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

Helmet Protectioin 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

Nhóm áp dụng YOLOv8 – một mô hình phát hiện vật thể hiện đại, có tốc độ nhanh và độ chính xác cao. Dữ liệu được lấy từ dataset "Rider, With Helmet, Without Helmet, Number Plate" trên Kaggle của tác giả Penny.py:

- **Train:** 400 ảnh
- **Validation:** 100 ảnh
- **Classes:** helmet, no helmet, rider, number plate

Các chỉ số chính khi train:

- Epochs: 100
- Batch size: 16
- Optimizer: Adam
- Learning rate: 0.01

2.2 Mô hình Phát hiện Đối tượng: YOLOv8

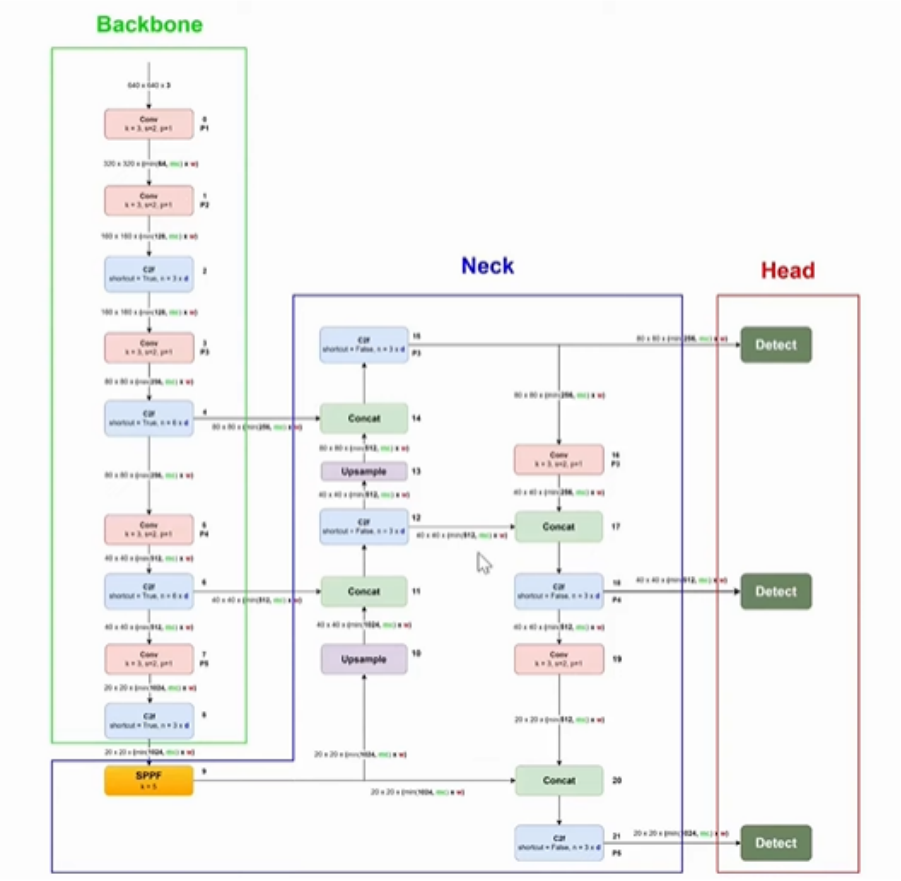
2.2.1 Giới thiệu về YOLOv8

YOLOv8 (**Y**ou **O**nly **L**ook **O**nce, phiên bản 8) là một mô hình **mạng nơ-ron tích chập (CNN)** tiên tiến được phát triển bởi **Ultralytics** để thực hiện các tác vụ thị giác máy tính, nổi bật là phát hiện đối tượng. Là mô hình **One-Stage Detector**, YOLOv8 thực hiện dự đoán hộp giới hạn và phân loại đối tượng chỉ trong **một lần truyền (forward pass)**, đảm bảo khả năng hoạt động ở tốc độ thời gian thực.

2.2.2 Cấu trúc Kiến trúc (Architecture)

Kiến trúc của YOLOv8 được tối ưu hóa để tăng cường khả năng trích xuất đặc trưng và cải thiện hiệu suất suy luận. Nó bao gồm ba thành phần chính:

1. **Backbone (Xương sống):** Chịu trách nhiệm trích xuất các **bản đồ đặc trưng (feature maps)** từ ảnh đầu vào. YOLOv8 sử dụng các khối tích chập hiệu quả, chẳng hạn như **C2f (Contextualized CSP-Former)** để tăng cường khả năng học đặc trưng phong phú với số lượng tham số ít hơn so với các phiên bản trước.
2. **Neck (Cổ):** Thực hiện tổng hợp đặc trưng. Sử dụng kết hợp kiến trúc **FPN** (Feature Pyramid Network) và **PAN** (Path Aggregation Network) để kết hợp các đặc trưng từ các cấp độ khác nhau của Backbone, tạo ra các đặc trưng đa tỉ lệ (multi-scale features) giúp phát hiện đối tượng ở nhiều kích thước khác nhau.
3. **Head (Đầu):** Sử dụng các đặc trưng được tổng hợp để đưa ra dự đoán cuối cùng. YOLOv8 áp dụng một **Decoupled Head** (đầu tách biệt), trong đó các nhánh cho **Phân loại (Classification)** và **Định vị (Localization)** được tách riêng, giúp cải thiện độ chính xác. Quan trọng hơn, YOLOv8 chuyển sang cơ chế **Anchor-Free**, loại bỏ nhu cầu về *anchor boxes* cố định, đơn giản hóa quá trình huấn luyện và tối ưu hóa hộp giới hạn.



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

2.2.3 Cách hoạt động chính

YOLOv8 là mô hình One-Stage Detector, nghĩa là nó thực hiện đồng thời phát hiện vùng (region proposal) và phân loại đối tượng trong một bước duy nhất.

- Ảnh đầu vào được đưa qua backbone để tạo đặc trưng.
- Đặc trưng được xử lý qua FPN để tạo ra các tầng đặc trưng đa tỉ lệ.
- Mỗi tầng đặc trưng (P3, P4, P5) được đưa vào head để dự đoán đầu ra (box, class, objectness).

So với các mô hình hai giai đoạn như Faster R-CNN (đề xuất vùng trước rồi mới phân loại), YOLOv8 xử lý nhanh hơn nhiều, đặc biệt hiệu quả trong các bài toán yêu cầu thời gian thực như:

- Giám sát an ninh
- Xe tự hành
- Thiết bị nhúng (embedded AI)

2.2.4 Thuật toán và Hàm Mất Mát (Loss Function)

Thuật toán phát hiện của YOLOv8 dựa trên nguyên tắc dự đoán trực tiếp (Direct Prediction) trên các ô lưới (grid cells) của các bản đồ đặc trưng.

- **Phát hiện Anchor-Free:** Thay vì dự đoán độ lệch so với các anchor boxes được xác định trước, YOLOv8 dự đoán trực tiếp tọa độ (x, y) và kích thước (w, h) của hộp giới hạn cùng với điểm số đối tượng (objectness score) và xác suất lớp (class probabilities) cho mỗi vị trí trên bản đồ đặc trưng.
- **Hàm Mất Mát Hồi quy (Localization Loss):** Sử dụng các hàm mất mát dựa trên **Intersection over Union (IoU)** nâng cao, chẳng hạn như **CIoU Loss** (Complete IoU) hoặc **D-IoU Loss**. Mục tiêu là tối

đa hóa sự chồng lấn và giảm thiểu khoảng cách trung tâm giữa hộp giới hạn dự đoán ($\mathcal{B}_{pred}$) và hộp giới hạn thật ($\mathcal{B}_{gt}$).

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

Trong đó, ρ^2 là bình phương khoảng cách Euclidean, c là đường chéo của hộp bao trọn, v là độ tương đồng về tỉ lệ khung hình (aspect ratio), và α là hệ số cân bằng.

- **Hàm Mất Mát Phân loại (Classification Loss):** Thường sử dụng **Binary Cross-Entropy Loss** hoặc **Focal Loss** để xử lý hiệu quả vấn đề mất cân bằng giữa các lớp (class imbalance) và giữa các mẫu nền/tiền cảnh (background/foreground samples).

2.3 Triển khai

Ví dụ sử dụng:

```python
from ultralytics import YOLO

# Load a COCO-pretrained YOLOv8n model
model = YOLO("yolov8n.pt")

# Display model information (optional)
model.info()

# Train the model on the COCO8 example dataset for 100 epochs
results = model.train(data="coco8.yaml", epochs=100, imgsz=640)

# Run inference with the YOLOv8n model on the 'bus.jpg' image
results = model("path/to/bus.jpg")
```

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" ]
```

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

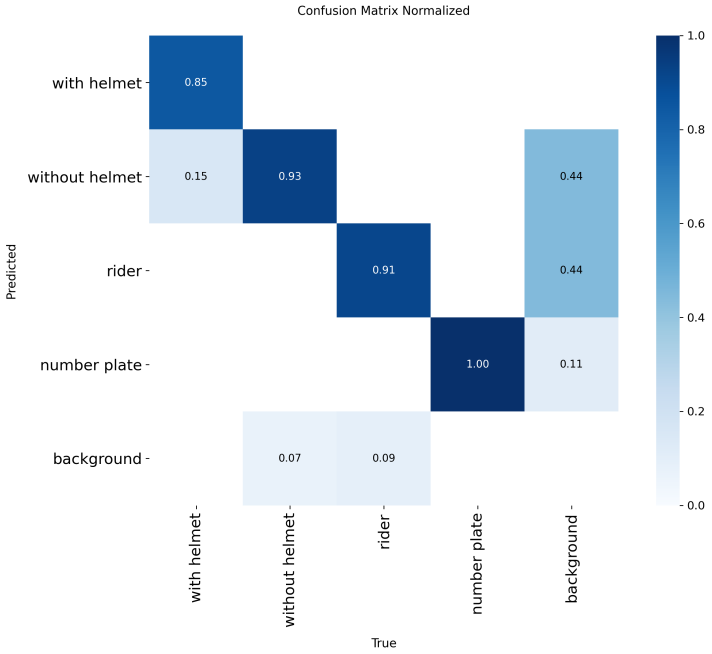
3.2 Phân tích

Các lớp như rider và number plate đạt mAP cao (0.898–0.82), trong khi hai lớp helmet và no_helmet có mức nhận dạng thấp hơn do đặc điểm hình dạng tương tự.

Mô hình thường nhầm lẫn giữa "helmet" và "no_helmet" trong các ảnh có ánh sáng yếu hoặc bị che khuất.

- **Precision (Độ chính xác) = 0.923**: Giá trị này **rất cao**, cho thấy mô hình có tỷ lệ dự đoán đúng cao trong số tất cả các dự đoán dương tính. Điều này ngụ ý rằng mô hình **ít xảy ra lỗi dương tính giả** (False Positives), tức là ít dự đoán sai một vật thể nào đó.
- **Recall (Độ phủ) = 0.876**: Giá trị **cao** này chỉ ra rằng mô hình phát hiện được phần lớn ($\approx 87.6\%$) các vật thể thực tế có mặt trong tập kiểm thử. Mô hình **ít xảy ra lỗi âm tính giả** (False Negatives), tức là ít bỏ sót vật thể.
- **mAP@50 = 0.901**: Giá trị **rất cao** (**90.1**%) của Mean Average Precision tại ngưỡng IoU (Intersection over Union) 0.50 chứng tỏ mô hình hoạt động **rất hiệu quả** trong cả việc phân loại và định vị vật thể.
- **mAP@50-95 = 0.764**: Chỉ số này (76.4%) thể hiện hiệu suất trung bình trên các ngưỡng IoU nghiêm ngặt hơn (từ 0.50 đến 0.95). Sự **sụt giảm** đáng kể so với mAP@50 cho thấy độ chính xác của hộp giới hạn (bounding box) bị giảm đi khi yêu cầu định vị chặt chẽ hơn, đây là điểm cần cải thiện.

Hình 3.1: Ảnh đã nhận diện bằng mô hình YOLOv8



Hình 3.2: Ma trận nhầm lẫn của mô hình YOLOv8

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 Đề xuất cải tiến

Dự án này đặt ra các mục tiêu cụ thể sau:

Cải tiến Kỹ thuật

- Xây dựng và huấn luyện mô hình YOLOv8 để đạt được hiệu suất phát hiện (`mAP50`) tối thiểu là 90%.
- Đảm bảo khả năng xử lý và phát hiện đối tượng với tốc độ phù hợp cho ứng dụng thời gian thực (ví dụ: tốc độ khung hình > 20 FPS).
- Phát triển giao diện hoặc API tích hợp để hiển thị trực quan kết quả phát hiện, bao gồm vị trí (bounding box) và độ tin cậy của đối tượng (`helmet`, `no helmet`, `rider`, `number plate`).

Ứng dụng

- Tạo ra một hệ thống có tính ổn định và khả năng hoạt động tốt trong nhiều điều kiện môi trường khác nhau (ngày, đêm, ánh sáng yếu, góc nhìn đa dạng).
- Cung cấp nền tảng để cơ quan chức năng có thể dễ dàng mở rộng, tích hợp thêm các tính năng như nhận dạng biển số xe (ANPR) để tự động hóa hoàn toàn quy trình xử phạt.

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