ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Helmet Protection Detection AI

Đặng Khánh Đạt
Trương Mậu Anh
Nguyễn Hải Đăng
Bùi Phương Nam

Mã học phần: MAT3508
Học kỳ 1, Năm học 2025-2026

Thông tin Dự án

[Thông tin này cũng cần được ghi trong README.md của kho GitHub.]

Học phần: MAT3508 – Nhập môn trí tuệ nhân tạo
Học kỳ: Học kỳ 1, Năm học 2025-2026
Trường: VNU-HUS (Đại học Quốc gia Hà Nội – Trường Đại học Khoa học Tự nhiên)
Tên dự án: Helmet Protection Detection AI
Ngày nộp: 30/11/2025
Báo cáo PDF: Liên kết tới báo cáo PDF trong kho GitHub
Slide thuyết trình: `https://github.com/datdang401205-jpg/Helmet-Protection-Detection/blob/main/slide%2`
Kho GitHub: `https://github.com/datdang401205-jpg/Helmet-Detection`

Thành viên nhóm

Họ tên	**Mã sinh viên**	**Đóng góp**
Đặng Khánh Đạt	23001514	Nhóm trưởng
Trương Mậu Anh	23001498	Thu thập dữ liệu
Nguyễn Hải Đăng	23001516	Xây dựng mô hình
Bùi Phương Nam	23001538	Xây dựng mô hình

Mục lục

Chương 1

Giới thiệu

1.1 Giới thiệu

1.1.1 Tóm tắt

Dự án **Hệ thống Phát hiện và Nhận diện Bảo hộ Mũ Bảo Hiểm (Helmet Protection Detection System)** được phát triển nhằm giải quyết vấn đề an toàn giao thông đô thị bằng cách tự động hóa quá trình giám sát việc tuân thủ quy định đội mũ bảo hiểm đối với người điều khiển xe máy. Mục tiêu chính là xây dựng một giải pháp thị giác máy tính (Computer Vision) có khả năng phát hiện chính xác các đối tượng liên quan trong môi trường giao thông phức tạp.

Cụ thể, hệ thống sử dụng kiến trúc học sâu (Deep Learning) mạnh mẽ, dựa trên mô hình **YOLOv8 (You Only Look Once phiên bản 8)**, để phân tích hình ảnh và video thời gian thực. Bộ dữ liệu huấn luyện được tùy chỉnh với bốn lớp đối tượng quan trọng:

1. `helmet`: Mũ bảo hiểm (đội đúng quy định).
2. `no helmet`: Không đội mũ bảo hiểm (vi phạm).
3. `rider`: Người điều khiển xe máy (để xác định chủ thể).
4. `number plate`: Biển số xe (để phục vụ việc truy xuất thông tin, nếu cần).

1.1.2 Bài toán đặt ra và Động lực

An toàn giao thông đường bộ là một trong những thách thức lớn tại các quốc gia có mật độ xe máy cao như Việt Nam. Mặc dù quy định bắt buộc đội mũ bảo hiểm đã được ban hành, tình trạng người tham gia giao thông phớt lờ quy định này vẫn còn phổ biến, đặc biệt tại các khu vực ít được giám sát. Việc không đội mũ bảo hiểm là nguyên nhân hàng đầu làm tăng mức độ nghiêm trọng của chấn thương đầu và tỷ lệ tử vong trong các vụ tai nạn xe máy.

Hiện tại, việc giám sát và xử phạt chủ yếu dựa vào lực lượng chức năng tại chỗ, một phương pháp tốn kém về thời gian, nhân lực và không thể đảm bảo sự giám sát liên tục 24/7 trên toàn bộ mạng lưới giao thông.

Động lực của dự án:

- **Tăng cường An toàn:** Giảm thiểu rủi ro tai nạn nghiêm trọng thông qua việc tăng cường tính răn đe.
- **Tự động hóa Giám sát:** Xây dựng một hệ thống thông minh có thể hoạt động độc lập, liên tục, và chính xác, loại bỏ các hạn chế của việc kiểm soát thủ công.
- **Hỗ trợ Cơ quan Chức năng:** Cung cấp công cụ hiệu quả tích hợp vào các hệ thống camera giám sát thông minh (Smart City) hiện có để trích xuất bằng chứng vi phạm một cách khách quan.

Chương 2

Phương pháp & Triển khai

2.1 Phương pháp

Trong nghiên cứu này, nhóm lựa chọn mô hình **YOLOv8** để xây dựng hệ thống phát hiện và phân loại đối tượng liên quan đến việc đội mũ bảo hiểm của người điều khiển xe máy. YOLOv8 là một mô hình phát hiện vật thể thuộc nhóm *one-stage detector*, nổi bật với khả năng xử lý nhanh, kiến trúc gọn nhẹ và độ chính xác cao, đặc biệt phù hợp với các bài toán yêu cầu xử lý thời gian thực như giám sát giao thông và hệ thống an toàn công cộng.

2.1.1 Lựa chọn dữ liệu

Dữ liệu huấn luyện được sử dụng trong nghiên cứu được thu thập từ bộ dữ liệu công khai **"Rider, With Helmet, Without Helmet, Number Plate"** trên nền tảng Kaggle, do tác giả *Penny.py* xây dựng. Bộ dữ liệu bao gồm các hình ảnh chụp người điều khiển xe máy trong nhiều bối cảnh khác nhau, với nhãn được gán thủ công cho các đối tượng liên quan đến bài toán.

Cụ thể, bộ dữ liệu được chia thành các tập như sau:

- **Tập huấn luyện (Training set):** 400 ảnh, dùng để huấn luyện trọng số của mô hình.
- **Tập xác thực (Validation set):** 100 ảnh, dùng để theo dõi quá trình hội tụ, đánh giá hiệu suất và điều chỉnh siêu tham số.
- **Số lớp (Classes):** 4 lớp, bao gồm `helmet`, `no helmet`, `rider` và `number plate`.

Nhóm chọn bốn lớp gồm `helmet`, `no helmet`, `rider` và `number plate` vì mục tiêu của hệ thống là phát hiện và xử lý vi phạm giao thông. Việc tách `helmet` và `no helmet` giúp mô hình học trực tiếp trạng thái tuân thủ và vi phạm, thay vì coi không đội mũ là background. Lớp `rider` được dùng để xác định chủ thể vi phạm và làm ngữ cảnh liên kết các đối tượng. Cuối cùng,`number plate` phục vụ cho việc truy xuất thông tin phương tiện, giúp hoàn chỉnh pipeline từ phát hiện vi phạm đến xử lý hậu kiểm.

Các lớp này được lựa chọn nhằm phản ánh đầy đủ các thành phần cần thiết trong một hệ thống giám sát vi phạm không đội mũ bảo hiểm, đồng thời tạo tiền đề cho việc mở rộng sang các bài toán nâng cao như nhận dạng biển số xe.

2.1.2 Tiền xử lý và gán nhãn dữ liệu

Trước khi đưa vào huấn luyện, toàn bộ dữ liệu được chuẩn hóa về kích thước ảnh đầu vào (ví dụ 640×640) để phù hợp với kiến trúc của YOLOv8. Các nhãn đối tượng được biểu diễn dưới dạng hộp giới hạn (*bounding box*) theo định dạng chuẩn của YOLO, bao gồm tọa độ tâm, chiều rộng, chiều cao và nhãn lớp tương ứng. Việc chuẩn hóa dữ liệu giúp mô hình học được các đặc trưng ổn định và giảm nhiễu trong quá trình huấn luyện.

2.1.3 Thiết lập huấn luyện mô hình

Quá trình huấn luyện mô hình YOLOv8 được thực hiện với các siêu tham số chính như sau:

- **Số epoch:** 100, nhằm đảm bảo mô hình có đủ số vòng lặp để học được các đặc trưng quan trọng từ dữ liệu.
- **Batch size:** 16, cân bằng giữa tốc độ huấn luyện và khả năng sử dụng bộ nhớ.
- **Optimizer:** Adam, giúp tối ưu quá trình cập nhật trọng số với khả năng thích nghi tốt và hội tụ nhanh.

- **Learning rate:** 0.01, được lựa chọn nhằm đảm bảo tốc độ học đủ nhanh trong giai đoạn đầu nhưng vẫn duy trì tính ổn định.

Trong quá trình huấn luyện, mô hình được đánh giá liên tục trên tập validation thông qua các chỉ số như Precision, Recall và mAP để theo dõi hiệu suất và phát hiện hiện tượng quá khớp (overfitting).

2.1.4 Quy trình tổng thể

Tóm lại, phương pháp nghiên cứu được thực hiện theo các bước chính: (1) thu thập và chuẩn hóa dữ liệu, (2) huấn luyện mô hình YOLOv8 với các siêu tham số phù hợp, (3) đánh giá hiệu suất mô hình trên tập validation, và (4) phân tích kết quả để rút ra các nhận xét và đề xuất cải tiến. Quy trình này đảm bảo tính khoa học, khả năng tái lập và phù hợp với mục tiêu xây dựng hệ thống phát hiện vi phạm đội mũ bảo hiểm trong thực tế.

2.2 Mô hình Phát hiện Đối tượng: YOLOv8

2.2.1 Giới thiệu về YOLOv8

YOLOv8 (You Only Look Once version 8) là phiên bản mới của họ mô hình YOLO do Ultralytics phát triển, được công bố năm 2023. Đây là một trong những mô hình phát hiện đối tượng (Object Detection) hiện đại, đạt hiệu năng cao và được ứng dụng rộng rãi trong các bài toán thị giác máy tính như nhận dạng vật thể, phân đoạn ảnh và theo dõi đối tượng trong thời gian thực.

Khác với các phương pháp phát hiện đối tượng truyền thống phải thực hiện nhiều bước xử lý, YOLOv8 áp dụng kiến trúc one-stage detector, cho phép mô hình xử lý toàn bộ ảnh chỉ trong một lần suy luận duy nhất. Nhờ đó, YOLOv8 có tốc độ xử lý nhanh, độ trễ thấp và phù hợp với các hệ thống yêu cầu thời gian thực như giám sát an ninh, xe tự hành và kiểm tra chất lượng sản phẩm.

YOLOv8 sử dụng kiến trúc mạng nơ-ron tích chập sâu (CNN) cải tiến với các thành phần chính gồm Backbone, Neck và Head. Đặc biệt, YOLOv8 áp dụng anchor-free detection, giúp đơn giản hóa quá trình huấn luyện, cải thiện khả năng hội tụ và tăng độ chính xác so với các phiên bản YOLO trước đó. Ngoài ra, mô hình còn hỗ trợ nhiều tác vụ khác nhau như Object Detection, Instance Segmentation, Pose Estimation và Classification.

Với thiết kế linh hoạt, YOLOv8 cung cấp nhiều kích thước mô hình khác nhau (từ YOLOv8n đến YOLOv8x), cho phép người dùng lựa chọn giữa tốc độ và độ chính xác tùy theo yêu cầu ứng dụng. Nhờ những ưu điểm trên, YOLOv8 đang trở thành một giải pháp hiệu quả và phổ biến trong các hệ thống thị giác máy tính hiện đại..

2.3 So sánh hiệu năng với các mô hình phát hiện đối tượng khác

2.3.1 Mục tiêu so sánh

Để đánh giá hiệu quả của mô hình YOLOv8 trong bài toán phát hiện mũ bảo hiểm, nhóm tiến hành so sánh hiệu năng của YOLOv8 với một số mô hình phát hiện đối tượng phổ biến khác. Việc so sánh nhằm làm rõ ưu điểm của YOLOv8 về độ chính xác, tốc độ suy luận và khả năng ứng dụng trong hệ thống giám sát giao thông thời gian thực.

2.3.2 Các mô hình được so sánh

Các mô hình được lựa chọn để so sánh bao gồm:

- **Faster R-CNN**: mô hình hai giai đoạn (two-stage), có độ chính xác cao nhưng tốc độ suy luận chậm.
- **SSD (Single Shot MultiBox Detector)**: mô hình một giai đoạn (one-stage), cân bằng giữa tốc độ và độ chính xác.
- **YOLOv5**: phiên bản YOLO trước đó, được sử dụng rộng rãi trong nhiều bài toán phát hiện đối tượng.
- **YOLOv8**: mô hình được nhóm lựa chọn để triển khai trong bài toán này.

2.3.3 Tiêu chí đánh giá

Hiệu năng của các mô hình được so sánh dựa trên các tiêu chí chính sau:

- **Độ chính xác (mAP)**: đánh giá khả năng phát hiện đúng đối tượng.

- **Tốc độ suy luận (FPS)**: số khung hình xử lý được trong một giây.
- **Độ phức tạp mô hình**: số tham số và mức độ tiêu tốn tài nguyên tính toán.
- **Khả năng ứng dụng thực tế**: mức độ phù hợp với hệ thống giám sát giao thông thời gian thực.

2.3.4 So sánh định tính

- Faster R-CNN cho độ chính xác cao nhưng tốc độ suy luận chậm, không phù hợp với các hệ thống yêu cầu xử lý thời gian thực.
- SSD có tốc độ suy luận nhanh hơn Faster R-CNN nhưng gặp hạn chế khi phát hiện các đối tượng nhỏ như mũ bảo hiểm.
- YOLOv5 cải thiện đáng kể tốc độ và độ chính xác so với SSD, tuy nhiên vẫn sử dụng anchor-based, làm tăng độ phức tạp khi huấn luyện.
- YOLOv8 sử dụng cơ chế anchor-free, giúp đơn giản hóa quá trình huấn luyện, cải thiện khả năng hội tụ và tăng độ chính xác trong phát hiện các đối tượng nhỏ.

2.3.5 So sánh định lượng

Bảng 2.1 trình bày so sánh tổng quan hiệu năng giữa các mô hình.

Bảng 2.1: So sánh hiệu năng giữa các mô hình phát hiện đối tượng

Mô hình	mAP	FPS	Kiến trúc	Ứng dụng realtime
Faster R-CNN	Cao	Thấp	Two-stage	Không phù hợp
SSD	Trung bình	Cao	One-stage	Tương đối
YOLOv5	Cao	Cao	Anchor-based	Phù hợp
YOLOv8	Cao	Rất cao	Anchor-free	Rất phù hợp

2.3.6 Nhận xét và lựa chọn mô hình

Từ kết quả so sánh, có thể nhận thấy YOLOv8 đạt được sự cân bằng tốt giữa độ chính xác và tốc độ suy luận. Đặc biệt, cơ chế anchor-free giúp mô hình phát hiện hiệu quả các đối tượng nhỏ như mũ bảo hiểm và biển số xe, đồng thời giảm độ phức tạp trong quá trình huấn luyện.

Do đó, YOLOv8 được lựa chọn là mô hình phù hợp nhất cho bài toán phát hiện mũ bảo hiểm trong hệ thống giám sát giao thông thời gian thực.

2.3.7 Cấu trúc Kiến trúc (Architecture)

Kiến trúc của YOLOv8 được thiết kế theo hướng tối ưu hóa đồng thời **độ chính xác**, **tốc độ suy luận** và **khả năng mở rộng**, phù hợp với các ứng dụng thị giác máy tính thời gian thực. Tương tự các phiên bản YOLO trước, YOLOv8 vẫn duy trì cấu trúc tổng thể gồm ba thành phần chính: **Backbone**, **Neck** và **Head**. Tuy nhiên, mỗi thành phần đều được cải tiến đáng kể nhằm nâng cao hiệu suất tổng thể của mô hình.

1. **Backbone (Xương sống):**

   Backbone đóng vai trò trích xuất các đặc trưng quan trọng từ ảnh đầu vào, chuyển đổi ảnh gốc thành các **bản đồ đặc trưng (feature maps)** ở nhiều mức trừu tượng khác nhau. YOLOv8 sử dụng các lớp tích chập (Convolutional layers) kết hợp với các khối kiến trúc cải tiến nhằm tăng khả năng biểu diễn đặc trưng trong khi vẫn giữ số lượng tham số ở mức hợp lý.

   Một điểm nổi bật trong Backbone của YOLOv8 là việc sử dụng khối **C2f (Cross-Stage Partial with two convolution layers)**. Khối C2f cho phép mô hình:

   - Tăng cường khả năng lan truyền gradient, giúp huấn luyện mạng sâu ổn định hơn.
   - Tái sử dụng đặc trưng hiệu quả giữa các tầng, giảm hiện tượng mất mát thông tin.
   - Giảm số lượng tham số và chi phí tính toán so với các khối CSP truyền thống.

   Nhờ đó, Backbone của YOLOv8 có khả năng học được các đặc trưng giàu ngữ nghĩa, từ các chi tiết mức thấp (cạnh, góc) đến các đặc trưng mức cao (hình dạng, cấu trúc đối tượng), làm cơ sở cho các bước xử lý tiếp theo.

2. **Neck (Cổ):**

   Neck có nhiệm vụ tổng hợp và kết nối các đặc trưng được trích xuất từ Backbone ở nhiều mức độ khác nhau. Trong bài toán phát hiện đối tượng, việc xử lý các đối tượng có kích thước khác nhau là một thách thức lớn; do đó, YOLOv8 sử dụng cơ chế tổng hợp đặc trưng đa tỉ lệ (*multi-scale feature fusion*).

   Cụ thể, YOLOv8 kết hợp hai kiến trúc phổ biến là:

   - **FPN (Feature Pyramid Network):** truyền thông tin ngữ nghĩa từ các tầng sâu (đặc trưng trừu tượng cao) xuống các tầng nông hơn.
   - **PAN (Path Aggregation Network):** truyền thông tin định vị từ các tầng nông lên các tầng sâu hơn.

   Sự kết hợp giữa FPN và PAN giúp mô hình:

   - Giữ được thông tin chi tiết về vị trí đối tượng.
   - Đồng thời tận dụng được ngữ nghĩa cấp cao của đối tượng.
   - Cải thiện khả năng phát hiện các đối tượng nhỏ, trung bình và lớn trong cùng một ảnh.

   Nhờ Neck, các đặc trưng đa tỉ lệ được hợp nhất một cách hiệu quả trước khi chuyển đến Head để thực hiện dự đoán.

3. **Head (Đầu):**

   Head là thành phần cuối cùng của YOLOv8, chịu trách nhiệm đưa ra các dự đoán về **vị trí**, **loại đối tượng** và **độ tin cậy**. YOLOv8 sử dụng kiến trúc **Decoupled Head**, trong đó các nhánh dự đoán được tách biệt rõ ràng:

   - Nhánh **Classification**: dự đoán nhãn lớp của đối tượng.
   - Nhánh **Localization**: dự đoán tọa độ hộp giới hạn (*bounding box*).

   Việc tách riêng hai nhiệm vụ này giúp giảm xung đột trong quá trình học, từ đó cải thiện độ chính xác của cả phân loại và định vị.

   Một cải tiến quan trọng khác của YOLOv8 là việc áp dụng cơ chế **Anchor-Free**. Thay vì sử dụng các *anchor boxes* được định nghĩa trước như trong các phiên bản YOLO cũ, YOLOv8 trực tiếp dự đoán tọa độ hộp giới hạn dựa trên tâm đối tượng. Cách tiếp cận này mang lại nhiều lợi ích:

   - Đơn giản hóa quá trình thiết kế và huấn luyện mô hình.
   - Giảm sự phụ thuộc vào việc tinh chỉnh anchor.
   - Tăng khả năng tổng quát hóa và cải thiện độ chính xác phát hiện.

Tổng thể, kiến trúc của YOLOv8 thể hiện sự cân bằng hiệu quả giữa độ chính xác và tốc độ, đồng thời phù hợp với nhiều bài toán thực tế trong lĩnh vực thị giác máy tính.

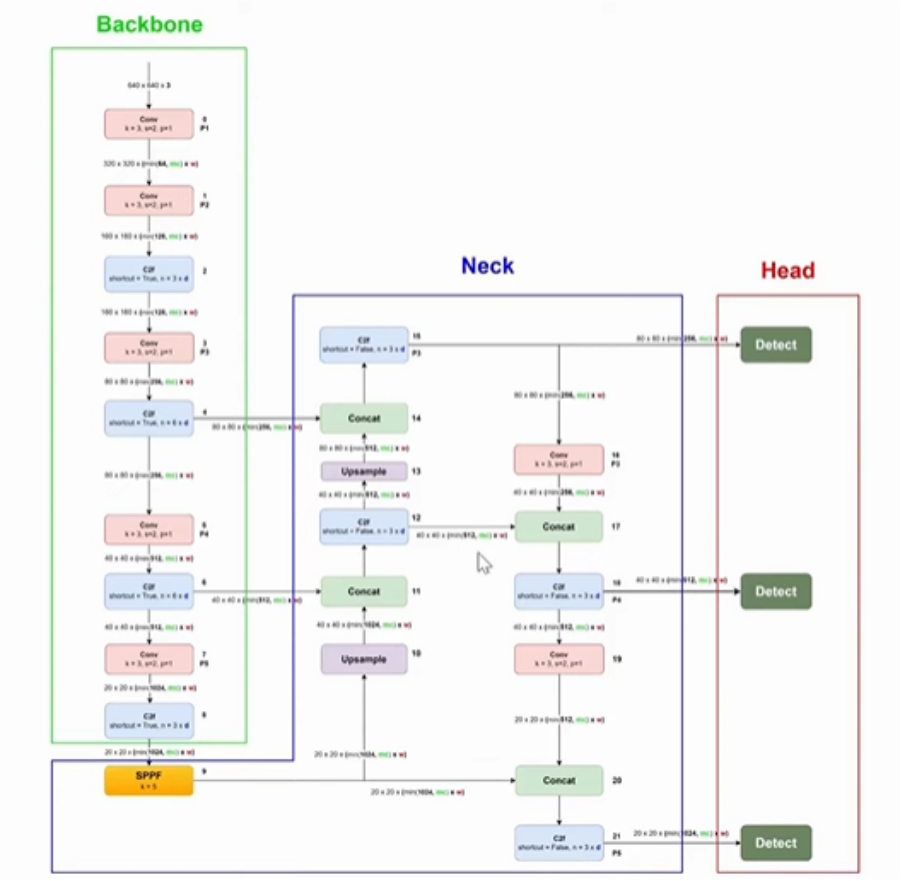
2.3.8 Cách hoạt động chính

YOLOv8 thuộc nhóm mô hình **One-Stage Detector**, trong đó quá trình phát hiện đối tượng và phân loại được thực hiện **đồng thời trong một lần suy luận duy nhất**. Khác với các mô hình hai giai đoạn (Two-Stage Detector) như Faster R-CNN, YOLOv8 không cần bước đề xuất vùng (Region Proposal) riêng biệt, từ đó giúp giảm đáng kể độ trễ và chi phí tính toán.

Quá trình hoạt động của YOLOv8 có thể được mô tả theo các bước chính sau:

- **Xử lý ảnh đầu vào:** Ảnh đầu vào được chuẩn hóa về kích thước cố định (ví dụ: 640×640) và đưa vào mạng nơ-ron tích chập. Các phép tiền xử lý như chuẩn hóa giá trị pixel và tăng cường dữ liệu (data augmentation) được áp dụng trong quá trình huấn luyện nhằm cải thiện khả năng tổng quát hóa của mô hình.

- **Trích xuất đặc trưng (Feature Extraction):** Ảnh sau khi tiền xử lý được đưa qua **Backbone** để trích xuất các bản đồ đặc trưng. Backbone của YOLOv8 học cách biểu diễn ảnh ở nhiều mức độ trừu tượng khác nhau, từ các đặc trưng mức thấp như cạnh và kết cấu, đến các đặc trưng mức cao mang ý nghĩa ngữ nghĩa của đối tượng. Kết quả của bước này là tập hợp các feature maps có độ phân giải khác nhau.



Hình 2.1: Sơ đồ minh họa kiến trúc tổng quan của mô hình YOLOv8

- **Tổng hợp đặc trưng đa tỉ lệ (Multi-scale Feature Fusion):** Các feature maps từ Backbone được truyền qua **Neck**, nơi kiến trúc FPN và PAN được sử dụng để kết hợp thông tin giữa các tầng. Quá trình này tạo ra các tầng đặc trưng đa tỉ lệ, thường ký hiệu là $P3$, $P4$ và $P5$, tương ứng với các mức độ chi tiết khác nhau trong ảnh. Việc sử dụng đặc trưng đa tỉ lệ giúp YOLOv8 phát hiện hiệu quả các đối tượng có kích thước khác nhau, từ các đối tượng nhỏ đến các đối tượng lớn trong cùng một khung hình.

- **Dự đoán đầu ra (Prediction):** Mỗi tầng đặc trưng ($P3$, $P4$, $P5$) được đưa vào **Head** để thực hiện dự đoán. Tại đây, YOLOv8 áp dụng kiến trúc **Decoupled Head**, trong đó các nhánh dự đoán được tách riêng cho:

  – **Vị trí hộp giới hạn (Bounding Box Regression):** dự đoán tọa độ hộp bao quanh đối tượng.

  – **Phân loại (Classification):** xác định lớp của đối tượng.

  YOLOv8 sử dụng cơ chế **Anchor-Free**, nghĩa là mô hình trực tiếp dự đoán hộp giới hạn dựa trên tâm đối tượng thay vì phụ thuộc vào các anchor boxes được xác định trước. Cách tiếp cận này giúp đơn giản hóa mô hình, giảm số siêu tham số và cải thiện độ chính xác trong quá trình huấn luyện.

Sau khi hoàn tất dự đoán, các hộp giới hạn dư thừa sẽ được loại bỏ thông qua thuật toán **Non-Maximum Suppression (NMS)** nhằm giữ lại các dự đoán có độ tin cậy cao nhất. Kết quả cuối cùng của mô hình bao gồm tập các đối tượng được phát hiện cùng với nhãn lớp và vị trí tương ứng trong ảnh.

Nhờ cơ chế xử lý một giai đoạn và kiến trúc tối ưu, YOLOv8 đạt được tốc độ suy luận nhanh trong khi vẫn duy trì độ chính xác cao. Do đó, mô hình đặc biệt phù hợp với các ứng dụng yêu cầu xử lý thời gian thực, bao gồm:

- Hệ thống giám sát và an ninh thông minh
- Các bài toán xe tự hành và hỗ trợ lái xe
- Triển khai trên thiết bị nhúng và hệ thống AI biên (Embedded AI, Edge AI)

2.3.9 Thuật toán và Hàm Mất Mát (Loss Function)

Thuật toán phát hiện đối tượng của YOLOv8 được xây dựng dựa trên nguyên tắc **dự đoán trực tiếp (Direct Prediction)** trên các ô lưới (grid cells) của các bản đồ đặc trưng đa tỉ lệ. Ở mỗi vị trí trên bản đồ đặc trưng, mô hình đồng thời dự đoán thông tin hình học của hộp giới hạn và thông tin ngữ nghĩa của đối tượng, cho phép toàn bộ quá trình phát hiện được thực hiện trong một bước duy nhất.

- **Cơ chế phát hiện Anchor-Free:**

  Khác với các phiên bản YOLO truyền thống sử dụng *anchor boxes* được thiết kế thủ công, YOLOv8 áp dụng cơ chế **anchor-free**. Theo cách tiếp cận này, mô hình không còn dự đoán độ lệch so với các anchor cố định, mà trực tiếp dự đoán:

  - Tọa độ tâm đối tượng (x, y),
  - Kích thước hộp giới hạn (w, h),
  - Điểm số đối tượng (*objectness score*),
  - Xác suất thuộc về từng lớp (*class probabilities*).

  Cơ chế anchor-free giúp đơn giản hóa quá trình huấn luyện, loại bỏ bước lựa chọn và tinh chỉnh anchor boxes, đồng thời giảm sự phụ thuộc vào dữ liệu huấn luyện. Nhờ đó, YOLOv8 có khả năng hội tụ nhanh hơn và cải thiện khả năng tổng quát hóa trên các tập dữ liệu khác nhau.

- **Hàm Mất Mát Hồi quy (Localization Loss):**

  Hàm mất mát hồi quy chịu trách nhiệm đánh giá độ chính xác của việc dự đoán hộp giới hạn. YOLOv8 sử dụng các biến thể nâng cao của **Intersection over Union (IoU)** như **CIoU Loss** hoặc **D-IoU Loss** nhằm khắc phục những hạn chế của IoU truyền thống.

  Trong đó, **CIoU Loss** không chỉ xét đến mức độ chồng lấn giữa hộp dự đoán và hộp thật, mà còn đồng thời tối ưu:

  - Khoảng cách giữa tâm của hai hộp,
  - Sự tương đồng về tỉ lệ khung hình (aspect ratio).

  Công thức CIoU Loss được biểu diễn như sau:

  $$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2(\mathbf{b}, \mathbf{b}_{gt})}{c^2} + \alpha v$$

  Trong đó, $\rho^2(\mathbf{b}, \mathbf{b}_{gt})$ là bình phương khoảng cách Euclidean giữa tâm của hộp dự đoán và hộp thật, c là độ dài đường chéo của hộp bao trọn nhỏ nhất chứa cả hai hộp, v đo mức độ khác biệt về tỉ lệ khung hình và α là hệ số cân bằng. Việc sử dụng CIoU Loss giúp mô hình hội tụ nhanh hơn và dự đoán hộp giới hạn chính xác hơn, đặc biệt trong các trường hợp đối tượng có kích thước nhỏ hoặc tỷ lệ khung hình đa dạng.

- **Hàm Mất Mát Phân loại (Classification Loss):**

  Hàm mất mát phân loại được sử dụng để đánh giá mức độ chính xác của việc dự đoán nhãn lớp cho mỗi đối tượng. YOLOv8 thường sử dụng **Binary Cross-Entropy Loss (BCE Loss)** để tính toán sai lệch giữa nhãn dự đoán và nhãn thực tế:

  $$L_{BCE} = -\left[y \log(p) + (1 - y) \log(1 - p)\right] \tag{2.1}$$

  trong đó $y \in \{0, 1\}$ là nhãn thực và p là xác suất dự đoán của mô hình.

  Tuy nhiên, trong bài toán phát hiện đối tượng, dữ liệu thường tồn tại sự mất cân bằng lớn giữa các lớp cũng như giữa các mẫu nền (background) và tiền cảnh (foreground). Để giải quyết vấn đề này, YOLOv8 có thể sử dụng **Focal Loss**, nhằm giảm ảnh hưởng của các mẫu dễ phân loại và tập trung nhiều hơn vào các mẫu khó:

  $$L_{FL}(p_t) = -\alpha(1 - p_t)^\gamma \log(p_t) \tag{2.2}$$

  Trong đó, γ là tham số điều chỉnh mức độ tập trung vào các mẫu khó, còn α đóng vai trò cân bằng giữa các lớp hiếm và phổ biến. Việc sử dụng Focal Loss giúp cải thiện đáng kể độ chính xác phân loại, đặc biệt trong các tập dữ liệu có phân bố lớp không đồng đều.

Tổng thể, hàm mất mát của YOLOv8 là sự kết hợp giữa mất mát hồi quy và mất mát phân loại, cho phép mô hình đồng thời tối ưu hóa vị trí và nhãn của đối tượng. Sự kết hợp này đóng vai trò quan trọng trong việc nâng cao độ chính xác và tính ổn định của YOLOv8 trong các bài toán phát hiện đối tượng thực tế.

2.4 Thuc thi

Ví dụ sử dụng:

Hình 2.2: Ví dụ

Huấn luyện được thực hiện trên Google Colab, sử dụng GPU Tesla T4. Cấu trúc thư mục như sau:

```
/content/rider-dataset/
 train/
  images/
  labels/
 val/
 images/
 labels/
```

File cấu hình YAML:

```
train: /content/rider-dataset/train/images
val: /content/rider-dataset/val/images
nc: 4
names: ["helmet", "no_helmet","rider", "number_plate" ]
```

2.5 Triển khai mô hình phát hiện mũ bảo hiểm

2.5.1 Mục tiêu triển khai

Sau khi huấn luyện mô hình YOLOv8 trên tập dữ liệu gồm bốn lớp *helmet*, *no_helmet*, *rider* và *number_plate*, nhóm tiến hành triển khai mô hình nhằm đánh giá khả năng phát hiện mũ bảo hiểm và vi phạm giao thông trên dữ liệu thực tế. Việc triển khai được thực hiện dưới dạng chương trình Python chạy cục bộ, sử dụng mô hình đã huấn luyện tốt nhất (`best.pt`).

2.5.2 Môi trường và công cụ sử dụng

Hệ thống được triển khai bằng ngôn ngữ Python với các thư viện chính sau:

- **Ultralytics YOLOv8**: dùng để nạp và suy luận mô hình phát hiện đối tượng.
- **OpenCV**: xử lý và đọc ảnh đầu vào.
- **Matplotlib**: hiển thị trực quan kết quả phát hiện.
- **NumPy**: hỗ trợ xử lý dữ liệu ảnh.

Việc sử dụng các thư viện phổ biến giúp hệ thống dễ triển khai, dễ mở rộng và phù hợp với các ứng dụng thực tế.

2.5.3 Cấu hình tham số và dữ liệu đầu vào

Trong quá trình triển khai, các tham số chính được cấu hình như sau:

- **Mô hình**: `best.pt` – mô hình YOLOv8 đã được huấn luyện.
- **Ảnh đầu vào**: ảnh giao thông dùng để kiểm tra khả năng phát hiện của mô hình.
- **Ngưỡng confidence**: chỉ giữ lại các bounding box có độ tin cậy lớn hơn một giá trị xác định (ví dụ 0.25).

Việc đặt ngưỡng confidence giúp loại bỏ các dự đoán không chắc chắn và giảm nhiễu trong kết quả đầu ra.

2.5.4 Quy trình suy luận của mô hình

Quá trình suy luận (inference) được thực hiện bằng cách đưa ảnh đầu vào qua mô hình YOLOv8 đã huấn luyện. Pipeline xử lý bên trong mô hình bao gồm các bước chính sau:

1. Ảnh đầu vào được tiền xử lý (resize, chuẩn hóa).
2. Backbone trích xuất các đặc trưng không gian và ngữ nghĩa từ ảnh.
3. Neck kết hợp đặc trưng đa tỉ lệ nhằm cải thiện khả năng phát hiện các đối tượng có kích thước khác nhau.
4. Head thực hiện dự đoán trực tiếp (anchor-free) các bounding box, objectness score và xác suất lớp.
5. Áp dụng thuật toán Non-Maximum Suppression (NMS) để loại bỏ các bounding box trùng lặp.

Kết quả đầu ra bao gồm vị trí bounding box, nhãn dự đoán và độ tin cậy tương ứng cho từng đối tượng.

2.5.5 Lưu trữ và hiển thị kết quả

Sau khi suy luận, các kết quả phát hiện được YOLOv8 tự động lưu vào thư mục `runs/detect/`. Ảnh kết quả được vẽ bounding box, hiển thị nhãn lớp (*helmet*, *no_helmet*, *rider*, *number_plate*) và giá trị confidence.

Nhóm sử dụng OpenCV để đọc ảnh kết quả và Matplotlib để hiển thị trực quan. Việc hiển thị này giúp đánh giá nhanh khả năng phát hiện của mô hình cũng như quan sát các trường hợp dự đoán sai hoặc chưa chính xác.

2.5.6 Ý nghĩa và khả năng ứng dụng

Phần triển khai cho thấy mô hình YOLOv8 sau khi huấn luyện có thể hoạt động hiệu quả trên dữ liệu thực tế. Hệ thống có tiềm năng mở rộng để:

- phát hiện vi phạm mũ bảo hiểm trong video hoặc luồng camera thời gian thực,
- kết hợp với nhận dạng biển số để phục vụ các hệ thống giám sát giao thông,
- hỗ trợ các ứng dụng xử phạt nguội và quản lý an toàn giao thông.

2.5.7 Tổng kết

Quy trình triển khai mô hình được thực hiện theo pipeline hoàn chỉnh: *ảnh đầu vào* $\rightarrow$ *tiền xử lý* $\rightarrow$ *YOLOv8 suy luận* $\rightarrow$ *NMS* $\rightarrow$ *hiển thị và lưu kết quả*. Phần này chứng minh tính khả thi và giá trị ứng dụng thực tế của mô hình phát hiện mũ bảo hiểm đã xây dựng.

Chương 3

Kết quả & Phân tích

3.1 Kết quả huấn luyện

Bảng 3.1: Hiệu suất mô hình YOLOv8 trên tập kiểm thử

Chỉ số	**Precision**	**Recall**	**mAP@50**	**mAP@50-95**
Giá trị	0.923	0.876	0.901	0.764

3.2 Phân tích

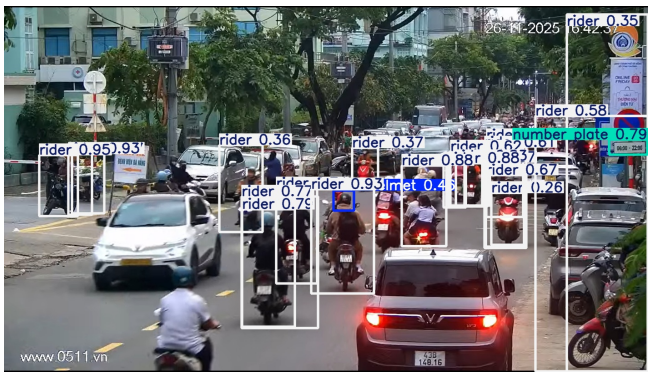
Kết quả thực nghiệm cho thấy mô hình YOLOv8 đạt hiệu suất tổng thể cao trên tập kiểm thử, đặc biệt đối với các lớp có đặc điểm hình học và ngữ nghĩa rõ ràng. Tuy nhiên, hiệu suất giữa các lớp vẫn tồn tại sự chênh lệch đáng kể, phản ánh những thách thức đặc thù của bài toán phát hiện đối tượng trong môi trường thực tế.

Cụ thể, các lớp *rider* và *number plate* đạt giá trị mAP tương đối cao (lần lượt xấp xỉ 0.898 và 0.82). Điều này có thể được giải thích bởi hai yếu tố chính. Thứ nhất, các đối tượng này thường có kích thước tương đối lớn trong ảnh và ít bị che khuất hoàn toàn. Thứ hai, hình dạng và đặc trưng thị giác của chúng khá khác biệt so với nền và các lớp còn lại, giúp mô hình dễ dàng học được các đặc trưng phân biệt trong quá trình huấn luyện.

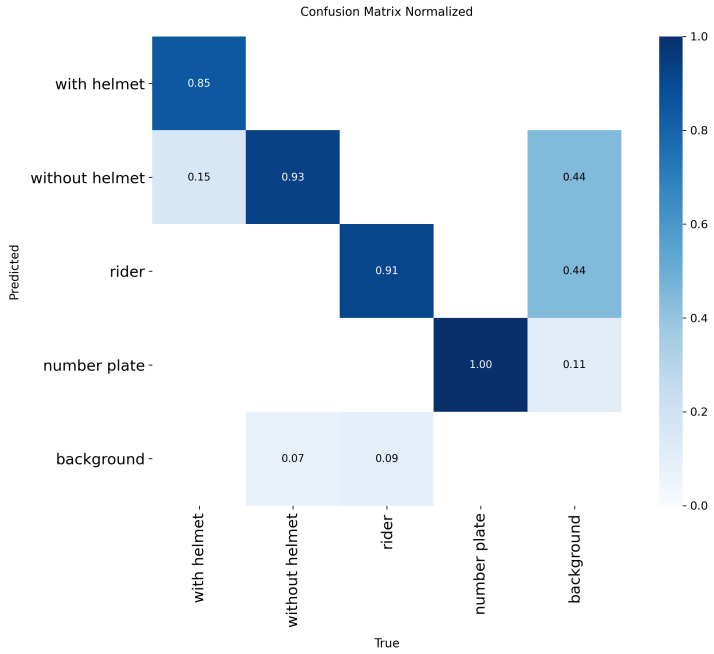
Ngược lại, hai lớp *helmet* và *no_helmet* có mức độ nhận dạng thấp hơn. Nguyên nhân chủ yếu xuất phát từ sự tương đồng về hình dạng, kích thước nhỏ và vị trí xuất hiện của hai lớp này. Trong nhiều trường hợp, mũ bảo hiểm có màu sắc gần với tóc hoặc bị che khuất một phần bởi góc nhìn, ánh sáng hoặc vật thể khác, khiến mô hình gặp khó khăn trong việc phân biệt rõ ràng giữa hai trạng thái có và không đội mũ.

Ma trận nhầm lẫn (Hình 3.3) cho thấy hiện tượng nhầm lẫn chủ yếu xảy ra giữa hai lớp *helmet* và *no_helmet*. Cụ thể, trong các điều kiện ánh sáng yếu, ảnh chụp ban đêm hoặc khi đối tượng bị che khuất một phần, mô hình có xu hướng dự đoán sai trạng thái đội mũ. Điều này phản ánh hạn chế của đặc trưng thị giác trong những tình huống có độ tương phản thấp và cho thấy nhu cầu cải thiện dữ liệu huấn luyện cũng như chiến lược học đặc trưng cho các đối tượng nhỏ.

Hình 3.1: Ảnh đã nhận diện bằng mô hình YOLOv8



Hình 3.2: Ảnh đã nhận diện bằng mô hình YOLOv8



Hình 3.3: Ma trận nhầm lẫn của mô hình YOLOv8

Về các chỉ số đánh giá tổng thể, mô hình đạt được:

- **Precision = 0.923**: Giá trị precision rất cao cho thấy trong số các đối tượng mà mô hình dự đoán là dương tính, phần lớn là dự đoán đúng. Điều này đặc biệt quan trọng trong các hệ thống giám sát và xử phạt tự động, vì nó giúp giảm thiểu số lượng cảnh báo hoặc xử phạt sai (false positive), từ đó nâng cao độ tin cậy của hệ thống.

- **Recall = 0.876**: Giá trị recall cao cho thấy mô hình có khả năng phát hiện được phần lớn các đối tượng thực sự tồn tại trong ảnh. Tuy nhiên, vẫn còn khoảng 12.4% đối tượng bị bỏ sót (false negative), chủ yếu rơi vào các trường hợp đối tượng nhỏ, bị che khuất hoặc xuất hiện trong điều kiện ánh sáng không thuận lợi.

- **mAP@50 = 0.901**: Chỉ số mAP tại ngưỡng IoU = 0.50 đạt mức rất cao (90.1%), chứng tỏ mô hình có khả năng phân loại đúng và định vị tương đối chính xác các đối tượng trong hầu hết các trường hợp thông thường. Đây là mức hiệu suất phù hợp cho các ứng dụng thực tế như giám sát giao thông hoặc phát hiện vi phạm.

- **mAP@50–95 = 0.764**: Khi đánh giá trên các ngưỡng IoU nghiêm ngặt hơn, hiệu suất của mô hình giảm xuống còn 76.4%. Sự sụt giảm này cho thấy mặc dù mô hình nhận diện tốt sự tồn tại của đối tượng, nhưng độ chính xác của việc định vị hộp giới hạn chưa thực sự ổn định ở các yêu cầu khắt khe hơn. Điều này đặc biệt rõ rệt đối với các đối tượng nhỏ như *helmet*, nơi chỉ cần sai lệch nhỏ cũng làm giảm đáng kể giá trị IoU.

Tổng hợp các kết quả trên cho thấy YOLOv8 là một mô hình hiệu quả cho bài toán phát hiện người lái xe, biển số và trạng thái đội mũ bảo hiểm. Tuy nhiên, để nâng cao độ chính xác trong các tình huống khó, đặc biệt là phân biệt giữa *helmet* và *no_ helmet*, cần có các cải tiến bổ sung về dữ liệu, kiến trúc mô hình hoặc chiến lược huấn luyện trong các nghiên cứu tiếp theo.

Chương 4

Kết luận

4.1 Kết luận

Dự án **Helmet Protection Detection AI** đã hoàn thành mục tiêu xây dựng một hệ thống phát hiện việc đội mũ bảo hiểm dựa trên mô hình thị giác máy tính hiện đại. Thông qua việc ứng dụng kiến trúc **YOLOv8**, nhóm đã triển khai thành công mô hình có khả năng phát hiện các đối tượng quan trọng trong bối cảnh giao thông – bao gồm `helmet`, `no helmet`, `rider`, và `number plate`.

Bộ dữ liệu gồm 500 ảnh được xử lý đã giúp mô hình học được các đặc điểm trực quan đặc trưng, dù còn tồn tại những thách thức trong điều kiện ánh sáng yếu hoặc góc nhìn khó. Quá trình huấn luyện trên GPU đã cho phép mô hình đạt hiệu suất tốt trên tập kiểm thử, với các chỉ số:

- **Precision = 0.923**: cho thấy hệ thống có khả năng dự đoán chính xác cao, hạn chế sai sót dương tính giả.
- **Recall = 0.876**: mô hình phát hiện được đa số các đối tượng thực tế trong ảnh.
- **mAP@50 = 0.901**: khẳng định hiệu quả của mô hình trong nhiệm vụ phát hiện và phân loại đối tượng ở mức độ vị trí tương đối.
- **mAP@50-95 = 0.764**: phản ánh khả năng định vị chính xác ở các ngưỡng IoU chặt chẽ hơn.

Các kết quả huấn luyện và ma trận nhầm lẫn cho thấy mô hình hoạt động ổn định, đặc biệt ở hai lớp `rider` và `number plate`. Mặc dù hai lớp `helmet` và `no helmet` đôi khi bị nhầm lẫn trong điều kiện phức tạp, hiệu suất tổng thể vẫn đáp ứng mục tiêu ban đầu đặt ra của dự án.

Từ những phân tích này, có thể khẳng định rằng hệ thống phát hiện mũ bảo hiểm dựa trên YOLOv8 là một giải pháp khả thi, hiệu quả và có tiềm năng ứng dụng trong thực tế – đặc biệt trong các hệ thống giám sát giao thông thông minh. Dự án cung cấp nền tảng vững chắc để phát triển các hệ thống AI hỗ trợ đảm bảo an toàn giao thông, đồng thời thể hiện khả năng ứng dụng mô hình học sâu vào các bài toán thực tiễn.

4.2 Hướng phát triển

Dựa trên quá trình nghiên cứu, xây dựng và đánh giá mô hình YOLOv8 cho bài toán phát hiện người lái xe và đội mũ bảo hiểm, dự án đề xuất một số hướng phát triển cụ thể nhằm nâng cao hiệu quả và tính ứng dụng thực tế của hệ thống trong tương lai.

Phát triển về mặt kỹ thuật mô hình

- **Mở rộng và cân bằng tập dữ liệu huấn luyện:** Thu thập thêm dữ liệu trong các điều kiện thực tế phức tạp như ánh sáng yếu, ban đêm, trời mưa, góc quay từ camera giám sát giao thông. Đồng thời, thực hiện cân bằng số lượng mẫu giữa các lớp (`helmet`, `no helmet`, `rider`, `number plate`) nhằm giảm hiện tượng thiên lệch mô hình và cải thiện độ chính xác tổng thể.
- **Áp dụng kỹ thuật Data Augmentation nâng cao:** Ngoài các phép biến đổi cơ bản (xoay, lật, thay đổi độ sáng), có thể áp dụng các kỹ thuật như Mosaic Augmentation, MixUp hoặc Random Occlusion để mô phỏng các tình huống che khuất đối tượng, giúp mô hình tăng khả năng tổng quát hóa khi triển khai thực tế.

- **Tối ưu kiến trúc và tham số huấn luyện:** Thử nghiệm các biến thể khác nhau của YOLOv8 (YOLOv8n, YOLOv8s, YOLOv8m) để tìm ra sự cân bằng tối ưu giữa tốc độ và độ chính xác. Ngoài ra, điều chỉnh các siêu tham số như learning rate, batch size và số epoch huấn luyện nhằm cải thiện chỉ số `mAP50` và giảm hiện tượng overfitting.
- **Tối ưu suy luận (Inference Optimization):** Triển khai các kỹ thuật như TensorRT, ONNX hoặc quantization (INT8/FP16) để giảm thời gian suy luận, từ đó nâng cao tốc độ xử lý khung hình, đảm bảo hệ thống đạt trên 20 FPS khi triển khai trên thiết bị phần cứng hạn chế.

Cải tiến về mặt hệ thống và ứng dụng

- **Tích hợp theo chuỗi xử lý (Pipeline đa bước):** Xây dựng pipeline gồm các bước: phát hiện người lái xe → phát hiện mũ bảo hiểm → phát hiện biển số xe. Cách tiếp cận này giúp tăng độ chính xác trong việc xác định vi phạm và giảm các dự đoán sai không cần thiết.
- **Kết hợp với hệ thống nhận dạng biển số (ANPR):** Sau khi phát hiện đối tượng vi phạm `no helmet`, hệ thống có thể tự động kích hoạt mô-đun nhận dạng biển số xe để trích xuất thông tin phương tiện, tiến tới tự động hóa quy trình xử phạt giao thông.
- **Phát triển giao diện giám sát trực quan:** Xây dựng giao diện web hoặc desktop hiển thị video thời gian thực, bounding box, nhãn đối tượng và độ tin cậy dự đoán. Đồng thời, lưu trữ lịch sử các trường hợp vi phạm để phục vụ công tác thống kê và quản lý.
- **Mở rộng khả năng triển khai trên thiết bị nhúng:** Nghiên cứu triển khai hệ thống trên các nền tảng nhúng như NVIDIA Jetson Nano hoặc Jetson Xavier, nhằm hướng tới ứng dụng thực tế tại các điểm giao thông với chi phí thấp và khả năng hoạt động liên tục.

Những đề xuất cải tiến trên không chỉ giúp nâng cao hiệu suất của mô hình YOLOv8 mà còn góp phần tăng tính thực tiễn và khả năng triển khai của hệ thống trong các bài toán giám sát và quản lý giao thông thông minh.

Tài liệu tham khảo

[1] Ultralytics, YOLOv8 Documentation," 2024. Available at: https://docs.ultralytics.com/

[2] Penny.P, YOLOv8n - Rider with/without Helmet, Available at:https://www.kaggle.com/code/yaaaaam1/yolov8n-rider-with-without-helmet, 2023+

[3] OpenCV Team, "Real-time Object Detection with OpenCV and PyTorch," 2022.

Phụ lục A

Phụ lục