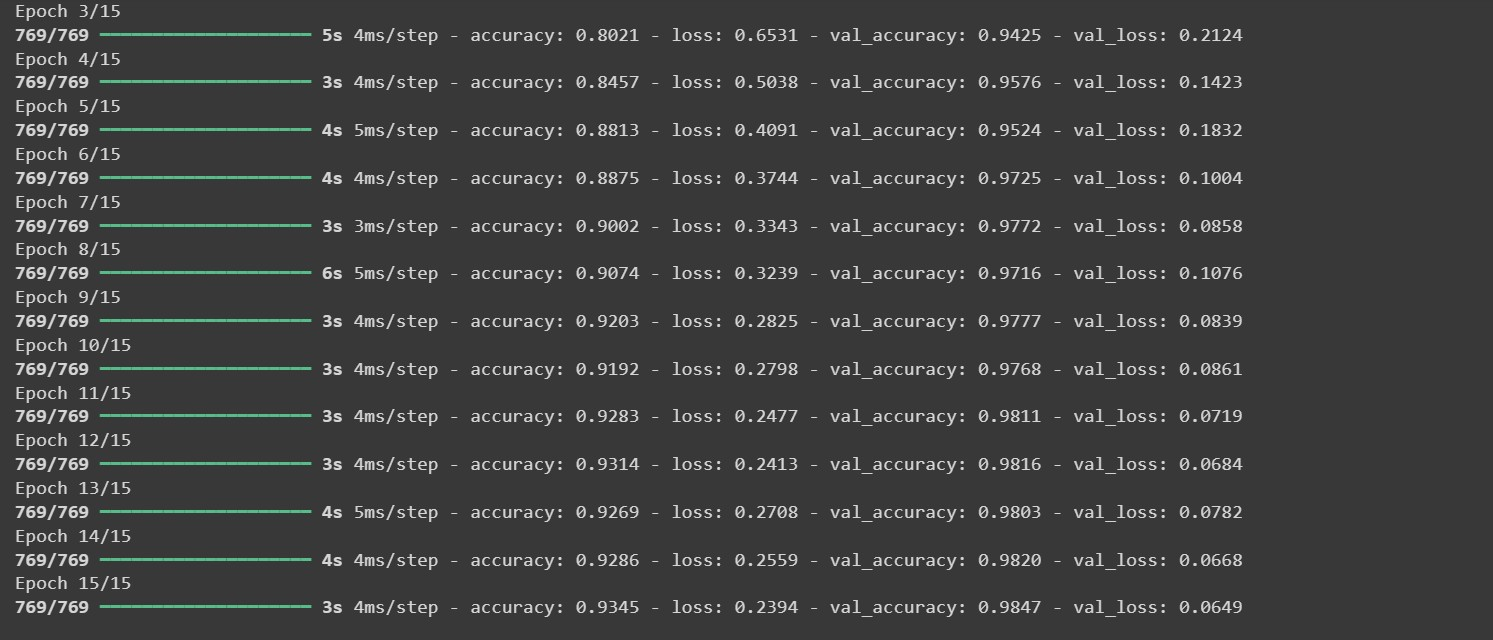
**Đánh Giá Hiệu Suất:**

* Kết quả của mô hình trên tập kiểm thử được truy xuất từ biến history.
* Sử dụng các biểu đồ để minh họa hiệu suất của mô hình qua các epoch trên cả tập huấn luyện và tập kiểm thử.



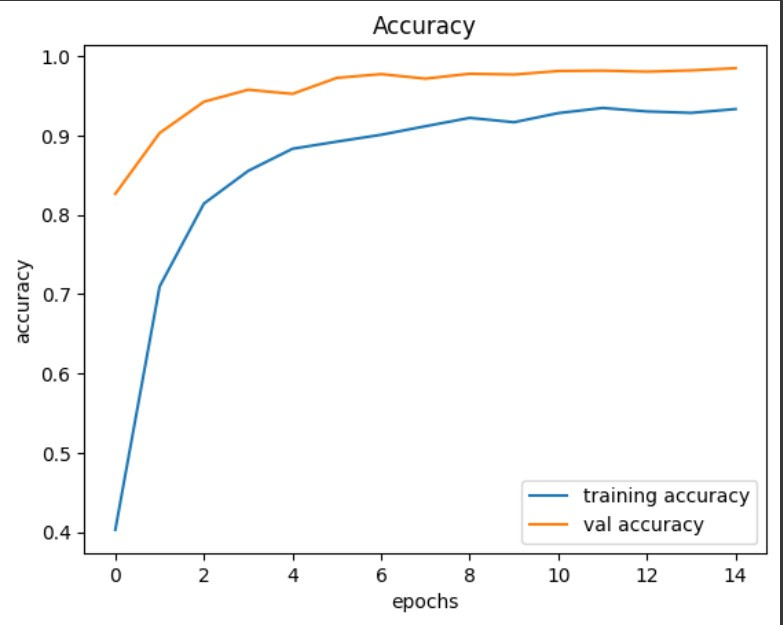
**Kết luận:** Quá trình này giúp đánh giá và theo dõi hiệu suất của mô hình trong quá trình huấn luyện, từ đó đưa ra quyết định về việc điều chỉnh hoặc cải thiện mô hình nếu cần thiết.

**Kết quả:**

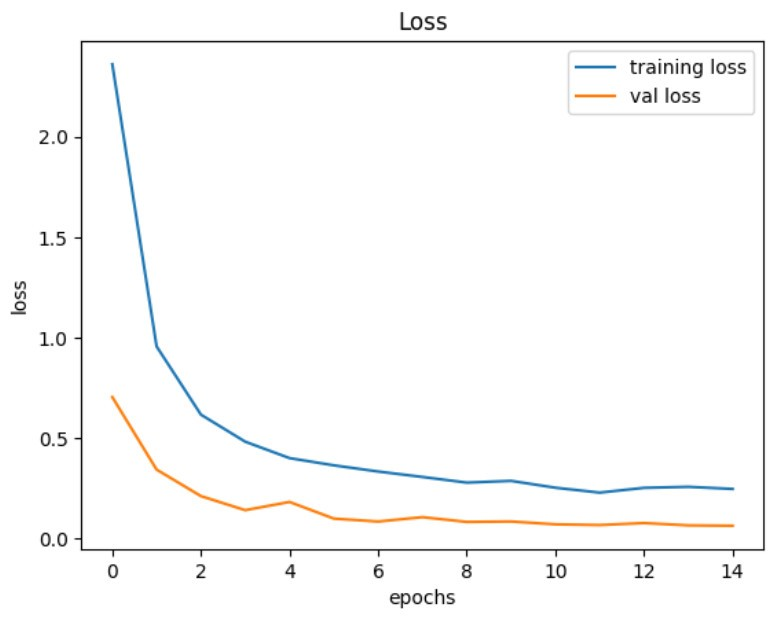


Hình 3.2 Kết quả huấn luyện model

Kết quả sau khi huấn luyện model với batch size là 32, model đã đạt được độ chính xác lên đến 95% trên các tập huấn luyện và đạt được độ ổn định qua 15 vòng lặp huấn luyện.



Hình 3.3 Đồ thị hiển thị giá trị chính xác của model

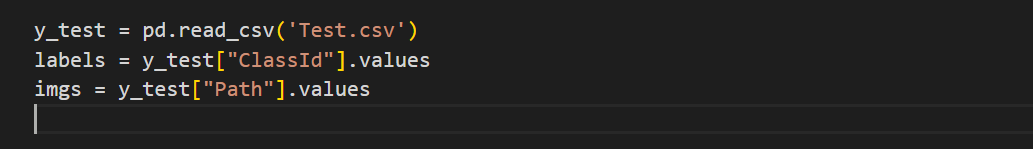
**

Hình 3.4 Đồ thị hiển thị giá trị mất mát của model

* 1. **Kiểm thử model**

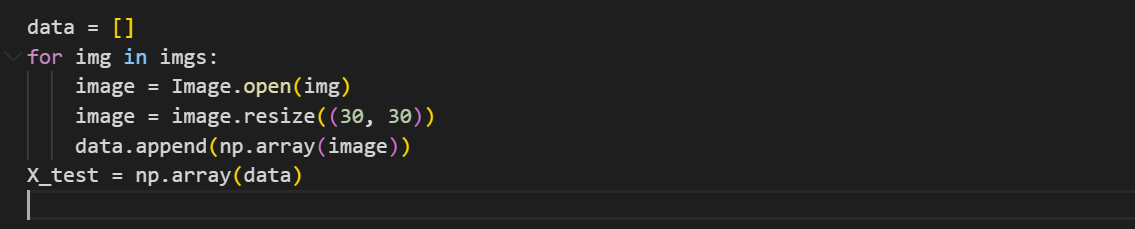
**Đọc Dữ Liệu Kiểm Thử:**

* Sử dụng thư viện pandas để đọc dữ liệu kiểm thử từ tệp CSV, trong đó chứa thông tin về các ảnh và nhãn.



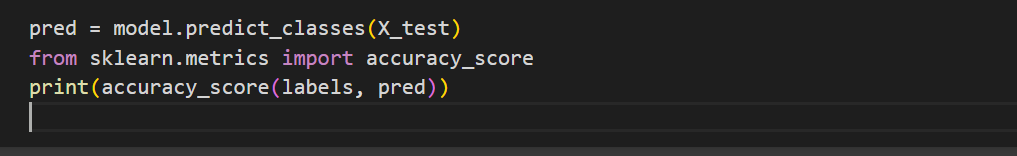
**Chuẩn Bị Dữ Liệu Kiểm Thử:**

* Đọc và chuẩn bị ảnh kiểm thử từ đường dẫn được cung cấp.
* Đảm bảo rằng dữ liệu kiểm thử được xử lý chính xác giống như trong quá trình huấn luyện.



**Dự Đoán và Đánh Giá:**

* Sử dụng mô hình đã được huấn luyện để dự đoán nhãn cho ảnh kiểm thử.
* So sánh kết quả dự đoán với nhãn thực tế để đánh giá hiệu suất của mô hình.



* 1. **Demo**
* Load hình ảnh chứa biển báo cần nhận diện



Hình 3.5 Giao diện để load hình ảnh

* Nhận diện biển báo



Hình 3.6 Giao diện đã load hình ảnh

* Kết quả sau khi nhận diện biển báo



Hình 3.7 Giao diện kết quả

1. **KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**
   1. **Kết quả và hạn chế**

Trong đồ án này, nhóm em đã đặt ra mục tiêu là nhận diện biển báo giao thông qua quá trình xây dựng model CNN. Nhóm em đã sử dụng mô hình CNN để thực hiện nhiệm vụ nhận diện này. Dưới đây là một số điểm chính và kết quả đạt được:

* + 1. **Kết quả**
* Nhóm em đã triển khai mô hình CNN, được đào tạo trên tập dữ liệu Traffic Sign Detection.
* Sử dụng các thư viện như keras, matplotlib, tensorflow,... và các thư viện hỗ trợ giao diện như Tkinter.
* Mô hình đạt được độ chính xác cao trên tập dữ liệu kiểm thử, với khả năng nhận diện chính xác biển báo giao thông trong nhiều điều kiện khác nhau.



* + 1. **Hạn chế**

**Dữ liệu chưa đa dạng:**

* Nếu tập dữ liệu không đủ lớn hoặc không đa dạng đủ, mô hình có thể không hiệu quả khi áp dụng trên các tình huống thực tế khác nhau.

**Tốc độ xử lý:**

* Mô hình Faster R-CNN có thể đưa ra dự đoán chính xác nhưng cần một lượng lớn thời gian tính toán.
* Trong môi trường thời gian thực, có thể cần xem xét các mô hình nhẹ và hiệu quả hơn để đảm bảo tốc độ xử lý nhanh chóng.
  1. **Hướng phát triển**

**Thêm Dữ Liệu Đa Dạng:**

* Thu thập thêm dữ liệu từ nhiều nguồn và điều kiện khác nhau để tăng độ đa dạng của tập dữ liệu.

**Tối Ưu Hóa Siêu Tham Số:**

* Tiến hành quá trình tối ưu hóa siêu tham số của mô hình để tìm ra các giá trị phù hợp nhất.
* Sử dụng các kỹ thuật như tìm kiếm ngẫu nhiên, tìm kiếm theo gradient, hoặc tối ưu hóa bayesian.

**Sử Dụng Mô Hình Mạng Nơ-Ron Mạnh Mẽ Hơn:**

* Thử nghiệm với các mô hình mạng nơ-ron mạnh mẽ hơn như EfficientNet, ResNeXt, hoặc mô hình pre-trained trên nhiều dữ liệu hơn.

**Tối Ưu Hóa Tốc Độ Xử Lý:**

* Thử nghiệm với các mô hình nhẹ và tối ưu hóa để đảm bảo tốc độ xử lý nhanh chóng.
* Sử dụng các kỹ thuật như quantization để giảm kích thước mô hình.

**TÀI LIỆU THAM KHẢO**

[1] GeeksforGeeks, “What is Python?”, *Sanchhaya Education Private Limited*, Last Updated: 05 Jun, 2023.

[2] GeeksforGeeks, “Libraries in Python?”, *Sanchhaya Education Private Limited*, Last Updated: 18 Oct, 2021.

[3] GeeksforGeeks, “Introduction to Deep Learning”, *Sanchhaya Education Private Limited*, Last Updated: 14 Apr, 2023.

[4] Valohai, “What Is a Machine Learning Pipeline?”, 2022.

[5] Ekin Keserer, “What Are machine Learning Pipelines, and Why Are They Important?”, 24 November 2023.

[6] GeeksforGeeks, “ML | Understanding Data Processing”, *Sanchhaya Education Private Limited*, Last Updated: 06 May, 2023.

[7] ReinTech, “Data Vailidation”, *Media*, 2023.

[8] Sarathi B, Adel Sepehr, “What are the most effective ways to validate data in image mining”, 12 November 2023.

[9] GeeksforGeeks, “What Is Feature Engineering”, *Sanchhaya Education Private Limited*, Last Updated: 21 Dec, 2023.

[10] GeeksforGeeks, “Introduction to Convolution Neural Network”, *Sanchhaya Education Private Limited*, Last Updated: 20 Dec, 2023.