BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

ĐỒ ÁN MÔN HỌC

HỌC PHẦN: TRÍ TUỆ NHÂN TẠO

ĐỀ TÀI: NHẬN DIỆN BIỂN BÁO GIAO THÔNG

KHOA: **CÔNG NGHỆ THÔNG TIN**

Sinh viên thực hiện: **Phan Thị Anh**
MSSV: 2011065095 – Lớp 20DTHB1
Kiều Thị Hà Duyên
MSSV: 2011064200 - Lớp 20DTHB2
Trần Vũ Ngọc Hiếu
MSSV: 2011063175 - Lớp 20DTHB1

Giảng viên hướng dẫn: **TS. Nguyễn Thị Hải Bình**

TP. Hồ Chí Minh, 2022

LỜI CAM ĐOAN

Trong quá trình thực hiện đề tài Báo cáo "Nhận diện biển báo giao thông", do khả năng tiếp thu thực tế còn nhiều hạn chế, kiến thức chưa sâu rộng. Mặc dù nhóm chúng em đã cố gắng hết sức nhưng chắc chắn bài báo cáo này khó tránh khỏi những thiếu sót, nhưng những gì được trình bày trong báo cáo này là sự thể hiện những kết quả đạt được dưới sự hướng dẫn của giảng viên môn Trí Tuệ Nhân Tạo – cô Nguyễn Thị Hải Bình đã nhiệt tình hướng dẫn, khuyến khích và tạo điều kiện cho chúng em hoàn thành tốt đề tài của mình.

Chúng em xin cam đoan rằng nội dung trình bày trong báo cáo đồ án, những số liệu và kết quả nghiên cứu là trung thực, hoàn toàn được thực hiện không sao chép bất kỳ nguồn nào khác. Ngoài ra, trong bài báo cáo có sử dụng một số nguồn tài liệu tham khảo đã được trích dẫn nguồn và chú thích rõ ràng. Nếu không đúng sự thật, chúng em xin chịu mọi trách nhiệm.

Sinh viên đồng thực hiện

Phan Thị Anh

Kiều Thị Hà Duyên

Trần Vũ Ngọc Hiếu

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN 1

1.1. Giới thiệu đề tài: 1

1.2. Nhiệm vụ đồ án: 1

1.2.1. Tính cấp thiết của đề tài: 1

1.2.2. Ý nghĩa khoa học và thực tiễn của đề tài: 2

1.2.3. Mục tiêu, đối tượng và phạm vi nghiên cứu của đề tài: 2

1.3. Cấu trúc đồ án: 2

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 4

2.1. Một số khái niệm cơ bản 4

2.1.1. Trí tuệ nhân tạo (AI) 4

2.1.2. Xử lý hình ảnh 4

2.1.3. Nhận dạng hình ảnh 5

2.1.4. OpenCV 6

2.2. Phương pháp đề xuất 7

2.2.1. Mô hình mạng nơ ron tích chập CNN 7

2.3. Các gói package cần cài: 9

CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM 11

3.1. Tổng quan về hệ thống: 11

3.2. Kết quả huấn luyện và đánh giá mô hình: 12

3.2.1. Bộ dữ liệu 12

3.2.2. Kết quả huấn luyện 13

3.2.3. Đánh giá mô hình 17

3.3. Chương trình Demo 18

CHƯƠNG 4: KẾT LUẬN 21

4.1. Kết quả đạt được 21

4.2. Vấn đề chưa đạt được 21

4.3. Hướng phát triển 21

TÀI LIỆU THAM KHẢO 22

DANH MỤC HÌNH VẼ, ĐỒ THỊ

Hình 1. Trí tuệ nhân tạo **4**
Hình 2. Cấu trúc thư viện OpenCV **7**
Hình 3. Convolutional Neural Networks (CNN) **7**
Hình 4. Mô hình CNN **8**
Hình 5. Sơ đồ hoạt động chương trình **12**
Hình 6. Thư mục train **13**
Hình 7. Tập dữ liệu test **13**
Hình 8. Xây dựng mô hình **14**
Hình 9. Quá trình huấn luyện **15**
Hình 10. Khởi tạo đồ thị và kết quả biểu đồ **15**
Hình 11. Biểu đồ thay đổi của training loss theo epoch **16**
Hình 12. Biểu đồ thay đổi của training Accuracy theo epoch **16**
Hình 13. Kiểm tra độ chính xác trên tập dữ liệu kiểm tra **17**
Hình 14. Độ chính xác **18**
Hình 15. Kết quả nhận dạng biển báo Đường đang thi công **18**
Hình 16. Kết quả nhận dạng biển báo Giới hạn tốc độ (60km/h) **19**
Hình 17. Kết quả nhận dạng biển báo Giao nhau với đường ưu tiên **19**

DANH MỤC TỪ VIẾT TẮT

CNN	Convolution Neural Network
IPP	Intergrated Performance Primitives
AI	Artificial Intelligence
OpenCV	Open Source Computer Vision Library
HSI	Hang Seng Index
RGB	Red Green Blue
PIL	Python Imaging Library
GTSRB	German Traffic Sign Recognition Benchmark
BTSD	Belgium Traffic Sign Dataset

CHƯƠNG 1: TỔNG QUAN

1.1. Giới thiệu đề tài:

Biển báo giao thông là quy định mà người điều khiển phương tiện khi tham gia giao thông phải tuân theo để đảm bảo an toàn cho người và phương tiện lưu thông trên đường. Một trong những thuật toán phát hiện và nhận biết các biển báo giao thông mà nhóm chúng tôi muốn nói đến là trong khi lái xe, chúng ta sẽ thấy nhiều biển báo giao thông khác nhau như đèn tín hiệu giao thông, rẽ trái hoặc phải, giới hạn tốc độ, không vượt xe nặng, cấm đi vào, trẻ em sang đường,... , mà chúng ta phải tuân theo để lái xe an toàn. Phương pháp nhận biết biển báo giao thông thuộc loại nào được gọi là phân loại biển báo giao thông. Trong dự án học sâu này, nhóm của tôi sẽ xây dựng một mô hình phân loại các biển báo giao thông có sẵn trong hình ảnh thành nhiều loại bằng cách sử dụng mạng nơ-ron phức tạp (CNN) và thư viện Keras.

Nghiên cứu của các tác giả: Wenhui Li, Daihui Li và Shangyou Zeng. Trong nhận dạng hình ảnh của biển báo giao thông, theo các đặc điểm hình ảnh của biển báo giao thông, các phương pháp phổ biến bao gồm phương pháp đối sánh mẫu truyền thống, phương pháp SVM, phương pháp Random forest và phương pháp Mạng nơ-ron tích chập (CNN) tốt nhất . Trong bài báo này, một CNN mới được đề xuất. Trích xuất tính năng, so với phương pháp CNN truyền thống, có độ chính xác cao hơn, ít tham số hơn, mô hình nhỏ hơn và đào tạo dễ dàng hơn, được đánh giá trên Điểm chuẩn nhận dạng biển báo giao thông của Đức (GTSRB) và Bộ dữ liệu biển báo giao thông của Bỉ (BTSD). Kết quả cho thấy phương pháp này ưu việt hơn phương pháp CNN truyền thống trong việc xác định lưu lượng.

1.2. Nhiệm vụ đồ án:

1.2.1. Tính cấp thiết của đề tài:

Biển báo giao thông là một phần thiết yếu của cơ sở hạ tầng đường bộ. Chúng cung cấp cho người đi đường những thông tin quan trọng, đôi khi là những khuyến nghị thuyết phục và do đó khuyến khích họ điều chỉnh hành vi lái xe của mình để tuân thủ các quy định hiện hành về giao thông đường bộ. Nếu không có những biển báo hữu ích này,

rất có thể chúng ta sẽ gặp nhiều tai nạn hơn, vì người lái xe sẽ không nhận được phản hồi quan trọng về tốc độ họ có thể lái xe an toàn, cũng như họ sẽ không được thông báo về công trình đường bộ, khúc cua gấp hoặc giao cắt trường học. Vì những lý do trên, nhóm chúng tôi muốn tạo một ứng dụng đơn giản và nhỏ gọn cho phép người dùng tìm kiếm thông tin biển báo một cách trực quan khi họ không nhớ nội dung của biển báo đó. Một bước phát triển nữa có thể là tích hợp hoặc kết nối các phương tiện giao thông và đưa ra các cảnh báo thích hợp.

1.2.2. Ý nghĩa khoa học và thực tiễn của đề tài:

- Để giúp người đi đường không bỏ sót các biển báo giao thông và dễ dàng nhận ra chúng, nhóm chúng tôi đã nghiên cứu một cách hiệu quả để kiểm tra một số hình ảnh có chứa các biển báo giao thông trong thời gian thực.
- Hỗ trợ lái xe nhận biết biển báo giao thông trên đường bằng cách hiển thị tên biển báo mỗi khi đi qua, được tích hợp với hệ thống của xe tự lái.

1.2.3. Mục tiêu, đối tượng và phạm vi nghiên cứu của đề tài:

- Đề tài giải quyết bài toán: Xây dựng chương trình thực nghiệm với giao diện sử dụng ngôn ngữ tiếng việt để phát hiện và nhận diện biển báo giao thông.
- Phạm vi nghiên cứu: Dữ liệu xử lý là tệp hình ảnh thu được từ Internet. Ảnh chụp trong điều kiện ánh sáng bình thường, được chụp theo nhiều định dạng, hướng khác nhau vừa mờ, vừa nghiêng, độ rung trong khung ảnh để hệ thống dễ dàng nhận diện biển báo đúng ở vị trí, khung cảnh khác nhau.

1.3. Cấu trúc đồ án:

Đồ án "Phát hiện và nhận diện biển báo giao thông" bao gồm tất cả 4 phần.

Chương 1 – Tổng quan: Giới thiệu khái quát và mục tiêu của đề tài. Giới thiệu về các kiến thức nền tảng cũng như công nghệ và phần mềm được sử dụng trong đề tài.

Chương 2 – Cơ sở lý thuyết: Trình bày các lý thuyết có liên quan tới đề tài biển báo giao thông Việt Nam, phương pháp đề xuất và từ đó triển khai thuật toán xây dựng để giải quyết bài toán nhận dạng biển báo.

Chương 3 – Kết quả thực nghiệm: Trình bày tổng quan về hệ thống, xây dựng một chương trình để nhận dạng biển báo giao thông và chạy demo.

Chương 4 – Kết luận: Tổng kết quá trình thực hiện và rút ra hướng phát triển sau này của đồ án.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

2.1. Một số khái niệm cơ bản

2.1.1. Trí tuệ nhân tạo (AI)

Trí tuệ nhân tạo là trí tuệ máy móc (nhân tạo) do con người tạo ra. Kể từ khi chiếc máy tính điện tử đầu tiên ra đời, các nhà khoa học máy tính đã nghiên cứu để khám phá ra một hệ thống máy tính (cả phần cứng và phần mềm) giúp nó có khả năng thông minh như con người.

Thông qua trí tuệ nhân tạo AI, các thiết bị điện tử - máy móc có hành vi, khả năng thích ứng và học hỏi thông minh. Để đảm bảo hoạt động trơn tru và mượt mà hơn các thiết bị thông thường.

Hình 1. Trí tuệ nhân tạo

2.1.2. Xử lý hình ảnh

Xử lý hình ảnh là một trong những lĩnh vực quan trọng nhất của kỹ thuật thị giác máy tính và là cơ sở của nhiều nghiên cứu trong lĩnh vực này. Hai nhiệm vụ chính của xử

lý ảnh là nâng cao chất lượng thông tin ảnh và xử lý dữ liệu để hỗ trợ các quá trình khác, bao gồm cả việc áp dụng tầm nhìn để điều khiển.

Xử lý nhận dạng hình ảnh được coi là quá trình xử lý một hình ảnh đầu vào để thu được kết quả mong muốn. Kết quả của quá trình xử lý ảnh có thể là ảnh "đã qua xử lý" hoặc một bản tóm tắt.

Một hình ảnh có thể được xem như một tập hợp các pixel và mỗi pixel được xem như một đặc điểm ánh sáng cụ thể hoặc điểm mốc tại một vị trí cụ thể của đối tượng trong không gian và có thể được biểu thị dưới dạng hàm của n biến P (c, c1, c2, ...). Do đó, trong quá trình xử lý ảnh, ảnh có thể được coi là ảnh n chiều.

2.1.3. Nhận dạng hình ảnh

Ngày nay, công nghệ sử dụng nhận dạng hình ảnh rất phổ biến. Điện thoại thông minh sử dụng công nghệ này để nhận dạng khuôn mặt; Nó cũng được sử dụng trong ô tô tự lái, hình ảnh y tế và hơn thế nữa. Hệ thống thị giác cho phép máy tính đưa ra quyết định chính xác dựa trên những gì chúng nhìn thấy. Điều này thường được gọi là thị giác máy tính.

Học sâu là một phương pháp thị giác máy tính tự động hóa mọi thứ mà thị giác của con người có thể làm. Nhận dạng hình ảnh ngày nay tương đương với thị giác của con người. Mọi người và doanh nghiệp sử dụng nó cho các mục đích khác nhau hàng ngày. Ví dụ điển hình như: các chuyên gia y tế sử dụng hệ thống phân bổ để phục vụ bệnh nhân tốt hơn. Điều này giúp họ dự đoán các vấn đề sức khỏe, phát hiện bệnh và cung cấp các dịch vụ tập trung vào bệnh nhân hơn cả tại chỗ và trực tuyến. Các chuyên gia khác chỉ ra rằng nhận dạng hình ảnh trong bán lẻ giúp họ quét cơ sở dữ liệu lớn. Điều này cho phép họ đáp ứng tốt hơn nhu cầu của khách hàng và cải thiện trải nghiệm khách hàng tại cửa hàng và trực tuyến.

Bản chất của việc phát hiện và nhận diện biển báo là lấy một hình ảnh đầu vào thô và cung cấp một thuật toán có khả năng nhận ra nội dung của hình ảnh biển báo và hiển thị

rõ ràng cho từng đối tượng. Máy móc không thể nhìn và nhận dạng hình ảnh biển báo theo cách con người chúng ta có thể.

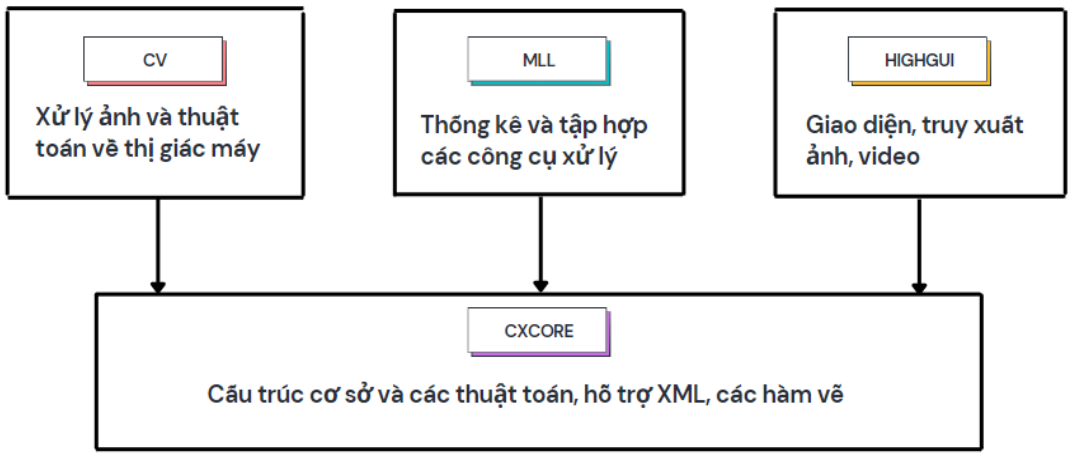
2.1.4. OpenCV

OpenCV (Open Source Computer Vision Library) là một thư viện phần mềm máy tính và thị giác máy tính mã nguồn mở. OpenCV được xây dựng để cung cấp một cơ sở hạ tầng chung cho các ứng dụng thị giác máy tính và để tăng tốc việc sử dụng cảm giác máy trong các sản phẩm thương mại. Là một sản phẩm được cấp phép Apache 2, OpenCV giúp các doanh nghiệp dễ dàng sử dụng và sửa đổi mã.

Thư viện OpenCV bao gồm một số tính năng nổi bật như:

- Bộ công cụ hỗ trợ 2D và 3D
- Phát hiện và nhận dạng khuôn mặt
- Xác định đối tượng
- Nhận diện cử chỉ
- Nhận dạng chuyển động, đối tượng, hành vi,
- Tương tác giữa con người và máy tính
- Điều khiển Robot
- Loại bỏ mắt đỏ khỏi hình ảnh được chụp bằng đèn flash

Cấu trúc cơ bản của OpenCV: CV chủ yếu bao gồm xử lý hình ảnh, phân tích cấu trúc hình ảnh, chuyển động và theo dõi, nhận dạng mẫu, hiệu chỉnh máy ảnh,... MLL chứa nhiều chức năng để phân cụm, phân loại và phân tích dữ liệu. HighGUI cung cấp giao diện người dùng đồ họa và lưu trữ hình ảnh. CXCore bao gồm cấu trúc dữ liệu, đại số ma trận, tính bền vững của đối tượng, xử lý lỗi, vẽ và toán học cơ bản. Nó cũng hoạt động để tải mã động.



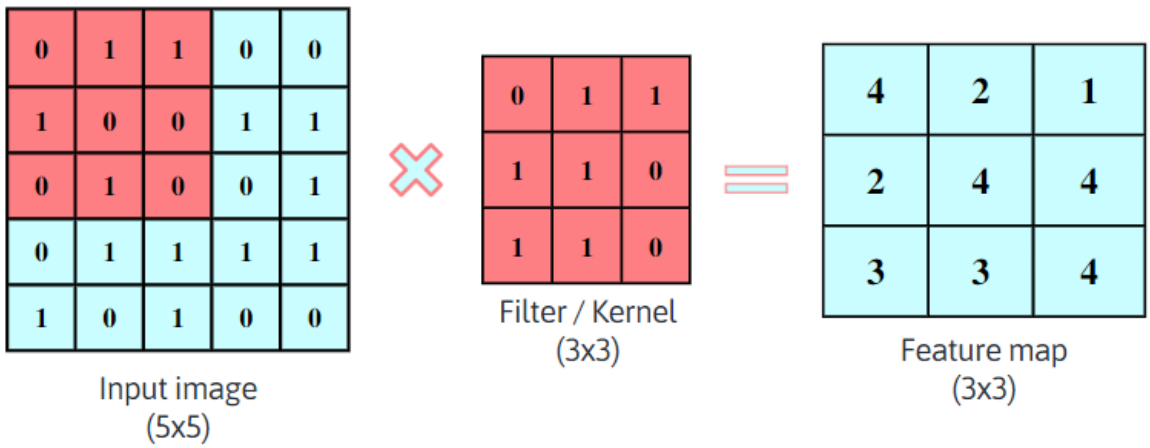
Hình 2. Cấu trúc thư viện OpenCV

2.2. Phương pháp đề xuất

2.2.1. Mô hình mạng nơ ron tích chập CNN

Mạng nơ-ron tích chập (CNN) là mạng nơ-ron học sâu được thiết kế để xử lý các mảng dữ liệu có cấu trúc như hình ảnh. Thường được sử dụng trong thị giác máy tính, mạng nơ-ron tích tụ rất phổ biến trong nhiều ứng dụng trực quan như phân loại hình ảnh, và cũng đã được chứng minh là hữu ích trong xử lý ngôn ngữ tự nhiên để phân loại văn bản.

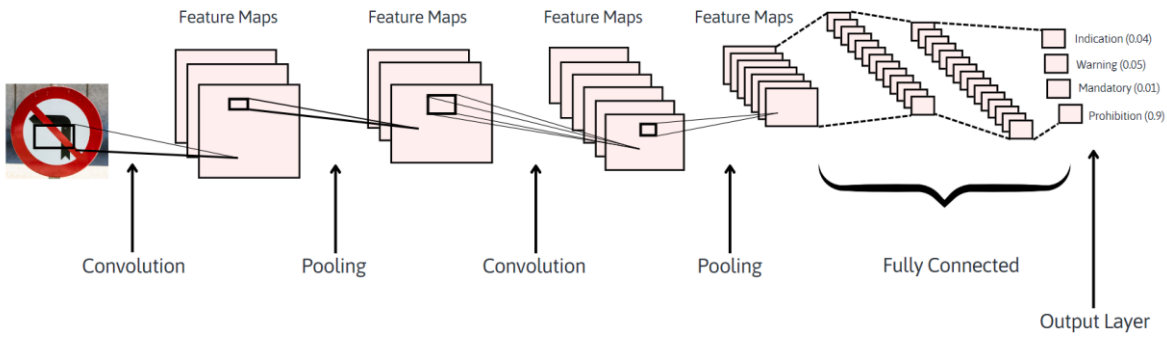
Sử dụng thuật toán CNN được thiết kế sẵn để đào tạo lại và điều chỉnh nhằm giải quyết bài toán.



Hình 3. Convolutional Neural Networks (CNN)

Kiến trúc của mạng nơ-ron tích tụ là một mạng nơ-ron truyền về phía trước nhiều lớp, được thực hiện bằng cách xếp chồng nhiều lớp ẩn chồng lên nhau theo trình tự. Chính thiết kế tuần tự này cho phép các mạng nơ-ron tích tụ tìm hiểu các tính năng phân cấp.

Các lớp ẩn thường là các lớp phức tạp theo sau là các lớp kích hoạt, một số trong số chúng tiếp theo là các lớp gộp.



Hình 4. Mô hình CNN

Cấu trúc của mạng CNN:

Mạng CNN là một tập hợp các lớp Convolution chồng lên nhau nhau kích hoạt trọng số node bằng cách sử dụng các hàm nonlinear activation (chẳng hạn như ReLU và tanh). Mỗi lớp tạo ra nhiều thông tin trừu tượng hơn cho lớp tiếp theo sau khi thông qua các hàm kích hoạt.

Mỗi lớp tạo ra nhiều thông tin trừu tượng hơn cho lớp tiếp theo sau khi thông qua các hàm kích hoạt. Trong mô hình mạng truyền ngược (feedforward neural network) thì mỗi nơ-ron đầu vào (input node) cho mỗi nơron đầu ra trong các lớp tiếp theo. Mô hình này gọi là mạng kết nối đầy đủ (fully connected layer).

Layer tiếp theo là kết quả convolution từ layer trước, vì vậy nó có các kết nối cục bộ. Do đó, mỗi nơ-ron trong lớp tiếp theo được tạo ra từ kết quả của bộ lọc được áp dụng cho vùng ảnh cục bộ của nơ-ron trước đó.

Mỗi lớp sử dụng một bộ lọc khác nhau thường sử dụng hàng trăm nghìn bộ lọc như vậy và kết hợp các kết quả của chúng. Ngoài ra, còn có một số lớp khác như

pooling/subsampling layer được sử dụng để lọc ra nhiều thông tin hữu ích hơn (loại bỏ thông tin nhiễu).

Cấu trúc cơ bản nhất của CNN sẽ bao gồm 3 phần chủ yếu, đó là:

- Local receptive field (trường cục bộ): Công việc của trường cục bộ là tách và lọc dữ liệu và thông tin hình ảnh để chọn ra những vùng hình ảnh hữu ích nhất.
- Shared weights and bias (trọng số chia sẻ): Trong mạng CNN, thành phần này có tác dụng lớn đến việc giảm thiểu số lượng tham số. Trong mỗi convolution sẽ chứa nhiều feature map khác nhau, mỗi feature lại có khả năng giúp nhận diện một số feature trong ảnh.
- Pooling layer (lớp tổng hợp): Pooling layer là lớp cuối cùng có chức năng đơn giản hóa thông tin đầu ra. Sau khi tính toán và quét qua các lớp, pooling layer được tạo để loại bỏ thông tin không cần thiết và tối ưu hóa kết quả đầu ra. Điều này cho phép người dùng có được kết quả hài lòng theo yêu cầu và mong muốn của họ.

2.3. Các gói package cần cài:

- Numpy: NumPy là một gói cơ bản cho tính toán khoa học bằng Python. Đây là một thư viện Python cung cấp các đối tượng mảng đa chiều, các đối tượng dẫn xuất khác nhau và một tập hợp các quy trình để thao tác nhanh với mảng.
- Matplotlib: Matplotlib là một thư viện toàn diện để tạo hình ảnh trực quan tĩnh, động và tương tác trong Python. Matplotlib biến những điều đơn giản trở nên dễ dàng và những điều khó khăn trở thành hiện thực.
- OpenCV: OpenCV là một thư viện xử lý hình ảnh, máy học và thị giác máy tính mã nguồn mở khổng lồ đóng một vai trò quan trọng trong các hoạt động thời gian thực ngày nay.
- Scikit-image: Scikit-image là một gói Python xử lý hình ảnh hoạt động với mảng NumPy, một bộ thuật toán xử lý hình ảnh.
- Tensorflow: TensorFlow là một thư viện phần mềm mã nguồn mở và miễn phí dành cho máy học và trí tuệ nhân tạo. Nó có thể được sử dụng cho nhiều nhiệm vụ khác nhau, đặc biệt tập trung vào đào tạo và suy luận mạng nơ-ron sâu.

- Keras: Keras là một API học sâu cấp cao do Google phát triển để triển khai mạng nơ-ron. Nó được viết bằng Python và được sử dụng để tạo điều kiện thuận lợi cho việc triển khai mạng thần kinh.
- Pillow: Pillow là một nhánh của thư viện cũ có tên PIL, là thư viện ban đầu cho phép Python xử lý hình ảnh. Nó cung cấp các tính năng xử lý hình ảnh tương tự như các tính năng được tìm thấy trong phần mềm xử lý hình ảnh.

CHƯƠNG 3: KẾT QUẢ THỰC NGHIỆM

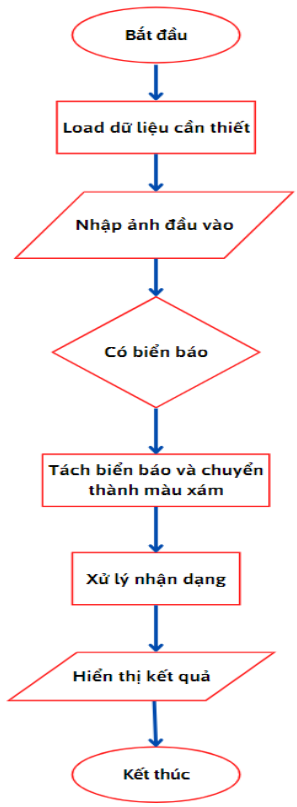
3.1. Tổng quan về hệ thống:

Đồ án này trình bày một phương pháp phát hiện và nhận diện biển báo giao thông kết hợp công nghệ phát hiện cạnh và phân tích hình dạng đối tượng để phát hiện các đối tượng biển báo giao thông tiềm ẩn. Sau đó, việc khai thác và đào tạo tính năng được thực hiện bởi một thuật toán CNN để nhận ra các điểm mốc.

Nhận biết biển báo giao thông: Có nhiều loại biển báo giao thông, chẳng hạn như biển báo hạn chế tốc độ, biển cấm đi vào, đèn giao thông, rẽ trái hoặc phải, trẻ em sang đường, giới hạn trọng lượng xe,... Nhận biết nội dung của tín hiệu và xác định loại tín hiệu.

Từ những hình ảnh thu được, nhóm của chúng tôi dựa vào các đặc điểm màu sắc và áp dụng thư viện OpenCV để xác định màu của một biển báo cụ thể, phân loại và xác định đó là biển báo gì. Sau đó, nó cung cấp thông tin về các ký tự được phát hiện và nhận diện.

Quy trình bài toán: Đầu tiên, các hình ảnh đầu vào là một logo và một hình nền. Sau khi vào hệ thống, hình ảnh được phân đoạn để loại bỏ nền và làm nổi bật các đối tượng có thể là biển báo, các đối tượng này được lọc để chọn vùng ứng viên làm biển báo giao thông. Các hình ảnh biển báo giao thông bị phân tách sau đó được đưa vào một mô hình CNN đã được đào tạo trước để xác định loại biển báo. Cuối cùng, hệ thống hiển thị thông tin cần cho người dùng.



Hình 5. Sơ đồ hoạt động chương trình

3.2. Kết quả huấn luyện và đánh giá mô hình:

3.2.1. Bộ dữ liệu

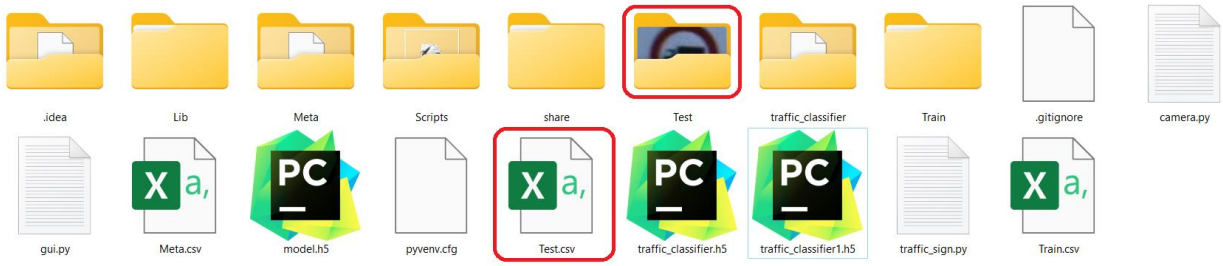
Tập dữ liệu chứa hơn 50.000 hình ảnh về các biển báo giao thông khác nhau, nó được phân loại thành 43 lớp. Các bộ dữ liệu khá khác nhau, với một số lớp có nhiều hình ảnh và một số lớp có rất ít hình ảnh. Tập dữ liệu có một thư mục train chứa các hình ảnh bên trong mỗi lớp và một thư mục test chứa các hình ảnh dùng để thử nghiệm mô hình.

- Thư mục train có chứa 43 thư mục con mỗi thư mục đại diện cho một lớp khác nhau được đánh số thứ tự từ 0 đến 42.

Hình 6. Thư mục train

Bộ dữ liệu chứa (39209, 30, 30, 3), có nghĩa là có 39209 hình ảnh, mỗi ảnh có kích thước 30x30 pixel, số 3 ở cuối thể hiện dữ liệu là hình ảnh màu (giá trị RGB) về các biển báo giao thông khác nhau.

- Tập dữ liệu của chúng tôi có chứa 1 thư mục test và 1 file test.csv.



Hình 7. Tập dữ liệu test

3.2.2. Kết quả huấn luyện

Nhóm chúng tôi xây dựng một mô hình CNN để phân loại các hình ảnh thành các danh mục tương ứng của chúng.

Kiến trúc mô hình CNN của chúng tôi như sau:

- 2 lớp Conv2D (filter=32, kernel_size=(5,5), activation="relu")
- Lớp MaxPool2D (pool_size=(2,2))
- Lớp Dropout (rate=0.25)

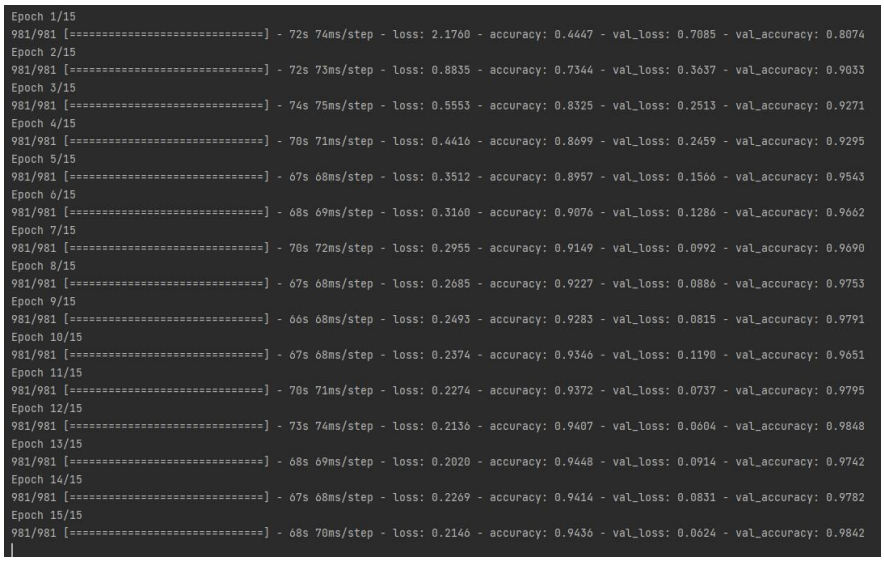
- 2 lớp Conv2D (filter=64, kernel_size=(3,3), activation=”relu”)
- Lớp MaxPool2D (pool_size=(2,2))
- Lớp Dropout (rate=0.25)
- Làm phẳng lớp để ép các lớp thành 1 chiều
- Lớp Dense Fully connected (256 nodes, activation=”relu”)
- Lớp Dropout (rate=0.5)
- Lớp Dense (43 nodes, activation=”softmax”)

```
# Xây dựng mô hình
model = Sequential()
model.add(Conv2D(filters=32, kernel_size=(5, 5), activation='relu', input_shape=X_train.shape[1:]))
model.add(Conv2D(filters=32, kernel_size=(5, 5), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(Conv2D(filters=64, kernel_size=(3, 3), activation='relu'))
model.add(MaxPool2D(pool_size=(2, 2)))
model.add(Dropout(rate=0.25))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(rate=0.5))
model.add(Dense(43, activation='softmax'))
```

Hình 8. Xây dựng mô hình

Mô tả quá trình training:

- Kết quả của quá trình huấn luyện: Sau 15 epochs, mô hình cho độ chính xác trên tập huấn luyện X_train khá cao.



Hình 9. Quá trình huấn luyện

Cụ thể: Mức độ training loss càng giảm xuống từ 2.1760 xuống 0.2146 và mức độ training accuracy càng tăng lên từ 0.4447 lên đến 0.9436.

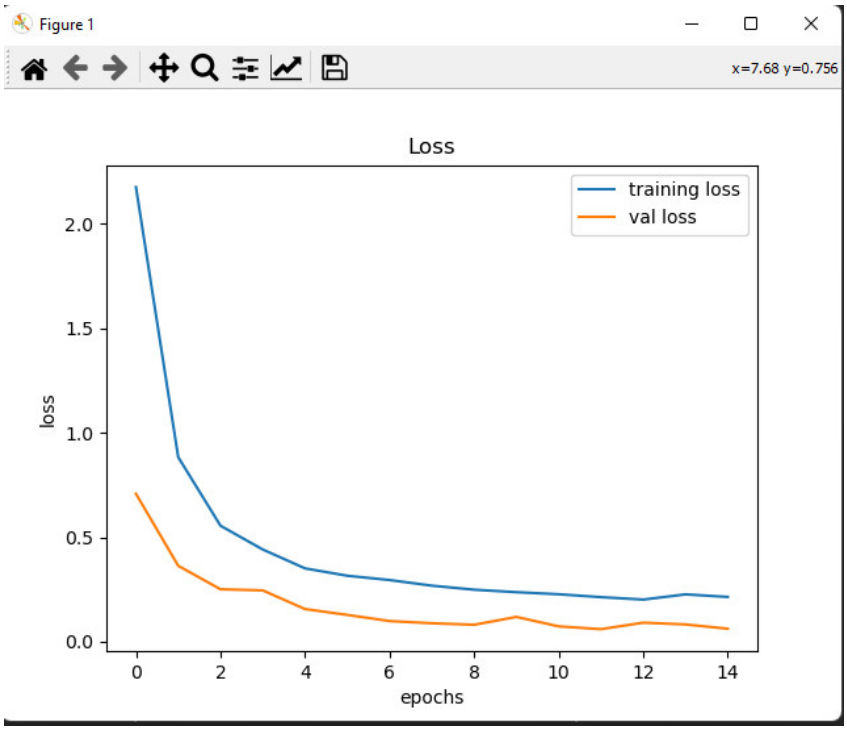
Mô hình của nhóm chúng tôi đã đạt đến độ chính xác gần 95% trên tập training set.

- Khởi tạo đồ thị kết quả biểu đồ:

```
# vẽ đồ thị về độ chính xác
plt.figure(0)
plt.plot(history.history['accuracy'], label='training accuracy')
plt.plot(history.history['val_accuracy'], label='val accuracy')
plt.title('Accuracy')
plt.xlabel('epochs')
plt.ylabel('accuracy')
plt.legend()
plt.show()
# vẽ đồ thị về độ thất bại
plt.figure(1)
plt.plot(history.history['loss'], label='training loss')
plt.plot(history.history['val_loss'], label='val loss')
plt.title('Loss')
plt.xlabel('epochs')
plt.ylabel('loss')
plt.legend()
plt.show()
```

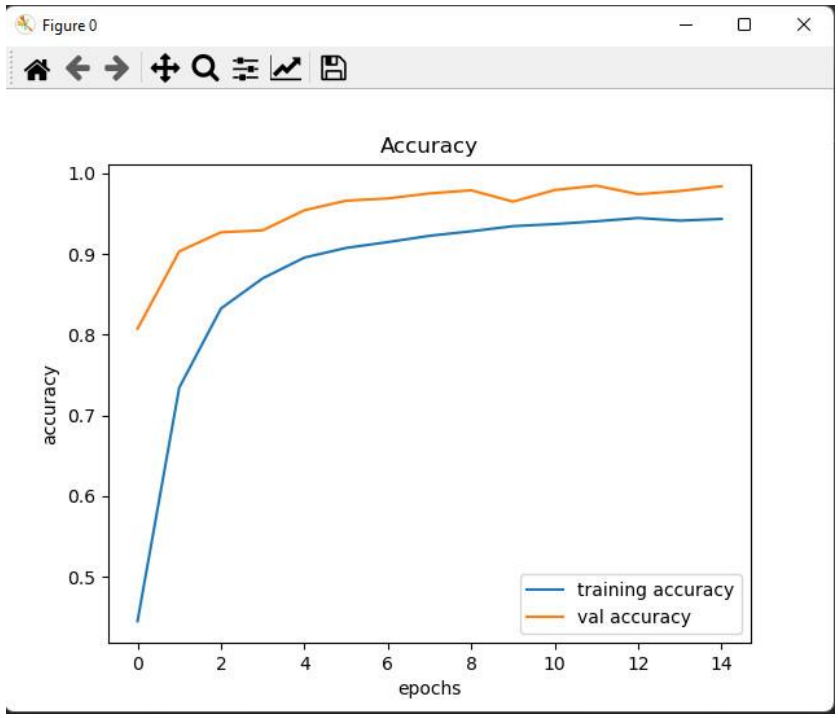
Hình 10. Khởi tạo đồ thị và kết quả biểu đồ

Biểu đồ thay đổi của training loss theo epoch để minh họa:



Hình 11. Biểu đồ thay đổi của training loss theo epoch

Biểu đồ thay đổi của training Accuracy theo epoch để minh họa:



Hình 12. Biểu đồ thay đổi của training Accuracy theo epoch

3.2.3. Đánh giá mô hình

Chúng ta cần phải trích xuất hình ảnh và nhãn tương ứng bằng cách sử dụng pandas. Sau đó, để dự đoán mô hình, chúng ta phải thay đổi kích thước hình ảnh thành 30 × 30 pixel và tạo một mảng numpy chứa tất cả dữ liệu hình ảnh. Sử dụng precision_score của sklearn.metrics để dự đoán các nhãn của bộ test. Có thể thấy ta đã đạt được độ chính xác là 96% trong mô hình này.

Kiểm tra độ chính xác trên tập dữ liệu kiểm tra:

```
# kiểm tra độ chính xác trên tập dữ liệu kiểm tra
from sklearn.metrics import accuracy_score

y_test = pd.read_csv('Test.csv')

labels = y_test["ClassId"].values
imgs = y_test["Path"].values

data = []

for img in imgs:
    image = Image.open(img)
    image = image.resize((30, 30))
    data.append(np.array(image))

X_test = np.array(data)

pred = model.predict_classes(X_test)

# Độ chính xác của dữ liệu thử nghiệm
from sklearn.metrics import accuracy_score

print(accuracy_score(labels, pred))
```

Hình 13. Kiểm tra độ chính xác trên tập dữ liệu kiểm tra

Hình 14. Độ chính xác

Dựa vào hình mô tả quá trình training, gồm 15 epoch cho thấy thời gian xử lý là 17p3s.

3.3. Chương trình Demo

- Giao diện:

Hình 15. Kết quả nhận dạng biển báo Đường đang thi công

Hình 16. Kết quả nhận dạng biển báo Giới hạn tốc độ (60km/h)

Hình 17. Kết quả nhận dạng biển báo Giao nhau với đường ưu tiên

- Chức năng: Chương trình Demo sẽ hiển thị kết quả nhận diện ra biển báo giao thông khi bạn đưa hình biển báo lên hệ thống.
- Môi trường hoạt động của chương trình demo: Được demo trên phần mềm Pycharm, ngôn ngữ Python phiên bản 3.9.

CHƯƠNG 4: KẾT LUẬN

4.1. Kết quả đạt được

- Tìm hiểu được các kĩ thuật xử lý ảnh trong nhận diện hình học, tạo ra một chương trình nhận diện biển báo giao thông.
- Đã xây dựng thành công được chương trình nhận diện biển báo giao thông với độ chính xác 95%. Điều này khá tốt cho một mô hình CNN đơn giản.

4.2. Vấn đề chưa đạt được

- Vì mang tính nghiên cứu nên hệ thống chỉ mới làm việc trên tập dữ liệu thử nghiệm với số ít loại biển báo khác nhau.
- Giao diện nhóm tôi còn một số hạn chế nhất định: Không thể nhận diện qua camera, giao diện chưa đẹp mắt,...
- Ứng dụng còn bị hạn chế khi làm việc với một số phần cứng không đáp ứng được các yêu cầu về xử lý.

4.3. Hướng phát triển

- Trong tương lai, nhóm có thể sẽ nghiên cứu cải tiến phương pháp phát hiện vùng chứa biển báo để giải quyết trường hợp các biển báo bị hư hỏng hoặc chồng lấp.
- Tăng số lượng mẫu huấn luyện và kiểm tra để nâng cao độ chính xác của hệ thống.
- Mở rộng hệ thống để phát hiện và nhận dạng thêm các kiểu biển báo khác.
- Đưa công nghệ AI nhận diện biển báo giao thông vào thực tiễn, giúp giảm thiểu tai nạn giao thông và các vấn đề về an toàn giao thông hiện nay.

TÀI LIỆU THAM KHẢO

1. Nghiên cứu của các tác giả: Wenhui Li , Daihui Li và Shangyou Zeng. Traffic Sign Recognition with a small convolutional neural network. https://iopscience.iop.org/article/10.1088/1757-899X/688/4/044034
2. [Phân loại biển báo giao thông bằng Deep Learning (CNN)] https://miai.vn/2020/08/03/phan-loai-bien-bao-giao-thong-bang-deep-learning-cnn/
3. Traffic Signs Recognition using CNN and Keras in Python: Nhận dạng biển báo giao thông bằng CNN và Keras trong Python (analyticsvidhya.com)
4. Recognising Traffic Signs With 98% Accuracy Using Deep Learning: Recognising Traffic Signs With 98% Accuracy Using Deep Learning | by Eddie Forson | Towards Data Science
5. Traffic Sign Classification with Keras and Deep Learning: Phân loai biển báo giao thông với Keras và Deep Learning - PyImageSearch
6. Python Project on Traffic Signs Recognition with 95% Accuracy using CNN & Keras - DataFlair
7. https://imageio.forbes.com/specials-images/imageserve/5f4339cb93822f4b6e1e74b8/7-Successful-Ways-To-Use-Artificial-Intelligence-To-Improve-Your-Business-Processes/960x0.jpg?format=jpg&width=960
8. Nhận diện biển báo giao thông bằng CNN và Keras: https://tek4.vn/nhan-dien-bien-bao-giao-thong-bang-cnn-keras
9. Tập dữ liệu có sẵn tại kaggle https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign
10. Thuật toán CNN là gì? Cấu trúc mạng Convolutional Neural Network: https://topdev.vn/blog/thuat-toan-cnn-convolutional-neural-network/