

## Article

# Train Station Pedestrian Monitoring Pilot Study Using an Artificial Intelligence Approach

Gonzalo Garcia <sup>1</sup>, Sergio A. Velastin <sup>2,3,\*</sup>, Nicolas Lastra <sup>4</sup>, Heilym Ramirez <sup>4</sup>, Sebastian Seriani <sup>5</sup>  
and Gonzalo Farias <sup>4</sup>

- <sup>1</sup> College of Engineering, Virginia Commonwealth University, Richmond, VA 23284, USA; garciaga3@vcu.edu  
<sup>2</sup> School of Electronic Engineering and Computer Science, Queen Mary University of London, London E1 4NS, UK  
<sup>3</sup> Department of Computer Engineering, Universidad Carlos III de Madrid, 28911 Leganés, Spain  
<sup>4</sup> Escuela de Ingeniería Eléctrica, Pontificia Universidad Católica de Valparaíso, Valparaíso 2362804, Chile; nicolas.lastra.m@mail.pucv.cl (N.L.); heilym.ramirez@pucv.cl (H.R.); gonzalo.farias@pucv.cl (G.F.)  
<sup>5</sup> Escuela de Ingeniería de Construcción y Transporte, Pontificia Universidad Católica de Valparaíso, Valparaíso 2362804, Chile; sebastian.seriani@pucv.cl  
\* Correspondence: sergio.velastin@ieee.org

**Abstract:** Pedestrian monitoring in crowded areas like train stations has an important impact in the overall operation and management of those public spaces. An organized distribution of the different elements located inside a station will contribute not only to the safety of all passengers but will also allow for a more efficient process of the regular activities including entering/leaving the station, boarding/alighting from trains, and waiting. This improved distribution only comes by obtaining sufficiently accurate information on passengers' positions, and their derivatives like speeds, densities, traffic flow. The work described here addresses this need by using an artificial intelligence approach based on computational vision and convolutional neural networks. From the available videos taken regularly at subway stations, two methods are tested. One is based on tracking each person's bounding box from which filtered 3D kinematics are derived, including position, velocity and density. Another infers the pose and activity that a person has by analyzing its main body key points. Measurements of these quantities would enable a sensible and efficient design of inner spaces in places like railway and subway stations.

**Keywords:** artificial intelligence; computer vision; convolutional neural networks; pedestrian detection; pedestrian tracking; human pose



**Citation:** Garcia, G.; Velastin, S.A.; Lastra, N.; Ramirez, H.; Seriani, S.; Farias, G. Train Station Pedestrian Monitoring Pilot Study Using an Artificial Intelligence Approach. *Sensors* **2024**, *24*, 3377. <https://doi.org/10.3390/s24113377>

Academic Editors: Antonio Fernández-Caballero and Dominique Gruyer

Received: 26 April 2024  
Revised: 20 May 2024  
Accepted: 21 May 2024  
Published: 24 May 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The increasing use of rail as a mass urban transportation is making the efficiency in managing stations and their daily operations, a pressing topic. An optimal distribution of the passengers during their stay on the platforms, while they move, and the way they board and alight from the trains have the potential of reducing the required time for these actions, allowing a safer and more comfortable operation for the passengers. For high-frequency services, the capacity of the station is mainly determined by the dwell time of the trains. A longer time than necessary with the train stopped at the station implies a longer waiting time for the passengers as well as the entire system's capacity. This justifies the need for studying the movements of the passengers while on the platforms and their boarding and alighting times; see [1,2].

Information about the motion of the passengers and their behavior is key in the design of the layout of fixed elements located at the station and on the train, including seats for passengers, areas for organized transiting, locations of trash cans, loudspeakers, etc., to name a few. As a case in point, the Metro Santiago in Chile averages 700 million passengers per year, peaking at around 2.5 million trips per day, where the platform–train

interface (PTI) is the space with the highest numbers of interactions [3]. Due to safety reasons, the PTI for reduced-mobility passengers, such as those in wheelchairs, becomes a complex space, and special measures must be taken to mitigate the chances of accidents and improve the quality of the travel experience [4].

Having reliable information on a person's position inside the station allows the estimation of key variables such as velocity, density, and others. These quantities are useful information to help determine the overall behavior of the passengers and the way they are affected by the architectural design of the station, in terms of the distribution of objects, signage, illumination, and audible devices. Due to the high volume of video data, manually based observation at the PTI becomes practically impossible. Historically, two indirect ways to automatize this task have been developed, by monitoring the use of RFID tickets for entrance to the station and the use of cellphones inside [5].

Nowadays, technology is available for a sufficiently accurate extraction of the location of elements of interests, such as passengers, from an image or a video in a fully automatized way. Computer vision is one of the most relevant techniques in the field. This is accomplished by training an artificial intelligence algorithm based on artificial neural networks to extract the location of predefined objects belonging to different classes; see [6,7]. A popular tool of such technique is YOLOv7 [8], an open-source software that is trained to detect different categories of objects from individual images. This was supplemented with a tracking capability [9,10].

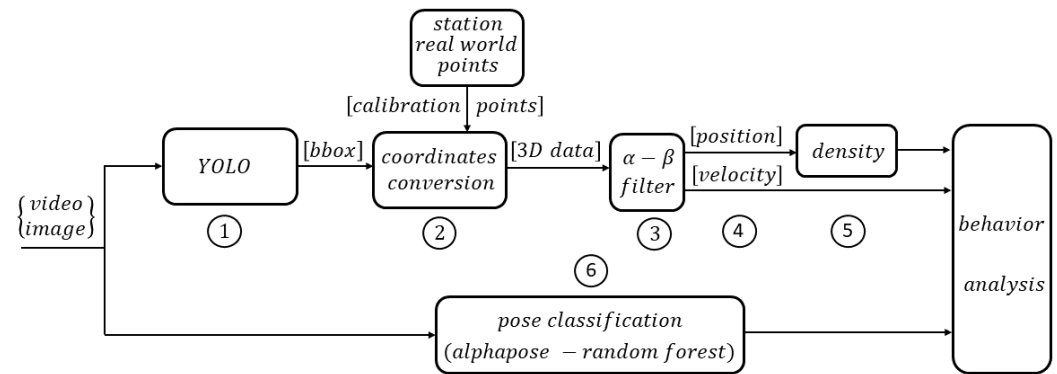
With the development of artificial intelligence, vision-based recognition of human activity has also become a field of great interest for researchers due to its significant application value. One application is face pattern recognition/extraction to infer specific activities using visible or infrared cameras [11]. Pedestrian activity recognition in a metro/train station can provide an important tool for the safe and stable operation of transportation systems. Systems are required not only to detect pedestrians but also to recognize pedestrian movements and make decisions accordingly; see [12–14]. Works such as [15] have shown that the use of pose estimators using skeletons to recognize the activity carried out by each pedestrian increases the performance of systems that predict activity. Typical daily activities in subway station scenarios are walking, sitting, standing, etc., but it is also important and possible to recognize activities that can trigger alarms such as a pedestrian falling, someone running, among other suspicious activities.

The present work is aimed in that direction by proposing a pipeline that performs the following steps as seen in Figure 1: (1) Using a pre-trained YOLOv7 model complemented with the addition of tracking capability, it extracts the positions in each image of all the detectable persons in sight, delivering a bounding box framing each individual with a unique identification number; (2) it converts the positions of each person from camera coordinates to station coordinates; (3) it applies an estimation/smoothing kinematic filter to improve the passengers dynamics by smoothing their positions penalizing outliers; (4) from the kinematic filter, it obtains velocities; (5) from the position, it determines the spatial density as a function of time. Finally, the paper presents (6) a process to recognize pedestrian activity in the station, through a person's pose classification approach.

The main contributions of this work are the following:

- Few works have been carried out on the detection and analysis of passengers by computer vision methods in metro stations (in the field of transport engineering) using the techniques described here. To the best of our knowledge, this specific application has not yet been reported in the literature.
- Obtaining transport-related variables such as speed, density, and distance, for a better assessment of the monitoring of passengers when entering/exiting the train (detection and analysis).
- A validation of the detection of passengers on the platform with respect to the video: manual/qualitative validation. The proposed approach's results are contrasted against a manual processing of the chosen videos, labeled as ground truth.

The manuscript is organized as follows: Section 1 is an introduction. Section 2 presents the algorithms proposed for extracting the passenger's basic data. Section 3 shows the results of the application of the proposed artificial intelligence methods to actual videos recorded inside a subway station, and a qualitative analysis from these results. The paper ends with the main conclusions in Section 4.



**Figure 1.** Steps carried out by the system.

## 2. Materials and Methods

### 2.1. Detection and Tracking

Detection was carried out by the open-source object detector YOLOv7, augmented with the addition of a tracking package. It accepts frames from a video or images as inputs and delivers a bounding box and tracking data of each person as output. The algorithm allows for the detection of 80 different classes of objects, but in the present work, only the person and train classes were used.

#### 2.1.1. YOLOv7 (See [8])

It is a single-shot detector, comprising several convolutional neural networks in cascade trained to detect and localize objects. YOLOv7 was trained on the MS COCO dataset [16] from scratch without using other datasets or pre-trained weights. It was trained to detect some specific classes of objects, about 80 classes, and outputs for each frame all the detected objects with the vertices of a bounding box and the probability associated with a confidence level. YOLOv7 has several advantages with respect to precision, accuracy, and computational time, in comparison with its previous versions or other algorithms: it is 120% faster, allowing a range between 5 to 160 FPS; it has a faster inference speed and higher detection accuracy, which is very effective for this work; and it allows us to work in real time with lesser computational time.

#### 2.1.2. Tracking Capability Added to YOLOv7

There is also an addition to YOLOv7 allowing for tracking capability and another with segmentation, but for this work, we used the tracking version only [9,10]. This was achieved by initially assigning each detected object a unique detection number, which was kept in the following frames corresponding to the same person or train, in this case. This particular version kept the identification number stored in case an object disappeared for a few frames and then reappeared. For the frames where this occurred, no estimated positions were generated. If the gap was larger, then the algorithm assigned a new number, i.e., it considered that it was a new object.

In other words, YOLOv7 has the capability of tracking objects throughout the image or frames. This is possible due to the Simple Online and Real-time Tracking (SORT) algorithm, which allows multiple-object tracking in real time in 2D. SORT is based on simple data association and state estimation techniques. When tracking is lost for a brief period of time it does not delete the label of a bounding box. However, this algorithm does not resolve problems with occlusion or re-entering objects, it just works as a method for tracking visible

objects. The key to the method is that data association (matching an object in the current frame to an object in the previous frame) is based on object appearance (e.g., as given by pixel values or an internal deep network representation).

YOLOv7 can be set up with different parameters when performing inference. This is a useful option for adjusting the weights, threshold, classes, augmented inferences, etc. In the near future, this work will be implemented in real time, and the YOLOv7 algorithm will be an appropriate option for detection due to its efficient use of convolutional neural networks. Detection performed with a deep learning approach is becoming faster with respect to algorithms that use regions of interest (ROIs) or multiples steps [8]. The YOLOv7 algorithm steps are: 1.- The input can be an image or video; if the input is a video it is separated into frames. Each frame is divided into cells, or grid divisions, a process that helps to better predict the object or class in the frame. 2.- Once the cells are generated, prediction is performed on each cell, generating a number of partially overlapping bounding boxes within the cells for each object. 3.- The model assigns a confidence value to each one of the bounding boxes. 4.- As the inference process generates multiple bounding boxes for the same object, a “Non Maximum Suppression” algorithm is applied, leaving the one with the highest confidence value, per object, and deleting the rest. 5.- The sizes of bounding boxes are converted into real values of the image (pixels), and then the algorithm filters the bounding boxes with a confidence threshold; see Figure 2.

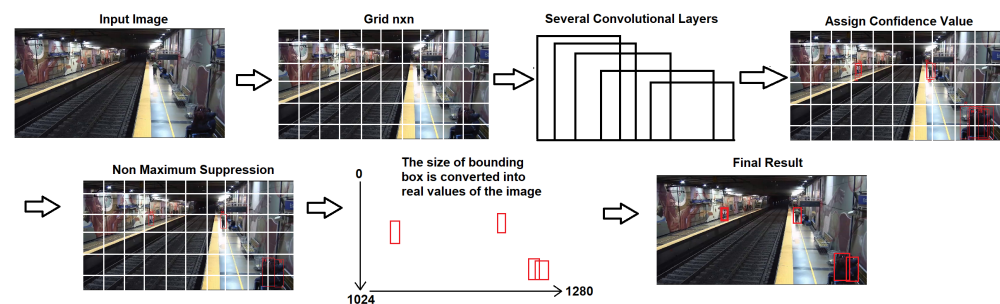


Figure 2. Steps carried out by YOLOv7.

## 2.2. Camera Calibration

One of the main steps in obtaining useful data is to convert the data from the camera's 2D image plane (given in pixels) to data in the real world in 3D, in distance units. The process of determining the conversion parameters is called camera calibration. Consider the geometry detailed in Figure 3.

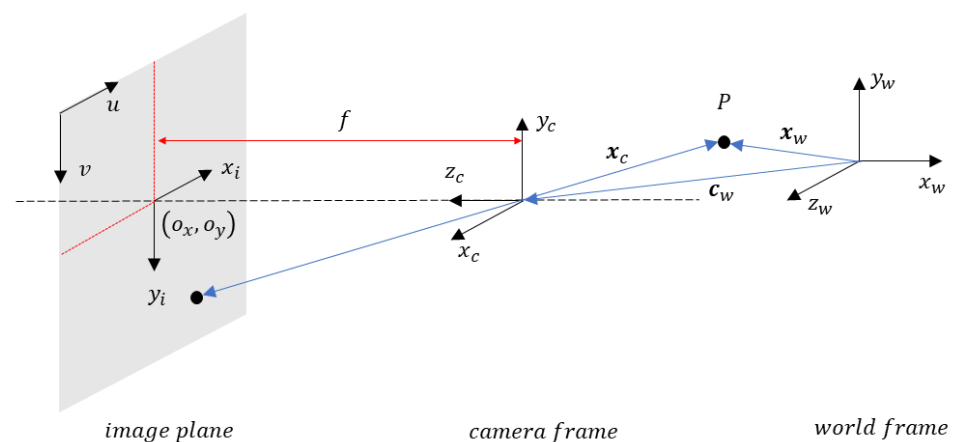


Figure 3. Camera calibration geometry.

It defines the geometry relating a point in three coordinate frames: a world coordinate frame, the camera coordinate frame, and the image plane coordinate plane. The 3D transformation from world to camera coordinate frames requires the camera translation

vector and orientation matrix, with respect to the world. Going from camera to image plane coordinate frames, a 3D-to-2D transformation, requires the focal length of the camera and the principal point.

### 2.2.1. Forward Imaging Modeling

It consists in converting a point from the world coordinates to the image plane coordinates. Consider a point P located in world coordinates at the vector  $\mathbf{x}_w = [x_w, y_w, z_w]^T$ . In the camera frame, it becomes:

$$\mathbf{x}_c = \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \mathbf{R}(\mathbf{x}_w - \mathbf{c}_w) = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix} \quad (1)$$

with  $\mathbf{R}$  the rotation matrix,  $\mathbf{c}_w$  the position vector, and  $-\mathbf{R}\mathbf{c}_w = [t_x, t_y, t_z]^T$  the translation vector of the camera with respect to the world coordinates. The two parameters  $(\mathbf{R}, \mathbf{c}_w)$  represent the extrinsic camera parameters.

On the other hand, applying the perspective projection principle allows the conversion from  $\mathbf{x}_c$  to image coordinates. From the geometry between the point and the image plane,

$$x_i = f \frac{x_c}{z_c}, y_i = f \frac{y_c}{z_c} \quad (2)$$

with  $f$  the focal length. Further conversion from image plane to pixel coordinates is done by

$$\begin{aligned} u &= m_x x_i + o_x = m_x f \frac{x_c}{z_c} + o_x = f_x \frac{x_c}{z_c} + o_x \\ v &= m_y y_i + o_y = m_y f \frac{y_c}{z_c} + o_y = f_y \frac{y_c}{z_c} + o_y \end{aligned} \quad (3)$$

with  $m_x$  and  $m_y$  the pixel densities per axis, and  $(o_x, o_y)$  the principal point. The four parameters  $(f_x, f_y, o_x, o_y)$  represent the intrinsic parameters of the camera.

### 2.2.2. Homogeneous Coordinates

To simplify the calculations, a linear model was derived by converting the previous equations to homogeneous coordinates as follows; see [17]. For the camera-to-image-plane conversions:

$$\begin{bmatrix} z_c u \\ z_c v \\ z_c \end{bmatrix} = \begin{bmatrix} f_x x_c + z_c o_x \\ f_y y_c + z_c o_y \\ z_c \end{bmatrix} = \begin{bmatrix} f_x & 0 & o_x & 0 \\ 0 & f_y & o_y & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \quad (4)$$

and for the world-to-camera conversion:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (5)$$

and from the world to image plane

$$\begin{bmatrix} z_c u \\ z_c v \\ z_c \end{bmatrix} = \begin{bmatrix} f_x x_c + z_c o_x \\ f_y y_c + z_c o_y \\ z_c \end{bmatrix} = \mathbf{P} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (6)$$

with the projection matrix:

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & p_{13} & p_{14} \\ p_{21} & p_{22} & p_{23} & p_{24} \\ p_{31} & p_{32} & p_{33} & p_{34} \end{bmatrix} \quad (7)$$

### 2.2.3. Projection Matrix Estimation

The projection matrix can be estimated from a set of 3D points; see [18]. These points can be randomly chosen and should be spread out through the volumetric space of interest. Assuming  $i = 1 \dots n$  3D points  $(x_w^i, y_w^i, z_w^i)$  and their corresponding pixel coordinates  $(u^i, v^i)$ , the following matrix equation is established:

$$\begin{bmatrix} x_w^1 & y_w^1 & z_w^1 & 1 & 0 & 0 & 0 & 0 & -u_1 x_w^1 & -u_1 y_w^1 & -u_1 z_w^1 & -u_1 \\ 0 & 0 & 0 & 0 & x_w^1 & y_w^1 & z_w^1 & 1 & -v_1 x_w^1 & -v_1 y_w^1 & -v_1 z_w^1 & -v_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_w^n & y_w^n & z_w^n & 1 & 0 & 0 & 0 & 0 & -u_n x_w^n & -u_n y_w^n & -u_n z_w^n & -u_n \\ 0 & 0 & 0 & 0 & x_w^n & y_w^n & z_w^n & 1 & -v_n x_w^n & -v_n y_w^n & -v_n z_w^n & -v_n \end{bmatrix} \begin{bmatrix} p_{11} \\ p_{12} \\ \vdots \\ p_{33} \\ p_{34} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad (8)$$

Equation (8), expressed in compact form,  $\mathbf{A}\mathbf{p} = \mathbf{0}$ , can be solved by minimizing the loss function  $L(\mathbf{p}, \lambda) = \mathbf{p}^T \mathbf{A}^T \mathbf{A} \mathbf{p} - \lambda(\mathbf{p}^T \mathbf{p} - 1)$  by solving the eigenvalue problem  $\mathbf{A}^T \mathbf{A} \mathbf{p} = \lambda \mathbf{p}$ . The vector  $\mathbf{p}$  corresponds to the eigenvector associated to the smallest eigenvalue.

### 2.2.4. Inverse Estimation

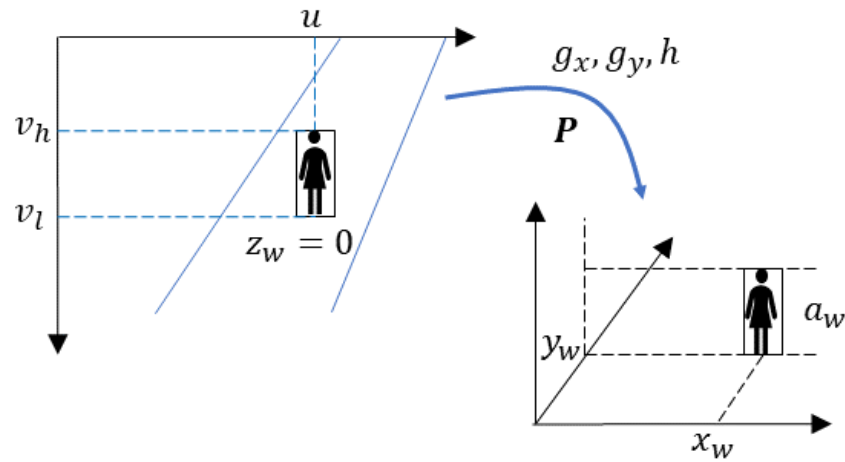
The forward imaging estimation suggests a mechanism to implement an inverse method for obtaining world coordinate points from image plane coordinates; see [19] for a similar approach. A truly 2D-to-3D transition is unambiguously achieved by the operation of two or more cameras with overlapping images. The use of a single camera imposes a limitation in the number of components to be converted. In this work, it was assumed that one of the components of a 3D point was known, i.e., the vertical component, as explained below. This assumption rendered the inverse problem feasible.

For a world point  $(x_w, y_w, z_w)$ , it is clear from (6) and (7), that by assuming that the component  $z_w$  is known, together with a matrix  $\mathbf{P}$  and its pixel coordinates  $(u, v)$ , the components  $x_w$  and  $y_w$  are uniquely determined as follows:

$$\begin{aligned} x_w &= g_x(\mathbf{P}, u, v_l, z_w) \\ y_w &= g_y(\mathbf{P}, u, v_l, z_w) \\ a_w &= h(\mathbf{P}, v_h, x_w, y_w) \end{aligned} \quad (9)$$

The idea is sketched out in Figure 4. Functions  $g_x$ ,  $g_y$ , and  $h$  carry out an inverse calculation based on (6) of the location of the person in the image plane in pixels, to the real world in world units (meters). In particular, functions  $g_x$  and  $g_y$ , using the data set  $\{\mathbf{P}, u, v_l, z_w\}$ , obtain  $x_w$  and  $y_w$ . On the other hand,  $h$  computes the height  $a_w$ , using the data  $\{\mathbf{P}, v_h, x_w, y_w\}$ , which include the pixel coordinate of the upper edge of the bounding box  $v_h$  and the already estimated position  $(x_w, y_w)$ , in world coordinates.

The main objective of the present work was to determine pedestrian parameters, such as position, velocity, and spatial density, inside the train stations. Given the horizontally leveled floors of the stations under study, the assumption of a fixed and known value for the component  $z_w$ , in this case the vertical component, was reasonable and did not involve an arbitrary approximation. This was supported by the fact that in general conditions, the bounding boxes normally go from the floor up, so using a point from the bottom edge is safe as its vertical component will be practically at the floor with a known value.



**Figure 4.** Inverse estimation.

### 2.3. Missing Data Estimation and Smoothing

Once the original data were converted from pixel coordinates to world coordinates, further processing was required. The method used for detection and tracking that obtains the bounding boxes per person per frame, has, from time to time, gaps for a number of consecutive frames with no information. If a person for some reason becomes not clearly visible for a short period of time, due to temporary occlusion with other people for example, the method manages to keep the same person linked before and after the gap but does not produce a bounding box.

To address this problem an estimation filter was devised. Due to the low complexity of the variables under study, i.e., position and velocity, and the lack of noise figures associated to the detection/identification process, an alpha-beta filter was used for the gap estimation. This filter also helped in smoothing the data, especially the estimated velocity. For the construction of the filter, the dynamics assumed zero acceleration, hence a constant velocity.

Given a set of points  $(x_w^j, y_w^j)$  associated to the same person  $j$ , for each frame  $k$ , with potential gaps in a few consecutive frames, the following prediction stage of an alpha-beta filter was constructed

$$\begin{bmatrix} p_{\hat{x}}^j(k|k-1) \\ p_{\hat{y}}^j(k|k-1) \\ v_{\hat{x}}^j(k|k-1) \\ v_{\hat{y}}^j(k|k-1) \end{bmatrix} = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} p_{\hat{x}}^j(k-1|k-1) \\ p_{\hat{y}}^j(k-1|k-1) \\ v_{\hat{x}}^j(k-1|k-1) \\ v_{\hat{y}}^j(k-1|k-1) \end{bmatrix} \quad (10)$$

where  $dt$  is the time interval between frames, a fixed quantity. Measured data consisted of positions  $(p_x, p_y) = (x_w^j, y_w^j)$ , and estimated/smoothed variables were position  $(p_{\hat{x}}, p_{\hat{y}})$  and velocity  $(v_{\hat{x}}, v_{\hat{y}})$ . After a set of measurements, the update stage became

$$\begin{bmatrix} p_{\hat{x}}^j(k|k) \\ p_{\hat{y}}^j(k|k) \\ v_{\hat{x}}^j(k|k) \\ v_{\hat{y}}^j(k|k) \end{bmatrix} = \begin{bmatrix} p_{\hat{x}}^j(k|k-1) \\ p_{\hat{y}}^j(k|k-1) \\ v_{\hat{x}}^j(k|k-1) \\ v_{\hat{y}}^j(k|k-1) \end{bmatrix} + \delta \begin{bmatrix} \alpha r_{\hat{x}}^j(k) \\ \alpha r_{\hat{y}}^j(k) \\ \beta r_{\hat{x}}^j(k) \\ \beta r_{\hat{y}}^j(k) \end{bmatrix} \quad (11)$$

with  $r_{\hat{x}}(k) = p_x(k) - p_{\hat{x}}(k|k-1)$  and  $r_{\hat{y}}(k) = p_y(k) - p_{\hat{y}}(k|k-1)$  the a priori estimation errors,  $\alpha$  and  $\beta$  the filter parameters to obtain by fine tuning, and  $\delta = \{0, 1\}$  a variable to suppress the update and only allow the prediction step, when no measurements are available during the gaps in the data.

#### 2.4. Ground-Truth Comparison

There are several sources of error that can potentially add up to produce biased or incorrect results. These include detection and tracking errors happening at the level of the video processing of the bounding boxes. Another source of errors can arise in the camera calibration (errors in measuring the calibration points in the field and in the extraction of their corresponding image coordinates), as this is a manual and graphical process. The latter error source can be worsened by the geometry of the station platforms, depending on the chosen camera and its orientation. When longer physical depths are involved, distant features in the image tend to get closer and be affected by the working resolution.

A preliminary assessment was carried out on the accuracy of the estimations in the two categories of errors by comparing their outcomes with a ground truth, i.e., data extracted manually from the video. First, a comparison was made between bounding boxes produced by YOLOv7 and those bounding boxes extracted manually frame by frame. Intersection over Union (IoU) gives a pertinent indicator of the goodness of the automated process, by computing the ratio of intersection and union areas between a prediction and its ground truth. Errors encountered here could include wrong classification, missed detections, and deformed bounding boxes. Secondly, the assessment was based on analyzing the resulting positions and speeds. This was achieved by contrasting the estimated values outputted by the algorithm with the values manually extracted after analyzing the video and in-site measurements. The evaluation of position accuracy was more easily performed by analyzing a person that remained in a fixed position, sitting down for example, and for speed, by choosing somebody that had a uniform motion.

#### 2.5. Activity Recognition

For the recognition of activity, there are several important aspects to take into account, for example, that this is a real scenario with multiple people in the same scene, that there is occlusion with objects and occlusion between people, etc. Therefore, an activity recognition method that works in this type of scenario was necessary.

It has been shown that the use of skeletons to estimate the pose of people delivers superior performance compared to other approaches for the recognition of activities, so in this work, the methodology carried out in [15] was used, as illustrated in Figure 5. The process began with the acquisition of the data (the analysis was carried out frame by frame), followed by the extraction of the poses of the people in the scene using AlphaPose [20], which generated skeletons with 17 key points (corresponding to the joints of the person's body) each with the x and y information of the image plane plus a detection confidence score. Once the features were extracted, a feature vector was formed, so that for each skeleton, there was a vector of 51 features ( $16 \times 3$ ) plus a label with the activity performed by the person in the scene (to carry out supervised learning). Finally, a machine learning model that predicts the activity performed by each individual was used. The workflow used for this work is shown in Figure 6, using a Random Forest (RF) model pre-trained as per the work reported in [15]. That model was trained with 12 activities, 5 falling activities (falling forward using hands, falling forward using knees, falling backward, falling sideways, and falling sitting on a flat chair) and 7 daily activities (walking, standing, sitting, picking up an object, jumping, lying down, and kneeling). In the station videos, we expected the model to mainly predict daily activities such as walking, standing, and sitting.

The main challenge was that this RF model was trained with a set of images with simulated activities, carried out by a single person in a controlled environment without any type of occlusion. However, the images of the station had multiple people, in a real uncontrolled setting and with occlusions.

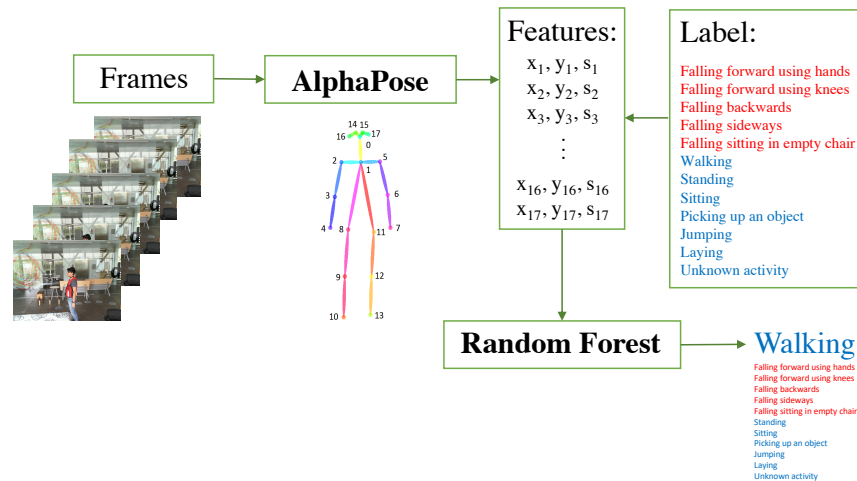


Figure 5. Workflow used in [15] for activity recognition.

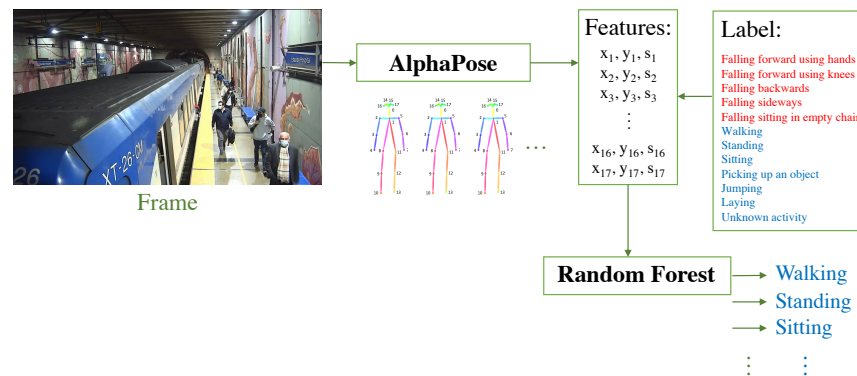


Figure 6. Workflow for activity recognition in this work.

### 3. Experimental Results and Preliminary Behavior Analysis

This section presents the results of the application of the algorithms detailed above to three different videos obtained from an underground train station. The process was carried out offline, but it is interesting to mention that a dedicated GPU computer would perform the same calculations in real time.

#### 3.1. First Case: Limited Number of Persons

This video was selected as a proof of concept, as it contained a reduced number of persons, allowing for a simpler analysis and confirmation of the results. It had a length of 11 s, and contained five people, two of them sitting down.

The application of the detection and tracking algorithms from YOLOv7 [8,9] generates a data set of the trajectory in pixel coordinates  $(u, v)$  with the trajectories for each detected person. Figure 7 shows a particular frame of the video with one person walking and two sitting down.

For the camera calibration,  $n = 8$  points were randomly selected to cover the entire volume of interest, with known world frame coordinates  $(x_w^i, y_w^i, z_w^i)$  with respect to a chosen reference point  $RP$ , and their corresponding image coordinates  $(u^i, v^i)$ , with  $i = 1 \dots n$ . The points used are shown on a snapshot of the camera in Figure 8 and Table 1. Points were tape-measured on site. The application of (8) gave the projection matrix  $P$ .

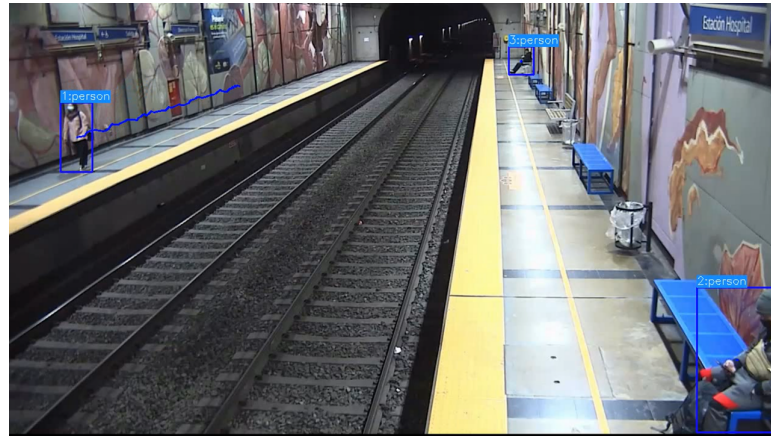


Figure 7. Frame example showing people in the station.

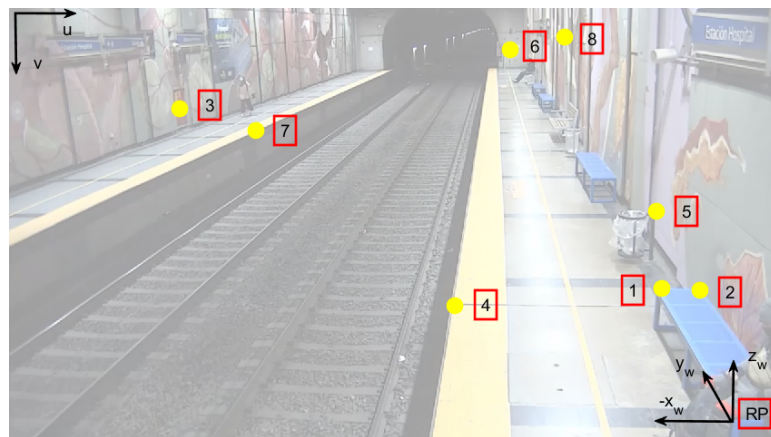
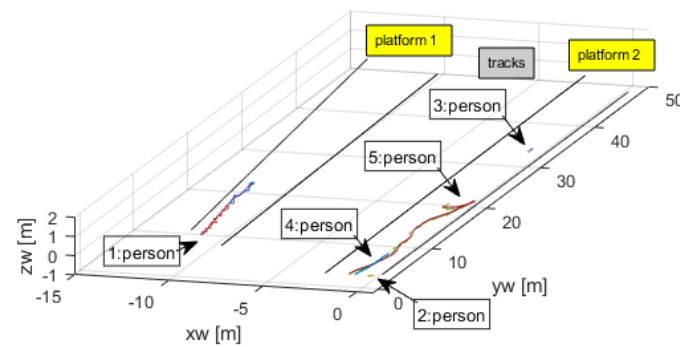


Figure 8. Camera snapshot of the station with points for calibration.

Table 1. Calibration points.

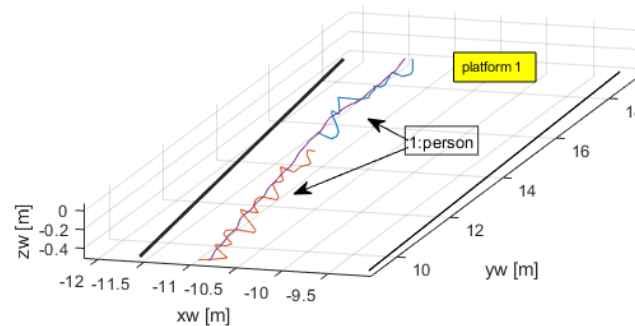
| Calibration Points | Component $x_w$ [m] | Component $y_w$ [m] | Component $z_w$ [m] |
|--------------------|---------------------|---------------------|---------------------|
| 1                  | 0.00                | 2.45                | 0.00                |
| 2                  | 0.45                | 2.45                | 0.00                |
| 3                  | −12.19              | 17.35               | 0.31                |
| 4                  | −2.10               | 3.00                | −0.45               |
| 5                  | 0.43                | 4.55                | 0.25                |
| 6                  | −0.62               | 47.30               | 1.05                |
| 7                  | −9.40               | 17.35               | −0.45               |
| 8                  | 0.30                | 17.44               | 2.12                |

Having the camera calibrated, it was then possible to apply (9) to obtain the coordinates  $(x_w^i, y_w^i)$  for a known third component, in this case  $z_w^i = -0.45$  m, which represented the vertical location of the platform floor with respect to the reference point. Figure 9 shows the world coordinate trajectories of the five persons in this video.



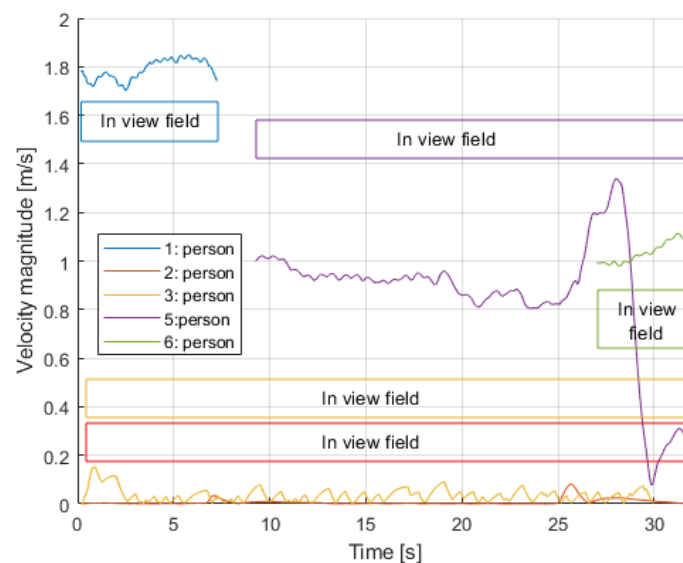
**Figure 9.** Trajectories of the tracked persons.

One of the features of the open-source code for tracking and detection is that it is able to keep the correct identification of a person even if it disappears from view for a few consecutive frames. This was the case of person 1 that was absent for some frames, but whose tracking number was correctly reassigned after reappearing. This is clearly seen in Figure 10, a zoom of Figure 9.



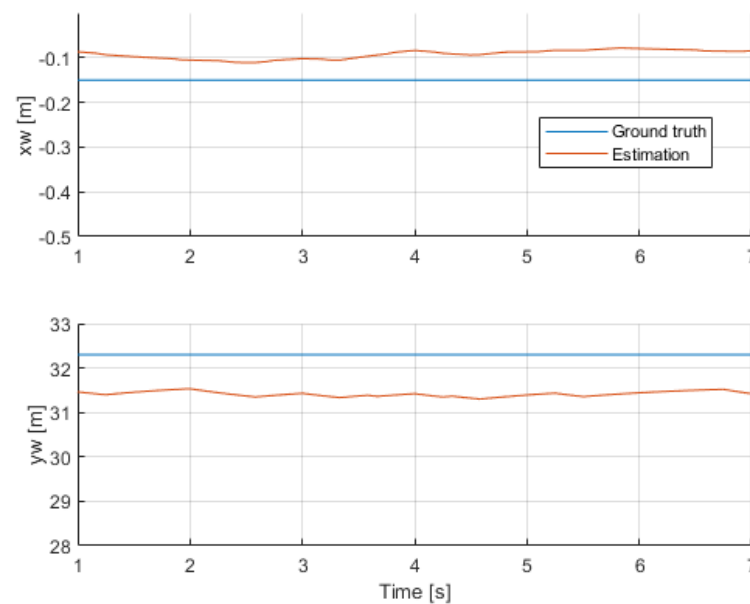
**Figure 10.** Tracking number was kept despite short absence of person.

The gap in tracking is apparent from the figure. This was fixed by applying an interpolating/smoothing filter, in this case an alpha-beta filter, which filled in between the two interconnected trajectories and also applied a smoothing effect for less noisy variables, specially velocity. Figure 11 shows the smoothing effect on the estimated velocities.

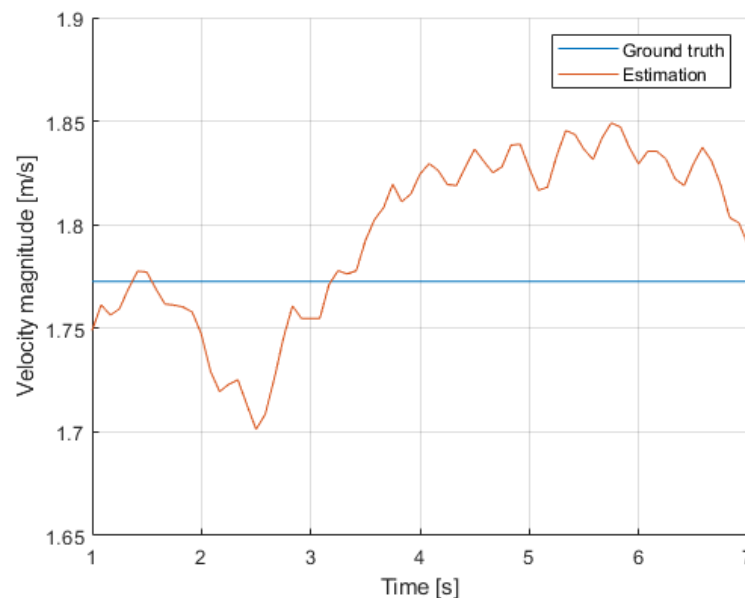


**Figure 11.** Velocity magnitude per person.

This video was also selected for a preliminary assessment of the accuracy of the overall estimations. It had both a person sitting down for the entire video (person 3), and a person that had a very stable motion in a portion of the video (person 1). Figures 12 and 13 show the comparisons. In terms of position, component  $xw$  showed a small error around 0.1 m, and component  $yw$  had one closer to 1 meter. The estimation of speed had a better agreement with its the ground-truth value. These results served as a proof of concept and validation of the method, but further fine-tuning was required to mitigate the effect of the different sources of errors.



**Figure 12.** Position comparison with respect to the ground truth (person 3).

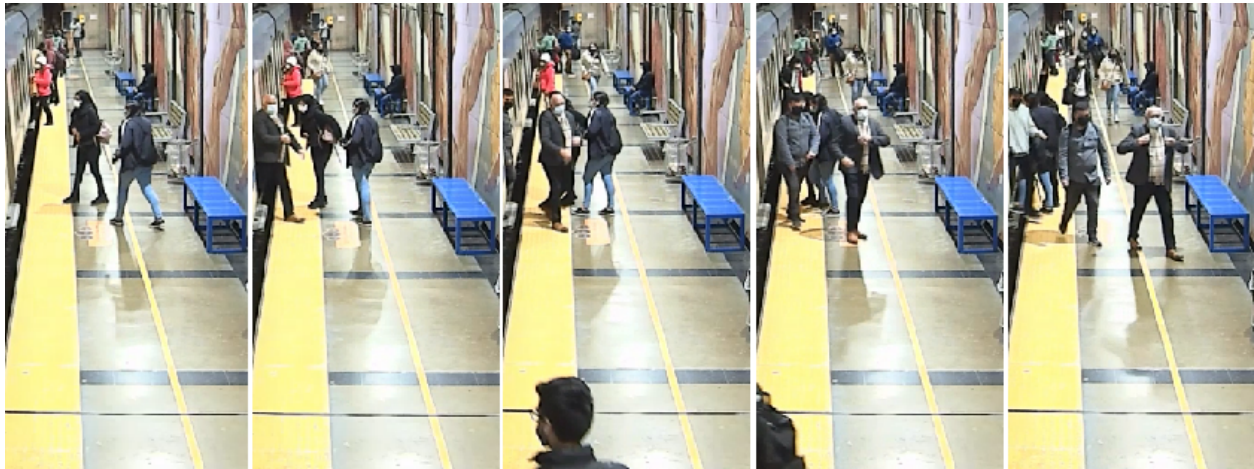


**Figure 13.** Speed comparison with respect to the ground truth (person 1).

### 3.2. Second Case: A Larger Number of Persons

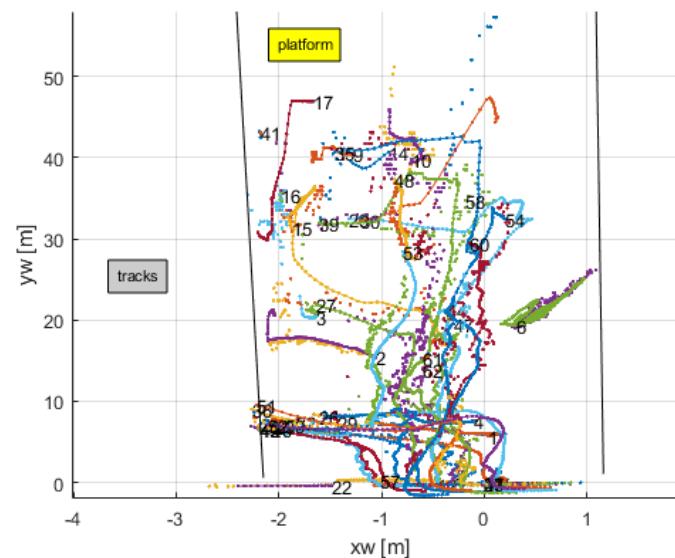
As a second step, a video with a considerably larger number of persons was used. The video included the presence of a train for the entire duration of the video, so only one platform was visible. Figure 14 shows a sequence of five snapshots equally time-spaced, covering 32 persons, taken from the same camera as the first video on a different day.

Camera calibration was performed with a new set of points, due to the presence of the train, so all points were chosen from the right side of the station.



**Figure 14.** Sequence of snapshots of the second video.

Figure 15 shows the trajectories of the 62 persons during the video. Black lines demarcate the platform. It includes the raw data after the 2D-to-3D conversion and the filtered data. As before, the effect of the filter is clear, smoothing the motions and penalizing non-realistic outliers.



**Figure 15.** Tracked trajectories of persons.

A zoom at the location of the second door (in Figures 14 and 15) shows the trajectory of some of the people as they approach to enter or leave the train, see Figure 16. Black lines demarcate the platform. Some interesting comments can be made in this case. In the upper sub-figure, it can be seen that persons 1 and 4, as they approached to board the train, suffered occlusion due to people leaving the train, and their track was lost and briefly reassigned with different identification numbers when they became visible again. On the other sub-figure, persons 20, 24, 33, and 40 left the train. Person 33 showed a departure from the trajectories the other persons were following while on the platform by walking temporarily in the opposite direction. This effect was not seen in the video, and person 33 walked along the same general path as the others. The error was due to a temporal miscalculation of its bounding box as its lower part was again occluded by other people; see Figure 17.

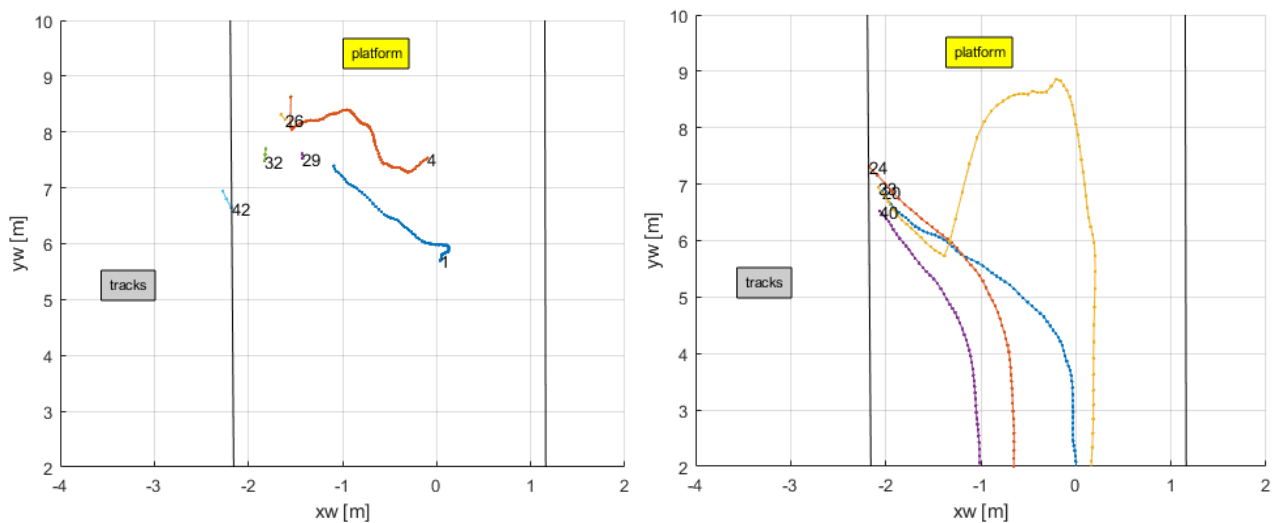


Figure 16. Zoom of second door in Figure 15.

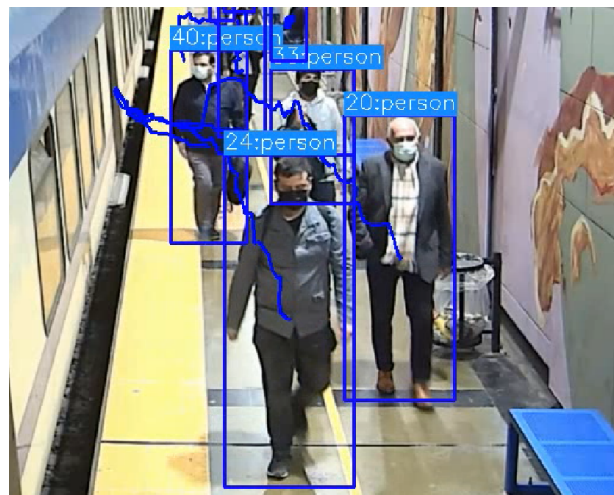


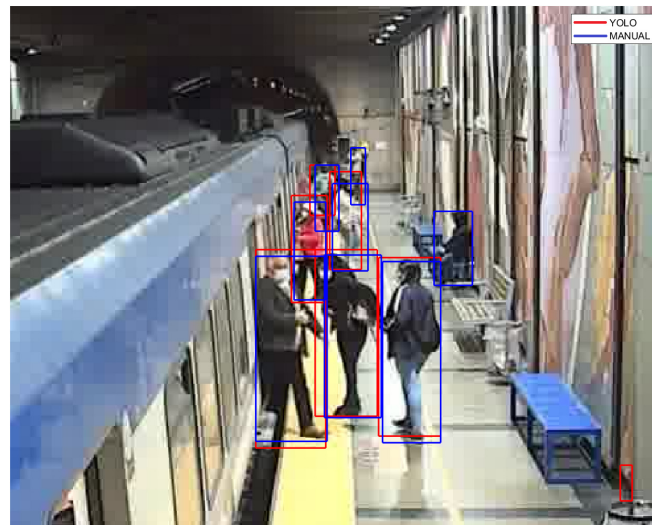
Figure 17. Error in bounding box of person 33.

Here, the assumption that the lower edge of the bounding box was at a known vertical coordinate was no longer appropriate, and the conversion from 2D to 3D suffered a projection error.

This video was also used for an evaluation of the performance of YOLOv7. Bounding boxes generated by the method were contrasted with bounding boxes manually obtained in a subset of consecutive frames. Table 2 summarizes the average of the outcomes of the two chosen criteria, for each frame: a comparison of the number of bounding boxes correctly assigned to the class of persons is included in the first row; for correctly assigned pairs of bounding boxes, the calculation of the IoU is shown in the second row. Figure 18 zooms in on a frame with both sets of bounding boxes, where the differences can be seen more clearly.

Table 2. YOLOv7 performance.

| Criterion                        | Outcome (%) |
|----------------------------------|-------------|
| Number of correct bounding boxes | 81.5        |
| IoU                              | 77.6        |



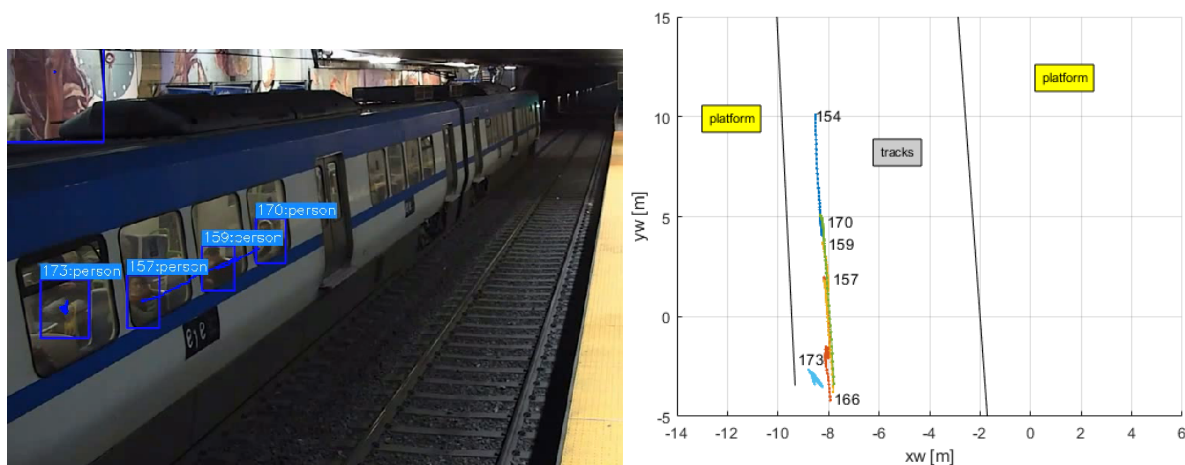
**Figure 18.** Comparison of bounding boxes.

### 3.3. Third Case: Larger-Duration Video

The final video was one of a much larger duration, around 11 min. For this reason, some trains were seen in and out of the station, and due to the orientation of the camera, in some frames, people inside the trains were detected. A total of 1081 detections were processed.

The left sub-figure in Figure 19 shows a snapshot when the train was stopped at the station, and the trajectories of some people onboard the train were detected and tracked. Some of their trajectories are shown in the right sub-figure. The black lines demarcate the tracks area.

This gives the possibility of detecting people inside the tracks area when there is no train present, which would require an immediate reaction to control the situation and avoid any potential accident.



**Figure 19.** Detection of people inside the train.

Figure 20 shows the detection of the train. As explained earlier, YOLOv7 was set to detect two classes: people and trains. The information of the train being present at the station allowed us to mask the people detected inside the train, and to focus on people detected when there is no train.



Figure 20. Detection of the train.

Another variable of interest was the spatial density. The analysis was conducted by subdividing the platforms into a fine grid and counted the number of persons per divisions as a function of time. Figure 21 shows the calculation of the density at  $t = 664$  s. The platform area was divided in squares of 1 square meter each. The left sub-figure shows a snapshot with the bounding boxes and part of the trajectories, and the right sub-figure shows the respective density as a number of persons sharing a division (colors do not provide extra information).

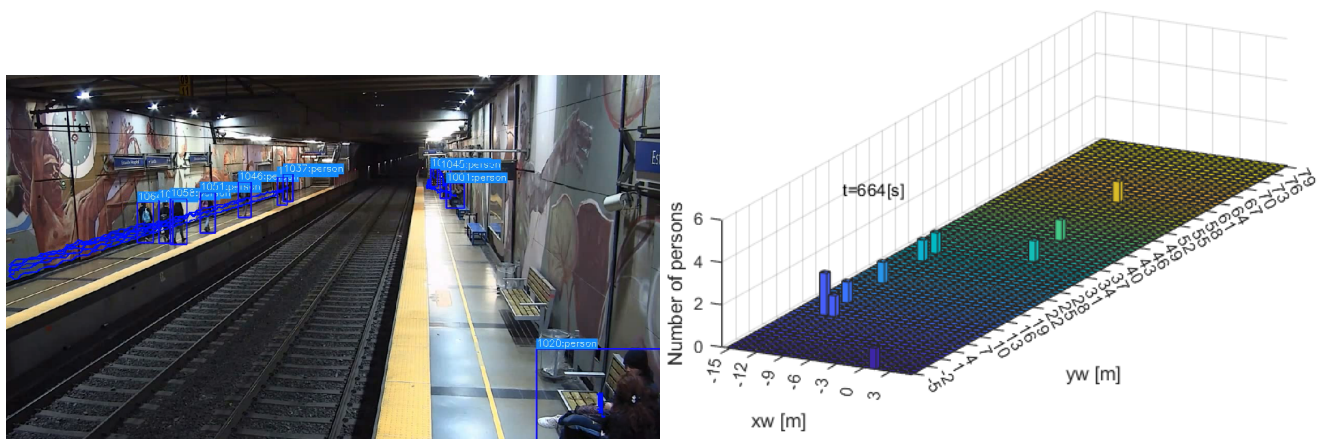


Figure 21. Spatial density at  $t = 664$  s, using a division of 1 square meter.

### 3.4. Activity Recognition

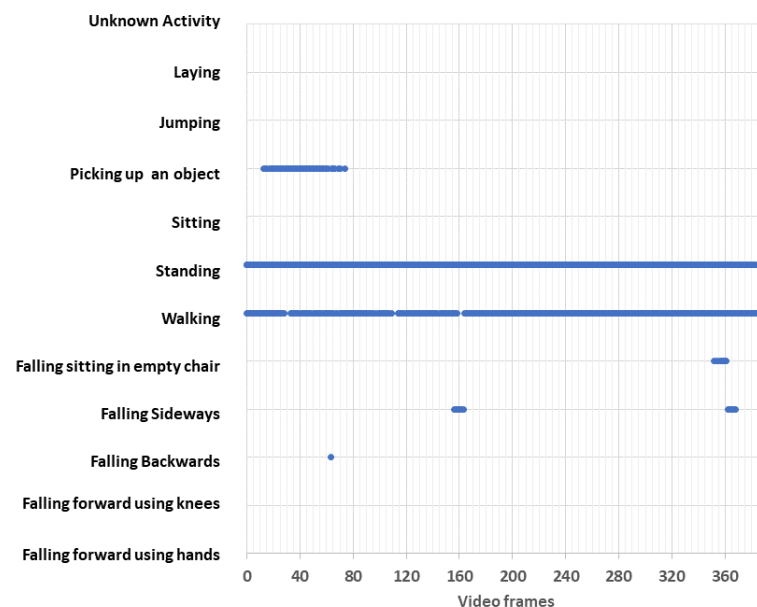
The last experiment consisted in recognizing the activities carried out by people in the subway station. This was achieved using the video of the first case. As mentioned in the previous section, the method consisted in passing the video frame by frame through the pre-trained Random Forest (RF) machine scanning model and predicting the activity performed by each person in the image.

Taking into account that the RF model only detected the activity of one skeleton at a time, the recognition of the activity in this work was carried out for each person in the image.

To carry out the tests, a video of the subway was used with 383 images in total, each with three or more people. All the images contained two people sitting, waiting for the subway, and at least one person walking.

Figure 22 shows the prediction made by the RF model. The graph shows the activity performed by people in each frame of the video. As can be seen, the model incorrectly predicted that in some images a person fell, and that in some images, a person was holding

an object. Nevertheless, the RF model was able to correctly predict that there were one or more people sitting in all the images, and it was also able to predict correctly that in most of the images, there were one or more people walking.



**Figure 22.** Activity recognition using the RF model.

Finally, the model incorrectly predicted three different types of falls in a total of 24 images and wrongly predicted that in 54 images, a person picked up an object. These figures are shown in Table 3.

**Table 3.** Number of frames containing each activity.

|                                | Total Frames   |          |
|--------------------------------|----------------|----------|
|                                | Original Video | RF Model |
| Falling forward using hands    | 0              | 0        |
| Falling forward using knees    | 0              | 0        |
| Falling backwards              | 0              | 1        |
| Falling sideways               | 0              | 15       |
| Falling sitting in empty chair | 0              | 8        |
| Walking                        | 383            | 361      |
| Standing                       | 0              | 0        |
| Picking up an object           | 0              | 54       |
| Jumping                        | 0              | 0        |
| Laying                         | 0              | 0        |
| Kneeling                       | 0              | 0        |

Although the approach delivered a total of 78 “False Alarms”, the model was quite accurate at recognizing people’s activity when they were walking. It might be possible to increase the performance of the proposed model and reduce the false alarms by re-training the model with other datasets in uncontrolled, real, and occluded environments, including different camera orientations and positions, or by using other types of machine learning or deep learning models.

### 3.5. Behavioral Analysis

#### 3.5.1. Combined Kinematics and Activity Approach

From the presented results, some preliminary analysis can be conducted regarding the behavior of passengers during their presence at the station. Both methods used, i.e., the extraction of kinematic information and the classification of the specific activity, are seen as complementary, enhancing the overall analysis. Using the video of the first case, analyzing the velocities of all passengers throughout the duration of the video, see Figure 11, we can infer the following classification in Table 4:

**Table 4.** Inferred classification from the estimated velocities.

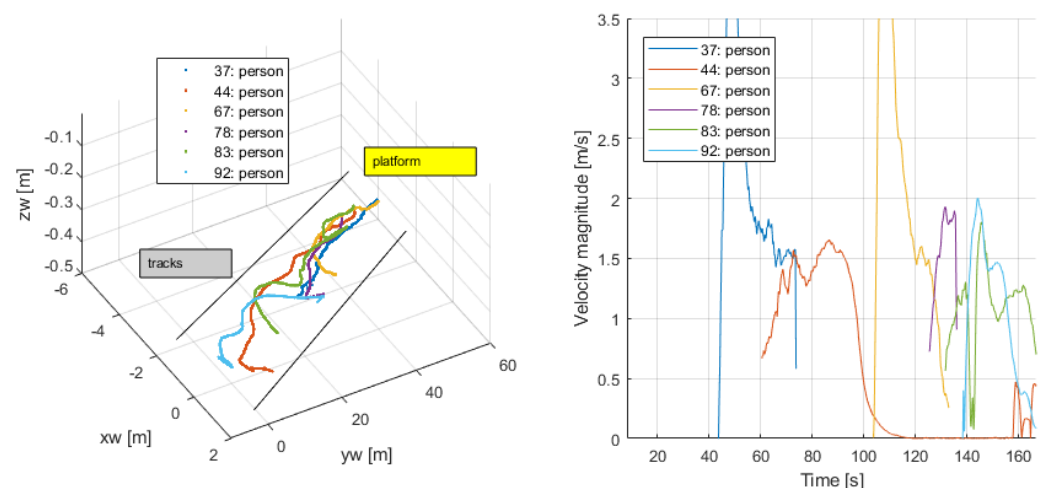
| Person   | Total Frames |     |     |     |    |
|----------|--------------|-----|-----|-----|----|
|          | 1            | 2   | 3   | 5   | 6  |
| Walking  | 85           | 0   | 0   | 271 | 60 |
| Standing | 0            | 381 | 381 | 0   |    |

This result complements the classification shown in Table 3, indicating that there were people sitting down the entire video (persons 2 and 3), and that there were people walking (persons 1, 5, and 6) in different but overlapping sections of the video, adding up to a value close to the one estimated in Table 3.

#### 3.5.2. Kinematics Approach

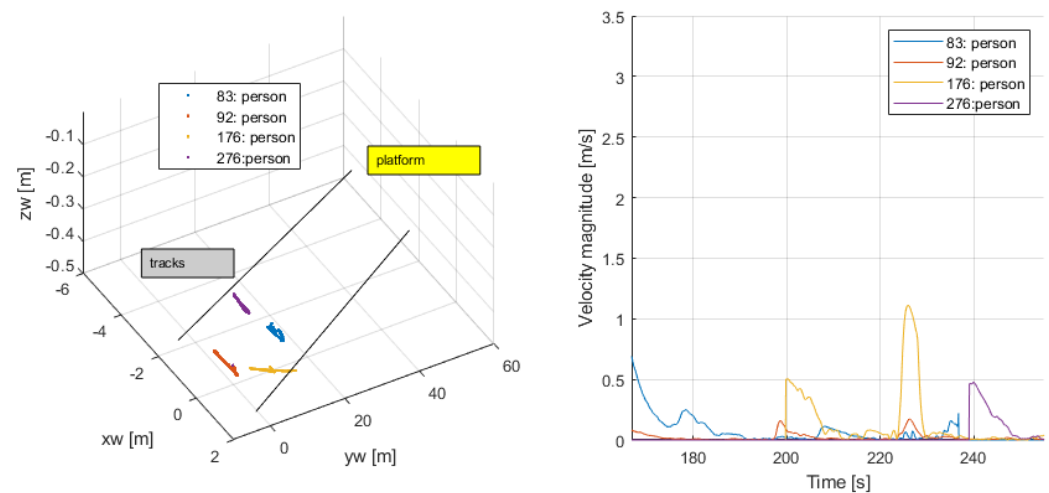
Part of the video from the third case was used for this analysis (the first 308 s). Four stages were singled out, choosing representative persons for each one. Their estimated behavior, including walking, standing or barely moving, boarding or leaving the train, can be inferred from a combination of their speeds and their estimated trajectories.

- Walking onto the platform (0 – 160 s): Persons {37, 44, 67, 78, 83, 92} entered the platform and reached a point and mainly stopped. Figure 23 shows their tracking history and speeds, from where this activity can be inferred. Their average speed was different from zero, indicating their motion until they settled in a fixed position.



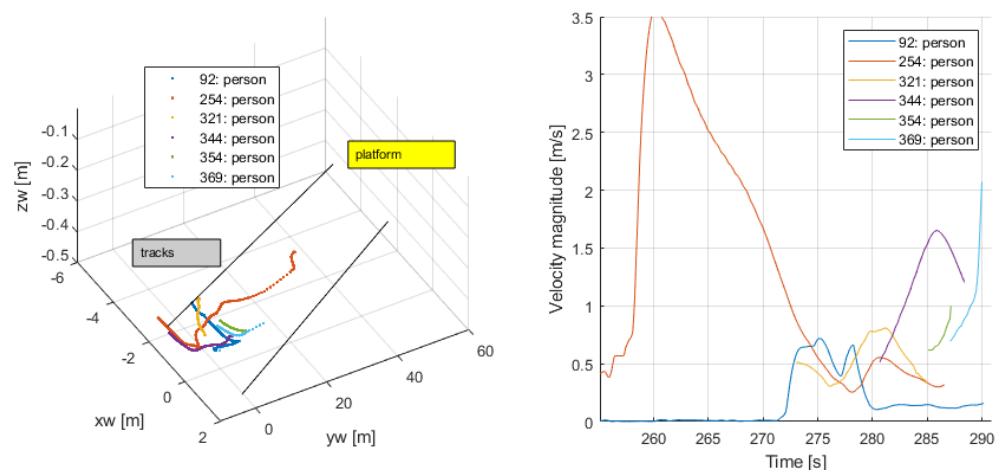
**Figure 23.** Entering the platform.

- Waiting for the train arrival (161–255 s): Persons {83, 92, 176, 276}, some of which may have entered the platform before the beginning of the video, remained with almost no motion waiting for the train, or briefly changed position; see Figure 24. Their average speed was close to zero, and their trajectories were closely centered at a fixed position.



**Figure 24.** Waiting for the train.

- Train approaching, boarding and alighting (256–290 s): Persons {92, 254, 321, 344} approached and entered the train through two different doors. Persons {354, 369} left the train; see Figure 25. Tracking for these last ones was available until they became visible. Their speeds clearly indicated their activity, as they changed abruptly from a value different from zero, either because the passengers appeared or disappeared from one frame to the next one.



**Figure 25.** Train approaching, boarding and alighting from the train.

- Leaving the platform (291–308 s): Persons 361, 372, after alighting from the train (from a door not in the visual field), walked out from the platform; see Figure 26. Their speed was consistent as they increased from lower values to higher ones reflecting their more stable motion towards the exit.

Figure 27 shows the average number of people per square meter on the platform, as people circulated.

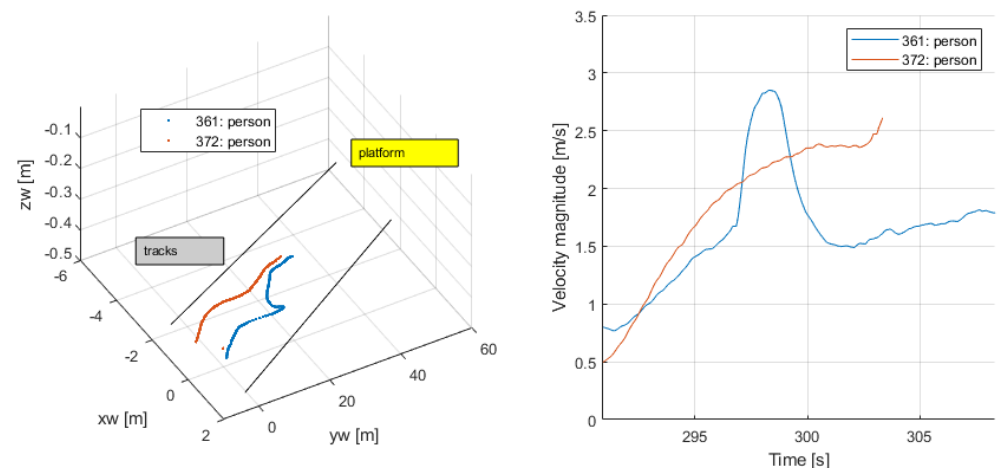


Figure 26. Leaving the platform.

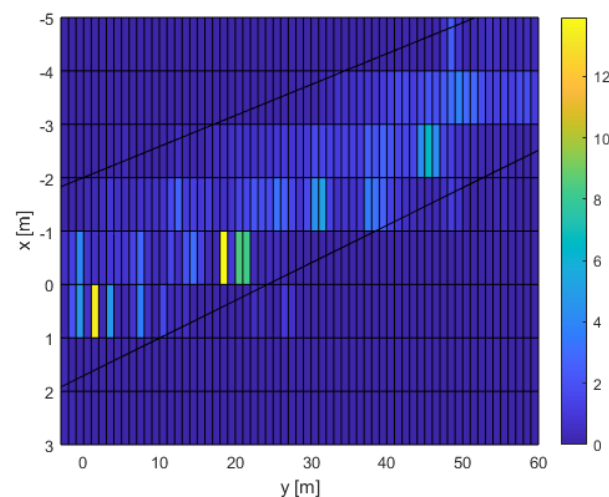


Figure 27. Average number of people per square meter.

#### 4. Conclusions

Knowledge of people's behavior in train stations is critical for any optimization of space design and services to be carried out. The article aimed to test the concept of extracting kinematic and pose information of passengers from cameras on a subway platform as they entered/exited trains and as they moved around. Two complementary methods were tested, both implemented on convolutional neural networks. One was based on a kinematic analysis, from where variables such as position, velocity, and spatial density were calculated; and the other used extracted key points from the body, from which specific attitudes or poses were estimated, especially walking. The results were promising in the sense that fusing both methods, it was possible to quantitatively and qualitatively extract information useful for transport management. The approach can be further improved, making it more robust, especially the activity recognition method, by performing a fine-tuning of the neural network models with new conditions, unforeseen situations, such as different stations, different cameras and their position and orientation, and changing light levels. As a proof of concept, the present work reached its objectives to create an accessible automatized technique with standard computational resources, delivering useful raw information for an in-depth analysis towards a better station design.

**Author Contributions:** Conceptualization, G.G. and S.S.; methodology, G.F.; software, G.G., N.L. and H.R.; validation, S.S.; writing—review and editing, S.A.V.; supervision, G.F. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was supported, in part, by the Chilean Research and Development Agency (ANID) under Project FONDECYT 1191188 and by Grant FONDEF ID22I10018, ANID.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data is contained within the article.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. National Research Council. *Highway Capacity Manual*; National Research Council: Washington, DC, USA, 2000; Volume 2.
2. National Academies of Sciences, Engineering, and Medicine. *Transit Capacity and Quality of Service Manual*; National Academies of Sciences, Engineering, and Medicine: Washington, DC, USA, 2003; Volume 42.
3. Basnak, P.; Giesen, R.; Muñoz, J.C. Estimation of crowding factors for public transport during the COVID-19 pandemic in Santiago, Chile. *Transp. Res. Part A Policy Pract.* **2022**, *159*, 140–156. [\[CrossRef\]](#) [\[PubMed\]](#)
4. Seriani, S.; Fernandes, V.A.; Moraga, P.; Cortes, F. Experimental Location of the Vertical Handrail to Improve the Accessibility of Wheelchair Passengers Boarding and Alighting at Metro Stations—A Pilot Study. *Sustainability* **2022**, *14*, 9224. [\[CrossRef\]](#)
5. Liu, Z.; Liu, Y.; Meng, Q.; Cheng, Q. A tailored machine learning approach for urban transport network flow estimation. *Transp. Res. Part C Emerg. Technol.* **2019**, *108*, 130–150. [\[CrossRef\]](#)
6. Calero, M.J.F.; Aldás, M.; Onofa, N.; Quinga, B. Detección de Peatones bajo oclusión parcial usando Inferencia Logística, HOG y SVM. *IEEE Lat. Am. Trans.* **2019**, *17*, 1552–1559.
7. Jeong, E.; You, S.I.; Lee, J.; Moon, D. Identifying the indoor space characteristics of an urban railway station based on pedestrian trajectory data. *J. Adv. Transp.* **2019**, 2019. [\[CrossRef\]](#)
8. Munawar, I. Yolov7-Object-Tracking. 2023. Available online: <https://github.com/RizwanMunawar/yolov7-object-tracking> (accessed on 10 May 2024).
9. Wang, C.Y.; Bochkovskiy, A.; Liao, H.Y.M. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv* **2022**, arXiv:2207.02696.
10. Bewley, A.; Ge, Z.; Ott, L.; Ramos, F.; Upcroft, B. Simple online and realtime tracking. In Proceedings of the 2016 IEEE International Conference on Image Processing, Phoenix, AZ, USA, 25–28 September 2016. [\[CrossRef\]](#)
11. Hermosilla, G.; Verdugo, J.; Farias, G.; Vera, E.; Pizarro, F.; Machuca, M. Face Recognition and Drunk Classification Using Infrared Face Images. *J. Sens.* **2018**, *2018*, 5813514. [\[CrossRef\]](#)
12. Chen, L.; Ma, N.; Wang, P.; Li, J.; Wang, P.; Pang, G.; Shi, X. Survey of pedestrian action recognition techniques for autonomous driving. *Tsinghua Sci. Technol.* **2020**, *25*, 458–470. [\[CrossRef\]](#)
13. Zheng, Z.; Wang, H.; Liu, W.; Peng, L. Toward real-time congestion measurement of passenger flow on platform screen doors based on surveillance videos analysis. *Phys. Stat. Mech. Its Appl.* **2023**, *612*, 128474. [\[CrossRef\]](#)
14. Lv, H.; Yan, H.; Liu, K.; Zhou, Z.; Jing, J. Yolov5-ac: Attention mechanism-based lightweight yolov5 for track pedestrian detection. *Sensors* **2022**, *22*, 5903. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Ramirez, H.; Velastin, S.A.; Meza, I.; Fabregas, E.; Makris, D.; Farias, G. Fall Detection and Activity Recognition Using Human Skeleton Features. *IEEE Access* **2021**, *9*, 33532–33542. [\[CrossRef\]](#)
16. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision—ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; pp. 740–755.
17. Zhang, Z. A flexible new technique for camera calibration. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1330–1334. [\[CrossRef\]](#)
18. Heikkila, J.; Silvén, O. A four-step camera calibration procedure with implicit image correction. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Juan, PR, USA, 17–19 June 1997; pp. 1106–1112.
19. Yuan, Z. Pedestrian Gait Analysis Based on Computer Vision Technology Application. Master’s Thesis, Imperial College and University College, London, UK, 2022.
20. Fang, H.S.; Xie, S.; Tai, Y.W.; Lu, C. Rmpe: Regional multi-person pose estimation. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2334–2343.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.