

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA VẬT LÝ**



**ĐỒ ÁN CUỐI KỲ HỌC PHÂN THỊ GIÁC MÁY TÍNH
SIGN LANGUAGE DETECTION AND CLASSIFY**

Sinh viên thực hiện: Chu Phạm Đình Tú

Nguyễn Văn Hùng

Phùng Phúc Hậu

Hà Nội - 2023

Lời cảm ơn

Lời đầu tiên nhóm em xin được cảm ơn thầy Phạm Tiến Lâm cùng anh trợ giảng Đặng Văn Báu đã nhiệt tình giảng dạy và hướng dẫn môn Thị Giác Máy Tính. Các thầy đã cung cấp cho em những kiến thức cơ bản quan trọng để phục vụ trong quá trình nghiên cứu. Chúng em rất trân trọng khoảng thời gian cùng thầy nghiên cứu môn học.

Do thời gian thực hiện có hạn kiến thức còn nhiều hạn chế nên bài làm của chúng em chắc chắn không tránh khỏi những thiếu sót nhất định. Chúng em rất mong nhận được sự đóng góp, phê bình của các thầy để đề tài của chúng em được hoàn thiện hơn. Cuối cùng chúng em xin kính chúc hai Thầy nhiều sức khỏe , thành công và hạnh phúc ,...

Mục lục

Lời cảm ơn	i
Danh sách hình vẽ	iii
MỞ ĐẦU	1
1 Tổng quan bộ dữ liệu	3
1.1 Bộ dữ liệu Sign language MNIST	3
1.2 Dữ liệu bổ sung	3
1.3 Dữ liệu sử dụng cho mô hình Yolo	3
2 Mô hình phát hiện bàn tay	5
2.1 Mô hình Yolov7	5
2.2 Mô hình Yolov8	7
2.2.1 Modes	8
2.2.2 Task	10
2.2.3 Models YOLOv8	11
2.3 So sánh YOLOv7 và YOLOv8	14
2.3.1 Điểm mấu chốt	14
2.3.2 So sánh hiệu suất YOLOv8 và YOLOv7	15
3 Xây dựng mô hình phân loại ký hiệu tay	17
3.1 Xây dựng mô hình CNN	17
3.2 Kết quả và đánh giá	19
4 Kết hợp kết quả của mô hình Yolov8 và CNN	21
4.1 Kết hợp hai mô hình	21
4.2 Ghép các ký tự thành từ, câu	22
5 KẾT LUẬN	23
5.1 Tóm tắt nội dung, kết quả đạt được và một số hạn chế cần khắc phục . .	23
5.2 Tiềm năng ứng dụng của đề tài	25

Danh sách hình vẽ

1.1	Roboflow	4
1.2	Chia tập dữ liệu	4
2.1	Kết quả mô hình yolov7	6
2.2	Tổng quan về các biến thể mô hình YOLOv8	12
2.3	Detection (COCO)	12
2.4	Results YOLOv8	12
2.5	Bảng so sánh	15
2.6	Tổng quan cấp cao về so sánh hiệu suất giữa YOLOv8 và YOLOv7 trên nền tảng nhúng.	16
3.1	Kết quả của 10 epochs cuối	19
3.2	Đồ thị accuracy giữa tập train và test	19
3.3	Đồ thị loss giữa tập train và test	20
4.1	Ví dụ viết một câu từ các ký tự	22
5.1	Kết quả nhận diện bàn tay của mô hình Yolov7	23
5.2	Kết quả nhận diện bàn tay của mô hình Yolov8	24
5.3	Kết quả phân loại thời gian thực của mô hình CNN	25

MỞ ĐẦU

Phát hiện và phân loại ngôn ngữ ký hiệu là một lĩnh vực nghiên cứu và ứng dụng của trí tuệ nhân tạo và công nghệ máy tính, nhằm nhận diện và hiểu các biểu đạt và cử chỉ trong ngôn ngữ ký hiệu được sử dụng bởi người khiếm thính/câm. Mục tiêu chính của phát hiện ngôn ngữ ký hiệu là tạo ra các hệ thống tự động có khả năng nhận biết và chuyển đổi ngôn ngữ ký hiệu thành ngôn ngữ tự nhiên hoặc thông tin khác có thể được sử dụng để giao tiếp và tương tác.

Để thực hiện phát hiện ngôn ngữ ký hiệu, các công nghệ và phương pháp như máy học, học sâu (deep learning), thị giác máy tính và phân tích chuyển động thường được sử dụng. Các hệ thống phát hiện ngôn ngữ ký hiệu hoạt động dựa trên dữ liệu đầu vào từ các thiết bị camera.

Trong dự án này, nhóm đã sử dụng mô hình Yolov7 và Yolov8 để phát hiện bàn tay, sau đó sử dụng mô hình CNN để phân loại ngôn ngữ ký hiệu, sử dụng bảng chữ cái ASL dựa trên mô hình trong học máy. Phát hiện ngôn ngữ ký hiệu có tiềm năng lớn trong việc nâng cao khả năng giao tiếp và tương tác giữa người khiếm thính/câm và những người không biết ngôn ngữ ký hiệu, cũng như đem lại sự tiện lợi và đa dạng trong việc truyền đạt thông tin.

Qua đồ án này chúng em mong muốn đóng góp một phần nhỏ sức lực vào sự phát triển công nghệ và giúp đỡ cho những người khiếm thính và câm.

Trong các phần tiếp theo của báo cáo này, chúng em sẽ trình bày chi tiết về phương pháp nghiên cứu và xây dựng mô hình, cũng như kết quả và phân tích từ các phần kiểm tra.

- **Chương 1: Tổng quan về bộ dữ liệu**

Sign language MNIST

Dữ liệu bổ sung

- **Chương 2: Mô hình phát hiện bàn tay**

Mô hình Yolov7

Mô hình Yolov8

So sánh hai mô hình

- **Chương 3: Xây dựng mô hình phân loại ký hiệu tay**

Xây dựng mô hình CNN

Kết quả và đánh giá

- **Chương 4: Kết hợp kết quả của mô hình Yolov8 và CNN**

Kết hợp hai mô hình

Ghép các ký tự thành từ, câu

- **Chương 5: Kết luận**

Tóm tắt nội dung và kết quả đạt được

Hạn chế chưa khắc phục được

Tiềm năng ứng dụng của đề tài

Chương 1 Tổng quan bộ dữ liệu

1.1 Bộ dữ liệu Sign language MNIST

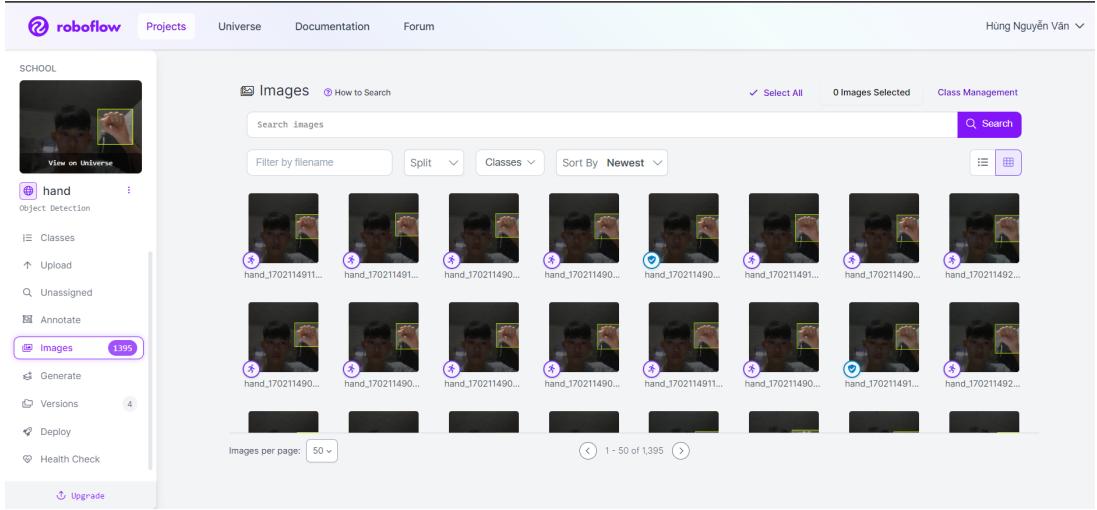
Bộ dữ liệu Sign language MNIST được phát hành dưới định dạng file csv gồm hai file train và test. Tập train có kích thước 27455 hàng và 785 cột, tập test có 7172 hàng và 785 cột. Số hàng tương ứng với số lượng ảnh trong bộ dữ liệu, 785 cột bao gồm cột đầu tiên là tên class và 784 cột còn lại là giá trị độ sáng của mỗi pixel. Trong bộ dữ liệu có 24 class là các chữ cái Mỹ Latinh: a, b, c ... x, y (không có ký tự 'j' và 'z'). Class có số ảnh ít nhất là 'e' với 957 ảnh, class có số ảnh nhiều nhất là 'r' với 1294 ảnh, các class còn lại đều có số lượng trên 1000 ảnh. Mỗi ảnh có kích thước là (28, 28, 1).

1.2 Dữ liệu bổ sung

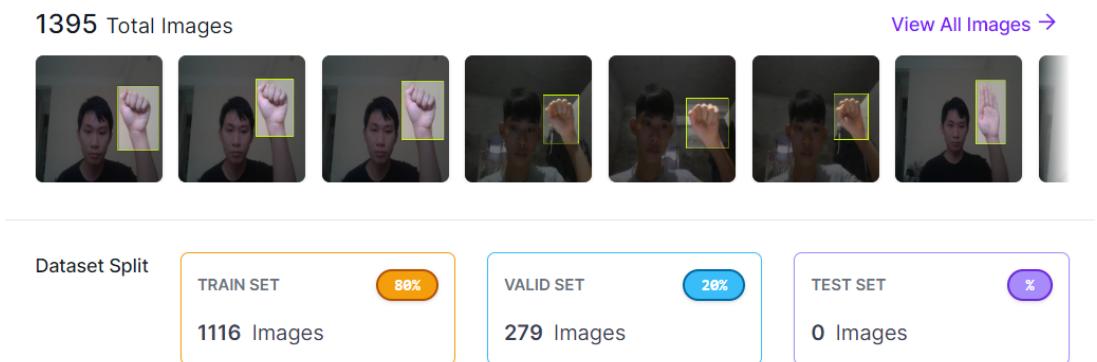
Dữ liệu được thu thập trực tiếp từ webcam, sử dụng thư viện cv2 để đọc, ghi ảnh, kết hợp với thư viện time để kiểm soát số lượng ảnh lấy tại mỗi thời điểm, tránh các ảnh trùng lặp quá nhiều. Ảnh ban đầu được lấy trực tiếp từ webcam có kích thước là (240, 240, 3). Số lượng ảnh bổ sung là 9600 ảnh, mỗi class có khoảng từ 350 đến 500 ảnh. Ảnh trước khi được thêm vào tập train của bộ dữ liệu mnist sẽ được xử lý qua các bước: chuyển ảnh từ BGR thành Gray, làm mờ bằng phương pháp Gaussian, thay đổi kích thước còn (28, 28), cuối cùng chuẩn hóa ảnh đảm bảo các giá trị độ sáng nằm trong giá trị từ 0 đến 255.

1.3 Dữ liệu sử dụng cho mô hình Yolo

Từ bộ dữ liệu MNIST và dữ liệu bổ sung được thực hiện vẽ các box bao quanh để phát hiện bàn tay ở vị trí nào. Khoanh vùng box cho bàn tay ở trang web roboflow 1395 ảnh rồi sau đó xuất tập dữ liệu dưới dạng yolov7, yolov8 rồi sau đó cho vào mô hình training.



Hình 1.1: Roboflow.



Hình 1.2: Chia tập dữ liệu.

Chương 2 Mô hình phát hiện bàn tay

2.1 Mô hình Yolov7

YOLO v7 là phiên bản thứ 7 của mô hình YOLO (You Only Look Once), một mô hình phát hiện đối tượng và nhận dạng đối tượng trong ảnh và video. YOLO v7 sử dụng mạng nơ-ron tích chập (CNN) để đồng thời dự đoán vị trí và phân loại các đối tượng trong ảnh một cách chính xác và nhanh chóng.

YOLO v7 có một số điểm nổi bật quan trọng. Nó cho phép phát hiện đối tượng và nhận dạng đối tượng theo thời gian thực, với tốc độ nhanh và hiệu suất cao. YOLO v7 sử dụng mạng nơ-ron tích chập sâu để học các đặc trưng của đối tượng, giúp nâng cao độ chính xác và độ phức tạp của nhiệm vụ. Hỗ trợ phát hiện và nhận dạng đa lớp đối tượng.

YOLO v7 cũng có độ phân giải cao hơn so với các phiên bản trước. Nó xử lý hình ảnh ở độ phân giải 608 x 608 pixel, cao hơn độ phân giải 416 x 416 được sử dụng trong YOLO v3. Độ phân giải cao hơn này cho phép YOLO v7 phát hiện các đối tượng nhỏ hơn và có độ chính xác tổng thể cao hơn.

Một trong những ưu điểm chính của YOLO v7 là tốc độ. Nó có thể xử lý hình ảnh với tốc độ 155 khung hình mỗi giây, nhanh hơn nhiều so với các thuật toán phát hiện đối tượng hiện đại khác. Ngay cả mô hình YOLO cơ bản ban đầu cũng có khả năng xử lý ở tốc độ tối đa 45 khung hình mỗi giây. Điều này làm cho nó phù hợp với các ứng dụng thời gian thực nhạy cảm như giám sát và ô tô tự lái, trong đó tốc độ xử lý cao hơn là rất quan trọng.

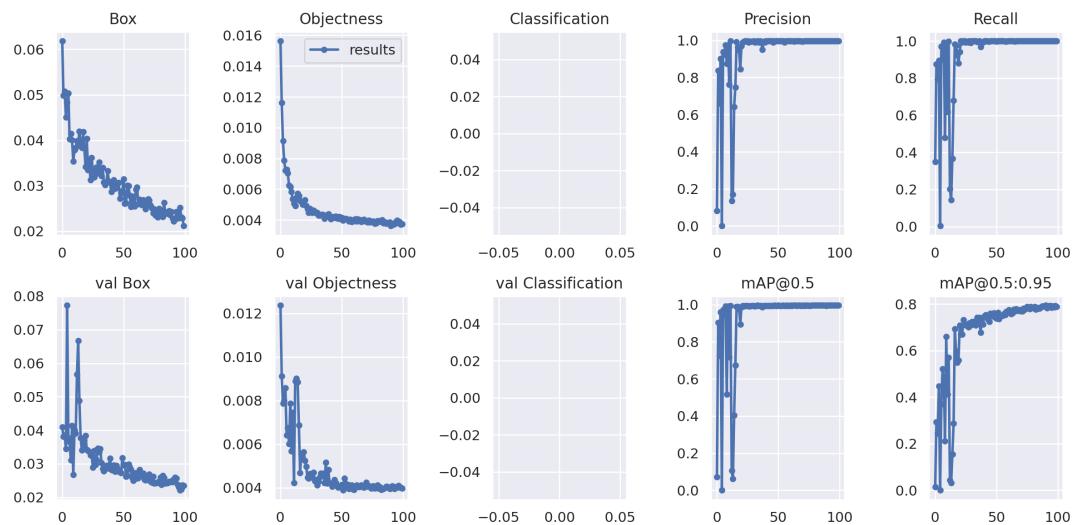
Về độ chính xác, YOLO v7 có thể so sánh với các thuật toán phát hiện đối tượng khác. Chúng tôi đạt được độ chính xác trung bình là 37,2% với IoU (Giao lộ khi hợp nhất) là 0,5 trên tập dữ liệu COCO được chia sẻ, có thể so sánh với các thuật toán phát hiện đối tượng tiên tiến nhất khác. Một so sánh hiệu suất định lượng được hiển thị dưới đây.

Tuy nhiên, cần lưu ý rằng YOLO v7 kém chính xác hơn so với các công cụ phát hiện hai giai đoạn như Faster R-CNN và Mask R-CNN, những công cụ này có xu hướng đạt

được độ chính xác trung bình cao hơn trên tập dữ liệu COCO nhưng cũng yêu cầu thời gian suy luận lâu hơn.

Chức năng:

- Phát hiện các đối tượng nhỏ hơn với độ chính xác cao hơn.
- Tăng cường hiệu suất.
- Hỗ trợ đa độ phân giải.



Hình 2.1: Kết quả mô hình yolov7

Box: Biểu đồ thể hiện hiệu suất của mô hình trong việc phát hiện các đối tượng ở các kích thước khác nhau.

Objectness: Biểu đồ thể hiện hiệu suất của mô hình nhận dạng đối tượng trong việc phát hiện các đối tượng tay trong ảnh. Biểu đồ này được tạo ra bằng cách sử dụng dữ liệu tập thử nghiệm của mô hình, bao gồm các ảnh có chứa các đối tượng tay ở các kích thước và vị trí khác nhau. Trục x là kích thước của đối tượng tay được biểu thị bằng tỷ lệ phần trăm của kích thước hình ảnh. Trục y là giá trị "Objectness", là một chỉ số cho biết mức độ chắc chắn của mô hình về việc có một đối tượng tay trong ảnh.

Precision: Biểu đồ thể hiện độ chính xác.

Recall: Biểu đồ thể hiện độ nhớ lại.

Val box: Biểu đồ này thể hiện giá trị của một hộp trong các khung thời gian khác nhau. Trục y thể hiện giá trị hộp, và trục x thể hiện số khung hình. Biểu đồ cho thấy rằng thuật toán có thể phát hiện đối tượng một cách khá chính xác trong khoảng thời gian từ 0 đến 50 khung hình. Tuy nhiên, độ chính xác của thuật toán giảm xuống trong khoảng thời gian từ 50 đến 100 khung hình.

Val objectness: Biểu đồ thể hiện giá trị trung bình của "val Objectness" trong một tập dữ liệu. Cho thấy rằng thuật toán đang hoạt động tốt trong việc phát hiện các đối tượng có độ "objectness" thấp.

mAP@0.5: Là một chỉ số được sử dụng để đánh giá hiệu suất của mô hình nhận dạng đối tượng. Chỉ số này được tính bằng cách lấy trung bình của các giá trị Average Precision (AP) ở các ngưỡng phân loại khác nhau. Ngưỡng phân loại được đặt thành 0.5. Có nghĩa là khi một mẫu được phân loại với độ tin cậy cao hơn hoặc bằng 0.5, thì mẫu đó được coi là được nhận dạng đúng.

mAP@0.5:0.95: Là một chỉ số được sử dụng để đánh giá hiệu suất của mô hình nhận dạng đối tượng. Chỉ số này được tính bằng cách lấy trung bình của các giá trị Average Precision (AP) ở các ngưỡng phân loại từ 0.5 đến 0.95, với bước nhảy là 0.05. Ngưỡng phân loại được đặt từ 0.5 đến 0.95, với bước nhảy là 0.05. Điều này có nghĩa là khi một mẫu được phân loại với độ tin cậy cao hơn hoặc bằng 0.5, nhưng nhỏ hơn hoặc bằng 0.55, thì mẫu đó được coi là được nhận dạng đúng ở ngưỡng 0.55. Tương tự, khi một mẫu được phân loại với độ tin cậy cao hơn hoặc bằng 0.55, nhưng nhỏ hơn hoặc bằng 0.60, thì mẫu đó được coi là được nhận dạng đúng ở ngưỡng 0.60.

2.2 Mô hình Yolov8

YOLOv8 là phiên bản YOLO mới nhất của Ultralytics. Là mô hình tiên tiến, hiện đại (SOTA), YOLOv8 được xây dựng dựa trên sự thành công của các phiên bản trước, giới thiệu các tính năng và cải tiến mới nhằm nâng cao hiệu suất, tính linh hoạt và hiệu quả. YOLOv8 hỗ trợ đầy đủ các nhiệm vụ AI về thị giác, bao gồm phát hiện, phân đoạn, ước

tính tư thế , theo dõi và phân loại . Tính linh hoạt này cho phép người dùng tận dụng khả năng của YOLOv8 trên nhiều ứng dụng và miền khác nhau.

2.2.1 Modes

Giới thiệu

Ultralytics YOLOv8 không chỉ là một mô hình phát hiện đối tượng khác; đó là một khung linh hoạt được thiết kế để bao trùm toàn bộ vòng đời của các mô hình học máy—từ nhập dữ liệu và đào tạo mô hình đến xác thực, triển khai và theo dõi trong thế giới thực. Mỗi chế độ phục vụ một mục đích cụ thể và được thiết kế để mang lại cho bạn sự linh hoạt và hiệu quả cần thiết cho các nhiệm vụ và trường hợp sử dụng khác nhau.

Sơ lược về các chế độ

Hiểu các chế độ khác nhau mà Ultralytics YOLOv8 hỗ trợ là rất quan trọng để tận dụng tối đa các mô hình của bạn:

- **Train mode:** Tinh chỉnh mô hình của bạn trên các tập dữ liệu tùy chỉnh hoặc được tải sẵn.
- **Val mode:** Điểm kiểm tra sau đào tạo để xác thực hiệu suất của mô hình.
- **Predict mode:** Điểm kiểm tra sau đào tạo để xác thực hiệu suất của mô hình.
- **Export mode:** Giải phóng sức mạnh dự đoán của mô hình của bạn trên dữ liệu trong thế giới thực.
- **Track mode:** Làm cho mô hình của bạn sẵn sàng triển khai ở nhiều định dạng khác nhau.
- **Benchmark mode:** Phân tích tốc độ và độ chính xác của mô hình của bạn trong các môi trường triển khai đa dạng.

Train

Chế độ đào tạo được sử dụng để đào tạo mô hình YOLOv8 trên tập dữ liệu tùy chỉnh. Ở chế độ này, mô hình được huấn luyện bằng cách sử dụng tập dữ liệu và siêu tham số

đã chỉ định. Quá trình huấn luyện bao gồm việc tối ưu hóa các tham số của mô hình để có thể dự đoán chính xác các lớp và vị trí của các đối tượng trong ảnh.

Val

Chế độ Val được sử dụng để xác thực mô hình YOLOv8 sau khi nó được đào tạo. Ở chế độ này, mô hình được đánh giá trên một bộ xác thực để đo lường độ chính xác và hiệu suất tổng quát của nó. Chế độ này có thể được sử dụng để điều chỉnh các siêu tham số của mô hình nhằm cải thiện hiệu suất của nó.

Predict

Chế độ dự đoán được sử dụng để đưa ra dự đoán bằng mô hình YOLOv8 đã được huấn luyện trên hình ảnh hoặc video mới. Ở chế độ này, mô hình được tải từ tệp điểm kiểm tra và người dùng có thể cung cấp hình ảnh hoặc video để thực hiện suy luận. Mô hình dự đoán các lớp và vị trí của các đối tượng trong hình ảnh hoặc video đầu vào.

Export

Chế độ xuất được sử dụng để xuất mô hình YOLOv8 sang định dạng có thể được sử dụng để triển khai. Ở chế độ này, mô hình được chuyển đổi sang định dạng có thể được sử dụng bởi các ứng dụng phần mềm hoặc thiết bị phần cứng khác. Chế độ này rất hữu ích khi triển khai mô hình vào môi trường sản xuất.

Track

Chế độ theo dõi được sử dụng để theo dõi các đối tượng trong thời gian thực bằng mô hình YOLOv8. Ở chế độ này, mô hình được tải từ tệp điểm kiểm tra và người dùng có thể cung cấp luồng video trực tiếp để thực hiện theo dõi đối tượng theo thời gian thực. Chế độ này rất hữu ích cho các ứng dụng như hệ thống giám sát hoặc xe tự lái.

Benchmark

Chế độ điểm chuẩn được sử dụng để lập hồ sơ về tốc độ và độ chính xác của các định dạng xuất khác nhau cho YOLOv8. Điểm chuẩn cung cấp thông tin về kích thước của định dạng được xuất, mAP50-95 số liệu của nó (để phát hiện, phân đoạn và tư thế đối tượng) hoặc accuracy top5 số liệu (để phân loại) và thời gian suy luận tính bằng mili giây trên mỗi hình ảnh trên các định dạng xuất khác nhau như ONNX, OpenVINO, TensorRT

và các định dạng khác . Thông tin này có thể giúp người dùng chọn định dạng xuất tối ưu cho trường hợp sử dụng cụ thể của họ dựa trên yêu cầu về tốc độ và độ chính xác của họ.

2.2.2 Task

Detection

Phát hiện là nhiệm vụ chính được YOLOv8 hỗ trợ. Nó liên quan đến việc phát hiện các đối tượng trong khung hình ảnh hoặc video và vẽ các hộp giới hạn xung quanh chúng. Các đối tượng được phát hiện được phân loại thành các loại khác nhau dựa trên tính năng của chúng. YOLOv8 có thể phát hiện nhiều đối tượng trong một khung hình ảnh hoặc video với độ chính xác và tốc độ cao.

Segmentation

Phân đoạn là một nhiệm vụ liên quan đến việc phân chia hình ảnh thành các vùng khác nhau dựa trên nội dung của hình ảnh. Mỗi khu vực được gán một nhãn dựa trên nội dung của nó. Nhiệm vụ này rất hữu ích trong các ứng dụng như phân đoạn hình ảnh và tạo ảnh y tế. YOLOv8 sử dụng một biến thể của kiến trúc U-Net để thực hiện phân đoạn.

Classification

Phân loại là một nhiệm vụ liên quan đến việc phân loại hình ảnh thành các loại khác nhau. YOLOv8 có thể được sử dụng để phân loại hình ảnh dựa trên nội dung của chúng. Nó sử dụng một biến thể của kiến trúc EfficiencyNet để thực hiện phân loại.

Pose

Phát hiện tư thế/điểm chính là một nhiệm vụ liên quan đến việc phát hiện các điểm cụ thể trong khung hình ảnh hoặc video. Những điểm này được gọi là điểm chính và được sử dụng để theo dõi chuyển động hoặc ước tính tư thế. YOLOv8 có thể phát hiện các điểm chính trong khung hình ảnh hoặc video với độ chính xác và tốc độ cao.

2.2.3 Models YOLOv8

Tổng quan

YOLOv8 là phiên bản mới nhất trong dòng máy dò tìm vật thể thời gian thực YOLO, mang lại hiệu suất vượt trội về độ chính xác và tốc độ. Dựa trên những tiến bộ của các phiên bản YOLO trước đó, YOLOv8 giới thiệu các tính năng và tối ưu hóa mới khiến nó trở thành lựa chọn lý tưởng cho các tác vụ phát hiện đối tượng khác nhau trong nhiều ứng dụng.

Các tính năng chính

- **Advanced Backbone and Neck Architectures:** YOLOv8 sử dụng kiến trúc xương sống và cổ tiên tiến, giúp cải thiện hiệu suất trích xuất tính năng và phát hiện đối tượng.
- **Anchor-free Split Ultralytics Head:** YOLOv8 sử dụng đầu Ultralytics phân chia không có neo, góp phần mang lại độ chính xác cao hơn và quy trình phát hiện hiệu quả hơn so với các phương pháp dựa trên neo.
- **Optimized Accuracy-Speed Tradeoff:** Với trọng tâm là duy trì sự cân bằng tối ưu giữa độ chính xác và tốc độ, YOLOv8 phù hợp cho các tác vụ phát hiện đối tượng theo thời gian thực trong các lĩnh vực ứng dụng đa dạng.
- **Variety of Pre-trained Models:** YOLOv8 cung cấp một loạt các mô hình được đào tạo trước để đáp ứng các nhiệm vụ và yêu cầu hiệu suất khác nhau, giúp bạn dễ dàng tìm thấy mô hình phù hợp cho trường hợp sử dụng cụ thể của mình.

Nhiệm vụ và chế độ được hỗ trợ

Dòng YOLOv8 cung cấp nhiều mẫu mã đa dạng, mỗi mẫu chuyên dụng cho các tác vụ cụ thể trong thị giác máy tính. Các mô hình này được thiết kế để đáp ứng nhiều yêu cầu khác nhau, từ phát hiện đối tượng đến các tác vụ phức tạp hơn như phân đoạn phiên bản, phát hiện tư thế/điểm chính và phân loại.

Mỗi biến thể của dòng YOLOv8 đều được tối ưu hóa cho nhiệm vụ tương ứng, đảm bảo hiệu suất và độ chính xác cao. Ngoài ra, các mô hình này tương thích với các chế độ

hoạt động khác nhau bao gồm Inference, Validation, Training, and Export, tạo điều kiện thuận lợi cho việc sử dụng chúng trong các giai đoạn triển khai và phát triển khác nhau.

Model	Filenames	Task	Inference	Validation	Training	Export
YOLOv8	yolov8n.pt yolov8s.pt yolov8m.pt yolov8l.pt yolov8x.pt	Detection	✓	✓	✓	✓
YOLOv8-seg	yolov8n-seg.pt yolov8s-seg.pt yolov8m-seg.pt yolov8l-seg.pt yolov8x-seg.pt	Instance Segmentation	✓	✓	✓	✓
YOLOv8-pose	yolov8n-pose.pt yolov8s-pose.pt yolov8m-pose.pt yolov8l-pose.pt yolov8x-pose.pt yolov8x-pose-p6.pt	Pose/Keypoints	✓	✓	✓	✓
YOLOv8-cls	yolov8n-cls.pt yolov8s-cls.pt yolov8m-cls.pt yolov8l-cls.pt yolov8x-cls.pt	Classification	✓	✓	✓	✓

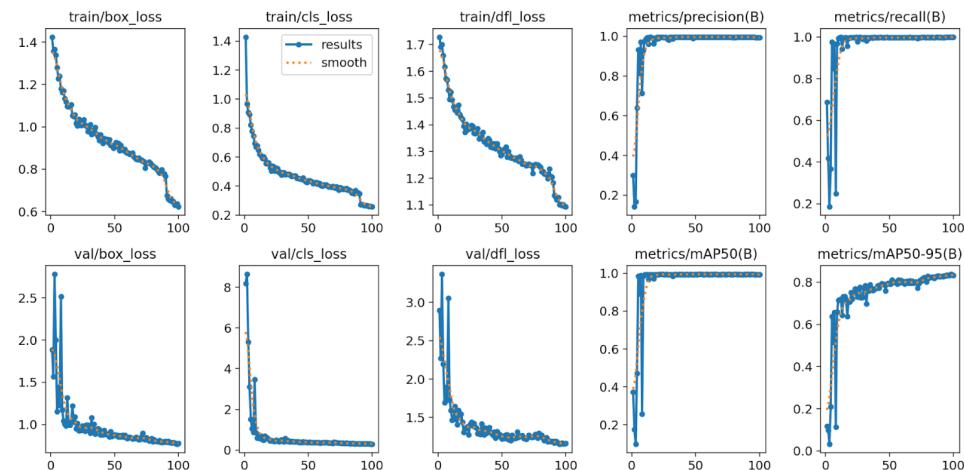
Hình 2.2: Tổng quan về các biến thể mô hình YOLOv8

Số liệu hiệu suất

Model	size (pixels)	mAP _{val} 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

Hình 2.3: Detection (COCO)

Kết quả của mô hình YOLOv8



Hình 2.4: Results YOLOv8

- **box loss**

Một thước đo tổn thất, dựa trên một hàm tổn thất cụ thể, đo lường mức độ "chặt chẽ" của các hộp giới hạn được dự đoán đối với các đối tượng thực tế cơ bản (nhãn trên hình ảnh của tập dữ liệu của bạn).

Giá trị thấp hơn cho biết mô hình của bạn đang được cải thiện để khai quát hóa và tạo các hộp giới hạn tốt hơn xung quanh các đối tượng mà tập dữ liệu đã được gắn nhãn để xác định.

- **cls loss**

Một thước đo tổn thất, dựa trên một hàm tổn thất cụ thể, đo lường tính chính xác của việc phân loại tất cả các hộp giới hạn được dự đoán. Mỗi hộp giới hạn riêng lẻ có thể chứa một lớp đối tượng hoặc một “nhãn nền” (hình ảnh rỗng).

- **mAP**

Để tính toán mAP để phát hiện đối tượng, bạn tính toán độ chính xác trung bình cho từng lớp trong dữ liệu của mình dựa trên các dự đoán của mô hình. Độ chính xác trung bình có liên quan đến vùng dưới đường cong thu hồi độ chính xác hoặc AUC cho một lớp nhất định trong tập dữ liệu của bạn. Giá trị trung bình của độ chính xác trung bình này đối với từng lớp riêng lẻ mang lại cho bạn Độ chính xác trung bình hoặc mAP trung bình.

Lưu ý: mAP bị ảnh hưởng bởi Giao lộ trên Liên minh hoặc IoU của nhãn sự thật mặt đất và các hộp giới hạn được dự đoán.

- **Giao điểm trên Liên minh (IoU)**

Được đo bằng số lượng hộp giới hạn được dự đoán trùng với hộp giới hạn thực tế trên mặt đất, chia cho tổng diện tích của cả hai hộp giới hạn.

- **mAP50(B)**

Được biết đến nhiều hơn với cái tên "Độ chính xác trung bình trung bình với IoU là 0,50 hoặc 50 phần trăm".

Độ chính xác trung bình trung bình (mAP) với các dự đoán được đánh giá là “đối tượng được phát hiện” tại Giao điểm trên Union (IoU) lớn hơn 0,5 hoặc 50 phần trăm.

- **mAP50-90(B)**

Được biết đến nhiều hơn với cái tên "Độ chính xác trung bình trung bình với khoảng IoU từ 0,50 đến 0,95 hoặc 50 phần trăm đến 95 phần trăm".

Độ chính xác trung bình trung bình (mAP) với các dự đoán được đánh giá là “đối tượng được phát hiện” tại Giao điểm trên Union (IoU) lớn hơn 0,50 và nhỏ hơn hoặc bằng 0,95 (50 phần trăm - 95 phần trăm).

2.3 So sánh YOLOv7 và YOLOv8

YOLOv8 và YOLOv7 đều là phiên bản của hệ thống phát hiện đối tượng YOLO (You Only Look Once) phổ biến . Trong bài viết này, chúng tôi sẽ so sánh các tính năng và cải tiến của YOLOv8 với YOLOv7 để hiểu những tiến bộ trong việc phát hiện đối tượng và xử lý hình ảnh theo thời gian thực .

So sánh song song giữa YOLOv8 và YOLOv7, cho thấy sự khác biệt về độ chính xác và tốc độ phát hiện đối tượng. Sử dụng màu sắc tương phản để làm nổi bật ưu và nhược điểm của từng phiên bản.

2.3.1 Điểm mấu chốt

- YOLOv8 và YOLOv7 là phiên bản của hệ thống phát hiện đối tượng YOLO.
- YOLOv8 cung cấp một số cải tiến và tính năng chính so với YOLOv7.
- Một số cải tiến trong YOLOv8 bao gồm tốc độ phát hiện nhanh hơn và độ chính xác được cải thiện trong việc phát hiện các vật thể nhỏ.
- YOLOv8 có kiến trúc không có neo, dự đoán đa quy mô và mạng đường trực được cải tiến.
- Các bài kiểm tra hiệu suất đã chỉ ra rằng YOLOv8 vượt trội hơn YOLOv7 về tốc độ và độ chính xác .

2.3.2 So sánh hiệu suất YOLOv8 và YOLOv7

Tính năng	YOLOv7	YOLOv8
Độ chính xác	Cao	Cao hơn
Tốc độ	Nhanh	Nhanh hơn
Kích thước mô hình	Lớn	Nhỏ hơn
Độ phức tạp	Cao	Thấp hơn
Khả năng phát hiện đối tượng nhỏ	Tốt	Tốt hơn
Khả năng phát hiện đối tượng trong các cảnh đông đúc	Tốt	Tốt hơn
Khả năng phát hiện đối tượng ở các tỷ lệ khác nhau	Tốt	Tốt hơn
Khả năng phát hiện đối tượng dưới sự thay đổi về ánh sáng	Tốt	Tốt hơn

Hình 2.5: Bảng so sánh.

Khi đánh giá hiệu suất của YOLOv8 và YOLOv7, một số yếu tố quan trọng sẽ được phát huy, chẳng hạn như tốc độ, độ chính xác, độ chính xác trung bình trung bình (MAP) và kiến trúc mô hình. Các thử nghiệm hiệu suất đã chứng minh rằng YOLOv8 vượt trội hơn YOLOv7 trong các lĩnh vực chính này, khiến nó trở thành lựa chọn ưu tiên để phát hiện đối tượng theo thời gian thực.

Speed

Một trong những cải tiến đáng kể của YOLOv8 so với YOLOv7 là tốc độ khung hình trên giây (FPS) nhanh hơn. Kiến trúc mô hình nâng cao và các thuật toán được tối ưu hóa góp phần phát hiện đối tượng nhanh hơn và hiệu quả hơn. Lợi thế về tốc độ này cho phép YOLOv8 xử lý số lượng khung hình cao hơn trên mỗi đơn vị thời gian, khiến nó rất phù hợp cho các ứng dụng thời gian thực.

Accuracy and MAP

YOLOv8 cũng vượt trội hơn YOLOv7 về độ chính xác. Kiến trúc mô hình nâng cao được triển khai trong YOLOv8 cho phép phát hiện đối tượng chính xác và đáng tin cậy hơn. Ngoài ra, điểm chính xác trung bình (MAP) của YOLOv8 đã cho thấy sự cải thiện so với YOLOv7.

Độ chính xác và độ chính xác rất quan trọng trong các hệ thống phát hiện đối tượng vì chúng xác định khả năng của mô hình trong việc xác định và định vị chính xác các đối tượng trong hình ảnh hoặc video. Hiệu suất vượt trội của YOLOv8 trong các khía cạnh này đảm bảo kết quả chính xác và đáng tin cậy hơn cho các ứng dụng trong thế giới thực khác nhau.

Model Architectu

Kiến trúc mô hình của YOLOv8 đã trải qua những cải tiến để cải thiện khả năng phát hiện đối tượng. Kiến trúc YOLOv8 có thiết kế không có neo, giúp loại bỏ nhu cầu về các hộp neo được xác định thủ công. Lựa chọn thiết kế này giúp đơn giản hóa quá trình đào tạo và nâng cao khả năng của mô hình trong việc phát hiện các đối tượng có kích thước và tỷ lệ khung hình khác nhau một cách hiệu quả.

Hơn nữa, YOLOv8 kết hợp dự đoán đa quy mô, cho phép mô hình đưa ra dự đoán ở nhiều độ phân giải. Cách tiếp cận đa tỷ lệ này giúp nâng cao khả năng của mô hình trong việc chụp các đối tượng ở nhiều tỷ lệ khác nhau, dẫn đến độ chính xác phát hiện đối tượng được cải thiện.

Nhìn chung, kiến trúc mô hình tinh tế và những cải tiến về hiệu suất của YOLOv8 khiến nó trở thành lựa chọn ưu việt hơn YOLOv7 cho các ứng dụng phát hiện đối tượng theo thời gian thực

Tham số	YOLOv8	YOLOv7
Tốc độ	Nhanh hơn	Chậm hơn
Sự chính xác	Cải thiện	Thấp hơn
Độ trễ	Thấp	Cao

Hình 2.6: Tổng quan cấp cao về so sánh hiệu suất giữa YOLOv8 và YOLOv7 trên nền tảng nhúng.

Chương 3 Xây dựng mô hình phân loại ký hiệu tay

3.1 Xây dựng mô hình CNN

Mô hình bao gồm 3 lớp Convolutional, 3 lớp MaxPooling, 1 lớp Flatten, 2 lớp Dense và 2 lớp Dropout. Kích thước ảnh đầu vào là (28, 28, 1), hàm kích hoạt là relu và softmax ở lớp Dense cuối cùng.

- Lớp Convolutional:

- Convolution 1:
 - Số lượng filters: 32
 - Kích thước kernel: (3, 3)
 - Activation function: ReLU
 - Input shape: (28, 28, 1)
- Convolution 2:
 - Số lượng filters: 64
 - Kích thước kernel: (3, 3)
 - Activation function: ReLU
- Convolution 3:
 - Số lượng filters: 128
 - Kích thước kernel: (3, 3)
 - Activation function: ReLU

- Lớp MaxPooling: (2, 2)

- Tỉ lệ Dropout: 0.2

- Lớp Dense:

- Dense 1:
 - Số lượng neurons: 256
 - Activation function: ReLU
- Dense 2:
 - Số lượng neurons: 24
 - Activation function: softmax
- Tham số:
 - Tổng tham số 131864 parameters
 - Tham số trainable 131864 parameters

3.2 Kết quả và đánh giá

- Tham số đầu vào: learning_rate=0.001, batch_size=64, epochs=100

Kết quả

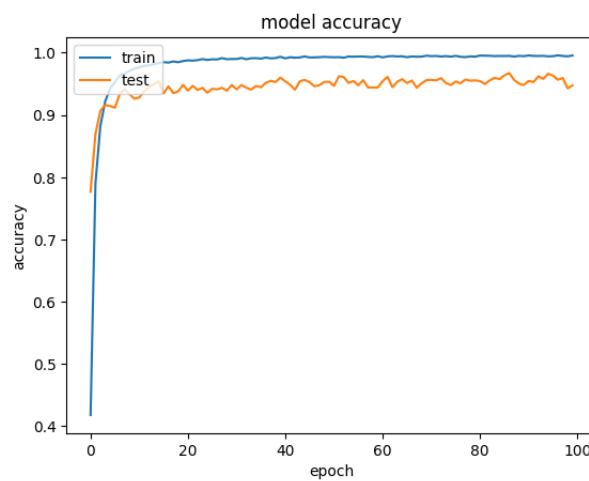
```
Epoch 98/100
533/533 [=====] - 3s 5ms/step - loss: 0.0173 - accuracy: 0.9945 - val_loss: 0.2474 - val_accuracy: 0.5483
Epoch 99/100
533/533 [=====] - 3s 7ms/step - loss: 0.0132 - accuracy: 0.9956 - val_loss: 0.2475 - val_accuracy: 0.9567
Epoch 100/100
533/533 [=====] - 3s 5ms/step - loss: 0.0184 - accuracy: 0.9949 - val_loss: 0.2322 - val_accuracy: 0.6529
Epoch 97/100
533/533 [=====] - 3s 5ms/step - loss: 0.0174 - accuracy: 0.9948 - val_loss: 0.2152 - val_accuracy: 0.9623
Epoch 98/100
533/533 [=====] - 3s 5ms/step - loss: 0.0165 - accuracy: 0.9950 - val_loss: 0.2060 - val_accuracy: 0.9580
Epoch 99/100
533/533 [=====] - 3s 6ms/step - loss: 0.0191 - accuracy: 0.9943 - val_loss: 0.3474 - val_accuracy: 0.5663
Epoch 100/100
533/533 [=====] - 3s 6ms/step - loss: 0.0193 - accuracy: 0.9945 - val_loss: 0.1866 - val_accuracy: 0.9632
Epoch 96/100
533/533 [=====] - 3s 5ms/step - loss: 0.0142 - accuracy: 0.9958 - val_loss: 0.1542 - val_accuracy: 0.9566
Epoch 97/100
533/533 [=====] - 3s 5ms/step - loss: 0.0171 - accuracy: 0.9948 - val_loss: 0.1699 - val_accuracy: 0.9594
Epoch 98/100
533/533 [=====] - 3s 5ms/step - loss: 0.0208 - accuracy: 0.9943 - val_loss: 0.3134 - val_accuracy: 0.9428
Epoch 99/100
533/533 [=====] - 4s 7ms/step - loss: 0.0156 - accuracy: 0.9950 - val_loss: 0.2654 - val_accuracy: 0.9476
```

Hình 3.1: Kết quả của 10 epochs cuối

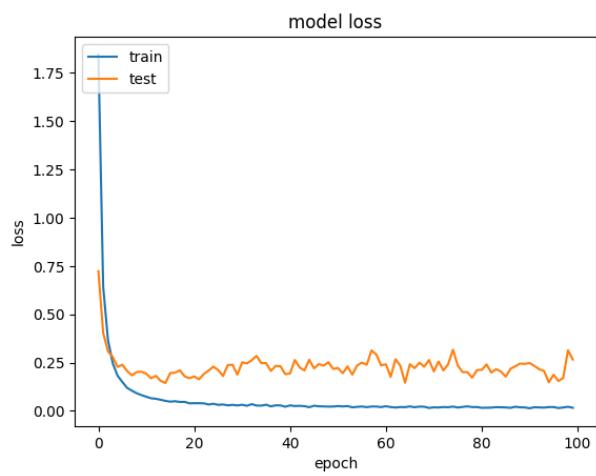
- Thông số đầu ra

- Số batch xử lý trong mỗi epoch: 533
- Thời gian xử lý một epoch: 3-4s
- Thời gian trung bình mỗi bước: 5-7ms/step
- Giá trị loss function trên tập dữ liệu huấn luyện ở epoch cuối cùng: 0.0156
- Độ chính xác của mô hình trên tập dữ liệu huấn luyện ở epoch cuối cùng: 99,56%

Mô hình CNN



Hình 3.2: Đồ thị accuracy giữa tệp train và test



Hình 3.3: Đồ thị loss giữa tập train và test

Từ hai đồ thị, có thể thấy kết quả đạt được rất tốt, với accuracy xấp xỉ bằng 1, loss tiến về 0.

Chương 4 Kết hợp kết quả của mô hình Yolov8 và CNN

4.1 Kết hợp hai mô hình

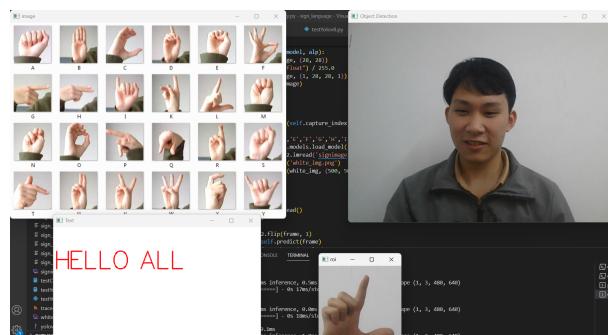
Để sử dụng kết hợp kết quả của mô hình Yolov8 và CNN, cần thông qua các bước sau:

- Bước 1: Detect bàn tay. Sau khi quá trình training của mô hình Yolov8 hoàn thành, sẽ thu được hai trọng số là weight.pt và best.pt, được sử dụng để detect bàn tay. Kết quả thu được chứa một số thông tin như tọa độ boxes, confidences(độ tin cậy), tên class...
- Bước 2: Từ kết quả của bước 1, trích xuất tọa độ boxes xyxy. Tọa độ này được lưu dưới định dạng numpy_array.
- Bước 3: Load mô hình CNN dưới định dạng h5
- Bước 4: Bật webcam hoặc camera, sau đó kiểm tra có bàn tay trong frame hay không, nếu có chuyển lên bước 5, nếu không chuyển lên bước 9.
- Bước 5: Chuyển xyxy thành tọa độ (x, y) và (x_max, y_max), sau đó trích xuất ROI từ tọa độ trên.
- Bước 6: Xử lý ảnh ROI. Ảnh ROI ban đầu sẽ cần lật ảnh, sau đó ảnh được chuyển thành ảnh gray, làm mờ bằng GaussianBlur, thay đổi kích thước về (28, 28) và chuẩn hóa giá trị độ sáng nằm trong [0, 1].
- Bước 7: Sử dụng ảnh ROI đã xử lý để tiến hành phân loại. Output của mô hình phân loại sẽ là 1 trong 24 chữ cái.
- Bước 8: Viết chữ và vẽ hình chữ nhật quanh bàn tay lên frame ban đầu.
- Bước 9: Hiển thị frame và ROI. Sau đó tiếp tục vòng lặp cho đến khi nhấn phím 'q'.

4.2 Ghép các ký tự thành từ, câu

Kết quả của việc kết hợp hai mô hình Yolov8 và CNN là ký tự được phân loại. Để tạo nên từ hoặc câu từ các ký tự được phân loại cần thông qua quá trình dưới đây:

- Phân loại ký tự: Kết quả phân loại được lưu lại và ký tự chỉ được xem xét khi nó xuất hiện liên tục trong khoảng thời gian 4 giây.
- Ghi ký tự lên ảnh trắng:
 - Sau mỗi chu kỳ 4 giây, ký tự được ghi lên ảnh trắng với vị trí xác định từ trước.
 - Vị trí của ký tự vừa viết được lưu lại để đảm bảo ký tự sau không bị đè lên ký tự trước.
- Quản lý vị trí và khoảng cách giữa các từ:
 - Vị trí của mỗi ký tự đã được ghi lên ảnh trắng được lưu lại để đảm bảo việc đặt ký tự tiếp theo không gây xung đột.
 - Khoảng cách giữa các từ cũng được xử lý dựa trên vị trí của ký tự để tạo thành một câu.



Hình 4.1: Ví dụ viết một câu từ các ký tự

Chương 5 KẾT LUẬN

5.1 Tóm tắt nội dung, kết quả đạt được và một số hạn chế cần khắc phục

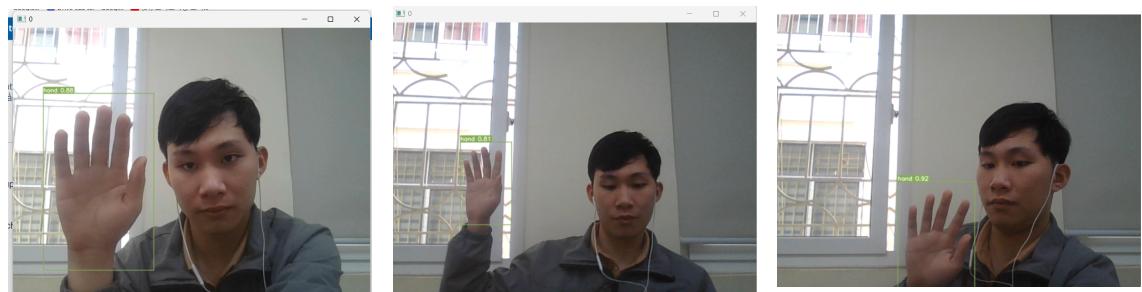
Qua nghiên cứu, tìm hiểu và thực hành, nhóm đã ứng dụng được hai mô hình object detection mạnh mẽ là Yolov7 và Yolov8, cũng như biết cách kết hợp với mô hình CNN. Ngoài ra, cũng ứng dụng một số kiến thức đã học như sử dụng thư viện cv2 để tiền xử lý ảnh và thu thập data.

Kết quả 3 mô hình khi chạy riêng lẻ cho kết quả nhận diện và phân loại đạt hiệu suất cao, ổn định. Tuy nhiên, trong một số điều kiện môi trường không tốt như thiếu sáng hay màu nền phức tạp, thì hiệu suất và tính ổn định cũng giảm đáng kể. Nguyên nhân là do bộ dữ liệu train được lấy trong điều kiện tốt, chưa có đủ lượng nhiễu tương ứng với môi trường thực tế.

Kết quả của việc kết hợp yolov8 và CNN cũng chưa thực sự như mong muốn, còn nhiều kí tự chưa phân loại đúng. Lý do cho kết quả không tốt là bộ dữ liệu train có kích thước ảnh vuông (28, 28), nhưng kết quả detect của yolov8 có kích thước ảnh đa dạng, tỉ lệ giữa width và height chênh lệch lớn, nên khi thay đổi về kích thước (28, 28), giá trị các pixel bị sai lệch, dẫn đến kết quả không tốt.

Dưới đây là một số kết quả nhóm đạt được, minh họa thông qua hình ảnh kiểm tra thực tế.

- Mô hình Yolov7



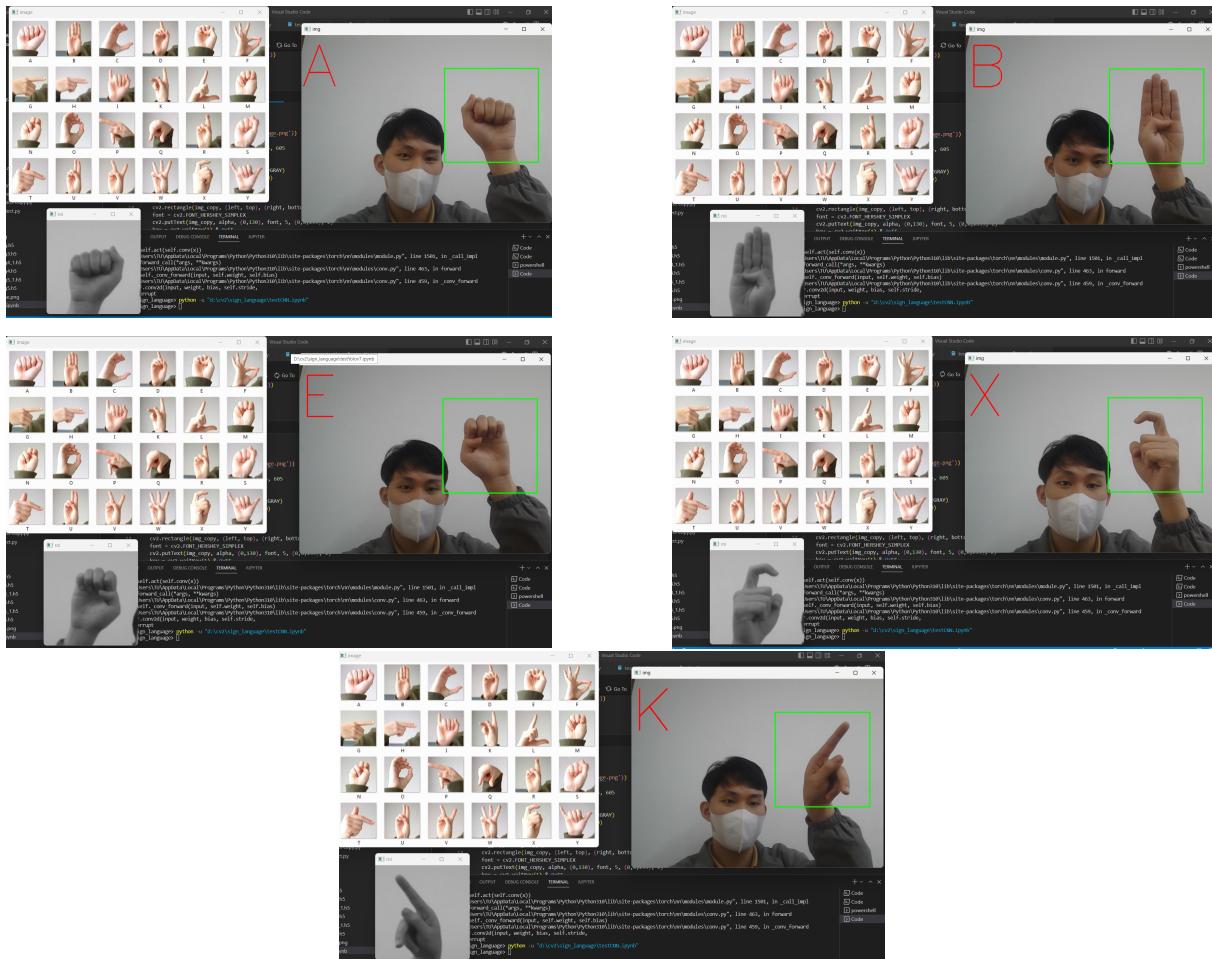
Hình 5.1: Kết quả nhận diện bàn tay của mô hình Yolov7

- Mô hình Yolov8



Hình 5.2: Kết quả nhận diện bàn tay của mô hình Yolov8

- Mô hình CNN



Hình 5.3: Kết quả phân loại thời gian thực của mô hình CNN

Dự án hiện tại đang phát triển ở những bước đầu, cần phải cải thiện, nâng cấp cũng như thời gian để có thể đưa gần với việc ứng dụng thực tế. Tuy nhiên, dự án cũng có những tiềm năng để có thể phát triển sau này.

5.2 Tiềm năng ứng dụng của đề tài

+ Giao tiếp và hỗ trợ cho người khiếm thính: Mô hình có thể được sử dụng để giúp người khiếm thính giao tiếp và tương tác với người khác. Việc chuyển đổi ngôn ngữ ký hiệu thành văn bản hoặc âm thanh có thể cung cấp một phương tiện truyền đạt thông tin hiệu quả và giúp người khiếm thính tham gia vào xã hội một cách đầy đủ hơn.

+ Giáo dục và đào tạo: Mô hình có thể được áp dụng trong lĩnh vực giáo dục và đào tạo để hỗ trợ việc học của người khiếm thính. Việc chuyển đổi ngôn ngữ ký hiệu thành

văn bản hoặc âm thanh có thể giúp người khiếm thính tiếp cận kiến thức và thông tin một cách dễ dàng và hiệu quả.

+ Tạo ra công cụ hỗ trợ truyền thông: Mô hình có thể được sử dụng để xây dựng các công cụ hỗ trợ truyền thông cho người khiếm thính. Các ứng dụng và công cụ này có thể cung cấp khả năng chuyển đổi giữa ngôn ngữ ký hiệu và văn bản/âm thanh, hỗ trợ giao tiếp hàng ngày, truyền đạt thông tin và truyền tải ý kiến.

+ Nghiên cứu ngôn ngữ ký hiệu: Mô hình có thể được sử dụng để nghiên cứu và phân tích ngôn ngữ ký hiệu. Việc áp dụng các phương pháp học máy và xử lý ngôn ngữ tự nhiên vào ngôn ngữ ký hiệu có thể giúp tìm ra các quy tắc và cấu trúc ngôn ngữ ký hiệu, nâng cao hiểu biết về ngôn ngữ này và đóng góp vào việc phát triển công cụ hỗ trợ ngôn ngữ ký hiệu.

+ Phát triển ứng dụng di động: Mô hình có thể được tích hợp vào các ứng dụng di động để cung cấp hỗ trợ truyền thông cho người khiếm thính. Việc sử dụng điện thoại di động hoặc tablet có thể giúp người khiếm thính giao tiếp và truy cập thông tin một cách dễ dàng và tiện lợi.