



# Visual and automatic bus passenger counting based on a deep tracking-by-detection system

Claire Labit-Bonis, Jérôme Thomas, Frédéric Lerasle

## ► To cite this version:

Claire Labit-Bonis, Jérôme Thomas, Frédéric Lerasle. Visual and automatic bus passenger counting based on a deep tracking-by-detection system. 2021. hal-03363502

**HAL Id: hal-03363502**

**<https://hal.science/hal-03363502v1>**

Preprint submitted on 4 Oct 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Visual and automatic bus passenger counting based on a deep tracking-by-detection system

Claire Labit-Bonis  
LAAS-CNRS  
ACTIA Automotive  
Toulouse, France

claire.labit-bonis@actia.fr

Jérôme Thomas  
ACTIA Automotive  
Toulouse, France

jerome.thomas@actia.fr

Frédéric Lerasle  
LAAS-CNRS  
Université UPS  
Toulouse, France

lerasle@laas.fr

## Abstract

*In this paper, we address the industrial constraints of automatic passenger counting in city buses through a deep architecture able to deal with images taken from low cost 2D cameras placed above the doorstep, from a zenithal point of view. The challenge is then to handle highly variable scenes due to passengers appearance (hair color, hats, height), bus population density at rush hour and changes in scene illumination. The scientific breakthrough related to deep learning applied to computer vision as well as the system embedding requirements for this task motivate us to integrate in this context a lightweight convolutional multi-object tracker which was especially designed for embedded applications and performed well on the MOT Challenge. We here evaluate it in an industrial context on our large scale in-situ dataset, labelled for detection, multi-target tracking and counting, and present a complete and embedded counting system meeting the requirements of our application.*

## 1. Introduction

In the literature, people counting is divided into two main areas: "region of interest" or ROI counting, and "line of interest" or LOI counting. The former consists in estimating the number of people present in an image at a given time, while the latter considers the individual and temporal counting of people crossing an artificial line in the image. For a public transport operator, counting passengers on buses allows both to establish statistics on bus lines occupancy and to estimate the fraud rate w.r.t. ticketing. To get the accurate and individual count required by these tasks, we place ourselves from the LOI counting perspective.

In order to avoid all the problems of occultation that can occur with ordinary perspective view surveillance systems, the existing precision counting sensors are generally placed from a zenithal point of view *i.e.* above each of the doors

to be monitored. The technologies used are mainly either stereoscopic vision sensors with depth reconstruction from two images taken from different points of view (passive 3D sensors), or infrared sensors that pulse light beams on the scene and calculate the return time of flight of the beam to estimate the depth of the elements in the observed scene (active 3D sensors). Despite the ease of acquisition of RGB images which would make the 2D modality a very attractive alternative as presented by Sun *et al.* [30] in their study on LOI counting, 2D image sensors are rarely used for passenger counting on the public transport commercial market. However, the technological breakthrough related to deep learning methods and convolutional neural networks (CNNs) applied to computer vision in recent years motivate us to develop a low cost system based on 2D cameras and offering counting performance equivalent to the market.

In addition to counting performance, we must meet industrial requirements such as (i) a reasonable cost for the client, but also (ii) on-board and near real-time processing – the videostream of passengers getting in/out of the bus can be analyzed between two bus stops. From this perspective, the recent and constant evolution of embedded hardware platforms designed for deep learning approaches is a real game changer: we can imagine a whole material architecture suited for this kind of method, able to handle several doorsteps/cameras at the same time, in parallel, as opposed to existing decentralized solutions with independent sensors and processing units for each doorstep to consider – thus multiplying the final costs.

One way to address the 2D visual counting problem is then to perform a frame-by-frame visual analysis of the scene in a multi-object tracking (MOT) framework. The vision community is very active on this problematic, and the detection-based tracking paradigm is widely used to tackle this challenge. A single image counting process based on one-shot detection is not suited for LOI counting because single image detection cannot infer if people are getting

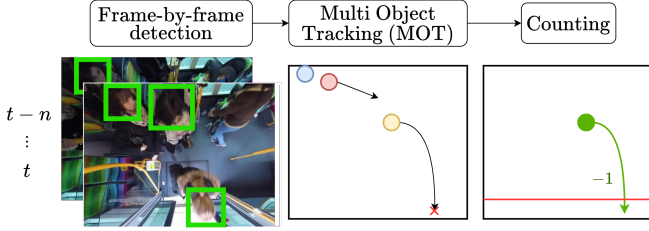


Figure 1. Online multi-object tracking-by-detection applied to passenger counting. A detector locates every object in the scene on a frame-by-frame basis, then a multi-object tracker links these detections together in order to reconstruct the distinct passengers trajectories. Finally, a tracklet crossing the doorstep line is counted in/out depending on its direction.

in/out of the bus – to this aim, knowing the passenger direction is essential. Moreover, a temporal analysis *i.e.* targets tracking, is more robust since it can catch up on the single image detector mistakes by filtering false positives and filling false negative gaps over time. On another note, where conventional motion flow analysis methods such as optical flow would have difficulty distinguishing passengers at rush hour when they are all moving together, tracking-by-detection allows to precisely detect each target at each instant, and to link these detections through time to reconstruct individual trajectories as it is illustrated by figure 1.

In this context and based on the industrial embedded constraints of our overall counting application, we integrate and evaluate the compact siamese multi-object tracker proposed by Labit-Bonis *et al.* [18]. Because their work was specifically motivated by the performance/speed trade-off required by such embedded applications, it is well-suited to our context as they proposed a lightweight architecture, benefiting from CNNs and which performed great in terms of tracking performance and speed on the well-known public benchmark for multiple people tracking, namely the MOT Challenge [19].

As a recap, our contributions are three-fold, with:

- the integration of a recent, compact and embeddable CNN-based siamese tracker which performed well on the MOT Challenge, into a 2D visual counting system thus benefiting from the recent breakthrough of deep learning for computer vision tasks;
- the description of the large scale bus passenger *in situ* dataset that we acquired in a city bus under operational circumstances both at off-peak and rush hour, at different times of the day, and that we annotated for detection, tracking and counting;
- a tracking and counting performance evaluation of the overall application on this dataset, along with em-

bedded considerations and preliminary results for the whole algorithm porting.

In the following, we propose a review of existing 2D visual methods applied to people counting in section 2. In section 3, we then describe the CNN-based siamese multi-object tracker integrated in the overall system [18], before delving into the industrial context of passenger counting with the presentation of our dataset in section 4. Finally in section 5, we evaluate our approach both in terms of tracking and counting metrics on this dataset, after what we show preliminary results towards the embedding of the final counting system.

## 2. Related work

2D vision and deep learning in LOI counting is rarely used by bus line operators. However, the considerable gains brought by these techniques on the problem of multi-target tracking and object detection show a real interest in evaluating their contribution in the industrial context of passenger counting. Several visual methods exist, though, and have been applied to passenger counting for image pre-processing, person detection or tracking and counting algorithms, but few propose to use deep learning and convolutional neural networks. In this section, we propose a review of the existing 2D vision systems for people counting in the literature, as well as an overview of the online multi-object tracking-by-detection techniques.

### 2.1. Visual passenger counting with 2D images

**In the context of city buses.** In the operational context of city buses, works are usually carried out on private image databases, in zenithal view. In 2008, Chen *et al.* [8] performs segmentation and region merging based on motion vectors extracted from the image. In 2012, Chen *et al.* [9] opt for circular shape detection using Canny filtering and a Hough transform, as well as colorimetric filtering on hair color. The detected circles are then compared with those of nearby instants to infer the passengers trajectories. In 2016, Perng *et al.* [25] apply background subtraction as well as morphological operators to focus on moving areas of interest, before performing the actual person detection *via* the correspondence between the region under consideration and an upstream defined head/shoulder geometric profile. In 2017, Liu *et al.* [20] use a mixture of Gaussians and background subtraction for target pre-localization, and then apply a CNN for automatic extraction of passenger visual features. They also mitigate target tracker drift by progressively updating a pheromone map capturing the memory of passing trajectories. In 2019, Nakashima *et al.* [24] achieve a 93.5% (resp. 36.1%) counting accuracy for people getting on (resp. off) of a bus on their dataset taken from a perspective view, by using a tracking-by-detection approach based

on YOLO [26] as the detector and DeepSORT [34] as the tracking method. They robustify their counting accuracy by adding a Random Forest Regressor dealing with additional data based on GPS and vehicle speed.

**Out of bus context.** Other works in perspective and/or zenithal view propose for example to manually extract visual features through HOG and LBP [36, 16, 1] before passing them to an SVM classifier. In addition, Zeng *et al.* [36] then use a particle filter for target tracking, Kocamaz *et al.* [16] associate detections with existing trajectories based on the spatial proximity and motion fluxes of the targets, García *et al.* [14] use a Kalman filter to predict the position of the targets prior to the measurements acquisition.

## 2.2. Online multi-object tracking-by-detection

Visual and online MOT considers sequences of successive 2D images. The targets positions are generally represented by a bounding box around them at each image instant, expressed in pixel coordinates *e.g.*  $[x, y, width, height]$  and to which a unique identifier related to the considered target is assigned. The tracking-by-detection paradigm is commonly used in 2D visual tracking and consists in **(i) locating the targets** present in each image *via* a dedicated detection algorithm and **(ii) linking these inter-image observations** by specifically identifying them. The tracking objective is twofold: to ensure the spatio-temporal coherence of the targets' trajectories, and to compensate for the errors of the detection algorithm which can generate false positives (object detected but not present in the image), false negatives (object present in the image but not detected) and imperfect measurements. In this context, detectors and trackers are generally evaluated separately on dedicated benchmarks such as ImageNet [28] and the MOT Challenge [19].

**Detection.** For a long time and before the advent of deep learning, object detection was done by means of a window of fixed size browsing the entire image in small regular steps and with overlap, which were each one going through a feature extraction and classification process before the gathering and filtering of every confident window to produce the final bounding box predictions. Since the formalization of the detection task within CNNs and the evolution of graphic cards specifically well designed for this kind of structure, all detectors today replace this slow sliding window process by using "region-based" and "one-shot" convolutional architectures such as Faster R-CNN [27], SSD [21] or YOLO [26]. They take an image as input and process it once, in its entirety, to directly produce as network outputs the box coordinates and classification scores for every detected object. SSD and YOLO both stand out from the rest

by their architecture compacity and their speed of execution, well suited for our application requirements, while still providing for good detection performance.

**Tracking.** At each image instant and after the detector is applied, detection outputs are given to the tracking algorithm, which creates new trajectories when objects are detected, predicts the current position of existing tracklets, and associates them with the remaining detections. Various methods exist for predicting, associating, and managing the tracklets status. For prediction, some of them use simple linear velocity models [4], Kalman filters [10, 34] or single object convolutional trackers [11, 13]. The vast majority of the approaches submitted to the MOT Challenge also use CNNs for appearance description within the association process [10, 34, 11, 13, 15, 35]. However, even for methods using CNNs for position prediction or appearance feature extraction, networks are trained and executed separately for the two different tasks, thus multiplying the execution time of the overall process. To overcome this issue, Labit-Bonis *et al.* [18] proposed a tracking method meeting the industrial requirements presented in section 1, by combining these two tasks into a single, unified and compact siamese convolutional architecture for one-shot regression and reidentification, demonstrating great speed and tracking performance on the MOT Challenge w.r.t. the literature. The tracker formalization will be further summarized in section 3.

This paper focuses on the behavior of this tracker within our industrial passenger counting system, coupled with two recent detectors of the literature: SSD, and YOLO. As a baseline to characterize the contribution of the tracking method, we also compare these couplings with DeepSORT [34] in place of the tracking technology brick: this popular <sup>1</sup> method uses a Kalman filter for position prediction and a CNN for appearance feature extraction.

## 2.3. Publicly available datasets

Since 2019, a large public database of bus passenger counting sequences, PCDS (for *Passenger Counting Data Set*) has been made available to the scientific community by Sun *et al.* [30]. However to our knowledge, most of the image databases dedicated to tracking or counting concern plunging or frontal views [7, 6, 23, 22] and there is still no public dataset annotated for the three tasks of detection, multi-object tracking and counting within this applicative context in a zenithal view. In section 4, we focus on our large scale *in situ* dataset, on which we evaluate the tracking system in section 5.

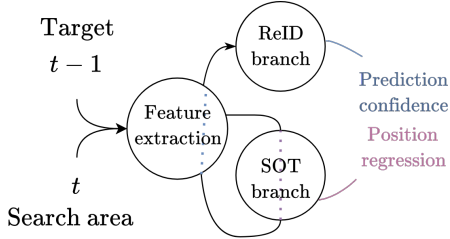


Figure 2. Tracker synopsis [18]. From two images cropped around targets position at  $t - 1$  and  $t$ , the tracker both regresses their position offset within the search area, and a reidentification score between the current target appearance and the predicted one, which can be interpreted as the prediction confidence.

### 3. Tracker overview

Labit-Bonis *et al.*'s work start from the two observations mentioned before. First, within the MOT context, many approaches use CNNs to efficiently describe targets, but only a few use them for the position prediction phase, even though CNN-based single object trackers (SOT) are among the best on the VOT Challenge [17]. Second, when trackers do use CNNs both for prediction and re-identification within MOT [11, 13], they show great tracking performance but their processing still remains cumbersome: different networks are used for both tasks, they are trained independently and executed in sequence.

**All-in-one architecture.** This single object tracker for position prediction is integrated within the MOT context and enriched with a reidentification purpose in a single, compact and unified architecture by taking advantage of Siamese networks. These structures are multi-input architectures with layers sharing all or part of their weights on each entry. Initially formulated for similarity/dissimilarity feature learning for verification applications [31, 29], they are now widely used by the visual tracking community for target visual motion prediction and have shown great progress on the VOT Challenge in particular.

The Siamese-based tracker proposed in the paper [18] consists of three parts: the joint visual feature extraction, the inter-frame target offset prediction and the prediction similarity computation *via* the generation of a reidentification descriptor. Figure 2 illustrates the high-level architecture and highlights the joint processing of regression and reidentification in a two-headed system.

From two enlarged  $t - 1$  and  $t$  image patches cropped around the target position at time  $t - 1$ , the network first infers the target position offset between the two instants through the SOT branch, and then produces a reidentification score between the current target appearance and the one at the predicted position after retrieving the feature maps

subsets thanks to a ROI Align layer. Based on this prediction confidence score, they are then able to adopt a strategy which keeps confident targets alive even if no detection has been made or associated. They thus fill the gaps produced by missing detections and show a good robustness to occlusions.

**Tracking performance on the MOT Challenge.** In 2021, the authors presented state-of-the-art tracking performance of their siamese-based tracker on the MOT Challenge SDP test set, comparatively to the most recent approaches of the literature at the time such as MOTDT [13], FAMNet [11], Tracktor++ [2] or DASOT [12]. Beyond tracking performance, they also exhibited a 2 to 17 times speedup over the approaches with the best MOTA.

Based on these promising results both in terms of tracking performance and speed, we want to integrate and evaluate it within the industrial context of passenger counting on our dataset we present hereafter.

### 4. Large-scale *in-situ* passenger dataset

**Dataset description.** For our evaluations, we recorded a video database by placing a GoPro camera above the central door of a bus in operation. These videos are acquired at different times of the day to capture the variability of the perceived scenes (day/night, bus congestion, shape/appearance of passengers). On these recordings, only some scenes contain people (at the terminus, the bus is waiting for its departure with closed doors, without passengers on board): only these sub-sequences are used for detection and tracking. For the counting evaluation, we then select only the moments between opening and closing of the bus doors at the stops. In addition to this segmentation, all videos go through a calibration phase to linearize the distortion induced by our on-boarded fish-eye camera.

**Detection, Tracking, and Counting Annotations.** We annotate our database for the three evaluation objectives of detection, multi-target tracking, and counting. Detection and tracking are annotated using the VATIC tool [33]; all the sequences details are given in Table 1, and illustrated by Figure 3. We estimate the annotation time for detection and tracking alone to be about 70 effective hours. Counting is evaluated only on the sequences of opening and closing of doors *i.e.* 100 sub-sequences accounting for 118 ascents and 239 descents. For each sub-sequence, we store the ground truth in a `.txt` file indicating the number of ascents and descents.

**Dataset splits.** For each experiment, the sequences Seq-2, Seq-3 and Seq-4 are used for training, Seq-1 and Seq-5 for validation, and finally Seq-6 for testing.

<sup>1</sup>  $\approx 1000$  citations.



Seq.	Images	$\neq$ ID	Labels	Clutter	Illumination
Seq-1	9,751	45	10,017	Medium (max 4)	Daylight
Seq-2	15,382	47	15,724	Medium (max 3 + seats)	Late + Artificial light
Seq-3	18,427	80	16,773	Large (max 6 + seats)	Daylight
Seq-4	29,889	96	33,947	Small (max 3)	Daylight
Seq-5	20,353	48	69,493	XLarge ( $\approx 10$ + seats)	Night + Artificial light
Seq-6	11,576	38	10,848	Small (max 2)	Night + Artificial light
$\approx 1h$	<b>105,378</b>	<b>354</b>	<b>156,802</b>		

Table 1. Passenger dataset description: there are several scenarios with more or less congestion (standing and sitting passengers) as well as changing illumination (day, night, with or without artificial lighting).



Figure 3. Illustration of the variability of the scenes in our database in terms of lighting, passenger appearance and environment clutter.

This distribution is chosen in order to have sequences of all types both in training and validation/testing sets *i.e.* with different illumination conditions and scene clutter.

## 5. Evaluations & discussion

### 5.1. Implementation details

The counting prototype in its first version (SSD + DeepSORT, as described in section 2) is embedded and functional on an NVIDIA Jetson TX2 development kit. At the time of the porting choice, NVIDIA was the only manufacturer on the market and the TX2 had the necessary capabilities for handling three video streams in parallel and running the considered networks.

The system in its latest version *i.e.* with the YOLO detector and the tracker from [18], is currently running on a fixed development PC equipped with a Titan Xp. Even if the official branch of YOLO is the one developed by Alexey Bochkovskiy [5], we chose the implementation of "YOLOv5", released in August 2020 by Ultralytics<sup>2</sup> because of its ease of use and portability. Precisely, its porting is in progress and the arrival of potential processing units such as Intel Myriad V2 or Google Coral motivates us to consider a new platform, better adapted than the Jetson TX2 to the industrial constraints stated above. We will further discuss and show preliminary results of the counting system on these platforms at the end of this section.

### 5.2. Tracking performance analysis

#### 5.2.1 Tracking metrics

As it is done in the MOT Challenge, we evaluate the tracker with the CLEARMOT [3] metrics: false positives FP, false

negatives FN, identity switches IDS, percentage of mostly tracked (MT) and mostly lost (ML) targets, fragmentations (FM) and most importantly the accuracy of the multi-object tracking MOTA which combines these metrics and gives relevant insight into the overall tracking performance.

#### 5.2.2 Tested configurations

In order to quantify the contribution of our method, we compare several variants in Table 2, in an incremental way:

- (A) *SSD + DeepSORT*: as stated in section 2, this coupling is used as a reference;
- (B) *SSD + [18]*: to illustrate the contribution of the lightweight siamese tracker, the DeepSORT tracker is replaced by [18];
- (C) *YOLOv5 + DeepSORT*: replacing SSD by YOLOv5 and keeping DeepSORT as a tracker shows the impact of the detection performance;
- (D) *YOLOv5 + [18]*: finally, we replace the tracker by [18] again to show its influence.

#### 5.2.3 Results interpretation

**Tracker [18] contribution: (A) vs. (B)** Considering exactly the same detections from SSD, the application of the new tracker instead of DeepSORT gains respectively +1.6%, +6.4% and +4.5% MOTA points on the three sequences, the most notable contribution being on the most crowded sequence, Seq-5. For SSD, which generates many false negatives, the trajectory keep-alive strategy presented in section 3 reduces them considerably (−431, −6581 and −937 decrease on FNs compared to SSD + DeepSORT). The visual position prediction strategy also

<sup>2</sup>Source: <https://github.com/ultralytics/yolov5>.

	Method	IDF1	Prcn	Rpl	GT	MT	ML	FP	FN	IDS	FM	MOTA
Seq-1	(A) SSD + DeepSORT	77,7	88,6	89,1	45	68.9	8.9	1151	1090	10	69	77,5
	(B) SSD + [18]	84,9	86,7	93,4	45	80.0	4.4	1433	659	5	22	79,1
	(C) YOLOv5 + DeepSORT	78.8	92.3	95.8	45	88.9	4.4	805	422	15	15	87.6
	(D) YOLOv5 + [18]	79.0	90.3	95.6	45	88.9	4.4	1044	346	11	6	86.0
Seq-5	(A) SSD + DeepSORT	30,6	92,1	55,6	48	25.0	22.9	3299	30860	233	1237	50,5
	(B) SSD + [18]	31,6	89,3	65,1	48	47.9	20.8	5432	24279	222	297	56,9
	(C) YOLOv5 + DeepSORT	61.3	92.6	93.0	48	70.8	10.4	5185	4897	123	414	85.3
	(D) YOLOv5 + [18]	65.9	91	95,1	48	85.4	8.3	6514	3394	112	143	85,6
Seq-6	(A) SSD + DeepSORT	51,7	90,9	55,64	38	36.8	13.2	604	4812	27	175	49,8
	(B) SSD + [18]	51,4	86,8	64,3	38	60.5	10.5	1061	3875	25	31	54,3
	(C) YOLOv5 + DeepSORT	77.7	96.6	83.4	38	81.6	7.9	317	1806	17	53	80.3
	(D) YOLOv5 + [18]	74.0	94.8	85.3	38	86.8	7.9	507	1598	17	35	80.4

Table 2. MOT evaluations of the different trackers on our dataset. (B) and (D) illustrate the interest of using the siamese tracker from [18], compared to the original configuration (A). (C) and (D) integrates YOLOv5 as a detector in replacement of SSD.

catches detection artifacts better than DeepSORT’s Kalman position prediction: trajectories are less fragmented (decrease of  $-47$ ,  $-940$  and  $-144$  on FMs), and better tracked (gain of  $+11.1\%$ ,  $+22.9\%$  and  $+23.7\%$  in MT).

**Importance of the detector for our application: (C) vs. (D)** The evaluation of the system against DeepSORT with YOLOv5 in detection instead of SSD shows the importance of the detection quality in the context of a zenithal view. Compared to the MOT Challenge for which the authors show the effectiveness of their approach in these complex situations, the zenithal view drastically reduces the number of occlusions: the application of a powerful detector like YOLOv5 mechanically reduces the potential gain brought by the tracker. However, we still observe a reduction of fragmentations, identity switches and a better coverage of the tracked targets.

### 5.3. Counting performance analysis

#### 5.3.1 Counting metrics

As for detection and tracking, an automatic passenger counting system can generate two types of errors within a full counting sequence at a bus stop during the doors opening: false positives (FP) and false negatives (FN). When the automatic count of people is greater (resp. lower) than it should be, it generates FP (resp. FN).

Few studies formalize precisely the metric used for passenger counting, but it is nevertheless common to see two types of errors advertised on the industrial market:

- the “compensated” error, which is based on the overall number of people on the bus, and for which the errors can cancel each other out<sup>3</sup>;

<sup>3</sup>A false negative (resp. positive) on the way up can be compensated for by a false negative (resp. positive) on the way down, or in the same direction of travel, a false positive catches up with a false negative.

- the “uncompensated” counting error where all the errors made on the count (false positives and false negatives) are cumulated in the metric.

A document published in 2018 by the German industry forum of transport companies VDV gives recommendations for the evaluation of automatic passenger counting systems [32].

In particular, it sets the measure of overall error with compensation as the difference between  $P_a$  the number of people counted automatically by the system and  $P_m$  the number of passengers counted manually *i.e.* the ground truth (cf. *equation 1*). This measurement is applied distinctly to the upward and downward directions; the compensation is then done in the same direction – a false positive makes up for a false negative.

$$E_{g(lobal)} = \frac{|P_a - P_m|}{P_m} \quad (1)$$

We can formulate this metric in terms of true/false positives and false negatives by the *equation 2*, considering  $P_a = FP + TP$ ,  $P_m = GT = FN + TP$ , and therefore :

$$E_g = \frac{|(FP + TP) - (FN + TP)|}{FN + TP} = \frac{|FP - FN|}{FN + TP} \quad (2)$$

We propose to formalize the cumulative error  $E_c$  without compensation by the *equation 3*, which accounts for the totality of errors w.r.t. the groundtruth.

$$E_c = \frac{FP + FN}{FN + TP} \quad (3)$$

For the sake of interpretability, we express the performance of the counting system in terms of count rate without/with compensation  $T_c = (1 - E_c) \times 100$  and  $T_g = (1 - E_g) \times 100$ .

Method	Up : GT = 118					Down : GT = 239					Global	
	FP	FN	TP	$T_c$	$T_g$	FP	FN	TP	$T_c$	$T_g$	$T_c$	$T_g$
SSD + DeepSORT	5	19	99	79.7%	88.1%	21	12	227	86.2%	96.2%	84.0%	93.6%
YOLOv5 + [18]	10	7	111	<b>85.6%</b>	<b>97.5%</b>	11	19	220	<b>87.5%</b>	<b>96.7%</b>	<b>86.8%</b>	<b>96.9%</b>

Table 3. Final comparison of the complete counting system between the SSD + DeepSORT combination and YOLOv5 + [18].

	Detection	Tracking + Counting	Resolution	Training framework	Optim.	Platform	FPS
SSD + DeepSORT	✓	✓	$360 \times 480$	Tensorflow	-	Jetson TX2	~20-23
YOLOv5 + [18]	✓	✓	$360 \times 480$	PyTorch	-	Titan Xp	~31-37
YOLOv5	✓	-	$192 \times 224$	PyTorch	TensorRT	Jetson TX2	~40*
YOLOv5	✓	-	$192 \times 224$	PyTorch	OpenVINO	Intel Core i5	~50-70

\*For a 3 images batch.

Table 4. Pre-port evaluations of the system. We evaluate SSD + DeepSORT on Jetson TX2, YOLOv5 + [18] on Titan Xp, as well as the optimization with TensorRT and OpenVINO of YOLOv5 alone on TX2 and on Intel Core i5.

### 5.3.2 Quantitative results

Table 3 contains the count rates without/with compensation for ups, downs, and for the overall count. In all cases, the final version performs better than the SSD + DeepSORT combination with gains of +5.9%, +1.3% and +2.8% (resp. +9.4%, +0.5% and +3.3%) on the count rates without compensation (resp. with compensation) in ascent, descent and overall.

### 5.3.3 Qualitative results

A comprehensive analysis of count errors on the SSD + DeepSORT method shows that among the count errors, about : (i) 58% are related to false detections resulting in the creation of false trajectories, (ii) 19% are caused by non-detections and therefore uncounted trajectories, (iii) 23% are due to bad associations from DeepSORT during tracking. The efficiency of the YOLOv5 detector as well as the better tracking performance brought by the siamese one allow to improve the counting performance for such types of errors.

### 5.3.4 Industrial requirements

The performance of our counting system is comparable to the best sensors on the market claiming a count rate of between 95 and 99% according to the VDV standard: we thus meet the specifications and present an industrially viable solution in terms of counting quality. Moreover, as we will show in the next section, our centralized architecture based on ordinary 2D sensors makes it possible to integrate the counting system while reducing the costs of the hardware targets.

## 5.4. Embedded prototype and preliminary results

In Table 4, we present the on-going evaluations of our system porting. In the first two rows, the complete counting system is evaluated with the detection, tracking and counting bricks, for an input image resolution of  $360 \times 480$ . The difference in FPS between SSD + DeepSORT and YOLOv5 + [18] can be explained by the fact that the Siamese tracker is applied on the patches at time instants  $t - 1$  and  $t$ , with a factor 2 to widen the search area, multiplying by 4 the amount of pixels to be processed. Even if the network used in [18] is lighter, it is executed a larger number of times and on larger images. However, in both cases the networks have not been transformed through TensorRT, so in addition to having more powerful platforms than the Jetson TX2, a margin of progress is possible thanks to the optimization phase during the embedding process.

As an example, we evaluate YOLOv5 alone on images whose size is divided by two, both on Jetson TX2 with TensorRT and on Intel Core i5 processor with OpenVINO (OpenVINO allows to optimize the networks for the Myriad V2 but more globally for any Intel processor, Core i5 included). The execution speed observed (respectively 40 FPS for 3 processed video streams and between 50 and 70 FPS for one stream) confirms that the porting of the new system is within our grasp while respecting the constraints of the specifications.

## 6. Conclusion

Automatic passenger counting in city buses is usually done thanks to independent processing units using 3D sensors placed above the doors. In this paper, we propose an embedded and centralized counting system using 2D cameras and achieving a count rate comparable to the best sensors on the market. To do so, we take advantage of recent



deep learning methods applied to computer vision and integrate a recent, compact and discriminative siamese multi-object tracker used for both position regression and reidentification, in the overall counting framework, enabling us to reconstruct passengers trajectories and count them as they cross the doorstep. We evaluate this method on our large scale bus passengers dataset, which we annotated for detection, tracking and counting, and demonstrate performance meeting the requirements both in terms of counting results and speed.

## References

- [1] I. Ahmed and A. Adnan. A robust algorithm for detecting people in overhead views. *Cluster computing*, 2018. 3
- [2] P. Bergmann et al. Tracking without bells and whistles. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2019. 4
- [3] K. Bernardin and R. Stiefelhausen. Evaluating multiple object tracking performance: the clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008. 5
- [4] A. Bewley et al. Simple online and realtime tracking. In *IEEE Int. Conf. on Image Processing (ICIP)*, 2016. 3
- [5] A. Bochkovskiy et al. Yolov4: Optimal speed and accuracy of object detection. *preprint arXiv:2004.10934*, 2020. 5
- [6] A. Chan et al. Analysis of crowded scenes using holistic properties. In *IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS)*, 2009. 3
- [7] A. Chan and N. Vasconcelos. Counting people with low-level features and bayesian regression. *IEEE Trans. on Image Processing (TIP)*, 2011. 3
- [8] C.-H. Chen et al. People counting system for getting in/out of a bus based on video processing. In *IEEE Int. Conf. on Intelligent Systems Design and Applications (ISDA)*, 2008. 2
- [9] J. Chen et al. Automatic head detection for passenger flow analysis in bus surveillance videos. In *IEEE Int. Congress on Image and Signal Processing*, 2012. 2
- [10] L. Chen et al. Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In *IEEE Int. Conf. on Multimedia and Expo (ICME)*, 2018. 3
- [11] P. Chu and H. Ling. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2019. 3, 4
- [12] Q. Chu et al. Dasot: A unified framework integrating data association and single object tracking for online multi-object tracking. In *AAAI Conf. on Artificial Intelligence (AAAI)*, 2020. 4
- [13] W. Feng et al. Multi-object tracking with multiple cues and switcher-aware classification. *preprint arXiv:1901.06129*, 2019. 3, 4
- [14] J. García et al. Directional people counter based on head tracking. *IEEE Trans. on Industrial Electronics*, 2012. 3
- [15] S. Karthik et al. Simple unsupervised multi-object tracking. *preprint arXiv:2006.02609*, 2020. 3
- [16] M. K. Kocamaz et al. Vision-based counting of pedestrians and cyclists. In *IEEE Winter Conf. on Applications of Computer Vision (WACV)*, 2016. 3
- [17] M. Kristan et al. The seventh visual object tracking vot2019 challenge results. In *IEEE Int. Conf. on Computer Vision Workshops (ICCVW)*, 2019. 4
- [18] C. Labit-Bonis et al. Compact and discriminative multi-object tracking with siamese cnns. *IEEE Int. Conf. on Pattern Recognition (ICPR)*, 2021. 2, 3, 4, 5, 6, 7
- [19] L. Leal-Taixé et al. Motchallenge 2015: Towards a benchmark for multi-target tracking. *preprint arXiv:1504.01942*, 2015. 2, 3
- [20] G. Liu et al. Passenger flow estimation based on convolutional neural network in public transportation system. *Knowledge-Based Systems*, 2017. 2
- [21] W. Liu et al. Ssd: Single shot multibox detector. In *European Conf. on Computer Vision (ECCV)*, 2016. 3
- [22] Z. Ma and A. Chan. Crossing the line: Crowd counting by integer programming with local features. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2013. 3
- [23] Z. Ma and A. Chan. Counting people crossing a line using integer programming and local features. *IEEE Trans. on Circuits and Systems for Video Technology (TCSVT)*, 2015. 3
- [24] H. Nakashima et al. Passenger counter based on random forest regressor using drive recorder and sensors in buses. 2019. 2
- [25] J.-W. Perng et al. The design and implementation of a vision-based people counting system in buses. In *IEEE Int. Conf. on System Science and Engineering (ICSSE)*, 2016. 2
- [26] J. Redmon et al. You only look once: Unified, real-time object detection. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 3
- [27] S. Ren et al. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems (NIPS)*, 2015. 3
- [28] O. Russakovsky et al. Imagenet large scale visual recognition challenge. In *Int. Journal of Computer Vision (IJCV)*, 2014. 3
- [29] F. Schroff et al. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015. 4
- [30] S. Sun et al. Benchmark data and method for real-time people counting in cluttered scenes using depth sensors. *IEEE Trans. on Intelligent Transportation Systems (ITS)*, 2019. 1, 3
- [31] Y. Taigman et al. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 4
- [32] V. D. Verkehrsunternehmen. Recommendations for the applications of apess within public transport and regional rail transport. 2018. 6
- [33] C. Vondrick et al. Efficiently scaling up crowdsourced video annotation. *Int. Journal of Computer Vision (IJCV)*, 2013. 4
- [34] N. Wojke et al. Simple online and realtime tracking with a deep association metric. In *IEEE Int. Conf. on Image Processing (ICIP)*, 2017. 3
- [35] J. Xu et al. Spatial-temporal relation networks for multi-object tracking. In *IEEE Int. Conf. on Computer Vision (ICCV)*, 2019. 3

- [36] C. Zeng and H. Ma. Robust head-shoulder detection by pca-based multilevel hog-lbp detector for people counting. In *IEEE Int. Conf. on Pattern Recognition (ICPR)*, 2010. 3