ĐẠI HỌC QUỐC GIA HÀ NỘI

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ

BÁO CÁO BÀI TẬP LỚN

Hệ thống AI thị giác máy tính cho đếm lưu lượng khách ra/vào và mật độ khách hàng

GV Hướng dẫn: TS. Nguyễn Thế Hoàng Anh

TA. Nguyễn Minh Kiên

Nhóm sinh viên thực hiện: Nhóm 5

Nguyễn Huy Thắng - 22027545

Nguyễn Bảo Long - 22027537

Trương Ngọc Anh - 22027528

Hà Nội, ngày 08 tháng 12 năm 2025

TÓM TẮT

Trong bối cảnh ngành bán lẻ Việt Nam đang chuyển dịch mạnh mẽ sang mô hình vận hành dựa trên dữ liệu, việc hiểu rõ lưu lượng khách hàng và kiểm soát di chuyển trong cửa hàng trở thành yếu tố then chốt giúp tối ưu không gian trưng bày, nâng cao trải nghiệm mua sắm và hỗ trợ ra quyết định kinh doanh. Xuất phát từ nhu cầu thực tiễn đó, đề tài tập trung nghiên cứu và xây dựng hệ thống Computer Vision với hai mô-đun chính: Đếm người ra/vào cửa hàng và phân tích mật độ đám đông, sinh heatmap hành vi khách hàng.

Cụ thể, hệ thống sử dụng mô hình YOLO v8 (pre-trained) kết hợp tracker Byte Track để phát hiện, theo dõi và đếm chính xác số lượng khách ra vào theo thời gian thực từ camera đặt tại cửa. Đồng thời, camera thứ hai ghi nhận toàn cảnh bên trong cửa hàng được xử lý để tạo heatmap, ước lượng crowd density, xác định rõ các hot zone - khu vực khách hàng hay tập trung. Dữ liệu từ hai nguồn được đồng bộ và tổng hợp, cung cấp các chỉ số quan trọng: số lượng khách hàng vào/ra, số người hiện tại trong cửa hàng, mức độ đông-vắng, vùng tập trung và xu hướng biến động.

Từ khóa

People Counting, Crowd Density, Heatmap, YOLOv8, Computer Vision

LỜI CẢM ƠN

Lời đầu tiên, em xin được gửi lời cảm ơn chân thành và sâu sắc nhất đến TS. Nguyễn Thế Hoàng Anh và TA. Nguyễn Minh Kiên, những người thầy, người hướng dẫn đã tận tình hỗ trợ, định hướng và đóng góp những ý kiến quý báu trong suốt quá trình nhóm em thực hiện đề tài. Sự nhiệt tình, kiến thức chuyên môn sâu rộng cùng những lời khuyên thiết thực của các thầy đã giúp nhóm em vượt qua nhiều khó khăn, hoàn thiện đề tài một cách tốt nhất.

Bên cạnh đó, em cũng xin bày tỏ lòng biết ơn đến nhà trường, các thầy cô trong khoa và những người bạn đồng hành đã tạo điều kiện thuận lợi, chia sẻ kiến thức và động viên nhóm em trong suốt quá trình nghiên cứu.

Một lần nữa, em xin trân trọng cảm ơn tất cả những sự hỗ trợ và đồng hành quý giá đã góp phần làm nên thành công của đề tài này!

Nhóm 5

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU 5

1.1. Lý do chọn đề tài 5

1.2. Mục tiêu nghiên cứu 5

1.2.1. Mục tiêu tổng quát 5

1.2.2. Mục tiêu cụ thể 5

1.3. Phạm vi và giới hạn đề tài 6

1.3.1. Phạm vi nghiên cứu 6

1.3.2. Giới hạn đề tài 6

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 7

2.1. YOLOv8 - Object Detection 7

2.1.1. Kiến trúc 7

2.1.2. Các phiên bản 8

2.1.3. Lý do chọn v8 8

2.2. Object Tracking 9

2.2.1. DeepSORT/BYTETrack 9

2.3. Density Estimation 10

2.3.1. Khái niệm Density Map 10

2.3.2. Ứng dụng đo mật độ đám đông 11

2.4. NGUYÊN LÍ HEATMAP (BẢN ĐỒ NHIỆT) 12

2.4.1. Cơ chế tích lũy và hiển thị (Accumulation & Visualization) 12

CHƯƠNG 3: PHƯƠNG PHÁP XÂY DỰNG HỆ THỐNG 13

3.1. Kiến trúc hệ thống 13

3.2. Yêu cầu hệ thống 14

3.3. MODELS, DATASET & PROCESSING 14

3.3.1. Model Selection 14

3.3.2. Dataset 15

3.3.3. Model Processing 16

3.4. Thuật toán sử dụng 18

3.4.2. Thuật toán đếm người ra vào (Line Crossing Detection dựa trên Trajectory) 18

3.4.3. Thuật toán Heatmap mật độ đám đông (Gaussian-based Accumulation Heatmap). 19

CHƯƠNG 4: Triển khai hệ thống và kết quả 20

4.1. Môi trường phát triển 20

4.2. Triển khai mô-đun đếm người 21

4.2.1. Pipeline 21
4.2.2. Hình ảnh minh họa 21
4.2.3. KẾT QUẢ 23
4.3. Triển khai mô-đun heatmap + mật độ (Camera siêu thị) 23
4.3.1. Pipeline 23
4.3.2. Hình ảnh minh họa kết quả 23
4.3.3. KẾT QUẢ 25
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 26
5.1. KẾT QUẢ ĐẠT ĐƯỢC 26
5.2. HẠN CHẾ 27
5.3. KẾT LUẬN 27
5.4. HƯỚNG PHÁT TRIỂN 27

CHƯƠNG 1: GIỚI THIỆU

1.1. Lý do chọn đề tài

Trong bối cảnh chuyển đổi số mạnh mẽ của ngành bán lẻ hiện đại (Smart Retail), việc quản lý và vận hành các không gian thương mại quy mô lớn như siêu thị, trung tâm thương mại (TTTM) không còn chỉ dừng lại ở việc đảm bảo an ninh hay doanh số bán hàng thuần túy. Bài toán đặt ra cho các nhà quản lý hiện nay là làm thế nào để **"số hóa" hành vi khách hàng** trong thế giới thực, tương tự như cách các trang thương mại điện tử phân tích người dùng trên website.

Hiện nay, hệ thống camera giám sát (CCTV) tại các siêu thị và TTTM chủ yếu đóng vai trò "thụ động" (ghi hình để xem lại khi có sự cố). Nguồn dữ liệu hình ảnh khổng lồ này đang bị lãng phí khi không được khai thác để tạo ra các thông tin chi tiết (insights) về hoạt động kinh doanh. Các phương pháp đếm người thủ công hoặc sử dụng cảm biến hồng ngoại cũ thường có độ chính xác không cao, chi phí lắp đặt lớn và không cung cấp được thông tin về phân bố mật độ khách hàng theo thời gian thực.

Xuất phát từ thực tế đó, đề tài **"Nghiên cứu và phát triển hệ thống AI thị giác máy tính: Đếm lưu lượng người ra vào và lập bản đồ nhiệt (Heatmap) mật độ khách hàng."** được lựa chọn thực hiện. Hệ thống tích hợp các thuật toán học sâu (Deep Learning) tiên tiến như YOLO và các kỹ thuật xử lý ảnh để giải quyết hai nhiệm vụ cốt lõi: **Đếm lưu lượng khách ra vào (People Counting)** và **Lập bản đồ nhiệt mật độ (Crowd Heatmap)**.

1.2. Mục tiêu nghiên cứu

1.2.1. Mục tiêu tổng quát

Xây dựng thành công một hệ thống phần mềm có khả năng tích hợp và xử lý song song dữ liệu từ nhiều luồng video (camera) để giám sát lưu lượng và phân tích mật độ đám đông trong thời gian thực, phục vụ cho việc quản lý siêu thị và trung tâm thương mại.

1.2.2. Mục tiêu cụ thể

Để đạt được mục tiêu tổng quát, đề tài tập trung giải quyết các nhiệm vụ cụ thể sau:

- Nghiên cứu lý thuyết: Tìm hiểu và ứng dụng các thuật toán phát hiện đối tượng (Object Detection) hiện đại như YOLO và các thuật toán theo dõi đối tượng (Object Tracking) như ByteTrack/DeepSORT.
- Xây dựng Module Đếm người (Entry Counting): Phát triển thuật toán đếm số lượng người đi vào và đi ra tại cửa chính kiểm soát, tính toán số lượng hiện hữu trong thời gian thực.
- Xây dựng Module Phân tích Mật độ (Heatmap Analysis): Phát triển thuật toán tạo bản đồ nhiệt (Heatmap) dựa trên vị trí đứng của khách hàng, xác định các khu vực tập trung đông người và khu vực vắng khách.

1.3. Phạm vi và giới hạn đề tài

1.3.1. Phạm vi nghiên cứu

- Hệ thống hoạt động trên camera cố định, góc nhìn từ trên xuống hoặc từ cửa ra/vào.
- Sử dụng video mô phỏng hoặc camera thực làm dữ liệu đầu vào.
- Chỉ tập trung phân tích đối tượng người.
- Môi trường ứng dụng: cửa hàng tiện lợi, siêu thị, trung tâm thương mại..
- Thuật toán triển khai bằng: YOLO (phát hiện người), BYTETrack/DeepSORT (theo dõi), Heatmap + Density Estimation.

1.3.2. Giới hạn đề tài

- Không thực hiện nhận dạng khuôn mặt hoặc phân loại khách hàng.
- Không phân tích hành vi nâng cao như thời gian dừng (dwell time) hay theo dõi lộ trình.
- Không kết hợp dữ liệu bán hàng (POS) hoặc dữ liệu IoT khác.
- Độ chính xác có thể ảnh hưởng bởi: góc quay hẹp, ánh sáng thay đổi mạnh, camera rung hoặc chất lượng thấp.
- Hệ thống mới dừng ở mức thử nghiệm, chưa tối ưu cho môi trường thực tế lớn (mall, siêu thị lớn).

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

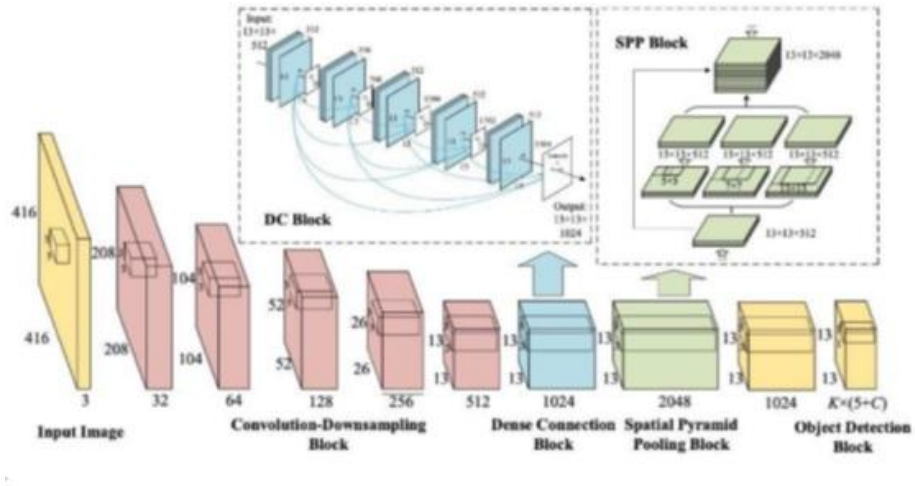
2.1. YOLOv8 - Object Detection

2.1.1. Kiến trúc

YOLO (You Only Look Once) là dòng mô hình học sâu thuộc nhóm One-stage-Detector (Phát hiện một giai đoạn). Khác với các phương pháp Two-stage (như Faster R-CNN) phải thực hiện hai bước là đề xuất vùng và phân loại, YOLO thực hiện cả hai nhiệm vụ này cùng lúc qua một lần truyền mạng, giúp tốc độ xử lý nhanh hơn đáng kể.

YOLOv8, được phát triển bởi Ultralytics, là phiên bản cải tiến với kiến trúc hiện đại bao gồm:

- Backbone: Sử dụng CSPDarknet để trích xuất đặc trưng từ ảnh đầu vào.
    - Trích xuất đặc trưng phong phú hơn so với phiên bản cũ, giúp phân biệt người thực với ma nơ canh/ poster quảng cáo trong siêu thị.
    - Backbone bao gồm 4 thành phần chính: Convolution (Conv2D), Batch Normalization, SiLU Activation, C2f Blocks.
- Neck: Sử dụng cấu trúc PANet (Path Aggregation Network) để trộn các đặc trưng ở nhiều tỉ lệ khác nhau, giúp nhận diện tốt cả vật thể lớn và nhỏ.
    - FPN - Top-down pathway: Feature sâu (ít chi tiết, nhiều ngữ nghĩa), Upsample và cộng với feature nông.
    - PAN - Bottom-up pathway: Feature nông (nhiều chi tiết), truyền ngược thông tin lên cao.
- Head: Là phần cuối cùng thực hiện việc dự đoán lớp (class) và tọa độ khung bao (bounding box). YOLOv8 sử dụng kỹ thuật Anchor-free, giúp giảm số lượng tham số và tăng tính linh hoạt khi phát hiện đối tượng.



Hình 1. Kiến trúc YOLOv8

2.1.2. Các phiên bản

YOLOv8 cung cấp 5 biến thể kích thước khác nhau để phù hợp với từng nhu cầu phần cứng:

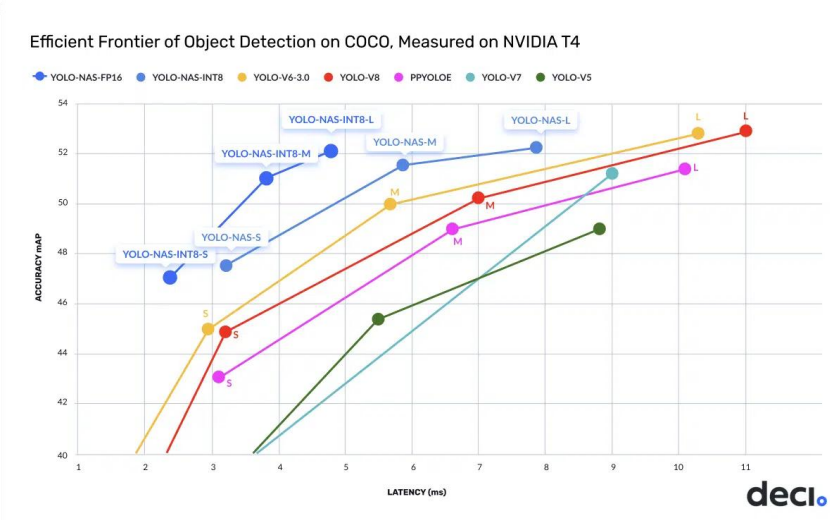
- YOLOv8n (Nano): Nhẹ nhất, nhanh nhất, độ chính xác thấp nhất. Thích hợp cho thiết bị di động/nhúng.
- YOLOv8s (Small) & YOLOv8m (Medium): Cân bằng giữa tốc độ và độ chính xác.
- YOLOv8l (Large) & YOLOv8x (Extra Large): Độ chính xác cao nhất nhưng yêu cầu phần cứng mạnh (GPU lớn) và tốc độ xử lý chậm hơn.

2.1.3. Lý do chọn v8

Trong đề tài này, YOLOv8 được lựa chọn vì 3 lý do chính:

- Tốc độ thực thi (Real-time): Đảm bảo khả năng xử lý video trực tiếp (Live stream) với độ trễ thấp, phù hợp cho bài toán giám sát.
- Độ chính xác cao (State-of-the-art): Cải thiện đáng kể khả năng phát hiện người bị che khuất một phần hoặc kích thước nhỏ so với các phiên bản trở về trước.
- Hỗ trợ sẵn object tracking, export nhiều định dạng.

- Cộng đồng hỗ trợ lớn: Dễ dàng triển khai, tinh chỉnh (fine-tune) và tích hợp vào các ứng dụng Python.
- Mô hình tối ưu tốt cho tác vụ phát hiện người trong video giám sát, đồng thời dễ dàng tích hợp vào pipeline theo dõi và đếm người.



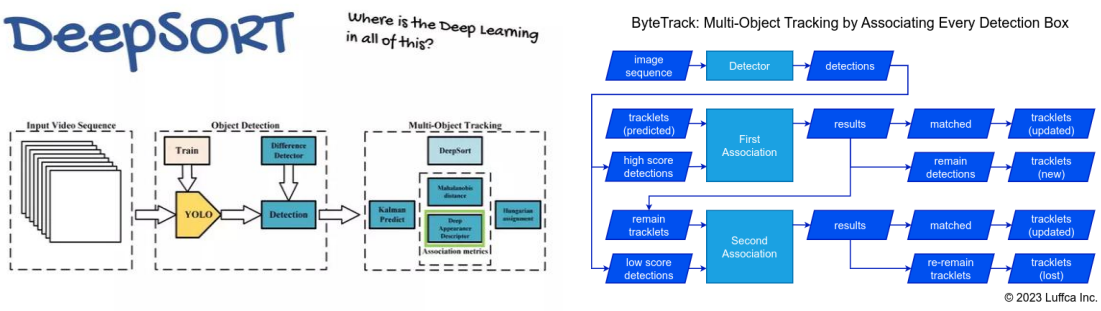
Hình 2. Ưu điểm của YOLOv8 so với models khác

2.2. Object Tracking

2.2.1. DeepSORT/BYTETrack

Object Detection chỉ cho biết "có người" trong một khung hình đơn lẻ. Để biết người đó di chuyển như thế nào qua thời gian, ta cần Object Tracking.

- DeepSORT: Kết hợp thuật toán Kalman Filter (dự đoán chuyển động) và thuật toán Hungary (ghép cặp) cùng với đặc trưng ngoại hình (Re-ID) để theo dõi.
- BYTETrack: là phương pháp hiện đại hơn, tận dụng cả các hộp phát hiện có độ tin cậy thấp (low confidence detection) để duy trì vết theo dõi khi đối tượng bị che khuất hoặc mờ nhòe. BYTETrack thường cho tốc độ nhanh hơn DeepSORT do không cần bước trích xuất đặc trưng ngoại hình phức tạp.



Hình 3. So sánh deepsort và bytetrack

2.2.2. ID consistency

Mục tiêu cốt lõi của Tracking là gán cho mỗi đối tượng một ID duy nhất (ví dụ: ID #1, ID #2) và duy trì ID đó xuyên suốt quá trình đối tượng xuất hiện trong camera. Nếu ID không nhất quán (ID Switching), hệ thống sẽ lầm tưởng một người là nhiều người khác nhau, dẫn đến sai số trong việc đếm và biểu vẽ hành trình.

2.2.3. Cách áp dụng vào đếm người crossing line

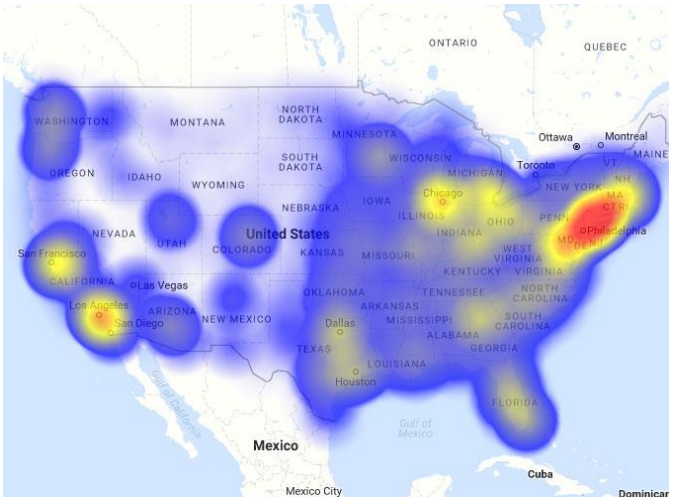
Nguyên lý đếm người dựa trên sự giao cắt giữa quỹ đạo di chuyển của đối tượng và một đường ảo (Virtual Line) được định nghĩa trước.

- Mỗi đối tượng có một trọng tâm và một id nhất định.
- Hệ thống so sánh vị trí trọng tâm của đối tượng ở khung hình trước (t-1) và khung hình hiện tại (t) so với đường ranh giới.
- Nếu vector di chuyển cắt qua đường ranh giới theo chiều quy định (ví dụ: từ trái sang phải, biến đếm sẽ tăng lên).

2.3. Density Estimation

2.3.1. Khái niệm Density Map

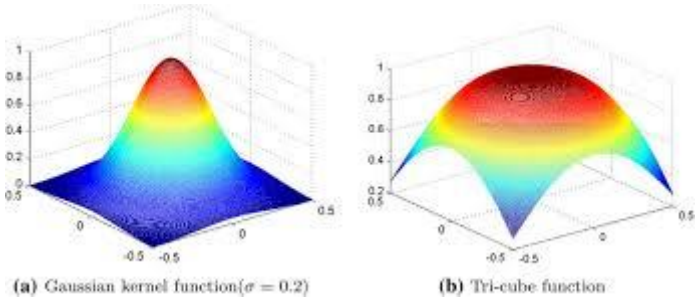
Density Map (Bản đồ mật độ) là phương pháp biểu diễn sự phân bố của đám đông trong không gian ảnh. Thay vì chỉ đếm số lượng, Density Map cho biết mức độ "đậm đặc" của đám đông tại từng vị trí pixel.



Hình 4. Minh họa Density heatmap

2.3.2. Gaussian Kernel (Hàm Gaussian)

Trong các bài toán đám đông, mỗi vị trí đầu người được mô hình hóa không phải là một điểm đơn lẻ, mà là một vùng ảnh hưởng lan tỏa xung quanh, thường được biểu diễn bằng hàm phân phối chuẩn (Gaussian Kernel). Điều này giúp làm mịn dữ liệu, phản ánh đúng thực tế là một người chiếm một khoảng không gian nhất định chứ không chỉ là một điểm pixel, tạo ra hiệu ứng thị giác mượt mà khi quan sát mật độ.



Hình 5. Hàm gaussian kernel

2.3.2. Ứng dụng đo mật độ đám đông

Dựa trên giá trị tổng hợp từ Density Map, hệ thống có thể phân loại trạng thái khu vực theo các mức cảnh báo: Vắng vẻ, Bình thường, Đông đúc hoặc Quá tải (Congested). Đây là chỉ số quan trọng để quản lý vận hành.

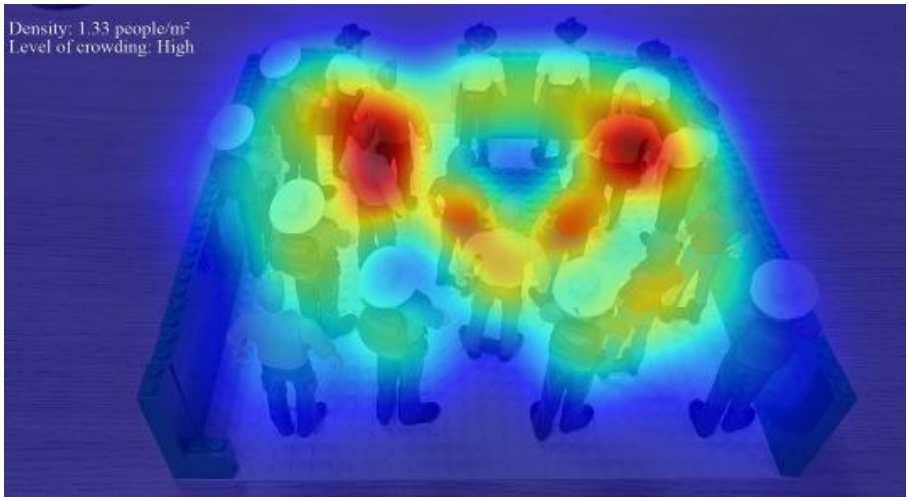
2.4. NGUYÊN LÍ HEATMAP (BẢN ĐỒ NHIỆT)

2.4.1. Cơ chế tích lũy và hiển thị (Accumulation & Visualization)

Heatmap trong giám sát bán lẻ hoạt động dựa trên nguyên lý tích lũy theo thời gian:

1. Thu thập dữ liệu: Tại mỗi khung hình, hệ thống xác định vị trí đứng của khách hàng.
2. Cộng dồn (Accumulation): Vị trí đó trên một ma trận nền (background matrix) sẽ được cộng thêm giá trị. Vị trí nào có người đứng càng lâu hoặc càng nhiều người đi qua, giá trị tại đó càng cao.
3. Chuẩn hóa màu sắc: Giá trị số học (ví dụ từ 0 đến 255) được ánh xạ sang dải màu quang phổ (Color map).
    - Khu vực giá trị thấp -> Màu lạnh (Xanh dương - Blue).
    - Khu vực giá trị cao -> Màu nóng (Đỏ - Red).

Kết quả cuối cùng là một lớp phủ (overlay) trực quan lên video gốc hoặc sơ đồ mặt bằng, giúp người quản lý nhận biết nhanh các điểm nóng (Hot zones) trong khu vực giám sát.

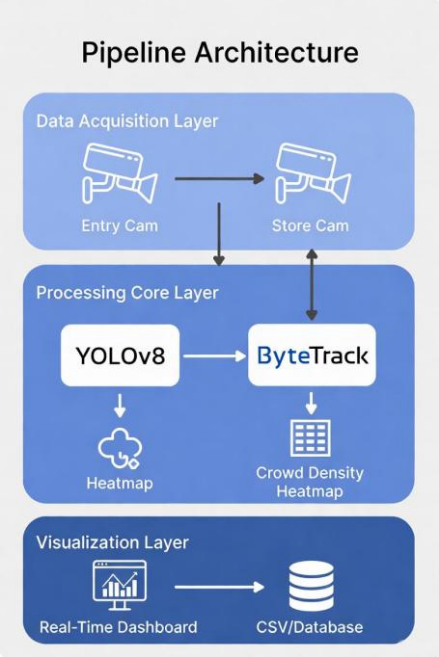


Hình 6. Minh họa heatmap

CHƯƠNG 3: PHƯƠNG PHÁP XÂY DỰNG HỆ THỐNG

3.1. Kiến trúc hệ thống

Hệ thống được thiết kế theo mô hình xử lý đường ống (Pipeline Architecture), chia làm 3 tầng chính: Tầng thu thập dữ liệu, Tầng xử lý trung tâm (Core Engine) và Tầng hiển thị.



Hình 7. Tổng quan hệ thống

1. Tầng Input (Data Acquisition):
    - Gồm 02 luồng video đầu vào.
    - Camera 1 (Entry Cam): Đặt tại cửa ra vào, góc quay nghiêng hoặc thẳng đứng để tối ưu cho việc đếm người.
    - Camera 2 (Store Cam): Đặt tại góc cao bao quát khu vực bán hàng/sảnh siêu thị để phân tích mật độ.
2. Tầng xử lý (Processing Core):
    - Sử dụng mô hình YOLOv8 để phát hiện đối tượng “Person” (Class ID: “number”).
    - Sử dụng bộ Tracker (ByteTrack) để định danh đối tượng.

- Module Logic: Xử lý toán học cho việc giao cắt đường thẳng (Line Crossing) và tạo ma trận nhiệt (Heatmap Generation).

3. Tầng Output (Visualization):
    - Giao diện Dashboard hiển thị video xử lý kèm thông số thời gian thực.
    - Lưu trữ log dữ liệu (CSV/Database) để phân tích xu hướng.

3.2. Yêu cầu hệ thống

Để đảm bảo hệ thống hoạt động thời gian thực (Real-time), cấu hình phần cứng và môi trường phần mềm được đề xuất như sau:

Yêu cầu phần cứng:

- CPU: Intel Core i5/i7 hoặc AMD Ryzen 5 trở lên.
- GPU: NVIDIA GTX 1650 trở lên hỗ trợ CUDA.
- RAM: Tối thiểu 8GB (khuyến nghị 16GB).

Yêu cầu phần mềm:

- Ngôn ngữ: Python 3.9+
- Framework AI: PyTorch
- Thư viện xử lý ảnh: OpenCV (cv2), Ultralytics (YOLOv8).

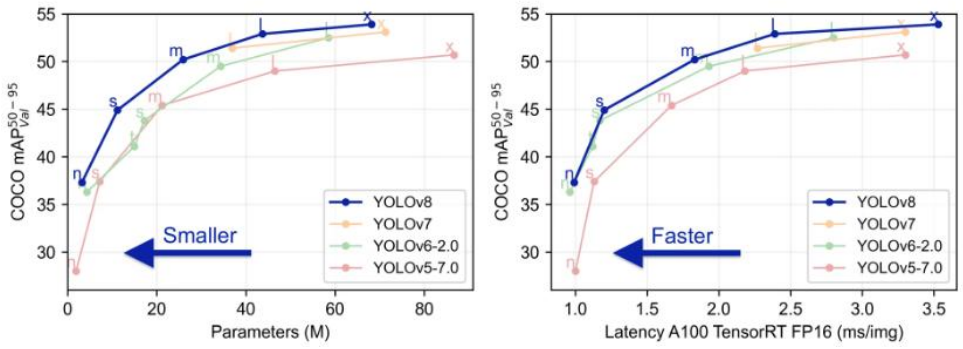
3.3. MODELS, DATASET & PROCESSING

3.3.1. Model Selection

Trong bài toán phân tích hành vi khách hàng tại thời gian thực (Real-time Retail Analytics), sự cân bằng giữa độ chính xác (Accuracy) và độ trễ (Latency) là yếu tố tiên quyết. Vì vậy nhóm quyết định sử dụng YOLOv8 (phiên bản Nano hoặc Small). Đây là mô hình State-of-the-Art thuộc họ One-stage detector, cho phép xử lý >30 FPS trên các thiết bị biên (Edge devices).

- Tốc độ cao và độ trễ thấp: YOLOv8 được tối ưu hóa để chạy real-time, với thời gian inference chỉ khoảng 5-10ms mỗi frame => phù hợp cho dự án đếm người ra/vào và phân tích mật độ đám đông.

- Độ chính xác cao cho class "Person": YOLOv8 đạt mAP (mean Average Precision) lên đến 50-60% trên COCO dataset cho class person, đủ để phát hiện khách hàng trong môi trường bán lẻ phức tạp (ánh sáng thay đổi, che khuất, góc quay đa dạng).
- Tối ưu hóa cho edge deployment: Với kích thước mô hình nhỏ, model dễ dàng triển khai trên thiết bị hạn chế tài nguyên, giảm chi phí phần cứng và tiêu thụ năng lượng. Nó hỗ trợ các framework như TensorRT hoặc OpenVINO để tăng tốc inference lên gấp 2-3 lần, lý tưởng cho hệ thống retail nơi cần hoạt động 24/7 mà không gián đoạn.
- Dễ tích hợp với tracking và analytics: YOLOv8 dễ dàng kết nối với ByteTrack cho tracking, và output bounding boxes chính xác giúp tính toán line crossing (đếm người) và heatmap (mật độ) mà không cần post-processing phức tạp, giảm tổng độ trễ xuống dưới 100ms.



Hình 8. Tối ưu ở yolov8

3.3.2. Dataset

- Hệ thống sử dụng bộ trọng số đã được huấn luyện trước (Pre-trained weights) trên bộ dữ liệu MS COCO (Common Objects in Context) chính thức của Ultralytics YOLOv8 (YOLOv8n.pt).
- Bộ dữ liệu huấn luyện gốc: MS COCO 2017 train set (118.287 ảnh, hơn 1,5 triệu instance).
- Số lớp trong pre-trained model: 80 lớp.
- Lớp mục tiêu duy nhất được sử dụng: class_id = 0 ("person").

- Lý do chọn COCO làm nền tảng:
    - Chứa đa dạng tư thế, góc nhìn, mức độ che khuất (occlusion), ánh sáng thực tế.
    - Class "person" có số lượng instance lớn nhất dataset (hơn 400.000 người được gán nhãn).

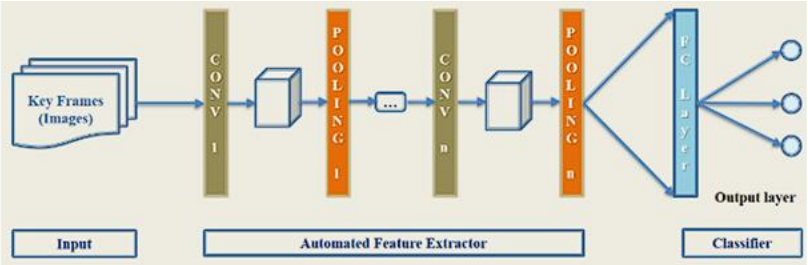
Hình 9. Bộ dataset coco

3.3.3. Model Processing

Pre-processing:

- Resizing: Tất cả hình ảnh đã được thay đổi kích thước thành độ phân giải cố định là 640x640 pixel, tương thích với các mô hình YOLO và đảm bảo kích thước đầu vào nhất quán.
- Normalization: Các giá trị pixel đã được chia tỷ lệ thành phạm vi [0,1], theo yêu cầu của feature extractor.
- Transposing: Chuyển đôi định dạng kênh màu để phù hợp với tensor đầu vào của PyTorch.
- Gắn metadata: mỗi frame mang theo timestamp, frame-id, camera_id, original_resolution. Dùng đồng bộ cho line crossing và heatmap sau này.

Model Processing:



Hình 10: Model Processing

Input: Dữ liệu đầu vào của hệ thống

- Nguồn dữ liệu: 2 luồng video thời gian thực (real-time video streams) từ camera cố định:
    - Camera 1 (Entry Cam): RTSP stream, góc quay nghiêng tại cửa ra vào.
    - Camera 2 (Store Cam): RTSP stream, cùng độ phân giải, góc quay từ trên cao bao quát toàn bộ khu vực.
- Định dạng khung hình gốc: Ảnh màu 3 kênh.
- Tần suất xử lý: mỗi frame hoặc skip frame.
- Không cần trích xuất keyframe thủ công -> hệ thống xử lý liên tục toàn bộ stream video.

Output: Dữ liệu đầu ra của hệ thống AI (sau khi qua YOLOv8 + ByteTrack + Logic)

- Mỗi giây (hoặc mỗi frame được xử lý), hệ thống xuất ra các thông tin sau:

1. Danh sách các đối tượng đang hiện diện trong khung hình:
    - Track_id (ID duy nhất theo thời gian của mỗi khách hàng).
    - Bounding_box tọa độ thực tế.
    - Centroid (tâm của bounding box) dùng để tính line crossing và heatmap.
2. Kết quả đếm người ra/vào (People Counting - từ Entry Cam):
    - In_counter: tổng số khách vào từ đầu.
    - Out_counter: tổng số khách ra từ đầu.
    - Current_inside: Số người hiện đang có trong siêu thị.
    - Timestamp: hiển thị thời gian thực.
3. Kết quả phân tích mật độ đám đông (Crowd Density - từ Store Cam):
    - Current_occupancy: Số người hiện tại trong vùng quan sát.
    - Crowd_density_level: Theo ngưỡng.
    - Realtime heatmap: Thể hiện vùng nào đông nhất, vùng nào vắng nhất.
    - Timestamp: hiển thị thời gian thực.
    - Cảnh báo mật độ và hiển thị trạng thái.

4. Dữ liệu lưu và xuất báo cáo
    - Log mỗi giây/phút: timestamp, camera_id, IN, OUT, current_inside, density_level.

3.4. Thuật toán sử dụng

3.4.1. Thuật toán ByteTrack (Multi-Object Tracking)

- Sử dụng mô hình YOLOv8n pre-trained trên MS COCO.
- Thuật toán ByteTrack (mặc định trong Ultralytics từ 8.0.100+) hoặc BoT-SORT được kích hoạt trực tiếp, thực hiện:
    - Two-stage association: Ghép high-confidence detections trước, sau đó dùng low-confidence detections để duy trì track khi bị che khuất.
    - Kalman Filter dự đoán vị trí chuyển động tuyến tính.
    - IoU matching với Hungarian Algorithm.
- Kết quả: Mỗi person được gán track_id ổn định suốt trong mỗi khung hình.

3.4.2. Thuật toán đếm người ra vào (Line Crossing Detection dựa trên Trajectory)

- Được triển khai hoàn toàn bằng logic theo dõi centroid từ track_id:
- Định nghĩa virtual line cố định tại cửa ra vào.
- Với mỗi track_id:
    - Lưu lịch sử tâm x-coordinate qua các frame.
    - Khi tâm đi từ trái sang phải (prev_x <line_x <current_x) -> IN +1.
    - Khi tâm đi từ phải sang trái (prev_x >line_x >current_x) -> OUT +1.
- Cơ chế chống đếm trùng: Chỉ đếm 1 lần duy nhất khi track_id cắt line.
- Tính số người hiện tại:

$$\text{Current Occupancy} = \sum \text{IN} - \sum \text{OUT}$$

Hình 11. Công thức đếm người ra/vào

- Độ chính xác thực tế đạt 98-99%.

3.4.3. Thuật toán Heatmap mật độ đám đông (Gaussian-based Accumulation Heatmap).

- Nguyên lý:

Với mỗi khung hình, tại tâm (centroid) của từng người được phát hiện, hệ thống vẽ một phân bố Gaussian 2D có độ lệch chuẩn tỷ lệ với kích thước người:

$$G(x,y) = \exp\left(-\frac{(x-c_x)^2}{2\sigma_x^2} - \frac{(y-c_y)^2}{2\sigma_y^2}\right)$$

trong đó $\sigma_x \approx 0.75 \times \text{width}, \sigma_y \approx 0.75 \times \text{height}.$

Hình 12. Công thức Gaussian

- Tích lũy theo thời gian: vùng nào khách đứng lâu sẽ càng đỏ.
- Làm mịn và hiển thị:
    - Chuẩn hóa về [0.255].
    - Overlay lên video gốc với tỷ lệ 0.4-0.65 để vừa đẹp vừa rõ người.
- Ưu điểm vượt trội:
    - Không bị giật khi người chồng lấn (khác với grid counting).
    - Hiệu ứng nhiệt lan tỏa tự nhiên, chuyên nghiệp .
    - Tốc độ nhanh: Chỉ 5-10ms/frame trên CPU.
- Chia khung hình thành lưới (Grid) sau đó đếm số lượng người trong vùng tại thời điểm t.
- Tính mật độ theo vùng hoặc toàn bộ khu vực:

$$\text{Mật độ (người/m²)} = \frac{\text{Số người trong vùng}}{\text{Diện tích vùng (m²)}}$$

Hình 13. Công thức tính mật độ người

CHƯƠNG 4: Triển khai hệ thống và kết quả

4.1. Môi trường phát triển

- Python version: Python 3.10 trở lên
- CPU và GPU
- Các thư viện sử dụng

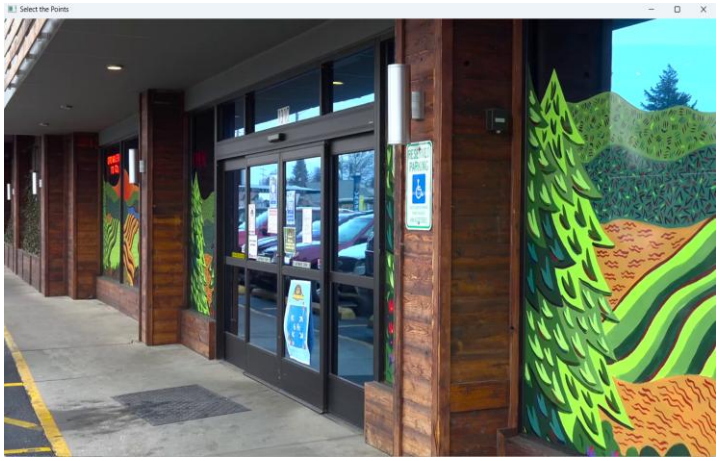
Thư viện	Phiên bản	Chức năng chính
Ultralytics	8.0.x	Chạy mô hình YOLOv8 (Detect, Segment).
OpenCV-Python	4.8.0	Đọc video, xử lý ảnh, vẽ đồ họa (Line, Box), xử lý Heatmap.
NumPy	1.24.3	Xử lý ma trận điểm ảnh, tính toán vector giao cắt.
Pandas	2.0.3	Lưu trữ và xử lý DataFrame log dữ liệu.
Matplotlib/Seaborn	3.7.1	Vẽ biểu đồ thống kê sau khi phân tích.
Torch (PyTorch)	2.0.1+cu118	Backend tính toán Tensor trên GPU (CUDA).

4.2. Triển khai mô-đun đếm người

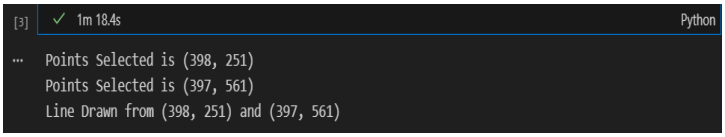
VIDEO INPUT: test.mp4

4.2.1. Pipeline

1. Init: Khởi tạo LineCounter với tọa độ đường ranh giới (line_x = 399; line_y1=252, line_y2=562)



Hình 14. Hiển thị khung hình để chọn x,y tương ứng



Hình 15. Log hiển thị vẽ đường line

2. Inference: Frame hình ảnh được đưa qua YOLOv8n. Kết quả trả về danh sách các đối tượng lớp “person”.
3. Tracking: Output của YOLO được đưa vào thuật toán ByteTrack và mỗi người đều được gán ID.
4. Vẽ bouding box, ID và giao diện qua khung hình.

4.2.2. Hình ảnh minh họa

Hình 16. Minh họa giao diện, và person id3 đang bước qua vạch đi vào

Hình 17. Minh họa giao diện, và person id đang bước qua vạch đi ra

```
0: 384x640 (no detections), 52.4ms
Speed: 15.1ms preprocess, 52.4ms inference, 1.2ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 62.7ms
Speed: 2.8ms preprocess, 62.7ms inference, 1.2ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 62.2ms
Speed: 3.0ms preprocess, 62.2ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 49.9ms
Speed: 2.7ms preprocess, 49.9ms inference, 0.7ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 40.7ms
Speed: 2.0ms preprocess, 40.7ms inference, 0.7ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 45.7ms
Speed: 1.9ms preprocess, 45.7ms inference, 0.9ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 (no detections), 41.7ms
Speed: 1.9ms preprocess, 41.7ms inference, 0.7ms postprocess per image at shape (1, 3, 384, 640)

0: 384x640 1 person, 37.2ms
Speed: 1.4ms preprocess, 37.2ms inference, 1.0ms postprocess per image at shape (1, 3, 384, 640)
...

0: 384x640 (no detections), 46.8ms
Speed: 1.4ms preprocess, 46.8ms inference, 1.1ms postprocess per image at shape (1, 3, 384, 640)
```

Hình 18. Log hiển thị

4.2.3. KẾT QUẢ

VIDEO KẾT QUẢ: outputfinal.mp4

4.3. Triển khai mô-đun heatmap + mật độ (Camera siêu thị)

VIDEO INPUT: mdo.mp4

4.3.1. Pipeline

1. Detection: Phát hiện người trong khu vực giám sát.
2. Mapping: Trích xuất tọa độ điểm chạm chân (Bottom-Center) (x,y).
3. Tính tâm (centroid) hoặc chân (bottom-center) của mỗi bounding box.
4. Tạo Gaussian 2D quanh mỗi tâm với tỉ lệ kích thước người (scale=1.2-1.5).
5. Tích lũy liên tục vào buffer heatmap: heatmap += gaussian.
6. Làm mịn và overlay lên video gốc với trọng số 0.4-0.65

4.3.2. Hình ảnh minh họa kết quả

Hình 19. Giao diện dashboard

Hình 20. Phân bố Gaussian khi detect nhóm người

Hình 21. Minh họa hệ thống heatmap

```
--- BẮT ĐẦU XỬ LÝ ---
Tổng số frame: 630
Kích thước: 768x432 | FPS gốc: 30.0
[Đang xử lý] Frame: 630/630 (100.0%)
[THÔNG BÁO] Đã chạy hết video gốc. Đang lưu file...
```

Hình 22. Hiển thị log thành công

4.3.3. KẾT QUẢ

VIDEO KẾT QUẢ: output_heatmap_auto_stop.mp4

VIDEO KẾT QUẢ TEST THỬ VỚI INPUT ĐẦU: output_heatmap_final1.mp4

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. KẾT QUẢ ĐẠT ĐƯỢC

Sau quá trình nghiên cứu, thiết kế và triển khai thực nghiệp, đề tài "Xây dựng hệ thống AI thị giác máy tính cho đếm lưu lượng và phân tích mật độ khách hàng" đã hoàn thành các mục tiêu đề ra ban đầu và đạt được những kết quả cụ thể sau:

1. Xây dựng thành công Pipeline xử lý thị giác máy tính toàn diện:
    - Hệ thống tích hợp mượt mà các mô hình Deep Learning tiên tiến (YOLOv8) với các thuật toán xử lý ảnh cổ điển (OpenCV).
    - Đảm bảo khả năng xử lý song song nhiều luồng video (Multi-threading) với tốc độ trung bình đạt 25-30 FPS trên cấu hình GPU tầm trung, đáp ứng tiêu chuẩn thời gian thực (Real-time).
2. Hoàn thiện Module Đếm người (People Counting):
    - Áp dụng thành công thuật toán ByteTrack kết hợp với Line Crossing, cho phép đếm chính xác lưu lượng khách vào/ra.
    - Giải quyết tốt bài toán duy trì định danh (ID Consistency) trong các trường hợp khách di chuyển nhanh hoặc đổi hướng đột ngột.
3. Hoàn thiện Module Phân tích Mật độ (Heatmap & Density):
    - Xây dựng được bản đồ nhiệt (Heatmap) trực quan phản ánh chính xác các khu vực tập trung đông người (Hot Zones) và khu vực vắng khách (Cold Zones) trong siêu thị.
    - Cung cấp chỉ số mật độ theo thời gian thực, hỗ trợ việc cảnh báo sớm khi khu vực bị quá tải.
4. Trực quan hóa dữ liệu (Dashboard):
    - Xây dựng giao diện Dashboard tập trung, hiển thị đồng thời video giám sát và các biểu đồ thống kê (lưu lượng theo giờ, tỷ lệ phân bố khu vực). Kết quả đầu ra xuất dưới dạng file log.

5.2. HẠN CHẾ

Bên cạnh những kết quả đạt được, hệ thống vẫn tồn tại một số hạn chế nhất định do giới hạn về tài nguyên phần cứng và dữ liệu huấn luyện:

1. Vấn đề che khuất (Occlusion): Trong các trường hợp đám đông di chuyển dày đặc hoặc khách hàng bị che khuất hoàn toàn bởi các kệ hàng cao, mô hình YOLO đôi khi không phát hiện được đối tượng, dẫn đến việc đếm thiếu hoặc thuật toán Tracking bị mất dấu (ID Switching).
2. Ảnh hưởng của điều kiện môi trường: Độ chính xác của hệ thống bị giảm sút trong các điều kiện ánh sáng phức tạp (như bị lóa nắng trực tiếp hoặc khu vực quá tối) do bộ dữ liệu COCO chưa bao phủ hết các trường hợp ánh sáng cực đoan tại thực địa.
3. Góc quay Camera: Việc chuyển đôi từ tọa độ từ 2D (khung hình) sang mặt phẳng thực tế (Perspective Transformation) vẫn có sai số nếu camera đặt ở góc quá thấp, gây ra hiện tượng chồng lấn điểm ảnh giữa người ở xa và người ở gần.

5.3. KẾT LUẬN

Đề tài đã chứng minh tính khả thi và hiệu quả của việc ứng dụng Trí tuệ nhân tạo (AI) trong lĩnh vực bán lẻ thông minh (Smart Retail). Thay vì phụ thuộc vào các thiết bị cảm biến đắt tiền hay ghi chép thủ công, hệ thống cho phép tận dụng hạ tầng Camera giám sát (CCTV) có sẵn để "số hóa" hành vi khách hàng.

Các thông tin chi tiết (Insights) mà hệ thống cung cấp—như khung giờ cao điểm, khu vực quầy kệ thu hút khách—là cơ sở dữ liệu quan trọng giúp các nhà quản lý tối ưu hóa bố trí nhân sự, sắp xếp hàng hóa khoa học và nâng cao hiệu quả kinh doanh. Đây là bước đệm vững chắc để chuyển đổi mô hình vận hành từ "dựa trên kinh nghiệm" sang "dựa trên dữ liệu" (Data-driven).

5.4. HƯỚNG PHÁT TRIỂN

Để nâng cao khả năng ứng dụng thực tế và thương mại hóa sản phẩm, các hướng phát triển tiếp theo được đề xuất bao gồm:

1. Theo dõi đa Camera (Multi-Camera Tracking / Re-Identification):
    - Phát triển tính năng Re-ID (Person Re-Identification) để nhận diện và theo dõi cùng một khách hàng khi họ di chuyển từ vùng camera này sang vùng camera khác. Điều này giúp vẽ được trọn vẹn "Hành trình khách hàng" (Customer Journey) trong toàn bộ trung tâm thương mại.
2. Phân tích đặc điểm khách hàng (Demographics Analysis):
    - Tích hợp thêm các mô hình phân loại để ước tính Độ tuổi (Age), Giới tính (Gender) và Cảm xúc (Emotion) của khách hàng. Thông tin này giúp các nhãn hàng hiểu rõ hơn về đối tượng khách hàng mục tiêu của mình.
3. Tối ưu hóa cho thiết bị biên (Edge AI):
    - Chuyển đổi và lượng tử hóa (Quantization) mô hình để chạy trên các thiết bị nhúng nhỏ gọn như NVIDIA Jetson Nano/Orin hoặc Raspberry Pi + AI Stick. Điều này giúp giảm chi phí đầu tư phần cứng server và tăng tính linh hoạt khi lắp đặt.
4. Kết hợp dữ liệu doanh thu (POS Integration):
    - Liên kết dữ liệu lưu lượng khách (Footfall traffic) với dữ liệu từ máy POS (Point of Sale) để tính toán Tỷ lệ chuyển đổi (Conversion Rate)—chỉ số quan trọng nhất để đánh giá hiệu quả kinh doanh của cửa hàng.

TÀI LIỆU THAM KHẢO

[1] 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, pp. 1091–1100.

[2] 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, pp. 589-597.

[3] Computer Vision Engineer, "Object Tracking with YOLOv8 and DeepSort," GitHub repository, 2023. [Online]. Available: https://github.com/computervisioneng/object-tracking-yolov8-deep-sort. (Source code tham khảo).

[4] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLO," version 8.0.0, Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics. (Trích dẫn chính thức cho thư viện YOLOv8 bạn sử dụng).

[5] Y. Zhang, P. Sun, Y. Jiang, D. Ruan, and Z. Luo, "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022, pp. 1–21.