Advances in Science, Technology and Engineering Systems Journal
Vol. 7, No. 5, 113-128 (2022)
www.astesj.com
Special Issue on Innovation in Computing, Engineering Science & Technology

ASTES Journal
ISSN: 2415-6698

Automatic Counting Passenger System using Online Visual Appearance Multi-Object Tracking

Javier Calle[*], Itziar Sagastiberri, Mikel Aramburu, Santiago Cerezo, Jorge García

Fundación Vicomtech, Basque Research and Technology Alliance (BRTA), Donostia-San Sebastián, 20009, Spain

ARTICLE INFO

Article history:
Received: 31 August, 2022
Accepted: 08 October, 2022
Online: 25 October, 2022

Keywords:
People-counting
Multi-object tracking
Fisheye camera
Public transport
Visual appearance modelling

ABSTRACT

In recent years, people-counting problems have increased in popularity, especially in crowded indoor spaces, e.g., public transport. In peak hours, trains move significant numbers of passengers, producing delays and inconveniences for their users. Therefore, analysing how people use public transport is essential to solving this problem. The current analysis estimates how many people are inside a train station by using the number of people entering and leaving the ticket gates or estimating the train occupancy based on conventional CCTV cameras. However, this information is insufficient for knowing the train occupancy. The required data includes vehicle usage: how many people enter or leave a vehicle or which door is the most used. This paper presents a solution to the stated problem based on a multi-object tracker with a sequential visual appearance predictor and a line-based counting strategy to analyse each passenger's trajectory using an overhead fisheye camera. The camera selection inside the train was made after profoundly studying the railway environment. The method proposes a module to compute the total train occupancy. The solution is robust against occlusions thanks to the selected tracker and the fisheye camera field of view. This work shows a proof of concept dataset containing pseudo-real case scenarios of people's affluence in train doors recorded by fisheye cameras. Its purpose is to prove the system's functionality in these scenarios. The proposed approach achieved an overall accuracy of counting people getting on and off of 90.78% in the pseudo-real dataset, proving that this approach is valid.

1 Introduction

In the context of security surveillance, people counting is one of the most interesting tasks in analysing crowds. Although it seems like a simple task, when the crowd density is high, the number of occlusions increases the problem's complexity. During the last few years, this problem has risen significantly in the security field because it allows estimating the number of people in a room, thus limiting access to it. However, people-counting algorithms are also used in outside environments to estimate the number of pedestrians in a specific area.

There are two main circumstances where it is practical to count people in an area. The first one is in an outdoor space, where it is usual to use a CCTV camera on a building wall to overlook pedestrians. This case and the technologies related to it are called crowd counting. For this purpose, it is common to use crowd-density approaches. They behave exceptionally well when dealing with a highly dense crowd. On the contrary, they do not work well with small groups of people, where an error of a single person can mean a significant change in accuracy.

This paper is an extension of the work initially presented at the IEEE International Conference on Advanced Video and Signal-Based Surveillance [1]. In this case, the multi-object tracker analyses the people's trajectory to count the number of passengers inside a train. Even though the experiments were held in a train-related environment, the offered solution can work in situations where there is a need to count people crossing a narrow space and a limited height to locate the camera. Examples of this may include building doors (banks, shopping centres, stadiums, companies, etc.), other transport (buses) or multitudinous events (festivals, sports events, etc.).

As the objective is to count the number of people inside the train and analyse the influx of people using it, the crowd density approaches are not accurate enough, so other options are considered. The second case is an indoor area CCTV or fisheye cameras. The choice of the camera will depend on the characteristics of the room. Another approach is to know the flow of people getting in and out of a room, especially if there are multiple doors in an area that a single

[*] Corresponding Author: Javier Calle, email: jcalle@vicomtech.org

https://dx.doi.org/10.25046/aj070514

camera can not cover. In this case, it is common to use flow-based or detection-based methods to track the people.

In the context of a train, there is a very particular situation; people can only get on and off at train stops. This means that to know the number of people on the train, it is not necessary to count the number of people at all times, just at the train stops.

Another condition to be considered is that, on a train, it is common to find people blocking the door or entering and stopping in the middle of the door. This causes more occlusions and can generate counting errors if the chosen method is not prepared for having a person at the edge of the door.

The next encountered challenge was choosing the counting method. For this, the strategy used is based on route estimation, thus being possible to conclude whether people enter or leave through the door. A multitracking method with a flow door counting strategy has been chosen. All the people in the scene are tracked. And the method can add those who are entering and leaving. So the passenger flow is calculated by comparing the number of people crossing the train door.

Another important point to tackle this issue is the study of the different cameras that could be used. The study was conducted to have the best possible view of the door at the time of entry and exit to ease the video analysis.

The camera that best suited all of the needs was the overhead fisheye camera, as explained in more detail in Section 3. However, using a fisheye camera entails solving a problem of distortion in the shape of people. It is especially challenging at the train's entrance, as it is common to find a different height between the train and the station. That step causes an acceleration in the person's movement through the camera, exacerbated by the camera's distortion. This causes tracking methods based on movement modelling to lose some targets as the complexity of the target's movement increases in this situation.

Given this, the best-suited tracking method is the one that uses visual appearance to aid in the tracking, but it also needs to be robust against the distortion of the cameras. For the detection and tracking of people, the online multi-object tracking approach with an affinity model from previous work [1] is proposed.

This method is based on FairMOT [2]. Still, the addition of the affinity model has been inspired to deal with the visual appearance transformation of the object while it goes through the camera's field of view. This method will allow accurate tracking of people on the scene, as it is fully prepared to deal with the deformation caused by the fisheye cameras.

The model uses a convolutional LSTM encoder-decoder architecture to learn the space-time transformation metric between consecutive re-ID embeddings extracted from the object trajectory. This allows obtaining the next re-ID embedding by considering the long-term appearance information. Furthermore, the tracking algorithm can also handle temporal occlusions in video sequences by feeding back predicted re-ID embeddings into the affinity model.

With the contributions of this work, four problems are solved:

1. Camera positioning and type of camera. After realising a synthetic simulation-based analysis, an overhead fisheye camera is selected to be installed inside the train.

2. The addition of the visual modelling for the tracking to help the Kalman filter with the issues caused by the distortion of the fisheye camera.

3. Problems caused by the coach step and the train's environment. Solved by the visual appearance-based tracker.

4. Counting people method. The solution is based on all train doors' flow using the tracking results and a counting line.

The report is organised as follows. Section 1 is the current introduction followed by the related work (section 2). This section studies different avant-garde methods to count people based on estimating via area occupancy or trajectory analysis. Additionally, it inspects various people-tracking solutions. Section 3 analyses the railway environment, i.e., it considers the different types and settings of the cameras inside a train car and selects the best option for people counting. Section 4 describes the approach chosen to count people inside a train based on overhead fisheye cameras and the passenger's trajectory. It explores the main stages of the counting people solution, detection, tracking, flow control and occupancy computation. Section 5 presents the obtained results. For this purpose, it first introduces the used datasets for performance evaluation. Straightaway the experimental and global results are given along with a comparison with state-of-the-art works. Section 6 is the conclusion. Here the results are discussed, and conclusions about the people-counting solution are presented. The future work, section 7, describes conceivable improvements and suggestions. Lastly, some acknowledgements for APPRAISE European project are given in Section 8.

2 Related work

As the main method of this article specialises in trajectory-based people counting, this section will first analyse previous SOTA work on occupancy, detection, and trajectory analysis. And it will also present different tracking methods from the SOTA.

2.1 *Area occupancy*

During the first months of the COVID-19 pandemic, occupancy problems raised their importance as it was essential to prevent disease transmission in closed environments. The main objective was to know how many people were inside an area at a particular moment. These methods are divided into object-detection or density map-based methods. Each person is detected in the first case, while a general estimation is done in the second.

2.1.1 *Object based*

The approach proposed in [3] uses an overhead video camera. They transform each video frame into a grey-scale picture and subtract the background using the empty scene knowledge. Once verified that the tracked object is a person, they keep track of it, knowing if the person is inside the area.

Another option is the one proposed in [4], where a thermal camera was used to obtain the semantic segmentation for the human. After that, they use a classification model using Adaptive Boosting

(AdaBoost) and a regression model using a shallow neural network to estimate the occupancy.

2.1.2 Density map based

When the pedestrians' density is high, the number of occlusions is also high, and person detection is almost impossible.

In these cases, state of the art is to use methods that estimate a density map and calculate its integral, obtaining the number of objects in the image as [5]–[7]. Also, some "density map" based strategies focus on the most visible part of a human in a crowd, the head, reducing the occlusion effect. The approach made in [8] not only estimates the number of people in a group but localises them with a point in the middle of their heads. They propose a new metric to achieve high localisation errors and counting performances called density Normalised Average Precision (nAP).

2.2 Trajectory analysis

These methods use different strategies to know how many people have entered or left a room. They are divided into image-based or non-image-based methods.

2.2.1 Image based methods

One of the image-based methods is to use a multi-object tracker and analyse the trajectory to count how many people have entered or left a room, as the authors do in [9]. They use a line as a frontier between the inside and the outside of the train. They focus on head detection using a standard overhead camera placed on a train platform roof over the train's door. This method is based on a detector and tracker. Once the trajectory is known, the counting module verifies if the person has crossed the limit line. If the person crosses the line, the counting method checks the direction to determine if the person is entering or leaving the train.

Another similar approach is [10], which tracks heads and uses a frontier line to count people. Their contribution is the analysis of the reference line's height. Alternatively, it is possible to use a region of the image to do the tracking as in [11].

There are different methods to analyse the trajectory. The approach proposed in [12] uses multiple independent lines for counting people. Each line counts the number of people crossing without analysing the direction, just the number of people crossing the lines.

Other methods like [13] propose a strategy that obtains the direction of the trajectories generated by the tracker by computing the angle between the position of the mass centre in the actual frame and its position in a previous frame. In [14], they propose a method consisting of two lines that define an Area of Interest that the person has to cross to be counted. In this case, the direction is determined by which line was crossed first.

In [15], the author defines a Region of Interest (ROI) where they track the objects and obtain the direction by looking at the increasing or decreasing of the y-axis coordinate of the mass centre.

2.2.2 Non image based methods

Some approaches do not use RGB cameras, such as [16], where they use an infrared array sensor. The sensor is equipped with a wide-angle lens that covers 110º and 75º on each axis, so, if the sensor is placed on the ceiling at $2.6m$ the sensor will cover $6x3.2meters$. The method allows one to count and locate the person in each frame with a margin of 0.3 meters. The dataset used contained data from up to three people simultaneously.

Another method was proposed in [17] using two infrared sensors located at a distance of 15 cm between them. The idea is to use a device that counts how many people have passed through a door. Depending on which IR detect the object first, it is possible to know the direction of that object. This method works with environments that assure that only one person is crossing that door, for example, at ticket gates.

2.3 Tracking

A tracker is needed to be able to analyse the trajectories of people, so different options for tracking were studied. Multi-object tracking approaches are typically categorised into offline and online methods. Offline methods can use all frames in a sequence (present, past and future), whereas online methods only use past and current frames for inference. So the only option to analyse people's trajectory currently getting in or out of the train is to use online methods.

Online methods frequently use the tracking-by-detection approach [18, 19]. In order to connect tracklets or detections between frames, recent online multi-object tracking systems use an affinity model in the data association step. Using pairwise affinity ratings, affinity models attempt to account for occlusions and changes in appearance. These methods can be divided into two categories: (i) robust and discriminant re-ID embeddings-based methods and (ii) sophisticated scoring functions-based methods. For the first, siamese or triplet networks are frequently suggested, as appearance cues are essential. The first approach introduced by [20] proposes a combination of the standard region loss with a triplet loss for maximising and minimising the distance between similar and dissimilar identities. In [21], authors propose deep collaborative reinforcement learning under a unified network.

Sophisticated scoring functions-based methods simultaneously output detected objects and their associated re-ID embeddings, and they focus the attention on an affinity metric design. In [22], a siamese network is explicitly designed to estimate the affinity between the detections by adding the object's appearance. In [23], a quadruplet loss is proposed to emphasise both the object's appearance and its temporal proximity. A more recent proposal is the UMA triplet network, proposed in [24] to learn the single object tracking and affinity prediction tasks simultaneously, creating a unified multi-task learning framework.

Regardless of the approach, it should be noted that all of these algorithms only consider short-term temporal appearance information between successive frames. By teaching an appearance transformation metric, our approach promotes the use of long-term temporal appearance information.

3 Railway environment

Passenger counting in train carriages is a very characteristic problem, as people can only enter and exit at train stops. They also move

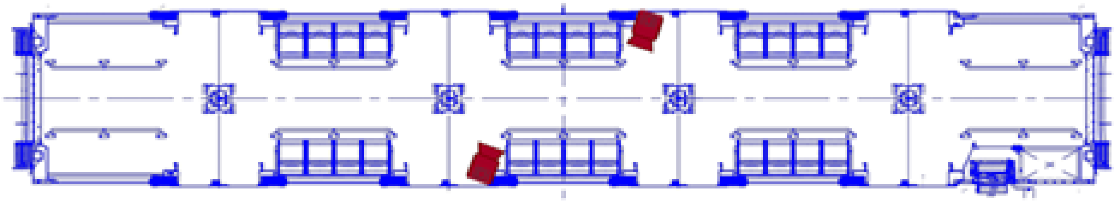


Figure 1: Schema for CCTV use.

together in groups, cross paths, wait at the door until it opens, and move all together at once. Therefore, different situations should be analysed, not only for choosing the best counting strategy but also for setting up the proper type of camera in the correct position to make this task less challenging.

It is important to note that, at certain times of the day, the density of people inside the train can be very high. This implies that the chosen method must be able to operate with a high number of simultaneous objects. In addition, the space inside a coach is minimal, implying that there will be many occlusions. Consequently, finding a counting method that works with high and low densities will be necessary.

While the train is moving, monitoring or tracking people is not necessary, as the train doors are closed, meaning there is no change in the influx of people. Therefore, the trajectory analysis of the users is only necessary when the train stops and the doors to the train open.

As stated before, there is a flow of passengers between carriages in a train, and there is an increased number of occlusions in situations of high occupancy. It would be necessary to count the number of people changing carriages to know the total count in each carriage. The door between carriages is an extra door, which implies a higher number of cameras.

Finally, it is essential to note that it is not necessary to strictly keep each person's ID throughout their journey. It is only needed during their entry or exit of the train, as the objective is to count entries and exits of the carriage through the door. Therefore, considering all this, the number of people will be controlled by focusing on door analysis. To know the total number of people inside the train, counting the number of people entering and exiting at each door is enough. Each door can be analysed separately and then added or subtracted from the people that joined or left the train.

It is crucial to select the best type of camera and its position to make the task easier. So we deeply analysed the possible camera positions and configurations in the wagon to choose the most appropriate for our study.

3.1 Explored camera configurations

It was decided to perform a study based on synthetic simulations to represent the most common and problematic situations in access to a train car. All these simulations were performed with different camera configurations inside the train. Outside will produce more useless tracking due to the people walking near the train but not going in or out. The most representative of which are shown here.

3.1.1 Train's CCTV camera

The first approach is using the CCTV cameras that are already available on the train. For security reasons, these cameras are angled, so the two front doors are checked, and the actions of users are seen to be able to act in case of an emergency, robbery, etc. The layout schematic of the cameras' positions in the coach can be seen in Figure 1.

To check the type of images that would be obtained by this type and the position of the camera, some 3D simulations were carried out. The simulated situations were (i) an empty train with one person entering, (ii) a train with a certain number of passengers already in and a group of people entering through the door.

In the situation where only one passenger is entering, Figure 2, they can be easily monitored. Through analysing images, it is easy to know whether the target is inside or not using this camera.

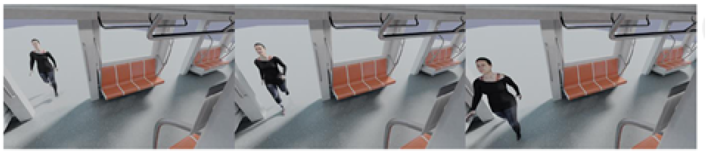


Figure 2: Example of a CCTV image.

In the second case, Figure 3, there is a greater number of occlusions. Passengers already on the train may occlude the entering people, depending on the distribution of the passengers. If rush hour images were analysed, where the train is almost fully packed before people enter, door visibility would be lost.

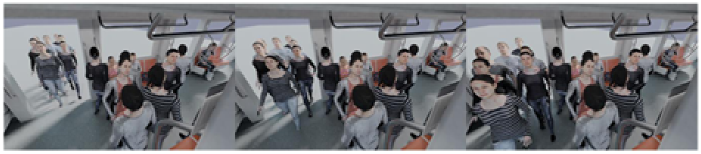


Figure 3: Example of a crowded space captured by a CCTV camera.

However, at the moment the group enters the carriage, it is not possible to differentiate whether the passengers in the back have already entered the train or not. There is no visibility of the people in the back of the group. Occlusions with this view will be a big problem, so the position and type of camera must be changed to get a more suitable angle of vision.

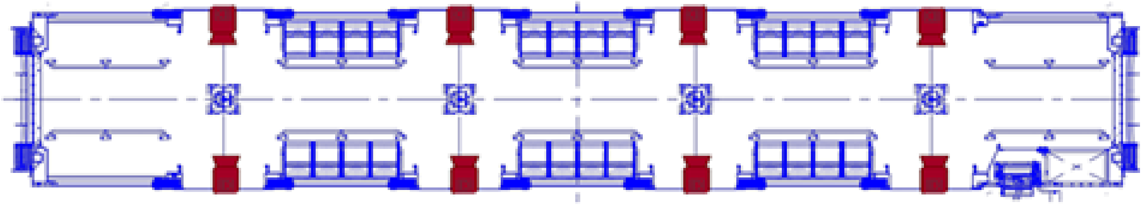


Figure 4: Schema for overhead camera use.

3.1.2 *Standard overhead camera*

The use of overhead cameras positioned over the doors is considered next. The exact position of the camera is inside the roof of the coach instead of directly above the door, as cameras can not be placed where the door's mechanism is. This means the camera is slightly displaced towards the inside of the carriage. This position also has the advantage of having the camera at a higher altitude to get a wider field of view of the scene. This kind of view can allow seeing the heads of every person going through the door, reducing the number of occlusions, and it is easier to extract the passenger's position with respect to the door. A schematic representing the distribution of the cameras can be seen in Figure 4.

The number of cameras is doubled with respect to the previous case, as there is one camera per door.

Figure 5 illustrates a situation where a single person enters the train. The placement of the camera allows for complete visualisation of the passenger. Having complete visibility of the door line allows for distinguishing whether people entered the train or not. However, the field of vision is relatively small, and we can only see the lower part of the door, but the middle and upper parts are not visible. Moreover, the field of view may not be enough to see the entire door if the door is wider or with a lower ceiling.

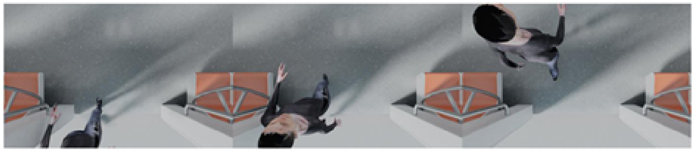


Figure 5: Example of an overhead camera.

For the second case, in Figure 6, it can be observed that the passengers already in the carriage do not generate any occlusion, as most of them are not visible. Passengers entering the carriage do not occlude each other either, but due to the limited field of vision, we have fewer frames to analyse the trajectory of the passengers.

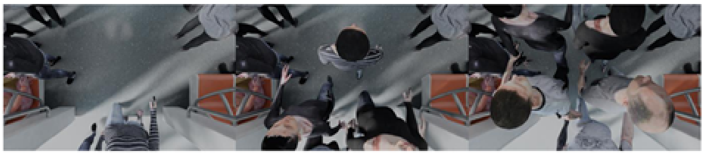


Figure 6: Example of a crowded space captured by an overhead camera.

3.1.3 *Fisheye overhead camera*

An overhead camera allows a better perspective of the passengers' flow through the door. But this solution with a standard camera has a narrow field of view. So, the fisheye camera has been proposed to solve the lack of visibility. The placement is the same as in the previous case (Figure 4), but omnidirectional 180º field of view cameras are used in this one.

Figure 7 illustrates the first experiment, where the person is completely visible. In addition, the image shows a more extensive area inside and outside the train compared to a regular overhead camera. Seeing a larger area of the outside and the coach allows us to analyse the trajectory of the passengers more robustly, as there are more frames where they are visible.

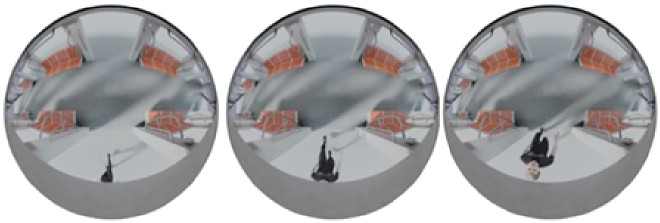


Figure 7: Example of a fisheye overhead camera.

In the second experiment (Figure 8), it can be observed that, similarly to previous cases, occlusions only appear in the case of the group entering the door when they are away from it. At the moment of getting in or out of the carriage, there are no occlusions, although the height of both the person and the camera will affect the perceived distortion.

It should be noted that the camera can only count correctly at the door below the camera, as there will be many occlusions at the opposite door. Hence one camera per door is needed.

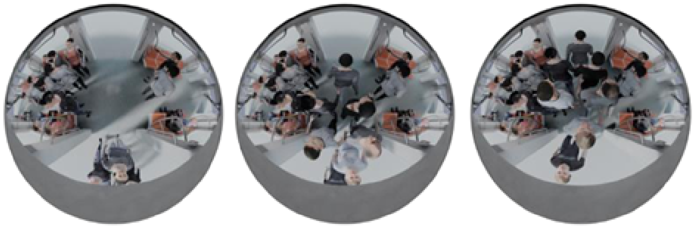


Figure 8: Example of a crowded space captured by a fisheye overhead camera.

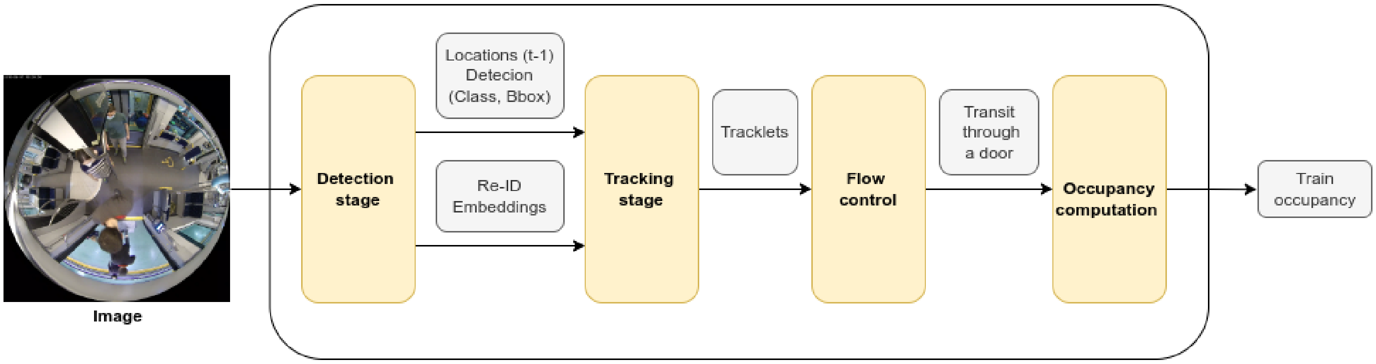


Figure 9: Overview of our overall approach.

3.2 Final configuration of the cameras

The first conclusion is that the camera must be above the door to have a good perspective. Otherwise, the door area is occluded, and the solution can not accurately analyse if a person has boarded or left the train in areas close to the door.

Secondly, after comparing the different types of cameras, it is concluded that both overhead camera options are suitable for the task. As the height of the train ceiling limits the camera location, the best option is to use a fisheye camera to obtain an omnidirectional view, even though the image will be distorted. This camera configuration allows greater flexibility when analysing each person's trajectory since they appear in a more significant number of frames.

In conclusion, the final configuration of our cameras is the one in Figure 4, using overhead fisheye cameras next to each train door.

4 Selected Approach

This section proposes a method to count people inside a train based on overhead fisheye cameras and the passenger's trajectory. Figure 9 shows the schema related to the method.

Both detection and tracking modules are the same as in our previous paper [1], specifically designed to track objects in videos recorded by omnidirectional cameras like the ones in this work. It relies on two main steps for each frame: (i) the detection of the object instances and (ii) the matching of detections to their corresponding tracklets. Once the tracking stage is done, the next step is the counting strategy for each door, where we know the passengers' transit over each doorway. The last step is to count the total number of passengers on the train.

4.1 Detection stage

In keeping with the prior work, the detection pipeline is based on the FairMOT object detection work [2]. This paper's solution uses a network structure which consists of two homogeneous branches to detect the objects and obtain re-ID embeddings in a single step. These embeddings are the feature vector of the detected object, which should ideally give us a smaller distance between the detections of the same object and maximise the space to the other things. For this, the detection network has a convolutional layer with 128 kernels that extract the re-ID features, explained with further detail in the FairMOT work [2].

The essential advantage of this network is that because it does not prioritise object detection over re-ID, the features in the output are ideal for both detection and recognition tasks. Although their work is based on DeepSORT [25], the authors present the results.

4.2 Tracking stage

The structure of this stage can be seen in Figure 10. Notably, the processing pipeline is as follows. Periodically, for each time instant t, any detected object in the scene is represented by $\{c_t^i, r_t^i, e_t^i\}$, where c_t^i is the object class, r_t^i represents the bounding box coordinates $\{x_t^i, y_t^i, w_t^i, h_t^i\}$ and e_t^i is the visual appearance representation of the object, known as re-ID embedding. The bounding box coordinates are given in MOT format [26]; (x,y) represent the top left point of the bounding box, and w and h are the width and height.

Then, depending on the detections in the first frame, a set of tracklets is initialised. As new items enter the scene and new tracklets are created, we continue to associate detections with the scene's existing tracklets for consecutive frames. In the association step, both motion (m) and affinity (a) models are used to calculate the pairwise matching scores s_{ij}^m and s_{ij}^a for every detected object i with every object that has previously been tracked j across successive time instances. A final score s_{ij}^g is calculated by a global scoring function that combines s_{ij}^m and s_{ij}^a scores.

As noted earlier, to link detections with active tracklets, we combine a motion model and an affinity model. This post-processing step is based on DeepSORT [25]. We contribute to this phase by having a single network to predict the embeddings necessary to link active tracklets and detections. As we shall demonstrate in Section 5, the features our network learns are more robust than those of the re-ID module of DeepSORT. Our visual appearance transformation network also learns the appearance evolution over time and estimates the embedding in the next frame once we get the embeddings from detections.

To estimate where the future tracklets will be, the motion model combines a constant velocity model and the Kalman Filter [27]. The scores s_{ij}^m are computed by utilising the squared Mahalanobis distance [28] to calculate the proximity between the predicted and detected bounding boxes. Finally, the unlikely matches are ruled out by thresholding the inverse chi distribution to a 95% confidence interval. In a similar process, the scores s_{ij}^a are computed utilising

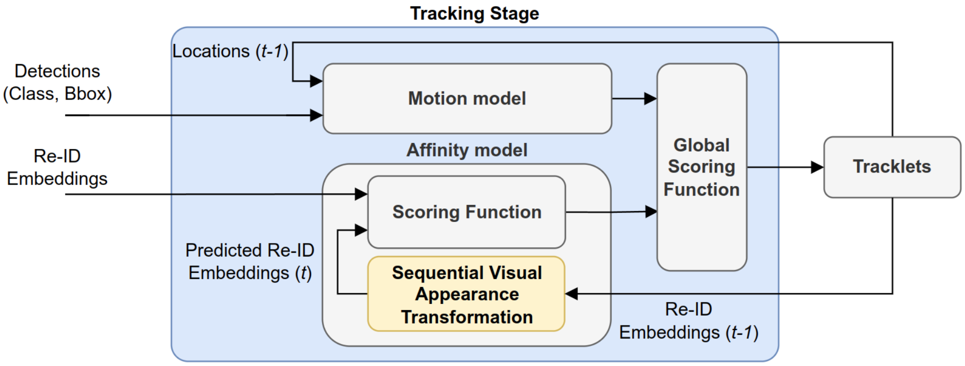


Figure 10: Overview of our online multi-object tracking approach.

the cosine distance to link the visual appearance information between the predicted and detected re-ID embeddings, all of which are based on DeepSORT [25]. The sequential visual appearance transformation network outputs the predicted embeddings online. Finally, using the following equation [25], we combine both distances:

$$s^g_{ij} = \lambda s^m_{ij} + (1 - \lambda) s^a_{ij} \tag{1}$$

where the parameter λ is utilised to balance the impact of the motion and affinity models.

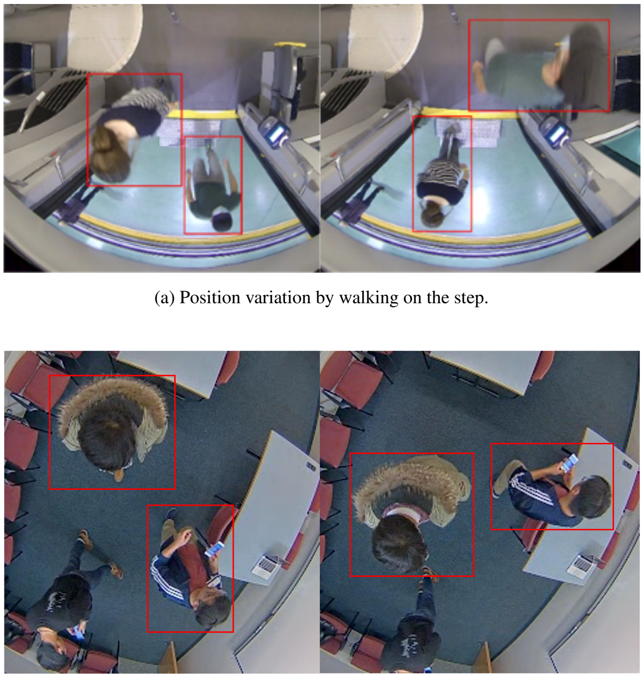
After obtaining the confusion matrix that describes the distances between detections and tracklets, we utilise the Jonker-Volgenant [29] implementation of the linear assignment problem (LAP) algorithm to match pairings that minimise the overall distance. This is a faster implementation of the Hungarian method for LAP. We save a pool of potential active tracks that we keep for n frames when none of the current detections can be matched to them. Although our method is quite similar to DeepSORT [25], we maintain both the new and lost tracks until we get a matching detection for 30 frames. For both new and lost tracks, the number of frames is the same (30). Once the target is detected again, we ought to be able to match it to one of the active tracks visually.

It is crucial to balance the effects of the Kalman and the re-ID embeddings when matching detections with tracks to get accurate results, When examining the image, the Kalman filter can effectively track the targets. Nonetheless, due to the camera's distortion, its constant velocity model causes some problems. The camera is centred on the door, the main area of interest, but that part of the camera is the one that suffers from the highest deformation. This means the target appears to move faster in the central part of the image, so constant velocity can not be assumed. The visual appearance allows us to obtain robustness against occlusions, especially against the high deformation in the door, where we want to know if the passenger is coming into or leaving the train.

4.2.1 *Sequential visual appearance transformation network*

The importance of this model comes from the cameras and environment in which we are developing this work. In the sequence in Figure 11a we can observe the deformation suffered by two people due to the step at the train's entrance. The deformation is more significant when the target is closer to the camera. The height difference created by the step changes the visual appearance of people more than under normal circumstances. We can see in Figure 11b the deformation in a regular fisheye image taken from the HABBOF [30] public dataset.

In both cases, the people in the sequence have taken only two steps and walked roughly the same distance. But if we look at the person with the green shirt on the right side of the image in Figure 11a, they seem to have moved a lot, and their appearance has changed more when compared to the person in the striped sweatshirt in the right side of the image in Figure 11b.



(a) Position variation by walking on the step.

(b) Position variation by walking on a flat floor.

Figure 11: Compare movement distance due to step

This is the main reason why the visual appearance transformation network is beneficial to model better the change in the person's appearance. The change in appearance is so significant that simply using the embeddings is not enough. We need a dedicated model that learns the visual transformation. On the other hand, we already mentioned that the Kalman filter is not robust enough in this situation. However, it's not only because the main area of interest is the central part of the image. The step between the train and the ground magnifies the ”acceleration” effect. Due to the distortion and the step in the door, our main area of interest, the target’s velocity in the image is far from constant when getting in or out of the train.

Now, let us explain the architecture of the proposed network, which we maintained from the previous work. We are working with time data; thus, it makes sense to think about using a recurrent neural network (RNN). RNNs have an internal state known as the hidden state that holds the data relevant to what has been seen, and they sequentially process the new input data to maintain track of the temporal information.

Our sequential visual appearance transformation network has an encoder-decoder structure of deep convolutional long short-term memory (convLSTM) architectures [31]. We can see a graphical representation of a single cell in Figure 12.

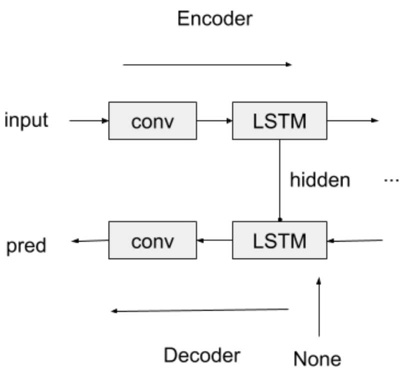


Figure 12: ConvLSTM with an encoder-decoder structure.

As noted, both the encoder and decoder parts are represented by a convLSTM cell. The convLSTM decoder uses the convLSTM encoder’s hidden states as its hidden states. The suggested network may be implemented by iteratively concatenating many encoder-decoder structures. We decide to utilise convLSTM since it provides a method for transferring information across sequences. This keeps track of the appearance information for later and stops earlier signals from progressively vanishing. [32]. For this, the encoder captures the context of the visual appearance information, represented by our re-ID embeddings, summarising the previous states of the object trajectory. Conversely, the decoder uses the sequential accumulated transformation, in this instance, the anticipated re-ID embedding for the current frame, to generate the future object appearance.

Using ground truth object sequences, we learn the offline transformation of the visual appearance. We produce a set of re-ID embeddings $e^i_{gt,j}$ for each object identity i that are taken from ground truth detections of the object trajectory. For every $e^i_{gt,j}$, we infer the predicted re-ID embedding $e^i_{p,j}$ by using the proposed network. Finally, we compute the affinity loss as follows:

$$L_{\text{affinity}} = 1 - |\frac{1}{N}\sum_{i=1}^{N} D_c(e^i_{p,j}, e^i_{gt,j})| \qquad (2)$$

where D_c represents the cosine distance and N is the batch size. We split the ground truth trajectories into sub-trajectories with a single length j because the length of the ground truth trajectories is unequal. As a result, we can do the training in batch mode.

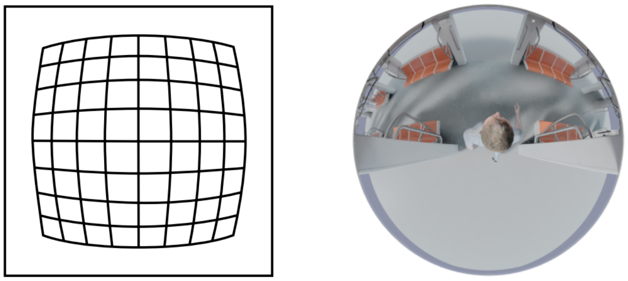
4.3 Counting strategy for flow control

Once explained how people are tracked during training, the following section describes how counting people is done. For this purpose, two main factors are taken into account. Firstly, the position and effect of the fisheye camera and how this affects the appearance of people in the image and, therefore, the selection of an appropriate reference point for a person. And, secondly, the logic behind the people counting module.

4.3.1 Fisheye camera effect

Fisheye cameras generate large deformations in objects as they move through space, as explained in section 4.2.1. They specifically produce barrel distortion (Figure 13a), which has more effect at the wide-angle end of the range of the image. The deformation may cause difficulties in developing a good people-counting strategy. So selecting a suitable reference point for a person to identify when he has entered or left a train door is key to obtaining a high hit rate.

Just in the centre of the image of a fisheye camera, an object’s appearance will be the least deformed. This can be seen in Figure 13a: the squares located in the middle of the image have no distortion, while the extremes of the image suffer a great distortion. In a real case of a person crossing the door line, only their head and shoulders will be visible in the centre of an image, with no distortion. (Figure 13b).



(a) Barrel distortion [33]

(b) Overhead fisheye image with the person above the camera, and camera in line with the door.

Figure 13: Fisheye camera distortion

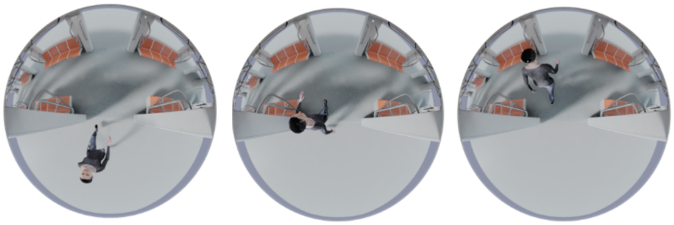


Figure 14: Simulation of an overhead fisheye camera in line with the door. *Left: Person approaching the door. Centre: Person below the camera and above the door line. Right: Person walking away from the door.*

When an object is further from the centre, the object's image suffers more deformation. For a person, once they are walking away from the camera's centre point, their entire body will appear in the image as long as there are no occlusions. An example can be seen in the left and right images from Figure 14. So the distortion is related to the object's position or person respective to the camera.

Ideally, the best place to locate the camera and detect peoples' positions would be just above the limit line; in this case, the train door axis, as shown in Figure 14. Using the central point of the bounding box would work perfectly as a reference point for the person's position. However, the camera is placed inside the train because of design and structural reasons. Therefore, placing the fisheye camera outside the door axis results in a person's appearance distortion. This makes it somewhat more complicated to define its reference point and to know whether a person is inside or outside the carriage in borderline situations, as in Figure 15. Moreover, this distortion is even increased by some trains' steps that may have at the doors.

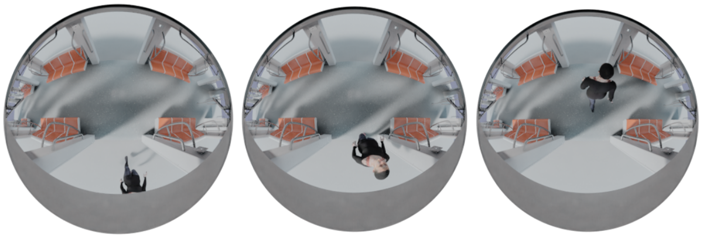


Figure 15: Simulation of an overhead fisheye camera inside the train. *Left: Person approaching the door. Centre: Person just above the door line. Right: Person walking away from the door.*

The position variation of an object with a constant height moving towards or away from the centre of an overhead fisheye camera is different for the highest and lowest part of the object. This means that when an overhead camera captures the movement of a person, the head's location suffers more variation than the feet' location. For example, in Figure 16 the variation of the head doubles the variation of the feet of the person.

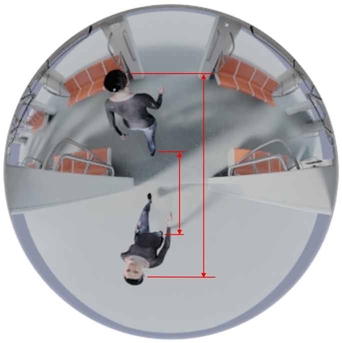
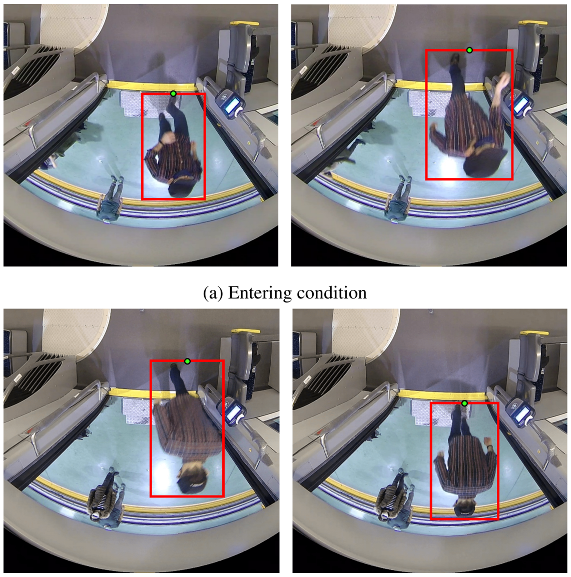


Figure 16: Deformation of top and bottom parts of an object moving in a fisheye camera. In this case, the person's head suffers more variation than the feet.

Usually, with standard CCTV cameras, it is common to use the bounding box centre or the person's head as a reference point, as the head is the most visible part. However, based on the problems stated before, with this camera configuration, the most suitable reference for defining the position of a person is their feet. As for the bounding box, the selected reference point is the upper-centred point related to the feet. Figure 17 shows the entering and exiting conditions picking the feet, located in the upper-centred point of the bounding box, as a reference point to consider that a person has crossed the door line axis.

While entering the train door (Figure 17a) the person is first entirely outside the train. The right image shows that most of the body appears outside the train, but the feet are inside. That means that the person has entered the train. This happens due to the previously explained distortion that depends on the height of an object. While exiting the train (Figure 17b) the left image shows that most of the body is outside the train, but is not considered that the person has left the train until the feet are out of it. This condition is fulfilled in the right image, where the feet are finally outside the train door limit.



(a) Entering condition

(b) Exiting condition

Figure 17: Entering and exiting conditions of a person through a train door (The train door limit is the yellow line).

This way, if a person is standing right at the door's limit, even if their body's bounding box is mostly outside the train, the person will be considered inside the train, as in Figure 18.

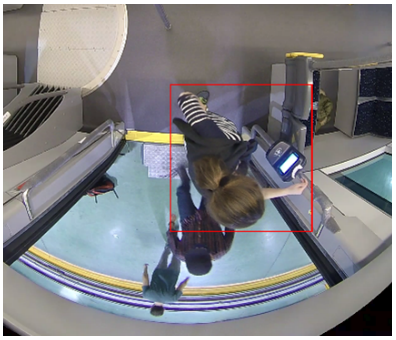


Figure 18: Example of a person standing right under the door.

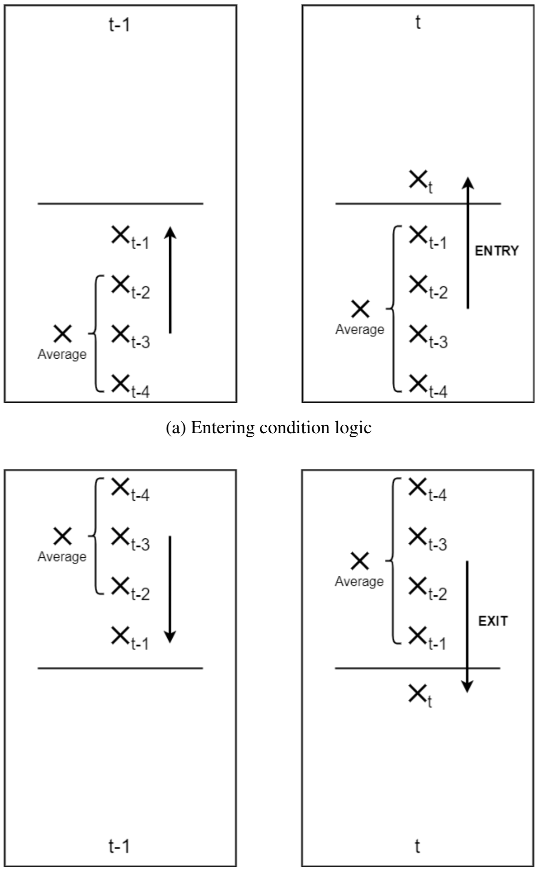
4.3.2 Counting logic

The counting logic is based on the state-of-the-art of [34]. The main idea is to determine a line and count how many people have crossed it in each direction. The line is defined as a horizontal line in the centre of the image.

To determine if an object has crossed a limit line, the author of the paper mentioned above uses the vertical (y-axis) coordinate of the centre of the bounding box of a person in each frame. Then, the current position and the mean value of the previous positions of the tracked person are considered to evaluate the crossing. As it states, the mean of all the previous points is selected because:

> ”The reason we take the mean is to ensure our direction tracking is more stable. If we stored just the previous centroid location for the person, we leave ourselves open to the possibility of false direction counting. [...] by taking the mean, we can make our people counter more accurate.”

In the current case of this paper, the limit line will be defined as the limit of the door (notice the yellow line in the Figures 17). So it will also be a horizontal line in the image. A person will be considered that has crossed the line if the limit line's vertical (y-axis) value is between the current position and the mean of the previous ones. A representation of the stated logic can be seen in Figure 19.



(a) Entering condition logic

(b) Exiting condition logic

Figure 19: Entering and exiting logic of a person crossing the line.

- **Enter condition logic** (Figure 19a): A person has entered the train if:
  - The current reference point is above the limit line.
  - The average of the previous points is below the line.

  (The person is moving to the upper part of the image, representing the inside of the train.)

- **Exit condition logic** (Figure 19b): A person has exited the train if:
  - The current reference point is below the limit line.
  - The average of the previous points is above the line.

  (The person is moving to the lower part of the image, representing the outside of the train.)

Some changes were made to the approach mentioned above due to some requisites specific to using the fisheye camera. Instead of using the bounding box centre, the reference point is the top middle point. This point coincides with the location of the feet in the fisheye images, as explained in Section 4.3.1; this reference point is more suitable to determine if a person is inside or outside the train.

The objective of the counting logic is to analyse the flow through the door. If a person goes in or out of the train multiple times in the same recording, the system must count every time the person crosses the line limit (see Figure 20). This may carry problems related to the counting logic, especially with the average of the previous reference points. If the person goes through the door several times, the average point will be unsuitable for the stated logic. That is why the previous tracking points are reset after a person crosses the line limit to avoid further problems. This way, the average position is kept on one side of the limit line, and if the person crosses again, even with the same tracking ID, the logic will still work.

The paper proposes a solution to reset the trajectories based on separating the whole trajectory into sub-trajectories. Every time a tracked person crosses the limit line in either direction, a sub-trajectory containing the previous points of the tracking is saved and set aside. Then, a new sub-trajectory is started, containing only the first reference point of the moment of entering or exiting the door. Figure 21 shows how the trajectory of Figure 20 is separated into four sub-trajectories.

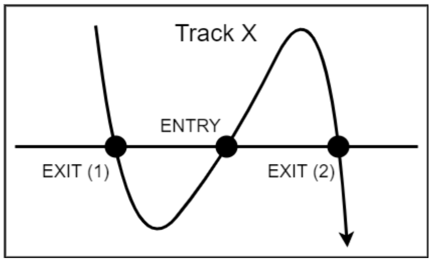


Figure 20: Example of the trajectory of a person entering and exiting multiple times.

This way, the average of the previous points is kept on one side of the line, and as soon as a person crosses the line, the module will count as the entry or exit and reset the previous points for the next sub-trajectory. For example, for the first exit in Figure 21 (Sub-track X_1) the average of the previous points is on the upper side of the line. Once the person crosses the line (green circle), another sub-trajectory is started (Sub-track X_2). This sub-track ends once the person crosses the line and enters (orange circle). The average point in red is kept on the downside of the line. Similarly happens with the next sub-trajectory (Sub-track X_3), but the average is kept on the upper side. The trajectory ends when the person is no longer tracked (Sub-track X_4).

Finally, the flow of a door is composed of two positive integer numbers:

- **Enter flow**: each person that enters the train counts as a **positive enter**.
- **Exit flow**: each person that exits the train counts as a **positive exit**.

Separating the enter and exit values gives more information about the flow of a door. If a unique number was given as a flow, the information of people who entered or exited the door would be lost. For example, it would be the same for the cases where two people enter and two people exit and for no one entering and exiting the door (flow equal to zero in both cases).

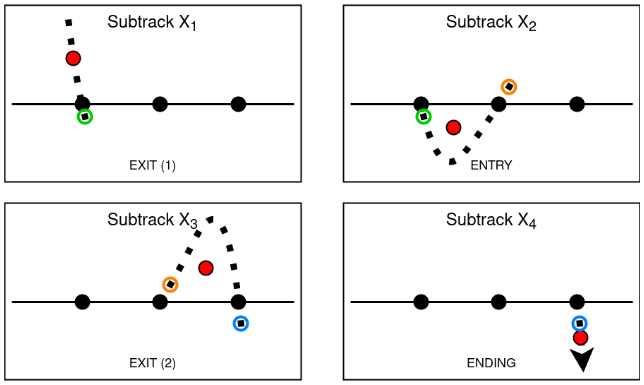


Figure 21: Considered sub-trajectories to evaluate if a person has crossed or not the line in multiple crossings within the same recording. The red dot indicates the average point of the sub-trajectory. The dotted line indicates each person's step. The coloured circle over a dot indicates the end/beginning of a sub-track.

4.4 Occupancy computation

The methodology for calculating the occupancy of a train is formed by the sum of the people entering minus the sum of the people leaving the train at each door; see Equation 3. It's important to notice that a particular door's flow can have either positive, zero or negative value. But the overall train occupancy will be either zero or positive.

$$Total\_flow_{door} = Enter\_flow_{door} - Exit\_flow_{door} \quad (3)$$

Once the total flow of each door is known, to calculate the overall train occupancy, all door flow values are added together, as shown in 4. This particular value will indicate the number of passengers on the train at a given time.

$$Total\_occupancy_{train} = \sum_{d=0}^{N_{door}} Total\_flow(d) \quad (4)$$

$flow(d)$ is the flow door value, and d is the door number.

5 Results

5.1 Datasets

The next section presents the datasets used for the training and testing of the detection and tracking models and the use case of people counting through trajectory analysis. The datasets are separated into two sections: Public datasets (Section 5.1.1), which were used to train the detection and tracking models; and Proof of concept dataset (Section 5.1.2), which consists of video recordings of a real train fisheye camera where people enter and exit a train.

5.1.1 Public datasets (Detection and Tracking oriented)

The performance of our method is assessed over a certain amount of data obtained from omnidirectional cameras. We considered seven publicly available datasets specifically created to address the problem of people tracking and detection. They combine multiple indoor environments, including a wide range of challenging scenarios: crowded room, severe body occlusions, various body poses, head camouflage (e.g., hoods, hats) and low-light conditions.

Table 1 provides a summary of those datasets, including the number of frames, the number of people (IDs), and the data source for each of them.

It should be noted that the HABBOF, FES, and PIROPO databases lack tracking information.

We added some data captured in trains with the specific camera setting we proposed for these datasets.

Table 1: Public omnidirectional datasets.

Dataset	Images	IDs	Data Source
CEPDOF [35]	25.5k	51	Real
HABBOF [30]	5.8k	-	Real
FES [36]	301	-	Real
Bomni [37]	12.9k	85	Real
THEODORE [36]	100k	2307	Synthetic
Mirror World [38]	7k	63	Real
PIROPO [39]	3k	-	Real
PATHTRACK [40]	276.4k	16,287	Real
Total	430.9k	18,793	-

5.1.2 Proof of concept dataset (People counting)

The dataset for obtaining the results of counting people consists of several video clips. The recordings were captured by a fisheye camera located at the door of a train coach. The videos contain different cases of people entering and exiting the door:

- Cases of one person to multiple people entering and exiting the door one by one.
- People entering or exiting together in groups on the train.
- Video recordings with two or more people, usually crossing their paths.

- Cases where people enter and exit the train at varying walking speeds (Fastly and slowly moving passengers).

- Other actions such as stopping in the middle of the door.

All the clips were classified into three categories depending on the difficulty of the video stream:[1]

- **"Low"**: In these videos, there is usually one person who enters and/or exits the train door.

- **"Medium"**: There are usually 2 or 3 people who enter and/or exit the train.

- **"High"**: There are more than five people who enter and/or exit the train.

Once all the clips are grouped in the three types of difficulty, the dataset remains as in Table 2.

Table 2: Number of clips separated into difficulties.

Difficulty	**No. of clips**
Low	41
Medium	39
High	22

The ground truth data of the different clips consists of the tally of how many people enter and exit the train door, i.e., the total count of people crossing the limit line (door flow). Alongside the difficulty level and the distinction between people entering and exiting the train door, the ground truth data is composed as shown in Table 3.

Table 3: Ground truth data of the count of people crossing the train door line. The difference between entry and exit and the three difficulty levels of clips are reflected.

	People crossing count (GT)		
	Low	**Medium**	**High**
Entry	27	47	68
Exit	23	49	68
Total	50	96	136

In total, the dataset contains **282** door flows or entries and exits, which are divided into "Low", "Medium", and "High" difficulties.

5.2 *Experimental evaluation*

Our previous work explained how we obtain the best tracker configuration to achieve a state-of-the-art tracker [1].

Although our main contribution to the tracking stage was proposed in the association part, we also trained the FairMOT object detector network [2] following the same implementation details as the authors. The original network is trained on frontal view images, so its performance on omnidirectional data drops dramatically. Consequently, we have used all omnidirectional datasets to fine-tune the pre-trained network.

We select trajectories with a sequence length of $j = 40$ frames for our sequential visual appearance transformation network This value was chosen because in a camera recording at 15-20 fps, 40 frames are approximately 2 seconds of the recording, and a person will approximately take that time to enter a train; thus, we have 40 consecutive re-ID embeddings for each person's identity.

The network can keep the identification of the tracked individual for as many frames as they are in the scene, although it was designed to function with as little as 40 frames in training.

The tracklet is retained in the pool of potential identities for 30 frames if the system loses track of a person. If, after 30 frames, the system cannot match the tracklet with any further detections, we assume the individual has departed the scene. This part is important for evaluating the public datasets. However, as the goal is to know whether a person has walked through a door and the direction they walked in, keeping track of them for longer while they are inside the train is not essential.

We have used the same configuration as in our previous work [1] where we prove that using a high λ value and, therefore, giving more weight to the score of the Kalman filter results in a significantly lower outcome. This shows the importance of using appearance-based matching, adding robustness against occlusions. The best option is to use a low λ value but different from zero, so we use $\lambda = 0.1$.

The system's ability to re-ID people comes from visual information. In case of occlusion, if the target moves through the image, a motion-only model would discard the detection and start a new tracklet. That means we would be unable to count the action if the occlusion happens during an entry or exit if it wasn't for the visual model.

Following the standard practices in multi-object tracking, we use the multi-object tracking accuracy (MOTA) [41] and the ratio of correctly identified detections over the average number of ground-truth and computed detections (IDF1) for rigorously evaluating re-ID features with ground-truth detections.

To evaluate the performance of our tracker, we have considered a unique scenario containing the test set from all datasets that include ground truth tracking information, meaning: CEPDOF, MWR, and Bomni, with six different sequences in total. Also, to achieve a better performance, we train our tracker with the PATHTRACK dataset [40]. This dataset is one of the largest publicly available multiple objects tracking data sets.

Following the same implementation details as in our previous work [1], we obtain three proposed networks. Table 4 shows the MOTA and IDF1 results for the baseline FairMOT method and our three proposed networks. The convLSMT-Enc-Dec-3 achieves the best MOTA and IDF1 results with 88.21% and 87.77%, respectively.

[1] *Even though the video difficulties are separated into three subsets, the "Low" and "Medium" video complexities do not differ too much. The major dissimilarity is with the "High" difficulty videos.*

Table 4: Evaluation of our online tracking by detection approach including PATH-TRACK dataset [40]

Model	MOTA (%)	IDF1 (%)
FairMOT	82.94	80.25
FairMOT + convLSMT-Enc-Dec-1	87.85	87.73
FairMOT + convLSMT-Enc-Dec-3	**88.21**	**87.77**
FairMOT + convLSMT-Enc-Dec-5	87.15	87.55

We compare the performance of our approach on the Bomni [37] and Mirror World [38] datasets as we only found results for these two datasets. Table 5 demonstrates the benefits of our affinity model concerning state-of-the-art methods. As state-of-the-art methods do, we also include multi-object tracking precision (MOTP) for this comparison. Results show that our approach outperforms all existing methods considering the omnidirectional perspective. In particular, it improves the best performance (93.5%) obtained by BTLD [42] by almost 2% using the Bomni dataset. Similarly, we outperform the best performance (38.4%) obtained so far by SORT [38] by more than 40% using the Mirror World dataset.

Due to the lack of recent results for omnidirectional datasets, we decided to compare ourselves with DeepSORT [25]. We use the detections of our model and perform the tracking with their approach. Table 5 and 6 show results across the six test sequences. We used the model with $\lambda = 0$ for comparison, as that is the value they used in their final implementation. Table 5 shows that using FairMOT instead of DeepSORT yields slightly better results, proving the re-ID embeddings of their work are better than those of the original DeepSORT. Table 6 shows that our method improves tracking given the increase in the number of mostly and partially tracked people.

Table 5: Comparison with state-of-the-art methods.

Approach	MOTA (%)	MOTP (%)	IDF1 (%)
Bomni Dataset			
FTMO [37]	73.52	72.00	-
FTMO (updated) [43]	86.27	72	-
RTMOT [42]	78.55	76.74	-
BTLD [44]	93.5	-	-
Ours	**94.27**	**92.14**	**95.14**
Mirror World Dataset			
SORT [38]	38.4	-	32.1
Ours	**84.14**	**81.93**	**88.68**
Across all datasets (MW, Bomni and CEPDOF)			
DeepSORT[25]	81.37	80.15	79.98
FairMOT[2]	82.94	80.34	80.25
Ours	**88.03**	**84.25**	**86.87**

Finally, Figure 22 shows that the presented method is more robust to the appearance distortion caused by overhead cameras than DeepSORT. Moreover, it can recognise the target in the centre of the image, whereas DeepSORT changes its ID due to appearance distortion.

Taking all of these results into account, we prove that adding the sequential visual appearance transformation network helps with the reidentification task in the case of fisheye cameras. Due to the deformation, the appearance of the objects is less constant than with regular cameras, so we need the aid of a model specialised in predicting the object's appearance and comparing the embeddings. We can't use the embeddings directly because the appearance changes between frames.

Table 6: Tracking quality compared with DeepSORT.

Model	Mostly tracked	Partially tracked	Totally lost
DeepSORT	16	13	6
FairMOT + convLSMT-Enc-Dec-3	**20**	**11**	**4**

5.3 Global results

The proof of concept dataset defined in Section 5.1.2 is used to obtain the results of the people counting method. This dataset is exclusively used to get the results of the people counting module because it is the closest dataset to a real-case scenario, the main objective is to check the feasibility of the people counting logic from Section 4.3.2. The dataset is not considered significant enough to present the results of the detection and tracking module due to its size.

The accuracy results for each difficulty level were based on the total count of entries and exits from each video divided by its ground truth (GT), as shown in Equation 5. The following Table 7 shows the accuracy results of the people counting logic.

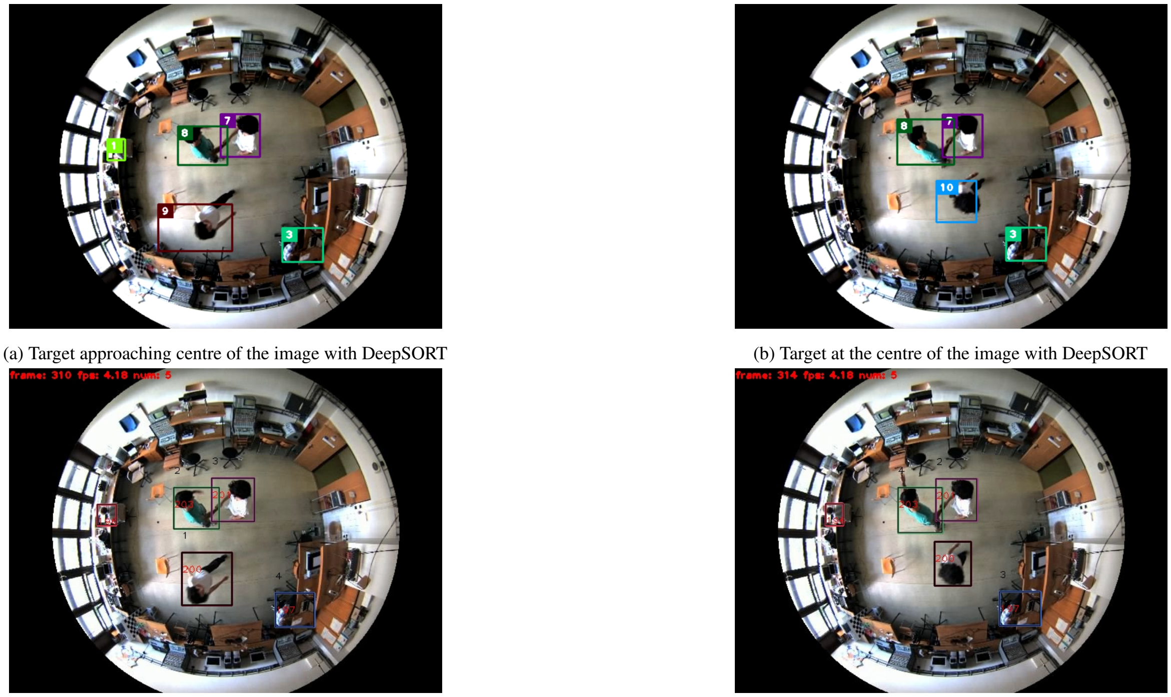
$$accuracy_{Entry/Exit}(\%) = \frac{\sum Output}{\sum GT} \tag{5}$$

Table 7: People counting accuracy (%).

	People counting accuracy (%)		
	Low	**Medium**	**High**
Entry	100.00	97.87	91.18
Exit	100.00	89.80	79.41
Mean	100.00	93.75	85.29

The overall accuracy of counting people getting on and off is **90.78%**. The worst results are presented with "High" difficulty clips, as the detector may fail in crowded situations or when people cross their paths. Even though the dataset used to obtain these results is based on real case scenarios (recorded with fisheye cameras and with people entering and exiting the train door), the number of door flows is not representative enough for a generalization. So the results might not be reliable. Future work aims to create a more extensive dataset with more realistic scenarios to validate the people-counting module alongside the detection and tracking models.

The approach proposed in [9] was similar to this project's strategy. As the author uses a tracker and a counting line to know how



(a) Target approaching centre of the image with DeepSORT

(b) Target at the centre of the image with DeepSORT

(c) Target approaching centre of the image with our method

(d) Target at the centre of the image with our method

Figure 22: One of the scenarios where our method can keep track of a target compared to state-of-the-art methods.

many people entered or left the train. In this case, it uses a standard overhead camera, so there is no image distortion. This type of camera needs a higher position to have enough vision to analyze the passengers' flow, so the point of view (camera location) is outside the train, on the roof of a train platform. The accuracy metric used in this proposal is based on a confusion matrix. The ground truth data of the videos specifies when a person has entered or exited the train. The proposal achieves an accuracy of 92.01% and 92.47% in counting people leaving and entering the train, respectively.

The approach in this paper instead is based on an overhead fisheye camera inside the train. A fisheye camera can be located in a lower position maintaining a wide viewing angle, and, therefore, inside the train, close to the door. This location's advantage is that the cameras can follow the passengers' trajectory inside and outside the train at the expense of having distortion. But, as explained in section 4.3.1, the tracker solves the fisheye camera's image distortion problem. The accuracy metric used in this paper is based on counting entries and exits, achieving an overall accuracy of 90.78% against the proof of concept dataset.

Comparing the results and affirming which method is better does not make sense since the cameras and the databases are different. The only viable comparison is to evaluate the accuracy of the two solutions and check that the results are similar. It is impossible to affirm that one methodology is better than the other, as their goal is only to count people, but we also want to analyze the trajectory and the influx of people using the train.

6 Conclusions

This paper presents a method based on video sequences to count the number of people inside a train car based on the doors' flow. The technique can be applied in environments with narrow spaces and a limited height to locate the camera, such as buildings, transports or events where a door limits the entry. The proposed approach is based on an appearance-based multi-object tracker and a door's flow counting method that analyze the trajectory of each passenger, counting how many times the people enter or leave the coach.

A study of the best camera type and its location was done to obtain the best possible vision of the passengers at the door. After analyzing the train's environment and carrying out different simulations, the best results were given by an overhead fisheye camera. This camera causes distortion, which our appearance model can handle, improving the performance achieved by the motion modelling-only models. Moreover, the selected trackers can also handle the step acceleration problem.

Once the trajectory was known, a flow control module based on a counting line was developed to calculate the number of times the passengers crossed the line in each direction. Different options for the reference point of a bounding box were regarded, and the central upper point was selected as the best reference point to decide whether a person was on either side of the line. Finally, the overall train occupancy module knows the train occupancy based on all the door's flow information.

After all this work, it is possible to affirm that using fisheye cameras allows for better tracking of people thanks to their broader field of view. The selected tracker can solve the distortion problems caused by the fisheye camera. Moreover, the counting method is robust against the different situations found on a train, and it is successfully tested on a proof-of-concept dataset. Finally, it is confirmed that the previous tracker [1] works as well as expected in a different and more complex environment.

7 Future work

It has been proven that the method works, but its accuracy cannot be confirmed because the dataset used is small and in a controlled environment. The future line of work is to obtain a dataset taken in a real environment, annotate it and check its absolute precision. As for the algorithm, the next step would be to try a Generative adversarial network (GAN) [45] to model the appearance of the detections, as this model has proven to be a very effective way of predicting visual appearance in a time sequence.

8 Acknowledgments

This work has been partially funded by the European Union's H2020 research and innovation programme under grant agreement 101021981, project APPRAISE (Facilitating public & private security Operators to mitigate terrorism scenarios against soft targets).

References

[1] I. Sagastiberri, N. v. d. Gevel, J. García, O. Otaegui, "Learning Sequential Visual Appearance Transformation for Online Multi-Object Tracking," in 2021 17th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 1–7, 2021, doi:10.1109/AVSS52988.2021.9663809.

[2] Y. Zhang, C. Wang, X. Wang, W. Zeng, W. Liu, "FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking," arXiv preprint arXiv:2004.01888, 2020.

[3] E. Khoumeri, H. Fraoucene, E. Khoumeri, C. Hamouda, R. Cheggou, People Counter with Area Occupancy Control for Covid-19, 405–415, 2021, doi:10.1007/978-3-030-63846-7_38.

[4] A. Naser, A. Lotfi, J. Zhong, "Adaptive Thermal Sensor Array Placement for Human Segmentation and Occupancy Estimation," IEEE Sensors Journal, **21**(2), 1993–2002, 2021, doi:10.1109/JSEN.2020.3020401.

[5] Z.-Q. Cheng, J.-X. Li, Q. Dai, X. Wu, A. Hauptmann, "Learning Spatial Awareness to Improve Crowd Counting," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 6151–6160, 2019, doi:10.1109/ICCV.2019.00625.

[6] Y. Li, X. Zhang, D. Chen, "CSRNet: Dilated Convolutional Neural Networks for Understanding the Highly Congested Scenes," 1091–1100, 2018, doi:10.1109/CVPR.2018.00120.

[7] Y. Miao, Z. Lin, G. Ding, J. Han, "Shallow Feature Based Dense Attention Network for Crowd Counting," in The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, 11765–11772, AAAI Press, 2020.

[8] Q. Song, C. Wang, Z. Jiang, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, Y. Wu, "Rethinking Counting and Localization in Crowds:A Purely Point-Based Framework," 2021.

[9] S. A. Velastin, R. Fernández, J. E. Espinosa, A. Bay, "Detecting, Tracking and Counting People Getting On/Off a Metropolitan Train Using a Standard Video Camera," Sensors, **20**(21), 2020, doi:10.3390/s20216251.

[10] D. Kuplyakov, Y. Geraskin, T. Mamedov, A. Konushin, "A Distributed Tracking Algorithm for Counting People in Video by Head Detection," paper26–1, 2020, doi:10.51130/graphicon-2020-2-3-26.

[11] J.-W. Kim, K.-S. Park, B.-D. Park, S.-J. Ko, "Real-time vision-based people counting system for the security door," in Proceedings of the IEEK Conference, 1416–1419, The Institute of Electronics and Information Engineers, 2002.

[12] J. Barandiaran, B. Murguia, F. Boto, "Real-time people counting using multiple lines," in 2008 Ninth International Workshop on Image Analysis for Multimedia Interactive Services, 159–162, IEEE, 2008.

[13] J. Ahmad, H. Larijani, R. Emmanuel, M. Mannion, A. Javed, "An intelligent real-time occupancy monitoring system using single overhead camera," in Proceedings of SAI Intelligent Systems Conference, 957–969, Springer, 2018.

[14] S. Yu, X. Chen, W. Sun, D. Xie, "A robust method for detecting and counting people," in 2008 International conference on audio, language and image processing, 1545–1549, IEEE, 2008.

[15] P. Chato, D. J. M. Chipantasi, N. Velasco, S. Rea, V. Hallo, P. Constante, "Image processing and artificial neural network for counting people inside public transport," in 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM), 1–5, IEEE, 2018.

[16] M. Bouazizi, C. Ye, T. Ohtsuki, "Low-Resolution Infrared Array Sensor for Counting and Localizing People Indoors: When Low End Technology Meets Cutting Edge Deep Learning Techniques," Information, **13**(3), 2022, doi:10.3390/info13030132.

[17] R. L. dos Santos, H. C. de Oliveira, M. C. de Almeida, D. F. Vieira, E. P. L. Junior, T. Ji, "A Low-Cost Bidirectional People Counter Device for Assisting Social Distancing Monitoring for COVID-19," Journal of Control, Automation and Electrical Systems, **33**(4), 1148, 1160, 2022, doi:10.1007/s40313-022-00916-z.

[18] P. Bergmann, T. Meinhardt, L. Leal-Taixe, "Tracking Without Bells and Whistles," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[19] W. Tian, M. Lauer, L. Chen, "Online Multi-Object Tracking Using Joint Domain Information in Traffic Scenarios," IEEE Transactions on Intelligent Transportation Systems, **21**(1), 374–384, 2020, doi:10.1109/TITS.2019.2892413.

[20] W. V. Ranst, F. De Smedt, J. Berte, T. Goedemé, "Fast Simultaneous People Detection and Re-identification in a Single Shot Network," in 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018, doi:10.1109/AVSS.2018.8639489.

[21] L. Ren, J. Lu, Z. Wang, Q. Tian, J. Zhou, "Collaborative Deep Reinforcement Learning for Multi-object Tracking," in V. Ferrari, M. Hebert, C. Sminchisescu, Y. Weiss, editors, Computer Vision – ECCV 2018, 2018.

[22] X. Dong, J. Shen, "Triplet Loss in Siamese Network for Object Tracking," in Proceedings of the European Conference on Computer Vision (ECCV), 2018.

[23] J. Son, M. Baek, M. Cho, B. Han, "Multi-object Tracking with Quadruplet Convolutional Neural Networks," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 3786–3795, 2017, doi:10.1109/CVPR.2017.403.

[24] J. Yin, W. Wang, Q. Meng, R. Yang, J. Shen, "A Unified Object Motion and Affinity Model for Online Multi-Object Tracking," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[25] N. Wojke, A. Bewley, D. Paulus, "Simple Online and Realtime Tracking with a Deep Association Metric," 2017.

[26] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. D. Reid, S. Roth, K. Schindler, L. Leal-Taixé, "MOT20: A benchmark for multi object tracking in crowded scenes," CoRR, **abs/2003.09003**, 2020.

[27] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Transactions of the ASME–Journal of Basic Engineering, **82**(Series D), 35–45, 1960.

[28] P. C. Mahalanobis, "On the Generalised Distance in Statistics," Proceedings of the National Institute of Sciences of India, **2**(1), 49—55, 1936.

[29] R. Jonker, A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," Computing, **38**(4), 325–340, 1987.

[30] S. Li, M. Tezcan, P. Ishwar, J. Konrad, "Supervised People Counting Using An Overhead Fisheye Camera," in 2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 1–8, 2019, doi: 10.1109/AVSS.2019.8909877.

[31] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W. kin Wong, W. chun Woo, "Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting," 2015.

[32] S. A. Rahman, D. A. Adjeroh, "Deep Learning using Convolutional LSTM estimates Biological Age from Physical Activity," Scientific Reports, 2019, doi:10.1038/s41598-019-46850-0.

[33] G. Vass, "Applying and removing lens distortion in post production," 2003.

[34] A. Rosebrock, "OpenCV People Counter," https://pyimagesearch.com/2018/08/13/opencv-people-counter/, 2018, [Online; accessed 12- Aug- 2022].

[35] Z. Duan, M. O. Tezcan, H. Nakamura, P. Ishwar, J. Konrad, "RAPiD: Rotation-Aware People Detection in Overhead Fisheye Images," 2020.

[36] T. Scheck, R. Seidel, G. Hirtz, "Learning from THEODORE: A Synthetic Omnidirectional Top-View Indoor Dataset for Deep Transfer Learning," 2020 IEEE Winter Conference on Applications of Computer Vision (WACV), 2020, doi:10.1109/wacv45572.2020.9093563.

[37] B. E. Demiroz, I. Ari, O. Eroglu, A. A. Salah, L. Akarun, "Feature-based tracking on a multi-omnidirectional camera dataset," in 2012 5th International Symposium on Communications, Control and Signal Processing, 1–5, 2012, doi:10.1109/ISCCSP.2012.6217867.

[38] R. B. Knapp, N. F. Polys, J.-B. Huang, A. Ibrahim, N. Ma, C. Hurt, Y. xiao, "MW-18Mar Dataset," .

[39] U. P. d. M. G.-U. Grupo de Tratamiento de Imágenes, "PIROPO Database: People in Indoor ROoms with Perspective and Omnidirectional cameras," .

[40] S. Manen, M. Gygli, D. Dai, L. V. Gool, "PathTrack: Fast Trajectory Annotation with Path Supervision," 2017.

[41] R. Stiefelhagen, K. Bernardin, R. Bowers, J. Garofolo, D. Mostefa, P. Soundararajan, "The CLEAR 2006 Evaluation," in R. Stiefelhagen, J. Garofolo, editors, Multimodal Technologies for Perception of Humans, 1–44, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[42] H. S. P. Brauer, Camera based Human Localization and Recognition in Smart Environments, Ph.D. thesis, University of the West of Scotland, 2014.

[43] B. Demiroz, A. Salali, L. Akarun, "Multiple person tracking using omnidirectional cameras," 1231–1234, 2014, doi:10.1109/SIU.2014.6830458.

[44] G. Gemignani, BTLD+:A Bayesian Approach to Tracking Learning Detection by Parts, Ph.D. thesis, UNIVERSIT'A DEGLI STUDI DI MILANO, 2013.

[45] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, "Generative adversarial nets," in Advances in neural information processing systems, 2672–2680, 2014.