Integrated passenger flow analysis and street-level classification for public transport management using deep learning and IoT

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING
ELECTRICAL AND INFORMATION ENGINEERING PH.D. PROGRAM
SSD: ING-INF/04

Integrated Passenger Flow Analysis and Street-Level Classification for Public Transport Management using Deep Learning and IoT

by

Paganelli Mariano Giuseppe

Supervisors:

Prof. David NASO

Prof. Paolo R. MASSENIO

Coordinator of Ph.D. Program:
Prof. Nicola Giaquinto

Course n° 38, 01/10/2022 – 31/10/2025

Politecnico di Bari

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING
ELECTRICAL AND INFORMATION ENGINEERING PH.D. PROGRAM
SSD: ING-INF/04

Integrated Passenger Flow Analysis and Street-Level Classification for Public Transport Management using Deep Learning and IoT

by

Paganelli Mariano Giuseppe

Supervisors:

Prof. David NASO

Prof. Paolo R. MASSENIO

Coordinator of Ph.D. Program:
Prof. Nicola Giaquinto

Course n° 38, 01/10/2022 – 31/10/2025

Abstract

In recent years, the rapid development of digital technologies and computer vision systems has profoundly transformed the way cities observe and manage urban mobility. Within this evolving context, public transportation serves as a key data source for understanding travel patterns and improving service quality. However, the real-time acquisition and processing of reliable information about passengers and the surrounding urban environment remain a complex challenge, often constrained by traditional sensor-based or indirect counting systems. This thesis introduces an **integrated approach for real-time passenger counting and street-level monitoring** in public transport vehicles, combining computer vision, artificial intelligence, and geospatial analysis. Unlike conventional methodologies, the proposed system leverages **pre-trained deep convolutional neural networks**, specifically the **YOLO** model, to simultaneously analyze video streams captured both inside and outside the vehicles. Inside the vehicles, cameras are used to perform accurate passenger counting after door closure and flow analysis during stops, substantially improving precision compared to flow-only approaches. Externally, additional cameras monitor the surrounding environment to detect relevant urban elements—such as **potholes, and bicycles**—thus providing a comprehensive view of street conditions and infrastructure quality. All detections are **georeferenced using GPS data** and transmitted to a cloud platform, where they are processed to generate informative maps and indicators that support urban maintenance and planning. A further contribution of this research lies in the integration of passenger flow data within **public transport network analysis tools**, enabling the identification of usage patterns, bottlenecks, and operational inefficiencies. The system was validated through a **large-scale deployment** carried out in the city of **Bari, Italy**, involving **50 buses** operating across **30 lines** of the local public transport network. The experimental results demonstrated **high accuracy, robustness, and real-world applicability**, confirming the potential of the proposed solution to support **intelligent mobility management** and the transition toward more **sustainable and data-driven urban systems**.

This thesis is organized into six chapters. **Chapter 1** introduces the problem of vision-based monitoring in public transport, motivates the use of onboard perception as a source of actionable mobility intelligence, and frames the main research contributions through a review of state-of-the-art passenger counting and monitoring approaches. **Chapter 2** provides the theoretical background required by the proposed framework, covering the core principles of neural networks as well as the optical, geometric, and calibration aspects that influence the quality and reliability of camera-based measurements in real operating conditions. **Chapter 3** focuses on the object-detection backbone adopted in this work, namely YOLOv5: after presenting the model fundamentals and representative application domains, it motivates the choice of YOLOv5 for both in-cabin passenger analytics and outdoor urban-scene monitoring under embedded, real-time constraints. **Chapter 4** describes the end-to-end system architecture and operating workflow, detailing the onboard hardware and networking setup, the dual internal/external processing pipeline, the training procedures, and the cloud-oriented data collection strategy that enables fleet-level scalability. **Chapter 5** reports the experimental evaluation and results, including passenger counting validation, cloud-based analytics for real-time fleet monitoring and network-level insights, and the analysis of geo-referenced external detections for infrastructure and urban-environment assessment. Finally, **Chapter 6** summarizes the main findings and limitations, and outlines future research directions and deployment perspectives; it also reflects on the dissemination outcomes of the work, including the formative experience.

Acknowledgements

Questa tesi non sarebbe stata possibile senza la guida e il supporto di molte persone, alle quali sono profondamente grato.

Desidero esprimere la mia più sincera gratitudine ai miei supervisori, **Prof. David Naso** e **Prof. Paolo R. Massenio**, per la loro guida preziosa, il costante sostegno e i suggerimenti sempre illuminanti durante lo sviluppo di questa ricerca.

Un ringraziamento speciale va al mio collega **Marco Gallo**, la cui collaborazione e amicizia hanno reso questa esperienza di ricerca più arricchente e stimolante.

Questa ricerca è stata realizzata con il supporto di **Techrail Srl**, il cui contributo è qui riconosciuto con sincera gratitudine.

Desidero infine esprimere la mia profonda riconoscenza alla mia famiglia — i miei genitori **Raffaele** e **Rosa**, e i miei fratelli **Aldo** e **Francesco** — per il loro amore incondizionato, la pazienza e l'incoraggiamento che mi hanno accompagnato lungo tutto questo percorso.

Un ringraziamento davvero speciale va a **Lorena**, il cui costante supporto, la comprensione e l'affetto sono stati una fonte di forza e ispirazione nei momenti più impegnativi di questo lavoro.

List of Publications

The work presented in this thesis has led to the following publications:

Journal Articles

1. **M. G. Paganelli**, M. Gallo, P. R. Massenio, and D. Naso, "Integrated Passenger Flow Analysis and Street-Level Monitoring for Public Transport Management Using Deep Learning and IoT," *IEEE Access*, vol. 13, pp. 143401–143413, 2025, doi: 10.1109/ACCESS.2025.3597327.

2. M. Gallo, **M. G. Paganelli**, P. R. Massenio, and D. Naso, "Real-Time Violence Detection in Urban Bus Environments," *IEEE Access*, vol. 14, pp. 55075–55089, 2026, doi: 10.1109/ACCESS.2026.3681675.

Conference Proceedings

1. **M. G. Paganelli**, M. Gallo, P. R. Massenio, and D. Naso, "Enhancing Public Transport Management with Deep Learning and IoT-Based Monitoring," in *Proc. 2025 13th Int. Conf. on Traffic and Logistic Engineering (ICTLE)*, 2025, pp. 346–350, doi: 10.1109/ICTLE67020.2025.11203545.

2. M. Gallo, **M. G. Paganelli**, D. Naso, and P. R. Massenio, "AI-Enabled Bus Surveillance: Real-Time Passenger Counting and Violence Detection," presented at the Int. Conf. on Artificial Intelligence and Smart Environments (ICAISE'25), 2025, Hammamet, Tunisia.

Contents

List of Figures

List of Tables

List of Abbreviations

EDU	**E**lectrical **D**istribution **U**nit
EMI	**E**lectro**m**agnetic **I**nterference
PoE	**P**ower-**o**ver-**E**thernet
DoF	**D**epth **o**f **F**ield
FOV	**F**ield **o**f **V**iew
OD	**O**rigin-**D**estination
CNN	**C**onvolutional **N**eural **N**etwork
SNR	**S**ignal-to-**N**oise **R**atio
CV2	Open**CV**

Chapter 1

INTRODUCTION

1.1 Passenger Analytics and Urban Mobility: Tools for Public Transport Improvement

Efficient public transportation management in urban areas faces increasing challenges due to growing mobility demand, dynamic population distribution, and evolving external conditions. Within this context, two fundamental tasks — accurate passenger counting and street-level monitoring — play a crucial role in achieving a more intelligent, adaptive, and sustainable transport system.

Passenger counting represents a cornerstone for data-driven fleet management, service optimization, and demand forecasting. Accurate and continuous monitoring of passenger flows enables transport agencies to make evidence-based decisions, shifting from reactive to proactive operations. Knowing precisely how many passengers board and alight at each stop allows operators to evaluate route performance, adjust vehicle allocation, and redesign timetables based on real demand dynamics rather than on estimations. This transition toward data-driven planning ensures that service provision accurately reflects the evolving mobility needs of the population throughout different times of the day, days of the week, and seasons.

Beyond operational efficiency, passenger counting also supports strategic planning and policy-making. The identification of overcrowded routes, underused services, and spatio-temporal demand patterns provides a solid foundation for long-term investment in transport infrastructure and the design of optimized services. Understanding where and when demand fluctuates enables public transport agencies to allocate resources more effectively, improve multimodal connectivity, and support environmentally sustainable mobility strategies. Reliable passenger flow data also serve as a foundation for key performance indicators (KPIs) such as load factors, dwell times, headway regularity, and punctuality — all of which are essential for improving operational reliability, cost-effectiveness, and passenger satisfaction [18, 19].

Despite these recognized benefits, current passenger counting systems available on the market and in the state of the art remain largely inefficient or underexploited. Many existing solutions rely on outdated sensor technologies, such as infrared or ultrasonic devices, which are limited in accuracy and strongly affected by environmental conditions. As a result, such systems are often perceived as unreliable or economically unjustified, and consequently are rarely integrated into the operational frameworks of public transport agencies. In most cases, operators continue to rely on manual counts or statistical estimations, which fail to capture the temporal and spatial variability of passenger demand. This lack of accurate, real-time data prevents true data-driven optimization, hindering the potential for dynamic adjustments in fleet deployment, frequency planning, and service design.

This work aims to overcome these limitations by proposing a practical and implementable solution that integrates real-time passenger counting and street-level monitoring in a single unified system. By leveraging deep learning, computer vision, and Internet of Things (IoT) technologies, the proposed approach demonstrates how accurate, scalable, and cost-effective passenger analytics can be embedded directly into public transport operations. This integration not only enhances decision-making and service adaptability but also lays the groundwork for a new paradigm of intelligent and responsive mobility management, where real-time data, automation, and contextual analysis work together to improve both operational efficiency and the quality of urban life.

In parallel, street-level monitoring provides complementary insights into the external environment of urban mobility. Detecting and analyzing conditions such as traffic congestion, road surface degradation, pedestrian density, or temporary obstacles contributes to safer and smoother journeys while supporting municipalities in infrastructure maintenance and planning [20]. In recent years, numerous research efforts have explored the integration of passenger counting and environmental perception through machine learning, computer vision, and IoT technologies, promoting the development of holistic, adaptive, and intelligent public transport management systems capable of addressing the growing complexity of modern urban mobility [21].

Most existing systems tackle these challenges separately. Current passenger counting systems typically use door-mounted sensors and flow-based estimations [1, 22], but exhibit intrinsic limitations: they cannot provide precise passenger tracking inside the vehicle, produce coarse estimates that degrade in crowded conditions, and do not leverage the capabilities of modern convolutional neural networks (CNNs) to operate in visually complex and occluded environments [4, 23, 24]. Moreover, validation is hindered by the lack of realistic, publicly available datasets for bus interiors. Instead, for external urban monitoring, such as pothole detection, scooter or pedestrian classification, the literature offers several CNN-based methods [25–28]. Nevertheless, these systems typically function without any integration with internal vehicle data (e.g., occupancy or passenger flow) or spatio-temporal correlation between street-level activity and on-board behaviour. This lack of integration reveals a technological gap: the absence of a unified framework combining internal and external perception, which is essential for enabling higher-level tasks such as multimodal optimization, context-aware fleet management, and safety enhancement.

In this work, we propose a novel and integrated framework that combines accurate passenger counting, based on CNNs and multi-object tracking, with the classification and geo-location of urban elements captured through outward-facing cameras. Alongside internal analytics, the system continuously monitors the external road environment, detecting road surface anomalies such as potholes, cracks, standing water, and general pavement degradation, as well as classifying a wide spectrum of street-level elements including scooters, bicycles, pedestrians, parked vehicles, signage, construction areas, and temporary obstacles. Through this dual perception layer, the vehicle becomes a mobile sensing platform capable of capturing both the dynamic behaviour of passengers and the evolving conditions of the surrounding urban space.

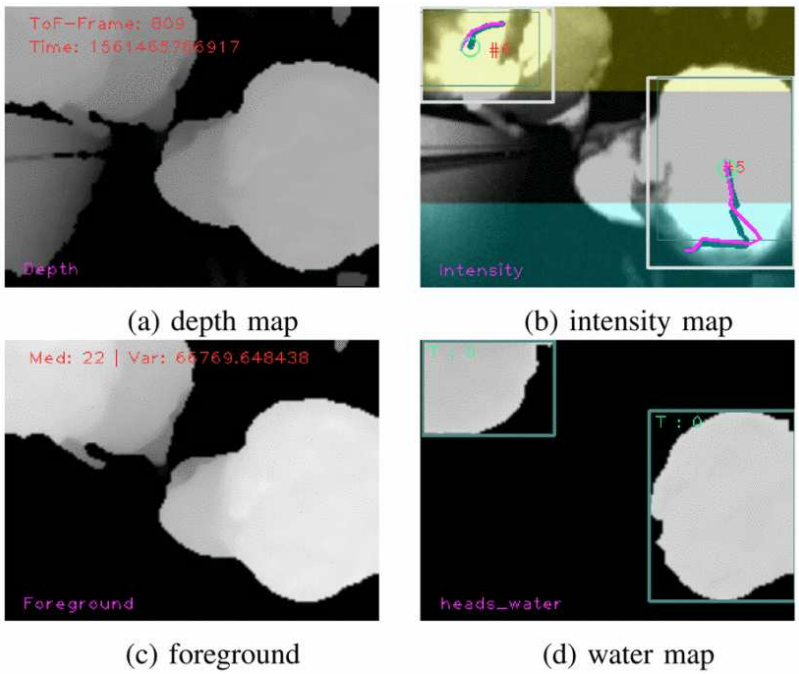
By embedding internal and external perception into a unified architecture, the system enables a multi-layered analytical process in which vehicle occupancy trends, passenger flow patterns, and street-level contextual factors are jointly interpreted rather than treated as independent information streams. This integration supports

advanced applications in route planning—where passenger demand can be correlated with infrastructural constraints or recurrent bottlenecks—and in fleet management, where dispatching and scheduling strategies can be adapted not only to ridership patterns but also to real-time environmental conditions. Moreover, the continuous acquisition of geo-referenced road condition data enables proactive and predictive maintenance workflows, allowing municipalities to detect degradation patterns early, optimize intervention priorities, and reduce long-term repair costs through data-driven asset management.

1.2 Literature Review

For passenger counting, previous studies have primarily focused on flow-based systems, where individuals are detected as they pass through predefined gates or door areas. For example, [23] employs computer vision and tracking algorithms to detect entering and exiting passengers, while [5] proposes a lightweight deep learning solution based on the MobileNetv2-SSD model, designed to operate efficiently on resource-constrained embedded devices. These approaches demonstrate the feasibility of visual detection on low-cost hardware, yet their applicability remains limited when scaled to complex real-world scenarios such as crowded buses or irregular passenger movements.

Other works, such as [1] (Kinect) and [29] (ToF), adopt overhead depth sensors and classical segmentation pipelines—often including morphological filling or "water-filling"–like strategies Fig. 1.1.—to improve robustness under partial occlusions.



(a) depth map (b) intensity map

(c) foreground (d) water map

FIGURE 1.1: Water-filling strategies [1]

[1] explicitly targets Kinect-based people counting at doorways, representing the line of work that relies on vertical depth acquisition for head/torso segmentation. Representative implementations of the water-filling family applied to vertical Kinect show how head segmentation from depth maps can enable real-time counts but remains sensitive to clutter and mutual occlusions.

[29] study Time-of-Flight (ToF) sensing for people counting, highlighting the appeal of compact depth hardware for embedded deployments; yet ToF ranging is prone to multipath interference and reflection-induced errors in realistic interiors with glossy, metallic, or glass surfaces. Similar scene-dependent artifacts are reported for Kinect-class depth sensors, where accuracy and noise vary with materials and lighting; measurements across the field of view can yield non-negligible depth errors and outliers.

Stereo-based pipelines constitute another historical strand. [30] introduced stereo for gate-crossing counts; subsequent analyses note that performance is highly sensitive to rig calibration/shift and lighting, which complicates field operation. [22] proposed dense close-range stereovision for bus passenger counting and reported very high accuracy on a large dataset including laboratory scenarios and some in-service bus sequences; nevertheless, scalability to crowded, highly dynamic situations remains constrained by disparity quality and segmentation thresholds.

Finally, [4] engineered a neural APC using depth video with LSTM-based cumulative summation and ensemble techniques to reduce bias; while this improves stability over single-pass detectors, crowding and simultaneous bidirectional motion still stress the counting pipeline in the absence of dedicated multi-target tracking. In sum, although depth-centric systems can be highly accurate in controlled or low-density conditions, their performance degrades in crowds where mutual occlusions inflate cumulative errors. An alternative approach, proposed by [4], processes video sequences of door-opening phases through a Long Short-Term Memory (LSTM) neural network combined with cumulative summation to estimate the number of boarding and alighting passengers. This method is relatively robust against noise and motion blur; however, it still exhibits performance degradation in high-density conditions, where distinguishing multiple overlapping passengers or determining movement direction becomes challenging. The lack of integrated tracking mechanisms exacerbates the risk of misclassifications when multiple individuals move simultaneously. In addition, most of these systems remain untested on real operating buses, where vibration, camera placement, and dynamic lighting further affect reliability.

A different line of research explores non-visual methods. For instance, [2] leverages Wi-Fi and Bluetooth signals to estimate passenger flows and infer origin destination (O–D) matrices.

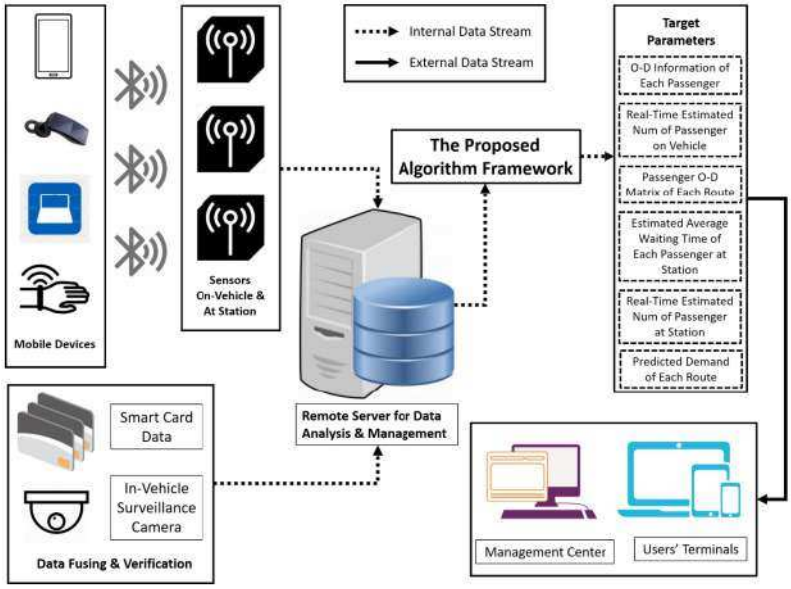


FIGURE 1.2: Wi-Fi and Bluetooth strategies [2]

While these methods can provide useful aggregate insights, they inherently exclude a significant portion of the population — such as users without devices, with disabled wireless connectivity, or with multiple overlapping devices — and their accuracy decreases in densely populated environments where signal interference and reflection are common.

Overall, these studies highlight both the potential and the limitations of existing passenger counting technologies. Despite numerous innovations, current systems are still fragmented, domain-specific, and difficult to integrate into operational frameworks of public transport agencies. The main features and limitations of the most representative approaches are summarized in Table 1.1, providing a clear comparative overview that motivates the need for a more integrated and deployable solution, as proposed in this work.

TABLE 1.1: Summary of flow-based passenger counting approaches.

Work	Sensor	Scope	Main Limitation
[1]	Kinect (Depth)	Passage way	Lighting sensitivity
[22]	Stereo camera	Door	Lighting sensitivity
[4]	LiDAR/ToF	Door	No interior tracking
[23]	RGB camera	Door	No interior tracking
[5]	RGB camera	Passage way	Not validated on buses
[29]	ToF sensor	Passage way	Narrow field of view
[30]	Stereo camera	Passage way	Not validated on buses

While recent research has made significant progress in vehicle detection and tracking using computer vision and deep learning techniques [31,32], comparatively fewer studies have focused on the detection of critical road surface elements—such as potholes, cracks, or debris—from the perspective of public transport vehicles. The ability to monitor road conditions from a bus-mounted viewpoint offers an underexplored yet highly valuable opportunity to gather continuous, large-scale data directly from operating fleets, supporting predictive maintenance and improving road safety and service quality.

In [25], the YOLOv7 model [3] is used to detect and geolocate potholes in road imagery Fig. 1.3.

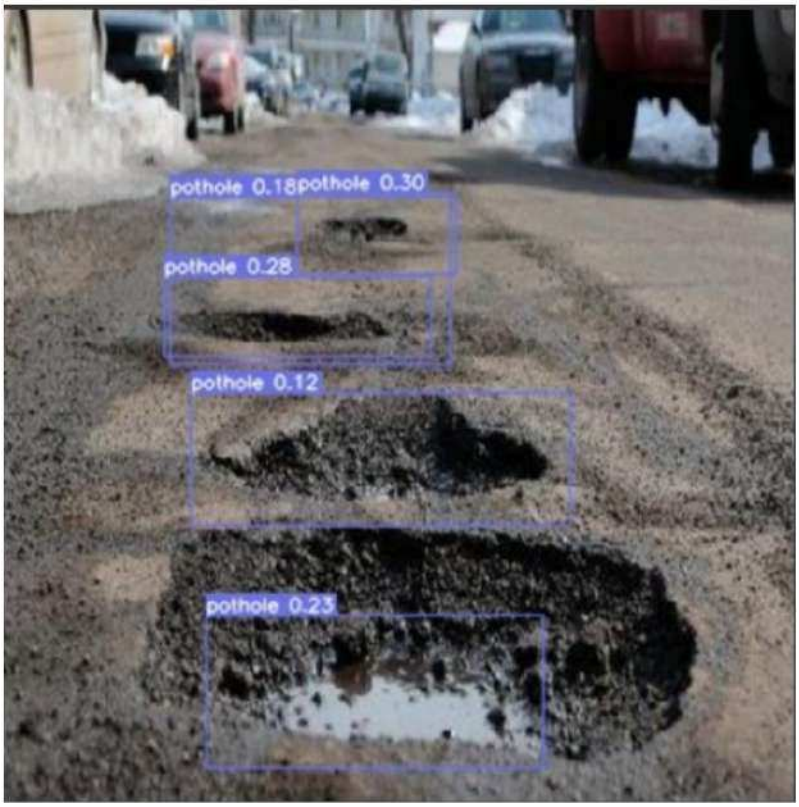


FIGURE 1.3: Potholes Detection [3]

Although promising, its overall accuracy was constrained by the limited size and diversity of the dataset and by the reliance on individual static images rather than continuous video sequences. This frame-by-frame approach neglects temporal information that could improve consistency and reduce false detections. Similarly, in [26], a comparison between MobileNetv2 and ResNet architectures demonstrated satisfactory performance for MobileNetv2 under controlled settings, yet the evaluation was again limited to static images, preventing the assessment of robustness in dynamic conditions such as motion blur, variable illumination, or occlusions caused by traffic.

A more comprehensive analysis was conducted in [27], where four different object detection models were compared for pothole detection in autonomous driving

contexts. The study identified several key challenges, including speed constraints (up to 60 km/h), dataset imbalance heavily favoring non-pothole images, and significant performance degradation under adverse conditions such as rain, poor lighting, or wet surfaces. In addition, false positives caused by road stains, manholes, or maintenance patches were frequent, further undermining the model's reliability in practical deployment scenarios.

Finally, [28] investigated the combination of stereo vision and deep learning techniques to enhance depth perception and improve surface anomaly recognition. However, despite the theoretical advantages of stereo-based depth cues, the experiments revealed substantial limitations due to restricted datasets, variability in illumination, and reflections on asphalt, which hindered the generalization of the model to real-world conditions.

Overall, the existing literature highlights that road anomaly detection from bus-mounted or vehicle-based perspectives remains an emerging field, constrained by dataset scarcity, environmental variability, and the lack of continuous, real-time analysis. These limitations underline the need for more robust and integrated approaches capable of leveraging temporal information, advanced data fusion, and adaptive algorithms to ensure reliable operation in diverse and dynamic urban environments.

1.2.1 Time-of-Flight (ToF) sensing for people counting

Time-of-Flight (ToF) technology is currently one of the most promising and scientifically established approaches for automatic passenger counting in public transportation systems. ToF sensors operate by emitting modulated or pulsed infrared radiation and measuring the round-trip propagation delay or phase shift of the reflected signal. This enables the reconstruction of a dense and accurate depth map, largely independent of ambient lighting conditions, which is a substantial advantage over traditional RGB or stereo vision systems that are sensitive to shadows, glare, contrast variations and complex illumination patterns [1]. The combination of illumination independence and high frame rates—typically between 30 and 100 fps—makes ToF particularly effective for monitoring rapid and dynamic events such as the transient flow of passengers across vehicle doors.

In public transport environments, ToF devices are typically installed in a top-down configuration above the entry and exit doors. This zenithal perspective provides a volumetric representation of the doorway area, facilitating the discrimination between boarding and alighting passengers, as well as the detection of simultaneous crossings or lateral movements. The vertical viewpoint Fig. 1.4 also reduces the likelihood of occlusions compared to frontal or side-mounted RGB cameras and provides a more direct and geometrically consistent basis for silhouette segmentation. Furthermore, the ToF measurement principle inherently separates static objects from moving individuals, enabling reliable differentiation between humans and carried objects such as luggage or strollers [1].

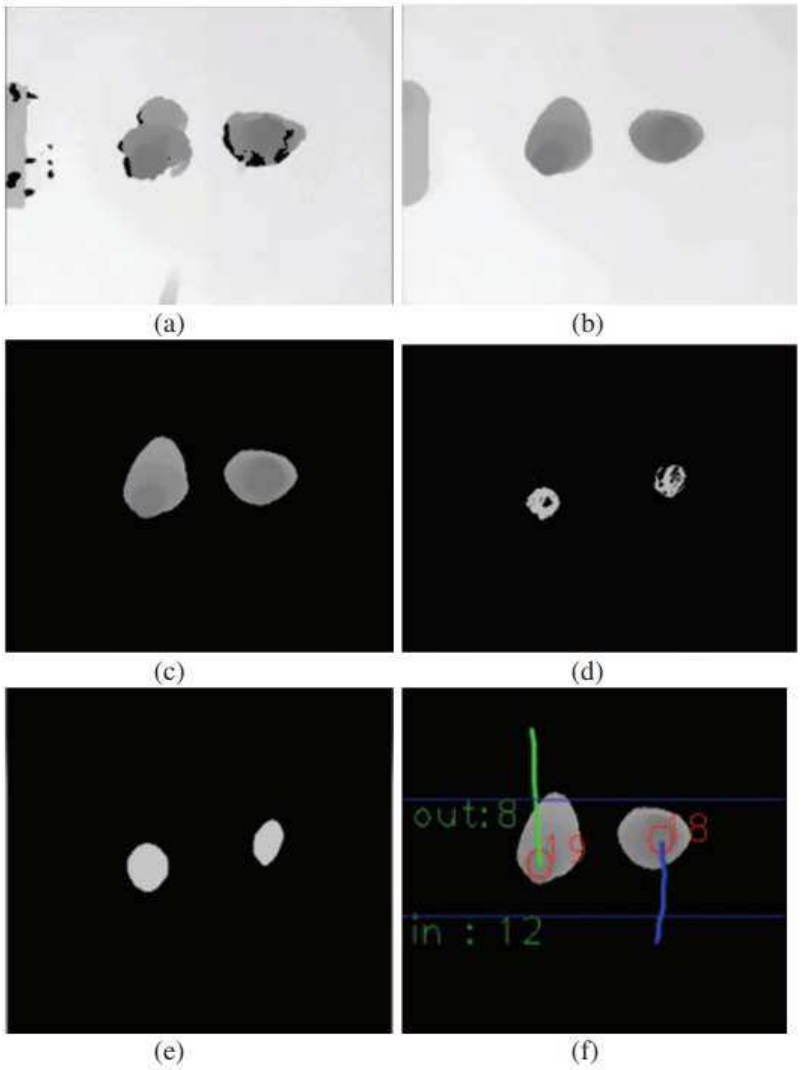


FIGURE 1.4: Vertical viewpoint [1]

Despite these strengths, the deployment of ToF in moving vehicles introduces several physical and computational challenges. Among these, multipath interference is one of the most significant. Due to the abundance of reflective surfaces inside modern buses—such as metallic poles, glossy floors, plastic panels and glass elements—the infrared signal can follow multiple optical paths before returning to the sensor. These secondary reflections introduce erroneous depth measurements, local distortions, phantom surfaces and unstable silhouette boundaries [1]. In harsh illumination conditions, especially in the presence of direct sunlight, sensor saturation and infrared contamination can further degrade measurement stability.

Mechanical vibrations and dynamic stresses generated by vehicle motion pose additional challenges. Irregular road surfaces, sharp braking, sudden accelerations and lateral oscillations cause continuous variations in sensor orientation and position, effectively altering the intrinsic and extrinsic calibration parameters in real time. Thermal fluctuations inside the passenger cabin can also affect sensor behaviour, leading to depth drifts and calibration inconsistencies that must be compensated to maintain measurement accuracy [1].

High passenger density contributes another significant source of error. During peak hours, occlusions between individuals become frequent, causing merged or fragmented blobs that hinder the reliable identification of single passengers. Children standing close to adults, groups entering simultaneously and complex body postures can all lead to systematic counting errors that accumulate throughout the journey, unless corrected by robust temporal and spatial filtering pipelines [1].

To ensure reliable operation, a sophisticated processing stack is required. This typically includes spatial and temporal noise filtering, multipath suppression algorithms, clustering and segmentation techniques in the depth domain, dynamic recalibration modules and probabilistic models for event detection. In more advanced scenarios, deep learning models operating on 3D data may further refine the segmentation and classification stages, although this comes at the cost of increased computational demands that may exceed the capabilities of low-power embedded devices commonly installed in public transport vehicles [1].

Overall, Time-of-Flight technology represents a theoretically promising and highly robust solution for passenger counting in public transport vehicles, owing to its active illumination principle, the high precision of depth measurements, and its relative insensitivity to ambient lighting conditions. However, when transitioning from controlled experiments to real operational environments, a series of structural limitations emerge that make this approach essentially unfeasible in real-world deployments.

Although a ToF sensor can indeed provide highly reliable depth information for a single acquisition, passenger counting is not performed on absolute measurements but rather through the incremental accumulation of boarding and alighting events, computed via additions and subtractions. Within this framework, a single detection error—caused, for example, by occlusions, multipath reflections, or a misinterpreted silhouette—propagates across all subsequent counts, accumulating progressively throughout the entire trip.

This phenomenon, known as cumulative error, is the fundamental weakness of ToF-based counting systems in public transport applications. Even a minimal initial discrepancy can escalate substantially and produce large deviations between the real and estimated passenger load by the end of the route. In extreme cases, this propagation may even result in mathematically impossible values, such as a negative number of passengers on board, clearly demonstrating the fragility of a counting method based solely on entrance–exit differentials.

For these reasons, despite the fact that ToF technology provides some of the most accurate and detailed depth data available at the frame level, its direct use as a reliable operational passenger counting system is not feasible, precisely because of the incremental counting mechanism and the unavoidable accumulation of propagated errors over time.

1.2.2 People counting using depth video LSTM-based

The use of depth video combined with LSTM-based neural networks for passenger counting represents one of the most ambitious attempts to leverage sequential modeling to interpret the complex dynamics of people moving through vehicle doors. In the work of Jahn and Siebert [4], the underlying idea is that a temporal model—fed not with RGB frames but with depth maps—can learn the structure of boarding and alighting events without the need for frame-level annotations. Instead, the LSTM is trained solely from the final total count assigned to each video, a strategy that

drastically reduces labeling effort and enables the training of large-scale models on thousands of real-world sequences.

The proposed system processes each depth frame by flattening it into a vector, which is then passed through a stack of LSTM layers. Although this flattening removes explicit spatial structure, it encourages the network to focus on the temporal evolution of shapes and movements. A particularly clever component is the cumulative summation (cumsum) layer placed at the output. Rather than forcing the LSTM to learn the arithmetic of cumulative counting, the model predicts the frame-to-frame differences while the cumsum layer integrates these increments over time. This significantly stabilizes long sequences and theoretically allows the model to handle arbitrary passenger counts.

Yet the paper also reveals substantial challenges 1.5 . LSTM architectures are known to suffer from drift and instability when dealing with long or highly variable sequences, and the authors demonstrate that without the cumsum layer the model reaches an effective upper bound of roughly 100 passengers. Even with the improved architecture, training outcomes vary enormously: results depend heavily on random initialization, batch ordering and training-set selection, to the extent that two models trained under identical conditions can behave completely differently. This indicates a level of statistical instability that is problematic for any system expected to operate reliably in real environments.

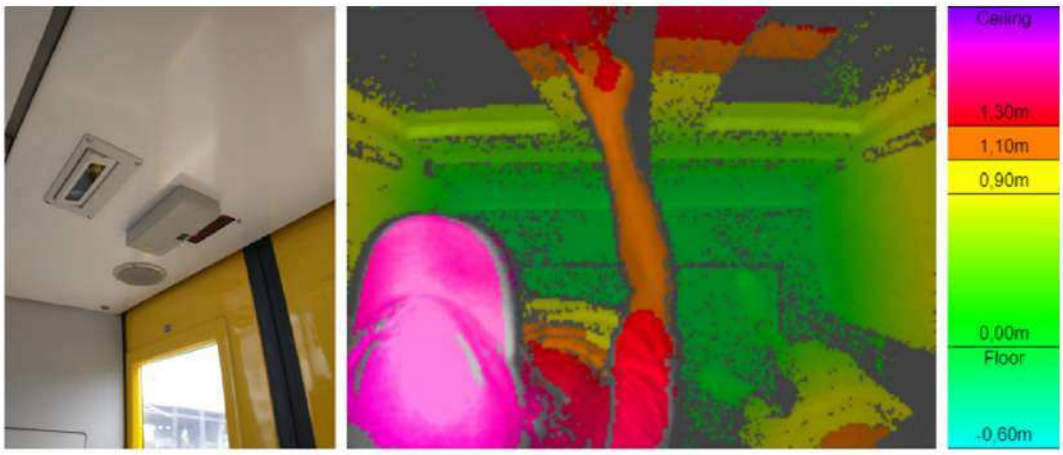


FIGURE 1.5: Depth video with LSTM-based strategies [4]

Another limitation arises from the dataset's distribution. Although the dataset is relatively large, high-density passenger scenarios are rare, which leads to poor generalization exactly in the situations that matter most. The model struggles with simultaneous crossings, tightly packed groups, child–adult combinations, and the presence of bulky objects like strollers or bicycles—scenarios that depth sensors often capture with partial occlusions and inconsistent silhouettes.

From an operational standpoint, these weaknesses have profound implications. Even if the LSTM performs reasonably well in isolated door-opening sequences, real-world transit systems rely on cumulative counts across entire trips. In such a

context, a single significant miscount can propagate through all subsequent calculations, producing large cumulative errors similar to those observed in traditional ToF-based APC systems. This makes the method unsuitable for operational use, where consistency, reproducibility and bias-free long-term accuracy are non-negotiable requirements. Figures in the paper clearly show cases where model performance collapses unpredictably, confirming that the approach—while innovative—is not yet stable enough for practical deployment.

In summary, the depth-video LSTM-based approach is an important scientific step forward, offering a sophisticated learning-based alternative to conventional counting systems and eliminating the need for costly frame-level annotations. However, the inherent volatility of sequential neural architectures, the strong sensitivity to randomness, and the unavoidable error propagation across operational timeframes make this method currently unsuitable for reliable use in public transport operations. It remains a promising line of research, but significant challenges must be overcome before it can meet the stringent requirements of real-world APC systems.

1.2.3 Wi-Fi and Bluetooth System

Using Wi-Fi and Bluetooth signals to estimate passenger counts and infer Origin – Destination [2] patterns has become a widely explored, non-intrusive approach due to the ubiquity of personal mobile devices. The underlying idea is that smartphones and other wireless devices continuously emit probe requests or inquiry responses, which can be captured by a sensor Fig. 1.6 placed inside a transit vehicle. By tracking these emissions over time and extracting signal-based features, it is possible—at least in principle—to approximate how many passengers are onboard, when they board and alight, and how they move along the route.

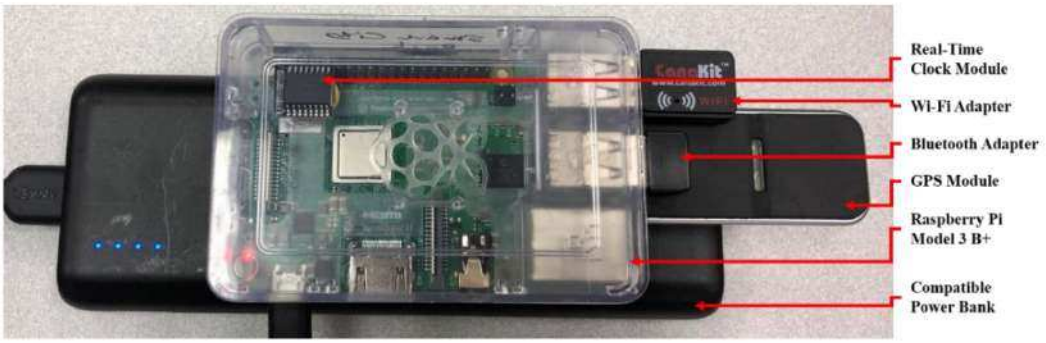


FIGURE 1.6: Data collection sensor [2]

To build such a system, a number of features must be extracted for each detected MAC address: the number of times the device is observed, the duration of its presence, received signal strength values, and the distance of the vehicle from nearby stops when the device first and last appears. Additional information on vehicle motion, such as speed and distance traveled during detection, helps distinguish whether a device is moving together with the vehicle or is stationary outside.

The most delicate part of the process is separating devices belonging to actual passengers from those belonging to pedestrians, cyclists, people waiting at stops, nearby buildings, or passengers in adjacent vehicles. Since the feature sets of these groups overlap substantially, the classification cannot be reliably performed using simple threshold-based filtering. Fuzzy clustering techniques are therefore often employed, allowing devices to have partial membership in both "passenger" and "non-passenger" groups. This soft assignment provides a way to handle the intrinsic ambiguity of the data and generally yields better results than hard, deterministic rules.

Once a subset of "likely passenger devices" is identified, a regression model is needed to map the number of detected devices to the actual number of passengers. Because only a fraction of passengers carry a detectable device—and because this fraction varies unpredictably—non-linear models such as Random Forest regression are typically used to capture the relationship between MAC counts and real ridership.

Conceptually, the approach has several attractive properties: the hardware is inexpensive, the deployment is simple, it does not require cameras or intrusive sensors, and it can provide continuous data throughout the trip. The combination of Wi-Fi and Bluetooth also increases the chances of observing a meaningful number of devices.

However, when examined through the lens of operational reliability, significant limitations emerge. In real settings, the majority of detected MAC addresses do not belong to onboard passengers but to devices in the surrounding environment. The proportion of actual passenger devices can be extremely small, making the separation process highly sensitive to noise, urban geometry and changes in the environment. A bus passing close to a building façade or traveling alongside another vehicle may detect dozens of non-passenger devices whose signal patterns closely resemble those of legitimate passengers.

A more critical limitation stems from the modern adoption of MAC address randomization, now standard on most mobile operating systems. Devices can change their MAC address frequently—sometimes with every probe request—making it nearly impossible to track a single individual through time or determine both boarding and alighting points. This fundamentally undermines the ability to reconstruct Origin–Destination flows or even to identify consistent device trajectories.

Even if the classification problem were perfectly solved, the core statistical issue remains: the ratio between detectable devices and actual passengers is not stable. It fluctuates across routes, times of day, demographic groups, and even according to individual user settings. This variability introduces systematic, non-recoverable errors that no regression model can reliably correct. As a result, the estimates may appear plausible in one context and fail dramatically in another.

In summary, while Wi-Fi and Bluetooth sensing provides a compelling research tool and a potentially useful source of qualitative insights into transit demand, it falls short of the robustness, stability and accuracy needed for operational passenger counting systems. Environmental noise, MAC randomization, low detection reliability and temporal variability collectively make this approach unsuitable for real-world deployment where precise, repeatable and certifiable measurements are required.

1.2.4 Computer vision and Tracking System

Computer vision and tracking–based passenger counting systems [5] are among the most widely adopted technologies in modern public transport due to their ability to detect and follow passengers in real time during boarding and alighting. These systems typically rely on an overhead camera positioned above the vehicle door 1.7 , capturing continuous video streams from which a deep learning detector identifies passengers and surrounds them with bounding boxes. Modern convolutional neural networks such as EfficientDet, SSD, Faster R-CNN or YOLOv3 enable highly accurate detection even under challenging conditions, including partial occlusions, varying illumination and high passenger densities.

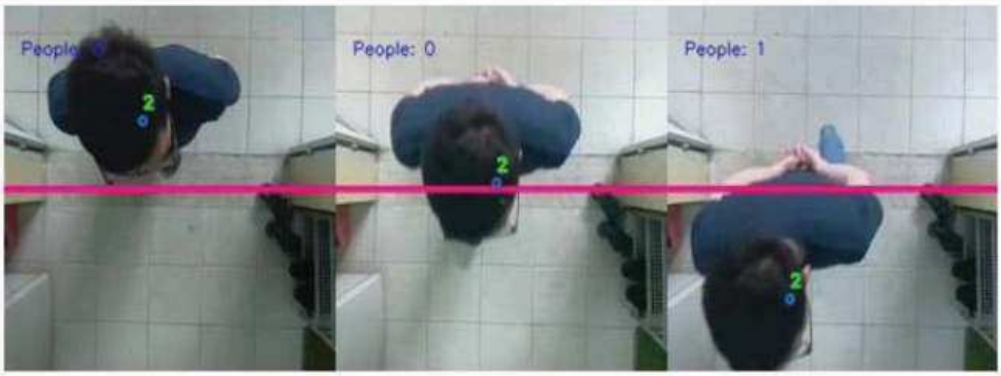


FIGURE 1.7: The result when one person coming in [5]

The general workflow involves initial preprocessing of the video frame, followed by the application of an object detection model and then a tracking algorithm Fig. 1.7 to maintain identity consistency across time. Centroid tracking is commonly used, relying on the movement of bounding box centroids to associate detections across consecutive frames. By monitoring how each centroid shifts within the scene, the system assigns persistent IDs to individuals and uses these trajectories to determine when a passenger crosses a virtual counting line, allowing the system to increment entry or exit counts.

Conceptually, this methodology has clear strengths. Deep learning–based detection is remarkably robust in dynamic public transport environments where lighting changes, passengers occlude each other or the camera's perspective introduces distortions. Tracking enables temporal continuity, converting per-frame detections into coherent passenger trajectories. The virtual line mechanism is intuitively interpretable and aligns well with the physical action of boarding and alighting.

However, transitioning from controlled experimental setups to actual operational environments reveals significant limitations. The performance of these systems is highly dependent on video quality: strong backlighting, reflections from glass doors, nighttime illumination or lens contamination can severely degrade detection accuracy. Vehicle motion introduces additional instability, as vibrations or sharp movements can interfere with the detector's ability to localize passengers.

Crowded conditions represent another major challenge for tracking consistency. When passengers overlap, cross paths or move very close to each other, the tracker may confuse identities, leading to ID switches or lost tracks. These errors directly impact the counting mechanism: a single mistaken association can produce multiple incorrect increments or missed counts, significantly reducing reliability over an entire trip. The issue becomes even more pronounced in highly congested buses where occlusions are constant.

The virtual line strategy, while simple, is sensitive to real-world passenger behavior. Individuals may hesitate, step back, stop in the counting zone or move unpredictably, causing the system to count them multiple times or not at all. The assumption of a clean, directional crossing rarely holds in practice, especially during peak hours or in scenarios involving strollers, wheelchairs, or passengers carrying large items.

Furthermore, although deep learning detection models achieve high accuracy in general, their reliability depends on the similarity between the training data and the operational environment. Variations in camera angle, vehicle model, passenger demographics or environmental lighting can degrade performance over time, necessitating frequent retraining and system calibration to maintain acceptable accuracy.

In summary, computer vision and tracking systems offer a powerful and sophisticated approach to automated passenger counting, capable of high performance under standard conditions. Nevertheless, their real-world deployment remains challenging due to occlusions, unpredictable human behavior, lighting variability and the intrinsic instability of tracking in crowded, dynamic scenes. These factors collectively limit their ability to deliver the level of repeatability and operational robustness required for certified, production-grade passenger counting.

1.3 Research Contributions and Innovations

This research proposes an integrated framework for real-time analysis of dynamic environments in public transport. The system leverages embedded high-performance computing, deep learning, and IoT technologies within a unified architecture that ensures cost efficiency, scalability, and high accuracy. Unlike traditional solutions that address operational aspects in isolation, this framework introduces a fully connected data ecosystem capable of processing multi-sensor information streams in real time, thus overcoming the fragmentation and limitations of conventional methods.

The developed system represents a significant step forward in the evolution of smart public transportation infrastructures, enabling continuous observation of passenger flow and street-level conditions through distributed IoT devices integrated onboard the fleet. Each vehicle functions as a mobile sensing node, equipped with GPU-enabled embedded PCs capable of performing deep learning inference directly in real time, without the need for constant cloud connectivity. This distributed approach reduces latency, enhances resilience, and allows for autonomous operation even in bandwidth-constrained environments.

The work introduces several key innovations aimed at improving robustness and applicability in real-world scenarios:

- **Integrated IoT architecture** — interconnected onboard devices enable continuous data acquisition, local processing, and remote synchronization through cloud-based dashboards, ensuring data consistency and system scalability across the fleet.

- **Elimination of cumulative counting errors** — unlike traditional flow-based systems affected by incremental inaccuracies, the proposed method adopts a pixel-to-Cartesian conversion process for spatial calibration combined with a deep object tracking mechanism (Deep SORT) to ensure precise localization and reliable passenger tracking over time.
- **Enhanced visual sensing** — the adoption of high-resolution RGB cameras, coupled with carefully trained convolutional neural networks (CNNs), mitigates the common drawbacks of sensor-based technologies such as time-of-flight (ToF) devices, including sensitivity to lighting variations and reflective surfaces.
- **Comprehensive external monitoring** — the system performs continuous analysis of road and environmental conditions through external cameras. By leveraging sequential image analysis rather than static frames, the approach allows for more accurate detection and tracking of potholes, pedestrians, bicycles, scooters, and other relevant urban elements, enabling a richer understanding of the surrounding infrastructure.
- **Operational validation** — the integrated solution was extensively deployed and tested under real operating conditions across 50 buses serving 30 routes within the public transport network of Bari, Italy. Over a two-month testing period, the system demonstrated stable performance, high accuracy, and practical applicability in day-to-day operations, proving its readiness for large-scale implementation.
- **Geospatial integration and visualization** — all detections are geo-referenced using GPS correlation and visualized through a dedicated web-based platform, providing decision-makers with intuitive tools for monitoring the network, detecting recurrent issues, and prioritizing maintenance interventions based on spatial density and severity.
- **Geo-referenced signage analysis** — the system extends its external monitoring capability to the detection and geo-referencing of road signs, traffic signals, and urban signage. Through deep learning-based classification and GPS correlation, the framework can support urban asset mapping, traffic regulation monitoring, and infrastructure maintenance. This functionality enhances the system's potential for integration with smart city platforms and transport authorities.

Overall, this research demonstrates the feasibility of combining deep learning, IoT, and real-time data analytics into a single operational system for intelligent mobility management. The proposed framework establishes the foundation for scalable smart mobility systems, promoting more efficient, adaptive, and sustainable urban transport management.

Chapter 2

VISION-BASED SYSTEMS IN PUBLIC TRANSPORTATION INFRASTRUCTURES

2.1 Neural Network Theory and Design Principles

Artificial neural networks represent one of the most powerful and versatile tools in modern machine learning, and they now play a central role in numerous applications related to intelligent mobility and public transport systems. The increasing availability of heterogeneous data from onboard sensors, interior cameras, APC systems (Automatic Passenger Counting), smart-ticketing infrastructures, and IoT devices has made neural networks essential for modelling complex phenomena in the transport field. Their ability to learn robust representations directly from raw, noisy, or unstructured data makes them particularly suitable for addressing the operational challenges of contemporary public transport, characterized by high variability, complex environmental conditions, and the need for real-time decision support. This section presents the fundamental theoretical principles that govern the design, mathematical formulation, and operational behaviour of neural networks, with a focus on concepts directly relevant to applications in urban mobility and public transport. The discussion begins with the mathematical foundations that describe information propagation across network layers, emphasizing the role of activation functions, weight matrices, and bias vectors in forming internal representations capable of capturing recurring patterns within data collected from vehicles in motion. Such concepts are essential for understanding, for example, how a CNN can detect passengers in heavily occluded environments or how sequence-based models can learn the temporal dynamics of boarding and alighting events. Subsequently, the section examines the theoretical principles behind neural learning mechanisms, loss functions, optimization algorithms, and generalization criteria—critical elements for ensuring the reliability of systems deployed in real operational contexts such as fleet management, service planning, or decision support for transit agencies. Special attention is given to gradient-based optimization and modern optimizers, which allow the effective training of deep models even when data are noisy or collected under uncontrolled environmental conditions. The backpropagation algorithm is then explored in detail, outlining its central role in minimizing prediction error and discussing numerical challenges that may arise in deep architectures. These aspects are particularly relevant in the public transport domain, where model robustness and stability are essential to guarantee reliable performance under highly dynamic and unpredictable scenarios. Finally, the section discusses key regularization and optimization strategies adopted to enhance the model's ability to generalize across

unseen conditions—such as changes in passenger flow, varying illumination, modifications in vehicle layout, or anomalies in traffic circulation. Complementing this, the operational principles guiding the training and tuning of the model used in this research are presented, including hyper-parameter selection, dataset preparation, validation schemes, and performance metrics specifically suited for counting and monitoring tasks in public transportation. Overall, this introduction provides the theoretical foundation needed to justify the neural architecture adopted in the present study and to understand the optimization strategies implemented within the developed system, ensuring coherence and relevance to the public transportation domain.

2.1.1 Neural Networks Model in Transport Field

In public transport applications, neural models are widely used for tasks such as automatic passenger counting, demand forecasting, arrival-time prediction, mobility flow analysis, visual recognition of street-level elements, and vehicle condition monitoring.

Among the most established architectures, the **Multilayer Perceptron (MLP)** represents one of the simplest yet most effective neural models. In transportation research, MLPs have been applied to traffic prediction, OD flow modelling, and smart-card data analysis. Their ability to approximate non-linear relationships makes them particularly suitable for rider ship forecasting and behavioural modelling, especially in systems where mobility patterns fluctuate throughout the day. Technically, an MLP processes information through a sequence of fully connected layers, where each neuron applies a non-linear transformation of the form:

$$h^{(l)} = \sigma\Big(W^{(l)}h^{(l-1)} + b^{(l)}\Big)$$

allowing the network to learn complex mappings between historical inputs and predicted outputs. Transportation studies applying MLPs to short-term traffic forecasting show that this structure is capable of capturing daily oscillations, peak-hour effects, and sudden fluctuations caused by congestion or exogenous factors. The training process relies on back-propagation, which iteratively adjusts weights to minimize prediction error. Evidence from empirical studies confirms that the MLP [33] can follow real traffic evolution with high fidelity Fig. 2.1 .

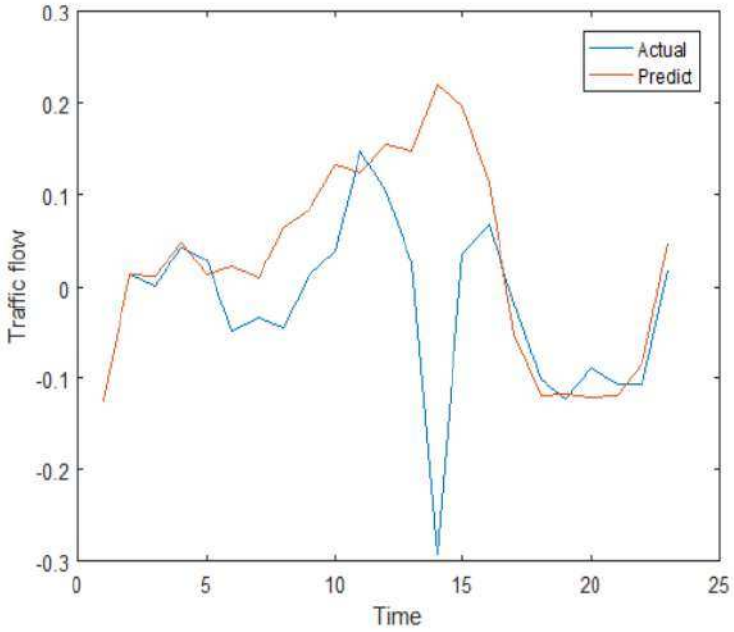


FIGURE 2.1: Network predicted and actual data

Even though it is not a recurrent model, it often learns temporal dependencies effectively, maintaining robust performance even when input data contain noise—a typical condition for loop-detector measurements. Error measures such as RMSE demonstrate high predictive accuracy, reinforcing the relevance of the MLP for modelling non-linear transport dynamics. These strengths translate naturally to public transport applications, where demand levels, passenger arrivals, and occupancy patterns follow similarly irregular and non-linear behaviours. The balance between simplicity and expressive power of MLPs makes them an effective solution, especially when datasets are limited or operational interpretability is required.

Another neural network architecture widely adopted in public transport applications is the **Convolutional Neural Network (CNN)**. CNNs have revolutionized visual perception in the transport domain due to their ability to automatically extract spatially meaningful features from images. This paradigm shift was largely initiated by the seminal work of Krizhevsky et al., who demonstrated the effectiveness of deep CNNs on large-scale visual recognition tasks through the introduction of AlexNet, achieving unprecedented performance on the ImageNet benchmark and marking a turning point in the adoption of deep learning for computer vision applications [6].

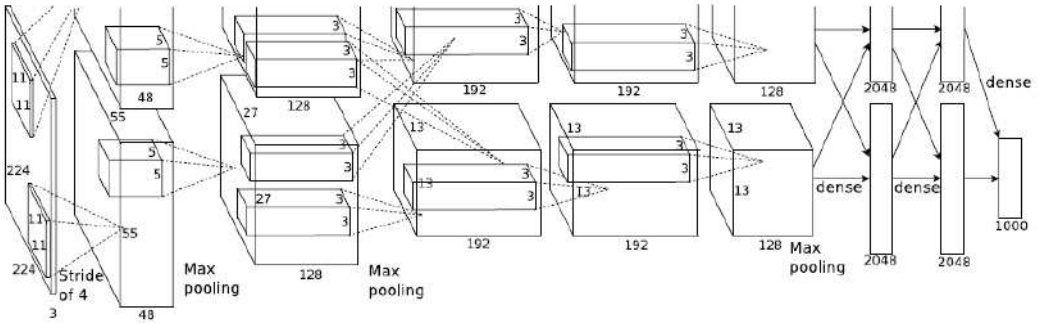


FIGURE 2.2: An illustration of the architecture of our CNN [6]

CNNs are extensively employed for passenger detection, onboard crowd density estimation, and street-level scene understanding, including pedestrian recognition, bicycle detection, obstacle identification, and pavement surface assessment. The layered architecture of CNNs—characterized by convolutional operations, nonlinear activations, pooling mechanisms, and hierarchical feature abstraction—makes them particularly well suited to handling the visual complexity of onboard monitoring systems. In such environments, variations in illumination, frequent occlusions, camera motion, and limited image resolution pose significant challenges to traditional computer vision techniques.

At the core of CNNs lies the convolution operation, which enables the network to focus on local spatial neighborhoods and progressively construct high-level semantic representations from low-level visual patterns. Mathematically, CNNs rely on the discrete convolution operation, traditionally defined as in LeCun [34]:

$$(F_k^{(l)})(x,y) = \sum_i \sum_j W_k^{(l)}(i,j)\, X^{(l-1)}(x+i,\, y+j) + b_k^{(l)},$$

where $W_k^{(l)}$ contains the learnable parameters of the filter, $X^{(l-1)}$ is the input feature map, and $b_k^{(l)}$ is a bias term. Through repeated applications of this operation, CNNs are able to extract edges, silhouettes, textures, and object-level Fig. 2.2 features that are essential for interpreting transport scenes in a reliable and scalable way. A representative example in this domain is the work by Alahi [7] , which proposes a robust real-time pedestrian detection system based on low-resolution cameras.

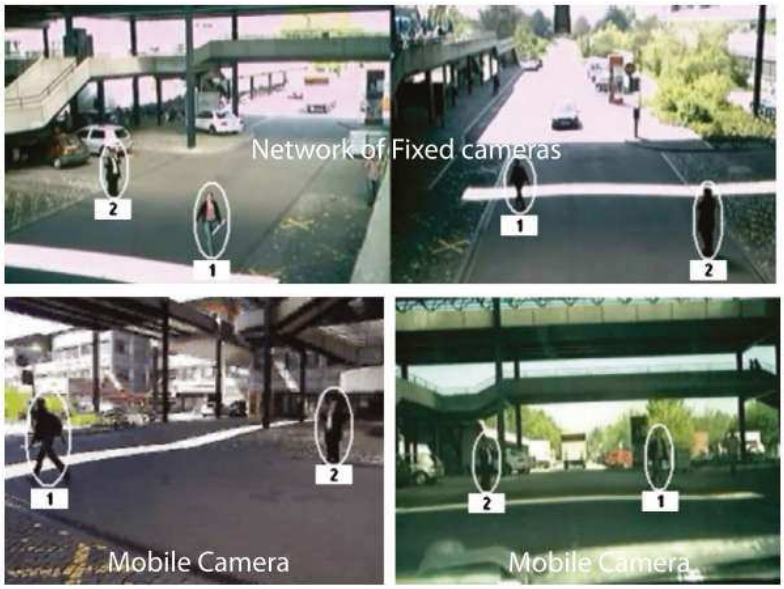


FIGURE 2.3: Collaborative network of cameras [7]

The paper highlights scenarios highly relevant Fig. 2.3 to public transport, where buses and trams operate in environments with dense pedestrian flow, frequent occlusions, rapid viewpoint changes, and cluttered urban backgrounds. The proposed approach integrates multi-view visual descriptors and sparse feature representations, inspired by convolutional filtering principles, enabling detection performance that remains robust even under degraded imaging conditions. The study demonstrates that combining information from fixed and mobile cameras increases detection distance and improves the system's responsiveness—an important requirement for safety-critical transport operations, where early identification of pedestrians around the vehicle can reduce collision risk and enhance situational awareness. Overall, CNNs constitute a fundamental building block for modern vision-based transport analytics. Their ability to generalize across different vehicle types, lighting conditions, and urban morphologies makes them a powerful tool for developing intelligent public transport systems capable of perception, monitoring, and decision-making in real time.

Another neural network architecture widely used in public transport analytics is the Recurrent Neural Network (**RNN**) family. Unlike feed-forward models, **RNNs** incorporate temporal feedback connections that allow them to process sequential data and learn dependencies that unfold over time. This capability was formally established in early recurrent learning frameworks based on backpropagation through time, which laid the theoretical foundation for sequence modelling in neural networks. Building upon this paradigm, Long Short-Term Memory (**LSTM**) networks were introduced to address the vanishing gradient problem affecting standard RNNs, enabling the effective learning of long-range temporal dependencies through gated memory mechanisms [35]. More recently, Gated Recurrent Unit (**GRU**) architectures have been proposed as a computationally efficient alternative to LSTMs, retaining comparable modelling capacity while reducing architectural complexity. These properties make RNN-based models particularly effective for capturing the

temporal dynamics that characterize mobility systems. In the public transport domain, RNNs are frequently employed for passenger demand forecasting, boarding and alighting sequence modelling, travel-time prediction, and the analysis of smart-card or onboard sensor time series, where sequential correlations and time-dependent patterns play a critical role.

From a mathematical perspective, LSTMs overcome the limitations of classical RNNs—such as vanishing and exploding gradients—by introducing gated memory units. The internal dynamics of an LSTM cell can be summarised through the following equations, where the forget gate f_t, input gate i_t, and output gate o_t regulate information flow across time [8]:

$$
\begin{aligned}
f_t &= \sigma(W_f h_{t-1} + U_f x_t + b_f), \\
i_t &= \sigma(W_i h_{t-1} + U_i x_t + b_i), \\
\tilde{c}_t &= \tanh(W_c h_{t-1} + U_c x_t + b_c), \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t, \\
o_t &= \sigma(W_o h_{t-1} + U_o x_t + b_o), \\
h_t &= o_t \odot \tanh(c_t),
\end{aligned}
$$

where x_t is the input at time t, h_t the hidden state, c_t the memory cell Fig. 2.4 , and $\sigma(\cdot)$ the sigmoid activation function.

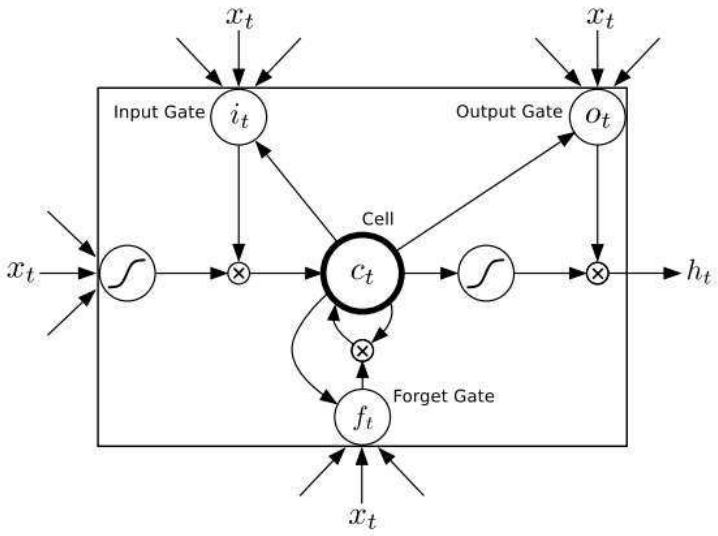


FIGURE 2.4: Long Short-term Memory Cell [8]

These mechanisms enable LSTMs to retain long-term dependencies while filtering irrelevant temporal fluctuations—crucial capabilities for transport data that naturally exhibit peaks, cycles, and irregular patterns. A representative example of RNN effectiveness in the transport field is the LSTM model proposed by Meng et al. [36], designed to predict short-term road traffic speed using large-scale GPS

positioning data. The article demonstrates that this architecture significantly outperforms conventional ANN and standard LSTM models across multiple prediction horizons, especially in conditions with fluctuating traffic patterns such as weekends or low-sample periods. The integration of DTW preprocessing helps the model adapt to slight temporal misalignments, while the attention mechanism enhances sensitivity to relevant road-speed features.

Collectively, MLPs, CNNs, and RNNs offer complementary capabilities for public transport analytics: MLPs capture nonlinear relationships in aggregated datasets, CNNs interpret complex visual information, and RNNs model the temporal evolution of mobility flows. Their integration enables more accurate monitoring and predictive systems, contributing to intelligent, adaptive, and data-driven public transport services.

2.1.2 Learning, Optimization, and Generalization in Neural Networks

Training deep neural networks relies on a rich set of theoretical principles that govern how models learn from data, how parameters are optimized, and which strategies are required to ensure robust generalization. Understanding these mechanisms is essential for designing reliable architectures, minimizing prediction errors, and preventing overfitting — especially in real-world applications such as intelligent mobility and public transport systems. In supervised learning, the model is provided with labeled data and attempts to approximate an unknown function by minimizing a **loss function**. Popular examples include the *Mean Squared Error (**MSE**)* for regression and the *Cross-Entropy Loss* for classification. The loss function plays a central role in shaping the learning dynamics, determining how the model interprets errors and how the optimizer adjusts the parameters. Foundational works such as Rumelhart [37] emphasise how the structure of the loss influences convergence stability and learning efficiency. Parameter optimisation is typically achieved via **gradient descent**, which updates the model weights in the direction that most reduces the loss. Given a cost function L, the update rule is:

$$W \leftarrow W - \eta \frac{\partial \mathcal{L}}{\partial W}$$

where η is the learning rate — a critical hyper parameter controlling the balance between convergence speed and training stability. Modern variants such as **Adam** [38], **RMSProp**, and **momentum-based SGD** have been widely adopted due to their ability to handle noisy gradients and highly non-convex loss landscapes. The computation of gradients is performed through the back-propagation algorithm, which applies the chain rule to propagate the error backward through the network. This enables efficient optimization of millions of parameters but also introduces challenges in deep architectures: gradients may vanish or explode, leading to slow convergence or numerical instability. To improve **generalization**, neural networks employ several **regularization techniques**. Among the most widely used are:

- **L1 and L2 weight decay**, which constrain parameter magnitudes;
- **dropout** which prevents co-adaptation by randomly deactivating neurons;
- **early stopping**, which halts training upon validation plateau;
- **data augmentation**, which expands dataset diversity by generating transformed versions of existing samples.

More recent approaches — including *batch normalization*, *mixup*, and *label smoothing* — have demonstrated additional gains in stability and generalization, with several studies published in IEEE Transactions on Pattern Analysis and Machine Intelligence. Altogether, these ingredients form the theoretical backbone of neural network training. By integrating appropriate loss functions, robust optimization algorithms, and effective regularization strategies, modern neural networks achieve high accuracy and stability, making them well-suited for complex environments such as public transportation systems, where data can be noisy, heterogeneous, and driven by dynamic temporal patterns.

2.1.3 Model Selection, Training and Optimization

In this study, a **Convolutional Neural Network (CNN)** was selected as the primary model for internal and external bus environment analysis due to its ability to effectively process video streams captured by onboard cameras. CNNs are the state-of-the-art in visual perception tasks thanks to their capacity to extract hierarchical spatial features and maintain robustness under challenging conditions such as illumination changes, occlusions, and camera motion. These properties are essential in public transport environments, where both interior scenes (passengers, luggage, crowding) and exterior scenes (pedestrians, bicycles, potholes, vehicles) exhibit high variability. Recenet studies, such as [39] and [9], demonstrate the reliability of CNN-based object detection in complex urban transport settings.

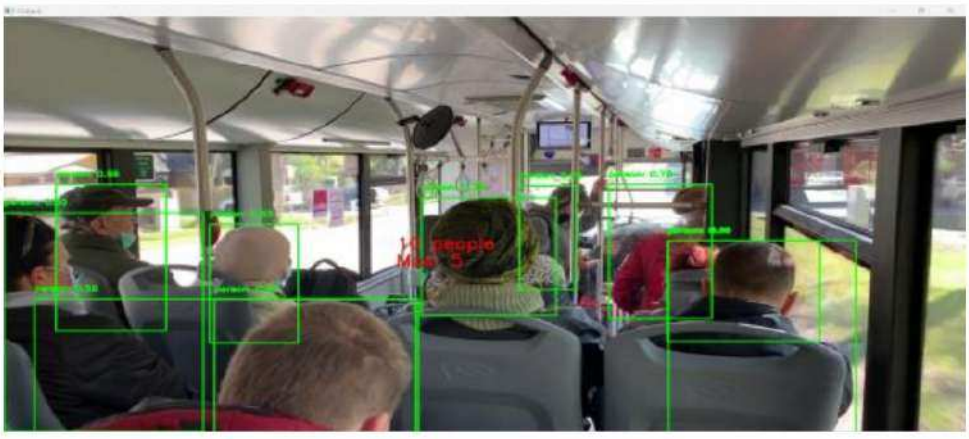


FIGURE 2.5: Visualization Result [9]

Within the CNN family, YOLOv5 was chosen due to its ability to perform real-time object detection Fig. 2.5 with high accuracy and low computational cost, which is fundamental in a moving-vehicle context. YOLOv5 integrates an optimized CNN backbone and detection head that allow efficient extraction and propagation of spatial features. The development of the model begins with building a dedicated dataset from video frames captured by cameras installed inside and outside public transport vehicles. Each image/frame is manually annotated with bounding boxes for target

classes (passengers, pedestrians, obstacles, street hazards, etc.). To increase robustness against variations such as lighting, viewpoint changes, occlusion, and crowd density, data augmentation, including rotations, brightness variation, scaling, flipping, is applied. Such techniques are essential in transport-focused vision systems and are highly recommended in the literature. The training process leverages transfer learning, initializing YOLOv5 with weights pretrained on large generic datasets (e.g., COCO) and fine-tuning on task-specific transport data. Hyperparameter tuning includes:

- selection of a suitable **learning rate**, often managed via decaying schedules;
- adjustment of **batch size** according to available computational resources;
- tuning **IoU thresholds** for detection, to handle small or partially occluded objects typical inside vehicles or in crowded streets;
- selection of the **model variant** (small/medium/large) depending on computational constraints and latency requirements.

Recent work demonstrates that a lightweight YOLOv5-based [40] bus passenger detector using cross-stage bottleneck modules can achieve high frame rates ($\tilde{4}0$ FPS) with reduced model size while preserving detection accuracy validating the feasibility of deploying such models in real transport vehicles. To assess model performance rigorously, standard object detection metrics are adopted:

- **mAP@0.5** and **mAP@0.5:0.95** for detection accuracy across varying IoU thresholds;
- **Precision** — fraction of correct detections among all detections;
- **Recall** — fraction of ground-truth objects correctly detected;
- **F1-score** — harmonic mean of precision and recall;
- **Inference speed (FPS)** — crucial for real-time deployment;
- **Counting error / occupancy error** when the goal is passenger counting or occupancy estimation.

A recent field study comparing a low-cost YOLOv5-based APC system against commercial sensors on metro trains demonstrated acceptable accuracy levels, confirming the practicality of camera-based deep learning approaches for public transport occupancy monitoring. The combination of CNN-based architectures (implemented through YOLOv5), onboard camera datasets, data augmentation techniques, transfer learning, and rigorous evaluation metrics constitutes a concrete, robust, and scalable solution for visual perception in public transportation. This configuration enables the system to manage the visual complexity of real-world environments, deliver real-time inference, adapt to highly variable operational scenarios, and provide actionable data for service planning, fleet management, safety enhancement, and overall operational optimization.

In summary, this architecture fully meets the objectives of the present thesis: achieving high accuracy, ensuring computational efficiency, enabling operational flexibility, and maintaining strong alignment with the practical requirements of modern intelligent mobility systems.

2.2 Optical Principles for Vision-Based Systems

Vision-based artificial intelligence systems fundamentally depend on the quality and fidelity of the visual data they receive, making the optical characteristics of the camera a critical component of the perception pipeline. This section introduces the key optical and photonic principles that guide the selection and configuration of imaging devices in public transport applications, with particular attention to their impact on the accuracy, robustness, and reliability of computer vision algorithms. Among the most relevant parameters, spatial resolution determines the level of detail captured in an image and directly affects the system's ability to identify small, distant, or partially occluded objects. The field of view (**FOV**) defines the angular extent of the observable scene and influences overall coverage, blind spots, and the need for multi-camera configurations inside and outside the vehicle. Parameters such as sensor sensitivity and signal-to-noise ratio (**SNR**) govern image quality under challenging illumination conditions, including nighttime operations, tunnels, or rapid light transitions , ensuring that AI models receive clean and informative inputs. Additional factors such as optical distortion, depth of field (DoF), and lens aberrations introduce geometric or photometric artifacts that must be considered when selecting a camera or corrected through calibration. Distortion and **DoF** constraints can significantly affect tasks such as object detection, tracking, and feature extraction in CNN-based systems, while noise and low-light limitations may degrade real-time inference performance. Overall, this section provides the theoretical foundation needed to understand how optical design choices influence data quality, deep learning performance, and the operational effectiveness of vision-based systems deployed in real public transport environments.

2.2.1 Fundamentals of Imaging and Optical Formation

Image formation is the physical foundation of all vision-based systems and arises from the complex interaction between light, the observed scene, and the camera sensor. Understanding these mechanisms is essential for the proper selection of imaging hardware and for anticipating how optical characteristics will influence the quality of the data that feed deep learning models. Image quality emerges from a chain of interconnected physical, optical, and electronic processes, each contributing to how faithfully reality is transcribed into digital form.

The formation process begins with the light–object interaction: incident illumination is reflected, absorbed, or scattered by surfaces in the scene. The reflected rays are then gathered by the lens and recombined according to the classical thin-lens model, whose governing equations are taken from Hecht's *Optics* [10] a foundational reference in geometric and physical optics. In particular, the thin-lens equation

$$\frac{1}{f} = \frac{1}{d_o} + \frac{1}{d_i},$$

which determines the geometry under which a real image is physically projected onto the sensor plane. Magnification is governed by

$$m = \frac{h_i}{h_o} = -\frac{d_i}{d_o},$$

indicating how the apparent size of objects varies with the distance from the camera.

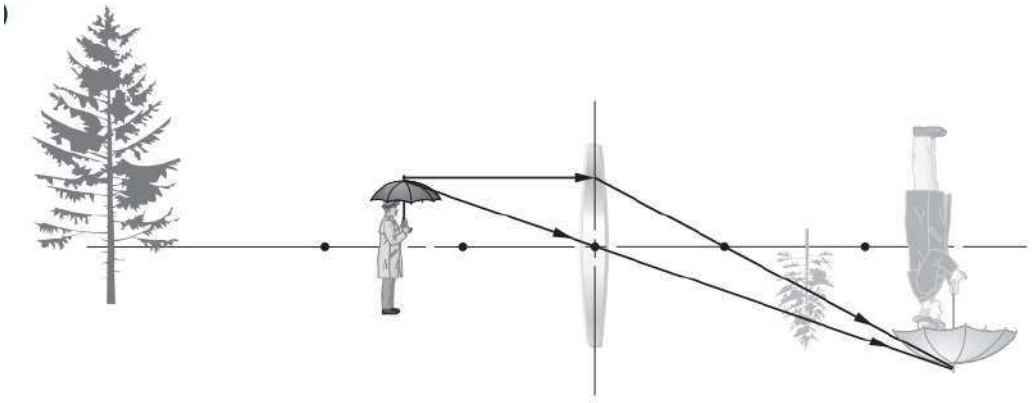


FIGURE 2.6: The image-forming behavior of a thin positive lens [10]

When geometric conditions are not met for physical projection, the system forms a virtual image based on the perceived extension of diverging rays. Key optical parameters strongly shape the resulting image. The focal length controls magnification and field of view (FOV); for a sensor of width W, the horizontal FOV Fig. 2.6 is given by

$$\mathrm{FOV}_h = 2\arctan\left(\frac{W}{2f}\right),$$

which explains why short focal lengths provide wide-angle coverage ideal for interior monitoring, while longer focal lengths enhance detail resolution for exterior pedestrian detection, bicycle tracking, or obstacle identification. The aperture regulates the amount of light reaching the sensor and determines the depth of field (DoF). A simplified approximation is

$$\mathrm{DoF} \approx \frac{2Ncs^2}{f^2},$$

where N is the f-number, c the circle of confusion, and s the focus distance. Wide apertures improve low-light performance, whereas narrower apertures provide greater image sharpness across multiple depth planes.

The quality of the lens further affects the image through chromatic aberrations, geometric distortions, and vignetting. Radial distortion, for instance, is often modelled as

$$r_{\mathrm{dist}} = r\left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right),$$

and can significantly alter the spatial consistency of the data. If uncorrected, such distortions disrupt the internal feature representations extracted by CNNs, reducing both performance and stability.

Equally critical is the image sensor, which converts light into digital signals and dictates sensitivity, dynamic range, and noise behaviour. In this study, a particularly relevant imaging component is the Sony IMX219, an 8-megapixel CMOS sensor frequently used in embedded vision platforms such as the NVIDIA Jetson. Featuring 1.12 μm pixels, the IMX219 provides a balanced trade-off between spatial resolution and light sensitivity, a stable signal-to-noise ratio (SNR), and reliable performance

under rapidly changing illumination—an everyday challenge in public transport environments (e.g., tunnels, shadows, glare). The SNR can be approximated by

$$\mathrm{SNR} = 20 \log_{10} \left(\frac{S}{N} \right),$$

highlighting the relationship between useful signal S and noise N. Its compatibility with interchangeable M12 optics additionally allows precise tuning of focal length and field of view, enabling the sensor to adapt seamlessly to both interior wide-angle monitoring and more focused exterior perception tasks. Sensors of this class provide sufficient image quality to support CNN-based detection in dynamic and unconstrained operational conditions.

Together, these optical and sensing principles provide the theoretical foundation required to understand how lens design, aperture, focal length, and sensor architecture jointly determine the quality of the data supplied to vision-based AI systems. In the context of on-board perception for public transport, mastering these interactions is crucial to ensuring accuracy, robustness, and operational reliability.

2.2.2 Spatial Resolution and Pixel Geometry

Spatial resolution is one of the fundamental determinants of image quality in vision-based systems, as it defines the system's ability to represent fine details and distinguish closely spaced structures within a scene. In digital imaging, spatial resolution depends on a combination of sensor characteristics—such as pixel size and pixel density—and optical factors including lens quality, focal length, and inherent aberrations. Understanding these interactions is essential for designing an imaging pipeline that provides data of sufficient fidelity for deep learning models, particularly in demanding environments like public transport vehicles.

At the sensor level, pixel geometry plays a central role. Smaller pixels allow more sampling points within the same sensor area, theoretically increasing the spatial resolution. However, reducing pixel size also decreases the number of photons captured per pixel, which in turn lowers the signal-to-noise ratio (SNR) and may introduce quantization artifacts, especially under low-light conditions. Conversely, larger pixels collect more photons, improving sensitivity and dynamic range but limiting the achievable spatial resolution for a given sensor size. This trade-off is well-documented in CMOS [41] sensor design and must be evaluated in relation to the operating conditions and the tasks assigned to the AI model. Pixel density—often expressed in megapixels—defines the total number of sampling sites available for image reconstruction. However, resolution cannot be assessed independently of optical quality. The resolving power of the lens, typically described by the modulation transfer function [42] (MTF), constrains the maximum spatial frequency that can be faithfully reproduced. Even a high-density sensor cannot capture fine details if the optical system introduces blur, chromatic aberration, or distortion, limiting the effective resolution of the entire imaging chain. Thus, spatial resolution is determined not only by sensor specifications but by the combined performance of sensor and optics. Geometric factors such as pixel pitch, fill factor, and microlens design further shape the effective sensitivity and resolution. For example, modern sensors like the Sony IMX219 employ microlens arrays to maximize photon collection efficiency, partially compensating for the reduced light sensitivity associated with small pixel sizes (1.12 µm). This improves both the contrast and clarity of high-frequency features,

supporting tasks such as passenger detection, object tracking, and anomaly recognition that rely on precise spatial detail. In practical applications, especially those involving convolutional neural networks, spatial resolution directly affects the quality of the extracted features. Higher-resolution imagery allows deeper CNN layers to encode more discriminative representations of edges, textures, and object boundaries. However, it also increases computational cost, memory load, and bandwidth requirements—factors particularly relevant for real-time processing on embedded platforms deployed in public transport vehicles.

Overall, spatial resolution and pixel geometry form the technical backbone of image representation. The careful balance between pixel size, sensor density, and optical quality determines the system's capacity to capture fine details reliably and to provide high-quality input data for deep learning models operating under real-world constraints.

2.2.3 Field of View and Lens Configuration

FOV is a fundamental geometric property of an imaging system and defines the angular extent of the scene that can be captured by the camera. Together with focal length and sensor dimensions, the FOV determines how much of the environment is visible in a single frame and directly influences the geometric perspective of the acquired image. As described in Hecht's [10], the FOV is governed by the basic geometry of lens projection and can be derived from the thin-lens approximation.

For a sensor of width W and a lens with focal length f, the horizontal field of view is expressed as:

$$\mathrm{FOV}_h = 2 \arctan\left(\frac{W}{2f}\right),$$

while the vertical FOV for a sensor of height H follows:

$$\mathrm{FOV}_v = 2 \arctan\left(\frac{H}{2f}\right).$$

The diagonal FOV, often used in camera datasheets, is correspondingly defined as:

$$\mathrm{FOV}_d = 2 \arctan\left(\frac{D}{2f}\right),$$

where $D = \sqrt{W^2 + H^2}$ is the diagonal of the sensor. These expressions, widely documented in the imaging literature [41], demonstrate how shorter focal lengths increase the angular coverage, producing a wide-angle perspective, whereas longer focal lengths restrict the viewing area and compress the spatial geometry of the scene.

Focal length not only defines the extent of the visible scene but also impacts geometric perspective. Lenses with short focal length cause parallel lines to diverge and exaggerate depth perception, while telephoto lenses reduce the apparent spatial separation between objects, flattening the scene. These effects directly influence tasks such as object detection, distance estimation, and geometric reconstruction, as the visual cues processed by deep neural networks depend on the inherent perspective encoded in the image. Moreover, the effective FOV is constrained by the optical design of the lens. Aberrations at the periphery of the image—such as distortion, coma, and astigmatism—can reduce the usable field of view even when the nominal FOV is large. Radial distortion, alters the apparent position of objects near the

edges of the frame and must be corrected through calibration to ensure accurate spatial measurements and stable feature extraction in CNN-based systems. In practical applications, especially onboard public transport vehicles, the selection of FOV and focal length determines whether the imaging system is optimized for wide-area situational awareness (interior cabin monitoring) or for narrow-angle, detail-specific perception (pedestrian detection, obstacle recognition, or road-surface inspection). Consequently, proper lens configuration is crucial to ensuring that vision-based AI modules receive geometrically consistent and information-rich imagery.

2.2.4 Impact of Optical Parameters on AI Data Quality

The performance of vision-based artificial intelligence systems is fundamentally constrained by the quality of the visual data they receive, and this quality is directly shaped by the optical configuration of the imaging system. Optical parameters such as focal length, field of view (FOV), aperture, lens distortion, pixel size, and sensor noise collectively influence how accurately the camera can represent the physical environment. Since convolutional neural networks operate on spatial patterns extracted from images, any degradation introduced at the optical level propagates through the entire perception pipeline, affecting both detection accuracy and model robustness. Spatial resolution and pixel geometry determine the system's ability to encode fine textures and object boundaries. If resolution is insufficient or pixel noise dominates, neural networks may fail to extract stable features, leading to loss of precision in tasks such as passenger counting, obstacle detection, and road condition analysis. Similarly, the choice of focal length and FOV shapes the geometric representation of the scene. Wide-angle lenses ensure broader coverage but introduce perspective exaggeration and potential edge distortion; narrower lenses provide higher spatial detail but restrict contextual awareness. These trade-offs must be balanced according to the operational requirements of public transport scenarios, where interior monitoring and exterior perception impose different geometric constraints.

In summary, optical parameters do not merely affect image aesthetics; they directly shape the statistical properties of the data on which AI models rely. The interaction between lens design, sensor architecture, and environmental illumination determines whether the captured data are sufficiently rich, stable, and noise-free to support high-performance deep learning. For public transport applications, where safety, reliability, and real-time operation are critical, understanding and optimizing these optical factors is essential for building perception systems that deliver consistent and trustworthy results.

2.3 Camera Geometry and Calibration Principles

Camera geometry plays a fundamental role in the performance of vision-based artificial intelligence systems, as the geometric and optical parameters of the camera determine how the three-dimensional world is projected onto the two-dimensional image plane. A thorough understanding of these parameters is essential to ensure that the data acquired by the camera are consistent, metrically meaningful, and suitable for deep neural network processing. This requirement is particularly critical in public transport environments, such as the interior of buses, where illumination conditions vary rapidly and crowding introduces frequent occlusions. When wide-angle lenses with large fields of view are employed, geometric distortions become

even more prominent, further emphasizing the importance of proper camera calibration.

Camera geometry is typically formalized through the pinhole camera model [43], in which a 3D point $P = (X, Y, Z)$ is projected onto the image plane by:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix},$$

where (u, v) are the image coordinates, s is a scale factor, $\mathbf{K}$ is the intrinsic camera matrix containing focal lengths and principal point, and $\mathbf{R}, \mathbf{t}$ represent the extrinsic rotation and translation [43]. In summary, understanding and calibrating the geometric properties of a camera is indispensable for achieving reliable AI perception. Especially in dynamic bus environments, characterised by motion, occlusions, illumination changes, and wide-angle distortions—accurate calibration ensures that deep neural networks operate on stable and geometrically coherent visual data.

2.3.1 Intrinsic Parameters and Camera Modelling

Intrinsic parameters describe the internal geometric and optical properties of a camera and define how three-dimensional points in the world are projected onto the two-dimensional image plane. These parameters characterize the camera independently of its spatial position and are therefore fundamental both for geometric reconstruction and for the coherent interpretation of visual data by artificial intelligence systems. In the pinhole camera model, intrinsic parameters are compactly represented by the intrinsic matrix:

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix},$$

where f_x and f_y denote the focal lengths expressed in pixel units along the horizontal and vertical axes, c_x and c_y indicate the coordinates of the principal point (typically located near the image center), and s represents the skew factor associated with non-orthogonality of the sensor axes. As described by Zhang [43], this matrix defines the mapping between 3D rays and their 2D projections under ideal, distortion-free conditions.

The focal length determines the projection scale and controls magnification: larger focal lengths narrow the field of view and increase spatial detail, whereas shorter focal lengths widen the scene but make the projection more susceptible to distortions. The principal point identifies the intersection of the optical axis with the image plane; inaccuracies in estimating this parameter introduce systematic shifts in image geometry. The skew factor, although often negligible in modern sensors, becomes relevant when the pixel grid is not perfectly rectangular.

Intrinsic parameters also include optical distortion coefficients, which model deviations from the ideal pinhole projection. Radial distortion, expressed as:

$$r_{\text{dist}} = r \left(1 + k_1 r^2 + k_2 r^4 + k_3 r^6\right),$$

and tangential distortion:

$$\Delta u = 2p_1uv + p_2(r^2 + 2u^2), \qquad \Delta v = p_1(r^2 + 2v^2) + 2p_2uv,$$

modify the apparent position of projected points, especially when using wide-angle lenses frequently employed in bus interiors. Without proper modelling and calibration, such distortions compromise the metric consistency of the image and destabilize feature extraction performed by convolutional neural networks.

By combining the intrinsic matrix $\mathbf{K}$ with the distortion coefficients, the camera model defines the transformation:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix},$$

which links the 3D camera coordinates (X_c, Y_c, Z_c) to the image coordinates (u, v). This relationship governs the projection geometry and determines how metric properties such as distances, angles, and shapes are preserved or distorted.

Accurate modelling of intrinsic parameters is therefore essential in AI applications for public transport. Tasks such as passenger counting, obstacle detection, tracking, and scene analysis depend on stable geometric relationships across frames. In dynamic environments such as buses, precise calibration ensures that neural networks operate on visually stable and geometrically reliable data.

2.3.2 Extrinsic Parameters and Spatial Transformations

Extrinsic parameters describe the position and orientation of the camera with respect to the external world and determine how three-dimensional points are expressed in the camera reference frame. While intrinsic parameters define the internal geometry of the imaging process, extrinsic parameters specify the spatial relationship between the camera and the observed environment. These parameters are fundamental for tasks requiring metrical interpretation of the scene, such as passenger localization, trajectory estimation, mapping, and multimodal sensor fusion.

The extrinsic parameters consist of a rotation matrix $\mathbf{R} \in SO(3)$ and a translation vector $\mathbf{t} \in \mathbb{R}^3$, which together form the rigid-body transformation between the world coordinate system and the camera coordinate system Fig. 2.7. A 3D point expressed in world coordinates $P_w = (X_w, Y_w, Z_w)^\top$ is transformed into the camera frame by:

$$P_c = \mathbf{R}P_w + \mathbf{t},$$

where $P_c = (X_c, Y_c, Z_c)^\top$ denotes the coordinates of the point as seen by the camera. As detailed in *Modelling and Control of Robot Manipulators* [11], such rigid transformations preserve distances and angles and constitute the mathematical foundation for many robotic and computer vision systems.

Using homogeneous coordinates, the transformation can be expressed as:

$$\begin{bmatrix} P_c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ 0\,0\,0 & 1 \end{bmatrix} \begin{bmatrix} P_w \\ 1 \end{bmatrix},$$

where the 4×4 homogeneous matrix encodes the complete spatial transformation. This representation is widely used because it allows rotations and translations to be combined through matrix multiplication, facilitating calibration and multi-view geometry.

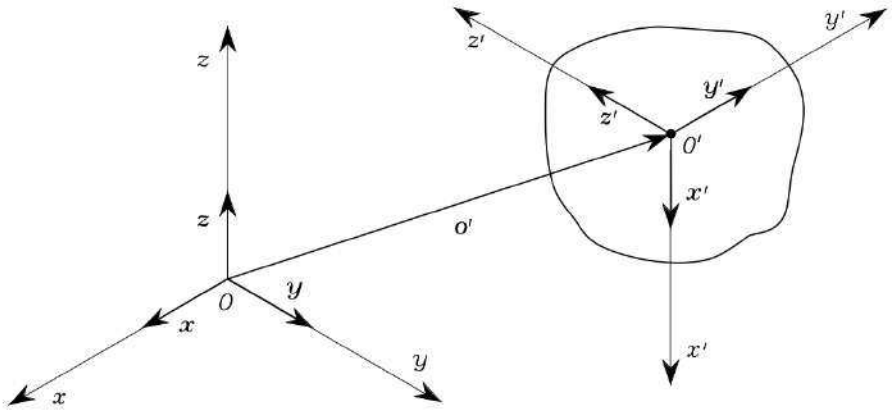


FIGURE 2.7: Position and orientation of a rigid body [11]

The rotation matrix **R** defines the camera orientation and can be parameterized by Euler angles, axis–angle representations, or quaternions. Errors in the estimation of **R** cause errors in depth perception and object localization, especially in scenes with strong perspective. The translation vector **t** represents the camera's position relative to the world origin and determines how the depth of the scene is mapped onto the camera coordinates.

In the standard projection equation:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix},$$

the extrinsic parameters **R** and **t** define how the 3D structure Fig. 2.7 of the world is mapped into the 2D image. They play a crucial role in determining spatial consistency across frames.

In public transport applications, extrinsic calibration becomes essential due to camera vibrations, non-static mounting positions, and the need for multi-camera fusion. Accurate estimation of extrinsic parameters ensures that geometric information remains consistent, enabling reliable detection, tracking, re-identification, and scene understanding. Precise modelling of spatial transformations is indispensable for any system requiring accurate interpretation of 3D geometry.

2.3.3 Perspective Projection and Object Geometry

Perspective projection governs the fundamental process through which the three-dimensional world is mapped onto the two-dimensional image plane of a camera. Under the ideal pinhole camera model, 3D points are projected through a single optical center onto the image surface, producing a geometry that preserves straight lines but induces non-linear scaling with depth. This representation captures the essence of how lenses alter the apparent size, position, and orientation of objects as their distance from the camera varies. Understanding these geometric effects is essential for designing reliable AI perception systems, especially in public transport

environments where objects (passengers, seats, obstacles) may appear at very different depths within the scene.

In the pinhole model [12], a 3D point $P = (X, Y, Z)$ is projected onto image coordinates (u, v) according to:

$$u = f_x \frac{X}{Z} + c_x, \qquad v = f_y \frac{Y}{Z} + c_y,$$

where f_x and f_y are the focal lengths in pixel units and (c_x, c_y) is the principal point. This formulation highlights the characteristic inverse-depth dependency of perspective projection: as Z increases, the projected dimensions shrink, producing the familiar foreshortening effect. Perspective also affects the perceived shape and aspect ratio of objects. The orientation of surfaces relative to the camera induces anisotropic scaling and edges parallel to the depth axis shrink more rapidly, while those orthogonal to it remain comparatively stable. This phenomenon influences the appearance of human silhouettes, seating geometry, and structural elements inside buses. Without an appropriate geometric model or calibration, AI systems may misinterpret object dimensions or incorrectly estimate spatial relationships.

Overall, perspective projection and its deviations from ideality profoundly influence the visual data used by AI systems. A correct understanding and calibration of these geometric effects enable more robust feature extraction, scale-aware detection, and consistent interpretation of object sizes and depths. For public transport applications—where wide-angle lenses, tight spaces, and dynamic scenes amplify perspective distortions—accounting for perspective geometry is essential to achieving reliable and accurate perception.

2.3.4 Calibration Techniques and Accuracy Evaluation

One of the most widely adopted approaches for camera calibration is Zhang's method [43], which introduced a flexible and easily implementable procedure based on observing a planar pattern—typically a checkerboard—from multiple orientations. The algorithm estimates intrinsic parameters, distortion coefficients, and extrinsic poses by minimizing the reprojection error between detected image points and their ideal projections under the pinhole model. Zhang's technique is particularly effective because it requires no prior knowledge of the camera pose, relies solely on a flat calibration target, and achieves high accuracy even with a relatively small set of images. Another classical approach is the Tsai calibration method [44], originally developed for industrial inspection systems. Tsai's method combines 3D reference measurements with nonlinear optimization to estimate focal length, principal point, and distortion parameters. Although historically influential, it requires more constrained acquisition setups and is generally less flexible than Zhang's method, especially when calibrating cameras with strong nonlinear distortion or extremely wide fields of view. For multi-camera and multi-view applications, calibration accuracy can be further enhanced through bundle adjustment, a nonlinear refinement technique that jointly optimizes all intrinsic and extrinsic parameters by minimizing the global reprojection error across all observations. As discussed by Hartley and Zisserman [45], bundle adjustment represents the state of the art in projective geometry optimization and is widely used in robotics, SLAM, and structure-from-motion.

To evaluate the quality of a calibration, several error metrics are commonly used. The most standard is the mean reprojection error:

$$e_{\text{repr}} = \frac{1}{N} \sum_{i=1}^{N} \|p_i - \hat{p}_i\|,$$

where p_i denotes a detected image point and $\hat{p}_i$ its corresponding projection based on the estimated parameters. A low reprojection error indicates strong agreement between the camera model and the real imaging geometry. Another widely used metric is the mean pixel deviation, which measures the average displacement between measured and predicted points, providing an intuitive assessment of practical accuracy.

In this study, Zhang's calibration method [43] was employed to estimate the intrinsic parameters of the onboard camera. The device features an extremely wide field of view of approximately 160°, which introduces significant radial distortion. To accurately characterize this distortion, a checkerboard calibration target Fig. 2.8 was photographed from multiple distances, angles, and perspectives, ensuring coverage of both central and peripheral regions of the field of view.

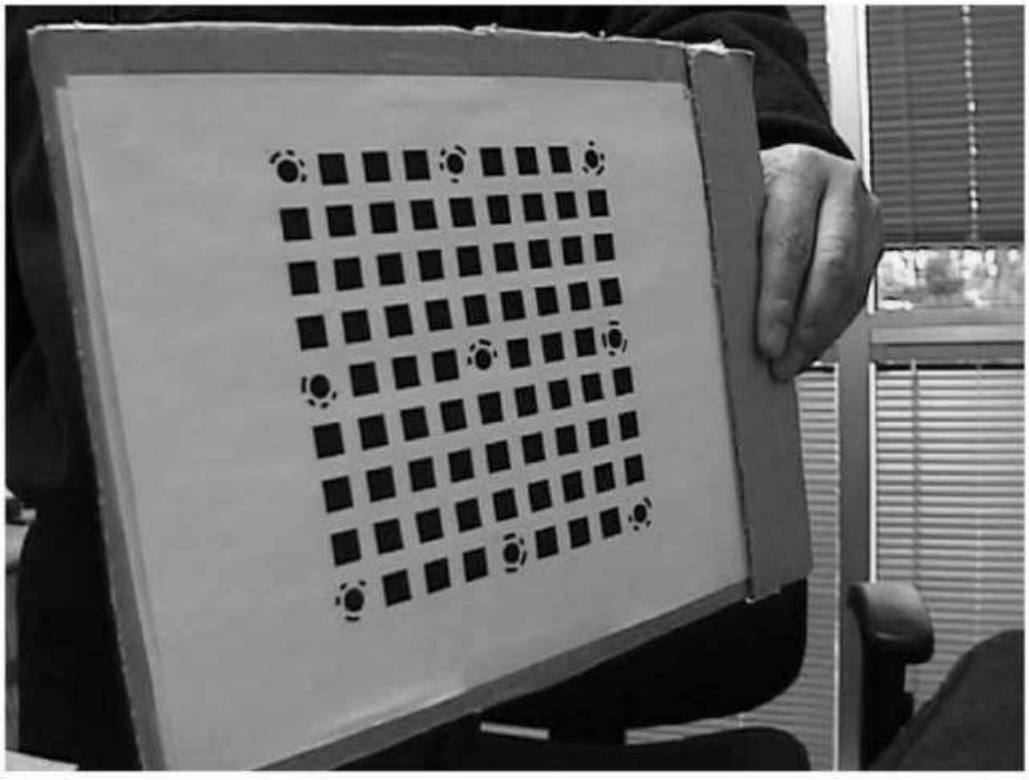


FIGURE 2.8: A sample image of the planar pattern used for camera calibration [12]

The intrinsic matrix **K** and distortion coefficients **D** were estimated using a Python

implementation based on the OpenCV library (`cv2`). The resulting parameters, extracted from the calibration output, are:

$$\mathbf{K} = \begin{bmatrix} 3.4764 \times 10^2 & 0 & 3.7454 \times 10^2 \\ 0 & 4.6088 \times 10^2 & 4.0666 \times 10^2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{D} = \begin{bmatrix} -0.33266 & 0.12279 & -0.00216 & 0.00105 & -0.02196 \end{bmatrix}$$

These matrices indicate a camera with anisotropic focal lengths ($f_x \neq f_y$), a principal point slightly offset from the geometric center of the image sensor, and distortion coefficients consistent with strong barrel distortion, typical of ultra-wide-angle lenses. The significant negative value of k_1 confirms the presence of pronounced radial distortion, while the small tangential terms (p_1, p_2) suggest minimal lens decentering.

The calibration process produced an accurate intrinsic model and a reliable distortion profile, enabling effective correction of the nonlinear warping inherent to wide-angle optics. This greatly improved the geometric consistency of the acquired images, which is essential for downstream AI tasks such as passenger detection, tracking, and semantic scene understanding in dynamic bus environments.

Chapter 3

YOLOv5 FOR THE URBAN MOBILITY MONITORING

3.1 The Core Model Behind the Detection Framework

At the foundation of this research lies an extensive and systematic investigation of deep learning architectures suitable for passenger counting and object detection in dynamic and resource-constrained environments. Throughout the preliminary analysis, several state-of-the-art models were reviewed and compared — ranging from traditional two-stage detectors such as R-CNN and Faster R-CNN to more efficient one-stage architectures including SSD, RetinaNet, and the YOLO family. This comparative evaluation highlighted that only a limited number of models achieve an effective compromise between accuracy, real-time inference, and computational efficiency — essential requirements for embedded and edge-based systems operating onboard public transport vehicles. Among these, the You Only Look Once (YOLO) architecture emerged as the most suitable solution, combining high detection performance with low-latency execution. This chapter provides an in-depth overview of the YOLOv5 model, which forms the core of the proposed detection framework, detailing its underlying principles, architecture, training methodology, and integration within the overall system.

3.1.1 Historical Background

The original YOLO algorithm, introduced by Joseph Redmon et al. in 2015, represented a fundamental paradigm shift in the field of object detection. Unlike the two-stage detectors such as R-CNN or Faster R-CNN—which first generate region proposals and then classify them—YOLO reformulated detection as a single regression problem. In this approach, bounding boxes and class probabilities are predicted directly from the entire image in a single network pass. This design drastically reduced computational overhead, allowing real-time performance without a significant loss of accuracy, and opened the door to early implementations in autonomous vehicles, robotics, and real-time surveillance.

Following this seminal work, the algorithm evolved through several major iterations. YOLOv2 (2016) introduced batch normalization and anchor box priors Fig. 3.1, improving localization and stability. YOLOv3 (2018) expanded the architecture with Darknet-53 as backbone and adopted multi-scale feature extraction to handle objects of different sizes, a key step toward robustness in complex scenes. YOLOv4 (2020) refined the approach further, integrating Spatial Pyramid Pooling (SPP), Path Aggregation Network (PANet), Mish activation functions, and advanced data augmentation techniques such as Mosaic and CutMix, significantly improving

the mean Average Precision (mAP) and generalization performance on real-world datasets [46].

Building on this lineage, YOLOv5, developed by Glenn Jocher and the Ultralytics team in 2020, marked a decisive turning point. It transitioned the entire framework from the original C/CUDA-based Darknet to the PyTorch deep learning framework. This migration was far more than a mere code translation—it enabled modularity, ease of customization, and compatibility with diverse hardware, including embedded GPU devices. The PyTorch implementation also allowed seamless integration with modern training pipelines, automatic mixed precision (AMP), and model export to ONNX and TensorRT formats, thereby expanding YOLO's usability for both research and production deployment [13]. Initially, the first PyTorch-based model incorporating these innovations was informally referred to as YOLOv4-PyTorch to reflect its architectural alignment with the contemporaneous Darknet YOLOv4. However, to avoid confusion and to emphasize the independence of its development branch, it was later rebranded as YOLOv5. This naming decision generated a brief debate within the computer vision community—some argued it should not be considered an official continuation of Redmon's YOLO line—but the superior modularity and training convenience of the PyTorch ecosystem soon consolidated YOLOv5 as the de facto standard for applied object detection. Unlike its predecessors, YOLOv5 is best understood not as a single static model, but as a continuously evolving framework. The open-source repository maintained by Ultralytics is under constant development, incorporating new architectures (e.g., YOLOv5n/s/m/l/x variants), enhanced data augmentations, and improved export pipelines.

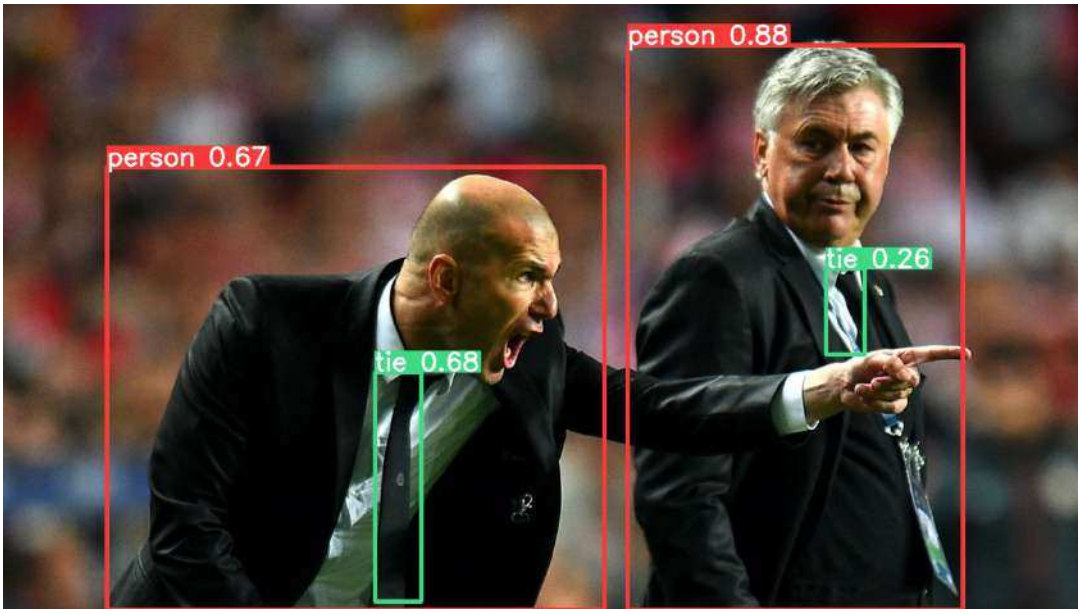


FIGURE 3.1: YoloV5 detection example [13]

This dynamism allows YOLOv5 to adapt quickly to new use cases—from autonomous driving to industrial inspection, public safety, and, as explored in this thesis, smart mobility and passenger analysis in public transport.

3.1.2 Model Architecture

Object detection, one of the core applications of YOLOv5, involves the extraction of salient visual features from an input image and their transformation into structured predictions that identify and localize multiple objects within the scene. Conventional object detection frameworks relied on multi-stage pipelines, where region proposals were first generated and then classified independently. The YOLO (You Only Look Once) architecture revolutionized this paradigm by introducing a fully end-to-end, differentiable detection process that unifies bounding-box regression and object classification into a single neural network. This design enables the model to process an entire image in a single forward pass, drastically reducing computational overhead while maintaining high detection accuracy — a key advantage for real-time and embedded inference.

The YOLO network is composed of three fundamental components: the backbone, neck, and detection head. The backbone, a convolutional neural network, encodes image information into multi-scale feature maps that capture both local and global spatial structures. These maps are then refined by the neck, a set of layers that integrate and enhance multi-level feature representations, ensuring that both high-resolution spatial detail and deeper semantic context are retained. Finally, the detection head interprets these fused features to generate bounding-box coordinates, confidence scores, and class probabilities for each detected object.

Over successive versions, particularly YOLOv4 and YOLOv5, the architecture has incorporated innovations from broader computer vision research — such as improved gradient flow mechanisms, multi-scale feature fusion, and advanced data augmentation — significantly improving the accuracy, efficiency, and generalization of object detection models. This synergistic evolution has positioned YOLOv5 as one of the most effective and practical frameworks for real-time visual perception tasks.

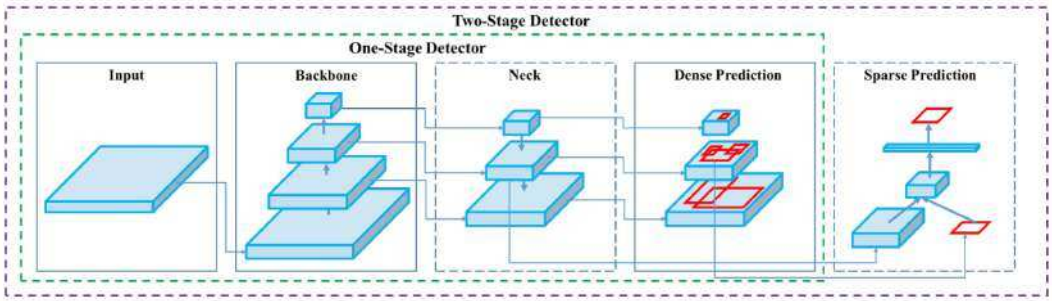


FIGURE 3.2: Yolo Detection Process [13]

Figure 3.2 illustrates the conceptual workflow distinguishing two-stage and one-stage detectors. Unlike two-stage architectures (e.g., Faster R-CNN), where object proposals and classification are handled sequentially, YOLOv5 performs all operations within a single, unified pipeline. The input image passes through three tightly connected components — the backbone, the neck, and the detection head — that collectively implement the one-stage detection strategy.

- **Backbone (CSPDarknet53)** [47] - The backbone is responsible for extracting hierarchical features from the input image. YOLOv5 adopts the CSPDarknet53 architecture, which applies the Cross Stage Partial (CSP) strategy to split feature maps into two parallel paths: one processed through dense convolutional blocks, and the other forwarded directly and later recombined. This division minimizes gradient duplication and redundant computation, improving both training stability and inference efficiency. CSP also optimizes GPU memory utilization, enabling high accuracy without increasing resource consumption — a crucial property for edge deployment on embedded GPU platforms. The resulting feature maps retain both low-level details (edges, textures) and high-level semantics (shapes, object context), which are propagated to the subsequent layers.

- **Neck (SPPF [48] + PANet [49])** - The neck serves as a feature aggregation bridge between the backbone and the detection head, enhancing the model's ability to detect objects across multiple scales. YOLOv5 integrates two key modules in this stage,the SPPF (Spatial Pyramid Pooling – Fast), which expands the receptive field through multi-kernel pooling operations without increasing inference time, and the PANet (Path Aggregation Network), which merges features in both top-down and bottom-up directions to strengthen multi-scale representations. This design allows YOLOv5 to effectively recognize small, partially occluded, or overlapping objects — a common condition in crowded public transport interiors and urban environments. The neck ensures that the information passed to the head is spatially coherent and semantically rich, supporting accurate multi-resolution prediction.

- **Detection Head** - The detection head performs bounding box regression and object classification using the features provided by the neck. YOLOv5 predicts across three different scales (stride 8, 16, and 32) to handle small, medium, and large objects simultaneously. Each grid cell on the feature map generates several anchor boxes, each characterized by its position, size, confidence score, and class probabilities. During training, the AutoAnchor algorithm automatically adapts anchor dimensions to match the object distribution within the dataset, improving accuracy and generalization. At inference, Non-Maximum Suppression (NMS) eliminates redundant overlapping detections, retaining only the most confident bounding boxes for each object. This multi-scale anchor-based mechanism allows YOLOv5 to detect small internal objects — such as passenger heads and upper bodies — as well as larger external elements, including pedestrians, bicycles, and vehicles.

Overall, YOLOv5 transforms raw visual input into structured object predictions through a sequence of progressively abstract feature representations. The backbone captures and encodes visual information, the neck harmonizes and scales these features, and the detection head produces interpretable predictions in real time. Unlike traditional two-stage architectures, YOLOv5 performs dense, end-to-end detection, eliminating the need for region proposals and achieving high precision with minimal latency.

These architectural characteristics — modularity, scalability, and computational efficiency — make YOLOv5 particularly suitable for embedded and onboard applications. In the context of this research, it enables real-time passenger counting and street-level object classification directly on GPU-embedded devices installed inside

public transport vehicles, reducing cloud dependency, minimizing latency, and ensuring reliable operation under the electrical and mechanical constraints typical of urban buses.

3.1.3 Training and Optimization

YOLOv5 employs a sophisticated training pipeline integrating several advanced augmentation and optimization techniques:

- **Mosaic augmentation** combines four random images into one during training, allowing the model to generalize across multiple contexts, object scales, and occlusion levels.
- **MixUp and Copy-Paste augmentations** simulate object overlap and lighting variation, improving robustness under dynamic conditions.
- **Multi-scale training** randomly resizes images between 0.5× and 1.5× per batch, enhancing cross-scale generalization.
- **Loss Function**: YOLOv5 optimizes detection through a composite loss combining Binary Cross Entropy (BCE) for classification and objectness with Complete Intersection over Union (CIoU) for bounding box regression. This dual optimization ensures precise spatial localization and reliable confidence calibration.

During inference, the model supports mixed-precision (FP16) computation and Exponential Moving Average (EMA) of weights to stabilize training and reduce latency, making it ideal for embedded GPUs. The architecture is available in five scalable variants — n, s, m, l, and x — balancing performance and computational load. For this research, YOLOv5s and YOLOv5m were selected, achieving an optimal trade-off between inference speed, energy efficiency, and detection accuracy.

3.2 YoloV5 Use Cases

YOLOv5, thanks to its speed, lightweight design, and architectural flexibility, has become a reference model in the field of real-time object detection. Its versatility enables deployment across a wide range of sectors, from medical imaging to autonomous driving, as well as industrial production and automated inspection. Below, we present the main macro-areas of application, each accompanied by an overview of the most relevant and representative use cases.

3.2.1 Medical and Pharmaceutical Domain

YOLOv5 has gained substantial relevance in the medical and pharmaceutical domain due to its ability to detect complex visual patterns with high accuracy and low latency—two properties that are essential in clinical decision support systems. One of the most active application areas is colorectal polyp detection. Polyps often exhibit highly variable shapes, irregular textures, and low contrast relative to surrounding mucosal tissue, making traditional detection methods fragile. The work presented in [50] demonstrates how YOLOv5 achieves robust localization of both small and flat polyps in colonoscopy videos, even under challenging conditions such as motion blur, uneven illumination, and specular reflections. By applying different

YOLOv5 variants (s, m, l) and optimizing anchor configurations, the authors report significant improvements in sensitivity—an essential metric for early colorectal cancer prevention. Another clinically relevant field is thoracic imaging, specifically the identification of lung nodules in X-ray and CT scans. Detecting early-stage nodules is notoriously difficult due to their subtle radiographic appearance, variable density, and potential overlap with ribs or vascular structures. The study in [14] applies YOLOv5 to datasets such as LIDC-IDRI and demonstrates that the model can accurately detect nodules Fig. 3.3 of varying diameters while maintaining a low false-positive rate.

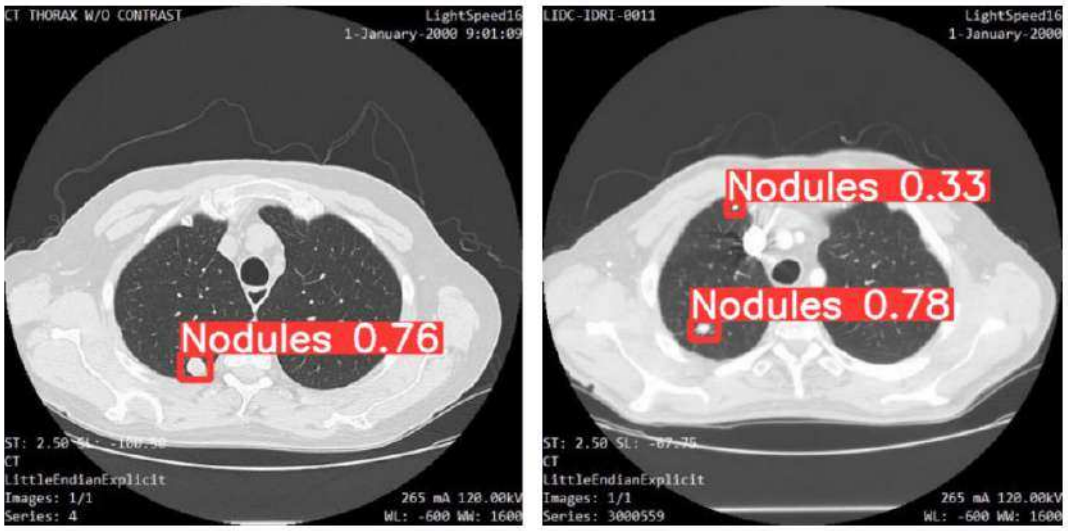


FIGURE 3.3: The Predicted location of Nodules [14]

The network's capability to generalize across noise levels and heterogeneous acquisition conditions illustrates why YOLOv5 is increasingly considered a reliable tool for assisting radiologists in screening workflows and reducing diagnostic burden. A third prominent research direction concerns skin cancer detection, where YOLOv5 is used to classify and localize malignant lesions such as melanoma, basal cell carcinoma, and squamous cell carcinoma. Skin lesions present significant variability in color, texture, shape, and boundary sharpness, making detection particularly challenging. The study in [51] extends the YOLOv5 architecture with additional attention mechanisms—such as SimAM and SEAttention modules—to enhance feature discrimination in dermoscopic images. Experimental results show improved performance on datasets like ISIC, with gains in precision, recall, and mAP. These enhancements make the method suitable for real-time diagnostic support, potentially deployable on portable dermatology tools or telemedicine systems. YOLOv5 balance of speed, accuracy, and architectural flexibility enables deployment in a variety of clinical contexts, from endoscopy and radiology suites to mobile diagnostic platforms. YOLOv5 effectively bridges traditional computer vision and modern AI-driven diagnostics, contributing to earlier detection, improved screening accuracy, and overall enhancement of healthcare workflows.

3.2.2 Industrial and Engineering Domain – Visual Inspection

YOLOv5 has become a key architecture in industrial visual inspection, where reliability, speed, and robustness against environmental variability are essential. A major application concerns surface defect detection in manufacturing processes. The study in [15] proposes an improved YOLOv5 framework that integrates Coordinate Attention and BiFPN to enhance multi-scale feature extraction and boost recall on fine-grained defects. Industrial surface imperfections—such as scratches, contamination, and deformation Fig. 3.4 .

FIGURE 3.4: Surface imperfections Detection [15]

The improved model achieves a recall of 91.6% and inference speeds up to 95 FPS, demonstrating its suitability for real-time deployment on assembly lines. A second engineering application involves printed circuit board (PCB) inspection, where YOLOv5 is used to detect missing components, soldering defects, and assembly errors. The work in [52] shows that YOLOv5 can outperform traditional rule-based inspection methods by learning robust feature representations even under variable lighting and imaging conditions. PCBs present complex backgrounds, densely packed components, and highly repetitive patterns; YOLOv5's multiscale detection capability is therefore crucial for distinguishing subtle defects such as micro-cracks or insufficient solder joints. Experimental evaluations highlight strong improvements in mAP and defect localization accuracy compared to older CNN-based approaches. A third widely studied application is the detection of defects in power-grid insulators, a safety-critical task where false negatives can lead to system instability or worker hazards. In [53], the authors propose enhancements to YOLOv5 based on self-attention and depthwise separable convolutions, enabling more effective detection of insulators of varying sizes and orientations. Their improved YOLOv5 network achieves 94.79% accuracy and 63.9 FPS, outperforming YOLOv3, YOLOv4, and Faster R-CNN. This demonstrates the model's suitability for UAV-based inspection along high-voltage transmission lines, where real-time performance and robustness to viewpoint changes are essential. Together, these studies confirm YOLOv5's status as a high-performance detector for industrial and engineering inspection tasks. Its architectural balance between speed and accuracy, combined with adaptability through attention mechanisms, feature pyramids, and lightweight convolutions,

makes it ideal for automated quality control, predictive maintenance, and safety monitoring across diverse industrial environments.

3.2.3 Automotive, Transportation, and Autonomous Driving Domain

YOLOv5 has been widely adopted in automotive and intelligent transportation systems due to its ability to combine high detection accuracy with real-time inference, two essential requirements for safety-critical perception tasks. One important line of research concerns aerial and elevated-view detection, which plays a key role in traffic monitoring, incident detection, and cooperative perception for connected vehicles. The study in [54] proposes an improved YOLOv5 architecture for UAV imagery by integrating attention mechanisms such as the Convolution-Swin Transformer Block (CSTB) and CBAM. These enhancements significantly increase detection performance for small and densely distributed objects frequently encountered in aerial traffic scenes, demonstrating notable gains in mean Average Precision without compromising model efficiency. A second contribution to the transportation domain relates to small-object detection, a long-standing challenge in autonomous driving. Vehicles must detect far-away traffic cones, road markers, signage, and other early-warning elements with limited pixel footprint. The work in [55] introduces model variants known as YOLO-Z, derived from YOLOv5 but modified through architectural optimizations such as high-resolution feature routing, DenseNet backbones, and enhanced feature pyramid structures. These adaptations yield significant improvements in detecting small objects under high-speed and dynamic racing conditions, highlighting how subtle structural changes can extend the perceptual range of autonomous vehicles. Another emerging application concerns traffic sign detection, foundational for Advanced Driver Assistance Systems (ADAS) and autonomous driving. The study in [16] shows that YOLOv5, when fine-tuned on domain-specific datasets, effectively localizes and classifies traffic signs Fig. 3.5 even across varying illumination, occlusions, and visual clutter.

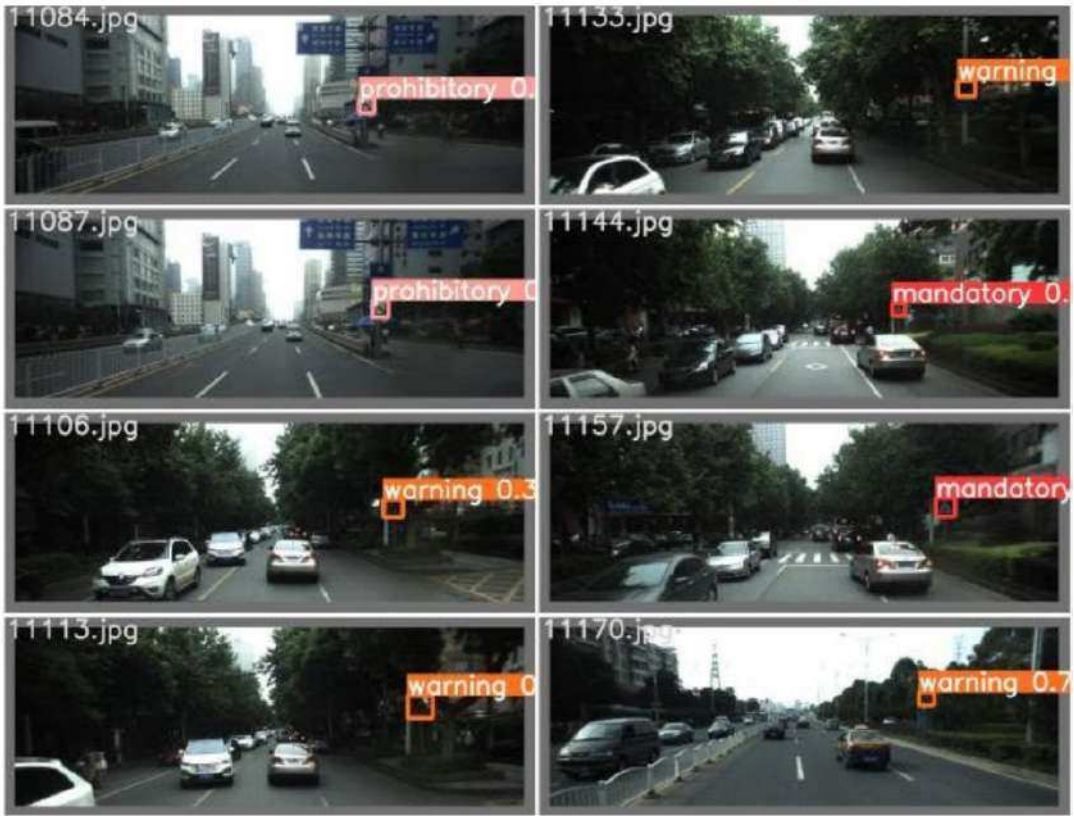


FIGURE 3.5: Traffic Sign Detection [16]

This robustness is essential for maintaining lane discipline, speed compliance, and safe navigation under real-world urban traffic variability. Finally, YOLOv5 has been successfully used to support traffic safety enforcement, particularly in detecting helmet violations among motorcyclists. The work in [56] presents a real-time detection pipeline capable of identifying riders, passengers, and helmet-wearing compliance across diverse environmental conditions. The system employs an ensemble of YOLOv5 models to improve robustness against motion blur, night-time scenes, and adverse weather, achieving competitive mAP performance in large-scale evaluations. Such systems demonstrate the growing role of deep learning in traffic law enforcement, risk mitigation, and smart-city analytics. Collectively, these studies confirm YOLOv5's prominent role as a versatile, high-performance detector capable of supporting a broad range of perception tasks within the automotive and transportation ecosystem, from autonomous driving to infrastructure monitoring and urban traffic safety.

3.2.4 Smart Cities, Security, and Urban Surveillance

YOLOv5 is playing an increasingly central role in smart-city applications, particularly in the domains of urban safety, public-space monitoring, and automated environmental management. One representative application concerns abandoned object detection in outdoor surveillance. The work in [17] proposes an enhanced YOLOv5s model integrated with StrongSORT tracking to address typical challenges of urban environments, such as illumination changes, shadows, occlusions, and moving backgrounds.

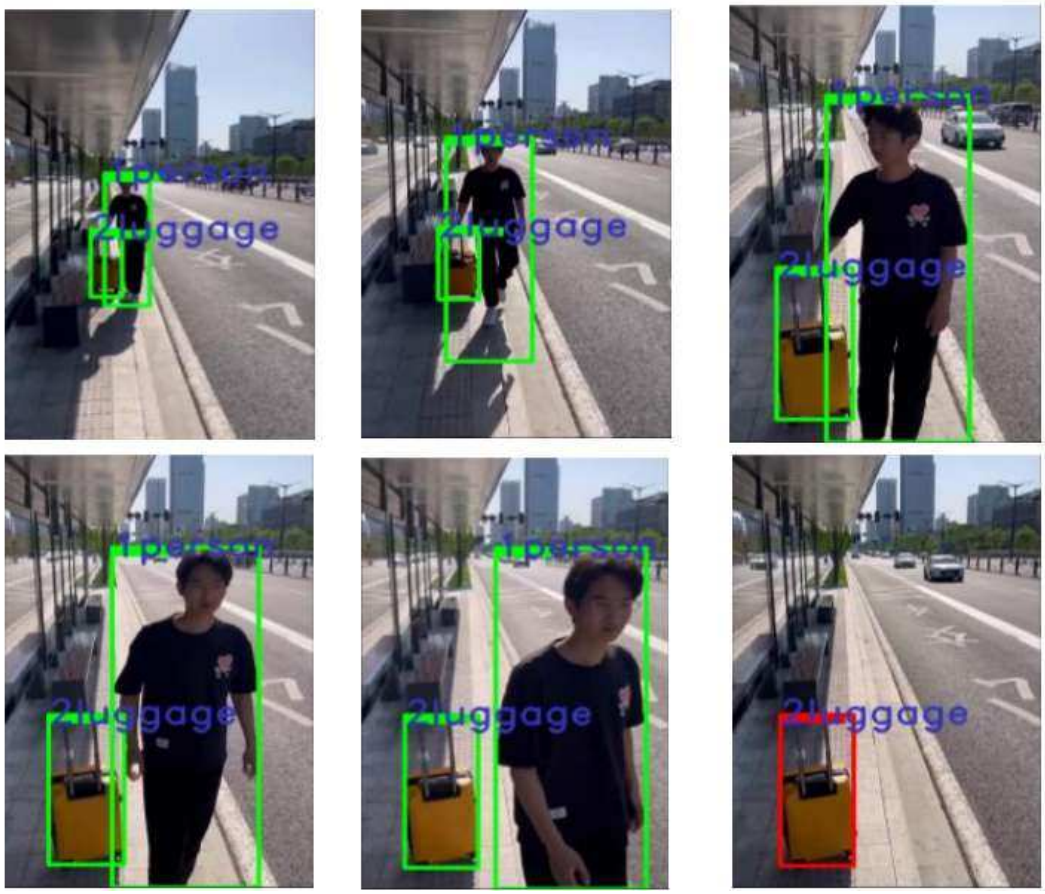


FIGURE 3.6: Detection results for video sequence [17]

By introducing improved feature-extraction modules—including the SPPFC2fC block—the system enhances multi-scale representation and maintains stable detection accuracy in complex public settings such as bus stops or pedestrian zones. The combination of detection and trajectory-based association allows robust identification of potentially abandoned items, offering clear benefits for public safety in smart-city infrastructures Fig. 3.6. A second application area involves automated waste classification, a critical component of sustainable urban management. The study in [57] presents an improved YOLOv5-based model designed to classify garbage directly at the source, supporting intelligent waste-sorting systems for smart cities. By simplifying network layers to increase inference speed and adapting the loss function to improve large-object detection efficiency, the model enables high-accuracy waste recognition even when deployed on low-power embedded devices such as Raspberry Pi. This approach demonstrates how lightweight deep-learning detectors can support scalable environmental monitoring, reduce manual sorting effort, and facilitate efficient recycling workflows. Finally, YOLOv5 has been successfully adopted in crowd monitoring and safety analysis, particularly during large-scale events. The research in [58] explores a detection-based framework to estimate crowd density and identify critical behavioral patterns in stadiums and other high-capacity venues. YOLOv5 is used to detect individual spectators under conditions of high density, variable viewpoints, and strong occlusions—typical scenarios in urban mass

gatherings. By providing accurate real-time estimates of the number of people in different sectors, the system supports proactive safety management, evacuation planning, and resource allocation, all of which are essential for modern smart-city operations. Overall, these studies demonstrate how YOLOv5 constitutes a versatile and scalable foundation for intelligent urban surveillance, enabling cities to enhance public safety, improve sustainability, and optimize operational efficiency across a wide range of real-world scenarios.

3.3 Rationale for Selecting YOLOv5 in the Proposed System

The selection of YOLOv5 as the primary detection model in the proposed system is motivated by a combination of architectural, operational, and empirical factors. These elements collectively demonstrate its suitability for real-time perception in public transport environments. The main reasons are summarized below:

- **Dual-layer applicability for internal and external perception.** YOLOv5 proves highly effective for the integrated detection strategy adopted in this thesis, supporting both passenger analysis inside the vehicle and environmental object recognition outside. Its unified architecture allows a single model family to perform consistently across heterogeneous visual domains.
- **Optimized architecture for edge deployment.** The lightweight convolutional design of YOLOv5, combined with its streamlined inference pipeline, enables efficient execution on embedded GPU platforms. This allows the entire perception system to operate directly onboard public transport vehicles, eliminating dependency on cloud infrastructure and reducing latency.
- **Robustness under challenging real-world conditions.** YOLOv5 exhibits strong resilience to occlusions, fluctuating lighting, dynamic backgrounds, reflections, and dense scenes—conditions frequently encountered both inside buses (crowded interiors, reflective surfaces) and outside (urban traffic, pedestrians, road irregularities).
- **Consistent performance demonstrated across multiple research domains.** Its effectiveness is supported by extensive literature:
  - In the *medical domain*, YOLOv5 accurately detects subtle and fine-grained patterns, demonstrating sensitivity to small anomalies.
  - In *industrial and engineering inspection*, it maintains stable performance under difficult imaging conditions and reliably identifies structural and surface defects.
  - In the *automotive and transportation sector*, it excels in small-object detection, traffic sign recognition, and hazard identification—tasks analogous to those required for bus exterior monitoring.
  - In *smart-city surveillance*, it has proven capable of abandoned-object detection, crowd counting, and waste classification in highly dynamic environments.

  The consistency of YOLOv5 across these diverse domains highlights its generalization capabilities and operational reliability.

- **Advanced multi-scale feature extraction.** YOLOv5 integrates Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) mechanisms, enabling robust detection of objects of varying sizes—crucial for differentiating passengers inside the bus and identifying small or distant elements in urban environments.
- **Mature data augmentation strategies.** Techniques such as Mosaic augmentation, adaptive anchor computation, and Class Label Smoothing enhance the model's ability to generalize, especially in visually complex and unstable environments.
- **Compatibility with embedded inference optimization tools.** The model integrates seamlessly with TensorRT, ONNX Runtime, and half-precision (FP16) execution, enabling real-time inference on GPU-equipped onboard systems. This ensures scalability, maintainability, and cost efficiency.
- **Alignment with system-level operational constraints.** YOLOv5 offers a balanced trade-off between accuracy, latency, and computational load, making it suitable for continuous deployment in moving vehicles with limited energy and processing budgets.

In conclusion, YOLOv5 emerges as the most suitable detection architecture for the proposed public transport monitoring system. Its balance between computational efficiency, perceptual robustness, and deployment flexibility enables accurate, autonomous, and scalable detection of both passengers and external environmental elements, ultimately supporting data-driven mobility analysis and the optimization of public transport operations.

Chapter 4

SYSTEM ARCHITECTURE AND OPERATING WORKFLOW

4.1 General System Architecture

The design of the proposed setup, illustrated in Figure 4.1, is the result of an extensive experimental campaign aimed at determining the optimal installation configuration for both the internal and external monitoring subsystems. During the development phase, multiple camera placements and viewing angles were evaluated inside the bus to achieve a full and continuous coverage of the passenger area while minimizing occlusions and redundant overlapping regions. Various mounting heights and tilt angles were tested, as even small changes in camera orientation significantly affect detection accuracy, especially in high-density scenarios or under variable lighting conditions. The adopted configuration represents the best trade-off between visibility, accuracy, and mechanical feasibility, ensuring comprehensive visual coverage of the bus interior and door areas with a limited number of cameras and minimal calibration effort.

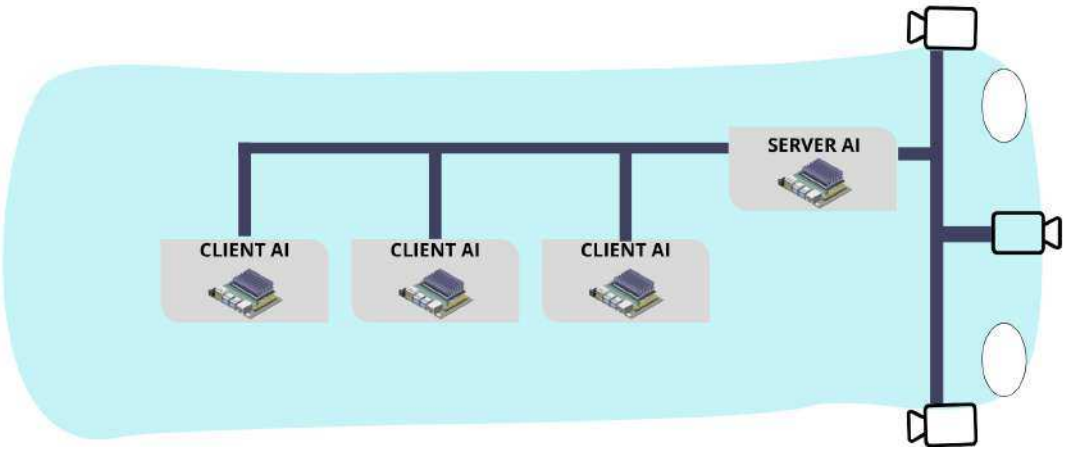


FIGURE 4.1: Setup architecture.

Similarly, several external camera placements were assessed to guarantee complete coverage of the surroundings, focusing on the frontal road section, the vehicle sides, and the driver's blind spots. The final layout allows continuous detection of

obstacles, pedestrians, and road surface conditions while maintaining a balance between field of view and image distortion. Compared to conventional single-camera approaches, the proposed distributed setup ensures redundancy and robustness, reducing blind zones and enabling accurate spatial correlation between the detected elements and their real-world locations.

To properly contextualize the proposed system, it is useful to consider the architecture of conventional passenger-counting solutions commonly employed in public transport. These systems typically rely on infrared or Time-of-Flight (ToF) sensors installed above the vehicle doors and connected to dedicated control units. While such technologies perform adequately for basic in–out flow detection, their operation is highly sensitive to reflections, lighting variations, and environmental interference. In crowded scenarios, where multiple passengers pass simultaneously or stand close to the sensor, signal overlap and occlusions significantly degrade accuracy, often leading to missed detections or false counts.

Furthermore, these traditional systems estimate total occupancy indirectly, by calculating the cumulative difference between entries and exits. This differential counting mechanism makes them particularly vulnerable to error propagation — a single miscounted event can compromise the overall total, producing significant deviations over the course of a route or a service day. As a result, their effectiveness decreases markedly under real operating conditions, especially during peak hours or in vehicles with frequent passenger movements.

In contrast, the proposed framework introduces a fully integrated and vision-based architecture, in which internal monitoring, external perception, and positional data are processed within a coordinated, intelligent structure. This configuration overcomes the limitations of conventional systems by providing direct observation and tracking of passenger flow, while generating multimodal, geo-referenced data that support real-time analysis and advanced transport management applications.

4.1.1 Electrical Layout

From a hardware perspective, the proposed system is organized around a centralized electrical distribution unit that supplies power and manages signal routing between the main processing modules. As illustrated in the electrical layout 4.2, the 24 V power line from the bus is routed to the electrical distribution unit, which serves as the main hub for both energy and signal management.

The Electrical Distribution Unit (EDU) supplies regulated 24 V power to all AI processing nodes — one AI server and three AI clients — through independent fused lines, ensuring electrical protection and full compliance with automotive safety standards. In addition to power distribution, the EDU interfaces with the bus door electrical signals (P1, P2, P3), enabling real-time detection of door-opening and -closing events. These signals are forwarded to the AI server, which uses them to synchronize passenger counting operations with the vehicle's stop phases, ensuring precise temporal alignment between video analysis and door status.

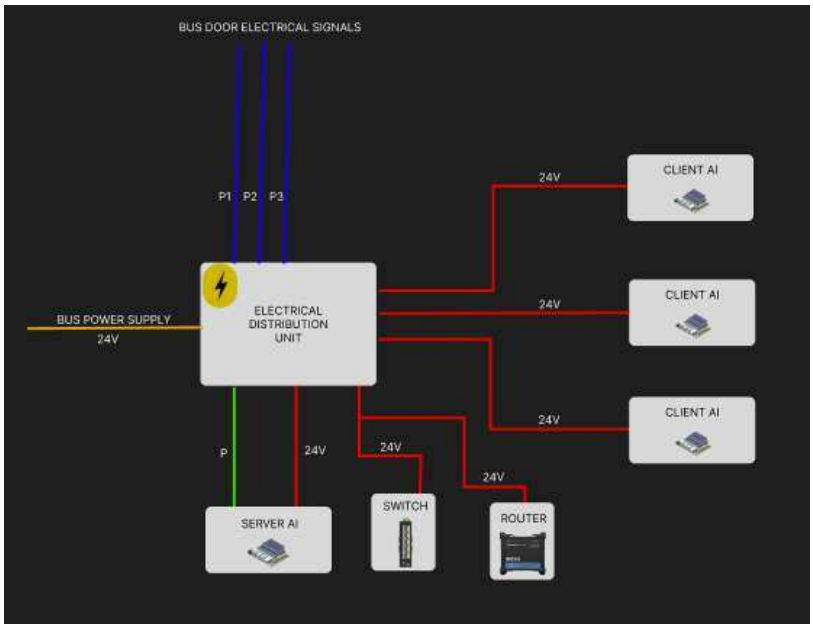


FIGURE 4.2: Electrical layout of the integrated system.

The EDU also powers a single 8-port Power-over-Ethernet (PoE) switch, which simultaneously provides power and data connectivity to all system components. On the same 24 V power line, a dedicated industrial router is supplied, enabling network connectivity via a SIM card for remote data transmission, system monitoring, and cloud synchronization. This configuration ensures a reliable communication link between the onboard AI infrastructure and the central management platform, even in mobile or low-coverage environments.

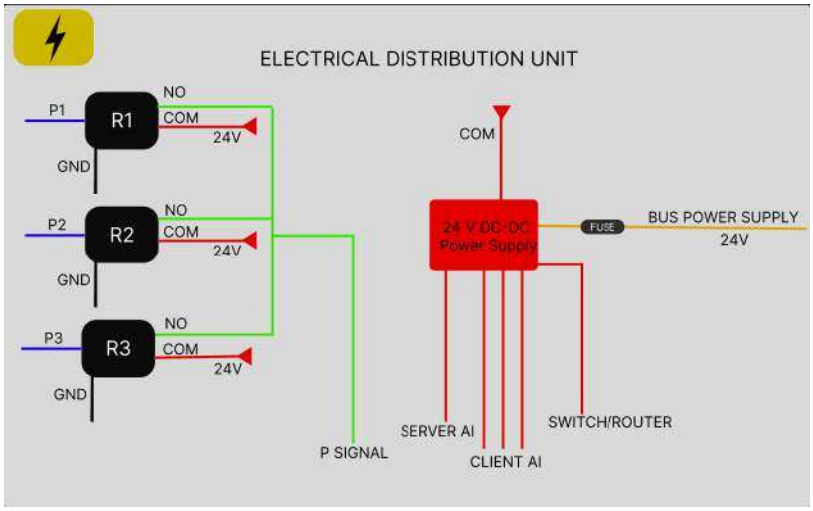


FIGURE 4.3: Electrical Distribution Unit

The Electrical Distribution Unit (EDU) illustrated in Figure4.3performs both power regulation and signal interface functions within the bus's AI monitoring architecture. Its purpose is twofold:

1. Provide stabilized 24 V DC power to all electronic subsystems (AI Server, AI Clients, PoE Switch and Router),
2. Acquire and isolate the door electrical control signals (P1, P2, P3) for synchronization of passenger counting events.

At the heart of the EDU is a Mean Well HDR-60-24 4.4 power module. This compact, ultra-slim DIN-rail device delivers up to 60 W at 24 V DC / 2.5 A, with a high efficiency of approximately 90%, and a universal AC input range from 9 V to 264 V AC. It is fully compliant with UL 508, IEC 60950-1, and EN 61558-2-16, offering Class II insulation, short-circuit, overload, and over-voltage protection, and a working temperature range from $-30\,^{\circ}\mathrm{C}$ to $+70\,^{\circ}\mathrm{C}$.

FIGURE 4.4: 24V DC-DC Power Supply

Mounted on a standard TS-35/7.5 or TS-35/15 rail, the HDR-60 ensures compactness and easy maintenance. Its constant-current limiting mode enables stable operation even with the inductive and capacitive loads typical of AI-based embedded systems.

A 10 A fuse positioned on the 24 V input line from the bus power supply protects the downstream equipment against over-current conditions and accidental short circuits. The regulated output from the HDR-60 feeds the AI Server, the AI Clients, and the PoE Switch, which powers the external cameras. This configuration guarantees electrical isolation from the vehicle's primary supply and reduces the propagation of voltage spikes generated by the alternator or auxiliary circuits.

The entire setup operates on the 24 V electrical standard commonly adopted in public transport vehicles. Power and communication lines are arranged in parallel to minimize electromagnetic interference (EMI) and to facilitate maintenance and scalability. The modular and distributed design ensures robustness and continuous operation under the mechanical and electrical constraints typical of urban buses, while remaining expandable for future AI modules or additional sensors. The EDU integrates three TE Connectivity F4 automotive relays 4.5 (V23134 series) , identified in the schematic as R1, R2, and R3.

FIGURE 4.5: F4 automotive relays

Each relay operates with a 24 V DC coil and uses Form A (1 NO) contact arrangement, capable of switching up to 40 A continuous current at 28 V DC under automotive conditions. The F4 relays feature a contact gap > 0.8 mm, silver-based contacts, and a rugged construction conforming to ISO 7588-1 (Mini ISO plug-in). Their electrical endurance exceeds 100,000 operations under nominal load, while mechanical endurance is above 1 million cycles. They are rated for operation from $-40\,^{\circ}\mathrm{C}$ to $+125\,^{\circ}\mathrm{C}$, with vibration resistance > 5 g and shock resistance > 20 g. This ensures reliability even in the harsh vibration and temperature environment typical of public-transport vehicles. Each relay is connected to one of the door control lines (P1, P2, P3), with the common (COM) terminal at 24 V and the normally open (NO) terminal switching the door status signal (P Signal) to the AI server input. When a door opens, the respective relay energizes, closing the NO contact and pulling the P Signal line high. This clean, galvanically isolated signal allows precise correlation between door states and passenger-counting events, minimizing noise interference from the vehicle's electrical network. The EDU architecture separates high-current power delivery from low-voltage control signals, achieving both safety and electromagnetic compatibility. By using automotive-grade F4 relays and a certified HDR-60 power module, the system ensures compliance with EN 61000-3-2, EN 55032, and EN 61000-4 EMC standards. All wiring follows the 24 V bus standard, and the use of a single grounded COM line avoids potential loops. Parallel routing of power and communication lines reduces electromagnetic interference (EMI) and simplifies servicing. The modular design allows the EDU to be replaced or extended without altering the rest of the onboard architecture. It guarantees stable power, signal isolation, and fault-tolerant operation, ensuring that the AI server continues to synchronize passenger detection even under adverse electrical conditions, such as voltage transients or partial system faults.

4.1.2 Network Connection

Figure 4.6 shows the network interconnection architecture of the onboard system, where the AI Server operates as the central coordination node of the entire infrastructure. It is responsible for data aggregation, external video stream processing, GPS data integration, and the synchronization of distributed AI clients, which manage local camera processing and transmit metadata in real time over a high-speed Gigabit Ethernet network.

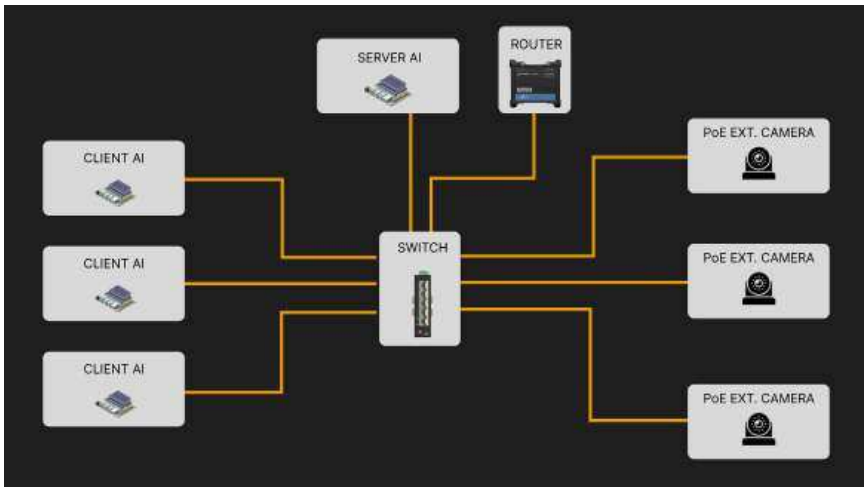


FIGURE 4.6: Network interconnection diagram of the onboard AI system.

Each AI Client handles two high-resolution video streams from its associated cameras, performing object detection, passenger tracking, and local inference before forwarding processed results to the central AI Server. Communication between all AI units, external cameras, and network peripherals is achieved through an industrial managed PoE+ switch (Teltonika TSW202), which provides both power distribution and data connectivity across the system. This switch features 8 Gigabit PoE+ ports (IEEE 802.3af/at) capable of delivering up to 30 W per port with a total power budget of 248 W, and 2 SFP uplink ports for potential fiber extensions. It supports Layer 2 management with additional Layer 3 functions, VLAN segmentation, QoS prioritization, and industrial protocols such as PROFINET, EtherNet/IP, and MRP, ensuring deterministic and fault-tolerant communication.

The operating range of $-40\,^{\circ}\mathrm{C}$ to $+75\,^{\circ}\mathrm{C}$ and IP30-rated metal housing make it suitable for the demanding thermal and vibrational environment of public transport vehicles. Integrated DIN-rail mounting simplifies installation and maintenance.

In parallel, the network includes an industrial cellular router Teltonika RUT951 4.7 , powered on the same 24 V DC line as the switch. The router provides 4G LTE Cat 4, 3G, and 2G mobile connectivity through dual SIM slots, ensuring automatic failover between operators to guarantee continuous service availability.

FIGURE 4.7: RUT951

Equipped with four 10/100 Ethernet ports (one configurable as WAN), dual Wi-Fi antennas, and operating under RutOS (OpenWrt-based Linux), the RUT951 acts as the gateway between the onboard network and the cloud infrastructure, supporting secure VPN tunnels (OpenVPN, WireGuard, IPsec) and remote management via Teltonika RMS. The system transmits real-time data on passenger counts, GPS location, and environmental detections to the cloud using a dedicated SIM-based connection, allowing remote diagnostics, updates, and data visualization. The PoE+ switch distributes both power and data to the AI server, three AI clients, and three external IP cameras, as shown in Figure 4.6. This configuration eliminates redundant wiring, minimizes voltage drop across long runs, and enhances network reliability by integrating energy and data over a single, structured Ethernet backbone. The presence of the RUT951 router ensures redundant, broadband connectivity, enabling the system to remain fully operational and connected even in mobile or low-signal conditions, while also facilitating remote monitoring, configuration, and firmware updates. Overall, the combined use of the TSW202 and RUT951 establishes a robust and scalable communication backbone, capable of supporting high-bandwidth AI workloads and ensuring secure, continuous connectivity for fleet-wide data exchange and monitoring.

4.2 General Algorithm Schema

The proposed approach is organized into two concurrent and complementary procedures: internal monitoring and external monitoring. Both are managed by the onboard AI infrastructure, whose distributed architecture—powered through the 24 V bus system and interconnected via the industrial PoE switch and router described in the previous chapters—ensures real-time data acquisition, synchronization, and processing between the AI server and clients.

Once the convolutional neural network (CNN) modules deployed on both the AI server and clients are initialized and the video streams are activated, the two monitoring processes begin simultaneously. During internal monitoring, the AI server continuously acquires the door status signals from the vehicle's electrical system

through the EDU interface, detecting in real time when doors are opened or closed. Upon detecting an open door event, the server instructs the AI clients to activate the passenger flow analysis module, which observes the movement patterns of individuals near the doors using the onboard high-resolution cameras. This module does not perform the actual counting but rather estimates flow intensity and directionality—information that serves to characterize the dynamics of the stop, providing valuable contextual data about boarding and alighting activity, crowd density, and stop occupancy.

The passenger counting itself is instead performed after the doors are closed, ensuring that all passengers who entered or exited during the open-door phase are correctly accounted for. At that moment, the AI server synchronizes the counting results from all AI clients and associates them with the GPS position provided by the integrated module. This enables precise spatial referencing of passenger events, allowing the system to correlate boarding patterns with specific stops or segments of the route. By integrating both flow and counting data, the system provides a comprehensive characterization of each stop, including not only the number of boarding and alighting passengers but also temporal patterns, density indicators, and passenger circulation trends.

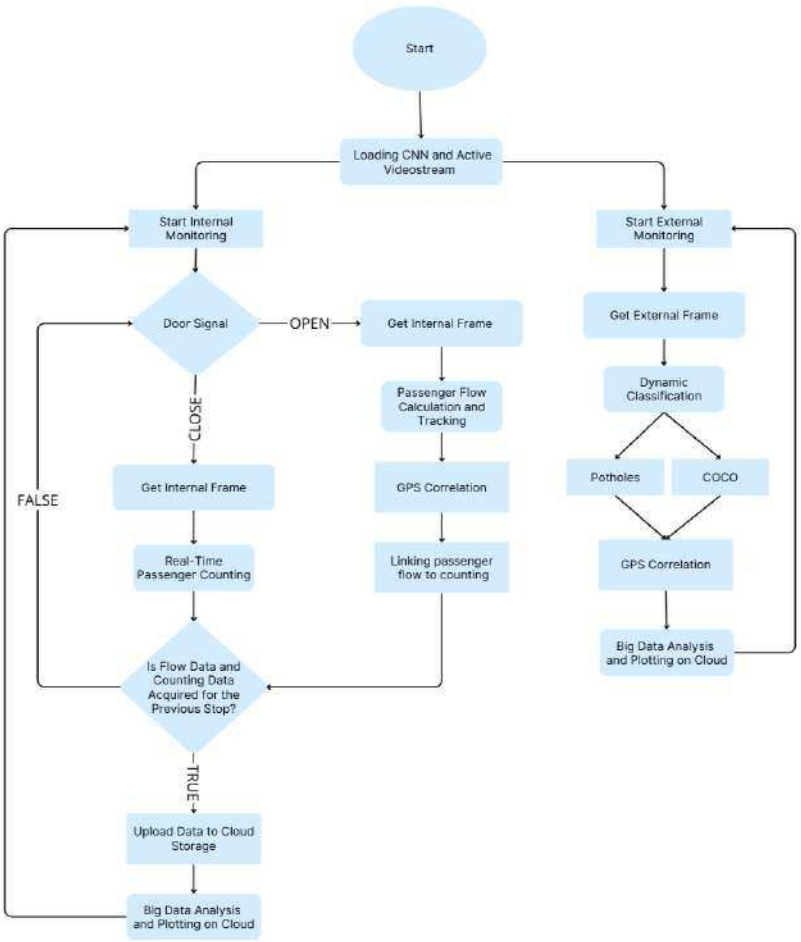


FIGURE 4.8: Flowchart of the proposed algorithmic procedure.

In the external monitoring process, the AI server continuously receives real-time video streams from the vehicle's external PoE cameras through the onboard Ethernet network. These streams are processed to detect and classify a variety of roadside and environmental objects, such as potholes, waste, bicycles, scooters, and pedestrians. This process, referred to as dynamic classification, emphasizes that object recognition and categorization occur while the vehicle is in motion, adapting dynamically to variable lighting conditions and environmental complexity. Through optimized inference and tracking algorithms, the system ensures that each detected object is recognized and geolocated only once, even when captured across consecutive frames. This mechanism minimizes redundant detections and maintains consistent tracking despite continuous vehicle movement. Each classified object is subsequently linked to GPS data, providing accurate spatial references and enabling the generation of geo-referenced urban maps that support infrastructure maintenance, urban safety management, and environmental analysis. Figure 4.8 illustrates the overall system workflow, which integrates both internal and external monitoring procedures into a single, unified framework. The process begins with the loading of CNN models and the activation of video streams on the AI server and clients. From that point, two concurrent processes operate in parallel:

- the internal monitoring pipeline, responsible for passenger flow tracking and counting, triggered by bus door signals;
- the external monitoring pipeline, focused on dynamic object detection and classification.

When the doors open, the internal subsystem activates the passenger flow calculation and tracking module, linking each detected movement with the current GPS position. Once the doors close, the real-time passenger counting process begins, updating occupancy data for the next trip segment. Meanwhile, the external module continues analyzing the environment, detecting and classifying objects in real time using YOLOv5-based models, and correlating them with precise geospatial coordinates. Both internal and external data streams are then transmitted through an industrial router equipped with a dedicated SIM module, ensuring continuous 4G/5G connectivity between the onboard system and the cloud platform. All data, including passenger information, environmental detections, and GPS metadata, are uploaded in real time to the cloud, where they are processed, aggregated, and visualized. The cloud dashboard provides operators with a clear, data-driven overview of passenger occupancy, environmental conditions, and detected infrastructure anomalies across the fleet. Moreover, the platform supports historical data analysis, enabling long-term performance evaluation, trend identification, and the development of predictive insights for operational planning. This integrated architecture establishes a closed-loop information ecosystem, where onboard edge AI and cloud analytics operate synergistically to support intelligent transport management, predictive maintenance, and data-driven service optimization across the fleet. Beyond the software layer, the overall system combines electrical robustness, ensured by the regulated 24 V power distribution, with network resilience and scalability provided by the Ethernet-based PoE infrastructure and secure cellular routing through the industrial router. This integrated design allows synchronized, real-time monitoring of both internal and external phenomena, bridging the gap between vehicle-level sensing and centralized data-driven decision-making. By connecting reliable onboard perception with cloud-level intelligence, the architecture delivers a comprehensive and scalable framework for the future of public transport management.

4.3 Internal And External Monitoring Training

4.3.1 Train Internal Monitoring Model

In this work, the YOLOv5 object detection framework [13,46] is adopted for passenger classification and localization tasks. The selection of YOLOv5 is motivated by its favorable trade-off between detection accuracy, inference speed, and computational efficiency, which makes it particularly suitable for real-time deployment in public transport environments characterized by constrained hardware resources and strict latency requirements. Thanks to its optimized convolutional backbone, anchor-based detection strategy, and efficient multi-scale feature aggregation, YOLOv5 is capable of maintaining high precision–recall performance while ensuring stable inference on embedded GPU platforms. The training dataset was constructed by integrating multiple heterogeneous data sources to maximize variability and robustness. Specifically, publicly available datasets were employed, including Kaggle ($\approx$10,000 images), Roboflow ($\approx$15,000 images), and GitHub repositories ($\approx$5,000 images), and were complemented with a large set of images acquired directly from onboard cameras installed inside the vehicle ($\approx$50,000 images). The inclusion of real operational data is particularly relevant, as it captures domain-specific conditions such as occlusions, perspective distortions, non-uniform illumination, and passenger density variations typical of real-world public transport scenarios. To further enhance dataset diversity and improve generalization capability [59], extensive data augmentation techniques were applied in order to synthetically reproduce challenging visual conditions encountered during daily operations. The applied augmentations are summarized as follows:

1. **Random brightness adjustment** (20,000 images): image brightness was randomly varied within a range of $\pm$20% with respect to the original illumination conditions, simulating different lighting environments inside the vehicle.

2. **Random horizontal flipping** (20,000 images): each image had a 50% probability of being horizontally flipped, accounting for viewpoint symmetry and camera placement variability.

3. **Random rotation** (20,000 images): images were rotated within a range of $\pm$15 degrees to reproduce camera misalignment and vehicle motion effects.

4. **Gaussian noise injection** (20,000 images): zero-mean Gaussian noise with standard deviation $\sigma \in [0.01, 0.05]$ was added and normalized to the image intensity range, emulating sensor noise and compression artifacts.

Subsequently, the Roboflow platform [60] was employed for dataset management, annotation, and preprocessing. A set of preprocessing filters was applied to improve image consistency and feature visibility across different acquisition conditions:

1. **Contrast normalization, sharpening, and denoising** were applied to enhance salient visual features such as body contours, clothing patterns, and facial regions, which are critical for reliable passenger detection and classification.

2. **Image resizing and cropping** ensured spatial uniformity across data sources, enabling stable batch processing during training.

3. **Selective background blurring** was introduced in specific cases to reduce visual clutter and emphasize passenger-related regions within the scene.

The resulting dataset, composed of more than 160,000 labeled images, was divided into training, validation, and test subsets following a 70–20–10 split. All images were manually annotated using bounding boxes to distinguish passengers, empty seats, and partial occlusions. To optimize convergence and leverage transfer learning, YOLOv5 models were initialized using pre-trained COCO weights and subsequently fine-tuned on the passenger-specific dataset by refining the final detection layers.

TABLE 4.1: Comparison of YOLOv5n, YOLOv5s, YOLOv5m, and YOLOv5l model results over 100 epochs.

Model	Precision	Recall	Inference
YOLOv5n	92.5	89.3	0.05
YOLOv5s	94.8	91.6	0.1
YOLOv5m	96.1	93.7	0.4

The training procedure was conducted over 100 epochs, and a comparative evaluation was performed among different YOLOv5 variants, namely YOLOv5n, YOLOv5s, and YOLOv5m. All experiments were executed on a workstation equipped with an AMD Ryzen 9 5900X 12-Core CPU (3.70 GHz), 64 GB RAM, and an NVIDIA GeForce RTX 3090 GPU. Table 4.1 reports a comparative evaluation of three YOLOv5 model variants trained for 100 epochs, highlighting the trade-off between detection accuracy and computational efficiency. The results show a clear and consistent performance progression as model complexity increases from YOLOv5n to YOLOv5m. YOLOv5n, the lightest architecture, achieves the lowest precision (92.5%) and recall (89.3%) while providing the fastest inference time (0.05 s per frame). Although its low latency makes it suitable for extremely resource-constrained scenarios, the reduced recall indicates a higher rate of missed detections, which is particularly critical in passenger counting applications where underestimation directly affects system reliability. YOLOv5s provides a balanced improvement, reaching 94.8% precision and 91.6% recall with a moderate inference time of 0.1 s. This configuration represents a compromise between accuracy and speed; however, the observed recall gap with respect to more complex models suggests that challenging cases such as partial occlusions, overlapping passengers, and variable illumination are not consistently handled. YOLOv5m exhibits the best overall performance, achieving the highest precision (96.1%) and recall (93.7%), at the cost of an increased inference time of 0.4 s. Despite this higher computational demand, the inference latency remains fully compatible with real-time passenger counting requirements in public transport scenarios. The improved recall is particularly relevant, as it indicates a stronger capability to detect passengers under crowded conditions and complex visual configurations, reducing systematic undercounting errors. Overall, the results demonstrate that increasing model capacity yields substantial gains in detection robustness and accuracy, while the corresponding increase in inference time remains acceptable for onboard deployment. Consequently, YOLOv5m was selected as the final model, as it provides the most favourable balance between detection accuracy and computational cost, while maintaining an inference latency fully compatible with real-time passenger counting requirements in operational public transport environments.

4.3.2 Train External Monitoring Model

The external monitoring component of the proposed system is designed to analyze road surface conditions and surrounding traffic elements using vision-based techniques. To address these two complementary objectives, two distinct YOLOv5m models were trained: a specialized model dedicated to pothole detection and a multi-class model trained on selected object categories from the COCO dataset for general road scene understanding. Accurate pothole classification plays a crucial role in enabling timely identification of critical road segments, supporting predictive maintenance strategies and improving traffic safety and flow. For this purpose, a YOLOv5m model was trained exclusively for pothole detection using a custom dataset composed of approximately 13,000 annotated images. The dataset integrates heterogeneous sources, including 3,800 images from Kaggle, 3,400 from Roboflow, 3,600 from GitHub, and 2,200 images acquired directly from vehicle-mounted cameras during real-world operations. The inclusion of onboard camera data is particularly relevant, as it captures realistic conditions such as varying illumination, motion blur, perspective distortions, and partial occlusions. Training was initialized using pre-trained YOLOv5m weights and fine-tuned on the pothole-specific dataset.

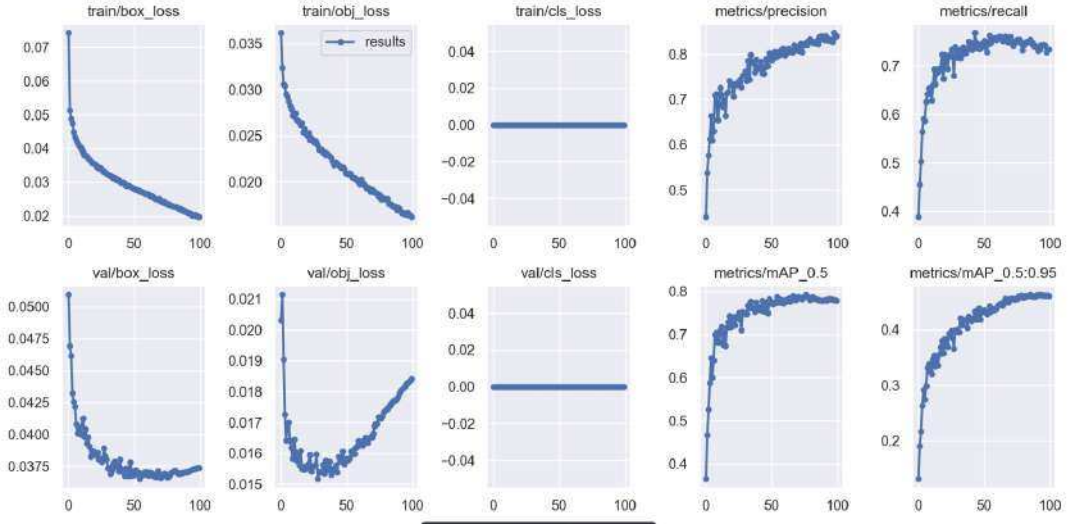


FIGURE 4.9: Training results with potholes dataset.

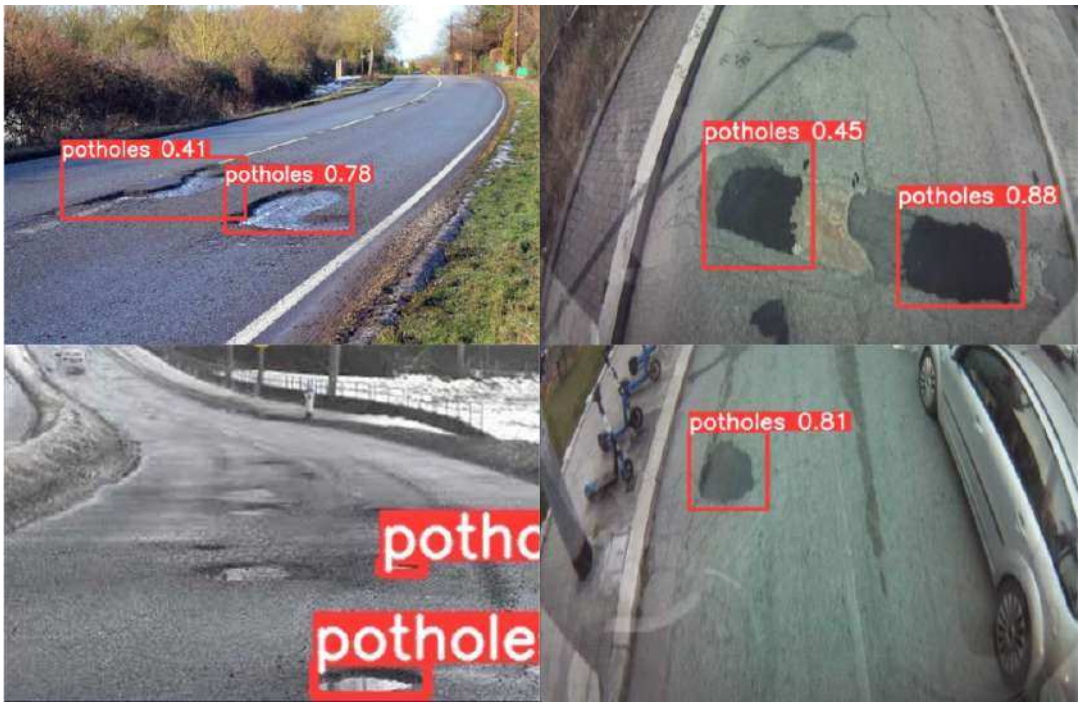


FIGURE 4.10: Potholes detection example.

As shown in Fig. 4.9, the training process exhibits a stable and progressive reduction in localization and classification losses, indicating effective convergence. The resulting model achieves a precision of approximately 0.8 and a recall of 0.7, with an mAP@0.5 close to 0.8, demonstrating robust detection performance. Although minor signs of overfitting are observed in the later training stages, these effects can be mitigated through additional data augmentation or regularization strategies. As illustrated in Fig. 4.10, the trained model is not limited to identifying potholes but is also capable of detecting related road surface anomalies, such as temporary patches and asphalt degradations, providing a more comprehensive assessment of road conditions. However, real-time classification alone is insufficient to reliably estimate pothole occurrences in dynamic scenarios. When the vehicle is in motion, a detected pothole may persist across multiple consecutive frames, potentially leading to duplicate counts.

In parallel, a second YOLOv5m model was trained to perform general external environment monitoring using lateral and front-facing cameras installed on the bus. This model was fine-tuned on selected object classes from the COCO (Common Objects in Context) dataset [61], which is widely adopted for large-scale object detection and scene understanding tasks. The COCO dataset comprises 39,475 images and 357,872 labelled instances across multiple object categories, with a strong emphasis on urban and traffic-related scenes. For this work, a subset of COCO classes relevant to road and traffic analysis was selected, including person, car, bus, truck, bicycle,train, traffic light, and stop sign. The dataset includes a total of 39.475 images and 357,872 labelled objects across multiple classes. Below the image and label counts per class:

- Bicycle: 1,929 images, 7,370 labels
- Car: 6,116 images, 45,451 labels

- Bus: 1,745 images, 6,344 labels
- Person: 23,449 images, 268,029 labels
- Train: 3,073 images, 4,760 labels
- Truck: 1,084 images, 10,384 labels
- Traffic Light: 949 images, 13,476 labels
- Stop Sign: 1,130 images, 2,058 labels

The dataset is heavily dominated by the *person* class, followed by vehicle-related categories, ensuring strong representation of typical urban mobility scenarios. A 75–20–5 split was adopted for training, validation, and testing, respectively. Training

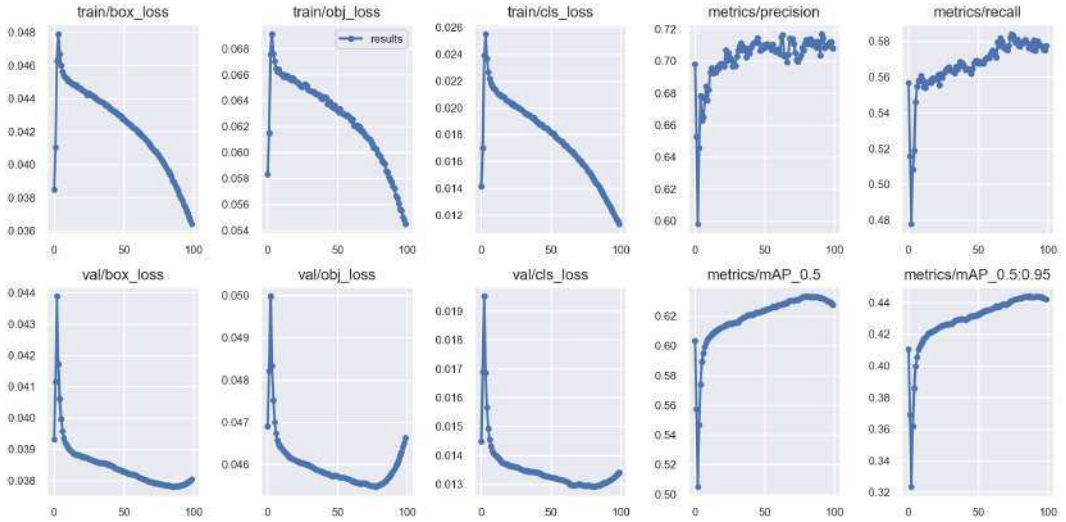


FIGURE 4.11: Train with COCO dataset

curves reported in Fig. 4.11 show a consistent reduction in loss values, confirming effective learning. The model reaches a stabilized precision of approximately 0.7 and a recall of 0.58, indicating reliable detection with some residual missed detections in complex scenes. The achieved mAP@0.5 of 0.62 demonstrates good overall accuracy, while the mAP@0.5:0.95 value of 0.44 suggests potential room for improvement through further fine-tuning or domain-specific data augmentation.

FIGURE 4.12: COCO detection example.

Figure 4.12 presents examples of inference results obtained from frames captured in real traffic conditions in the city of Bari, Italy. Each detected object is associated with its corresponding GPS coordinates, enabling spatio-temporal mapping of vehicles, pedestrians, and traffic infrastructure elements. Together, the two trained models provide a comprehensive external monitoring solution capable of jointly assessing road surface quality and surrounding traffic dynamics in real time.

4.4 Internal Monitoring Process

4.4.1 2D-to-3D Object Conversion

The YOLOv5 framework provides, for each detected object, a set of 2D bounding boxes associated with their corresponding semantic categories. However, in the context of passenger monitoring inside public transport vehicles, a simple count of detected bounding boxes is insufficient to guarantee accurate and reliable results. This limitation becomes particularly evident when multiple cameras with partially overlapping fields of view are deployed, as the same passenger may be simultaneously detected by different cameras, leading to systematic double counting. To address this issue, it is necessary to introduce an absolute spatial reference system capable of defining the effective coverage area of each camera and explicitly identifying overlapping regions. Within this framework, each detection must be uniquely associated with a position in a shared three-dimensional coordinate space.

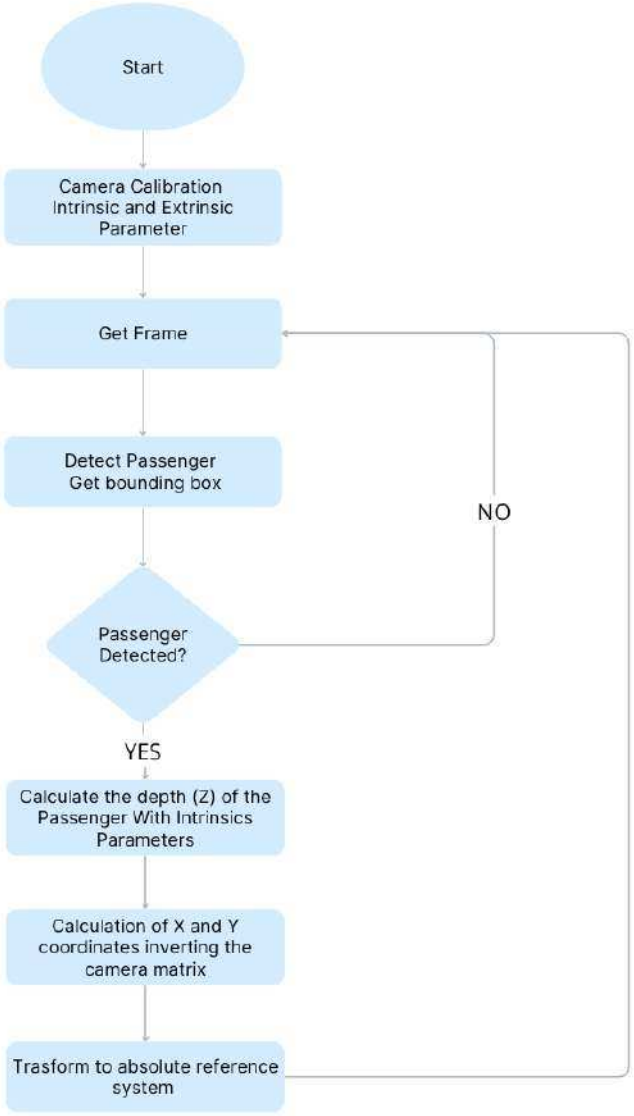


FIGURE 4.13: 2D to 3D object conversion.

For this reason, a 2D-to-3D conversion mechanism was developed to project image-based detections into an absolute 3D reference frame, as illustrated in Fig. 4.13. Building on the formulation proposed in [62], a similar geometric approach was adopted, although the underlying reasoning was inverted. Instead of estimating object dimensions from a known distance, the distance of the detected object from the camera is inferred from its known physical size. This assumption is particularly suitable for passenger monitoring, where approximate anthropometric dimensions can be reasonably constrained. The relationship between a point in the 3D world and its projection onto the 2D image plane is described by the standard pinhole camera

model:

$$\begin{pmatrix} u \\ v \\ w \end{pmatrix} = \underbrace{\begin{pmatrix} \frac{f}{\rho_u} & 0 & c_x & 0 \\ 0 & \frac{f}{\rho_v} & c_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}}_{\text{Intrinsic Camera Matrix}} \cdot \underbrace{\begin{pmatrix} R_{3\times3} & t_{3\times1} \\ 0_{1\times3} & 1 \end{pmatrix}}_{\text{Extrinsic Camera Matrix}} \cdot \begin{pmatrix} X_W \\ Y_W \\ Z_W \\ 1 \end{pmatrix}, \tag{4.1}$$

where (u, v, w) represents the homogeneous coordinates of a point in the 2D image plane, obtained through the projection of its 3D world coordinates. The intrinsic camera matrix encodes the optical properties of the camera, including the focal length f, the pixel scaling factors ρ_u and ρ_v, and the principal point coordinates (c_x, c_y). The extrinsic camera matrix describes the rigid transformation between the world reference frame and the camera reference frame through a rotation matrix $R_{3\times3}$ and a translation vector $t_{3\times1}$. Finally, (X_W, Y_W, Z_W) denotes the coordinates of the object in the absolute 3D reference system. Given the 2D image coordinates extracted from the YOLOv5 bounding box and assuming a known physical dimension of the detected object, it is possible to invert Eq. (4.1) and recover the corresponding 3D position in the world reference frame. This operation ensures spatial consistency across multiple cameras and allows all detections to be expressed within a common coordinate system. The intrinsic camera parameters were estimated using a chequerboard calibration pattern and multiple image acquisitions processed with OpenCV in Python, following Zhang's camera calibration method [43], as previously detailed in Chapter 2. Once the 3D spatial coordinates of detected passengers are available, the effective working area of each camera can be uniquely defined. This enables the explicit delineation of camera coverage regions and prevents double counting in areas where fields of view overlap.

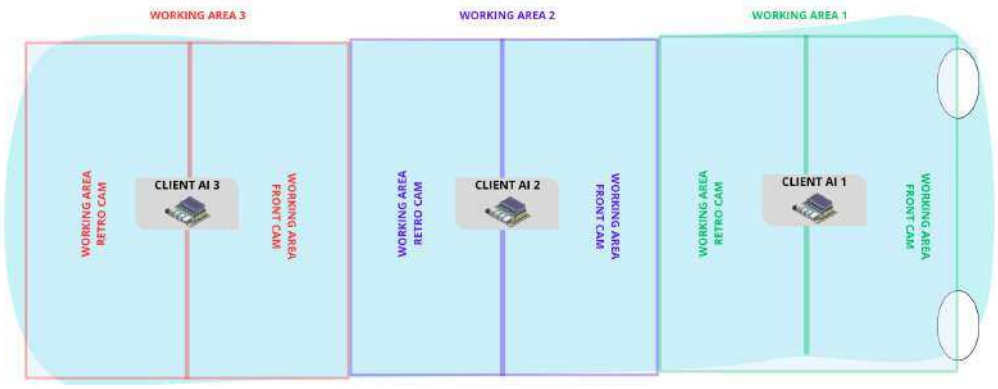


FIGURE 4.14: Working areas of the AI client cameras.

Figure 4.14 illustrates the working areas associated with the cameras in the considered setup, where each AI Client is equipped with two cameras: one oriented toward the front of the vehicle and the other toward the rear. By mapping 2D detections into the 3D space, each passenger is counted exclusively by the camera whose working area contains the corresponding 3D position.

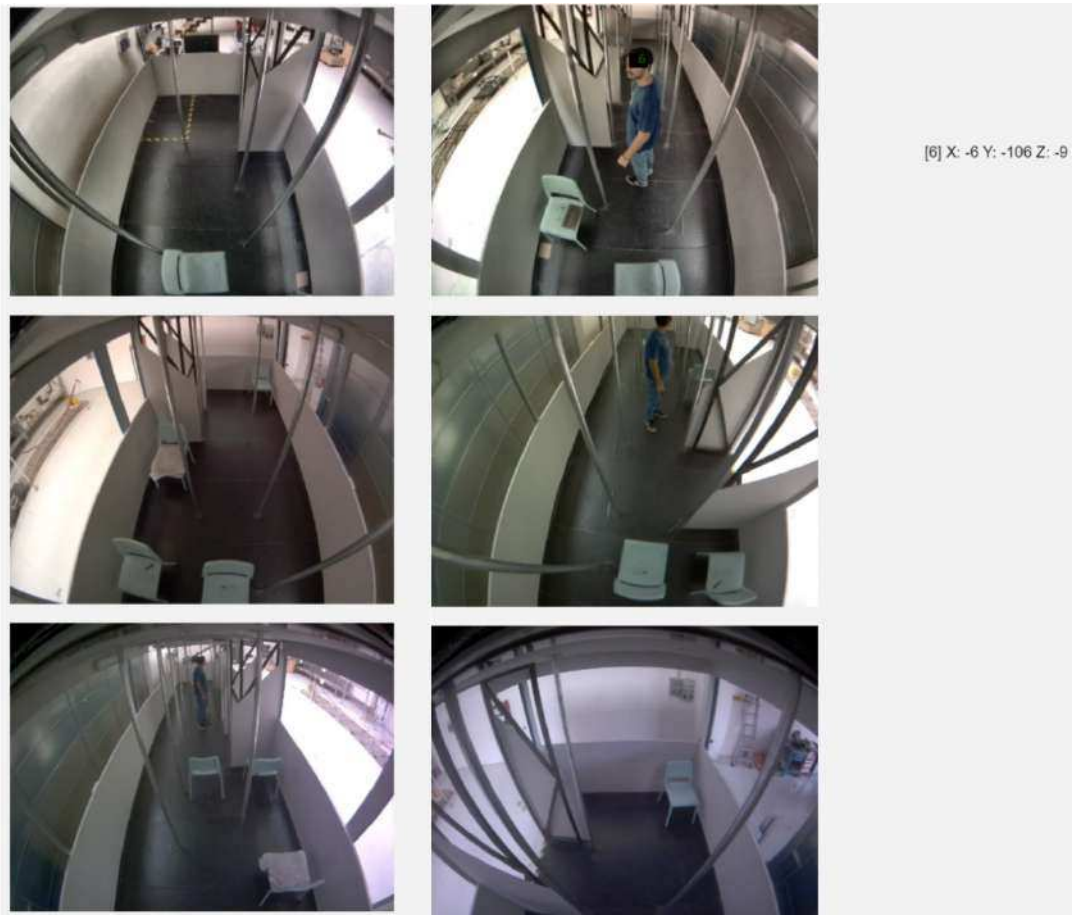


FIGURE 4.15: Example of managing overlaps between cameras.

An illustrative example is shown in Fig. 4.15, obtained from a 1:1 laboratory-scale emulation of the real bus environment. In this scenario, a subject is simultaneously detected by three cameras: the rear-facing camera of client 1, the front-facing camera of client 2, and the front-facing camera of client 3. Nevertheless, the subject is counted only by the rear-facing camera of client 1, as its 3D coordinates fall within the defined working area of that camera. The lateral column in Fig. 4.15 reports the subject's position in absolute (X, Y, Z) coordinates. This 2D-to-3D conversion strategy ensures robust and consistent passenger counting, even in complex multi-camera configurations with overlapping fields of view, and represents a fundamental component of the proposed onboard monitoring system.

4.4.2 Passenger Flow Calculation and Tracking with Deep SORT

To estimate passenger inflow and outflow, position tracking of the detected people is essential. In this work, we employ the Deep SORT (Simple Online and Realtime Tracking with a Deep Association Metric) algorithm [63, 64] for passenger flow estimation. Deep SORT extends the SORT algorithm by integrating motion and appearance-based metrics, improving identity consistency even in crowded or occluded scenes. First, YOLOv5 detects individuals in a sequence of video frames, generating bounding boxes around each detected person. Then, Deep SORT models

each tracked object using an 8-dimensional state vector To maintain tracking stability, a Kalman filter is employed to predict object states, estimating position, velocity, and acceleration. This predictive capability allows the system to maintain smooth tracking by compensating for noise and handling missed detections effectively. Finally, the Deep SORT algorithm integrates these components, performing multi-object tracking by combining motion and appearance information, ensuring accurate and consistent identity assignment throughout the video sequence. Deep SORT (Simple Online and Realtime Tracking with a Deep Association Metric) extends the SORT algorithm by integrating both motion-based and appearance-based metrics to improve tracking robustness. This enhancement ensures that objects detected in consecutive frames are consistently assigned the same tracking ID, even in occluded or crowded environments. The algorithm builds upon SORT's Kalman filter-based motion modelling but introduces a deep appearance descriptor to enhance identity preservation. This combination significantly reduces identity switches and ensures reliable object tracking across frames. Below, we provide a step-by-step breakdown of how Deep SORT operates, incorporating relevant mathematical formulations [64]. Deep SORT models each tracked object using an 8-dimensional state vector :

$$\mathbf{x} = \begin{bmatrix} x & y & a & h & \dot{x} & \dot{y} & \dot{a} & \dot{h} \end{bmatrix}^T \tag{4.2}$$

where (x, y) represent the center coordinates of the bounding box, a denotes the aspect ratio, and h represents the height of the bounding box. This state allows the system to predict future positions while maintaining shape consistency. To ensure continuity in tracking, Deep SORT employs a Kalman filter that predicts the next state of each object based on a constant velocity motion model:

$$x_{k+1} = Fx_k + w_k \tag{4.3}$$

where F is the state transition matrix and $w_k \sim \mathcal{N}(0, Q)$ is the process noise modeled as a Gaussian distribution with covariance Q that was set empirically by trial-and-error. The matrix F is given by:

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & \Delta t & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & \Delta t & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.4}$$

where Δt is the time step between consecutive frames. To maintain object identities across frames, Deep SORT matches new detections with existing tracks using two complementary metrics:

- Mahalanobis distance to evaluate motion consistency by comparing the Kalman filter's predictions with new detections;
- Cosine distance to preserve identity by matching visual features, reducing errors and reconnecting lost tracks after occlusions.

The two association metrics are combined into a cost function:

$$c_{i,j} = \lambda d_M(i, j) + (1 - \lambda) d_C(i, j) \tag{4.5}$$

In the cost function, i and j refer to the indices of an existing track and a new detection, respectively. The terms $d_M(i,j)$ and $d_C(i,j)$ denote, respectively, the Mahalanobis distance and the cosine distance between them. The parameter λ balances the contribution of these two metrics, thus combining spatial prediction and visual features. This cost matrix is then optimized using the Hungarian algorithm, which efficiently finds the best global assignment of detections to tracks.

Finally, to enhance robustness, Deep SORT incorporates a track retention and deletion mechanism. When an object temporarily disappears, its track is retained to handle short occlusions. A new detection must be confirmed across three consecutive frames before being assigned a unique ID, reducing false positives. Tracks that remain unmatched for an extended period are deleted to avoid outdated data. After the detection-to-track assignment, new detections are added to active tracks, and unmatched detections initiate new track hypotheses, while outdated tracks are removed. By applying the Deep SORT tracking approach in this study and setting the unmatched period to 2 seconds, the tracking process was implemented following the algorithm outlined in Fig. 4.16.

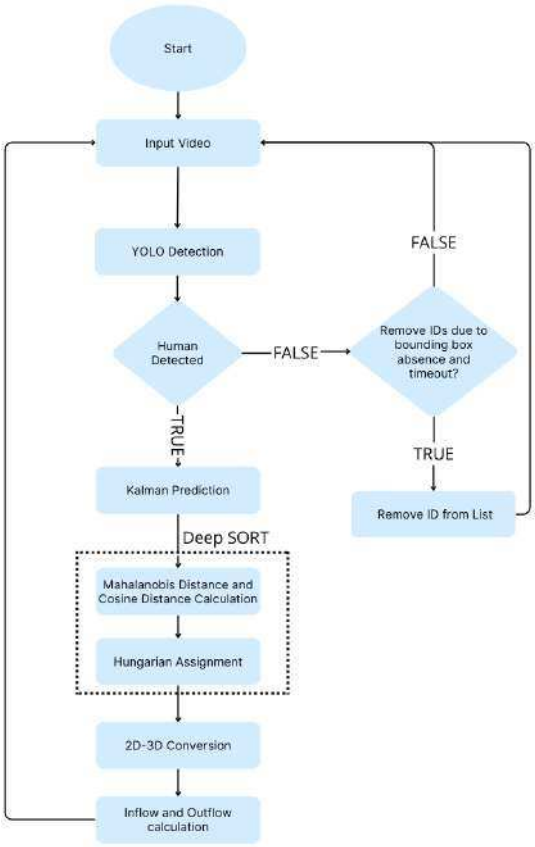


FIGURE 4.16: Passenger flow calculation using Deep SORT.

The bounding boxes generated by YOLO and tracked by Deep SORT algorithm are converted from 2D to 3D coordinates. Then, to calculate the inflow and outflow

through a passageway, thresholds that identify the passage of individuals through the area are defined. These thresholds allow for distinguishing between inward and outward movements, as illustrated in Fig. 4.17. Thus, using the position parameters (X, Y) associated with the passenger, the direction of their movement through the passageway can be determined. If the passenger crosses the first threshold before the second one, it is recorded as an exit. Conversely, if the second threshold is crossed before the first one, it is recorded as an entry.

FIGURE 4.17: Inward and outward thresholds.

4.5 External Monitoring Process

4.5.1 Flow Detection

The external monitoring process adopts a flow-based detection, filtering, and counting strategy inspired by the passenger inflow and outflow estimation methodology described in the previous sections. Rather than relying on frame-by-frame object counts, which would inevitably lead to multiple detections of the same physical object, the proposed approach explicitly exploits the dynamic nature of the system to ensure consistent and non-redundant counting. Two distinct flow detection strategies are implemented, depending on the type of monitored object. For road surface anomalies, such as potholes, a threshold-based flow detection mechanism is employed. As the vehicle moves forward, detected potholes remain visible across multiple consecutive frames.

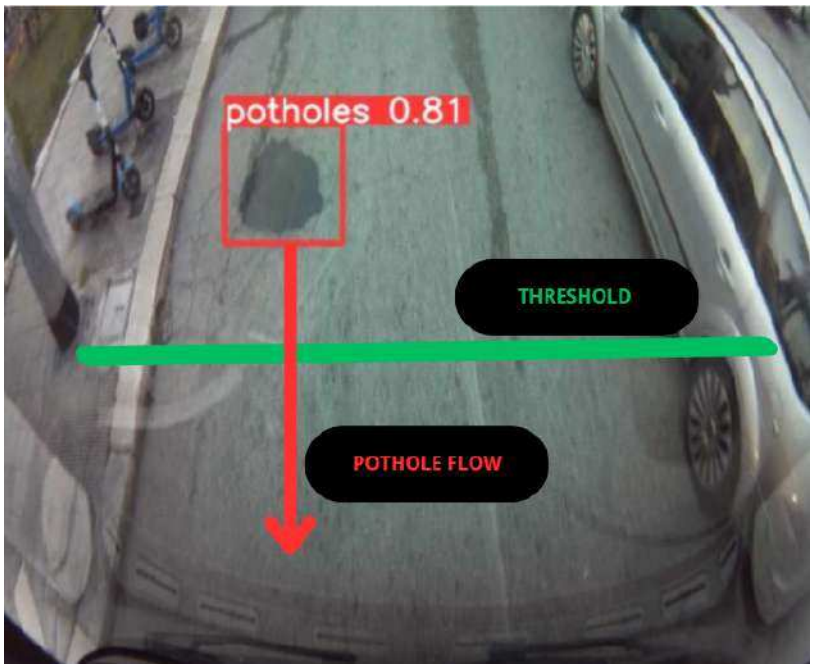


FIGURE 4.18: Pothole dynamic detection example.

To avoid duplicate counts, each pothole is tracked and counted only when it crosses a predefined virtual threshold in the image plane, as illustrated in 4.18. At the moment the threshold is crossed, the pothole's spatial position is recorded and the object is counted once, ensuring a unique association between physical road defects and recorded events. For all other external objects detected by the trained models, including selected COCO classes, a complementary ID-based flow detection strategy is applied. Given that these objects are observed from a single, fixed camera perspective, each detected instance is assigned a persistent tracking identifier through the Deep SORT framework. As long as an object remains within the camera's field of view, it retains the same tracking ID and is therefore considered an already counted instance. The counting event is triggered only when the tracked object disappears from the scene, indicating that it has fully exited the monitored area. This disappearance-based counting strategy prevents multiple counts of the same object across consecutive frames, as the presence of a persistent ID explicitly signals that the object has already been registered. Only when the object leaves the field of view and its associated track is terminated does the system increment the corresponding class counter. This logic is applied uniformly across all COCO-based object categories, ensuring consistent and reliable counting behaviour. Overall, the proposed flow detection mechanism leverages the intrinsic temporal continuity of the external monitoring scenario. By exploiting object persistence, motion dynamics, and tracking identities, the system avoids redundant counts and achieves accurate, real-time estimation of external events, even in highly dynamic operating conditions.

4.6 Data Collection on Cloud

The data collection architecture adopted in this work is based on a dual-layer structure, consisting of a local storage system deployed on the AI SERVER installed onboard the vehicle and a centralized cloud-based infrastructure. This design enables both low-latency local data availability and scalable, aggregated access to information at fleet level, which is particularly relevant when multiple vehicles are equipped with internal and external monitoring systems. The cloud infrastructure is built around a dedicated server running a Node.js-based software stack, responsible for both frontend visualization and backend data ingestion and management. The backend layer exposes a set of dedicated RESTful APIs that allow AI SERVER units to transmit collected data securely to the cloud. Each AI SERVER communicates with the cloud through authenticated HTTPS POST requests, using identification and security tokens to ensure data integrity and access control. Passenger counting data are transmitted to the cloud every time a new counting event is generated. These data are stored in a relational MySQL database, organized into structured tables designed to support efficient querying and aggregation. For example, passenger flow information is recorded in a dedicated table (e.g., *tablestat*), which includes the number of detected passengers, the associated bus stop identifier, GPS coordinates recorded at door opening, trip identifier, route identifier, and the timestamp corresponding to the data generation event. A separate set of cloud endpoints is dedicated to road surface monitoring, specifically pothole detection. For each detected pothole, the system stores georeferenced information obtained through the integration with the GNSS antenna of the onboard Teltonika RUT955 router. In addition to spatial coordinates, potholes are classified according to their estimated size into predefined categories (small, medium-small, medium, medium-large, and large), which are stored as indexed attributes within the database to facilitate statistical analysis and maintenance prioritization. Furthermore, external environment monitoring data related to COCO-based object detection are transmitted through additional dedicated routes. For these detections, the cloud infrastructure stores both the object class and the corresponding geographic location, including the road segment or street where the object was detected. This enables spatial analysis of traffic elements and environmental context across the monitored area. All data ingestion, validation, and storage processes are orchestrated by the Node.js backend server. Node.js was selected due to its event-driven, non-blocking I/O architecture, which allows efficient handling of a large number of concurrent requests generated by potentially thousands of distributed AI SERVER clients. This architectural choice ensures high scalability, robustness, and ease of horizontal expansion, making the proposed cloud infrastructure suitable for large-scale deployments involving entire fleets of public transport vehicles.

Chapter 5

RESULTS

The developed system was deployed on 50 buses operated by AMTAB, the public transport company of Bari, Italy. These vehicles serve 30 different routes, each consisting of 10 to 40 stops per line. However, over the course of a full working day, a single bus may perform between 50 and 100 total stops, as it completes multiple runs along different routes. Tests were conducted with the system running for a total of 2 consecutive months.

5.1 Passenger Counting Validation

To quantitatively assess the performance of the proposed passenger counting system, which integrates real-time occupancy estimation with passenger flow tracking, an extensive experimental evaluation was conducted on a public transport bus operating along a route composed of 58 stops over an entire working day. This experimental setup was designed to reflect realistic operational conditions, including varying passenger densities, boarding and alighting dynamics, and different levels of visual complexity inside the vehicle.

Ground truth data were obtained through remote validation tests carried out while the bus was in service. Passengers were manually counted using live video streams, and the observed occupancy values were systematically associated with the corresponding stop identifiers. These manually collected counts were then compared against the values estimated by the proposed system, allowing a direct and objective evaluation of counting accuracy at each segment of the route.

For each route segment i, defined as the interval between two consecutive stops, the accuracy is computed as:

$$\text{Accuracy}_i = \left(1 - \frac{|N_{Ri} - N_{Ci}|}{N_{Ri}}\right) \cdot 100, \tag{5.1}$$

where N_{Ri} denotes the number of passengers manually counted (reference value), and N_{Ci} represents the number of passengers estimated by the system. This metric captures the relative deviation between the estimated and actual occupancy, providing a normalized measure of performance for each segment.

To further analyze system behavior, accuracy was evaluated at different temporal intervals following door closure at each stop. This analysis aims to quantify how rapidly the estimated occupancy converges toward the true passenger count as the internal scene stabilizes after boarding and alighting events. Specifically, passenger counts were sampled at 5 seconds, 15 seconds, and immediately before the next door opening, corresponding to the final stabilized estimation.

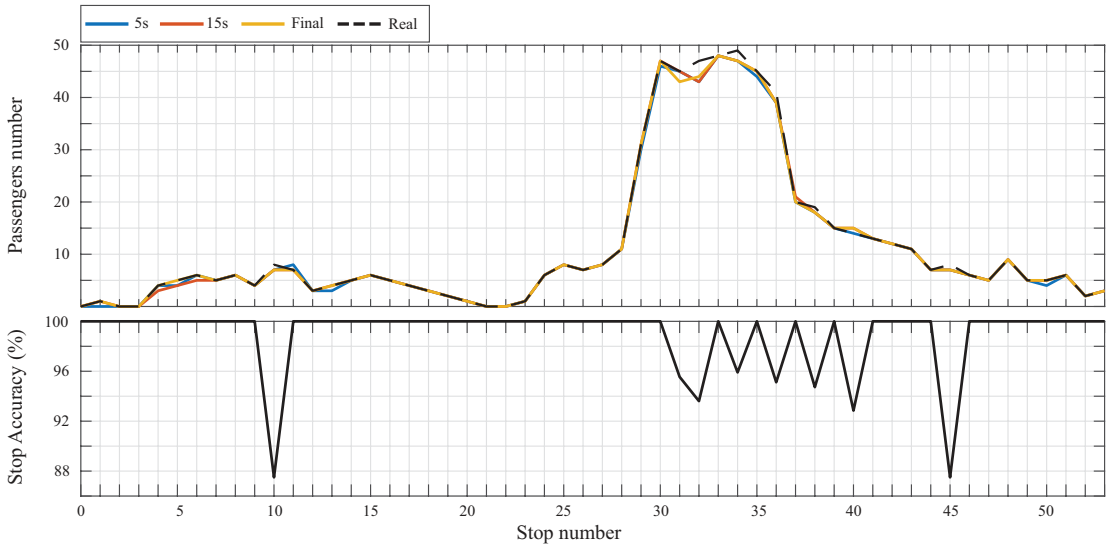


FIGURE 5.1: Performance of the proposed passenger counting approach on a test route with 53 stops.

Figure 5.1 illustrates the detected passenger counts at the selected temporal instants, alongside the corresponding ground truth values for each route segment. The figure also reports the per-segment accuracy computed according to Eq. (5.1). The results demonstrate that while early estimations at 5 seconds may exhibit small discrepancies due to highly dynamic conditions, subsequent measurements converge rapidly toward the actual occupancy. This behavior confirms the effectiveness of the proposed approach in stabilizing the passenger count within a short time window. To summarize performance over the entire route, an overall accuracy metric was computed using a weighted average formulation:

$$\text{Accuracy} = \frac{\sum_{i=1}^{n} \left(1 - \frac{|N_{Ri} - N_{Ci}|}{N_{Ri}}\right) \cdot 100 \cdot N_{Ri}}{\sum N_{Ri}}, \tag{5.2}$$

where n is the total number of route segments. The weighting factor N_{Ri} ensures that segments with higher passenger occupancy contribute proportionally more to the final accuracy, preventing extreme cases from disproportionately affecting the results. For example, detecting a single passenger when no passengers are present would otherwise result in a 100% relative error. Using this formulation, the proposed system achieves an overall accuracy of 94.04% at 5 seconds, 96.71% at 15 seconds, 98.1% at 30 seconds, and stabilizes at 98.15%. The initial performance drop observed at 5 seconds can be attributed to highly dynamic conditions inside the vehicle, as passengers are still boarding, moving, or searching for seats. These conditions increase occlusions and visual clutter, occasionally leading YOLOv5 to miss or double-count individuals. As the scene stabilizes, detections become increasingly consistent and precise, with accuracy stabilizing around 30 seconds, further confirming the robustness of the proposed method under visually stable conditions. It is important to note that the residual error of approximately 2% is non-cumulative. Miscounts occurring at a specific stop do not propagate to subsequent segments, as the system continuously re-estimates the absolute occupancy. This behavior contrasts with classical flow-based methods, in which early counting errors accumulate over time and may

significantly distort the final passenger count [1, 29]. Figure 5.1 also highlights the advantage of integrating real-time counting with flow-based estimation. To explicitly evaluate this aspect, the proposed system was compared against a flow-only variant obtained by disabling real-time passenger counting. This comparison was conducted on a test route composed of 30 stops.

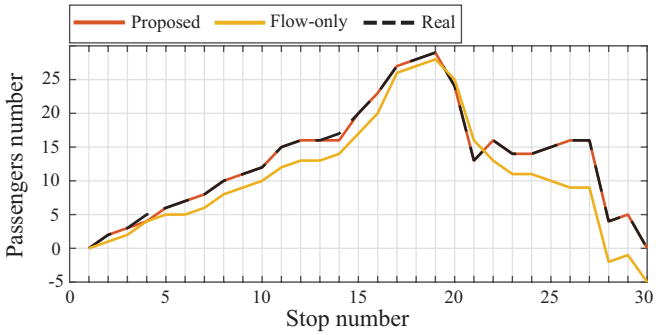


FIGURE 5.2: Comparison between the proposed method and the flow-only approach.

As shown in Fig. 5.2, the flow-only approach diverges significantly from both the proposed method and the ground truth passenger counts. Small errors at individual stops accumulate over time, in some cases leading to unrealistic outcomes such as negative passenger counts. In contrast, the integrated approach effectively constrains local errors, preventing drift and preserving global accuracy throughout the route. This improvement is largely attributable to the exploitation of internal video streams, which enable the system to dynamically refine the passenger count during each stop. To the best of our knowledge, no prior work in the literature has leveraged internal visual perception in this manner for continuous passenger counting, making the proposed framework particularly well suited for real-world deployment. Unlike traditional flow-based systems, which often require periodic resets to mitigate error accumulation, the proposed method maintains accuracy without the need for manual correction or end-of-trip recalibration.

5.2 Big Data Analysis and Plotting on Cloud

Having established the validity and robustness of the proposed monitoring and counting methodologies, the focus now shifts to the analysis and exploitation of the data collected by the centralized system. The availability of reliable, high-frequency data streams generated by both internal and external monitoring modules enables advanced data-driven analysis of public transport operations at both vehicle and fleet levels. All data produced by the onboard AI SERVER units are transmitted to a cloud-based infrastructure, whose core components are a Node.js backend and a relational MySQL database, as described in the previous section. This centralized architecture provides a unified and scalable repository for heterogeneous data, including passenger flow information, vehicle occupancy, road surface anomalies, and external environmental observations. Within this context, cloud-based data analysis

and visualization play a key role in transforming raw measurements into actionable insights. To this end, this thesis presents a set of dedicated dashboards specifically designed for fleet management, operational monitoring, and large-scale data analysis. These dashboards enable real-time and historical inspection of collected data, supporting both descriptive and exploratory analytics across multiple vehicles and routes. In the remainder of this section, several representative analysis scenarios are discussed, illustrating how the collected data can be aggregated, queried, and visualized to extract meaningful patterns related to passenger demand, service efficiency, infrastructure conditions, and environmental context. The presented analyses demonstrate the potential of the proposed cloud-based framework as a comprehensive decision-support tool for public transport operators.

5.2.1 Real Time Fleet Analysis

The proposed cloud-based framework enables advanced real-time analysis and supervision of the entire public transport fleet through a dedicated monitoring dashboard. This module represents the operational interface between the centralized data collection infrastructure and human operators, providing continuous situational awareness of vehicle activity, passenger occupancy, and onboard system status. An overview of the real-time fleet analysis dashboard is shown in Fig. 5.3.

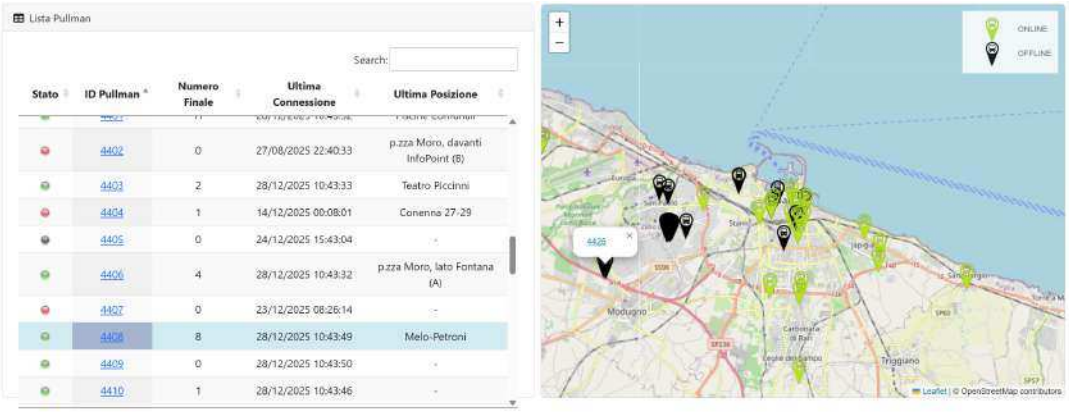


FIGURE 5.3: Dashboard for real time Fleet Analysis

The dashboard is structured around two complementary visualization components. On the left side, a dynamically updated table reports the list of all transport vehicles currently connected to the cloud platform. For each vehicle, the table displays a set of key attributes, including the vehicle identifier, current online/offline status, real-time passenger count estimated by the onboard AI system, timestamp of the last successful communication with the cloud, and the last recorded geographic position. This information allows operators to immediately assess fleet availability, identify inactive or disconnected vehicles, and monitor passenger load distribution across the fleet. The inclusion of the real-time passenger count directly within the fleet list provides immediate insight into vehicle occupancy levels without the need for additional queries or post-processing. This feature is particularly relevant for operational decision-making, such as detecting overcrowded vehicles, identifying

underutilized routes, or dynamically adapting service frequency based on demand patterns. On the right side of the dashboard, a geospatial visualization component presents the real-time positions of the vehicles on an interactive map. Each bus is represented by a marker positioned at its current GPS coordinates, enabling intuitive spatial interpretation of fleet distribution and movement within the service area. Figure 5.3 illustrates this functionality for the urban area of Bari, where the combined visualization of multiple vehicles allows operators to correlate spatial positioning with operational metrics such as occupancy and connectivity. The integration of tabular and map-based representations significantly enhances situational awareness, as it enables rapid cross-referencing between vehicle-specific data and geographic context. This dual-view approach supports both micro-level analysis of individual vehicles and macro-level assessment of overall fleet behaviour. By selecting a specific vehicle from the fleet list, the dashboard transitions to a detailed diagnostic view, as illustrated in Fig. 5.4.

FIGURE 5.4: Dashboard for real time Fleet Diagnostic

This secondary interface provides an in-depth overview of the operational status of the onboard monitoring infrastructure associated with the selected vehicle. In particular, it reports the status of the smart units and cameras installed on board, including network reachability (ping status), camera availability, and recording activity. This diagnostic capability enables rapid identification of hardware or connectivity issues, such as unreachable devices, camera malfunctions, or recording interruptions. Since all status indicators are updated in real time, operators can promptly detect anomalies and initiate maintenance or troubleshooting procedures without physical access to the vehicle. A particularly important feature of the dashboard is the presence of the “last connection” timestamp for each vehicle. This field allows operators to immediately identify vehicles that may have stopped circulating, experienced prolonged connectivity issues, or encountered unexpected shutdowns. In large-scale deployments involving dozens or hundreds of vehicles, this automated visibility is essential to ensure service continuity and to minimize operational downtime. Overall, the real-time fleet analysis dashboard constitutes a powerful and scalable tool for remote fleet supervision. By combining real-time passenger occupancy estimation, continuous vehicle localization, and detailed onboard device diagnostics, the system provides a holistic view of fleet operations. The event-driven and cloud-native nature of the underlying infrastructure ensures that the dashboard remains responsive even when handling data streams from a large number of vehicles. The clarity, transparency, and immediacy of the presented information support timely and informed decision-making by transport operators, enhancing operational

efficiency, service reliability, and passenger experience. As such, the proposed dashboard represents a key component of the overall intelligent public transport monitoring framework.

5.2.2 Analysis by Bus

Beyond fleet-level supervision, the proposed cloud platform enables detailed analytics at the granularity of a single vehicle. Through a dedicated page equipped with a dropdown selector, an operator can select a specific bus and query the centralized database for all records generated within a configurable observation window (here: 01/12/2025–28/12/2025). This design supports targeted inspection of vehicle-specific behavior and provides a controlled way to isolate operational phenomena that can be hidden in aggregated fleet statistics.

The per-vehicle analysis module fulfills a dual role. First, it acts as a diagnostic tool for the onboard sensing and communication infrastructure. Since the AI SERVER periodically transmits passenger counting, localization, and (when available) contextual information to the cloud, the presence (or absence) of data within a selected time interval can be used as an implicit health indicator. Extended gaps in the time series, unusually sparse records, or abrupt discontinuities may be symptomatic of camera outages, local computation failures, or connectivity degradation. In operational scenarios, this data-driven diagnostic layer is particularly valuable because it enables early detection of issues without physical access to the vehicle, thus supporting proactive maintenance scheduling and minimizing downtime. Second, the same interface provides a quantitative view of operational efficiency and usage patterns. By selecting medium- to long-range horizons (e.g., weekly or monthly), the platform makes it possible to observe recurrent demand patterns, load dynamics across service hours, and systematic peak windows.

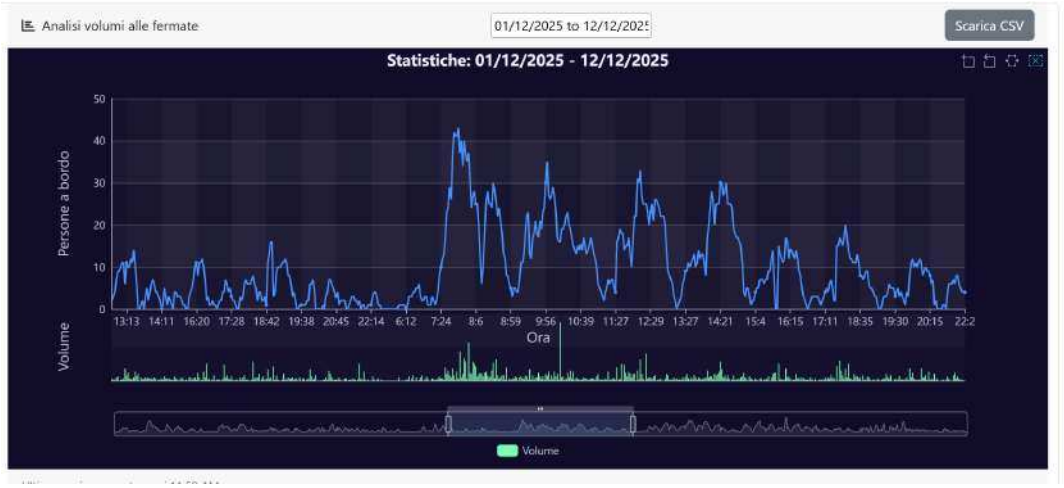


FIGURE 5.5: People counter time-based visualization.

Figure 5.5 shows the time-domain visualization adopted for this purpose, where

the information is displayed through two coordinated layers that capture complementary aspects of passenger activity. The primary layer (blue curve in Fig. 5.5) represents the estimated onboard occupancy as a function of time, i.e., the occupancy signal $N(t)$. This signal describes the internal load state of the vehicle and is driven by the net balance between boarding and alighting events occurring at stops. Sustained high values of $N(t)$ indicate prolonged periods of high utilization, whereas repeated oscillations or cyclic peaks can be associated with recurrent daily patterns (e.g., commuting peaks, school-related travel, or demand concentrated in specific time bands).

The secondary layer (green histogram in Fig. 5.5) represents the passenger exchange intensity, defined as the total number of passenger movements within the considered temporal discretization. Formally, the flow signal can be expressed as:

$$F(t) = B(t) + A(t), \tag{5.3}$$

where $B(t)$ denotes boardings and $A(t)$ denotes alightings. Notably, $F(t)$ is distinct from occupancy because it measures *activity* (exchange) rather than *state* (load). In discrete form, the two signals are linked through the occupancy update equation:

$$N_k = N_{k-1} + B_k - A_k, \tag{5.4}$$

which highlights that large spikes in $F(t)$ may correspond either to high-turnover stops (large exchange but limited net change, i.e., $B_k \approx A_k$) or to unbalanced events (dominant boarding or dominant alighting, producing step-like variations in $N(t)$). Therefore, the combined visualization provides a practical mechanism to distinguish *transit-like* intervals from *high-exchange* nodes, supporting operational decisions such as demand-driven scheduling, vehicle allocation, and the identification of high-impact stops.

To extract robust vehicle-level KPIs from asynchronous event logs, time-weighted metrics are computed from the ordered timestamp sequence $\{t_k\}$ and associated occupancies $\{N_k\}$. Since the dataset may contain long gaps due to off-service periods or communication interruptions, integration is performed by excluding intervals larger than a threshold τ (here, $\tau = 1800$ s). Let $\Delta t_k = t_k - t_{k-1}$. The effective service time is:

$$T_{\text{serv}} = \sum_k \Delta t_k \cdot \mathbb{I}\left(\Delta t_k \leq \tau\right), \tag{5.5}$$

where $\mathbb{I}(\cdot)$ is the indicator function. Passenger-hours quantify the cumulative onboard load over time and are computed via trapezoidal integration:

$$PH = \frac{1}{3600} \sum_k \frac{N_k + N_{k-1}}{2} \Delta t_k \cdot \mathbb{I}\left(\Delta t_k \leq \tau\right), \tag{5.6}$$

leading to the time-weighted mean occupancy:

$$\bar{N} = \frac{PH}{T_{\text{serv}}/3600}. \tag{5.7}$$

To summarize passenger activity, the total exchanges are computed as:

$$F_{\text{tot}} = \sum_k (B_k + A_k), \tag{5.8}$$

TABLE 5.1: Data coverage and load-related KPIs computed for the selected vehicle over 01/12/2025–28/12/2025.

KPI	**Value**
Days covered	28
Total records (events)	8,500
Effective service time T_{serv}	458.10 h
Average daily service time $T_{\text{serv}}/28$	16.36 h/day
Passenger-hours PH	2,872.73 pax·h
Average daily passenger-hours $PH/28$	102.60 pax·h/day
Time-weighted mean occupancy $\bar{N}$	6.27 pax
Peak occupancy $\max(N)$	52 pax
Occupancy 95th percentile	27 pax
Peak-to-mean ratio $\max(N)/\bar{N}$	8.29

and the corresponding flow intensity (exchange rate) is:

$$I_F = \frac{F_{\text{tot}}}{T_{\text{serv}}/3600}. \tag{5.9}$$

Moreover, to quantify high-load operation, the fraction of service time above an occupancy threshold θ is defined as:

$$S_{\geq\theta} = \frac{\sum_k \Delta t_k \cdot \mathbb{I}\left(N_{k-1} \geq \theta\right) \cdot \mathbb{I}\left(\Delta t_k \leq \tau\right)}{T_{\text{serv}}}. \tag{5.10}$$

Finally, to measure the spatial concentration of demand, a stop-level exchange score is defined as:

$$E_s = \sum_{k \in s} (B_k + A_k), \tag{5.11}$$

and the concentration of the top-m busiest stops as:

$$C_m = \frac{\sum_{s \in \text{Top}m} E_s}{\sum_s E_s}. \tag{5.12}$$

Tables 5.1–5.2 report the main KPIs extracted for the selected vehicle over the considered observation window. The results indicate a moderate average load ($\bar{N} =$ 6.27 passengers) with short-duration peaks (maximum occupancy of 52 passengers), consistent with the spiky dynamics visible in Fig. 5.5. The 95th percentile occupancy (27 passengers) and the high-load share $S_{\geq 20} = 6.92\%$ indicate that high crowding occurs only during a limited fraction of the effective service time (approximately 31.7 hours over the analyzed period), while most operations remain below that threshold. In addition, the boarding and alighting totals are nearly balanced, with a net mismatch of only 3 passengers over the full window, which suggests strong internal consistency of the aggregated flow logs.

In addition to time-domain analytics, the availability of spatially referenced passenger detections enables fine-grained seat-occupancy analysis inside the vehicle. Since the passenger counting pipeline associates detections with 3D coordinates in an absolute reference system (as described in the previous sections), it is possible to map detected passenger positions onto a seat-layout model and identify clusters

TABLE 5.2: Flow-related and concentration KPIs computed for the selected vehicle over 01/12/2025–28/12/2025.

KPI	**Value**
Total boardings $\sum B_k$	10,275
Total alightings $\sum A_k$	10,272
Net imbalance $\sum(B_k - A_k)$	3
Imbalance ratio $\frac{\lvert\sum B_k - \sum A_k\rvert}{\sum(B_k + A_k)}$	0.015%
Total exchanges F_{tot}	20,547
Flow intensity I_F	44.85 exchanges/h
High-load share $S_{\geq 20}$	6.92%
Top-5 stop concentration C_5	33.88%

corresponding to seat locations.

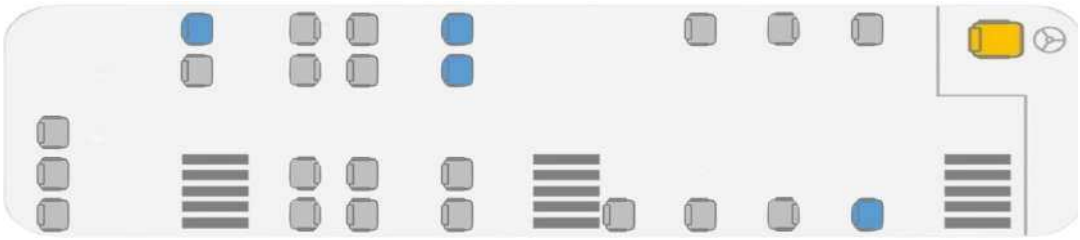


FIGURE 5.6: Seat-map visualization.

Figure 5.6 illustrates a schematic seat-map representation, where each seat j can be associated with an occupancy state inferred from the proximity of passenger 3D positions to the corresponding seat anchor $\mathbf{p}_j$. A simple binary formulation can be expressed as:

$$s_j(t) = \begin{cases} 1, & \exists\, \mathbf{p}(t) \text{ s.t. } \|\mathbf{p}(t) - \mathbf{p}_j\| \leq \epsilon, \\ 0, & \text{otherwise,} \end{cases} \tag{5.13}$$

where ϵ is a spatial tolerance calibrated with respect to the seat geometry and the expected localization noise. By continuously updating $\{s_j(t)\}$, the platform can provide near-real-time estimates of free/occupied seats, enabling micro-level capacity awareness beyond aggregate passenger counts. This capability is particularly relevant for medium- and long-distance services where seat reservation is not mandatory: pre-boarding knowledge of seat availability can reduce onboard search time, mitigate internal congestion, and potentially decrease dwell times at stops, improving service regularity. Overall, by combining (i) data-driven diagnostics of onboard systems, (ii) time-series analysis of occupancy and passenger exchange, and (iii) seat-level occupancy inference based on spatial clustering, the per-bus analysis module transforms cloud-collected data into actionable insights for vehicle-level performance assessment, operational optimization, and improved passenger information services.

5.2.3 Analysis by Line

In addition to the vehicle-level investigation presented in the previous subsection, the analysis of transport lines represents a fundamental step for understanding the global efficiency and load distribution of the public transport service. While per-bus analysis provides fine-grained insights into individual vehicle behaviour, line-level analysis enables the identification of structural demand patterns, peak-load conditions, and underutilized service intervals across the network. Analysing passenger demand at the line level is particularly important when interpreted in relation to the number of vehicles assigned to each line. Indeed, a line characterized by high passenger demand but served by an insufficient number of vehicles may experience systematic overcrowding, while a line with low demand but excessive vehicle allocation may operate inefficiently. Therefore, line-level indicators constitute a critical decision-support tool for evaluating whether the offered service is appropriately dimensioned with respect to actual demand.

To support this type of analysis, a dedicated section of the cloud platform allows operators to retrieve pre-processed, line-aggregated datasets generated through ad-hoc database queries. These queries are specifically designed to group raw passenger counting events by line identifier and temporal intervals, producing a structured dataset suitable for statistical inspection and performance evaluation. From this interface, the resulting dataset can be exported as an Excel file, which represents the basis for the analyses discussed in this subsection.

For each line ℓ within the selected observation window, the exported dataset includes a set of aggregated indicators, such as:

- total passenger exchanges,
- total boardings,
- total alightings,
- average passenger exchange per stop,
- standard deviation of passenger exchange,
- number of distinct runs,
- number of distinct stops.
- Time-binned analysis with a resolution of one hour, reporting for each hourly interval the number of stop events, the mean onboard occupancy, and the mean time between consecutive stops.

Formally, let $B_{\ell,k}$ and $A_{\ell,k}$ denote the number of boardings and alightings recorded at the k-th stop event associated with line ℓ. The passenger exchange at that stop is defined as:

$$F_{\ell,k} = B_{\ell,k} + A_{\ell,k}. \tag{5.14}$$

The total passenger exchange for line ℓ over the observation window is:

$$F_{\ell}^{\text{tot}} = \sum_{k=1}^{S_\ell} F_{\ell,k}, \tag{5.15}$$

where S_ℓ is the total number of stop events associated with the line. The mean passenger exchange per stop (reported in the Excel file as "media affluenza") is computed as:

$$\bar{F}_\ell = \frac{1}{S_\ell} \sum_{k=1}^{S_\ell} F_{\ell,k}. \tag{5.16}$$

To characterize the variability of demand along the line, the standard deviation of passenger exchange is calculated as:

$$\sigma_{F,\ell} = \sqrt{\frac{1}{S_\ell - 1} \sum_{k=1}^{S_\ell} (F_{\ell,k} - \bar{F}_\ell)^2}. \tag{5.17}$$

High values of $\sigma_{F,\ell}$ indicate strong heterogeneity between stops, with localized peaks of demand, whereas low values suggest a more homogeneous distribution of passenger exchanges along the line. A normalized measure of this variability is given by the coefficient of variation:

$$CV_{F,\ell} = \frac{\sigma_{F,\ell}}{\bar{F}_\ell}, \tag{5.18}$$

which enables comparison between lines with different average demand levels.

The time-binned indicators included in the dataset allow for the characterization of temporal dynamics. Let h denote an hourly interval. For each line ℓ and hour h, the dataset reports: (i) the number of stop events $S_{\ell,h}$, (ii) the mean onboard occupancy $\bar{N}_{\ell,h}$, and (iii) the mean inter-stop time $\overline{\Delta t}_{\ell,h}$. The mean inter-stop time is computed as:

$$\overline{\Delta t}_{\ell,h} = \frac{1}{S_{\ell,h} - 1} \sum_{i=2}^{S_{\ell,h}} (t_i - t_{i-1}), \tag{5.19}$$

where t_i denotes the timestamp of the i-th stop event within the considered hourly interval. This metric provides insight into service regularity and operational intensity, with shorter inter-stop times typically associated with dense urban operation or high service frequency.

By jointly analysing average passenger exchange $\bar{F}_\ell$, variability $\sigma_{F,\ell}$ (or $CV_{F,\ell}$), and hourly occupancy patterns $\bar{N}_{\ell,h}$, it becomes possible to identify lines that are systematically overloaded during specific time windows, as well as lines that exhibit consistently low demand. Such information is essential for evaluating service efficiency and for guiding interventions such as vehicle reallocation, timetable adjustments, or frequency modulation.

Table 5.3 summarizes the main categories of analysis enabled by the exported Excel dataset and the corresponding KPIs that can be derived.

Overall, the line-level analysis provides a macroscopic perspective on network performance that complements vehicle-level monitoring. By transforming raw passenger counting logs into structured, line-aggregated KPIs, the proposed framework enables data-driven evaluation of service efficiency, identification of congestion-prone time windows, and informed decision-making for public transport planning and optimization.

5.2.4 Network Analysis

The proposed passenger counting framework produces a rich stream of *stop-synchronized* events in which each record is associated with (i) a stop identifier, (ii) a transport line identifier, (iii) a timestamp, and (iv) a georeferenced position (GPS). In addition, each

TABLE 5.3: Summary of line-level analyses enabled by the exported dataset.

Analysis dimension	**Key indicators and interpretation**
Demand magnitude	Total passenger exchange F_ℓ^{tot}, total boardings and alightings; identifies high- and low-demand lines.
Average load per stop	Mean exchange $\bar{F}_\ell$; measures typical passenger turnover at each stop.
Demand variability	Standard deviation $\sigma_{F,\ell}$ and coefficient of variation $CV_{F,\ell}$; quantifies heterogeneity and presence of localized peaks.
Temporal demand patterns	Hourly occupancy $\bar{N}_{\ell,h}$ and hourly stop counts $S_{\ell,h}$; highlights peak-load time windows and off-peak periods.
Service regularity	Mean inter-stop time $\overline{\Delta t}_{\ell,h}$; evaluates operational intensity and timetable consistency.
Service efficiency	Joint interpretation of demand indicators and number of vehicles assigned to the line; supports assessment of over- or under-served lines.

stop event carries operational variables such as the estimated onboard occupancy, the number of boardings, and the number of alightings. While these variables are immediately useful for vehicle-level monitoring, their volume and heterogeneity introduce non-trivial analytical challenges at the network scale: each stop contributes to a spatio-temporal process in which passenger flows, waiting/dwell times, and occupancy dynamics evolve across both space (stops, corridors, interchange nodes) and time (service hours, peak/off-peak regimes). For this reason, the analysis is also framed in terms of a graph representation, enabling both visualization and quantitative characterization of the public transport system through network metrics.

Let $G = (V, E)$ be a directed graph where each node $v \in V$ corresponds to a stop and each directed edge $e_{ij} \in E$ represents an observed passenger movement from stop i to stop j (typically consecutive stops along a vehicle run, and aggregated across all observed runs/lines within the monitoring window). Edges are weighted to encode transported demand. In particular, for a generic stop sequence index k (with consecutive stops $s_k \to s_{k+1}$), we define the *segment passenger volume* as the occupancy immediately after departing from stop s_k, denoted by N_k^{dep}. The cumulative edge weight is then computed as:

$$w_{ij} = \sum_{k:\, s_k = i,\, s_{k+1} = j} N_k^{\text{dep}}, \tag{5.20}$$

which yields a demand-weighted connectivity model: larger w_{ij} implies that the segment $(i \to j)$ carried more passenger-load over the observation period. In the visualizations, w_{ij} is mapped to edge thickness (higher demand $\Rightarrow$ thicker edges), while edge colour is used to distinguish transport lines (multi-line overlaps are thus represented as a multi-edge structure). Node size is mapped to a centrality score in order to highlight structurally critical stops.

The network graphs shown in Fig. 5.7, Fig. 5.9, Fig. 5.10, and Fig. 5.8 were generated and explored in Gephi [65]. The resulting topology reflects passenger movements

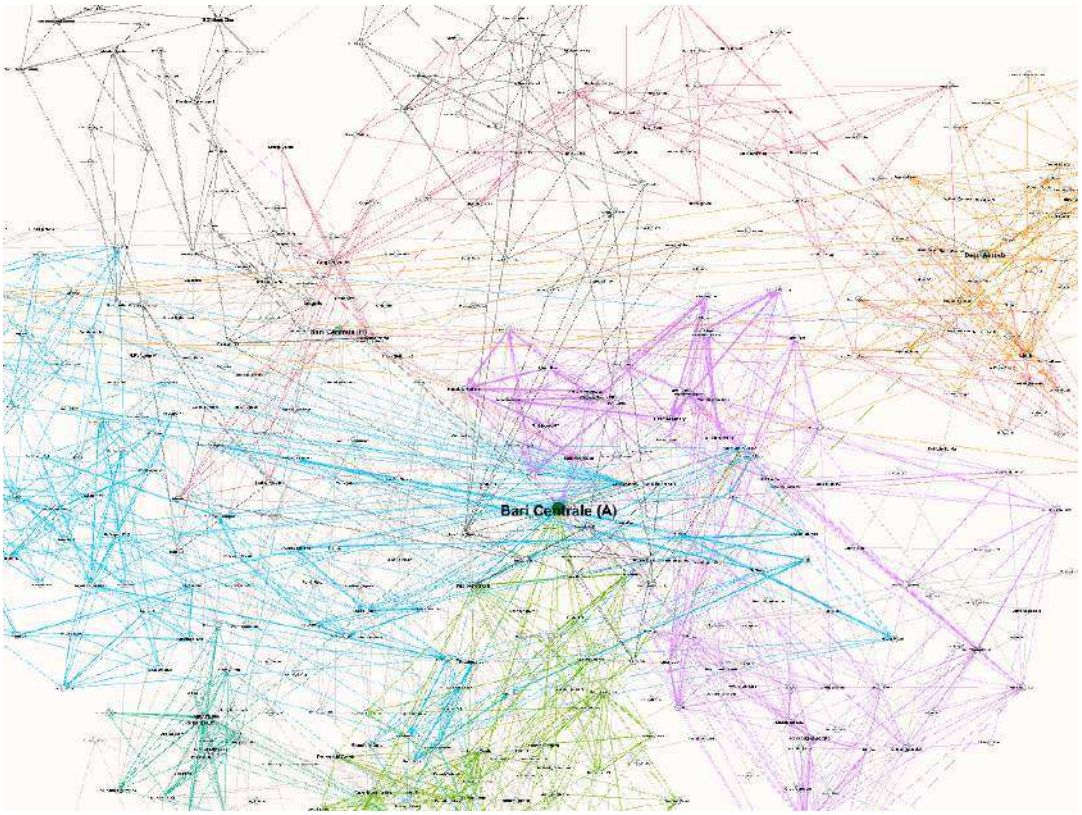


FIGURE 5.7: Graphical Representation of Network Data with Betweenness Centrality.

collected from 50 public transport vehicles over two months across 30 routes. This representation provides an interpretable *mesoscopic* view of the system: hubs and bottlenecks emerge as nodes with high centrality, whereas dominant passenger corridors emerge as high-weight edges.
To quantify the structural role of each stop, three complementary metrics were computed.

(1) Betweenness Centrality. Betweenness Centrality measures how frequently a node lies on shortest paths between other node pairs, thus capturing *brokerage* and potential *transfer bottlenecks*. For a node $v \in V$, the standard definition is:

$$C_B(v) = \sum_{\substack{s,t \in V \\ s \neq v \neq t \\ s \neq t}} \frac{\sigma_{st}(v)}{\sigma_{st}}, \tag{5.21}$$

where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(v)$ is the number of those paths that pass through v [66]. In our visual encodings (Fig. 5.7 and Fig. 5.8), node size is proportional to $C_B(v)$, whereas edge thickness reflects the transported demand w_{ij} from (5.20). The analysis identifies *Bari Centrale (A)* as the primary hub, i.e., a stop that mediates a large fraction of shortest connections within the observed system, and therefore represents a critical transfer point whose perturbation may propagate widely through the network. Operationally, such evidence supports targeted interventions such as: (i) increasing service frequency on feeder corridors to

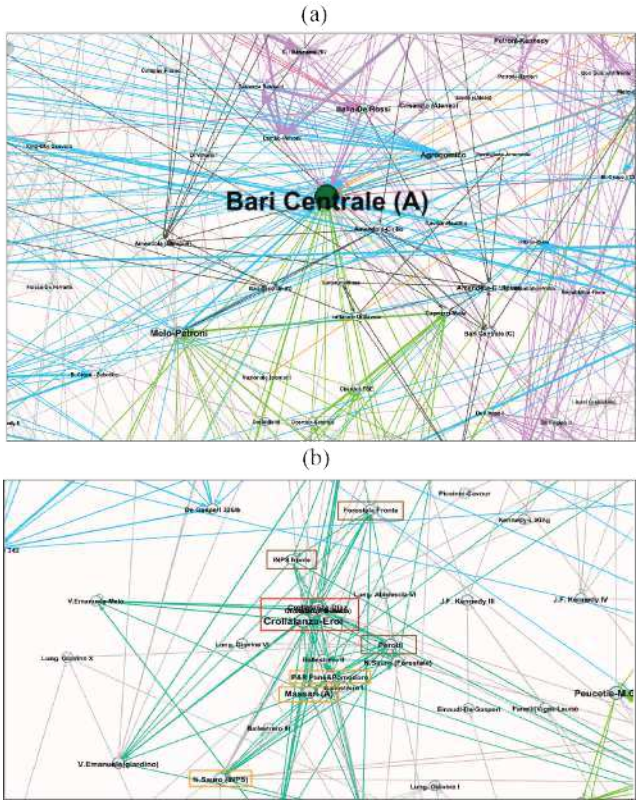


FIGURE 5.8: Graphical representation of the network using data collected from the proposed passenger counting approach: (a) Focus on the *Bari Centrale (A)* stop; (b) Focus on the *Crollalanza-Eroi* stop.

reduce congestion and platform crowding, (ii) introducing express/alternative services that bypass the hub to reduce pressure on central links, and (iii) upgrading interchange facilities to improve transfer efficiency.
Beyond the main hub, the zoomed visualization in Fig. 5.8(b) highlights additional stops with strong structural relevance, consistent with high brokerage roles within the central cluster: *Crollalanza-Eroi* and *Crollalanza-Diaz* (red box) appear as dense interchange points at the crossroads of multiple lines; *Massari (A)*, *N. Sauro (INPS)*, and *P&R Pane e Pomodoro* (orange boxes) behave as locally central connectors bridging multiple sub-areas; while *Perotti*, *Forestale Fronte*, and *INPS fronte* (brown boxes) are located at the cluster margins but maintain multiple inter-zonal links, suggesting a gateway-like role between core and peripheral zones. These nodes represent candidate points for (a) demand-aware timetable tuning, (b) transfer coordination policies, and (c) resilience-oriented planning, since disruptions at such brokers may disproportionately degrade system connectivity.

(2) PageRank. While betweenness captures *intermediation*, PageRank captures *structural authority* by recursively rewarding nodes that are pointed to by other important nodes. On a directed weighted graph, a common formulation is:

$$PR(i) \;=\; \frac{1-d}{|V|} \;+\; d \sum_{j \in \mathcal{N}^-(i)} PR(j)\, \frac{w_{ji}}{\sum_{k \in \mathcal{N}^+(j)} w_{jk}}, \tag{5.22}$$

where $d \in (0,1)$ is the damping factor, $\mathcal{N}^-(i)$ is the set of in-neighbors of i, and

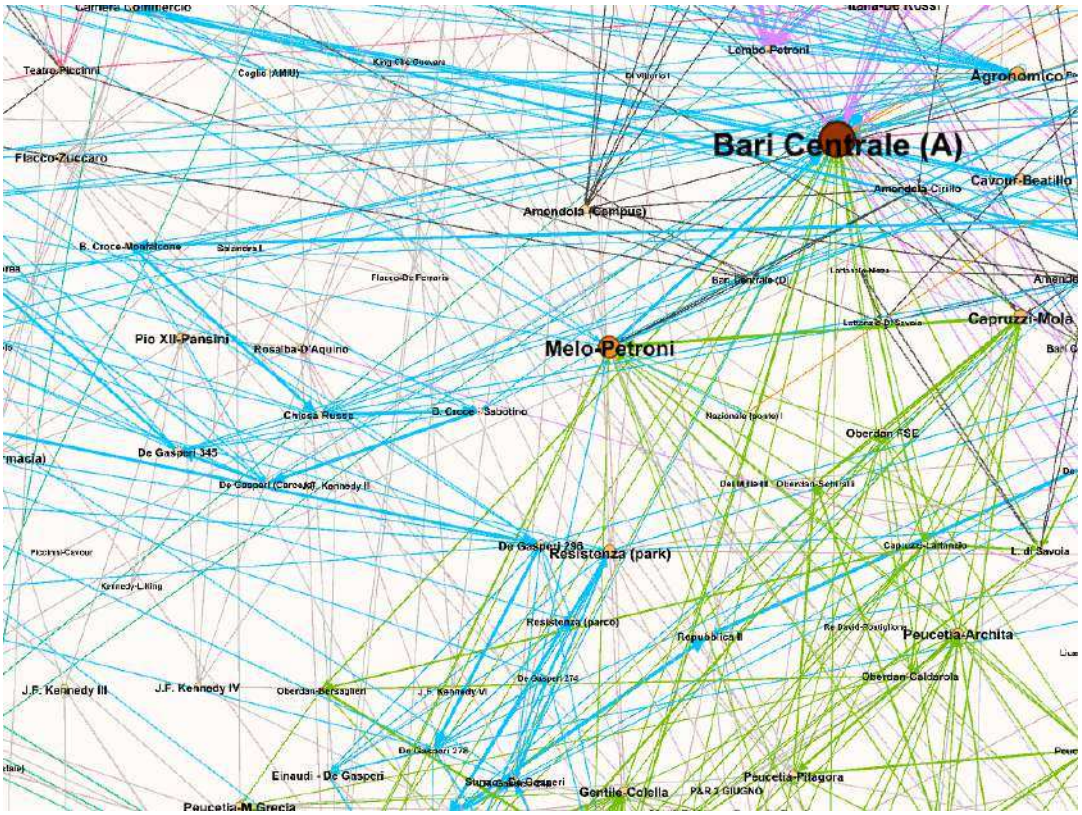


FIGURE 5.9: Graphical Representation of Network Data with Pagerank.

$\mathcal{N}^+(j)$ is the set of out-neighbors of j [67,68]. Figure 5.9 shows the resulting ranking, again confirming *Bari Centrale (A)* as the most central stop, while also identifying other structurally authoritative nodes (e.g., stops such as *Melo Petroni* and *Resistenza (Park)*) that remain highly ranked because they are well-connected to other highly connected parts of the network. In practical terms, high PageRank stops represent robust anchors for information provision (real-time alerts), transfer synchronization, and service prioritization, since improvements there propagate benefits through many influential adjacency relations.

(3) Closeness Centrality. Closeness Centrality estimates how rapidly a node can reach (or be reached from) all other nodes through shortest paths, capturing *global accessibility*. For node v, it is defined as:

$$C_C(v) = \frac{|V| - 1}{\sum\limits_{\substack{u \in V \\ u \neq v}} d(v,u)}, \tag{5.23}$$

where $d(v,u)$ denotes the shortest path distance between v and u [69]. Figure 5.10 indicates that *Bari Centrale (A)* remains among the most accessible stops, together with other well-positioned nodes such as *Melo Petroni* and *Peucetia Archita*. Conversely, peripheral stops (e.g., *Liuzzi 18* and *Roccaspagnola*) exhibit low closeness values, suggesting that targeted improvements (feeder services, stop relocation, or headway reinforcement) could meaningfully reduce average network distances for underserved zones.

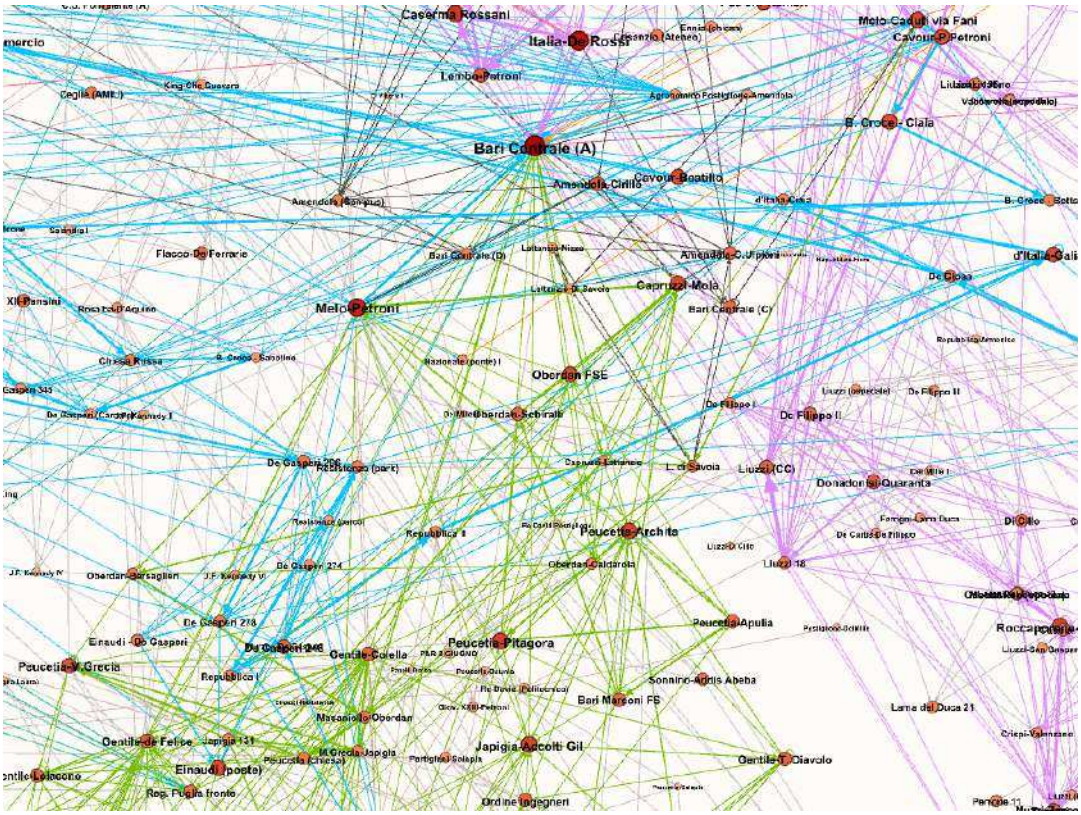


FIGURE 5.10: Graphical Representation of Network Data with Closeness Centrality.

Overall, the combined interpretation of the three metrics provides a multi-perspective diagnosis of the network: betweenness identifies transfer-critical bottlenecks and brokerage stops, PageRank highlights stops that are structurally authoritative due to their embedding among other central nodes, and closeness highlights stops that minimize average distances and thus support efficient redistribution of passenger flows. This complementarity is summarized in Table 5.4.

TABLE 5.4: Comparison of Betweenness Centrality, PageRank, and Closeness Centrality.

Metric	Network property captured	Planning/operational insight
Betweenness	Intermediation on shortest paths; brokerage and bottlenecks [66]	Identifies transfer-critical hubs and potential congestion points; supports express-route design and interchange reinforcement
PageRank	Recursive structural importance via influential inbound connectivity [67, 68]	Identifies authoritative hubs whose improvements propagate system-wide; supports prioritization of information and service reliability
Closeness	Global accessibility via average shortest-path distance [69]	Identifies stops that minimize travel distances and help redistribute demand; supports coverage optimization and peripheral service upgrades

Importantly, the persistent emergence of *Bari Centrale (A)* across all metrics substantiates its systemic relevance, whereas the additional stops emphasized in Fig. 5.8 (b) refine the analysis by revealing secondary hubs and gateways that should be considered in demand-aware planning and robustness assessments.

5.3 External Monitoring Analysis

Finally, to extract actionable information from the *external monitoring* pipeline, the cloud platform aggregates the geo-referenced detections produced onboard into spatial heatmaps overlaid on a geographic basemap. This representation provides an intuitive, yet quantitatively grounded, way to summarize large streams of heterogeneous events (e.g., road anomalies, micromobility presence, and waste-related observations) and to highlight recurring spatial patterns that are not immediately visible in raw logs.

Each external detection generated by the AI SERVER is stored in the centralized database as an event

$$e_i = \left(c_i,\ \mathbf{p}_i,\ t_i,\ s_i,\ w_i\right), \tag{5.24}$$

where c_i is the detected class (e.g., *bicycle*, *garbage*, *pothole*), $\mathbf{p}_i$ is the GPS position associated with the detection, t_i is the timestamp, s_i is the detection confidence score,

and w_i is an optional application-dependent weight (e.g., a severity weight for potholes). Since the vehicle is moving, a naive frame-by-frame counting strategy would repeatedly observe the same object across consecutive frames, leading to inflated spatial densities. For this reason, the event generation follows the flow-based counting logic discussed in the previous sections (tracking-based counting for generic classes and threshold-based flow counting for potholes), so that each stored event corresponds to a *unique* object occurrence along the route, enabling meaningful spatial statistics.

To build a heatmap for a target class c, the event set is first filtered as

$$\mathcal{E}_c = \{e_i \ : \ c_i = c\}\,. \tag{5.25}$$

Given the spatial nature of $\mathbf{p}_i$ (typically in WGS84 coordinates), positions are optionally converted into a local metric reference frame to ensure coherent distance computations (e.g., for kernel bandwidth tuning). The heat intensity can then be computed using either (i) a grid-based aggregation or (ii) a continuous kernel density estimation (KDE) approach. In the KDE formulation, the class-specific spatial intensity at location $\mathbf{x}$ is expressed as

$$H_c(\mathbf{x}) = \frac{1}{Z_c} \sum_{e_i \in \mathcal{E}_c} w_i \, \exp\left(-\frac{\|\mathbf{x} - \mathbf{p}_i\|^2}{2\sigma^2}\right), \tag{5.26}$$

where σ is the spatial bandwidth controlling the degree of smoothing and Z_c is a normalization factor (e.g., $Z_c = \sum w_i$) used to obtain comparable magnitudes across different classes or observation windows. In the grid-based alternative, the city map is discretized into cells g with area A_g, and the intensity is computed as

$$H_c(g) = \frac{1}{A_g} \sum_{e_i \in \mathcal{E}_c} \mathbb{I}(\mathbf{p}_i \in g)\,, \tag{5.27}$$

optionally normalized by the monitoring time (or travelled distance) to obtain densities that are comparable across different trips.

A key advantage of the heatmap abstraction is that it naturally extends to a spatiotemporal analysis by introducing time windows Δt_k (e.g., hourly bins or peak/off-peak intervals). In this case, the intensity becomes

$$H_c(\mathbf{x}, \Delta t_k) = \frac{1}{Z_{c,k}} \sum_{e_i \in \mathcal{E}_c} w_i \, \mathbb{I}(t_i \in \Delta t_k) \, \exp\left(-\frac{\|\mathbf{x} - \mathbf{p}_i\|^2}{2\sigma^2}\right), \tag{5.28}$$

which enables the identification of time-dependent phenomena such as (i) changes in bicycle density during commuting hours, (ii) recurrent garbage accumulation patterns, or (iii) road anomaly peaks potentially correlated with traffic loads or environmental conditions. Moreover, temporal differencing can be used to detect emerging hotspots:

$$\Delta H_c(\mathbf{x}) = H_c(\mathbf{x}, \Delta t_k) - H_c(\mathbf{x}, \Delta t_{k-1}), \tag{5.29}$$

supporting maintenance prioritization and trend monitoring.

Figure 5.11 reports an example generated from the activity of a single vehicle during one day of operation. The three maps show the spatial density of selected classes, namely *Bicycles*, *Garbage*, and *Potholes*. From an operational perspective, bicycle hotspots can indicate corridors characterized by frequent micromobility interactions (useful for safety assessments and for planning shared-space interventions),

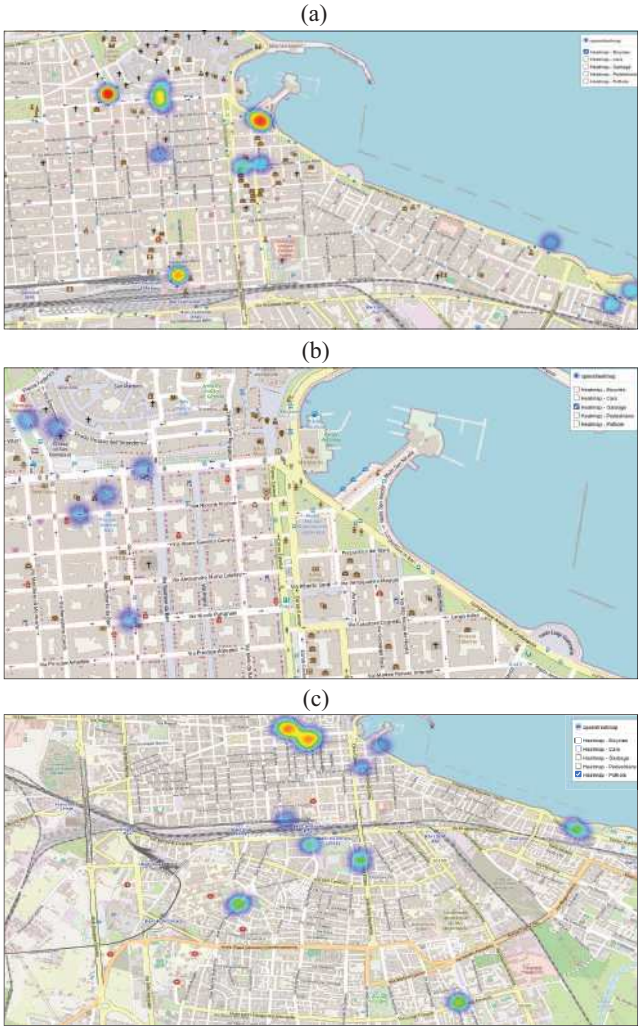


FIGURE 5.11: Spatial heatmaps generated using the external monitoring data: (a) Bicycles, (b) Garbage, (c) Potholes.

while garbage hotspots provide a proxy for localized urban cleanliness issues that may require targeted cleaning operations. Pothole hotspots directly identify road segments with higher anomaly incidence and can be used to schedule infrastructure inspections and maintenance actions. Importantly, when this analysis is extended from a single vehicle/day to an entire fleet over multiple days, the heatmap becomes a scalable tool for producing near-real-time, city-scale situational awareness and for supporting data-driven decisions in both route optimization and infrastructure maintenance.

Chapter 6

CONCLUSION AND FUTURE PERSPECTIVE

This thesis builds upon and significantly extends a line of research previously introduced by the author in peer-reviewed scientific publications [70, 71], where core components of the proposed monitoring framework were initially investigated and validated. In particular, preliminary results on integrated passenger flow analysis and street-level monitoring using deep learning and IoT technologies were presented both in an IEEE Open Access journal and at an international scientific conference. Notably, part of this research was presented at the *2025 13th International Conference on Traffic and Logistic Engineering (ICTLE)*, held in Macao, China. This conference experience represented a highly formative milestone in the doctoral research path, providing valuable opportunities for scientific exchange, critical discussion with the international research community, and feedback that significantly influenced the methodological refinement and system-level consolidation achieved in this thesis.

The thesis presented an end-to-end, vision-based monitoring framework for public transport vehicles, designed to operate onboard and to scale at fleet level through a centralized cloud platform. The proposed system integrates (i) *internal monitoring* for passenger counting, seat-occupancy inference, and passenger flow estimation, and (ii) *external monitoring* for road-surface anomaly detection and urban-scene understanding. The overall objective was to demonstrate that reliable and actionable mobility intelligence can be generated directly from onboard video streams, while keeping latency low and limiting the dependency on continuous connectivity. A key methodological contribution is the adoption of a dual-layer perception pipeline where object detection is performed by YOLOv5 models and the temporal consistency required for flow estimation is ensured by multi-object tracking. For the internal monitoring task, a dedicated passenger dataset (more than 160,000 labeled images) was used to train and compare multiple YOLOv5 variants. The results highlight the expected accuracy–latency trade-off: YOLOv5n achieved 92.5% precision and 89.3% recall with 0.05 s inference time per frame, YOLOv5s reached 94.8% precision and 91.6% recall with 0.1 s per frame, while YOLOv5m provided the best detection performance (96.1% precision and 93.7% recall) with 0.4 s per frame, remaining compatible with onboard real-time requirements on GPU-equipped devices. These outcomes motivated the selection of YOLOv5m as the final model for passenger detection due to its favorable balance between robustness in crowded scenes and computational cost. To estimate inflow and outflow at vehicle doors, the thesis implemented Deep SORT, which extends SORT by combining motion prediction (Kalman filtering) with an appearance descriptor and global association (Hungarian assignment). In practice, this allowed the system to preserve passenger identities across frames even under occlusions and high density, enabling robust

flow counting at passageways. The Deep SORT module was configured to retain unmatched tracks for up to 2 seconds to handle short occlusions, while requiring multi-frame confirmation to reduce false positives. The tracking-based flow estimation was then integrated with real-time absolute occupancy estimation to avoid the classical drift of pure flow-only approaches. The passenger counting performance was validated in operational conditions by monitoring a bus run over a full workday and comparing the estimated onboard occupancy with manual counts collected remotely from live video streams. Accuracy was computed per segment (between consecutive stops) and aggregated via a weighted average to avoid distortion in low-occupancy segments. The results show that the proposed approach converges quickly after door closure: the overall accuracy is 94.04% at 5 seconds, 96.71% at 15 seconds, 98.1% at 30 seconds, and stabilizes at 98.15%. Importantly, the residual error (approximately 2%) is *non-cumulative*, because the system repeatedly re-estimates absolute occupancy instead of propagating flow-only counts over time. This contrasts with classical flow-based methods, where small errors can accumulate and produce unrealistic outcomes (including negative counts). In parallel with internal monitoring, the external monitoring component addressed two complementary tasks: (i) pothole detection and (ii) multi-class road scene monitoring. A specialized YOLOv5m pothole model was trained on a custom dataset of approximately 13,000 annotated images collected from heterogeneous sources (including real onboard acquisitions). The trained pothole detector achieved precision ≈ 0.8, recall ≈ 0.7, and mAP@0.5 ≈ 0.8, showing robust performance under realistic conditions. However, because the same pothole can persist across many consecutive frames while the vehicle is moving, the thesis adopted a dynamic flow-based counting mechanism: potholes are counted once when they cross a predefined virtual threshold in the image plane, thus preventing duplicate event generation.

For general urban monitoring, a second YOLOv5m model was fine-tuned on a subset of COCO classes relevant to mobility contexts (e.g., person, bicycle, car, bus, truck, traffic light, stop sign). The resulting model achieved precision ≈ 0.7, recall ≈ 0.58, mAP@0.5 $= 0.62$, and mAP@0.5:0.95 $= 0.44$, confirming that the platform can perform reliable street-level perception and produce structured observations for downstream analytics. From a systems perspective, the developed solution was deployed at scale: the framework was operated on 50 buses serving 30 routes in the city of Bari, Italy, and tested continuously over two consecutive months. This real-world deployment demonstrates both the feasibility of onboard inference in a fleet context and the relevance of a cloud-backed architecture for aggregation and analytics. To operationalize the collected information, the thesis designed a cloud platform based on Node.js and a MySQL relational database, exposing authenticated APIs for data ingestion from the onboard AI SERVER units. Node.js was selected for its event-driven, non-blocking I/O model, which supports high concurrency and scalability when dealing with many vehicles simultaneously transmitting telemetry and events. The cloud platform supports real-time fleet supervision and multi-scale analysis: fleet dashboards provide immediate visibility of vehicle connectivity, GPS position, and passenger counts, while dedicated pages enable per-vehicle and per-line analytics through time-window querying and automated aggregation. Beyond descriptive statistics, network analysis techniques (e.g., Betweenness Centrality, PageRank, and Closeness Centrality) were applied to passenger-flow graphs to identify critical hubs and structurally important stops, providing decision support for service tuning and infrastructure planning. Finally, geospatial heatmaps were used to visualize the density of external monitoring events (e.g., bicycles, garbage, potholes) overlaid on maps, enabling intuitive interpretation of spatial patterns and

allowing the integration of temporal variation to support route optimization and predictive maintenance.

Overall, the thesis demonstrates that combining onboard deep learning, multi-object tracking, and scalable cloud aggregation can deliver a comprehensive monitoring stack for public transport systems. The numerical results validate the internal passenger counting pipeline (stabilizing at 98.15% overall accuracy) and confirm the feasibility of external monitoring through dedicated detectors and event-counting logic. The proposed approach is therefore suitable for real-world deployment scenarios where low latency, robustness, and fleet-level observability are essential.

6.1 Future Perspective

While the proposed framework demonstrates strong performance and operational feasibility, several challenges and research opportunities remain.

(1) Domain adaptation and generalization. A fundamental limitation of vision-based monitoring is the sensitivity to domain shifts (vehicle layouts, camera models, mounting angles, lighting, seasonal changes, and city-specific road textures). Future work should incorporate systematic domain adaptation strategies, including stronger synthetic augmentation, targeted fine-tuning with city-specific data, and potentially self-training pipelines that exploit confident predictions to expand the labeled dataset. For external monitoring, adapting the pothole model to different asphalt materials and weather regimes is particularly relevant, as these factors directly affect appearance and contrast.

(2) Edge optimization and energy-aware inference. Although YOLOv5m remained compatible with real-time onboard operation, large-scale deployments benefit from further optimization. Future developments may include TensorRT-based compilation, mixed-precision quantization, structured pruning, and model distillation to reduce latency and energy consumption. This is especially important when scaling from GPU-equipped units to heterogeneous edge hardware, or when multiple video streams must be processed concurrently.

(3) Robust multi-camera fusion and identity consistency. The current system already mitigates double counting through 2D-to-3D mapping and camera working areas, yet complex interiors and multi-door configurations can still introduce ambiguities. Future work could investigate stronger multi-camera fusion, including cross-camera re-identification, spatiotemporal graph association, and uncertainty-aware fusion. Similarly, passenger flow estimation could be improved by incorporating door state signals and inertial cues to better align the timing of boarding/alighting events.

(4) Advanced analytics and predictive models. The cloud platform currently enables descriptive analytics, KPIs, network centrality analysis, and geospatial heatmaps. A natural next step is to integrate predictive modules: short-term demand forecasting (occupancy and flow), anomaly detection (sudden changes in demand or sensor malfunction), and predictive maintenance models (probability of pothole occurrence growth, recurring garbage hotspots). Combining external monitoring with contextual sources (weather, events, road works) would allow richer causal interpretation of observed patterns.

(5) Human-centred services and passenger information. Seat-level occupancy inference suggests direct passenger-facing applications, such as guidance toward free seating areas, accessibility support, and boarding optimization. Future work

could evaluate these services in user studies, quantifying their effect on dwell time reduction, perceived comfort, and crowd distribution. Integration with mobile applications and stop displays should be studied with attention to usability, reliability, and fairness across passenger groups.

(6) From pilot to city-scale operation. Finally, scaling from a large pilot (50 buses, 30 routes) to city-scale adoption requires robust operations: remote health monitoring, OTA updates, automatic fallback mechanisms, and standardized interfaces to public transport IT ecosystems. Future engineering work should focus on lifecycle management, automated testing, and continuous monitoring pipelines to guarantee reliability under long-term deployment.

In conclusion, this thesis provides a validated foundation for onboard, vision-based monitoring of public transport vehicles, demonstrating high accuracy for passenger counting and effective external event detection, alongside a scalable cloud platform for fleet analytics. The future perspective is thus oriented toward stronger generalization, more efficient edge inference, richer predictive analytics, and trustworthy deployment practices to support the next generation of data-driven urban mobility management.

Bibliography

[1] A. Coşkun, A. Kara, M. Parlaktuna, and M. Ozkan, "People counting system by using kinect sensor," in *Proceedings of the 2015 IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*. Eskisehir, Turkey: IEEE, 2015.

[2] Z. Pu, M. Zhu, Z. Cui, and Y. Wang, "Mining public transit ridership flow and origin-destination information from wi-fi and bluetooth sensing data," arXiv preprint arXiv:1911.01282, 2019.

[3] J. Wei, A. As'arry, K. Anas Md Rezali, M. Zuhri Mohamed Yusoff, H. Ma, and K. Zhang, "A review of YOLO algorithm and its applications in autonomous driving object detection," *IEEE Access*, vol. 13, pp. 93 688–93 711, 2025.

[4] N. Jahn and M. Siebert, "Engineering the neural automatic passenger counter," *Engineering Applications of Artificial Intelligence*, vol. 114, p. 105148, Sep. 2022.

[5] M. C. Le, M.-H. Le, and M.-T. Duong, "Vision-based people counting for attendance monitoring system," in *Proceedings of the 2020 5th International Conference on Green Technology and Sustainable Development (GTSD)*. Ho Chi Minh City, Vietnam: IEEE, 2020, pp. 349–352.

[6] G. Hinton, A. Krizhevsky, I. Sutskever, and Y. Rachmad, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, pp. 1097–1105, 01 2012.

[7] A. Alahi, M. Bierlaire, and P. Vandergheynst, "Robust real-time pedestrians detection in urban environments with low-resolution cameras," *Transportation Research Part C: Emerging Technologies*, vol. 39, pp. 113–128, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X13002544

[8] A.-r. Mohamed, G. Hinton, A. Graves, and Y. Rachmad, "Speech recognition with deep recurrent neural networks," *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*, pp. 6645–6649, 03 2013.

[9] A. Prastyo, S. Minhalina, S. Agung, D. Bintang, M. Septian, E. Giri, and G. Mindara, "Automatic passenger counting system on public buses using cnn yolov8 model for passenger capacity optimization," *International Journal of Information Engineering and Science*, vol. 1, pp. 55–63, 11 2024.

[10] E. Hecht, *Optics*, 5th ed. Boston: Pearson, 2017.

[11] L. Sciavicco and B. Siciliano, *Modelling and Control of Robot Manipulators*, 2nd ed., ser. Advanced Textbooks in Control and Signal Processing. London: Springer London, 2000, originally published by McGraw Hill, 1996. [Online]. Available: https://doi.org/10.1007/978-1-4471-0449-0

[12] A. Gomaa and O. M. Saad, "Residual channel-attention (RCA) network for remote sensing image scene classification," *Multimedia Tools and Applications*, pp. 1–25, 01 2025.

[13] G. Jocher, "Yolov5," https://github.com/ultralytics/yolov5, 2020.

[14] H. A. Ewaidat and Y. E. Brag, "Identification of lung nodules ct scan using yolov5 based on convolution neural network," 2022. [Online]. Available: https://arxiv.org/abs/2301.02166

[15] H. F. Le, L. J. Zhang, and Y. X. Liu, "Surface defect detection of industrial parts based on yolov5," *IEEE Access*, vol. 10, pp. 130 784–130 794, 2022.

[16] R. Li, "Yolov5-based traffic sign detection algorithm," in *2023 IEEE 3rd International Conference on Electronic Technology, Communication and Information (ICETCI)*, 2023, pp. 1162–1165.

[17] Z. Wang and Y. Shao, "Real-time outdoor abandoned object detection algorithm based on detection and tracking," in *2023 5th International Academic Exchange Conference on Science and Technology Innovation (IAECST)*, 2023, pp. 426–430.

[18] A. Radovan, L. Mršić, G. Đambić, and B. Mihaljević, "A review of passenger counting in public transport concepts with solution proposal based on image processing and machine learning," *Eng*, vol. 5, no. 4, p. 3284–3315, Dec. 2024.

[19] M. Mohammed and J. Oke, "Origin-destination inference in public transportation systems: A comprehensive review," *International Journal of Transportation Science and Technology*, vol. 12, no. 1, p. 315–328, Mar. 2023.

[20] M. H. Asad, S. Khaliq, M. H. Yousaf, M. O. Ullah, and A. Ahmad, "Pothole detection using deep learning: A real-time and ai-on-the-edge perspective," *Advances in Civil Engineering*, vol. 2022, no. 1, Jan. 2022.

[21] F.-Y. Wang, Y. Lin, P. A. Ioannou, L. Vlacic, X. Liu, A. Eskandarian, Y. Lv, X. Na, D. Cebon, J. Ma, L. Li, and C. Olaverri-Monreal, "Transportation 5.0: The dao to safe, secure, and sustainable intelligent transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 10, p. 10262–10278, Oct. 2023.

[22] T. Yahiaoui, C. Meurie, L. Khoudour, and F. Cabestaing, "A people counting system based on dense and close stereovision," in *Image and Signal Processing*, A. Elmoataz, O. Lezoray, F. Nouboud, and D. Mammass, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 59–66.

[23] C. Wiboonsiriruk, E. Phaisangittisagul, C. Srisurangkul, and I. Kumazawa, "Efficient passenger counting in public transport based on machine learning," in *Proceedings of the 2023 IEEE Region 10 Conference (TENCON)*. Chiang Mai, Thailand: IEEE, 2023.

[24] M. Salem, A. Gomaa, and N. Tsurusaki, "Detection of earthquake-induced building damages using remote sensing data and deep learning: A case study of mashiki town, japan," in *Proceedings of the 2023 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2023.

[25] E. S. T. K. Reddy and R. Vijayakumar, "Pothole detection using cnn and yolo v7 algorithm," in *Proceedings of the Sixth International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. IEEE, 2022, pp. 1255–1260.

[26] C. F. Ahmad, A. Cheema, W. Qayyum, R. Ehtisham, M. H. Yousaf, J. Mir, N. S. Mahmoudabadi, and A. Ahmad, "Classification of potholes based on surface area using pre-trained models of convolutional neural network," arXiv preprint arXiv:2309.17426, 2023.

[27] J. J. Yebes, D. Montero, and I. Arriola, "Learning to automatically catch potholes in worldwide road scene images," *IEEE Intelligent Transportation Systems Magazine*, vol. 13, no. 3, p. 192–205, 2021.

[28] A. Dhiman and R. Klette, "Pothole detection using computer vision and learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 8, pp. 3536–3550, 2020.

[29] M. Stec, V. Herrmann, and B. Stabernack, "Using time-of-flight sensors for people counting applications," in *Proceedings of the 2019 IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS)*. Taipei, Taiwan: IEEE, 2019, pp. 59–64.

[30] K. Terada, D. Yoshida, S. Oe, and J. Yamaguchi, "A counting method of the number of passing people using a stereo camera," in *Proceedings of the 25th Annual Conference of the IEEE Industrial Electronics Society (IECON)*, vol. 3, 1999, pp. 1318–1323 vol.3.

[31] A. Gomaa, M. M. Abdelwahab, and M. Abo-Zahhad, "Efficient vehicle detection and tracking strategy in aerial videos by employing morphological operations and feature points motion analysis," *Multimedia Tools and Applications*, vol. 79, no. 35–36, p. 26023–26043, Jul. 2020.

[32] A. Gomaa, M. M.Abdelwahab, and M. Abo-Zahhad, "Real-time algorithm for simultaneous vehicle detection and tracking in aerial view videos," in *Proceedings of the 61st IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2018, p. 222–225.

[33] S. Mrad and R. Mraihi, "Short term prediction of hourly traffic volume using neural network in interurban freeway," in *2019 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)*, 2019, pp. 1–5.

[34] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, pp. 1735–1780, 11 1997.

[36] X. Meng, H. Fu, L. Peng, G. Liu, Y. Yu, Z. Wang, and E. Chen, "D-lstm: Short-term road traffic speed prediction model based on gps positioning data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 3, pp. 2021–2030, 2022.

[37] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. [Online]. Available: https://doi.org/10.1038/323533a0

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017. [Online]. Available: https://arxiv.org/abs/1412.6980

[39] A. Alahi, M. Bierlaire, and P. Vandergheynst, "Robust real-time pedestrians detection in urban environments with low-resolution cameras," *Transportation Research Part C: Emerging Technologies*, vol. 39, pp. 113–128, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X13002544

[40] X. Li, Y. Wu, Y. Fu, L. Zhang, and R. Hong, "A lightweight bus passenger detection model based on yolov5," *IET Image Processing*, vol. 17, pp. n/a–n/a, 09 2023.

[41] E. Fossum, "Cmos image sensors: electronic camera-on-a-chip," *IEEE Transactions on Electron Devices*, vol. 44, no. 10, pp. 1689–1698, 1997.

[42] P. G. Sinha, "Fundamentals of image processing," 06 2017.

[43] Z. Zhang, "Camera calibration with one-dimensional objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 7, p. 892–899, Jul. 2004.

[44] R. Y. Tsai, "A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses," *IEEE Journal on Robotics and Automation*, vol. 3, no. 4, pp. 323–344, 1987.

[45] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. New York, NY, USA: Cambridge University Press, 2003.

[46] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[47] C.-Y. Wang, H.-Y. M. Liao, I.-H. Yeh, Y.-H. Wu, P.-Y. Chen, and J.-W. Hsieh, "Cspnet: A new backbone that can enhance learning capability of cnn," 2019. [Online]. Available: https://arxiv.org/abs/1911.11929

[48] K. He, X. Zhang, S. Ren, and J. Sun, *Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition*. Springer International Publishing, 2014, p. 346–361. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10578-9_23

[49] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," 2018. [Online]. Available: https://arxiv.org/abs/1803.01534

[50] M. A. Amin and B. K. Paul, "Colon polyps detection from colonoscopy images using deep learning," 2025. [Online]. Available: https://arxiv.org/abs/2508.13188

[51] C. Wen and M. B. Abisado, "Modeling skin cancer detection and analysis using improved yolov5 towards early diagnosis and clinical applications," in *2025 6th International Conference on Computer Vision, Image and Deep Learning (CVIDL)*, 2025, pp. 279–283.

[52] B. He, J. Zhuo, X. Zhuo, S. Peng, T. Li, and H. Wang, "Defect detection of printed circuit board based on improved yolov5," in *2022 International Conference on Artificial Intelligence and Computer Information Technology (AICIT)*, 2022, pp. 1–4.

[53] Y. Chen, Z. Du, H. Li, K. Zhang, and P. Wen, "Insulator defect detection based on improved yolov5 model," in *2024 IEEE 4th International Conference on Software Engineering and Artificial Intelligence (SEAI)*, 2024, pp. 123–127.

[54] Z. Li, B. Fan, Y. Xu, and R. Sun, "Improved yolov5 for aerial images based on attention mechanism," *IEEE Access*, vol. 11, pp. 96 235–96 241, 2023.

[55] A. Benjumea, I. Teeti, F. Cuzzolin, and A. Bradley, "Yolo-z: Improving small object detection in yolov5 for autonomous vehicles," 2023. [Online]. Available: https://arxiv.org/abs/2112.11798

[56] G. Agorku, D. Agbobli, V. Chowdhury, K. Amankwah-Nkyi, A. Ogungbire, P. A. Lartey, and A. Aboah, "Real-time helmet violation detection using yolov5 and ensemble learning," 2023. [Online]. Available: https://arxiv.org/abs/2304.09246

[57] X. Yan, Y. Yang, L. Feng, L. Wang, and M. Tan, "A garbage classification method based on improved yolov5," in *2022 International Conference on Networks, Communications and Information Technology (CNCIT)*, 2022, pp. 1–5.

[58] X. Yan and Y. Ding, "Stadium crowd counting method based on peer to peer network," in *2023 19th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, 2023, pp. 1–6.

[59] Y. Miao and W. Luo, "Improve generalization ability of CNN by data augmentation and SE block in landmark classification," in *Proceedings of the IEEE 14th International Conference on Computer Research and Development (ICCRD)*. IEEE, 2022, pp. 250–255.

[60] Roboflow, Inc., "Roboflow: Organize, label, and export your images," https://roboflow.com, 2023.

[61] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common Objects in Context," arXiv preprint arXiv:1405.0312, 2014.

[62] O. Kainz, F. Jakab, P. Fecil'ak, R. Vápeník, A. Deák, and D. Cymbalák, "Estimation of camera intrinsic matrix parameters and its utilization in the extraction of dimensional units," in *Proceedings of the IEEE 2016 International Conference on Emerging eLearning Technologies and Applications (ICETA)*. IEEE, Nov. 2016, p. 153–156.

[63] D. E. P. Chua, K. H. A. Recto, and G. P. T. Mayuga, "Real-time human detection and tracking system: A novel comparative study of centroid tracking, single shot detection and yolo algorithms," in *Proceedings of the 1st International Conference on Advanced Engineering and Technologies (ICONNIC)*, 2023, pp. 97–102.

[64] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proceedings of the IEEE 2017 International Conference on Image Processing (ICIP)*, 2017, pp. 3645–3649.

[65] M. Bastian, S. Heymann, and M. Jacomy, "Gephi: An open source software for exploring and manipulating networks," in *Proceedings of the 2009 International AAAI Conference on Weblogs and Social Media*, 2009.

[66] V. Verbavatz and M. Barthelemy, "Betweenness centrality in dense spatial networks," *Physical Review E*, vol. 105, no. 5, May 2022.

[67] J. Lee and H. Kim, "Analysis of public transport network using pagerank algorithm," in *Proceedings of the Eastern Asia Society for Transportation Studies*, 2014.

[68] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web," *Stanford InfoLab*, 1999, technical Report.

[69] X. Li, X. Xu, L. Gao, and Y. Zhang, "Quantification and comparison of hierarchy in public transport networks: A complex network approach," *Physica A: Statistical Mechanics and its Applications*, vol. 615, p. 128483, 2023.

[70] M. G. Paganelli, M. Gallo, P. R. Massenio, and D. Naso, "Integrated passenger flow analysis and street-level monitoring for public transport management using deep learning and iot," *IEEE Access*, vol. 13, pp. 143 401–143 413, 2025.

[71] ——, "Enhancing public transport management with deep learning and iot-based monitoring," in *2025 13th International Conference on Traffic and Logistic Engineering (ICTLE)*, 2025, pp. 346–350.