Applied Intelligence (2025) 55:1078
https://doi.org/10.1007/s10489-025-06954-9

An end-to-end distributed deep learning system for real-time passenger flow measurement in transport interchanges

Eduardo Salas[1] · Pedro J. Navarro[1] · Francisca Rosique[1] · Juan Benavente[2,3] · Ana María Rivadeneira[2]

Received: 16 May 2025 / Accepted: 10 October 2025

Abstract

As urban populations continue to grow, managing and optimizing urban mobility has become increasingly complex, especially in multimodal transport interchanges. Accurate passenger flow measurement has therefore become essential for operators to mitigate congestion and improve service efficiency. This work proposes a scalable and flexible end-to-end system designed to accurately measure and track passenger flow in real-time. The system integrates a distributed network of Edge-AI sensor nodes with deep learning algorithms for local passenger detection and tracking, while a central processing server aggregates node outputs to derive flow counts. This approach overcomes the limitations of traditional single-sensor solutions by effectively handling occlusion and complex spatial configurations across multiple access points. Validated in a high-transited transport hub, results show that the system achieves accuracy rates between 94.03% and 99.30% even under crowded conditions with flow rates of 100 persons per minute, demonstrating its robustness and practical applicability in dynamic, high-density environments.

Keywords End-to-end systems · Passenger flow measurement · Transport interchanges · Deep learning · Computer vision

1 Introduction

Urban mobility is facing unprecedented challenges due to the continuous increase in urban populations. The United Nations (UN) [1] estimates that by 2050, 68% of the global population will reside in urban areas. Optimizing and sizing various modes of transportation in cities is a significant challenge for public and private transportation management institutions. Factors such as the increasing urban population, migration to suburban areas due to real estate speculation and rising housing prices, and the growing number of remote workers have introduced greater dynamism and complexity into individual mobility and multimodal transportation connections. This demographic shift necessitates optimizing multimodal transportation systems to ensure efficient passenger flow and reduce congestion at transport interchanges. Consequently, the transportation prediction models previously employed, based on historical data and static predictive models, are becoming increasingly complex and ineffective, [2].

Transport interchanges are designed to facilitate multimodal transportation connections, enhancing accessibility and connectivity among different metropolitan and urban transportation systems [3]. These urban spaces increase transportation efficiency and optimize routes by consolidating various types of transport in one location. From a user perspective, transport interchanges reduce waiting times and offer recreational spaces and commercial activities, contributing to a better perception of the transportation service by the end-user [4]. Additionally, transport hubs help reduce emissions in cities. For instance, concentrating vehicles in underground hubs allows for the capture and filtration of CO2 emissions and noise reduction, mitigating the environmental impact of transportation and contributing

✉ Francisca Rosique
paqui.rosique@upct.es

[1] Escuela Técnica Superior de Ingeniería de Telecomunicación – DSIE, Universidad Politécnica de Cartagena, Cartagena 30202, Spain

[2] Transport Research Centre – TRANSyT, Universidad Politécnica de Madrid, Madrid 28040, Spain

[3] Departamento de Ingeniería Civil, Universidad de Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain

Published online: 06 November 2025

to sustainability [5]. However, one of the main operational challenges in these hubs is managing passenger flow, particularly during peak hours when overcrowding leads to inefficiencies and delays. Ensuring optimal transit operations requires agile and automated systems for measuring passenger flow that accurately reflect the congestion levels in key areas of the transport interchange. Real-time measurement of passenger flow transitioning from one mode of transport (such as rail, metro, subway, bus stations) to others (pedestrian, cyclist, private vehicle, taxi) will enable dynamic planning and optimization of the transport hub.

Several methodologies have been proposed for passenger flow measurement, ranging from manual counting to sensor-based tracking solutions. Traditional pedestrian flow monitoring systems typically rely on single-camera setups or sensor-based counting, which are often limited to controlled environments such as narrow corridors or specific entry/exit points. These methods, while effective in specific scenarios, struggle to provide reliable results in large, open spaces within transport hubs, where high-density passenger movements and occlusion effects complicate detection and tracking.

Recent advancements in Artificial Intelligence (AI) [6] and Computer Vision Systems [7] have enabled the development of more sophisticated passenger detection and counting solutions [8]. Deep learning architectures such as Convolutional Neural Networks (CNN) [9], Recurrent Neural Networks (RNN) [10], Long Short-Term Memory (LSTM) [11, 12] and Vision Transformer (ViT) [13] have demonstrated their effectiveness and robustness in automatically counting and measuring passenger flows in various scenarios. Some of the most used techniques include object detection models specifically trained to identify passengers, cars, trucks, and cyclists. However, most existing solutions are constrained to single-sensor setups operating in narrow or controlled environments. These systems often struggle with scalability, occlusion, and performance degradation in complex, high-density spaces like large transport hubs.

In recent years, several multi-sensor approaches have been proposed to address these issues, leveraging data fusion from multiple viewpoints to improve counting accuracy under challenging conditions such as severe occlusion, illumination changes, and high crowd density. Examples include transformer-based architectures for multi-view crowd counting [14], modal adaptative spatial aware fusion and propagation networks for multimodal vision crowd counting [15], and multi-view linkage strategies for indoor person counting [16]. Despite their improvements, many of these approaches require high-bandwidth data transmission to centralized servers or face limitations in adaptability to heterogeneous layouts, which constrains their scalability in large-scale deployments.

To address these limitations, this work presents a novel end-to-end system that combines Edge Artificial Intelligence (Edge AI) with a distributed deep learning architecture for real-time monitoring of passenger flow in transport interchanges. The proposed approach combines adaptive camera field-of-view segmentation with a robust re-identification strategy to accurately track individuals across overlapping sensor nodes. Unlike conventional methods, our system is designed to operate effectively in complex spatial configurations and under varying levels of congestion and lighting. By executing inference locally on each sensor node and exchanging only lightweight data, the proposed solution minimizes network load and latency, enabling efficient scalability across diverse transport hub layouts. Its performance is evaluated through extensive experimentation in a real-world transport hub, demonstrating high accuracy, scalability, and resilience. Furthermore, a comparative analysis with existing approaches highlights the advantages of our method in terms of precision, adaptability, and deployment feasibility.

The remainder of this paper is structured as follows: Section 2 reviews the related work in the field of passenger flow monitoring and intelligent transport systems. Then, Section 3 provides a detailed description of the materials and methods used, including the hardware setup, the proposed counting methodology, the deep learning models and techniques employed, and the case study. Section 4 presents the evaluation metrics, the experimental setup, the obtained results, and a comprehensive discussion of key findings, including comparison with other works. Finally, Section 5 concludes the document by summarizing the main contributions, outlining practical implications, and proposing future research directions to enhance and refine the system.

2 Related works

The development of automatic passenger counting (APC) and flow measurement in specific areas of transportation systems has attracted significant research attention in recent years, focusing primarily on specific transportation environments such as subway platforms, bus boarding and alighting areas, and high-density commercial spaces. Various approaches have been proposed, from classic sensor-based systems to advanced computer vision and artificial intelligence solutions.

As traditional sensor-based solutions reached their limitations, the field of passenger counting has increasingly adopted computer vision approaches, enhanced by advances in artificial intelligence [17]. These systems leverage visual data captured by cameras and process it using AI models

capable of detecting and tracking passengers with high accuracy, even in complex and crowded environments.

Convolutional Neural Networks (CNNs) have been widely used for people counting due to their ability to extract complex spatial features from images. A prominent example is RetailNet [18], which combines the traditional RGB image with an additional layer that indicates the probability that a pixel contains a person. This technique improves the accuracy in estimating the number of people present; however, its effectiveness is mainly limited to controlled indoor spaces, which restricts its application in transport exchangers with dynamic lighting and variable crowd densities. To address these limitations, LRCN-RetailNet [19], an advanced version that integrates Long-Short-Term Convolutional Neural Networks (LRCN). This hybrid architecture combines the capabilities of CNNs for spatial feature extraction with Recurrent Neural Networks (RNNs) to capture temporal coherence in video sequences, allowing to more accurately predict the number of people by taking advantage of temporal information and better handling occlusions and variations in people's postures.

Based on CNN, YOLO models have been widely used in passenger detection in public transport due to their ability to perform real-time detections with high accuracy. These approaches have demonstrated good performance in controlled environments such as buses, subway platforms, and narrow corridors, often using single-camera systems mounted in overhead or frontal positions [20–24]. In some cases, multi-stage architectures combining feature extraction with object detection are used to improve accuracy in dense areas [25].

Other works have focused on tracking and measuring passenger flow or congestion using video surveillance and trajectory analysis. For instance, entropy-based metrics have been proposed to estimate congestion levels near platform screen doors [26].

On the other hand, LSTMs, a variant of RNNs, are especially effective at capturing long-term dependencies on sequential data. LSTM models have been used to predict ridership in public transport areas [27, 28]. For example, a model called NAPC (Neural Algorithm Passenger Counting) is presented [11], based on an LSTM architecture for counting passengers in public transportation during boarding and alighting operations. The model exploits the features of LSTM architectures for long-term event memory in time series. By leveraging 3D LiDAR data, this model improves accuracy in tracking boarding and alighting passengers, offering an average relative error of 3%. However, its reliance on LiDAR technology makes it costly and challenging to deploy in large-scale, multi-node environments such as transport interchanges.

Inspired by advances in natural language processing, Transformers have been adapted for computer vision tasks and time series prediction in public transportation. A study utilized a Transformer model based on LSTM to estimate the flow of passengers in transfer corridors between integrated hubs in an urban agglomeration [29]. This approach demonstrated high adaptability and good performance in passenger flow prediction, assisting in the efficient management of multimodal transportation systems.

Deep learning-based architectures employed in intelligent transportation systems are evolving towards end-to-end systems [30–32]. This approach integrates all workflow stages into a single architecture: data collection, preprocessing, training, inference, and evaluation. For example, the latest end-to-end architectures for autonomous driving [33] consolidate all traditional phases of perception, localization, navigation, and control into a unified system that receives input from the vehicle's sensors and generates control actions for the vehicle's manoeuvring elements [34].

Edge Artificial Intelligence (Edge AI) is transforming passenger counting systems in public transportation by enabling data processing directly on local devices such as cameras, sensors, or onboard computers. This localized approach reduces latency, enhances reliability in areas with limited connectivity, and addresses privacy concerns by keeping sensitive data on-device. By analyzing data at the source, Edge AI facilitates real-time decision-making, improves operational efficiency, and enhances passenger safety and comfort. The integration of Edge AI into public transit systems represents a significant advancement in creating smarter, more responsive urban mobility solutions.

For instance, Parquery [35] developed a camera-based passenger counting system for Swiss Federal Railways (SBB CFF FFS), achieving 98% accuracy by analyzing video streams from existing onboard security cameras using AI algorithms. These cameras are standard RGB security devices primarily intended for surveillance, and the inference is performed centrally on cloud or on-premise servers. A recent study [36] developed a passenger counter for bus stops using YOLOv3 executed on a Maixduino board (RISC-V K210 microcontroller with AI accelerator). This low-cost device performs real-time person detection at the edge, demonstrating satisfactory accuracy in crowd estimation. On the other hand, specialized companies like Outsight [37] or Amorph Systems [38] offer Edge AI platforms for LiDAR, capable of converting 3D data into flow metrics instantly, allowing the measurement of queues and people flows in airports and other transportation environments. In urban transportation, passenger counting manufacturers like DILAX [39] have introduced sensors on bus and train doors to classify objects and count people with up to 98% accuracy.

Although these studies have made significant advances in passenger detection and counting in specific scenarios, they face limitations in adaptability and scalability across various transportation environments. Most of these approaches are designed for particular contexts and do not offer a comprehensive and flexible solution for measuring passenger flow in different areas of a transport hub.

3 Materials and methods

This study evaluated the implementation and performance of an advanced passenger flow measurement system in a multimodal transportation environment. The case study focused on the Moncloa transport interchange in Madrid, a critical transportation node integrating metro services with urban and interurban buses. The following sections detail the materials used and the methods applied in the research. The end-to-end solution is described, detailing the system's sensing elements, the central processing unit, the proposed counting method, the deep learning techniques applied, and finally, the case study.

3.1 End-to-end passenger flow measurement system

An end-to-end system based on deep learning and computer vision techniques was implemented to measure passenger flow. This solution employs a distributed computer vision system to (i) capture images in various areas of the transport interchange, (ii) perform Edge AI based preprocessing, and (iii) send the local passenger flow state to a local processing server for post-processing. The system is based on a distributed architecture, which uses intelligent sensor nodes to capture and process real-time passenger movement data. A local processing server integrates and processes the data received from the sensor nodes. The processing server identifies whether each movement corresponds to an entry or exit and sends the final processed information regarding passenger count and flow to the database.

Overall end-to-end architecture for measuring passenger flow in any transit area is shown in Fig. 1. Local movement data from each node is post-processed on the local processing server by dedicated instances for each measurement zone. This post-processing results in a simplified counting data, representing the number of passengers crossing in each direction within the monitored areas. This data is then sent to the transport interchange management system.

3.1.1 Sensor nodes and sensor node groups

In our proposed solution, sensor nodes are responsible for the system's heaviest workload. As depicted in Fig. 1, each node integrates image acquisition, deep learning-based passenger detection, passenger tracking, and lightweight script-based processing.

Each sensor node is equipped with an OAK-1 PoE camera, featuring an IMX378 sensor with a maximum resolution of 4056x3040 pixels and a processing capacity of 4 TOPS (Tera Operations per Second), of which 1.4 TOPS are dedicated to AI inference. This enables real-time execution of AI algorithms directly at the capture site (Edge AI [40]). In addition to image acquisition, this model supports deep learning-based object detection, object tracking and lightweight Python script execution. This reduces the computational load on the local processing server, ensuring a scalable and efficient implementation.

The processing pipeline at a sensor node involves the following steps:

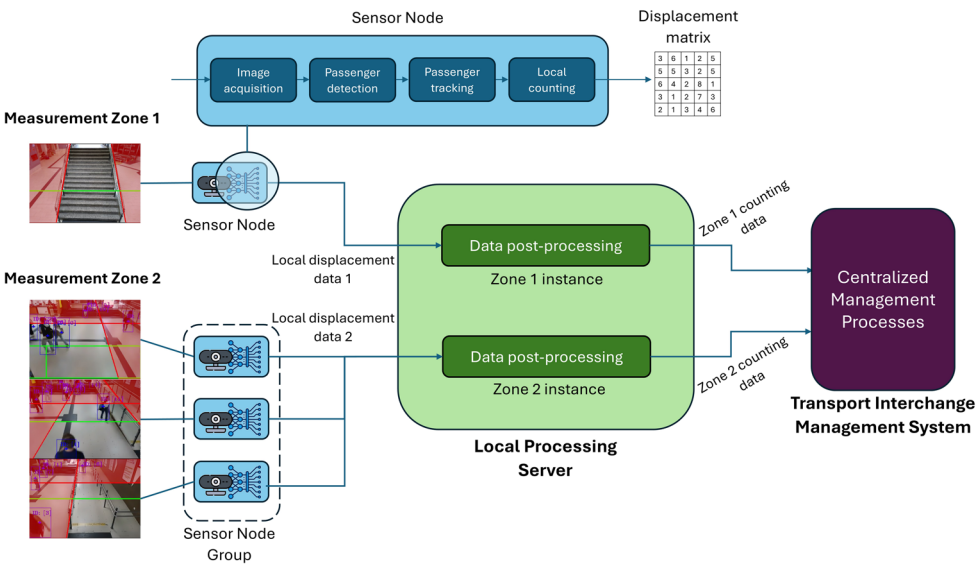


Fig. 1 Proposed end-to-end architecture for measuring passenger flow in any transit area. Each sensor node integrates image acquisition, passenger detection (detector module), passenger tracking, and local counting, whose outputs are post-processed at the local processing server before being forwarded to the transport interchange management system

1. Image acquisition: Continuous capture of RGB images from the monitored area.
2. Passenger detection: Passengers are identified within the node's field of view using a lightweight deep-learning object detection model.
3. Passenger tracking: Detected passengers are tracked across various areas within the node's field of view using an object tracker.
4. Movement detection: The field of view of the node is subdivided in different areas, facilitating thereby the recognition of passenger movements within the local environment of the node.

In more complex scenarios or those with higher passenger traffic, multiple sensor nodes can be grouped into a sensor node group (as shown on the left side of Fig. 1). This approach enables coordinated management of larger areas, ensuring complete coverage while avoiding duplication in passenger detection and counting.

The sensor nodes and sensor node groups provide as output passenger movement data in a local reference system unique to each sensor node.

3.1.2 Local processing server

Once the visual data is gathered by the sensor nodes and partially processed to obtain local movement data, it is transmitted to the local processing server. The server plays a critical role in the system by (1) connecting and configuring the sensor nodes, (2) integrating the collected data from individual nodes within a group and, (3) processing local displacements to generate the final passenger counts for each measurement zone.

The refined data is sent to the transport interchange management system and used in centralized management processes. This data can be stored in a local or cloud-based database, used for route optimization and planning, displayed on information screens within the transport hub, or even used to trigger alerts for the operators in the event of critical situations.

In our implementation and testing, an industrial-grade PC with 8 PoE ports was used as the local processing server, along with two PoE switches. However, the low computational load required allows for the use of lightweight computing devices in combination with the correspondent PoE switch networking.

3.1.3 Proposed counting method

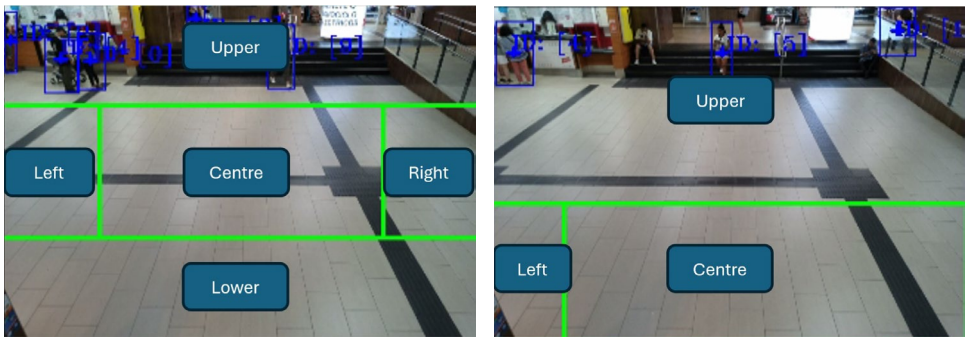
This subsection details the proposed method, divided into three parts: camera field of view subdivision, displacement detection, and entry/exit counting.

Camera field of view subdivision To detect passenger movements accurately, we employed an approach based on subdividing each sensor node's field of view into multiple areas. These areas are used to track and categorize passenger movements across the observed measurement zones. This subdivision is configured and adjusted based on the environment, allowing areas to be reduced or collapsed if necessary to simplify the analysis.

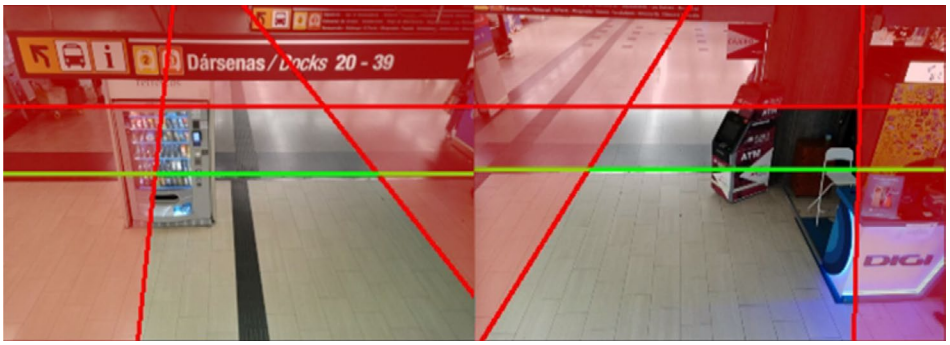
In the basic field of view subdivision template, each sensor node divides its field of view into five areas: upper, lower, left, centre and right (Fig. 2(a)). This template is conceived as a configurable design parameter rather than a fixed structure, and the number of areas can be reduced to the minimum required for accurate movement estimation. The subdivision defines the resolution at which movements are measured: more areas allow finer discrimination of directions, while fewer areas provide a simpler configuration without affecting the consistency of the calculations. In practice, most use cases require only two or three areas, while the full five-area configuration is reserved for more complex flows. Figure 2(b) illustrates a specific use case where the subdivision has been customized for a particular zone, adjusting the size and number of areas according to the environment. In that case, the left area of the node is used to handle the non-vertical movement, where upper and left areas represent the same movement direction.
As the camera field of view can contain non-interest places, the addition of restriction measurement areas is also supported. The use of these restriction areas is also intended to avoid measurement errors due to overlapping fields of view from neighbouring cameras (Fig. 2(c)).

When two nodes cover adjacent areas, it is essential that their subdivision configurations are consistently aligned to avoid discrepancies in tracking passengers moving horizontally from one node to another. As shown in Fig. 2(c), both nodes must be perfectly aligned in orientation, and their areas should maintain the same vertical size. This ensures that any movement of passengers between nodes is accurately captured in the corresponding area without duplication or omission in the count. To maintain same camera orientation, fixed angle supports were used during sensor installation procedure. With this configuration, strict temporal synchronization is not required, as transitions between nodes are consistently resolved by the spatial alignment of the subdivided areas.

To ensure the proper functioning of the restriction areas, the delimiting lines are positioned on a plane parallel to the ground at an average height of 0.825m, corresponding to the centre of the passengers' bounding box, based on an average person height of 1.65m. This approach ensures smooth transitions for lateral passenger movements between nodes.



(a) Standard subdivision (b) Customized subdivision



(c) Restricted measurement areas

Fig. 2 (a) Representation of the five-area subdivision template shown in green lines. (b) Specific area subdivision for a particular use case. (c) Adjacent cameras using restricted measurement areas (shaded in red)

Table 1 Example of a node origin–destination counting matrix. Each row represents the area where a passenger was first detected, and each column corresponds to the area where the passenger was last observed

Origin \Destination	Up	Down	Left	Centre	Right
Up	0	0	0	0	0
Down	0	0	0	0	0
Left	0	0	0	1	0
Centre	0	0	0	0	1
Right	0	0	0	0	0

In this example, the matrix encodes one displacement from the left to the centre area and another from the centre to the right area

A calibration procedure was carried out using sticks marked at 0.825m to delimitate restriction areas within each camera field of view.

Displacement detection For each node, all movement data is aggregated within a fixed time window into a 5×5 origin–destination matrix (M), where rows represent the area of the first detected position and the columns match the area of the final position. Each matrix cell represents the measured number of passenger displacements throughout the defined areas. This data is sent from the node to the local processing server. Inference speed becomes critical as low frame rates can lead to passenger disappearance without detecting its transit through a specific area, especially in the boundaries. An example of this counting matrix can be found in Table 1, representing one displacement from the left to the centre area and another one from the centre to the right area.

Entry and exit counting Once the nodes have transmitted the origin–destination matrices M to the local processing server, the server derives entry/exit events using the pre-assigned role of each area ("In", "Out", or "Unassigned"). Each time a passenger moves from one area to another within a node's field of view, the server evaluates the origin and destination roles and increments the corresponding counter:

- **Entry:** Incremented when a passenger moves from an "Out" or "Unassigned" area to an "In" area, indicating the passenger has entered the place of interest.
- **Exit:** Incremented when a passenger moves from an "In" or "Unassigned" area to an "Out" area, signifying that the passenger has left the monitored place.
- **None:** Movements into "Unassigned" areas are ignored for counting.

Formal modelling For each sensor node, M[i,j] stores the number of movements from area i to area j. Based on the

predefined area roles, passenger entries (E) and exits (X) are then computed according to (1) and (2), respectively.

$$\mathrm{E} = \sum_{\substack{i \in \{Out, Unassigned\} \\ j \in \{In\}}} M[i,j] \tag{1}$$

$$\mathrm{X} = \sum_{\substack{i \in \{In\} \\ j \in \{Out, Unassigned\}}} M[i,j] \tag{2}$$

This process ensures that only relevant movements are counted as entries or exits, minimizing unnecessary data accumulation and enhancing the system's accuracy.

3.1.4 Deep learning models for passenger movement detection

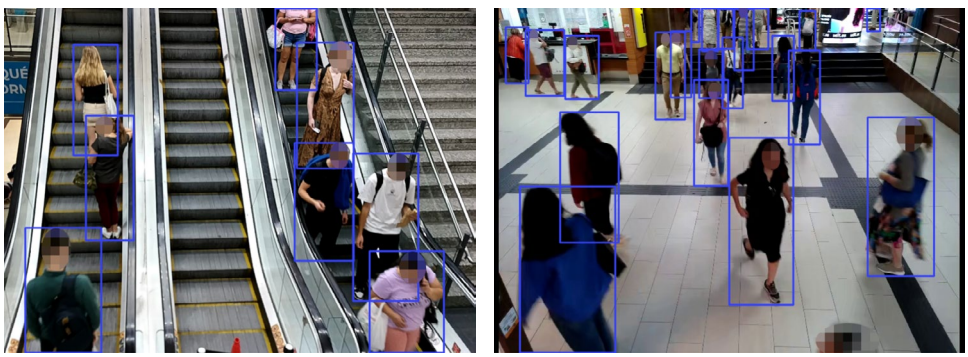
The core part of the end-to-end system is the deep learning model used in the sensor nodes. The model's average precision and inference time are key factors determining overall sensor performance. The selected OAK-1-PoE cameras are only compatible with the ".blob" MyriadX model format, which requires any used model to be converted into this format. The camera processors achieve 1.4 TOPS of AI processing power, restricting the use of medium and heavy models. It also supports built-in object trackers, including trackers such as Short-Term KCF Tracking (Kernelized Correlation Filter), Non-Visual Short-Term Tracking, Zero-Term Imageless Tracking and Colour Histogram Tracking [41–43].

In this work, pre-trained and custom-trained passenger detection models were compared. Using custom-trained deep learning models is highly beneficial, specifically in interiors where camera and light conditions are consistent. However, a comparison was conducted to identify whether the differences in accuracy are significant enough to justify the additional effort required to generate and label the dataset and train the custom models. Custom models can be trained for the defined working distances and optimized for the most common passenger sizes relative to the camera image at measurement zones. On the other hand, pre-trained models provide a fast implementation with high generalization for detection tasks.

The training process requires a large amount of data to be captured and labelled following an object detection format. This process can be carried out in two ways: manual or semi-automated. Semi-automated labelling can be performed using a pre-trained model to generate temporary bounding boxes requiring subsequent human adjustment. However, the data collection and labelling process in both methods can be time-consuming, and the amount of needed data and computational resources are not available for everyone. In those cases, the implementation of pre-trained models is highly beneficial. The selected light-weight person detection pre-trained models were MobileNet SSD [44, 45] and Person Detection Retail 0013 [46], both obtained from the OpenVINO toolkit [47].

Datasets Two distinct custom datasets were recorded for the specific scenarios analysed in this work. One dataset was acquired for the escalator access case study, while another was collected for wide corridor scenarios. Additionally, a mixed dataset combining data from both scenarios was generated to evaluate whether a more generalized model offers advantages over the scenario-specific ones.

The first escalator dataset consisted of 432 manually labelled images from an escalator access scenario with resolutions of 1920x1080, as illustrated in Fig. 3(a). The photos were taken from 4 installed sensor nodes at different heights, two of them at 3.8m height with at 65o inclination and a 4.1m distance, while the rest of the cameras were installed at a 2.9m height, 65o inclination, and 2.3m distance from the measurement section. Because of the labelling criteria, the generated bounding box had to cover the whole target using the smallest size possible for highly visible passengers. On the other hand, highly occluded persons had their bounding box cropped at the height where most of



(a) Escalator Dataset (b) Corridor Dataset

Fig. 3 Example frames from the custom datasets: (a) Escalator dataset and (b) Corridor dataset

the body is occluded. The dataset was shuffled and then subdivided into three sets: train (70%), validation (15%), and test (15%). Due to the low performance on crowded passenger flows, a second set of data was captured and labelled, including data with high passenger density. The additional data contained 63 images from the same cameras taken at crowded moments and subdivided using the same proportions as the first dataset, which resulted in a second version of the dataset with a total of 495 images containing 1893 passengers. All the extra images were labelled using semi-automatic labelling procedures, applying manual labelling adjustment to the pre-processed labels generated by a model trained on the first 432 images. Initially, more images were taken to expand the dataset; however, only 63 images were selected to be included in the extended dataset because the first model performed poorly on them.

A different dataset was acquired from wide corridor scenarios, collecting 178 images from eight sensor nodes in two wide corridors (Fig. 3(b)). The first corridor is 17.6m wide, requiring five nodes to cover the entire section. Installed cameras leverage the presence of columns to reduce overlapping areas at the measurement zones. Nodes were located at a 3.1m height, and columns left spaces of 3.9m, 6.6m, and 6.6m wide between them. In this scenario, most cameras used a 60o inclination, whereas the camera covering the 3.9m wide section used 62.5o to allow full coverage. While the original intention was to locate the cameras from 2.25m to 2.5m distance to the measurement section, local restrictions forced camera placement at 4.15m from the measurement section. The second corridor is 10m wide, where 6.8m of the measurement section is shared between two cameras and the remaining space is covered by a single node due to the presence of a ramp. The three cameras of this sensor node group located in this corridor had a height of 2.7m, a 60o inclination, and a distance to the measurement section of 1.56m. The first image set was expanded with 98 semi-automatic labelled images, resulting in a total size of 276 images containing 1297 passengers.

Model architecture Selecting a deep learning architecture to solve an object recognition problem is a complex task that depends on multiple variables, such as the need for real-time processing, the size of the available dataset, the computational capacity of the system, the accuracy required, and existing implementations.

Object recognition architectures fall into two main categories: (1) Two-Stage Detectors (e.g. R-CNN, Faster R-CNN and Mask R-CNN); (2) One-Stage Detectors (e.g. YOLO, SSD and RetinaNet). In this research work, single-stage detectors were chosen due to their suitability for real-time tasks. These architectures perform bounding box prediction and classes in a single pass of the network, which optimizes speed. Within this category, the YOLOv5, YOLOv8 and YOLOv11 [48–50] versions were specifically selected. This choice is justified because, chronologically, each version incorporates significant improvements in its design and capabilities. All the selected architectures have implementations ('n', 's', 'm', 'l', and 'x') with different numbers of parameters, which has facilitated their deployment in Edge AI devices, maximizing the flexibility and applicability of the system.

Given the deployment constraints previously outlined (MyriadX .blob compilation and the need to sustain a minimum on-node frame rate to maintain tracking continuity), we restricted the search to detectors that are both compatible with the OAK-1 PoE sensor and capable of achieving sufficient frame rate. While recent transformer-based detectors (e.g., DETR, Deformable-DETR, RT-DETR) perform well on GPU, they are not supported on MyriadX and therefore fall out of scope for on-node evaluation in this work.

Mdels in this work were trained using [48–50] the Ultralytics v8.3.51 framework, which implements the YOLO family using the PyTorch v2.1.1 library. Using this framework each YOLO model is initialized from COCO-pretrained weights provided by Ultralytics and fine-tuned on our datasets. The training was conducted in a NVIDIA RTX3070 with 8 GB of VRAM for 1000 epochs with a batch size of 32, image resolution of 320x320 pixels, initial learning rate of 0.01, momentum of 0.937, 3 warmup epochs, and optimizer set to auto. The learning rate followed the default linear decay schedule in Ulatralytics, decreasing from the initial value (0.01) to the final learning rate across training. An early stop criterion was applied, terminating training if no improvement was observed over the last 100 epochs. Additionally, data augmentation techniques were applied during the training process to improve model generalization. The augmentation criteria included HSV variations with a 1.5% maximum HUE change, 70% for saturation and 40% maximum brightness variation. Other modifications included image translation in both axes up to 10% of the image size, scale variations up to 50%, a 50% chance of horizontal flipping, and mosaic augmentation.

3.2 Case study: moncloa transport interchange

The case study is located at the Moncloa transport interchange in Madrid, Spain's capital. The Moncloa transport hub is a major transportation interchange that integrates metro services with urban and interurban buses.

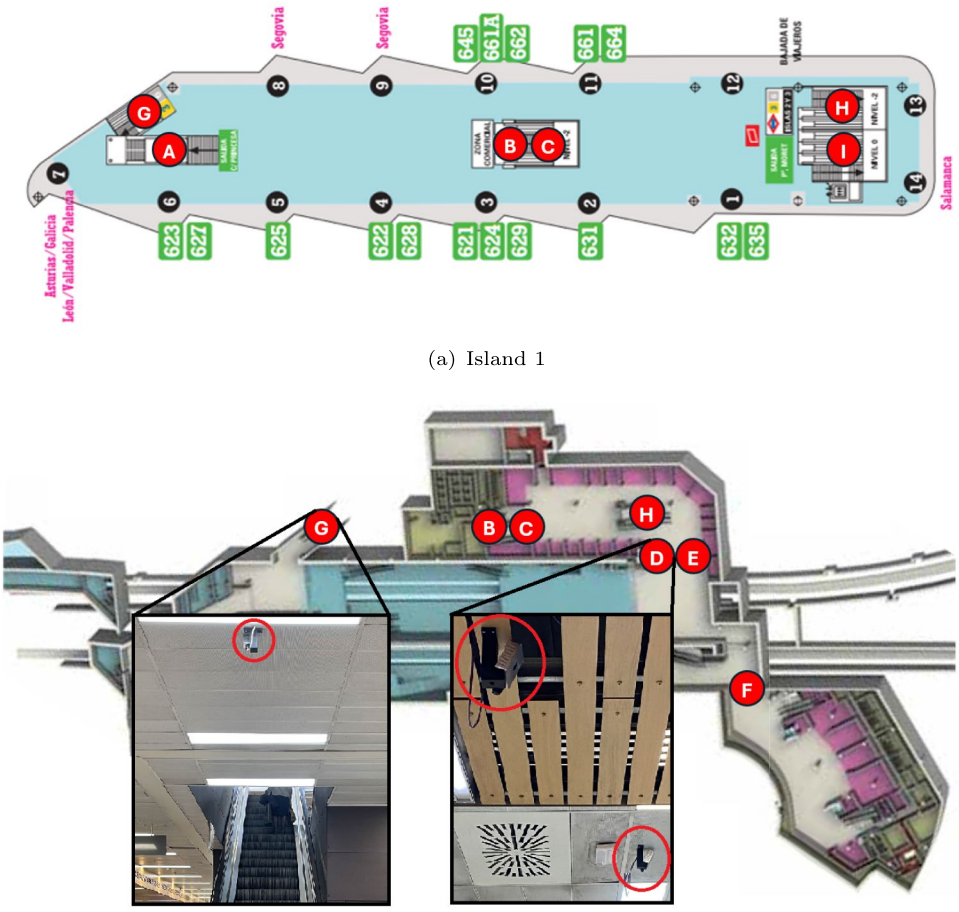
With an area of 46,000 m2, 34 bus platforms, 32 shops, 1,500 underground parking spaces, and 280,000 passengers circulating daily, this transport hub has become one of the most important in Madrid due to its ability to manage a large

volume of passengers each day. This makes it an ideal testing environment for this study.

This transport interchange is divided into three islands and several underground levels, featuring extensive facilities, including:

- Metro Platforms: The interchange connects with two main metro lines of the city. The 115-meter-long boarding and alighting platforms accommodate trains of up to six cars.
- Bus Terminals: The bus station is located on level -1, at an elevation of 651 meters above sea level. It has 15 bays, each with a 12-meter-long parking space. The terminals handle urban and interurban buses, approximately 1,500 buses daily.
- Passenger Walkways and Corridors: The interchange features wide corridors and walkways that allow passengers to move smoothly between different areas, minimizing waiting time and improving accessibility. The passenger corridors have ceiling heights ranging from 2.8 to 3.2 meters and widths varying from 3.3 to 17.6 meters.
- Stairs and Escalators: To facilitate access between different levels of the hub and efficiently manage vertical passenger flow, the hub includes both escalator and fixed stairs:
  - Escalators: The escalators have a 30-degree incline and a total width of 1.59 meters.
  - Fixed Stairs: These stairs have a 30-degree incline, widths ranging from 2.00 to 3.15 meters, and ceiling heights varying from 2.8 to 4.8 meters.
- Commercial and Recreational Areas: The interchange includes 1,500 m2 of commercial areas offering shops, restaurants, and other services, enhancing the user experience and making use of waiting times.

The proposed end-to-end system has been tested on Island 1 (Fig. 4(a)) of the Moncloa transport interchange and its lower level (Fig. 4(b)). Island 1 has five access points: two leading to the exterior, one to the Metro platforms, and two connecting the island to the lower level. The lower level has three access points: two connecting to Island 1 and one



(a) Island 1

(b) Lower level

Fig. 4 Deployed sensor nodes at the Moncloa transport hub in (a) Island 1 and (b) the lower level. A, C, and G correspond to single nodes on stairs/escalators, B, H, and I to node groups on stairs/escalators, and D, E, and F to node groups in wide corridors

leading to a connection with the Metro and the rest of the hub. The access points to Island 1 include wide corridors and stairs or escalators, posing a challenge due to the horizontal and vertical movements of passengers and the potential for significant occlusion in the field of view of the cameras.

As illustrate in Fig. 4, the sensor nodes (represented in red and labeled with letters A–I) were strategically placed at key points in the transport hub to maximize coverage and minimize occlusion. Specifically, two area types were addressed:

- Wide Corridors: Due to their size, these presented significant challenges, with ceiling heights between 2.8 and 3.2 meters and widths ranging from 3.3 to 17.6 meters.
- Stairs and Escalators: They present a 30-degree incline and widths varying from 2.00 to 3.15 meters for fixed stairs and 1.59 meters for escalators.

Although 17 nodes were installed across the nine access points of the transport hub shown in Fig. 4, only eight sensor nodes were used to evaluate system accuracy in this work. For the stair/escalator scenarios, two groups of nodes, each consisting of 2 cameras, were assessed in areas H and I. In contrast, areas D and E were used to evaluate wide corridor scenarios. Area D consisted of a group of 2 nodes, while Area E had a group of 3 nodes, of which only two were used for the system evaluation.

4 Results and discussion

4.1 Metrics

This section is divided in two parts: metrics for object detection models and selected criteria for determining overall system accuracy. The evaluation process of object detection models involves the use of precision-recall based metrics [51]. First, precision (3) is computed as the ratio of true positive detections over the total number of detections for a specific label. On the other hand, recall (4) is defined as the proportion of objects from the target class identified by the model. While precision analyses the accuracy of predictions, recall represents the model's capacity to identify positives.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

- TP (True Positive): Number of detections correctly classified as the target class.
- FP (False Positive): Number of detections being misclassified as the target class.
- FN (False Negative): Number of undetected target class objects.

Due to the trade-off between precision and recall, a robust model requires the consideration of other metrics derived from both. The Precision-Recall curve is a graphical representation of the trade-off between precision and recall, which shows when they are unbalanced. When the curve maintains high levels of precision across all levels of recall, it is considered an indicator of model robustness. Since different threshold values can be used to consider a detection valid, multiple precision-recall curves are usually drawn with different thresholds. Average Precision (AP) is a single-value metric that summarizes precision-recall quality for a specific model class. However, the most used metric in object detection is the Mean Average Precision (mAP), which extends the concept of AP to all classes supported by the model, resulting in an overall measure of the model's quality. It is very common to encounter this metric as mAP@0.5, representing that every AP has been computed using an IoU (Intersection over Union) threshold of 0.5. This value indicates that a detection is correct if the overlapping area between the ground truth and the detection bounding boxes is higher than 50% of their combined area.

Another widely used metric that combines precision, and recall is the F1 score. This metric is defined as the harmonic mean of precision and recall values, as shown in (5). The F1 score is considered reliable only when the different classes in a dataset are balanced. The F1 score is a particular case of a more general function called F_β, defined in (6)). Higher β values in F_β provide more weight to the recall value in the function, whereas lower β values give higher importance to precision. This metric is not restricted to evaluating deep learning models; it can be applied to more complex systems.

$$F_1 = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{\beta^2 \cdot \text{Precision} + \text{Recall}} \tag{6}$$

The first step for computing overall system performance is gathering ground truth data. The ground truth data acquisition process involved manual person counting in the whole access; for that matter, the counting was performed using a push-button-based smartphone app, which saves the push event time. As the installed system saves a single (in, out) vector with all gathered data in the same period, the ground

truth data is then transformed to the same system. Data comparison between ground truth and measured data for every minute is not a good practice as the time instant when the system increments the counter can be very different from the ground truth time. Occasionally, the tracking algorithm considers a tracked person lost and increments the counter several seconds after the passenger has disappeared from the image. Therefore, the relative error is computed by comparing gathered data for larger periods.

For each sensor node or sensor node group (in case of wide corridor scenarios), the measured passenger flow (P_M) is compared to ground truth data (P_{GT}) in both directions individually, obtaining the absolute error $\Delta P_i = |P_{GT,i} - P_{M,i}|$. Accuracy values shown in this work have been computed using (7), where i refers to any of both directions in any referenced camera or camera group.

$$\text{Accuracy} = 1 - \frac{\sum_i \Delta P_i}{\sum_i P_{GT,i}} \tag{7}$$

4.2 Quantitative results

Custom model training was performed in two stages. In the first stage, a YOLOv8 model of 's' size was trained for each scenario over the first iteration of both datasets with image sizes of 320x320. The escalator model achieved 97.7% mAP@0.5 on the test set; however, precision decreased during crowded moments. Additional images were taken during crowded moments and labelled using a combination of the capabilities of bounding box prediction of the first model and human adjustment. For the same set of 20 images, this semi-automated procedure took 25.3 seconds per image, while full human labelling took 42.8 seconds per image. The same procedure was performed for the corridor dataset, where the initial model achieved 97.4% mAP@0.5 on the test set of the first iteration of the corridor dataset. Additional data was also added to address the low accuracy of the model during crowded moments, which is more critical in this scenario, where occlusion has a significantly higher impact. For this dataset, semi-automated labelling took 28.3 seconds per image, while manual labelling took 44.3 seconds per image. As a result, manual labelling required an average of 43.6 seconds per image, while semi-automated labelling required an average of 26.8 seconds per image, saving 38.5% of labelling time.

Table 2 Results of YOLOv8 custom-trained models over test set

Dataset	Model variant	Image size	mAP@0.5	Average framerate
Escalator	n	640×640	0.987	8.6
Escalator	s	320×320	0.984	13.3
Corridor	n	640×640	0.970	8.2
Corridor	s	320×320	0.972	13.1
Mixed	n	640×640	0.981	8.0
Mixed	s	320×320	0.976	13.1

The second training stage addressed model training on extended datasets. Several YOLOv8 models were trained, using 'n' and 's' model sizes with image sizes of 640x640 and 320x320, respectively. Both architectures were used for the escalator, corridor, and mixed datasets containing all 699 images, resulting in a collection of six custom-trained models. No YOLOv8 architectures beyond 's' size were used to maintain a minimum framerate in sensor nodes, allowing consistent tracking. This test is performed to obtain accuracy and inference speed data to help in the selection of the most appropriate model family.

Table 2 represents custom-trained models, including test set average precision and their framerates using Zero-Term Imageless Tracking. Framerate tests showed that 'n' variants suffered higher FPS drops during high passenger density moments. Due to these results, only 's' size models from YOLO families have been used in the following tests.

To evaluate the performance of the proposed system over custom-trained models and selected pre-trained models, videos from different installed sensor nodes were recorded. The software has been adapted to perform simulation over recorded videos, allowing for a robust comparison of model performance on the same data. Each study case is represented using four nodes divided into two groups of 2 nodes each. For the wide corridor study case, both sensor nodes from each group share an overlapping area. The videos were recorded during both crowded and non-crowded times, each one two minutes long. In the sensor node groups the videos were recorded synchronously. In total, this test dataset comprises 16 video clips (4 nodes, 2 traffic conditions, 2 study cases), each lasting 120 seconds at 30 FPS with 1080p resolution.

Table 3 evaluates the performance of different trackers on each video using accuracy metric. Each video was processed using the custom-trained YOLOv8s model trained on the dedicated dataset for each location. A graphical representation of the results is provided in Fig. 5.

Simulation results show a similar performance across different trackers; however, the differences are more noticeable in corridor scenarios. The colour histogram tracker achieves slightly higher average accuracy than non-visual short-term and zero-term imageless tracking. For that reason, the colour histogram tracker has been selected as the tracking algorithm, and the following simulation results will include it. Once the most suitable tracker has been chosen, a similar process must be performed to identify the best passenger detection model for each study case.

Table 4 offers an accuracy comparison between selected models in both study cases, evaluated during crowded and

Table 3 System accuracy at different locations in crowded and non-crowded times using YOLOv8s scenario-specific models

Tracker	Escalator		Corridor		Weighted
	Non-Crowded	Crowded	Non-Crowded	Crowded	average
Non-Visual Short-Term	**0.9889**	0.9755	0.9237	0.9154	0.9543
Zero-Term Imageless	0.9833	0.9837	0.9322	**0.9353**	0.9624
Colour Histogram	0.9833	**0.9878**	**0.9407**	0.9303	**0.9637**
Passengers	180	245	118	201	744

Tracker algorithm comparison

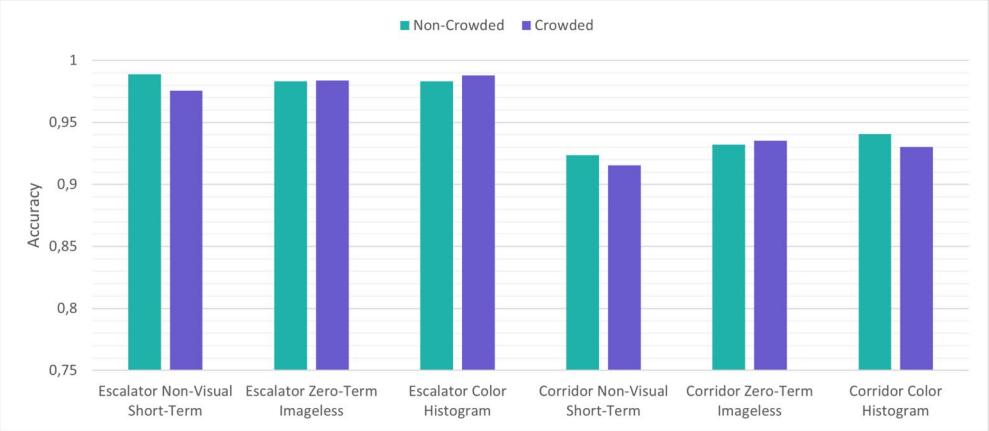


Fig. 5 Accuracy evolution across flow rates in escalator and wide corridor scenarios

Table 4 System accuracy at different locations in crowded and non-crowded times

Model	Escalator		Weighted	Corridor		Weighted
	Non-Crowded	Crowded	average	Non-Crowded	Crowded	average
Person Detection Retail-0013	0.8564	0.7796	0.8121	0.8814	0.8905	0.8871
MobileNet-SSD	0.5028	0.4980	0.5000	0.8983	0.7612	0.8119
YOLOv5su-Escalator	0.9779	0.9878	0.9836	–	–	–
YOLOv5su-Corridor	–	–	–	0.9407	**0.9602**	0.9530
YOLOv5su-Mixed	0.9945	0.9878	0.9906	0.9407	0.9453	0.9436
YOLOv8s-Escalator	0.9890	0.9878	0.9883	–	–	–
YOLOv8s-Corridor	–	–	–	0.9407	0.9303	0.9341
YOLOv8s-Mixed	**1.0000**	0.9878	**0.9930**	0.9746	0.9353	0.9498
YOLOv11s-Escalator	0.9834	**0.9918**	0.9882	–	–	–
YOLOv11s-Corridor	–	–	–	0.9576	0.9303	0.9404
YOLOv11s-Mixed	0.9834	0.9837	0.9836	**0.9915**	0.9403	**0.9592**
Passengers	180	245	–	118	201	–

Deep learning detection model comparison

non-crowded periods. The results indicate that the best pre-trained model, Person Detection Retail 0013, achieves an accuracy of 0.8121 in escalator scenario and 0.8871 in the corridor scenario. However, custom-trained models significantly outperform the best pre-trained model. During non-crowded periods, the best models trained on the mixed dataset show better results than the best models trained on the specific data, suggesting that greater model generalization enhances performance across both scenarios. However, during peak hours, scenario-specific models demonstrate superior performance compared to more generalized models. On average, mixed models are more accurate than specific ones. Among the tested models, YOLOv8s Mixed is the most reliable model for escalator scenarios, achieving 0.9930 average accuracy. Meanwhile, YOLOv11s Mixed offers best results in the most challenging scenario, wide corridors, with an average accuracy of 0.9592.

In Fig. 6 we compare the evolution of accuracy across different flow rates for both scenarios, using the models that achieved the highest average accuracy. In wide corridor scenario, the increase in passenger flow has significantly greater impact than that observed in the escalator scenario. The accuracy drop in wide corridor scenario is mainly attributed to the interface between the adjacent sensors. In most failures passenger displacement was not detected within the domain of any camera.

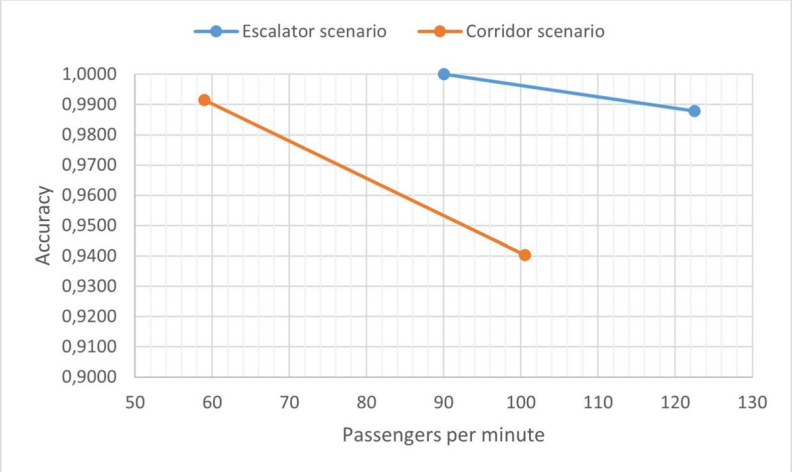
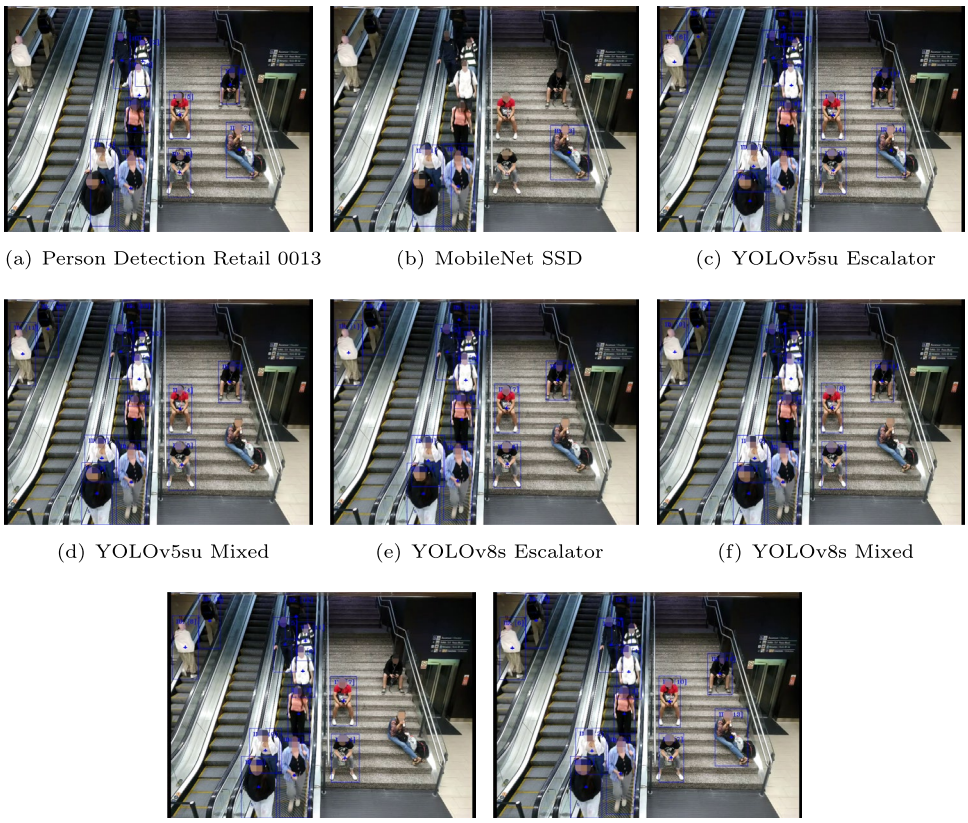


Fig. 6 Accuracy evolution across flow rates in escalator and wide corridor scenarios



(a) Person Detection Retail 0013 (b) MobileNet SSD (c) YOLOv5su Escalator

(d) YOLOv5su Mixed (e) YOLOv8s Escalator (f) YOLOv8s Mixed

(g) YOLOv11s Escalator (h) YOLOv11s Mixed

Fig. 7 Qualitative results per model at escalator scenario

To complement the model evaluation, we also quantified the hardware resource overhead of the OAK-1 PoE. In the escalator scenario with YOLOv8s-Mixed, the device operated at a stable temperature of 55.9 °C, with LEON CPU usage of ≈26.3% (CSS) and DDR occupancy of ≈36.6% (122.1/333.3 MiB). These results indicate a moderate computational load and memory footprint, supporting the feasibility of continuous deployment in transport hubs.

4.3 Qualitative results

To complement the quantitative analysis, Figs. 7 and 8 present qualitative examples for the escalator and corridor scenarios, respectively. In both cases, results from the eight evaluated models are displayed for the same representative frame. These comparisons illustrate how pre-trained models consistently underperform compared to our custom-trained models. In the escalator scenario, the selected frame includes people seated on the steps. Although this situation

(a) Person Detection Retail 0013 (b) MobileNet SSD (c) YOLOv5su Mixed

(d) YOLOv5su Corridor (e) YOLOv8s Mixed (f) YOLOv8s Corridor

(g) YOLOv11s Mixed (h) YOLOv11s Corridor

Fig. 8 Qualitative results per model at corridor scenario

is not critical for flow measurement, since seated individuals are not moving, custom-trained models sometimes fail to detect them. Nevertheless, all passengers in motion are correctly identified. In the corridor scenario, the chosen frame depicts a complex scene with several individuals, including a woman pushing a stroller. Some models erroneously classify the stroller as a person or miss certain detections. The best-performing model in this scenario (YOLOv11s Mixed) successfully detects all passengers but also produces a false positive for the stroller. These examples highlight both the robustness of the proposed models and the residual limitations under uncommon or ambiguous visual conditions.

4.4 Discussions

A comparison with other proposed passenger counting systems is performed in Table 5. Our final metrics were calculated as weighted averages across non-crowded and crowded scenarios. In addition to the accuracy metric, the F1 score was also calculated to perform a more appropriate comparison with other works. As the proposed systems in the literature focus on different locations with various spatial conditions, a simple metric review is insufficient to compare them. Works covered in this table have been analysed using precision, recall, accuracy and F1 score metrics. Proposals [23], and [21] cover vertical camera points of view for a mono-camera approach, suitable for small entrances or narrow corridors. Authors of [26] measure passenger flow

Table 5 Comparison with other passenger counting systems

Work	Scenario	Passengers	Sensor	Precision	Recall	F1 Score	Accuracy
[23]	Narrow corridor	100	Single-Camera	–	–	–	0.9900
[52]	Outdoors	5075	2D-LiDAR	0.9935	0.9829	0.9882	–
[21]	Corridor	300	Single-Camera	–	–	–	0.9326
[26]	Metro screen doors	13733	Single-Camera	0.9715	0.9797	0.9756	0.9916
[24]	Bus interior	37	Single-Camera	0.8810	0.8605	0.8706	–
Ours–Escalator	Escalator	426	Multi-Camera	**1.0000**	**0.9930**	**0.9965**	**0.9930**
Ours–Corridor	Wide corridor	319	Multi-Camera	0.9902	0.9498	0.9696	0.9592

Results are based on the values reported in the corresponding publications. When a metric was not explicitly provided, it was derived from the available data, for example the F1 score was calculated from the reported precision and recall

congestion at metro platform screen doors using surveillance video analysis and entropy-based evaluation of passenger trajectories. In outdoor scenarios, a single sensor can monitor wide areas when located at high altitudes. The work [52] achieves a high F1 score using 2D LiDAR in outdoors environments. On the other hand, [24] measures passenger flow inside bus transport vehicles.

As presented in Table 5, our results show an F1 score of 0.9965, and an accuracy of 0.9930 for the escalator scenario, while in the wide corridor scenario the system achieves a F1 score of 0.9696 and an accuracy of 0.9592. The system variant on the escalator scenario achieves the best metrics. This performance gap can be attributed to the more constrained movements of passengers on escalators compared to the irregular trajectories in wide corridors, the occasional transitions between cameras, and the impact of the different sensor inclination angle. The more pronounced decrease observed in corridors at higher passenger flows (Fig. 6) further reflects the effect of occlusions and boundary transitions, where overlapping passengers may prevent displacements from being correctly assigned to any camera domain. Although other works present better metrics than the corridor variant, these rely on a single sensor and does not address such complex environments. The multi-camera approach in our system allows complex scenario coverage, such as wide corridors in indoor locations where ceiling height restricts the width of the field of view. Additionally, some works only report accuracy, whereas the F1 score is a more precise metric, as it accounts for the balance between false positives and false negatives, which can otherwise distort accuracy.

Despite the advantages demonstrated by our multi-camera approach, potential limitations should be considered. The calibration procedure used for defining the restriction lines that delimit boundaries between adjacent cameras relies on an assumed average passenger height of 1.65m. Consequently, passenger transitions between areas in these boundaries can remain undetected, particularly when height of individuals significantly deviates from the average. However, a quantitative assessment of the impact of height variability on error rates remains to be addressed.

Finally, the system developed in this paper provides very useful information for the planning and management of operations and services in critical mobility hubs such as multimodal transport interchanges. The Highway Capacity Manual [53] establishes six levels of the quality of service provided by transportation infrastructures, ranging from 'A' (best) to 'F'. In pedestrian corridors, stairs, and waiting areas of a transport interchange, this level of service at a given time is quantified based on their geometries and the number of passengers passing through or staying in them. This paper presents an end-to-end system that directly provides real-time pedestrian flow data; while the number of people in a space where all entries and exits are monitored is known through the following conservation law (8). In this equation p_t and p_{t-1} represent the number of occupants inside an area at times t and $t-1$, respectively, while e_t and l_t denote the number of people entering or leaving the area between times $t-1$ and t.

$$p_t = p_{t-1} + e_t - l_t \tag{8}$$

The measurements of people flows and area occupancies, along with their derived levels of service at an interchange station, can be used in real time for proactive management during daily operations. This includes providing updated information to users, guiding passenger movement with crowd control barriers, or adjusting escalator directions. Analysing the trends of these measurements also supports tactical and strategic planning for the station. Examples include optimizing the assignment of bus bays, placing services and businesses, and determining their operation based on expected passenger flows and presence.

5 Conclusions

In this paper, we have proposed and developed an end-to-end system for passenger flow measurement in indoor environments using deep learning and computer vision techniques. The proposed system has demonstrated to be scalable and flexible, achieving average accuracy rates of 95.92% and

99.30%. It has maintained 94.03% accuracy in the corridor scenario with crowded flow, which is considered the worst-case scenario. The counting method defined for the shared areas between adjacent cameras allowed the system to maintain accuracy from 94.03% to 99.15% in complex locations such as high-traffic wide corridors.

During the development of the system, the semi-automatic labelling approach has been tested, and it demonstrated to reduce labelling time by 38.5%. The semi-automatic labelling process offered the possibility to feed the improved datasets only with the data the model failed to recognize. Also, the Colour Histogram tracking algorithm outperformed other tracking algorithms. The use of lightweight custom-trained models in comparison to pre-trained models achieved significant higher performance. This improvement in accuracy, from 7.21% to 18.09%, justifies the required effort to develop custom dataset and models. Besides, conducted tests have shown that deep learning models trained on larger and generalized datasets are more suitable for the system than models trained on smaller but more specific datasets.

The end-to-end system was tested at the Moncloa Transport Interchange, where 17 sensor nodes were installed at different locations covering a wide range of spatial conditions. The acquired data demonstrated 63.666 passenger displacements in a single business day at the escalator study case and 386.207 in the same week. In the wide corridor scenario, 45.408 displacements were registered in a single business day, with the week concluding at 276.964 displacements in that section.

Compared to other existing works, our solution improves accuracy in wide walkways and in environments with multiple outputs and inputs, where techniques based on single sensors or models trained on limited data have shown difficulties. In addition, the distributed vision approach mitigates occlusion issues and variations in lighting, key aspects in transport hubs where the flow of people is unpredictable.

Potential future works stemming from this proposed system include adapting it to new types of access points or alternative viewpoints, enabling broader coverage within transport hubs. For instance, using cameras with a top-down view could be beneficial in areas where the perspectives addressed in this paper may not be as effective. Another potential direction for future research is adapting the system for use inside public transport vehicles, such as buses or trains, where modifications would be required to accommodate the system to a mobile environment. Additionally, future work will include a quantitative analysis of the impact of passenger height variability on transition errors, in order to improve calibration procedures under diverse populations.

Author Contributions Eduardo Salas Fernández: Conceptualization, Investigation Methodology, Writing – original draft, Software. Pedro J. Navarro Lorente: Conceptualization, Methodology, Resources, Supervision, Validation, Writing – review & editing. Francisca Rosique Contreras: Validation, Writing – review & editing. Juan Benavente Ponce: Visualization, Writing – review & editing. Ana María Rivadeneira Muñoz: Visualization, Writing – review & editing.

Funding Open Access funding provided thanks to the CRUE-CSIC agreement with Springer Nature. This work was supported by Agencia Estatal de Investigación (AEI) of Spain under Grant PLEC2021-007609.

Data Availability The data that support the findings of this study are not publicly available due to participant privacy and ethical restrictions. Individual-level data cannot be shared under the terms of the informed consent and institutional ethical approval.

Declarations

Competing Interest The authors have no competing interests to declare that are relevant to the content of this article.

Research involving Human Participants and/or Animals This study did not involve direct interaction with human participants or the use of personally identifiable data. All image captures were processed on-device by Edge-AI cameras, without transmission or storage of raw video outside the device, and without identifying or tracking any individual beyond a transient, anonymous count. Therefore, institutional ethics committee approval for human research was not required.

Informed Consent The sensors operated in a public space where there is no reasonable expectation of privacy and no images were recorded that would allow personal identification. Consequently, obtaining individual informed consent was neither feasible nor required. All material captured exclusively for model training was immediately anonymized, used only for bounding-box annotation, and securely deleted after labeling was complete.

Data Recording for System Performance Evaluation To quantify metrics such as precision, recall, F1-score, and overall accuracy, two-minute video recordings were made with the installed sensors during periods of both high and low passenger flow. These recordings were used solely to simulate the system and extract aggregated passenger-flow data; at no point were faces or other identifiable features manually reviewed. All videos were encrypted and stored on a local, access-restricted server and retained only for the duration of the metric computation process, after which they were permanently destroyed.

References

1. Khor N, Arimah B, Otieno R, Oostrum M, Mutinda M, Martins J, Arku G, Castán Broto V, Chatwin M, Dijkstra L et al (2022) World Cities Report 2022: Envisaging the Future of Cities, United Nations Human Settlements Programme (UN-Habitat). Nairobi, Kenya
2. Banerjee N, Morton A, Akartunalı K (2020) Passenger demand forecasting in scheduled transportation. Eur J Operational Res 286(3):797–810. https://doi.org/10.1016/j.ejor.2019.10.032
3. Monzón A, Hernández S, Di Ciommo F (2016) Efficient urban interchanges: the city-hub model. Transportation Res Procedia 14:1124–1133. https://doi.org/10.1016/j.trpro.2016.05.183. Transport Research Arena TRA2016
4. Lucietti L, Hoogendoorn C, Cré I (2016) New tools and strategies for design and operation of urban transport interchanges. Transportation Res Procedia 14:1240–1249. https://doi.org/10.1016/j.trpro.2016.05.195. Transport Research Arena TRA2016
5. Conticelli E, Gobbi G, Saavedra Rosas PI, Tondelli S (2021) Assessing the performance of modal interchange for ensuring seamless and sustainable mobility in european cities. Sustainability 13(2). https://doi.org/10.3390/su13021001
6. Abduljabbar R, Dia H, Liyanage S, Bagloee SA (2019) Applications of artificial intelligence in transport: An overview. Sustainability 11(1). https://doi.org/10.3390/su11010189
7. Dilek E, Dener M (2023) Computer vision applications in intelligent transportation systems: A survey. Sensors 23(6). https://doi.org/10.3390/s23062938
8. Wong PK-Y, Luo H, Wang M, Leung PH, Cheng JCP (2021) Recognition of pedestrian trajectories and attributes with computer vision and deep learning techniques. Adv Eng Inf 49:101356. https://doi.org/10.1016/j.aei.2021.101356
9. Chua LO, Roska T (1993) The cnn paradigm. IEEE Trans Circuits Syst I Fundamental Theory Appl 40(3):147–156. https://doi.org/10.1109/81.222795
10. Salehinejad H, Sankar S, Barfett J, Colak E, Valaee S (2018) Recent Advances in Recurrent Neural Networks. arXiv:1801.01078
11. Seidel R, Jahn N, Seo S, Goerttler T, Obermayer K (2022) Napc: a neural algorithm for automated passenger counting in public transport on a privacy-friendly dataset. IEEE Open J Intell Transportation Syst 3:33–44. https://doi.org/10.1109/OJITS.2021.3139393
12. Yu Y, Si X, Hu C, Zhang J (2019) A review of recurrent neural networks: lstm cells and network architectures. Neural Comput 31(7):1235–1270. https://doi.org/10.1162/neco_a_01199
13. Liu Y, Zhang Y, Wang Y, Hou F, Yuan J, Tian J, Zhang Y, Shi Z, Fan J, He Z (2024) A survey of visual transformers. IEEE Trans Neural Netw Learn Syst 35(6):7478–7498. https://doi.org/10.1109/TNNLS.2022.3227717
14. Mo H et al (2025) Countformer: Multi-view crowd counting transformer. In: Leonardis A, Ricci E, Roth S, Russakovsky O, Sattler T, Varol G (eds) Computer Vision – ECCV 2024. Lecture Notes in Computer Science, vol 15110. Springer, Cham. https://doi.org/10.1007/978-3-031-72943-0_2
15. Liu K, Zou X, Zhu P, Sang J (2025) Modal-adaptive spatial-aware-fusion and propagation network for multimodal vision crowd counting. IEEE Trans Consumer Electron 71(2):3605–3616. https://doi.org/10.1109/TCE.2025.3571571
16. Guo J, Ju H, Ji M, Wang J, Zhang X (2024) Indoor person counting based on multi-view linkage. In: Proceedings of the 2024 7th international conference on data science and information technology (DSIT), Nanjing, China, pp 1–6. https://doi.org/10.1109/DSIT61374.2024.10881022
17. Radovan A, Mršić L, Đambić G, Mihaljević B (2024) A review of passenger counting in public transport concepts with solution proposal based on image processing and machine learning. Eng 5(4):3284–3315. https://doi.org/10.3390/eng5040172
18. Nogueira V, Oliveira H, Augusto Silva J, Vieira T, Oliveira K (2019) Retailnet: A deep learning approach for people counting and hot spots detection in retail stores. In: 2019 32nd SIBGRAPI conference on graphics, patterns and images (SIBGRAPI), pp 155–162. https://doi.org/10.1109/SIBGRAPI.2019.00029
19. Massa L, Barbosa A, Oliveira K, Vieira T (2021) Lrcn-retailnet: a recurrent neural network architecture for accurate people counting. Multimed Tools Appl 80:5517–5537. https://doi.org/10.1007/S11042-020-09971-7/TABLES/5
20. Prastyo AD, Minhalina SA, Agung S, Bintang DN, Septian MY, Giri EP, Mindara GP (2024) Automatic passenger counting system on public buses using cnn yolov8 model for passenger capacity optimization. Int J Inf Eng Sci 1:55–63. https://doi.org/10.62951/IJIES.V1I4.121
21. Wu J, Xie Z, Qin Y, Jia L, Guan L (2024) Bi-directional passenger flow tracking and statistics analysis in station passageways based on an improved deep-sort algorithm. Measurement Control 57(2):152–163
22. Hsu Y-W, Chen Y-W, Perng J-W (2020) Estimation of the number of passengers in a bus using deep learning. Sensors 20(8):2178
23. Kim H, Sohn M-K, Lee S-H (2022) Development of a real-time automatic passenger counting system using head detection based on deep learning. J Inf Process Syst 18(3):428–442
24. Rawat N, Rai A, Agarwal A (2024) Deep learning-based passenger counting system using surveillance cameras. In: 2024 16th International Conference on COMmunication Systems & NETworkS (COMSNETS), IEEE, pp 234–239
25. Zhang J, Liu J, Wang Z (2021) Convolutional neural network for crowd counting on metro platforms. Symmetry 13(4):703
26. Zheng Z, Wang H, Liu W, Peng L (2023) Toward real-time congestion measurement of passenger flow on platform screen doors based on surveillance videos analysis. Phys A Stat Mech Appl 612:128474
27. Balasubramani K, Natarajan UM (2024) Improving bus passenger flow prediction using bi-lstm fusion model and smo algorithm. Babylonian J Artif Intell 2024:73–82
28. Siswanto J, Manongga D, Sembiring I, Wijono S (2024) Deep learning based lstm model for predicting the number of passengers for public transport bus operators. Jurnal Online Informatika 9(1):18–28
29. Yue M, Ma S (2023) Lstm-based transformer for transfer passenger flow forecasting between transportation integrated hubs in urban agglomeration. Appl Sci 13(1):637
30. Jin J, Guo H, Xu J, Wang X, Wang F-Y (2020) An end-to-end recommendation system for urban traffic controls and management under a parallel learning framework. IEEE Trans Intell Transportation Syst 22(3):1616–1626
31. Pirgazi J, Pourhashem Kallehbasti MM (2022) Ghanbari sorkhi a (2022) an end-to-end deep learning approach for plate recognition in intelligent transportation systems. Wireless Commun Mobile Comput 1:3364921
32. Le Mero L, Yi D, Dianati M, Mouzakitis A (2022) A survey on imitation learning techniques for end-to-end autonomous vehicles. IEEE Trans Intell Transportation Syst 23(9):14128–14147
33. Chib PS, Singh P (2023) Recent advancements in end-to-end autonomous driving using deep learning: a survey. IEEE Trans Intell Veh 9(1):103–118
34. Navarro PJ, Miller L, Rosique F, Fernández-Isla C, Gila-Navarro A (2021) End-to-end deep neural network architectures for speed and steering wheel angle prediction in autonomous driving. Electronics 10(11):1266
35. AG P (2023) Automated passenger counting on Swiss trains. Company web page. https://parquery.com/98-accurate-anonymo

us-passenger-counting-on-swiss-trains/ Accessed 03 September 2025
36. Soy H (2023) Edge ai-based crowd counting application for public transport stops. In: Convergence of deep learning and internet of things: computing and technology, pp 182–205. IGI Global Scientific Publishing, Hershey, PA. https://doi.org/10.4018/978-1-6684-6275-1.ch009
37. Outsight (2024) LiDAR Solutions for Airports. Company web page. https://www.outsight.ai/use-cases/airports Accessed 03 September 2025
38. Amorph Systems and Veovo collaborate on LiDAR technology (2023) Company news. https://amorph.aero/amorph-systems-and-veovo-collaborate-on-lidar-technology/ Accessed 03 September 2025
39. DILAX (2024) People Counting in Retail: Sensors & Software. Product page. https://www.dilax.com/es/products/visitor-counting Accessed 04 September 2025
40. Singh R, Gill SS (2023) Edge ai: a survey. Int Things Cyber-Physical Syst 3:71–92
41. Henriques JF, Caseiro R, Martins P, Batista J (2014) High-speed tracking with kernelized correlation filters. IEEE Trans Pattern Anal Mach Intell 37(3):583–596
42. Leal-Taixé L, Milan A, Schindler K, Cremers D, Reid I, Roth S (2017) Tracking the trackers: an analysis of the state of the art in multiple object tracking. arXiv:1704.02781
43. Wang G, Song M, Hwang J-N (2022) Recent advances in embedding methods for multi-object tracking: A survey. arXiv:2205.10766
44. Howard AG (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861
45. OpenVINO (2023) mobilenet-ssd — OpenVINO. Documentation. https://docs.openvino.ai/2023.3/omz_models_model_mobilenet_ssd.html Accessed 03 September 2025
46. OpenVINO (2025) person-detection-retail-0013 — OpenVINO. Documentation. https://docs.openvino.ai/2024/omz_models_model_person_detection_retail_0013.html Accessed 03 September 2025
47. OpenVINO (2025) OpenVINO Toolkit. Documentation. https://docs.openvino.ai/2024/index.html Accessed 03 September 2025
48. Terven J, Córdova-Esparza D-M, Romero-González J-A (2023) A comprehensive review of yolo architectures in computer vision: from yolov1 to yolov8 and yolo-nas. Mach Learn Knowl Extraction 5(4):1680–1716
49. Jocher G, Qiu J, Chaurasia A (2023) Ultralytics YOLO. Software. https://github.com/ultralytics/ultralytics Accessed 03 September 2025
50. Khanam R, Hussain M (2024) YOLOv11: An Overview of the Key Architectural Enhancements. arXiv:2410.17725
51. Padilla R, Netto SL, Da Silva EA (2020) A survey on performance metrics for object-detection algorithms. In: 2020 International conference on systems, signals and image processing (IWSSIP), IEEE, pp 237–242
52. Lesani A, Nateghinia E, Miranda-Moreno LF (2020) Development and evaluation of a real-time pedestrian counting system for high-volume conditions based on 2d lidar. Transportation Res Part C Emerging Technol 114:20–35
53. National Academies of Sciences, Engineering, and Medicine (2022) Highway Capacity Manual 7th Edition: A Guide for Multimodal Mobility Analysis. The National Academies Press, Washington, DC. https://doi.org/10.17226/26432