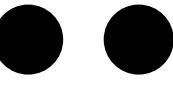


HELMET PROTECTION DETECTION AI



Nhóm thực hiện

Nhóm 15 – Môn Nhập môn Trí tuệ Nhân tạo

Họ và tên	MSSV	GitHub username
-----------	------	-----------------

----- ----- -----

Đặng Khánh Đạt 23001514 @datdang401205-jpg (https://github.com/datdang401205-jpg)

Nguyễn Hải Đăng 23001516 @dawn-ds-15 (https://github.com/dawn-ds-15)
--

Trương Mậu Anh 23001538 @truongmauanh (https://github.com/truongmauanh)

Bùi Phương Nam 23001498 @buiphuongnam23001538 (https://github.com/buiphuongnam23001538)



HELMET PROTECTION DETECTION AI



01

Giới thiệu

02

Phương pháp & Triển khai

03

Kết quả & Phân tích

04

Kết luận

01

GIỚI THIỆU

TÓM TẮT

BÀI TOÁN ĐẶT RA

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).

1.2 BÀI TOÁN ĐẶT RA VÀ ĐỘNG LỰC

Bài toán :

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.

02

PHƯƠNG PHÁP & TRIỂN KHAI

2.1 PHƯƠNG PHÁP

2.2 MÔ HÌNH PHÁT HIỆN ĐỐI TƯỢNG: YOLOV8

2.3 TRIỂN KHAI

2.3 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

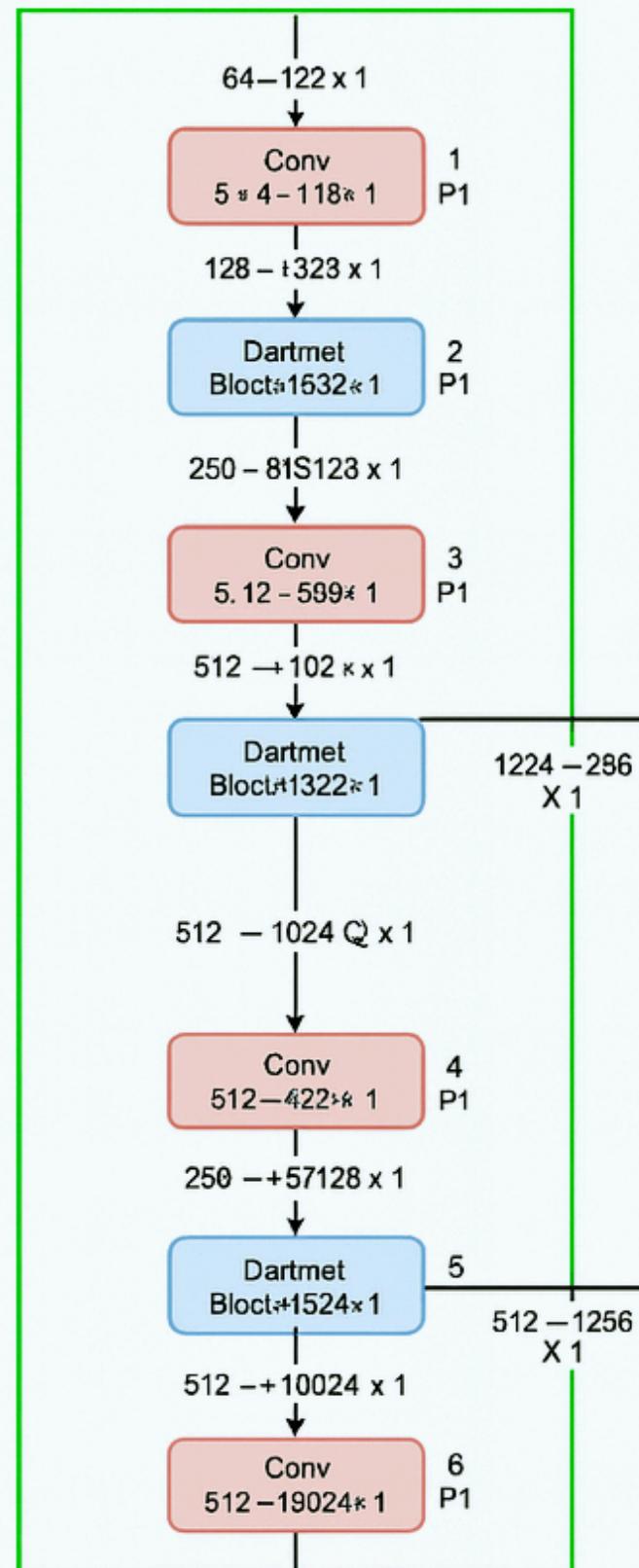
YOLOv8 (You Only Look Once, phiên bản 8) là một mô hình mạng **nơ-ron tích chập (CNN)** tiên tiến được phát triển bởi Ultralytics để thực hiện các tác vụ thị giác máy tính, nổi bật là phát hiện đối tượng. Là **mô hình One-Stage Detector**, YOLOv8 thực hiện dự đoán hộp giới hạn và phân loại đối tượng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

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:

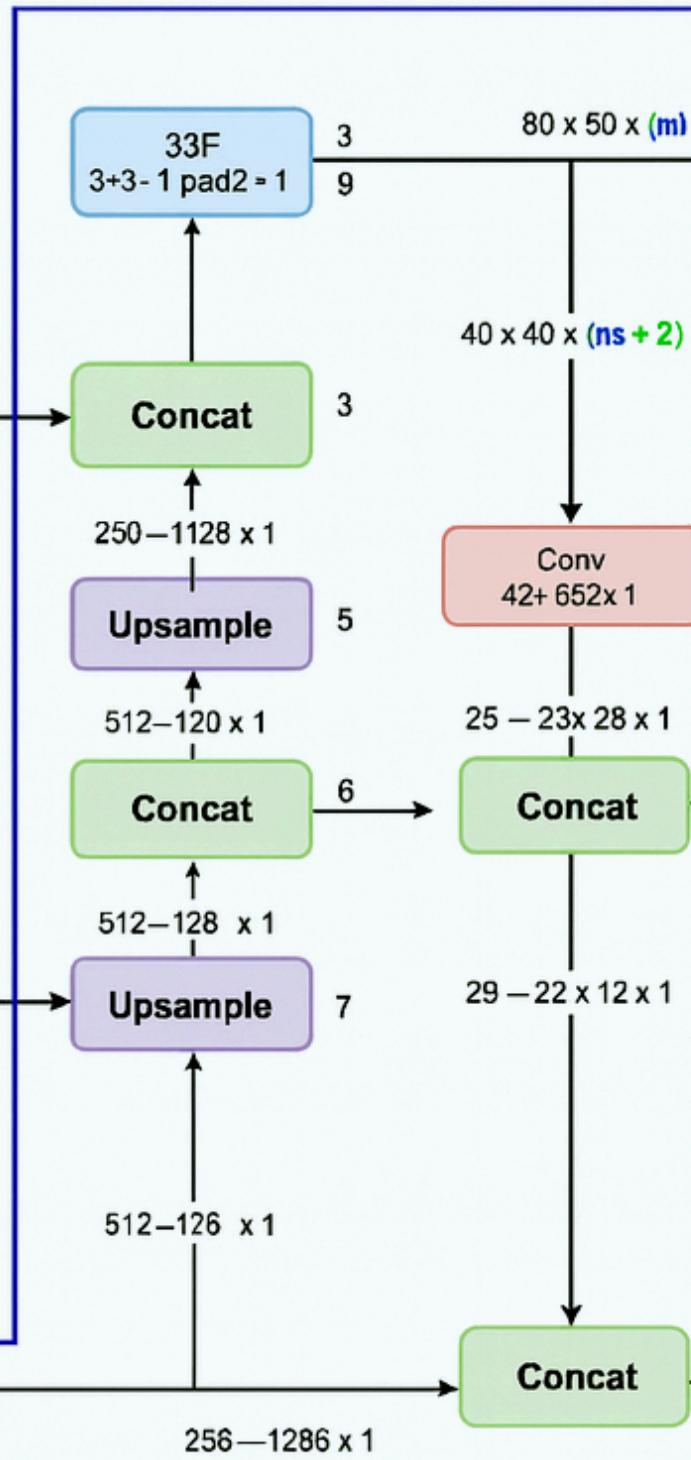
- 1. Backbone (Xương sống):** Chịu trách nhiệm trích xuất các bản đồ đặc trưng (feature maps) từ ảnh đầu vào. YOLOv8 sử dụng các khối tích chập hiệu quả, chẳng hạn như C2f (Contextualized CSP-Fomer) để tăng cường khả năng học đặc trưng phong phú với số lượng tham số ít hơn so với các phiên bản trước.
- 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.

Backbone



Sep
(2+4)

Neck



Head

Detect

Detect

Detect

SƠ ĐỒ MINH HỌA KIẾN TRÚC TỔNG QUAN CỦA MÔ HÌNH YOLOV8

Cách hoạt động chính

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.

1. 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.

2. Hàm Mất Mát Hồi quy (Localization Loss): Sử dụng các hàm mất mát dựa trên Intersection over Union (IoU) nâng cao, chẳng hạn như CloU 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 (Bpred) và hộp giới hạn thật (Bgt).

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

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.

3. 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

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" ]
```



3

KẾT QUẢ & PHÂN TÍCH

3.1 KẾT QUẢ HUẤN LUYỆN

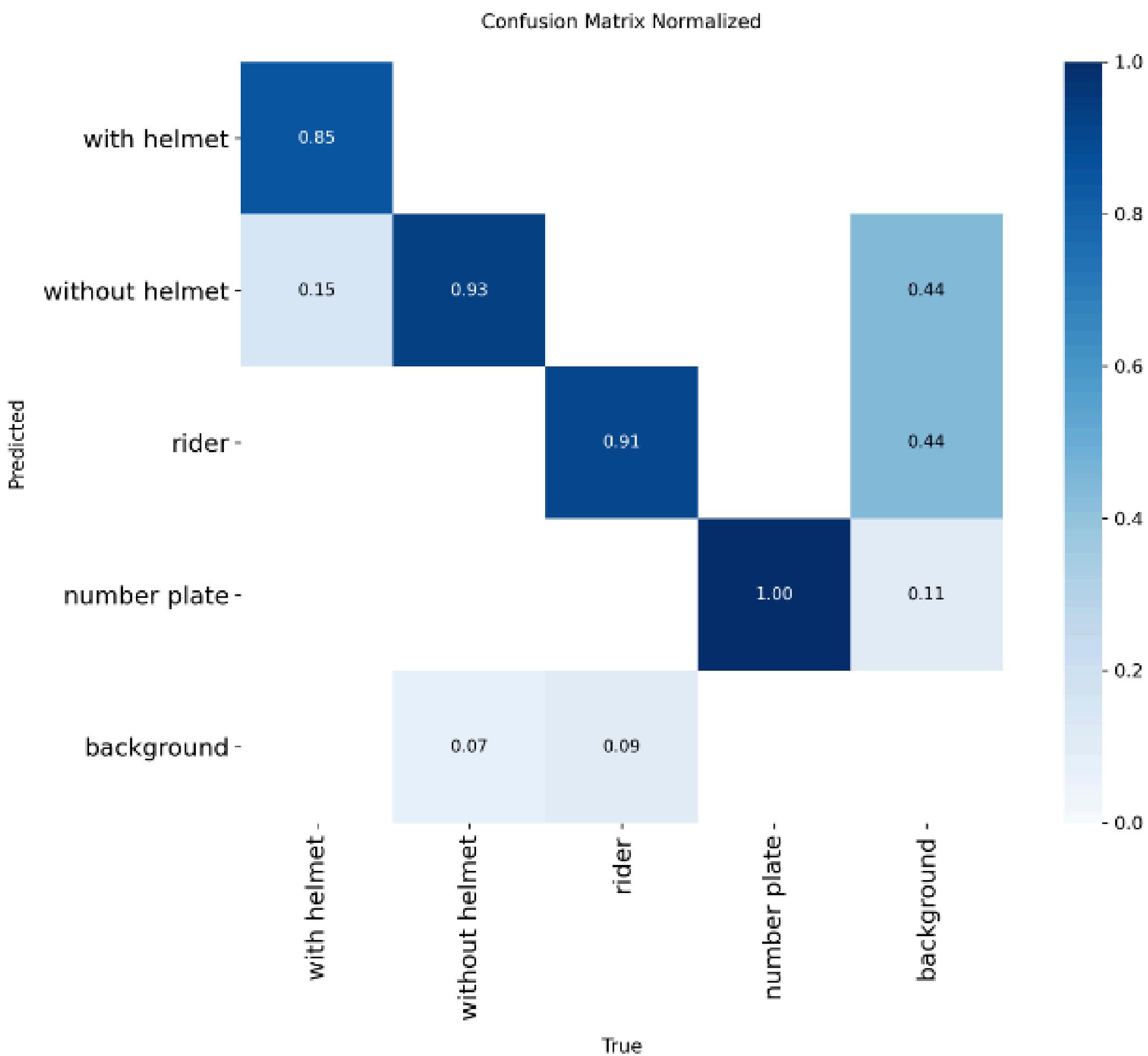
3.2 PHÂN TÍCH

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



MA TRẬN NHẦM LẦN CỦA MÔ HÌNH YOLOV8

3.2 PHÂN TÍCH

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.

04

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.

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**.

4.2 ĐỀ XUẤT CẢI TIẾN

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.

**THANK YOU
FOR LISTENING**