Báo cáo Lab Week 5 – Object Detection

Mask Wearing Detection

**Môn học**: Xử lý ảnh và Ứng dụng – CS406.Q12

GVHD: Cáp Phạm Đình Thăng

Sinh viên thực hiện: Trần Bảo Trân – 23521623

# Giới thiệu bài toán

Trong bối cảnh hậu đại dịch COVID-19 và các vấn đề về ô nhiễm không khí, việc giám sát tuân thủ đeo khẩu trang tại các khu vực công cộng là vô cùng cần thiết. Bài toán đặt ra là xây dựng một hệ thống thị giác máy tính (Computer Vision) có khả năng:

* **Object Detection:** Xác định vị trí khuôn mặt trong ảnh (Bounding Box)
* **Classification:** Phân loại khuôn mặt đó thuộc lớp with\_mask (có đeo) hay without\_mask (không đeo)

Trong báo cáo này, mô hình **YOLOv11 (Ultralytics YOLO)** được fine-tune để:

* Phát hiện khuôn mặt -> bbox cho khuôn mặt -> phân loại thành 2 lớp:
  + **with\_mask**
  + **without\_mask**

# Mô tả dataset

Dữ liệu được sử dụng gồm:

853 ảnh (file .png) chụp thực tế tại các khu vực đông người hoặc ảnh selfie tự chụp có chứa một hoặc nhiều người, đám đông trong nhiều điều kiện ánh sáng, góc nhìn, đối tượng người trong ảnh có đeo khẩu trang hoặc không, được chụp khi đang thực hiện nhiều động tác, hành động đa dạng. Dataset gồm 2 folders: images và annotations.

* **Định dạng ảnh (folder “images”):** 853 file ảnh màu (.png), chụp ngang hoặc dọc. Mỗi ảnh chứa một hoặc nhiều đối tượng ở các cự ly khác nhau, trong đó có người đeo khẩu trang và người không đeo, với kích thước khuôn mặt đa dạng.

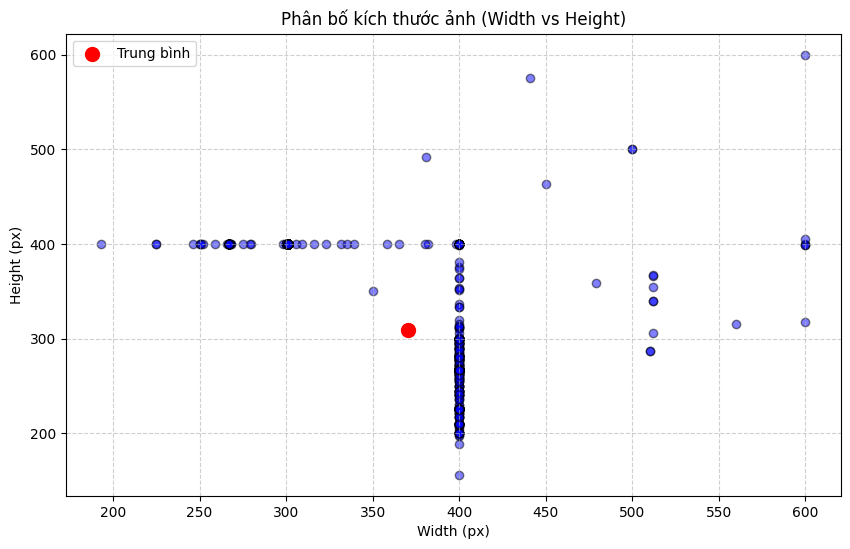
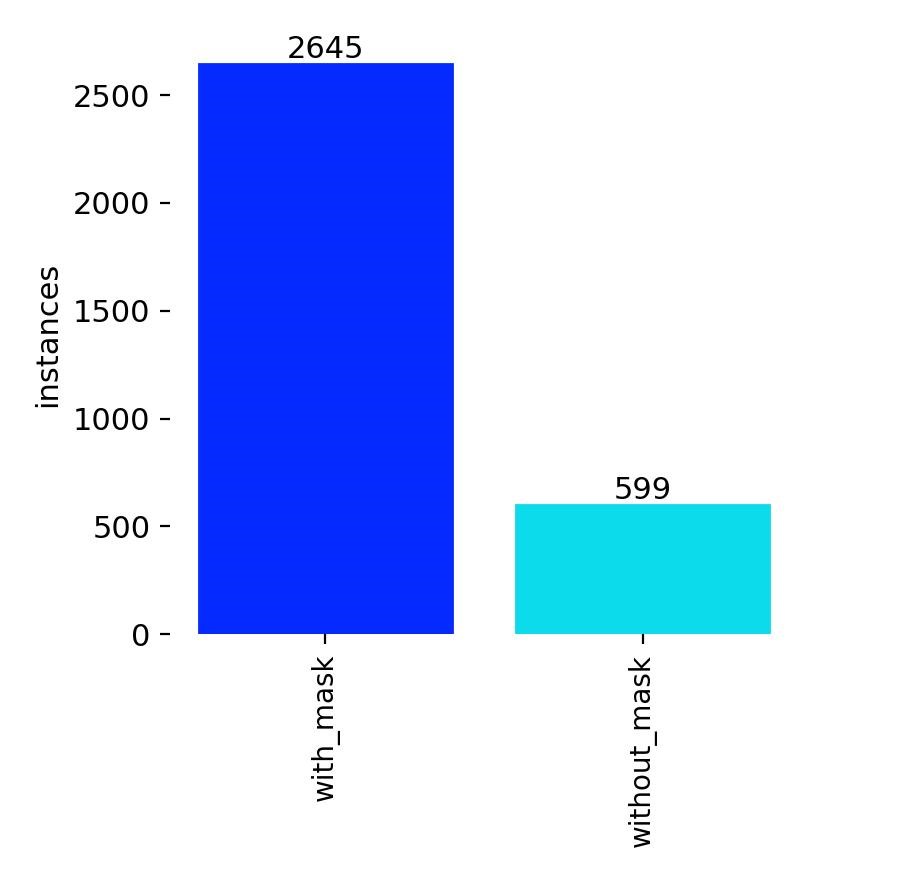


Figure 1: Image size distribution

* **Định dạng nhãn (Label):** Dữ liệu gán nhãn được cung cấp dưới dạng Pascal VOC (file XML).
  + Mỗi file XML chứa thông tin về kích thước ảnh (width, height) và danh sách các đối tượng (object).
  + Mỗi đối tượng bao gồm tên lớp (name) là with\_mask, without\_mask, mask\_weared\_incorrect và toạ độ khung bao (bndbox: xmin, ymin, xmax, ymax) (những đối tượng thuộc lớp “mask\_weared\_incorrect” sẽ bị bỏ qua)



Figure 2: labels distribution



* **Tiền xử lý dữ liệu:**
  + Chuyển đổi annotation VOC (file xml) → YOLO (file txt)

VOC format [x\_min, y\_min, x\_max, y\_max], hệ toạ độ pixel trên ảnh gốc.

Cần chuyển sang Yolo format [x\_center, y\_center, width, height]  
Công thức chuyển đổi: (với w = width, h = height)

**1. Tính tọa độ tâm box**

**2. Tính chiều rộng & chiều cao box**

* + Chuẩn hóa giá trị pixel về [0, 1]
  + Chia train, val set (tỉ lệ 80/20)

# Kiến trúc mô hình

**Base Model:** YOLO11 – công cụ phát hiện đối tượng thời gian thực, có độ chính xác, tốc độ và hiệu quả vượt trội do Ultralytics phát triển

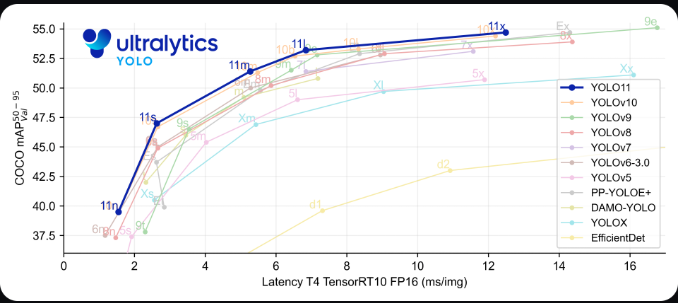


Figure 3: So sánh hiệu năng các phiên bản YOLO – Nguồn: <https://raw.githubusercontent.com/ultralytics/assets/refs/heads/main/yolo/performance-comparison.png>

**Kiến trúc mạng:**

* Backbone: gồm các khối Conv và C3k2 để trích xuất feature maps đa tầng từ ảnh input
* Neck: kết hợp khối C2PSA + các khổi C3k2 + SPFF để trộn đặc trưng, kết hợp features từ các tầng nông với các tầng sâu -> model phát hiện tốt các đối tượng ở nhiều kích thước trong ảnh.
* Head: **Decoupled detection head** tách thành 2 nhánh tích chập:
  + Nhánh **classification** dự đoán label
  + Nhánh **regression (bbox)** dự đoán toạ độ bbox

Hàm loss:

* Box loss: dùng CloU + DFL tối ưu độ chính xác của bbox
* Class loss: dùng BCE tính xác suất phân loại

Đầu ra: toạ độ (x, y, w, h) của bbox, confident\_score của lớp đối tượng. Dùng kỹ thuật NMS (Non-maximum Suppression) để loại bỏ các box trùng lặp

# Quy trình huấn luyện

Trong bài tập này, em fully fine-tune backbone YOLO11 Nano

**Thông số huấn luyện**

|  |  |
| --- | --- |
| **Tham số** | **Giá trị** |
| epochs | 100 |
| batch size | 16 |
| image size | 640 |
| learning rate | 1e-3 |
| optimizer | SGD (mặc định YOLO) |
| loss | Box + Obj + Class (YOLO Loss) |

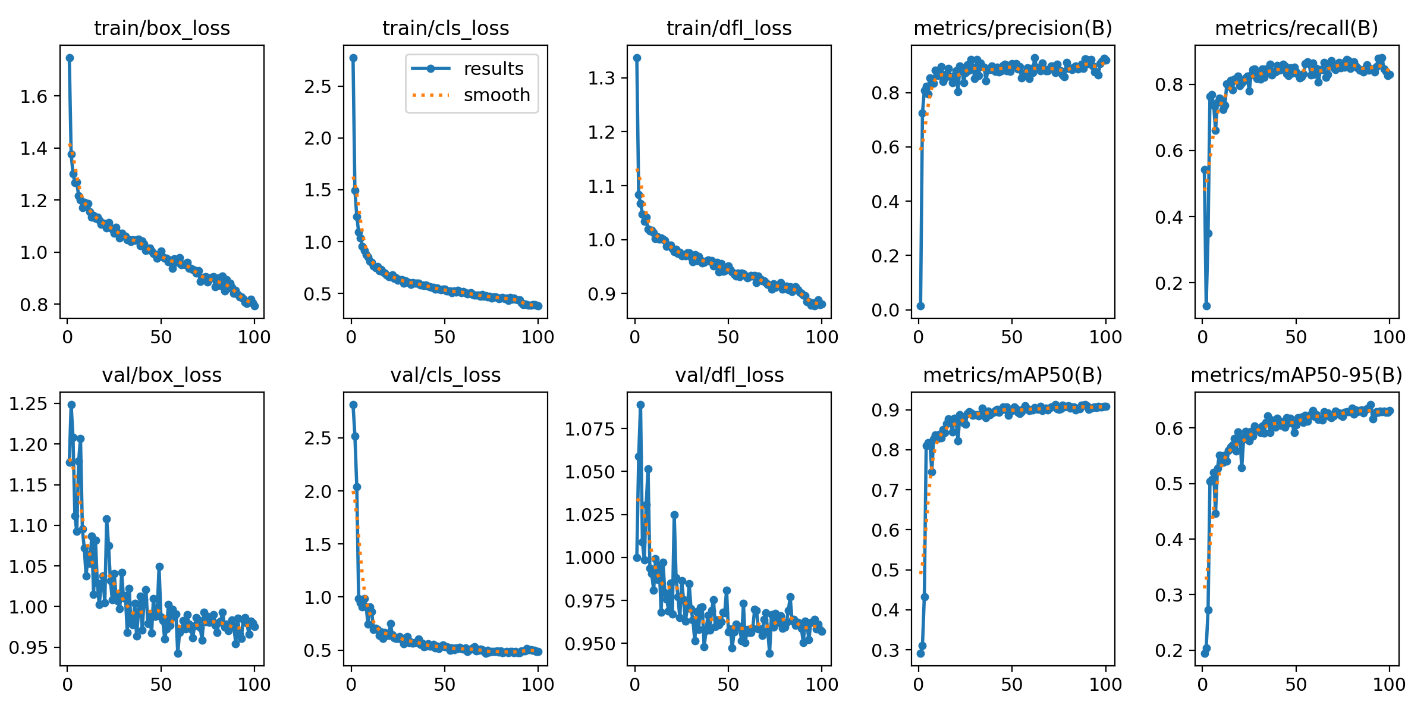
**Quá trình huấn luyện:**

sau mỗi epoch:

* **train/box\_loss** – độ mất mát dự đoán bbox
* **train/cls\_loss** – mất mát phân loại
* **val/box\_loss**, **val/cls\_loss** – kiểm định
* **metrics/precision (B)** – precision trung bình
* **metrics/recall (B)** – recall trung bình
* **metrics/mAP50** – độ chính xác ở IoU=0.5
* **metrics/mAP50-95** – trung bình IoU (0.5 → 0.95)

File log kết quả được lưu trong results.csv.

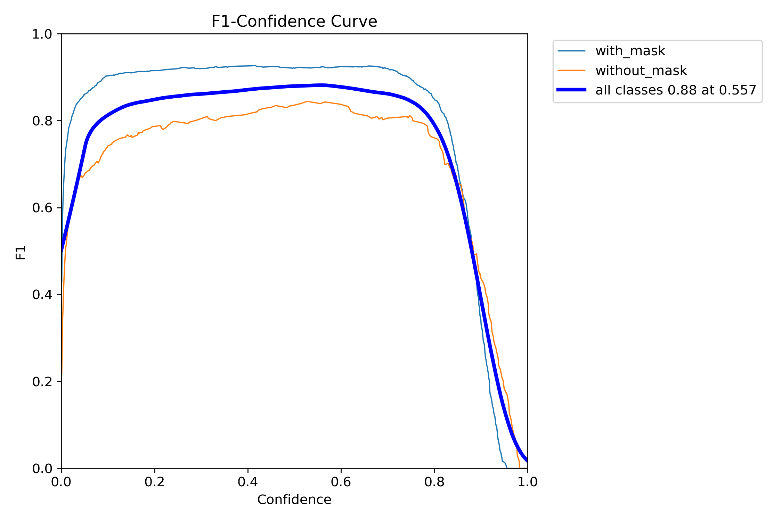
# Kết quả đánh giá



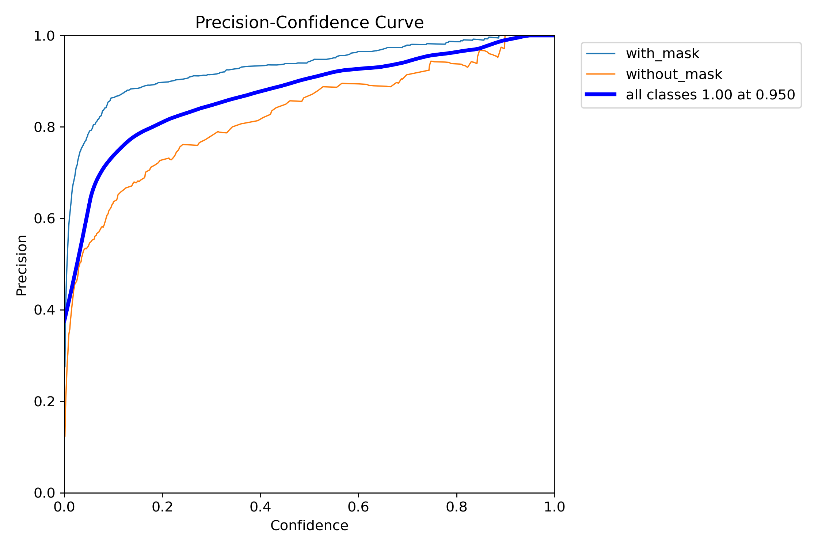
Nhận xét:

**mAP theo từng lớp**

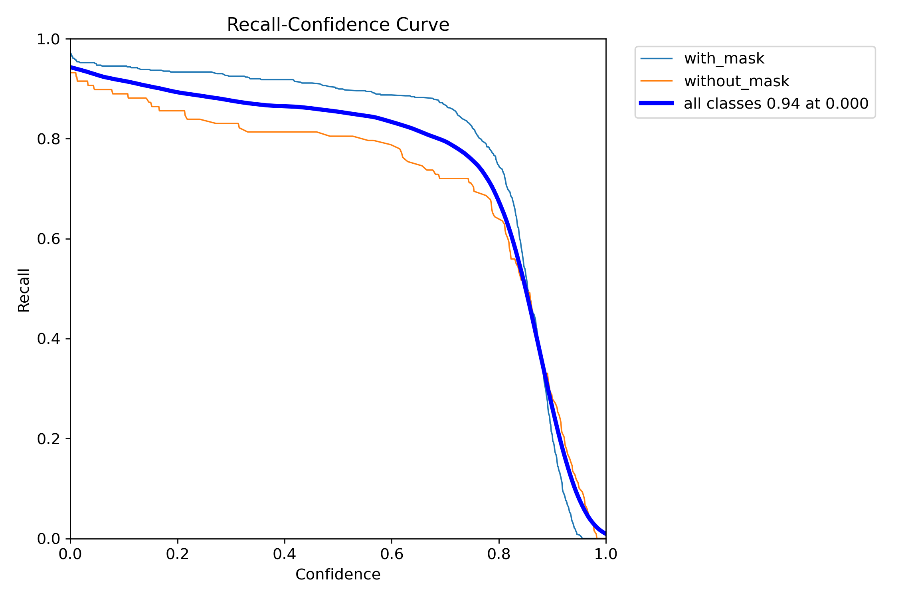
* with\_mask: **AP = 0.955**
* without\_mask: **AP = 0.865**
* mAP50 tổng: **0.910**
* mAP50–95 tổng: **≈ 0.62**
* mAP cao ~0.9 cho thấy model dự đoán tốt, mAP tổng thấp ~0.6 vì unbalanced dataset (label “without\_mask” chiếm số lượng khoảng ¼ label “with\_mask”) khiến mAP tổng bị kéo xuống



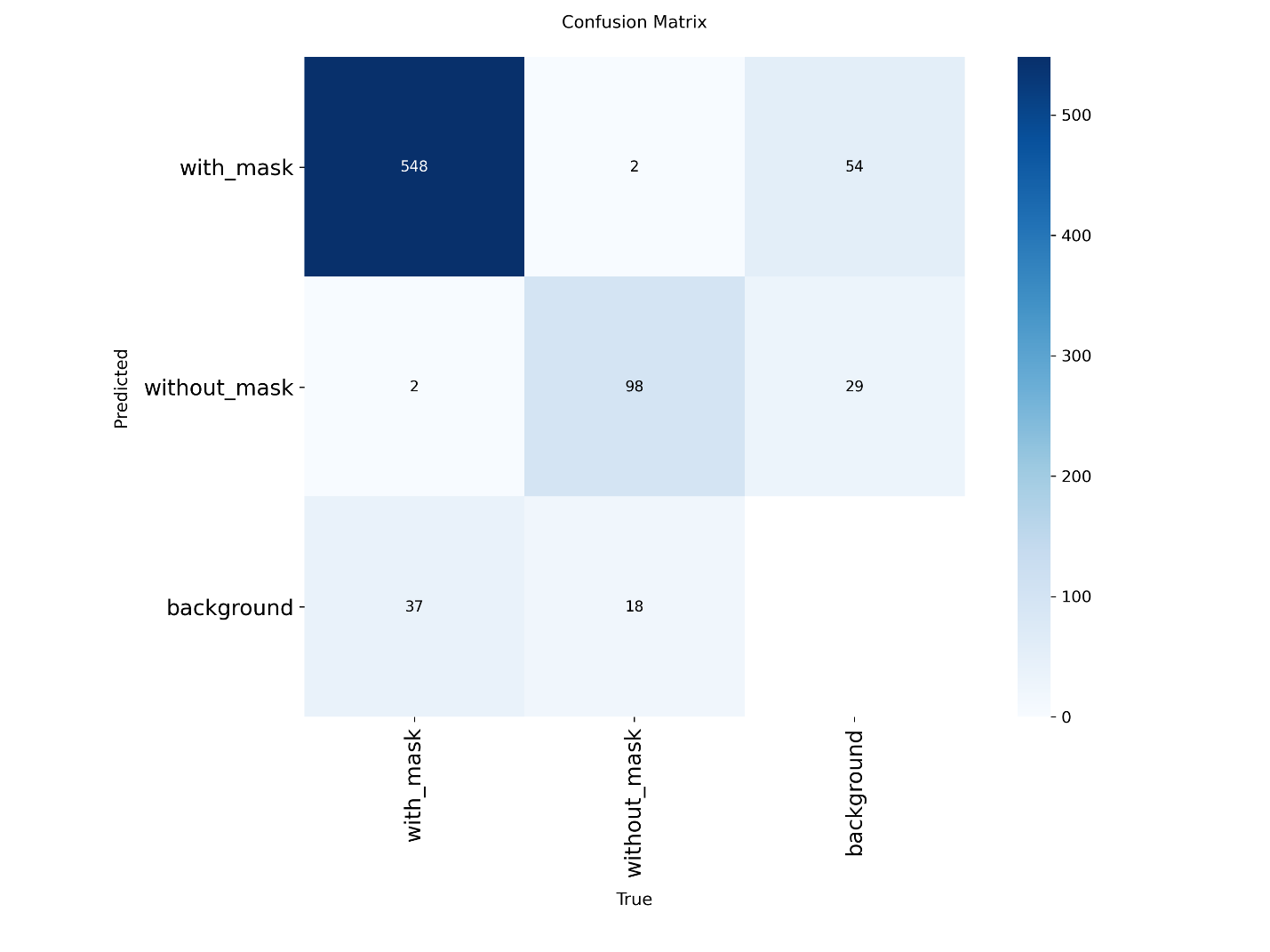
F1-score đạt giá trị cao nhất **0.88** tại ngưỡng confidence **0.557** -> cân bằng tốt giữa precision và recall.



Với conf = 0.95, precision gần như **1.0 -> dự đoán chắc chắn và chính xác**

****

Recall gần **0.94** ở conf = 0 → mô hình không bỏ sót nhiều đối tượng



Hầu hết ảnh **with\_mask** được nhận dạng đúng (548 mẫu → accuracy rất cao).

Lớp **without\_mask** bị nhầm sang with\_mask và background nhiều hơn → lý do chính:

* dataset mất cân bằng
* khuôn mặt không đeo khẩu trang có nhiều biến thể (há miệng, che nửa mặt, tối, noise…)

Tuy nhiên, model lại xác định các vùng background thành người đeo/ không đeo mask khá nhiều (54+29 = 83 bboxes)

# Hình minh họa dự đoán





Figure 4: thử nghiệm trên ảnh không nằm trong dataset

# Nhận xét & hạn chế

Model dự đoán tốt các đối tượng lớn, rõ ràng, các đối tượng lớn trung bình, nhiều đối tượng dày đặc với confidence score cao, thời gian inference nhanh (~ vài giây)

Hạn chế:

* những đối tượng có tấm bảo hộ khiến vùng mặt có nhiều điểm sáng (do ánh sáng phản xạ) thì dù cho đối tượng lớn chiếm vùng lớn trong ảnh nhưng không detect được
* model không detect được các khuôn mặt quá nhỏ nằm ở background

# Đề xuất cải thiện

Mở rộng dataset cho nhiều điều kiện khác nhau:

* ánh sáng yếu, bóng tối, ngoài trời, nhiều góc quay, hoàn cảnh, tình huống, góc nhìn,…để tăng độ tổng quát cho model
* Hiện tại đang dùng YOLOv8n (Nano) → ưu điểm nhanh, nhưng độ chính xác còn thấp, có thể thử nghiệm các backbone mạnh hơn