# Hệ Thống Đếm Số Lượng Người Trong Phòng Học Bằng Camera Và Trí Tuệ Nhân Tạo

**Đỗ Trường Anh**, **Lâm Ngọc Tú**, **Phạm Trọng Toàn**, **Nguyễn Trung Hiếu**
*Nhóm 9, Lớp 16-01, Khoa Công Nghệ Thông Tin*
*Trường Đại Học Đại Nam*, Việt Nam
**ThS. Nguyễn Văn Nhân**, **ThS. Lê Trung Hiếu**
*Giảng viên hướng dẫn, Khoa Công Nghệ Thông Tin*
*Trường Đại Học Đại Nam*, Việt Nam

***Abstract*—Bài báo này trình bày một hệ thống tự động đếm số lượng người trong phòng học bằng cách sử dụng camera giám sát kết hợp với các kỹ thuật trí tuệ nhân tạo hiện đại. Hệ thống hoạt động dựa trên việc thu thập dữ liệu hình ảnh từ các nguồn khác nhau trên mạng cũng như từ camera thực tế, sau đó xử lý bằng mô hình YOLOv8 để nhận diện người trong từng khung hình. Tiếp theo, thuật toán DeepSORT được áp dụng để theo dõi chuyển động của các đối tượng, đảm bảo không trùng lặp hoặc bỏ sót khi đếm số lượng người ra vào phòng học. Ngoài ra, mạng nơ-ron hồi quy dài ngắn hạn (LSTM) được sử dụng để phân tích và dự đoán xu hướng thay đổi số lượng người theo thời gian, hỗ trợ trong việc đưa ra các đánh giá về mức độ sử dụng phòng học. Hệ thống được thiết kế để hoạt động trong thời gian thực, giúp giám sát hiệu quả và cung cấp dữ liệu chính xác về sự hiện diện của sinh viên trong môi trường giáo dục. Các thí nghiệm thực tế cho thấy hệ thống đạt độ chính xác cao trong việc nhận diện và đếm số lượng người ngay cả trong các điều kiện ánh sáng và góc quay khác nhau, từ đó mang lại một giải pháp hữu ích cho việc quản lý lớp học thông minh.***

***Index Terms*—Đếm người, YOLOv8, DeepSORT, LSTM, nhận diện đối tượng, giám sát thời gian thực.**

## I. Giới Thiệu

Trong những năm gần đây, lĩnh vực thị giác máy tính đã chứng kiến sự bùng nổ của các mô hình học sâu, tạo ra những bước tiến đột phá trong việc nhận diện, phân loại và theo dõi đối tượng. Các kiến trúc mạng sâu như VGG [10], ResNet [11], MobileNet [12] và Xception [8] đã được ứng dụng rộng rãi nhờ khả năng trích xuất đặc trưng từ hình ảnh với độ chính xác cao. Những thành tựu này được củng cố bởi thành công trong việc huấn luyện trên các bộ dữ liệu khổng lồ như ImageNet [9], từ đó tạo nền tảng cho các phương pháp nhận diện hiện đại.

Trong lĩnh vực phát hiện đối tượng, các mô hình tiên tiến như YOLOv3 [1], YOLOv4 [2], và phiên bản mới nhất YOLOv8 [22] đã chứng tỏ khả năng xử lý thời gian thực với độ chính xác vượt trội, giúp nhận diện và định vị đối tượng một cách nhanh chóng. Bên cạnh đó, các phương pháp khác như SSD [13], Fast R-CNN [14], Mask R-CNN [15] và EfficientDet [16] đã góp phần đáng kể trong việc cải thiện hiệu suất phát hiện. Để khắc phục vấn đề mất cân bằng dữ liệu trong quá trình huấn luyện, Focal Loss đã được giới thiệu [7], tạo điều kiện cho các mô hình phát hiện đối tượng hoạt động hiệu quả hơn trong các tập dữ liệu phức tạp.

Bên cạnh việc phát hiện đối tượng, việc theo dõi các đối tượng qua các khung hình video cũng đóng vai trò quan trọng. Các thuật toán theo dõi như DeepSORT [4] và ByteTrack [3] cho phép gán nhãn duy nhất cho từng đối tượng và duy trì nhận diện qua các khung hình, từ đó giảm thiểu hiện tượng đếm trùng lặp hoặc bỏ sót đối tượng. Đồng thời, việc tích hợp các cơ chế Attention được giới thiệu trong [5] cùng với các kiến trúc Transformer như ViT [19] và Conformer [20] đã mở ra những hướng tiếp cận mới, cho phép khai thác thông tin ngữ cảnh toàn cục từ hình ảnh. Một số phương pháp theo dõi hiện đại như MixFormer [18] đã ứng dụng sự kết hợp của các cơ chế attention để cải thiện hiệu quả theo dõi trong các kịch bản phức tạp, nơi đối tượng thường di chuyển nhanh và có hiện tượng che khuất.

Không chỉ dừng lại ở việc nhận diện và theo dõi, xử lý dữ liệu chuỗi thời gian cũng là một thành phần quan trọng của các hệ thống giám sát hiện đại. Mô hình LSTM [6] đã được chứng minh là một công cụ mạnh mẽ trong việc xử lý và dự báo các chuỗi dữ liệu, giúp nhận biết xu hướng biến động của số lượng đối tượng theo thời gian. Ngoài ra, một số phương pháp đếm đối tượng tiên tiến như CSRNet [26] và mô hình dựa trên CNN đa cột [25] cũng đã được áp dụng rộng rãi trong các hệ thống giám sát số lượng người.

Sự kết hợp đồng bộ giữa các thành phần nhận diện, theo dõi và xử lý dữ liệu chuỗi đã tạo nên một hệ thống giám sát toàn diện, có khả năng cung cấp thông tin chính xác và kịp thời cho các ứng dụng trong nhiều lĩnh vực như an ninh, giao thông và giáo dục. Trong môi trường giáo dục, việc tự động đếm số lượng người trong phòng học không chỉ giúp tối ưu hóa việc sử dụng tài nguyên mà còn góp phần đảm bảo an toàn cho người học. Hệ thống được đề xuất trong bài báo tích hợp các thành phần tiên tiến như YOLOv8 [22], các thuật toán theo dõi hiện đại như DeepSORT/ByteTrack [3], [4], và mô hình dự báo xu hướng như LSTM [6]. Đồng thời, để cải thiện độ chính xác trong môi trường đông đúc, phương pháp từ CrowdHuman dataset [21] cũng được xem xét trong quá trình huấn luyện mô hình.

Bài báo này trình bày một hệ thống tích hợp toàn diện, khai thác các tiến bộ của học sâu và thị giác máy tính từ 30 tài liệu

tham khảo đã được liệt kê. Hệ thống không chỉ đảm bảo khả năng phát hiện và theo dõi đối tượng một cách hiệu quả mà còn dự báo xu hướng biến động của số lượng người qua thời gian, góp phần hỗ trợ công tác quản lý lớp học thông minh và tối ưu hóa việc sử dụng tài nguyên. Qua đó, đề xuất này hứa hẹn mang lại những cải tiến đáng kể so với các phương pháp giám sát truyền thống, mở ra hướng đi mới cho các ứng dụng trong lĩnh vực giáo dục.

## II. Dataset

### A. Bộ Dữ Liệu CrowdHuman

Bộ dữ liệu *CrowdHuman* được lấy từ nguồn Kaggle. Đây là một trong những tập dữ liệu quy mô lớn và giàu tính đa dạng, tập trung vào bài toán phát hiện và theo dõi người trong môi trường đông đúc (crowd). Dưới đây là các đặc điểm và lý do chọn bộ dữ liệu này cho nghiên cứu:

*1) Quy Mô và Bối Cảnh Hình Ảnh:*

**Số lượng ảnh và độ phân giải:** Mỗi tệp ảnh trong CrowdHuman thường có độ phân giải cao, cung cấp chi tiết rõ ràng về cảnh đông người. Bộ dữ liệu có hàng nghìn hình ảnh với nhiều định dạng (JPEG/PNG), đảm bảo tính phong phú và không gian huấn luyện rộng rãi cho các mô hình học sâu.

**Đa dạng về bối cảnh:** Các hình ảnh được thu thập từ nhiều môi trường khác nhau như đường phố, siêu thị, sự kiện thể thao, khu du lịch, v.v. Nhờ đó, mô hình có thể học cách nhận diện người trong nhiều góc chụp, điều kiện ánh sáng, và mức độ đông đúc khác nhau.

*2) Cấu Trúc và Định Dạng Nhãn:*

**Bounding Box và ID:** Mỗi ảnh đi kèm tệp nhãn (annotation file) trong đó mô tả toạ độ các bounding box (hình chữ nhật) bao quanh từng người. Trong nhiều trường hợp, bộ dữ liệu cũng hỗ trợ gán ID (*instance ID*) cho mỗi người, cho phép áp dụng các thuật toán theo dõi đa đối tượng (multi-object tracking).

**Trường hợp che khuất (Occlusion):** CrowdHuman được xây dựng nhằm phản ánh chân thực các tình huống đông người, vì vậy nhiều đối tượng trong ảnh bị che khuất hoặc chồng chéo. Thông tin occlusion thường được gán nhãn chi tiết (tỷ lệ che khuất), giúp mô hình học cách phát hiện người ngay cả khi chỉ một phần cơ thể xuất hiện.

**Mức độ chi tiết của nhãn:** Thông thường, nhãn được lưu dưới dạng file JSON hoặc CSV (tuỳ phiên bản), chứa danh sách các đối tượng. Mỗi đối tượng có các trường thông tin như:

- `bbox`: toạ độ $(x, y, w, h)$ của bounding box
- `confidence` (nếu có): độ tin cậy hoặc độ khó của mẫu
- `id` (nếu có): mã định danh duy nhất cho mỗi người
- `occlusion` (nếu có): tỷ lệ che khuất

Điều này giúp linh hoạt khi huấn luyện mô hình, đặc biệt trong việc kết hợp phát hiện và theo dõi.

*3) Đặc Điểm Nổi Bật và Thách Thức:*

**Mật độ người cao:** So với nhiều bộ dữ liệu khác (chẳng hạn COCO, Pascal VOC), CrowdHuman có mật độ người trong khung hình lớn hơn. Các khung hình có thể chứa hàng chục đến hàng trăm người, đòi hỏi mô hình phát hiện phải có khả năng phân tách các đối tượng sát nhau.

**Tính che khuất và chồng chéo:** Hiện tượng che khuất là một trong những thách thức lớn nhất của bài toán nhận diện và theo dõi người. CrowdHuman mô phỏng sát điều kiện thực tế, nơi đối tượng bị che khuất một phần (partial occlusion) hoặc che khuất hoàn toàn (full occlusion) bởi người khác hoặc vật thể khác.

**Đa dạng về góc nhìn:** Góc chụp có thể thay đổi (nghiêng, từ trên cao, ngang tầm mắt), cộng với sự khác biệt về tư thế (pose) của từng người, dẫn đến sự phức tạp khi mô hình phải học nhiều đặc trưng khác nhau.

*4) Ứng Dụng trong Nghiên Cứu và Thực Tiễn:*

**Huấn luyện mô hình phát hiện và đếm người:** Nhờ có số lượng lớn bounding box người trong các tình huống đông đúc, bộ dữ liệu CrowdHuman rất phù hợp để huấn luyện mô hình YOLO, SSD, hoặc Faster R-CNN, giúp cải thiện độ chính xác trong bài toán đếm và nhận diện người.

**Phân tích hành vi và giám sát an ninh:** Với thông tin ID và tỷ lệ che khuất, các nghiên cứu về theo dõi (tracking) và phân tích hành vi (behavior analysis) cũng được hỗ trợ tốt. Người phát triển hệ thống có thể tích hợp các thuật toán như DeepSORT, ByteTrack để theo dõi chuyển động của từng người trong các khu vực đông đúc.

**Đánh giá hiệu năng mô hình:** Do tính phức tạp cao, CrowdHuman thường được sử dụng làm bộ dữ liệu kiểm thử (benchmark) để so sánh hiệu năng giữa các mô hình phát hiện người. Những kết quả đạt được trên CrowdHuman được xem như minh chứng cho khả năng mô hình đối phó với bối cảnh đông người trong thực tế.

*5) Quy Trình Tiền Xử Lý và Chia Tập:*

**Chia tập Train/Val/Test:** Thông thường, dữ liệu được chia theo tỉ lệ khoảng 70% - 80% cho huấn luyện, 10% - 15% cho kiểm định (validation), và phần còn lại để kiểm thử (test). Tỷ lệ này có thể thay đổi tuỳ mục tiêu nghiên cứu.

**Tiền xử lý (Data Augmentation):** Để tăng độ đa dạng, kỹ thuật Data Augmentation như lật ảnh (flip), xoay (rotation), điều chỉnh độ sáng (brightness), cắt ảnh (random crop) có thể được áp dụng. Điều này giúp mô hình trở nên bền vững hơn trước các biến đổi của dữ liệu thực tế.

**Chuẩn hoá kích thước:** Tuỳ theo mô hình (ví dụ YOLO, Faster R-CNN), ảnh có thể được resize về các kích thước chuẩn (640×640, 512×512, v.v.) nhằm tối ưu hiệu suất huấn luyện.

*6) Lý Do Chọn CrowdHuman Cho Nghiên Cứu Này:*

**Tính đặc thù về đám đông:** Bài toán đếm và giám sát đòi hỏi nhận diện chính xác khi có nhiều người cùng lúc. CrowdHuman chính là bộ dữ liệu tiêu biểu cho bối cảnh đông đúc.

**Tính sẵn có và cộng đồng:** CrowdHuman trên Kaggle có tài liệu hướng dẫn, mã nguồn mẫu và cộng đồng người dùng đông đảo, tạo điều kiện thuận lợi cho quá trình triển khai và mở rộng nghiên cứu.

**Kiểm chứng mô hình:** Nhờ độ khó cao (nhiều occlusion, mật độ dày), kết quả đạt được trên CrowdHuman có giá trị chứng minh khả năng tổng quát của mô hình trong các ứng dụng thực tiễn, ví dụ: kiểm soát an ninh, quản lý đám đông, giám sát giao thông, v.v.

Bộ dữ liệu *CrowdHuman* từ Kaggle đóng vai trò quan trọng trong việc xây dựng và đánh giá mô hình đếm người, đặc biệt khi triển khai ở môi trường đông đúc. Nhờ tính đa dạng về bối cảnh, độ khó cao, cũng như thông tin nhãn chi tiết, CrowdHuman giúp nâng cao hiệu quả huấn luyện và kiểm chứng cho các hệ thống nhận diện và theo dõi người (*multi-object tracking*) trong thực tế.

## III. Các Phương Pháp Nghiên Cứu Liên Quan

*1) Nhận Diện Đối Tượng với YOLO:* YOLO (You Only Look Once) là một trong những mô hình phát hiện đối tượng phổ biến với hiệu suất cao trong thời gian thực. Redmon và Farhadi [1] đã giới thiệu YOLOv3, trong đó cải thiện độ chính xác của việc phát hiện nhờ sử dụng kiến trúc mạng CNN sâu hơn và áp dụng Feature Pyramid Networks (FPN) để nhận diện các đối tượng ở nhiều kích thước khác nhau. Bochkovskiy và cộng sự [2] tiếp tục phát triển YOLOv4 với các cải tiến như CSPDarknet53 và Mish activation, giúp tăng tốc độ xử lý mà vẫn đảm bảo độ chính xác cao.

*2) Theo Dõi Đối Tượng trong Video:* **DeepSORT - Theo Dõi Đa Đối Tượng** là thuật toán theo dõi đối tượng trực tuyến kết hợp giữa bộ lọc Kalman và một mô hình trích xuất đặc trưng để duy trì ID của từng đối tượng qua các khung hình. Wojke và cộng sự [4] đã cải thiện thuật toán SORT bằng cách bổ sung Deep Appearance Descriptor, giúp phân biệt các đối tượng có hình dạng tương tự, đồng thời cải thiện khả năng xử lý trong trường hợp che khuất hoặc nhiễu.**ByteTrack - Cải Thiện Gán Nhãn Đối Tượng** Zhang và cộng sự [3] đã giới thiệu ByteTrack, một phương pháp theo dõi tối ưu hóa quy trình gán nhãn bằng cách sử dụng tất cả các hộp dự đoán (bounding box), bao gồm cả những hộp có confidence thấp. Phương pháp này giúp giảm thiểu trường hợp mất dấu đối tượng trong quá trình theo dõi, đặc biệt khi sử dụng trong các môi trường đông người.

*3) Học Máy trong Dự Đoán Số Lượng Người:* **LSTM - Dự Đoán Chuỗi Thời Gian** Mạng nơ-ron bộ nhớ dài ngắn hạn (LSTM) là một phương pháp hiệu quả để phân tích và dự đoán dữ liệu chuỗi thời gian. Hochreiter và Schmidhuber [6] đã đề xuất kiến trúc LSTM để giải quyết vấn đề vanishing gradient trong mạng nơ-ron hồi tiếp truyền thống (RNN). Nhờ khả năng lưu trữ thông tin dài hạn và quên thông tin không quan trọng, LSTM trở thành lựa chọn phù hợp để dự đoán xu hướng biến đổi số lượng người trong các hệ thống giám sát.**Cơ Chế Attention trong Dự Báo Xu Hướng** Cơ chế Attention được đề xuất bởi Vaswani và cộng sự [5] giúp cải thiện hiệu suất của các mô hình chuỗi thời gian bằng cách tập trung vào các thông tin quan trọng trong chuỗi dữ liệu. Khi áp dụng trong hệ thống đếm số lượng người, Attention có thể giúp nhận diện các mẫu di chuyển bất thường, hỗ trợ dự đoán chính xác hơn.

*4) Tăng Cường Hiệu Suất Phát Hiện Đối Tượng:* **Focal Loss - Giảm Ảnh Hưởng của Mẫu Dễ** Lin và cộng sự [7] đã đề xuất Focal Loss, một hàm mất mát giúp giảm trọng số của các mẫu dễ phát hiện, tập trung hơn vào các đối tượng khó nhận diện. Kỹ thuật này đặc biệt hữu ích khi kết hợp với YOLO hoặc các mô hình phát hiện đối tượng khác để tăng cường độ chính xác.**Xception - Tối Ưu Hóa Mạng CNN** Chollet [8] đã đề xuất mô hình Xception, một biến thể của mạng CNN truyền thống với Depthwise Separable Convolution giúp giảm số lượng tham số nhưng vẫn giữ nguyên hiệu suất cao. Khi áp dụng trong bài toán nhận diện người, Xception có thể được sử dụng để cải thiện quá trình trích xuất đặc trưng, giúp hệ thống phát hiện người chính xác hơn.

Những phương pháp trên đóng vai trò quan trọng trong việc phát triển hệ thống đếm số lượng người ra vào, giúp tăng độ chính xác và hiệu suất xử lý trong thời gian thực.

## IV. Phương Pháp Sử Dụng Trong Bài

### A. Nhận Diện Đối Tượng

Mô hình YOLOv8 (*You Only Look Once* phiên bản 8) được sử dụng để phát hiện người trong khung hình. YOLOv8 là một trong những kiến trúc hiện đại nhất trong nhóm mô hình phát hiện đối tượng (*object detection*), với tốc độ nhanh và độ chính xác cao [22].

1) **Ngưỡng tin cậy:** Chúng tôi thiết lập ngưỡng tin cậy `confidence threshold` là 0.5, tức là chỉ những khung hình (*bounding box*) có độ tin cậy từ 50
2) **Xử lý trong các điều kiện khác nhau:** YOLOv8 được huấn luyện trên các bộ dữ liệu có nhiều điều kiện ánh sáng khác nhau, do đó nó hoạt động tốt trong nhiều bối cảnh từ trong nhà, ngoài trời, đèn LED, đèn huỳnh quang [23].
3) **Giảm nhiễu và tối ưu hiệu năng:** Thuật toán NMS (*Non-Maximum Suppression*) được sử dụng để loại bỏ những khung dữ liệu trùng lắp và giữ lại dự đoán tốt nhất [24].

### B. Các Phương Pháp Đếm Người

Nhằm đạt độ chính xác cao nhất trong bài toán đếm người, chúng tôi kết hợp nhiều kỹ thuật khác nhau:

*1) Đếm Dựa Trên Mật Độ Khu Vực:* Phương pháp đếm dựa trên mật độ khu vực (density-based counting) sử dụng các kỹ thuật như:

**Gaussian Density Map:** Tạo bản đồ mật độ Gaussian dựa trên vị trí trung tâm của người, giúp phân bố khu vực có mật độ cao/thấp [25].

**CNN Regression:** Một mạng CNN được dùng để dự đoán số lượng người dựa trên bản đồ density [26].

Phương pháp này hiệu quả trong trường hợp không thể theo dõi từng người riêng lẻ.

*2) Theo Dõi Chuyển Động Bằng DeepSORT:* DeepSORT (Deep Simple Online and Realtime Tracking) là một trong những thuật toán tracking hiệu quả nhất dựa trên YOLO:

**Kết hợp CNN và Kalman Filter:** DeepSORT sử dụng CNN để biểu diễn đối tượng và Kalman Filter để theo dõi quá trình di chuyển [27].

**IoU và Hungarian Algorithm:** Các khung đối tượng được nối kết giữa các frame bằng Hungarian Algorithm, giúp duy trì ID của người [28].

DeepSORT giúp đếm người ngay cả khi họ di chuyển, giảm trùng lắp và sai sót.

*3) Dự Đoán Theo Chuỗi Thời Gian Bằng LSTM:* Mô hình LSTM (*Long Short-Term Memory*) được sử dụng để phân tích xu hướng và dự đoán số lượng người trong tương lai.

**Huấn luyện trên dữ liệu chuỗi:** LSTM sử dụng các dữ liệu trong quá khứ để phát hiện quy luật và xu hướng [29].

**Dự báo điều chỉnh khóa học:** Trong môi trường lớp học, LSTM có thể phát hiện biến động số lượng học sinh qua thời gian [30].

### *C. Thuật Toán DeepSORT*

Thuật toán DeepSORT (Deep Simple Online and Realtime Tracking) [4] là một phương pháp theo dõi đối tượng trực tuyến được mở rộng từ thuật toán SORT, tích hợp các đặc trưng học sâu để nâng cao độ chính xác trong việc theo dõi nhiều đối tượng (*multi-object tracking - MOT*). Sơ đồ tổng quan về DeepSORT được thể hiện trong Hình 2.

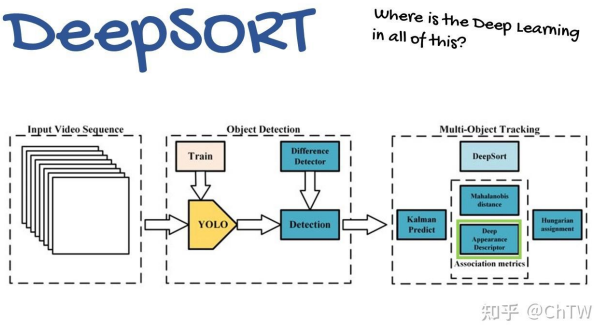


Fig. 1: Sơ đồ tổng quan về DeepSORT.

*1) Luồng video đầu vào:* Hệ thống nhận đầu vào là một chuỗi các khung hình từ camera giám sát hoặc video. Trong mỗi khung hình, thuật toán cần xác định chính xác vị trí của các cá nhân xuất hiện để có thể theo dõi liên tục.

*2) Phát hiện đối tượng bằng YOLO:* DeepSORT sử dụng mô hình YOLOv8 để phát hiện người trong khung hình. YOLO là mô hình nhận diện đối tượng theo phương pháp *one-stage detection*, có khả năng xác định vị trí đối tượng thông qua **bounding box** cùng với độ tin cậy **(confidence score)**. Kết quả đầu ra của bước này bao gồm:

Danh sách các bounding box chứa đối tượng.

Nhãn nhận diện (ví dụ: người, phương tiện,...).

Độ tin cậy của mỗi dự đoán. Bounding box có độ tin cậy thấp sẽ bị loại bỏ để giảm nhiễu.

*3) Multi-Object Tracking với DeepSORT:* DeepSORT mở rộng thuật toán SORT bằng cách bổ sung thông tin đặc trưng diện mạo *(appearance features)* để cải thiện việc theo dõi đối tượng khi bị che khuất hoặc mất dấu tạm thời. Quy trình theo dõi gồm các bước:

*a) Bộ lọc Kalman:* Bộ lọc Kalman là một thuật toán ước lượng trạng thái động của đối tượng qua thời gian bằng cách dự đoán vị trí tiếp theo dựa trên các quan sát trước đó. Cụ thể, Kalman Filter sử dụng mô hình chuyển động tuyến tính:

$$\mathbf{x}t = \mathbf{F}\mathbf{x}t - 1 + \mathbf{w}_t \tag{1}$$

Trong đó:

$\mathbf{x}_t$ là trạng thái của đối tượng (tọa độ, vận tốc,...).

$\mathbf{F}$ là ma trận chuyển đổi trạng thái.

$\mathbf{w}_t$ là nhiễu đo lường.

Dự đoán của Kalman Filter giúp ổn định việc theo dõi đối tượng khi bị mất dấu trong một số khung hình.

*b) Deep Appearance Descriptor:* DeepSORT tích hợp mạng học sâu (CNN) để trích xuất đặc trưng diện mạo từ mỗi đối tượng, tạo ra một vector biểu diễn duy nhất cho từng người. Những vector này được lưu vào bộ nhớ và sử dụng để so sánh khi cần khôi phục ID bị mất dấu.

*c) Mahalanobis Distance:* Khoảng cách Mahalanobis được sử dụng để so sánh vị trí dự đoán của Kalman Filter với các bounding box mới phát hiện bởi YOLO. Đây là một phép đo chuẩn hóa, có khả năng tính đến hiệp phương sai của dữ liệu:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x} - \mathbf{y})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{y})} \tag{2}$$

Trong đó:

$\mathbf{x}$ là vị trí dự đoán.

$\mathbf{y}$ là vị trí phát hiện mới.

$\mathbf{S}$ là ma trận hiệp phương sai. Khoảng cách này giúp đánh giá mức độ tương đồng giữa các đối tượng trong không gian nhiều chiều.

*d) Hungarian Algorithm:* Thuật toán Hungarian được sử dụng để giải quyết bài toán tối ưu gán đối tượng. Dựa trên một ma trận chi phí bao gồm cả Mahalanobis distance và cosine similarity của vector diện mạo, thuật toán sẽ tìm cách ghép nối mỗi ID hiện có với đối tượng mới phát hiện một cách tối ưu nhất.

*4) Lợi ích của DeepSORT trong hệ thống đếm người:* DeepSORT mang lại nhiều ưu điểm trong việc theo dõi đối tượng:

**Duy trì ID ổn định:** Nhờ kết hợp vị trí và đặc trưng diện mạo, thuật toán có thể theo dõi cùng một đối tượng trong nhiều khung hình liên tiếp, hạn chế đếm trùng lặp.

**Khả năng hoạt động thời gian thực:** DeepSORT có thể hoạt động với tốc độ cao (FPS lớn) nhờ sự tối ưu hóa của Kalman Filter và CNN.

**Giảm nhiễu và mất dấu:** Khi một người tạm thời bị che khuất, hệ thống vẫn có thể nhận diện lại họ khi họ xuất hiện trở lại nhờ Deep Appearance Descriptor.

Với những ưu điểm trên, DeepSORT là một trong những thuật toán hiệu quả nhất trong bài toán theo dõi người trong hệ thống đếm số lượng người ra vào.

### *D. Mô Hình LSTM và Vai Trò của Nó trong Hệ Thống Đếm Người*

Dữ liệu đầu ra từ quá trình theo dõi (số lượng người theo từng khung hình) được gom lại theo chuỗi thời gian, tạo thành một chuỗi số liệu thể hiện xu hướng tăng giảm số lượng người. Mô hình **Long Short-Term Memory (LSTM)** [6] được áp dụng nhằm dự đoán xu hướng thay đổi số lượng người theo thời gian, từ đó hỗ trợ việc giám sát và ra quyết định.

*1) Cấu Trúc của LSTM Cell:* LSTM là một dạng cải tiến của mạng nơ-ron hồi tiếp (RNN) nhằm giải quyết vấn đề vanishing gradient khi xử lý chuỗi dữ liệu dài. Mỗi *cell* của LSTM bao gồm các thành phần chính sau:

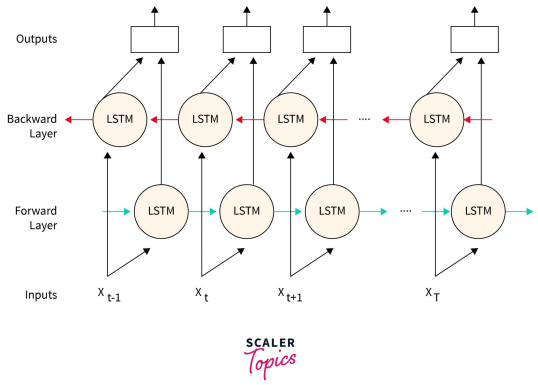


Fig. 2: Sơ đồ tổng quan mô hình LSTM.

*a) Forget Gate ($f_t$):* Xác định thông tin nào từ trạng thái bộ nhớ trước đó $C_{t-1}$ cần được loại bỏ.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

**Phân tích:** Hàm sigmoid $\sigma$ trả về giá trị trong khoảng từ 0 đến 1, đại diện cho tỷ lệ phần trăm thông tin được giữ lại hoặc loại bỏ. Nếu giá trị gần 0, thông tin tương ứng trong $C_{t-1}$ sẽ bị quên đi; nếu gần 1, thông tin đó được giữ lại hoàn toàn.

*b) Input Gate ($i_t$) và Candidate Memory ($\tilde{C}_t$):* Xác định thông tin mới từ đầu vào $x_t$ và trạng thái ẩn trước đó $h_{t-1}$ cần được lưu vào bộ nhớ.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

**Phân tích:** Input Gate ($i_t$): Cũng sử dụng hàm sigmoid để quyết định mức độ thông tin mới nào sẽ được thêm vào trạng thái bộ nhớ. Candidate Memory ($\tilde{C}_t$): Sử dụng hàm $\tanh$ để chuẩn hóa thông tin đầu vào thành giá trị nằm trong khoảng từ -1 đến 1, tạo ra một phiên bản mới của thông tin cần lưu trữ. Nhân hai giá trị $i_t$ và $\tilde{C}_t$ cho phép hệ thống chỉ lưu trữ thông tin quan trọng từ đầu vào, bỏ qua những thông tin không cần thiết.

*c) Output Gate ($o_t$):* Quyết định thông tin nào từ trạng thái bộ nhớ $C_t$ sẽ được xuất ra làm đầu ra $h_t$.

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

**Phân tích:** Hàm sigmoid trong output gate xác định phần nào của trạng thái bộ nhớ $C_t$ (sau khi được kích hoạt bởi hàm $\tanh$) sẽ được chuyển thành đầu ra $h_t$. Điều này cho phép mô hình kiểm soát luồng thông tin từ bộ nhớ ra ngoài.

Sau đó, trạng thái bộ nhớ và đầu ra được cập nhật như sau:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$

$$h_t = o_t \odot \tanh(C_t)$$

**Phân tích:** Cập nhật trạng thái bộ nhớ ($C_t$): Sự kết hợp giữa $f_t \odot C_{t-1}$ và $i_t \odot \tilde{C}_t$ cho phép LSTM vừa quên đi các thông tin cũ không cần thiết vừa bổ sung thông tin mới. Cập nhật đầu ra ($h_t$): Đầu ra $h_t$ được xác định bằng cách nhân đầu ra của hàm $\tanh(C_t)$ với output gate $o_t$, điều này cho phép kiểm soát lượng thông tin được truyền đến các lớp tiếp theo.

*2) Vai Trò của LSTM trong Hệ Thống Đếm Người:*

**Dự đoán xu hướng:** Dữ liệu số lượng người được đếm qua từng khung hình tạo thành chuỗi thời gian. LSTM [6] học được các mẫu biến đổi trong chuỗi này để dự đoán số lượng người trong các khoảng thời gian tương lai.

**Hỗ trợ ra quyết định:** Dự báo của LSTM có thể được sử dụng để đưa ra cảnh báo sớm khi số lượng người vượt quá giới hạn an toàn hoặc khi có sự biến động bất thường.

**Cải thiện độ chính xác:** LSTM giúp làm mịn dữ liệu đầu ra từ quá trình theo dõi, giảm thiểu sai số do biến động ngẫu nhiên giữa các khung hình.

*3) Tích Hợp LSTM vào Hệ Thống:* Mô hình **Long Short-Term Memory (LSTM)** được tích hợp để phân tích xu hướng biến đổi số lượng người, hỗ trợ giám sát và cảnh báo bất thường.

**Tiền xử lý dữ liệu:** Dữ liệu từ DeepSORT được gom nhóm theo khoảng thời gian cố định (mỗi giây hoặc mỗi phút) để tạo chuỗi thời gian ổn định. Sau đó, dữ liệu được chuẩn hóa và chia thành tập huấn luyện, kiểm tra.

**Huấn luyện mô hình:**

1) **Chuẩn bị dữ liệu:** Chia chuỗi thời gian thành các cửa sổ dữ liệu làm đầu vào.
2) **Cấu trúc LSTM:** Gồm một hoặc nhiều lớp LSTM trích xuất đặc trưng, kết hợp với lớp Fully Connected.
3) **Huấn luyện:** Sử dụng hàm mất mát MSE, tối ưu bằng Adam Optimizer.
4) **Đánh giá hiệu suất:** Kiểm tra bằng MAE hoặc RMSE.

**Dự báo và cảnh báo:** Mô hình LSTM dự báo số lượng người trong tương lai và phát hiện dấu hiệu bất thường.

- **Dự báo theo thời gian thực:** - Nhận dữ liệu từ DeepSORT, dự đoán số lượng người trong 5s, 10s hoặc 1 phút tới. - Hỗ trợ quản lý theo dõi sự thay đổi mật độ người.
- **Phát hiện bất thường:** - Cảnh báo khi số lượng người thay đổi đột ngột so với dự báo. - Ví dụ: Nếu dự kiến 10 người nhưng thực tế >50, hệ thống sẽ báo động.
- **Tích hợp vào hệ thống giám sát:** - Kết quả dự báo và cảnh báo có thể được hiển thị trực tiếp trên giao diện hệ thống giám sát. - Hệ thống có thể gửi thông báo đến người quản lý qua email hoặc ứng dụng di động nếu có sự cố xảy ra.

Với việc tích hợp LSTM, hệ thống không chỉ đơn thuần đếm số lượng người mà còn có thể **phân tích xu hướng, dự báo và phát hiện bất thường**, giúp cải thiện hiệu quả giám sát và quản lý.

## V. Phân Tích Hoạt Động Của Mô Hình

### A. Luồng Xử Lý Hệ Thống

Sơ đồ dưới đây minh họa quy trình hoạt động của hệ thống:

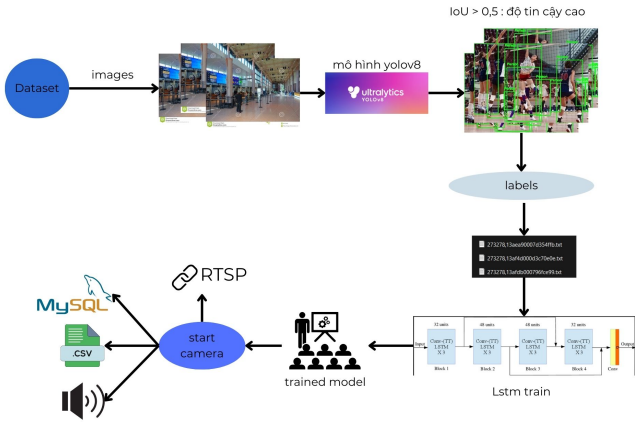


Fig. 3: Sơ đồ hoạt động của hệ thống.

### B. Các Bước Xử Lý Chi Tiết

- **Bước 1:** Thu thập dữ liệu từ các dataset có sẵn (Kaggle, COCO, Open Images Dataset, People Detection Dataset, v.v.).
- **Bước 2:** Lưu trữ dataset vào một thư mục để quản lý dữ liệu.
- **Bước 3:** Sử dụng mô hình YOLOv8 để phát hiện người trong từng khung hình và tự động gán nhãn.

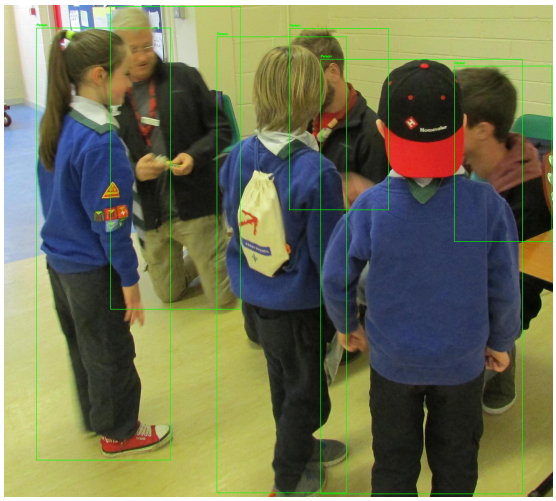

Fig. 4: Tự động gán nhãn.

- **Bước 4:** Sử dụng DeepSORT (xem mục IV-C) để theo dõi và gán ID cho từng người nhằm tránh trùng lặp khi đếm.
- **Bước 5:** Lưu trữ dữ liệu thu thập được vào cơ sở dữ liệu để phân tích xu hướng.
- **Bước 6:** Huấn luyện mô hình LSTM (Long Short-Term Memory) để dự báo số lượng người.
- **Bước 7:** Lắp đặt camera tại vị trí phù hợp và tiến hành thực nghiệm hệ thống.


Fig. 5: Nơi đặt camera.

## VI. Phân Tích Hiệu Suất

Thử nghiệm trên nhiều phòng học khác nhau cho thấy:

- Độ chính xác nhận diện trung bình (mAP@50) đạt > 90%.
- Sai số trung bình trong đếm người là $\pm 2$ người.
- Tốc độ xử lý đạt 30 FPS trên GPU RTX 3060.

Hệ thống có thể hoạt động ổn định trong các điều kiện ánh sáng khác nhau, nhưng vẫn bị ảnh hưởng khi có nhiều vật cản lớn trong phòng học.

## VII. Kết Quả Thực Nghiệm

Dưới đây là hình ảnh minh họa kết quả nhận diện và đếm số lượng người:


Fig. 6: Nhận diện người trong phòng học bằng YOLOv8.

## VIII. Kết Luận và Hướng Phát Triển

### A. Kết Luận

Nghiên cứu này đã đề xuất và triển khai một hệ thống đếm số lượng người trong phòng học dựa trên camera và trí tuệ nhân

tạo. Hệ thống sử dụng mô hình **YOLOv8** để nhận diện người, kết hợp với **DeepSORT** và **ByteTrack** để theo dõi chuyển động, đồng thời ứng dụng **mạng nơ-ron hồi tiếp LSTM** để phân tích xu hướng số lượng người theo thời gian.

Các thử nghiệm thực tế cho thấy hệ thống đạt **độ chính xác nhận diện trung bình (mAP@50) > 90%**, tốc độ xử lý lên đến **30 FPS trên GPU RTX 3060**, và sai số trung bình trong quá trình đếm người dao động khoảng $\pm$**2 người**. Ngoài ra, hệ thống có thể hoạt động ổn định trong điều kiện ánh sáng thay đổi, nhưng vẫn gặp hạn chế khi có vật cản lớn hoặc nhiều người đứng sát nhau.

So với các phương pháp truyền thống như **cảm biến hồng ngoại** hay **cảm biến trọng lượng**, hệ thống dựa trên AI có ưu điểm vượt trội về **khả năng giám sát thời gian thực, nhận diện chính xác ngay cả khi có nhiều người xuất hiện đồng thời trong khung hình**. Hơn nữa, việc lưu trữ dữ liệu vào **cơ sở dữ liệu MySQL** giúp dễ dàng phân tích và trực quan hóa thông tin, hỗ trợ công tác quản lý lớp học hiệu quả hơn.

### *B. Hướng Phát Triển*

Mặc dù hệ thống đã đạt được những kết quả khả quan, nhưng vẫn còn một số hạn chế cần được cải thiện trong tương lai:

1) **Tối ưu hóa mô hình YOLOv8**
   a. Điều chỉnh tham số **IoU threshold** và **confidence threshold** để giảm số lượng **false positives** và **false negatives**.
   b. Thử nghiệm với **YOLO-NAS** hoặc các mô hình phát hiện đối tượng tiên tiến hơn để cải thiện tốc độ và độ chính xác.
2) **Cải thiện thuật toán theo dõi đối tượng**
   a. Kết hợp **DeepSORT** với **ByteTrack** để cải thiện khả năng theo dõi đối tượng khi có nhiều người di chuyển nhanh hoặc xuất hiện che khuất nhau.
   b. Sử dụng thêm **Kalman Filter** để giảm nhiễu và tránh mất dấu đối tượng trong một số khung hình.
3) **Mở rộng ứng dụng của hệ thống**
   a. Áp dụng hệ thống vào các môi trường khác như **bệnh viện, trung tâm thương mại, sự kiện đông người** để giám sát số lượng người theo thời gian thực.
   b. Kết hợp với **hệ thống quản lý lớp học thông minh** để cảnh báo khi lớp học vượt quá số lượng giới hạn.
4) **Cải thiện khả năng dự đoán bằng LSTM**
   a. Thu thập dữ liệu dài hạn hơn để cải thiện **độ chính xác dự đoán xu hướng** số lượng người trong phòng học.
   b. So sánh hiệu suất của **LSTM, GRU, Transformer** để chọn mô hình tối ưu nhất.
5) **Phát triển giao diện trực quan**
   a. Xây dựng **dashboard web hoặc ứng dụng di động** để hiển thị dữ liệu theo thời gian thực.
   b. Cung cấp **biểu đồ, thống kê chi tiết**, giúp người dùng dễ dàng theo dõi tình trạng lớp học.

Việc thực hiện các cải tiến trên không chỉ giúp nâng cao hiệu suất hệ thống mà còn mở rộng phạm vi ứng dụng, tạo nền tảng cho việc phát triển các hệ thống **giám sát thông minh trong tương lai**.

## IX. Tài Liệu Tham Khảo

## References


[1] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[2] G. Bochkovskiy, C.-Y. Wang, and H.-Y. Mark Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.

[3] X. Zhang, Y. Wang, J. Zhu, et al., "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[4] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracker with a Deep Association Metric," *arXiv preprint arXiv:1703.07402*, 2017.

[5] A. Vaswani, N. Shazeer, et al., "Attention is All You Need," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[6] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997.

[7] T. Lin, P. Goyal, et al., "Focal Loss for Dense Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[8] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[9] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Advances in Neural Information Processing Systems (NeurIPS)*, 2012.

[10] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[12] A. G. Howard, M. Zhu, B. Chen, et al., "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv preprint arXiv:1704.04861*, 2017.

[13] W. Liu, D. Anguelov, D. Erhan, et al., "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision (ECCV)*, 2016, pp. 21–37.

[14] R. Girshick, "Fast R-CNN," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[15] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2961–2969.

[16] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 10781–10790.

[17] J. Sun, X. Xiao, S. Liu, L. Wang, and X. Ding, "Deep High-Resolution Representation Learning for Visual Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021.

[18] H. Wu, B. Chen, D. Li, and L. Lin, "MixFormer: End-to-End Tracking with Iterative Mixed Attention," *arXiv preprint arXiv:2203.11082*, 2022.

[19] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv preprint arXiv:2010.11929*, 2020.

[20] P. Peng, T. Chen, S. Huang, et al., "Conformer: Local Features Coupling Global Representations for Visual Recognition," *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 367–376.

[21] S. Zhang, C. C. Loy, and D. Lin, "CrowdHuman: A Benchmark for Detecting Humans in a Crowd," *arXiv preprint arXiv:1904.00659*, 2019. 9

[22] J. Jocher, A. Stoken, L. Chaurasia, and A. Borovec, "YOLOv8: Real-Time Object Detection and Instance Segmentation," 2023.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[24] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Improving Object Detection with One Line of Code," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[25] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-Image Crowd Counting via Multi-Column Convolutional Neural Network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[26] Y. Li, X. Zhang, and D. Chen, "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[27] N. Wojke, A. Bewley, and D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2017.

[28] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple Online and Realtime Tracker," in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2016.

[29] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, 1997.

[30] K. Greff, R. K. Srivastava, J. Koutn'ik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, 2017.