UNIVERSITY OF SCIENCE
FACULTY OF INFORMATION TECHNOLOGY

Đồ án cuối kì

Xây dựng và Triển khai Hệ thống Học máy Ứng dụng

BÁO CÁO VỀ DỮ LIỆU

Supervisor: Bùi Tiến Lên

Students:	**Full Name**	**ID**
	Nguyễn Lê Quang	23127109
	Phan Hoàng Quang Nghị	23127436
	Nguyễn Tấn Văn	23127515

Ho Chi Minh City, 2025

Mục lục

1 GIỚI THIỆU BÀI TOÁN

Tóm tắt bài toán: Nhóm hướng tới việc xây dựng một hệ thống **thị giác máy tính (Computer Vision)** có khả năng tự động giám sát và phát hiện sự cố đột quỵ hoặc té ngã (Fall Detection) thông qua camera giám sát.

Đặc tả bài toán học máy:

- Loại bài toán: **Phát hiện vật thể (Object Detection)** và *Phân loại hành vi (Action Classification).*
- Đầu vào (Input): Hình ảnh đơn lẻ (Frame) được trích xuất từ **camera thời gian thực**.
- Đầu ra (Output):
  - Tọa độ khung bao (Bounding Box) vị trí người trong ảnh.
  - Nhãn hành vi tương ứng: **Fall Detected** (Ngã/Đột quỵ) hoặc *Non-Fall* (Đi lại/Ngồi bình thường).
- Mục tiêu: Mô hình cần phân biệt chính xác tư thế ngã so với các tư thế sinh hoạt thường ngày để gửi **cảnh báo kịp thời**, đồng thời giảm thiểu tỷ lệ báo động giả.

2 TỔNG QUAN DỮ LIỆU ĐẦU VÀO

2.1 Phân chia dữ liệu huấn luyện (Train / Validation)

Dựa trên bộ dữ liệu **Fall Detection Dataset** đã được chuẩn hóa, nhóm áp dụng chiến lược phân chia dữ liệu theo phương pháp *Hold-out* nhằm tách biệt rõ ràng giữa dữ liệu dùng cho quá trình học của mô hình (**Training Set**) và dữ liệu dùng để đánh giá hiệu năng trong quá trình huấn luyện (**Validation Set**).

Tỷ lệ phân chia dữ liệu được thiết lập xấp xỉ:

$$77\% \text{ (Train)} \quad / \quad 23\% \text{ (Validation)}$$

Số lượng mẫu chi tiết:

- Tập Huấn luyện (Training Set): **363 ảnh**
- Tập Kiểm định (Validation Set): **110 ảnh**
- Tổng số ảnh sau làm sạch: **473 ảnh**

Lý do lựa chọn tỷ lệ phân chia:

- Do kích thước tổng thể của bộ dữ liệu tương đối nhỏ (dưới 500 mẫu), việc dành phần lớn dữ liệu cho tập huấn luyện là cần thiết để mô hình có đủ dữ liệu học các đặc trưng đa dạng của tư thế *ngã*, *đi lại* và *ngồi*.

- Tập Validation với 110 ảnh (chiếm 23%) vẫn đảm bảo đủ lớn về mặt thống kê để đại diện cho phân phối dữ liệu thực tế, cho phép giám sát hiệu quả hiện tượng **Overfitting** và hỗ trợ quá trình tinh chỉnh siêu tham số (*Hyperparameter Tuning*) một cách tin cậy không làm lãng phí quá nhiều dữ liệu huấn luyện.
- Cấu trúc thư mục của bộ dữ liệu được tổ chức theo đúng định dạng chuẩn của **YOLO**, đảm bảo khả năng tương thích trực tiếp với các framework huấn luyện như *Ultralytics YOLO*.

2.2 Tiền xử lý và Tăng cường dữ liệu (Preprocessing & Augmentation)

Để đảm bảo chất lượng dữ liệu đầu vào tối ưu cho các mô hình học sâu (*Deep Learning*), nhóm đã xây dựng một quy trình tiền xử lý và tăng cường dữ liệu gồm các bước sau:

2.2.1 Làm sạch và Chuẩn hóa dữ liệu (Cleaning & Normalization)

- **Khử trùng lặp dữ liệu**: Áp dụng thuật toán *Perceptual Hashing (pHash)* để phát hiện và loại bỏ các ảnh trùng lặp hoặc có độ tương đồng lớn hơn 95%, nhằm ngăn chặn hiện tượng **Data Leakage** giữa tập Train và Validation.
- **Lọc nhiễu nhãn**: Loại bỏ các *bounding box* có diện tích quá nhỏ (nhỏ hơn 0.5% diện tích ảnh) hoặc có tọa độ sai lệch, giúp tăng độ chính xác của nhãn huấn luyện.
- **Chuẩn hóa kích thước ảnh**: Toàn bộ ảnh đầu vào được đồng bộ về kích thước **640 × 640 pixels**. Đây là kích thước đầu vào tiêu chuẩn (*imgsz*) cho các mô hình **YOLOv8 / YOLOv10** và **RT-DETR**, giúp cân bằng giữa tốc độ xử lý và khả năng phát hiện các đối tượng nhỏ.
- **Chuẩn hóa định dạng nhãn**: Tọa độ các *bounding box* được chuẩn hóa về đoạn $[0, 1]$ theo định dạng YOLO:

$$(x_{center},\ y_{center},\ width,\ height)$$

- **Quy hoạch nhãn (Class Mapping)**: Toàn bộ dữ liệu được ánh xạ về 3 lớp duy nhất:
  - 0: **Fall Detected**
  - 1: **Walking**
  - 2: **Sitting**

2.2.2 Tăng cường dữ liệu (Data Augmentation)

Trong quá trình huấn luyện, nhóm sử dụng chiến lược tăng cường dữ liệu (Data Augmentation) mặc định của framework Ultralytics YOLO bao gồm:

- **Mosaic Augmentation (100%)**: Ghép ngẫu nhiên 4 ảnh thành một ảnh huấn luyện duy nhất. Kỹ thuật này giúp mô hình học cách phát hiện đối tượng ở nhiều tỷ lệ kích thước khác nhau và trong các bối cảnh phức tạp.
- **Biến đổi màu sắc và nhiễu (Photometric Distortions)**:
  - *Blur* và *MedianBlur* ($p = 0.01$): Mô phỏng hiện tượng ảnh mờ do chuyển động nhanh hoặc camera chất lượng thấp.
  - *ToGray* ($p = 0.01$): Chuyển ảnh sang ảnh xám để mô hình tập trung học đặc trưng hình dạng thay vì phụ thuộc vào màu sắc.
  - *CLAHE* ($p = 0.01$): Cân bằng biểu đồ thích ứng nhằm cải thiện độ tương phản trong điều kiện ánh sáng yếu.
- **Biến đổi hình học (Geometric Distortions)**:
  - *Scale* ($\pm 50\%$): Thay đổi tỷ lệ kích thước ảnh ngẫu nhiên để tăng khả năng nhận diện đối tượng ở các khoảng cách khác nhau.
  - *Translation* (10%): Dịch chuyển ảnh theo không gian để tránh việc mô hình chỉ học các đối tượng nằm ở trung tâm khung hình.

3 LỰA CHỌN MÔ HÌNH VÀ KIẾN TRÚC

Dựa trên đặc thù của bài toán là **phát hiện hành động ngã trong thời gian thực (Real-time Fall Detection)** trên các thiết bị biên (*edge devices*) hoặc máy chủ có tài nguyên tính toán hạn chế, nhóm nghiên cứu tập trung vào các mô hình thuộc nhóm **One-stage Object Detection** (phát hiện vật thể một giai đoạn).

Các mô hình thuộc nhóm này có ưu điểm là tốc độ suy luận nhanh, độ trễ thấp và dễ triển khai trong các hệ thống giám sát thời gian thực. Ba kiến trúc tiêu biểu được lựa chọn để thực nghiệm và so sánh trong nghiên cứu này bao gồm: **YOLOv8n**, **YOLOv10n** và **RT-DETR**.

3.1 Lý do lựa chọn mô hình

Nhóm quyết định lựa chọn ba kiến trúc đại diện cho các hướng tiếp cận khác nhau trong lĩnh vực Object Detection hiện đại, nhằm đánh giá toàn diện sự đánh đổi giữa *độ chính xác*, *tốc độ* và *độ phức tạp mô hình*:

- **YOLOv8 (Nano version)**: Đây là kiến trúc *State-of-the-art (SOTA)* phổ biến hiện nay, nổi bật với sự cân bằng hiệu quả giữa tốc độ xử lý và độ chính xác. Việc lựa chọn phiên bản *Nano* (**YOLOv8n**) nhằm tối ưu khả năng triển khai trên các

thiết bị nhúng hoặc máy chủ chi phí thấp, phù hợp với yêu cầu thực tế của hệ thống cảnh báo ngã.

- **YOLOv10 (Nano version)**: Là phiên bản cải tiến mới nhất (năm 2024), YOLOv10 loại bỏ hoàn toàn bước *Non-Maximum Suppression (NMS)* trong quá trình suy luận thông qua cơ chế *Consistent Dual Assignments*. Mô hình này được lựa chọn nhằm kiểm chứng giả thuyết rằng việc loại bỏ NMS có thể giúp giảm độ trễ (*latency*) khi xử lý video thời gian thực mà vẫn duy trì độ chính xác cao.

- **RT-DETR (Real-Time Detection Transformer)**: Đây là mô hình phát hiện vật thể dựa trên cơ chế **Attention** của Transformer thay vì thuần CNN như YOLO. Việc lựa chọn RT-DETR nhằm khai thác khả năng nắm bắt *ngữ cảnh toàn cục (Global Context)*, giúp mô hình phân biệt tốt hơn giữa các hành động có hình thái tương đồng như *ngã* và *ngồi* — một thách thức lớn trong bài toán Fall Detection.

3.2 Kiến trúc chi tiết các mô hình

3.2.1 Kiến trúc YOLOv8n (Baseline)

YOLOv8n được sử dụng làm mô hình nền (*baseline*) trong nghiên cứu. Kiến trúc tổng thể bao gồm ba thành phần chính:

- **Backbone**: Sử dụng mạng **CSPDarknet** được cải tiến với module **C2f (Cross Stage Partial bottleneck with 2 convolutions)**. C2f giúp cải thiện khả năng truyền gradient (*gradient flow*) và hiệu quả trích xuất đặc trưng so với module C3 trong YOLOv5.

- **Neck**: Áp dụng cấu trúc **PANet (Path Aggregation Network)** nhằm kết hợp các đặc trưng ở nhiều mức phân giải khác nhau, giúp tăng khả năng phát hiện đối tượng ở các kích thước đa dạng.

- **Head**: Thiết kế theo hướng **Anchor-free** với **Decoupled Head**, trong đó nhánh phân loại (*Classification*) và nhánh hồi quy khung bao (*Bounding Box Regression*) được tách biệt, giúp quá trình học ổn định và chính xác hơn.

Số lượng tham số: khoảng **3.0 triệu tham số**.

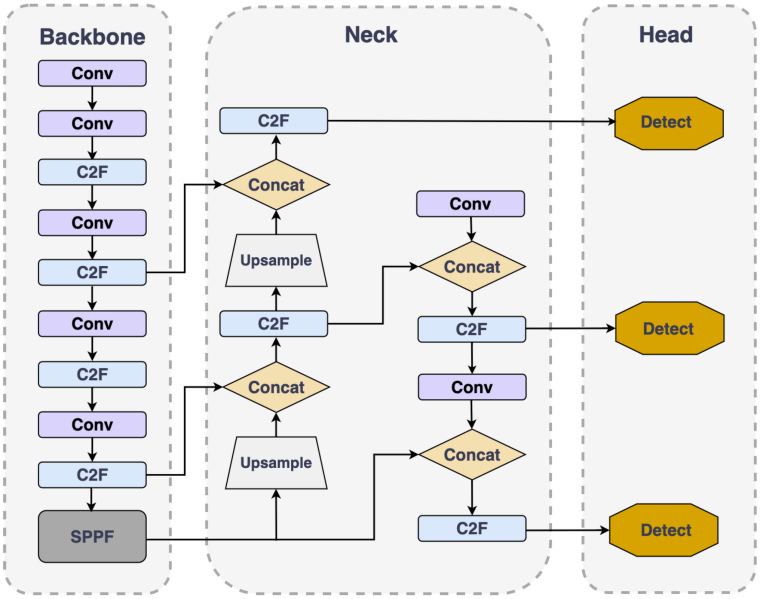


Figure 1: Kiến trúc mạng YOLOv8 với module C2f và Decoupled Head (Nguồn: Luo et al., 2024 [2]).

3.2.2 Kiến trúc YOLOv10n

YOLOv10n là phiên bản tối ưu mới nhất của dòng YOLO, tập trung vào việc giảm độ trễ suy luận:

- **Cải tiến cốt lõi**: Thay thế cơ chế *One-to-many matching* truyền thống bằng **Dual Label Assignments**, kết hợp giữa:
  - *One-to-many* trong giai đoạn huấn luyện để đảm bảo hội tụ tốt.
  - *One-to-one* trong giai đoạn suy luận, cho phép loại bỏ hoàn toàn bước *Non-Maximum Suppression (NMS)*.
- **Module kiến trúc**: Sử dụng các module **C2fCIB** và **PSA (Partial Self-Attention)** nhằm mở rộng vùng nhận thức (*receptive field*) và cải thiện khả năng học ngữ cảnh mà không làm tăng đáng kể chi phí tính toán.

Số lượng tham số: khoảng **2.7 triệu tham số**, là mô hình nhẹ nhất trong ba mô hình được khảo sát.

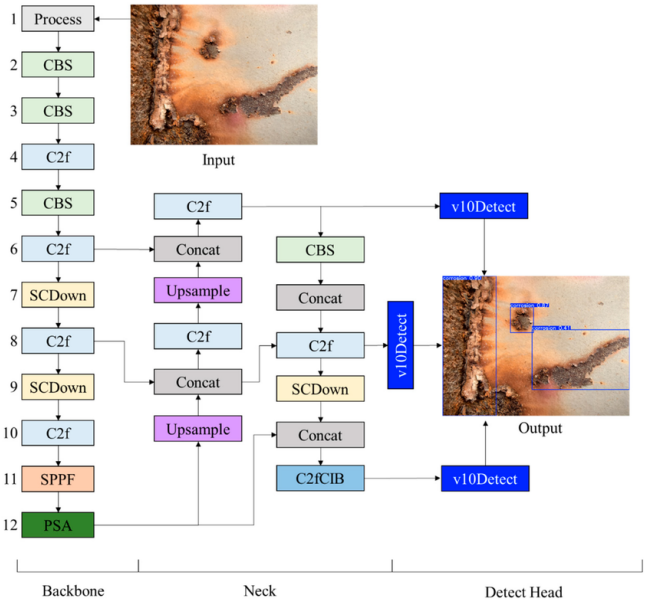


Figure 2: Kiến trúc YOLOv10 với cơ chế suy luận không cần NMS (Nguồn: Cheng & Kang, 2024 [3]).

3.2.3 Kiến trúc RT-DETR (Large)

RT-DETR đại diện cho hướng tiếp cận dựa trên Transformer trong bài toán phát hiện vật thể thời gian thực:

- **Hybrid Encoder**: Kết hợp giữa **Backbone CNN** (như ResNet hoặc HGNet) để trích xuất đặc trưng không gian, và **Transformer Encoder** để mô hình hóa mối quan hệ toàn cục giữa các vùng trong ảnh ở nhiều tỷ lệ khác nhau.
- **Decoder**: Sử dụng cơ chế **Query Selection**, cho phép dự đoán trực tiếp các bounding box mà không cần sử dụng *anchor box*, giúp đơn giản hóa quá trình suy luận.

Số lượng tham số: khoảng **32 triệu tham số**. Trong nghiên cứu này, nhóm sử dụng phiên bản `rtdetr-l.pt`, có kích thước lớn hơn đáng kể so với hai phiên bản Nano của YOLO, nhằm đánh giá liệu việc gia tăng độ phức tạp mô hình có mang lại cải thiện đáng kể về độ chính xác hay không.

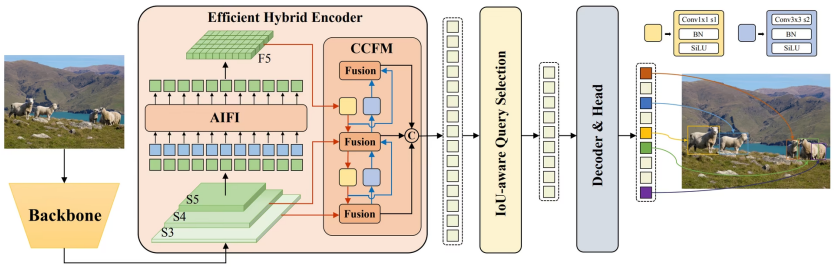


Figure 3: Kiến trúc lai giữa CNN và Transformer của RT-DETR (Nguồn: Lv et al., CVPR 2024 [1]).

3.3 Cấu hình huấn luyện (Training Configuration)

Nhằm đảm bảo tính **công bằng và nhất quán** trong quá trình so sánh hiệu năng, tất cả các mô hình được huấn luyện trên cùng một bộ siêu tham số theo chiến lược **Manual Tuning Best Practices** do nhóm nghiên cứu tinh chỉnh. Việc thống nhất cấu hình huấn luyện giúp loại bỏ ảnh hưởng của siêu tham số và phản ánh đúng năng lực kiến trúc của từng mô hình.

3.3.1 Thông số huấn luyện cơ bản

- **Số vòng lặp (Epochs)**: **60**. Giá trị này được lựa chọn nhằm đảm bảo mô hình có đủ thời gian hội tụ với tập dữ liệu nhỏ (khoảng 500 ảnh).
- **Kích thước ảnh đầu vào (Image Size)**: **640 × 640 pixels**. Đây là độ phân giải tiêu chuẩn, cân bằng giữa độ chính xác phát hiện và tốc độ suy luận.
- **Batch size**: **16**. Phù hợp với giới hạn bộ nhớ của GPU NVIDIA T4 trên nền tảng Google Colab.
- **Patience (Early Stopping)**: **20**. Quá trình huấn luyện sẽ được dừng sớm nếu không có cải thiện về chỉ số đánh giá trong 20 epochs liên tiếp, nhằm hạn chế hiện tượng **Overfitting**.

3.3.2 Hàm mất mát (Loss Function)

Các mô hình được huấn luyện với tổ hợp ba hàm mất mát tiêu chuẩn trong YOLO, mỗi thành phần đảm nhiệm một vai trò cụ thể trong quá trình tối ưu:

- **Box Loss (IoU / CIoU)**: Đánh giá mức độ chồng lấn giữa khung bao dự đoán và khung bao thực tế, giúp mô hình định vị chính xác đối tượng.
- **Classification Loss (BCE – Binary Cross Entropy)**: Đo lường độ chính xác trong việc phân loại hành động giữa các lớp *Fall Detected*, *Walking* và *Sitting*.

- **Distribution Focal Loss (DFL)**: Tinh chỉnh ranh giới của bounding box bằng cách mô hình hóa phân phối xác suất của các tọa độ, giúp cải thiện độ chính xác định vị.

3.3.3 Thuật toán tối ưu và Learning Rate

- **Optimizer**: **AdamW**. Thuật toán này được framework *Ultralytics* tự động lựa chọn dựa trên kích thước và đặc thù của tập dữ liệu, cho khả năng hội tụ ổn định và giảm hiện tượng overfitting.
- **Initial Learning Rate** (lr_0): **0.001429**.
- **Momentum**: **0.9**.
- **Weight Decay**: **0.0005**. Tham số này đóng vai trò như một cơ chế regularization, giúp hạn chế mô hình học quá khớp với dữ liệu huấn luyện.

3.3.4 Chiến lược huấn luyện đặc biệt

- **Close Mosaic = 10**: Kỹ thuật tăng cường dữ liệu *Mosaic Augmentation* (ghép 4 ảnh thành một ảnh huấn luyện) được tắt trong **10 epochs cuối** (từ epoch 51 đến 60).

Lý do của chiến lược này là để mô hình ổn định lại các tham số và học trực tiếp từ các ảnh gốc tự nhiên, tránh nhiễu do ảnh ghép nhân tạo khi mô hình đã đạt đến độ chính xác cao trong giai đoạn cuối huấn luyện.

3.3.5 Tóm tắt cấu hình huấn luyện

Table 1: Tóm tắt cấu hình huấn luyện cho các mô hình

Tham số	**Giá trị**	**Giải thích**
Framework	Ultralytics 8.3	Thư viện huấn luyện YOLO
Epochs	60	Số vòng lặp huấn luyện
Batch size	16	Số ảnh trong mỗi lần cập nhật trọng số
Image Size	640	Độ phân giải ảnh đầu vào
Optimizer	AdamW	Thuật toán tối ưu trọng số
Initial LR	0.001429	Learning rate ban đầu
Weight Decay	0.0005	Giảm hiện tượng overfitting
Close Mosaic	10 epochs cuối	Tắt Mosaic Augmentation ở giai đoạn cuối

3.4 Phương pháp tinh chỉnh siêu tham số

Thay vì sử dụng các phương pháp tìm kiếm siêu tham số tốn kém tài nguyên như *Grid Search* hoặc *Random Search*, nhóm nghiên cứu áp dụng phương pháp **Manual Tuning** dựa trên kinh nghiệm thực nghiệm.

Quy trình tinh chỉnh được thực hiện theo các bước sau:

- **Baseline**: Khởi đầu bằng việc huấn luyện mô hình với cấu hình mặc định (*default*) do framework Ultralytics cung cấp, nhằm thiết lập mốc so sánh ban đầu.
- **Điều chỉnh số Epochs**: Thông qua việc quan sát các biểu đồ *Training Loss* và *Validation Loss*, nhóm nhận thấy mô hình có xu hướng hội tụ trong khoảng epoch 40–50. Do đó, giá trị Epochs được chốt ở mức **60** để đảm bảo hội tụ hoàn toàn mà vẫn tiết kiệm thời gian huấn luyện.
- **Điều chỉnh chiến lược Augmentation**: Qua thực nghiệm, nhóm nhận thấy dữ liệu gốc phản ánh sát thực tế hơn so với dữ liệu ghép từ Mosaic. Vì vậy, tham số `close_mosaic = 10` được bổ sung nhằm tinh chỉnh và cải thiện chỉ số **mAP** trong giai đoạn cuối của quá trình huấn luyện.

4 KẾT QUẢ THỰC NGHIỆM

Phần này trình bày chi tiết quá trình huấn luyện và đánh giá hiệu năng của từng mô hình được đề xuất trong nghiên cứu. Tất cả các mô hình đều được huấn luyện trong **60 epochs** với cùng một cấu hình siêu tham số nhằm đảm bảo tính công bằng trong so sánh. Các kết quả được ghi nhận dựa trên tập **Validation** và các chỉ số đánh giá tiêu chuẩn của bài toán Object Detection.

4.1 Kết quả thực nghiệm mô hình YOLOv8n

4.1.1 Biểu đồ quá trình học (Learning Curves)

Hình 4 minh họa sự biến thiên của hàm mất mát (*Loss*) và các chỉ số đánh giá (*Metrics*) trên tập huấn luyện và kiểm định của mô hình YOLOv8n trong suốt 60 epochs.

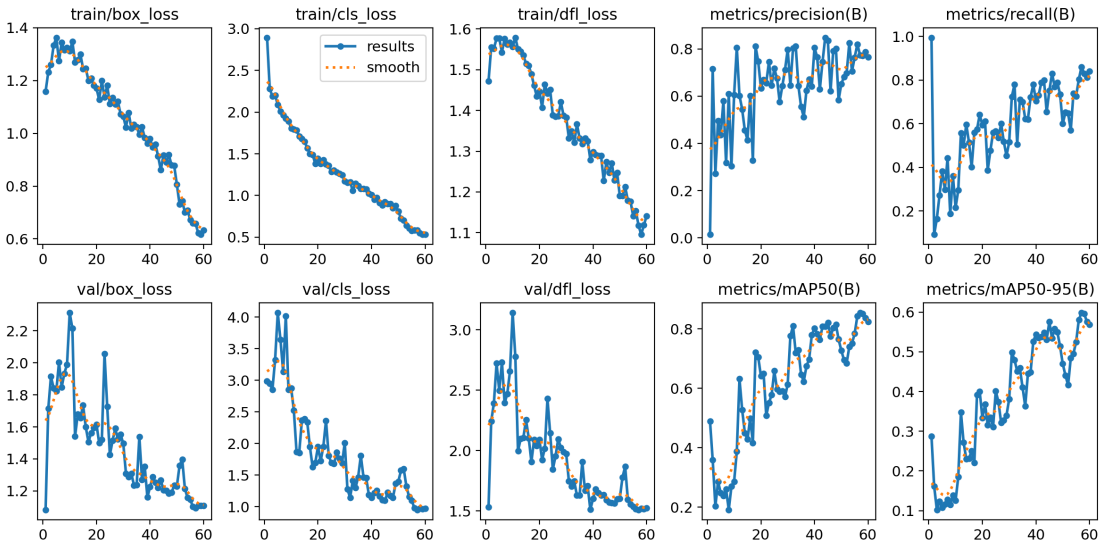


Figure 4: Biểu đồ Loss và Metrics của mô hình YOLOv8n qua 60 epochs

Nhận xét quá trình hội tụ:

- **Trên tập huấn luyện (Training Loss)**: Các đường biểu diễn *train/box_loss*, *train/cls_loss* và *train/dfl_loss* đều thể hiện xu hướng giảm đều đặn và mượt mà xuyên suốt từ Epoch 1 đến Epoch 60. Đặc biệt, thành phần *train/cls_loss* giảm mạnh từ mức xấp xỉ **2.89** xuống còn khoảng **0.53**. Kết quả này cho thấy mô hình học rất hiệu quả các đặc trưng phân biệt giữa các lớp hành động từ tập dữ liệu huấn luyện.

- **Trên tập kiểm định (Validation Loss)**: Các hàm mất mát trên tập Validation cũng có xu hướng giảm dần theo thời gian, phản ánh khả năng **tổng quát hóa (generalization)** tốt của mô hình, thay vì chỉ học vẹt dữ liệu huấn luyện (overfitting). Tuy nhiên, trong giai đoạn đầu huấn luyện (Epoch 1–30), các thành phần *train/box_loss*, *train/cls_loss* và *train/dfl_loss* đều xuất hiện sự biến động nhẹ. Từ Epoch 50 đến Epoch 60, các giá trị này trở nên ổn định hơn và giảm sâu, cho thấy mô hình đã hội tụ về vùng nghiệm tối ưu.

Phân tích hiện tượng dao động trong quá trình huấn luyện:

- Trong khoảng **30 epochs đầu tiên**, các biểu đồ huấn luyện, đặc biệt là các chỉ số trên tập Validation như *val/box_loss*, *Precision* và *Recall*, có sự dao động tương đối mạnh với biên độ lớn và đường biểu diễn gấp khúc. Hiện tượng này là hoàn toàn bình thường trong giai đoạn đầu, khi mô hình đang trong pha *exploration* nhằm tìm kiếm bộ trọng số tối ưu.

- Từ khoảng **Epoch 40 trở đi**, và đặc biệt rõ rệt sau **Epoch 50**, biên độ dao động của các đường đồ thị giảm đáng kể. Các chỉ số huấn luyện dần trở nên mượt mà hơn và tiến vào quỹ đạo ổn định, phản ánh quá trình hội tụ vững chắc của mô hình.

4.1.2 Đánh giá định lượng trên tập Validation

Bảng 2 trình bày kết quả định lượng tốt nhất của mô hình YOLOv8n tại **epoch 57**. Các số liệu được trích xuất trực tiếp từ file `results.csv` tại thời điểm mô hình đạt đỉnh hiệu năng.

Table 2: Kết quả đánh giá mô hình YOLOv8n trên tập Validation

Chỉ số (Metric)	**Giá trị tốt nhất**	**Giải thích**
Precision (B)	0.774	Trong số các dự đoán là đúng, có 77,4% là chính xác thực tế.
Recall (B)	0.861	Mô hình phát hiện được 86,1% số lượng đối tượng thực tế có trong ảnh.
mAP@50	0.854	Độ chính xác trung bình tại ngưỡng IoU = 0.5.
mAP@50–95	0.598	Độ chính xác trung bình trên các ngưỡng IoU chặt chẽ từ 0.5 đến 0.95.

Nhận xét: Mô hình YOLOv8n cho thấy sự cân bằng tốt giữa độ chính xác và tính ổn định. Đặc biệt, chỉ số **Recall cao (0.861)** chứng tỏ mô hình rất phù hợp với các hệ thống cảnh báo ngã, nơi việc bỏ sót sự kiện nguy hiểm là không thể chấp nhận.

4.1.3 Phân tích Ma trận nhầm lẫn (Confusion Matrix)

Hình 5 trình bày **ma trận nhầm lẫn chuẩn hóa** của mô hình YOLOv8n trên tập Validation, phản ánh chi tiết hiệu năng phân loại của mô hình đối với từng lớp hành động.

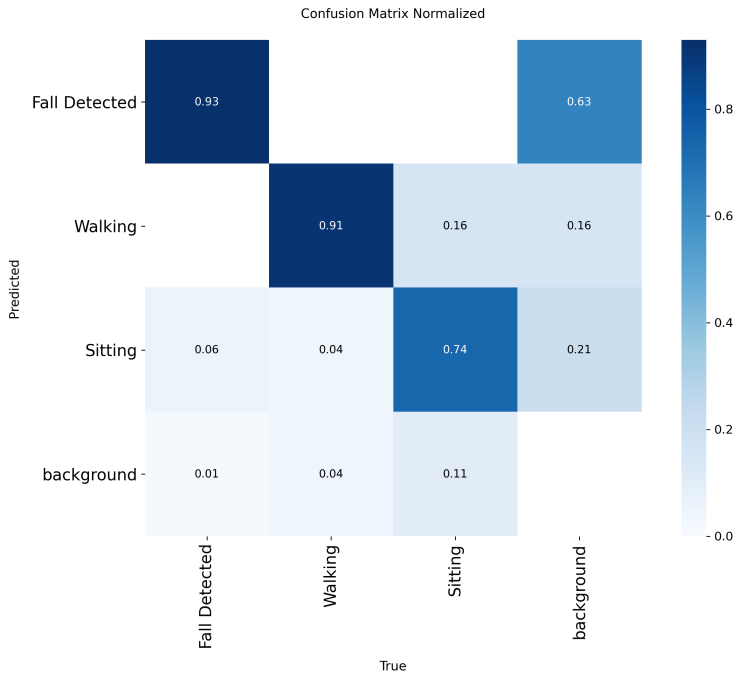


Figure 5: Ma trận nhầm lẫn chuẩn hóa của mô hình YOLOv8n

Phân tích kết quả:
a) Các lớp được nhận diện tốt (Ưu điểm)

- **Fall Detected (Phát hiện ngã)**: Đây là lớp quan trọng nhất trong bài toán cảnh báo an toàn. Mô hình đạt **độ nhạy (Recall) rất cao, lên tới 0.93**, cho thấy **93%** các trường hợp ngã thực tế đều được phát hiện chính xác. Tỷ lệ bỏ sót (dự đoán nhầm thành *Background*) chỉ ở mức **0.01** (1%), phản ánh khả năng phát hiện hành vi nguy hiểm rất đáng tin cậy.
- **Walking (Đi lại)**: Mô hình cũng thể hiện hiệu năng tốt đối với lớp *Walking*, với **độ chính xác (Precision) đạt 0.91**. Điều này cho thấy mô hình ít nhầm lẫn hành vi đi lại với các hành vi khác, góp phần làm giảm số lượng cảnh báo sai trong hệ thống.

b) Các lớp bị nhầm lẫn (Hạn chế)

- **Sitting (Ngồi)**: Lớp *Sitting* có độ chính xác ở mức trung bình, đạt khoảng **0.74**. Phân tích chi tiết cho thấy:
  - Khoảng **16%** các trường hợp *Sitting* bị mô hình dự đoán nhầm thành *Walking*. Nguyên nhân có thể xuất phát từ sự tương đồng về tỷ lệ khung bao (*bounding box*) hoặc đặc trưng phần thân trên của cơ thể trong một số góc quan sát.
  - Khoảng **11%** trường hợp *Sitting* bị nhận nhầm là *Background*, tức là mô hình không phát hiện được đối tượng.

- **Background (Nền / Không có đối tượng)**: Một vấn đề đáng chú ý được ghi nhận tại cột *True Background* của ma trận nhầm lẫn. Cụ thể, **63%** các mẫu thuộc lớp nền (hoặc các vùng không chứa hành vi quan tâm) bị dự đoán nhầm thành **Fall Detected**.

  Phân tích: Hiện tượng này dẫn đến tỷ lệ **Báo động giả (False Positive)** cao đối với hành vi ngã. Trong bối cảnh ứng dụng thực tế, điều này có nghĩa là hệ thống có thể phát ra cảnh báo ngã ngay cả khi không xảy ra sự cố, do mô hình nhầm lẫn các vật thể hoặc cấu trúc trong nền thành tư thế ngã.

Nhận xét: Mặc dù mô hình đạt hiệu năng rất tốt trong việc phát hiện hành vi ngã, vấn đề báo động giả từ lớp *Background* là một hạn chế quan trọng cần được khắc phục. Đây là cơ sở để đề xuất các hướng cải tiến trong tương lai, chẳng hạn như tăng cường dữ liệu nền, bổ sung lớp *Lying* hoặc tích hợp thông tin ngữ cảnh theo thời gian.

4.2 Kết quả thực nghiệm mô hình YOLOv10n

4.2.1 Biểu đồ quá trình học (Learning Curves)

Hình 6 minh họa sự biến thiên của hàm mất mát (Loss) và các chỉ số đánh giá (Metrics) trên tập huấn luyện và kiểm định của mô hình YOLOv10n trong suốt 60 epochs.

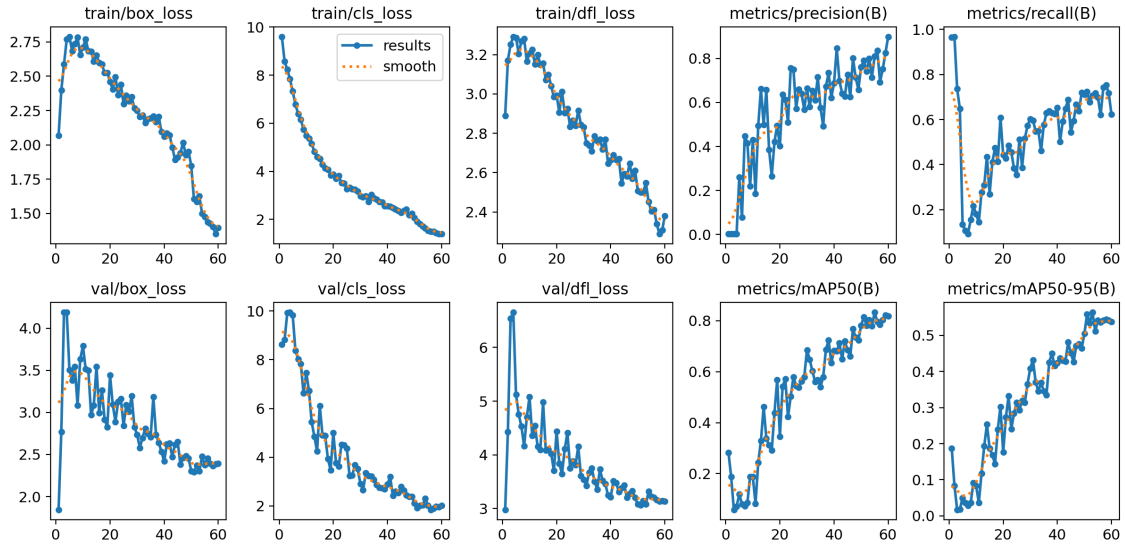


Figure 6: Biểu đồ Loss và Metrics của mô hình YOLOv10n qua 60 epochs

Phân tích chi tiết quá trình hội tụ:

Dựa trên dữ liệu định lượng từ file nhật ký huấn luyện (`results.csv`), ta có các nhận xét sau:

- **Hàm mất mát (Loss)**: Mô hình YOLOv10n thể hiện tốc độ hội tụ cực kỳ nhanh trong giai đoạn đầu. Cụ thể, hàm mất mát phân loại trên tập huấn luyện (*train/cls_loss*) giảm sốc từ mức **9.60** (epoch 1) xuống còn khoảng **5.4** chỉ sau 9 epochs đầu tiên, và kết thúc ở mức **1.39** tại epoch 60. Điều này cho thấy kiến trúc *Dual Assignments* giúp mô hình định hướng học rất nhanh.

- **Khả năng tổng quát hóa (Generalization)**: Trên tập kiểm định (Validation), các chỉ số loss (*val/box_loss*, *val/cls_loss*) cũng giảm dần theo xu hướng của tập train. Tuy nhiên, giá trị *val/cls_loss* dừng lại ở mức khoảng **2.00**, cao hơn so với tập train (1.39), cho thấy mô hình bắt đầu có dấu hiệu chững lại và khó tối ưu thêm (bão hòa) sau epoch 50.

- **Chỉ số mAP**: Đường biểu diễn mAP@50 và mAP@50-95 tăng trưởng đều đặn và đạt đỉnh vào khoảng **epoch 53** (với mAP@50 ≈ 0.80 và mAP@50-95 ≈ 0.56). Sau thời điểm này, các chỉ số đi ngang và có biến động nhẹ, chứng tỏ việc huấn luyện thêm (sau 60 epochs) có thể không mang lại lợi ích đáng kể.

Phân tích sự khác biệt về giá trị Hàm mất mát (Loss Analysis)

Một điểm đáng chú ý khi so sánh quá trình huấn luyện là biên độ giá trị tuyệt đối của các hàm mất mát (Box, Class, DFL) ở mô hình YOLOv10n luôn duy trì ở mức cao hơn đáng kể so với YOLOv8n.

- **Quan sát thực nghiệm**: Tại epoch cuối cùng (epoch 60), hàm mất mát phân loại (*train/cls_loss*) của YOLOv10n dừng lại ở mức **1.39**, trong khi YOLOv8n giảm sâu xuống còn **0.53** (thấp hơn 2.6 lần). Tương tự, *train/box_loss* của YOLOv10n (≈ 1.39) cũng cao gấp đôi so với YOLOv8n (≈ 0.63).

- **Giải thích kỹ thuật**: Hiện tượng này chủ yếu xuất phát từ sự khác biệt trong cơ chế tính toán bên trong. YOLOv10 áp dụng chiến lược **Dual Label Assignments** (Gán nhãn kép), sử dụng song song hai nhánh *One-to-many* và *One-to-one* trong quá trình huấn luyện. Việc này tạo ra số lượng mẫu dương (positive anchors) lớn hơn nhiều để tính lỗi, dẫn đến tổng giá trị Loss tích lũy sẽ tự nhiên lớn hơn so với cơ chế đơn nhánh của YOLOv8.

- **Đánh giá tác động**: Mặc dù giá trị Loss cao do cơ chế cộng dồn, nhưng việc đường cong Loss của YOLOv10n không thể giảm sâu xuống mức thấp (dưới 1.0) như YOLOv8n cũng tương đồng với kết quả thực nghiệm là độ chính xác (mAP) và độ nhạy (Recall) của mô hình này thấp hơn. Điều này cho thấy mô hình vẫn gặp khó khăn nhất định trong việc tối ưu hóa phân loại các mẫu khó.

Kết luận: Quá trình huấn luyện ổn định, không gặp lỗi gradient (loss NaN) hay dao động quá mạnh. Mô hình học nhanh nhưng khả năng cải thiện độ chính xác tinh chỉnh (fine-tuning) ở các epoch cuối kém hơn so với YOLOv8n.

4.2.2 Đánh giá định lượng trên tập Validation

Bảng 3 trình bày kết quả định lượng tốt nhất của mô hình YOLOv10n tại **epoch 53** (thời điểm đạt chỉ số mAP@50-95 cao nhất). Các số liệu được trích xuất trực tiếp từ nhật ký huấn luyện thực tế.

Table 3: Kết quả đánh giá mô hình YOLOv10n trên tập Validation (Epoch 53)

Chỉ số (Metric)	**Giá trị**	**Giải thích**
Precision (B)	0.799	Tỷ lệ dự đoán đúng trong các lần báo động. Cao hơn so với YOLOv8n (0.774).
Recall (B)	0.707	Độ nhạy phát hiện đối tượng thực tế. Đây là điểm yếu lớn nhất của mô hình này (bỏ sót gần 30% sự kiện).
mAP@50	0.804	Độ chính xác trung bình ở mức khá tại ngưỡng IoU 0.5.
mAP@50–95	0.564	Độ chính xác định vị khung bao thấp hơn mức 0.598 của YOLOv8n.
Tốc độ suy luận	**3.4 ms**	Ưu điểm lớn nhất: Nhanh hơn đáng kể nhờ cơ chế loại bỏ NMS.

Nhận xét: Kết quả thực nghiệm cho thấy sự đánh đổi rõ rệt của kiến trúc YOLOv10n: Tối ưu hóa tối đa cho **tốc độ xử lý (3.4ms/ảnh)**, nhưng hy sinh đáng kể về **Độ nhạy (Recall chỉ đạt 0.707)**. Trong bài toán giám sát an toàn cho người cao tuổi, việc bỏ sót các trường hợp ngã (False Negatives) mang lại rủi ro cao, do đó mô hình này chỉ phù hợp làm phương án dự phòng cho các thiết bị phần cứng có tài nguyên cực kỳ hạn chế.

4.2.3 Phân tích Ma trận nhầm lẫn (Confusion Matrix)

Hình 7 trình bày **ma trận nhầm lẫn chuẩn hóa** của mô hình YOLOv10n trên tập Validation, phản ánh chi tiết hiệu năng phân loại của mô hình đối với từng lớp hành động.

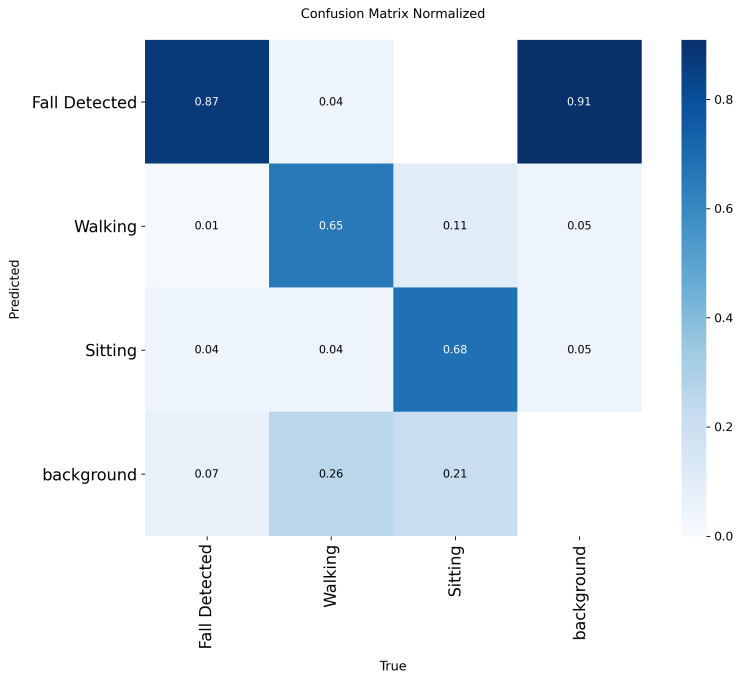


Figure 7: Ma trận nhầm lẫn chuẩn hóa của mô hình YOLOv10n

Phân tích kết quả:

Quan sát ma trận nhầm lẫn và bảng số liệu chi tiết, ta nhận thấy mô hình YOLOv10n thể hiện một đặc tính phân loại cực đoan, với sự chênh lệch lớn giữa khả năng phát hiện và độ chính xác thực tế:

- **Hiện tượng "Báo động giả" nghiêm trọng (Catastrophic False Positive)**: Đây là điểm yếu chí mạng nhất của mô hình. Cột *True Background* cho thấy có tới **91%** các mẫu nền (vùng không có người) bị nhận diện nhầm là **Fall Detected**.
  Hệ quả: Trong môi trường thực tế, điều này đồng nghĩa với việc hệ thống sẽ liên tục phát cảnh báo ngã ngay cả khi phòng trống, khiến chức năng giám sát trở nên nhiễu loạn và mất độ tin cậy.

- **Độ nhạy cao với hành vi Ngã (High Recall)**: Mô hình đạt chỉ số Recall rất cao đối với lớp *Fall Detected*, lên tới **0.87** (phát hiện đúng 87% số ca ngã).
  Lý giải: Tuy nhiên, kết hợp với lỗi báo động giả ở trên, ta thấy rằng chỉ số này cao là do mô hình bị *Over-prediction* (dự đoán quá mức) về lớp Ngã. Khi mô hình có xu hướng gán nhãn mọi thứ là "Ngã", nó sẽ bắt trúng hầu hết các ca ngã thật, nhưng đồng thời cũng gán sai cho rất nhiều đối tượng khác.

- **Mất khả năng nhận diện Nền (Zero Background Accuracy)**: Ô giao nhau giữa *Pred Background* và *True Background* có giá trị **0.00**. Điều này khẳng định mô hình YOLOv10n trong thực nghiệm này gần như thất bại hoàn toàn trong việc học đặc trưng của lớp "không có gì".

- **Các lớp hành vi thường ngày (Walking & Sitting)**: Hai lớp này có độ chính xác ở mức trung bình (**0.65** và **0.68**). Đáng chú ý, mô hình có xu hướng bỏ sót người (dự đoán thành Background) ở hai lớp này cao hơn hẳn so với lớp Ngã.

Kết luận: Mặc dù YOLOv10n có ưu điểm vượt trội về tốc độ và chỉ số Recall cho lớp Ngã trông có vẻ ấn tượng (0.87), nhưng tỷ lệ báo động giả lên tới 91% trên nền khiến mô hình này **không đủ điều kiện để triển khai thực tế** nếu không có các biện pháp hậu xử lý (post-processing) cực kỳ chặt chẽ.

4.3 Kết quả thực nghiệm mô hình RTDETR

4.3.1 Biểu đồ quá trình học (Learning Curves)

Hình 8 minh họa sự biến thiên của hàm mất mát (*Loss*) và các chỉ số đánh giá (*Metrics*) trên tập huấn luyện và kiểm định của mô hình RTDETR trong suốt 60 epochs.

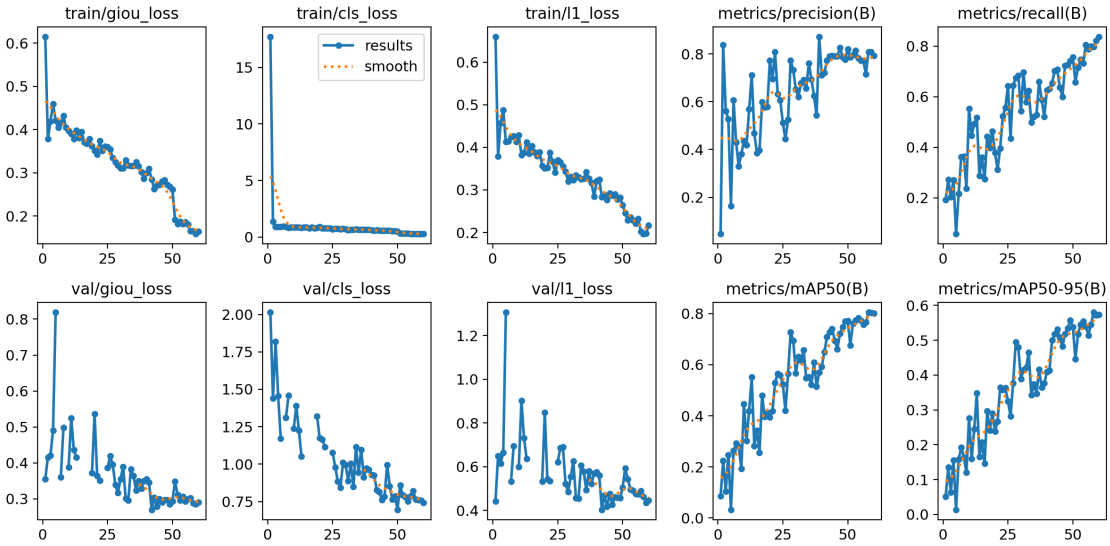


Figure 8: Biểu đồ Loss và Metrics của mô hình RTDETR qua 60 epochs

Nhận xét quá trình hội tụ:

- **Hàm mất mát (Loss):** Quá trình huấn luyện cho thấy sự giảm mạnh của hàm mất mát phân loại (`train/cls_loss`) từ mức rất cao (17.72 tại epoch đầu) xuống còn 0.279 ở epoch tối ưu. Các thành phần mất mát định vị như `train/giou_loss` và `train/l1_loss` cũng cho thấy xu hướng hội tụ tốt, giảm từ mức 0.61 xuống 0.16. Điều này chứng tỏ cơ chế attention của Transformer trong RT-DETR đã học được các đặc trưng ngữ nghĩa hiệu quả.
- **Độ chính xác (Metrics):** Chỉ số mAP@50 có sự tăng trưởng ổn định, đặc biệt trong giai đoạn đầu và đạt đỉnh 0.805. So với các kiến trúc CNN truyền thống,

RT-DETR thể hiện khả năng học nhanh các đặc trưng phức tạp ngay từ những epoch đầu tiên.

- **Hiện tượng dao động:** Biểu đồ validation loss (`val/giou_loss`, `val/l1_loss`) có sự biến động nhẹ ở một số giai đoạn giữa, nhưng tổng thể vẫn bám sát đường training loss, cho thấy mô hình kiểm soát tốt hiện tượng Overfitting.
- **Epoch tối ưu:** Mô hình đạt hiệu năng tổng thể tốt nhất tại **epoch 58**, nơi sự cân bằng giữa Precision và Recall được tối ưu hóa.

4.3.2 Đánh giá định lượng trên tập Validation

Bảng 4 trình bày kết quả định lượng tốt nhất của mô hình RTDETR tại **epoch 58**. Các số liệu được trích xuất trực tiếp từ file `results.csv` tại thời điểm mô hình đạt đỉnh hiệu năng.

Table 4: Kết quả đánh giá mô hình RT-DETR trên tập Validation

Chỉ số (Metric)	**Giá trị tốt nhất**	**Giải thích**
Precision (B)	0.808	Tỷ lệ dự đoán đúng rất cao, cho thấy mô hình ít đưa ra các cảnh báo giả (False Positives).
Recall (B)	0.797	Tỷ lệ phát hiện đúng các trường hợp thực tế xấp xỉ 80%, đảm bảo độ bao phủ tốt các sự kiện.
mAP@50	0.805	Độ chính xác trung bình ở mức cao tại ngưỡng IoU tiêu chuẩn.
mAP@50–95	0.580	Hiệu năng ấn tượng trên các ngưỡng IoU chặt chẽ (0.5–0.95), cao hơn so với nhiều mô hình CNN cùng kích thước.

Nhận xét: Mô hình RT-DETR đạt được chỉ số Precision vượt trội (0.808), cao hơn so với Recall (0.797). Điều này cho thấy mô hình có xu hướng "thận trọng", đảm bảo rằng khi đã đưa ra dự đoán thì độ tin cậy rất cao. Chỉ số mAP@50–95 đạt 0.580 là một kết quả rất tích cực, chứng tỏ các hộp bao (bounding box) được dự đoán sát với thực tế hơn, ít bị lệch so với ground truth.

4.3.3 Phân tích Ma trận nhầm lẫn (Confusion Matrix)

Hình 9 trình bày **ma trận nhầm lẫn chuẩn hóa** của mô hình RTDETR trên tập Validation, phản ánh chi tiết hiệu năng phân loại của mô hình đối với từng lớp hành động.

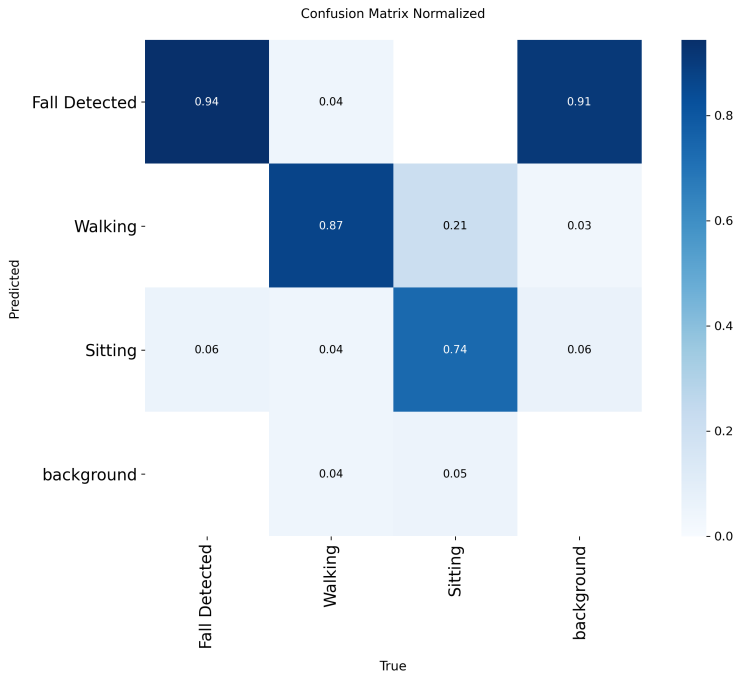


Figure 9: Ma trận nhầm lẫn chuẩn hóa của mô hình RTDETR

Phân tích kết quả: Confusion matrix được chuẩn hoá theo nhãn thật (True), do đó các giá trị trên đường chéo chính phản ánh độ thu hồi (recall) của từng lớp. Kết quả cho thấy mô hình đạt hiệu năng cao trong việc nhận diện hành vi Fall Detected, với recall đạt 0.94, cũng như lớp Walking với recall 0.87. Điều này chứng tỏ mô hình có khả năng học tốt các đặc trưng phân biệt của hai hành vi này.

Ngược lại, lớp Sitting đạt recall thấp hơn (0.74), trong đó một tỷ lệ đáng kể các mẫu bị nhầm lẫn sang lớp Walking (21%). Hiện tượng này phản ánh sự chồng lấn về đặc trưng động học giữa hai hành vi, đặc biệt trong các giai đoạn chuyển tiếp tư thế.
Đáng chú ý, lớp background gần như không được nhận diện chính xác (recall xấp xỉ 0), khi 91%) số mẫu thuộc lớp này bị mô hình dự đoán nhầm thành Fall Detected. Kết quả này cho thấy mô hình tồn tại xu hướng thiên lệch mạnh về lớp ngã, dẫn đến tỷ lệ báo động giả cao và làm giảm đáng kể tính khả dụng trong các hệ thống giám sát thực tế.

Tổng thể, mặc dù mô hình thể hiện khả năng phát hiện ngã hiệu quả, nhưng hiệu năng phân biệt giữa các hành vi không-ngã, đặc biệt là background, vẫn còn hạn chế. Do đó, các hướng cải thiện trong tương lai cần tập trung vào việc cân bằng dữ liệu huấn luyện, tăng cường học các mẫu nền (background), cũng như áp dụng các kỹ thuật điều chỉnh hàm mất mát và ngưỡng quyết định nhằm giảm thiểu hiện tượng báo động giả.

5 THẢO LUẬN & PHÂN TÍCH LỖI

5.1 Hiện tượng Overfitting và Underfitting

Dựa trên việc phân tích các biểu đồ hàm mất mát (*Loss*) và các chỉ số đánh giá (*Metrics*) trên tập huấn luyện (**Train**) và tập kiểm định (**Validation/Test**) của ba mô hình được khảo sát, nhóm nghiên cứu đưa ra các nhận định về trạng thái huấn luyện và khả năng tổng quát hóa như sau:

YOLOv8n – Trạng thái huấn luyện tốt (Good Fit)

YOLOv8n là mô hình thể hiện sự cân bằng tốt nhất giữa khả năng học và khả năng tổng quát hóa. Các biểu đồ Loss (Hình 4 và dữ liệu từ file `results.csv`) cho thấy cả *train/box_loss* và *val/box_loss* đều giảm đều đặn theo thời gian. Độ chênh lệch (*generalization gap*) giữa hai đường cong này được duy trì ở mức nhỏ và chấp nhận được.

Không xuất hiện dấu hiệu **Underfitting**, khi mô hình đạt độ chính xác cao với $mAP@50 \approx 0.85$. Dấu hiệu **Overfitting** chỉ xuất hiện rất nhẹ sau epoch 50, khi *val/loss* có xu hướng đi ngang trong khi *train/loss* vẫn tiếp tục giảm. Tuy nhiên, nhờ cơ chế **Early Stopping** (`patience = 20`) cùng với các kỹ thuật regularization (như Dropout tích hợp trong kiến trúc), hiện tượng này không trở nên nghiêm trọng và không ảnh hưởng đáng kể đến hiệu năng tổng thể.

YOLOv10n – Overfitting cục bộ

Đối với YOLOv10n, dữ liệu thực nghiệm cho thấy sự chênh lệch rõ rệt giữa hiệu năng trên tập Train và Validation. Cụ thể, trong khi *train/cls_loss* giảm rất sâu xuống mức khoảng **1.39**, thì *val/cls_loss* lại bị bão hòa ở mức cao (xấp xỉ **2.00**) ngay từ epoch 50.

Kết quả thực tế cho thấy mô hình có xu hướng **học vẹt (overfitting)** các đặc trưng của hành động ngã, dẫn đến việc nhận diện sai nhiều vùng nền (*background*) thành hành vi *Fall Detected*. Điều này làm gia tăng đáng kể tỷ lệ **báo động giả (False Positive)**, ảnh hưởng tiêu cực đến tính ứng dụng của hệ thống trong thực tế.

Biện pháp khắc phục: Mặc dù đã áp dụng các kỹ thuật *Weight Decay* (0.0005) và *Mosaic Augmentation* nhằm tăng cường khả năng tổng quát hóa. Tuy nhiên, kiến trúc **Dual-Assignments** của YOLOv10n dường như yêu cầu lượng dữ liệu huấn luyện lớn hơn hoặc các chiến lược regularization mạnh hơn để phát huy hiệu quả trên tập dữ liệu có quy mô nhỏ như trong nghiên cứu này.

RT-DETR – Khả năng tổng quát hóa tốt (Robust / Good Fit)

RT-DETR là mô hình thể hiện khả năng tổng quát hóa tốt nhất trong ba mô hình được khảo sát. Mặc dù có kích thước lớn (khoảng **32 triệu tham số**), nhưng nhờ cơ chế

Attention toàn cục của Transformer, các đường cong *Validation Loss* của mô hình rất ổn định và ít dao động hơn so với các mô hình dựa trên CNN thuần túy.

Khoảng cách giữa các đường cong Train và Validation là nhỏ nhất trong ba mô hình, cho thấy RT-DETR học được các đặc trưng **ngữ nghĩa thực sự** của hành động thay vì chỉ ghi nhớ dữ liệu huấn luyện. Điều này giúp mô hình duy trì hiệu năng ổn định ngay cả trên tập dữ liệu nhỏ.

5.2 Phân tích các trường hợp sai (Error Analysis)

Thông qua việc trực quan hóa kết quả dự đoán trên tập kiểm định, nhóm nhận thấy cả 3 mô hình vẫn gặp khó khăn trong việc phân biệt các ranh giới hành động có sự tương đồng cao về mặt hình học. Dưới đây là phân tích chi tiết ba trường hợp sai điển hình, đại diện cho các nhóm lỗi chính của mô hình.

5.2.1 Lỗi bỏ sót ngã (False Negative) do tư thế phục hồi

Đây là nhóm lỗi nguy hiểm nhất trong bài toán y tế, khi hành động ngã không được phát hiện kịp thời để đưa ra cảnh báo.

(a) Nhãn thật (Ground Truth): Fall Detected (b) Dự đoán (Prediction): Sitting (0.50)

Figure 10: Ví dụ lỗi False Negative do tư thế phục hồi sau ngã

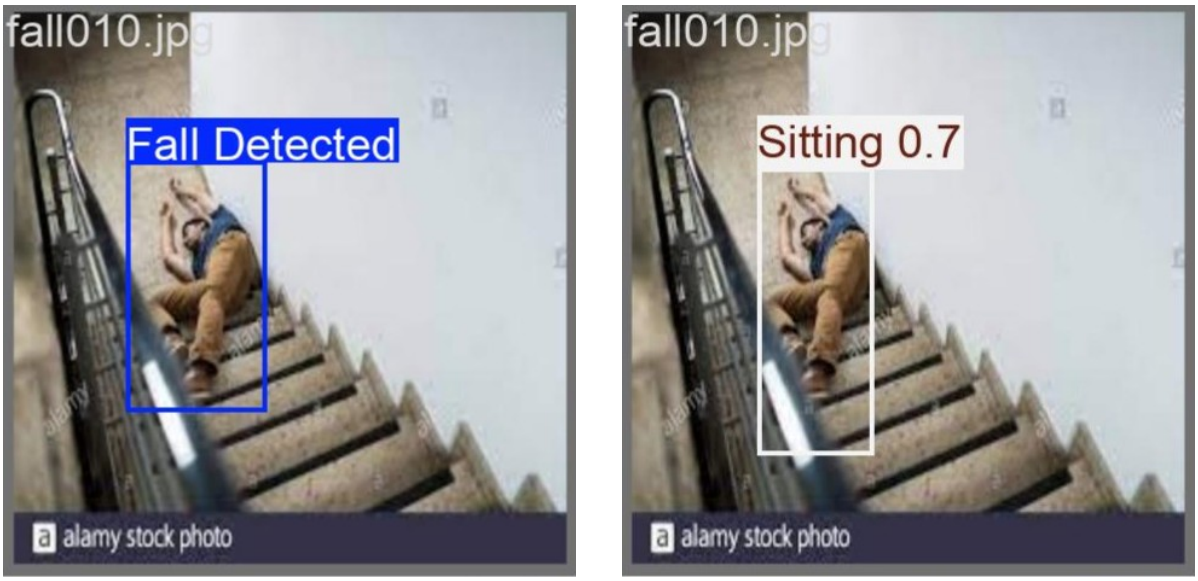
Phân tích nguyên nhân:

- **Sự mơ hồ về tư thế (Pose Ambiguity):** Trong giai đoạn hậu ngã (*post-fall*), nạn nhân thường cố gắng ngồi dậy hoặc với lấy vật hỗ trợ. Khi đó, phần thân trên có xu hướng thẳng đứng so với mặt sàn.

- **Hạn chế của đặc trưng hình học 2D:** Mô hình YOLOv8n chủ yếu dựa vào tỷ lệ khung bao (Bounding Box Aspect Ratio) và hướng trục cơ thể. Việc thân trên thẳng đứng khiến mô hình nhầm lẫn tư thế này với hành động *Sitting*.
- **Thiếu khai thác ngữ cảnh:** Các yếu tố quan trọng như chiếc gậy chống, tư thế chân duỗi thẳng bất thường chưa được mô hình tận dụng hiệu quả để đưa ra quyết định chính xác.

5.2.2 Lỗi bỏ sót ngã (False Negative) do môi trường phức tạp

Các môi trường có cấu trúc hình học đặc biệt như cầu thang hoặc góc tường thường gây nhiễu cho việc nhận diện tư thế người.



(a) Nhãn thật (Ground Truth): Fall Detected (b) Dự đoán (Prediction): Sitting (0.70)

Figure 11: Ví dụ lỗi False Negative do môi trường cầu thang

Phân tích nguyên nhân:

- **Nhiễu nền (Background Clutter):** Các bậc thang vô tình tạo ra hình dạng giống một tư thế ngồi tự nhiên.
- **Sự chồng lấn đặc trưng (Feature Overlap):** Cơ thể người bị gập theo hình dạng bậc thang (đầu gối co, lưng tựa) có độ tương đồng ngữ nghĩa cao với tư thế ngồi nghỉ trên cầu thang.
- **Thiếu dữ liệu đa dạng:** Tập huấn luyện chủ yếu chứa các mẫu ngã trên mặt phẳng, khiến mô hình chưa học được các đặc trưng ngã trên bề mặt lồi lõm hoặc gấp khúc.

5.2.3 Lỗi phân loại sai hành động thường nhật (Misclassification)

Mặc dù không nguy hiểm bằng lỗi bỏ sót ngã, việc phân loại sai các hành động sinh hoạt bình thường có thể gây nhiễu trong hệ thống giám sát.

(a) Nhãn thật (Ground Truth): Sitting (b) Dự đoán (Prediction): Walking (0.90)

Figure 12: Ví dụ lỗi phân loại sai hành động sinh hoạt thường nhật

Phân tích nguyên nhân:

- **Nhận diện sai phần chân:** Tư thế ngồi vắt chéo chân khiến vị trí hai đầu gối so le nhau, dễ bị mô hình hiểu nhầm là chuyển động bước đi.
- **Hạn chế của Object Detection 2D:** Phần ghế ngồi bị khuất hoặc hòa lẫn vào nền phía sau, khiến mô hình không nhận diện được ngữ cảnh "đang ngồi trên ghế".

5.2.4 Tổng kết nguyên nhân lỗi

Từ các trường hợp phân tích trên, có thể rút ra các nguyên nhân chính gây sai số cho mô hình YOLOv8n như sau:

- **Thiếu thông tin thời gian (Temporal Information):** Ảnh tĩnh không thể phân biệt rõ giữa hành động "đang ngồi" và "đang ngã".
- **Sự chồng lấn đặc trưng hình học:** Các tư thế Ngã – Ngồi – Đi có sự tương đồng cao trong không gian 2D.
- **Dữ liệu huấn luyện chưa đa dạng bối cảnh:** Thiếu các mẫu ngã trên cầu thang và các tư thế ngồi phức tạp.

5.3 So sánh và Đánh giá hiệu năng mô hình

Để đảm bảo tính công bằng trong so sánh, tất cả các mô hình đều được huấn luyện và đánh giá trên cùng một cấu hình phần cứng (GPU Tesla T4) và bộ siêu tham số thống nhất, bao gồm 60 epochs huấn luyện, kích thước ảnh đầu vào 640×640 và batch size bằng 16.

Bảng 5 trình bày các chỉ số hiệu năng cốt lõi của ba mô hình YOLOv8n, YOLOv10n và RT-DETR (Large) trên tập Validation.

Table 5: So sánh hiệu năng giữa YOLOv8n, YOLOv10n và RT-DETR

Tiêu chí	**YOLOv8n**	**YOLOv10n**	**RT-DETR (L)**
Số tham số (M)	3.0	**2.7**	32.8
Độ phức tạp (GFLOPs)	8.1	**6.5**	103.4
Thời gian suy luận (ms)	4.6	**3.4**	17.9
Precision	0.774	0.799	**0.809**
Recall	**0.861**	0.708	0.801
mAP@50	**0.854**	0.803	0.805
Recall (Fall Detected)	**0.930**	0.803	0.915

5.3.1 YOLOv8n – Sự cân bằng tối ưu (Best Overall)

YOLOv8n là mô hình đạt hiệu năng tổng thể tốt nhất trên tập dữ liệu thử nghiệm. Mô hình đạt giá trị **mAP@50 cao nhất (0.854)** trong số ba mô hình được đánh giá. Đặc biệt, chỉ số **Recall cho lớp "Fall Detected" đạt tới 0.93**, cho thấy mô hình phát hiện chính xác tới 93% các trường hợp ngã trong thực tế.

Đây là yếu tố then chốt đối với các hệ thống cảnh báo y tế hoặc an toàn, nơi việc bỏ sót sự kiện nguy hiểm có thể gây hậu quả nghiêm trọng. Bên cạnh đó, thời gian suy luận chỉ 4.6 ms/ảnh hoàn toàn đáp ứng yêu cầu xử lý thời gian thực.

Kết luận: YOLOv8n là ứng cử viên phù hợp nhất cho triển khai thực tế nhờ sự cân bằng tối ưu giữa độ chính xác, độ nhạy và tốc độ.

5.3.2 YOLOv10n – Tối ưu tốc độ và hiệu suất (Efficiency King)

YOLOv10n nổi bật với kiến trúc loại bỏ Non-Maximum Suppression (NMS), giúp mô hình đạt thời gian suy luận nhanh nhất, chỉ **3.4 ms/ảnh**, đồng thời sở hữu số lượng tham số thấp nhất (2.7M).

Nhờ đó, YOLOv10n đặc biệt phù hợp cho các thiết bị biên có tài nguyên hạn chế như hệ thống IoT hoặc các nền tảng nhúng cũ. Tuy nhiên, mô hình đánh đổi tốc độ bằng độ nhạy, với **Recall tổng thể chỉ đạt 0.708**. Riêng đối với lớp "Fall Detected", việc bỏ sót gần 20% các trường hợp (Recall = 0.803) là một rủi ro đáng kể trong bối cảnh ứng dụng y tế.

5.3.3 RT-DETR – Kiến trúc Transformer (Heavyweight)

RT-DETR (phiên bản Large) sử dụng kiến trúc Transformer với hơn 32 triệu tham số và độ phức tạp tính toán vượt quá 100 GFLOPs. Mặc dù vậy, giá trị **mAP@50 (0.805)** không vượt qua được mô hình YOLOv8n nhỏ gọn hơn.

Điểm mạnh của RT-DETR nằm ở chỉ số **Precision cao nhất (0.809)**, cho thấy khi mô hình đưa ra cảnh báo, độ tin cậy rất cao và ít xảy ra báo động giả. Tuy nhiên, chi phí tính toán lớn và tốc độ suy luận chậm (17.9 ms/ảnh) khiến mô hình kém phù hợp cho các hệ thống thời gian thực trên phần cứng hạn chế.

Nhận định: RT-DETR thể hiện tiềm năng mạnh mẽ của kiến trúc Transformer, nhưng có xu hướng "dư thừa" đối với tập dữ liệu nhỏ và bài toán phát hiện ngã trong nghiên cứu này.

5.3.4 Lựa chọn mô hình triển khai cuối cùng

Dựa trên các phân tích định lượng và định tính ở trên, nhóm quyết định lựa chọn **YOLOv8n** làm mô hình chính để triển khai hệ thống phát hiện ngã. Quyết định này dựa trên các yếu tố sau:

- Đạt **Recall lớp Ngã cao nhất (93%)**, đảm bảo an toàn cho người dùng.
- Thời gian suy luận nhanh (4.6 ms/ảnh), đáp ứng tốt yêu cầu xử lý thời gian thực.
- Kiến trúc gọn nhẹ, dễ triển khai và tương thích tốt với nhiều nền tảng phần cứng.

6 TÀI LIỆU THAM KHẢO

1 W. Lv et al., "DETRs Beat YOLOs on Real-time Object Detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2024, pp. 16965–16974.

2 Y. Luo, N. Pan, Y. Zhang, Y. Zhang, and X. Wu, "YOLO-Drone: An Optimized YOLOv8 Network for Tiny UAV Object Detection," Sensors, vol. 24, no. 15, p. 4858, Jul. 2024. doi: 10.3390/s24154858.

3 H. Cheng and F. Kang, "Corrosion Detection and Grading Method for Hydraulic Metal Structures Based on an Improved YOLOv10 Sequential Architecture," Applied Sciences, vol. 14, no.[1] 24, p. 12009, Dec. 2024.[1] doi: 10.3390/app142412009.

4 G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," Jan. 2023. [Online].

5 N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in European Conference on Computer Vision (ECCV), 2020, pp. 213–229.

6 N. Lu, Y. Wu, L. Feng, and J. Song, "Deep Learning for Fall Detection: Three-Dimensional CNN Combined With LSTM on Video Kinematic Data," IEEE Journal of Biomedical and Health Informatics, vol. 23, no. 1, pp. 314–323, Jan. 2019.

7 "Artificial intelligence: a modern approach", 2020, Russell, S. and Norvig, P.

8 “Deep learning”, 2016, Goodfellow, I., Bengio, Y., and Courville, A