applied sciences

MDPI

Article

Convolutional Neural Networks Refitting by Bootstrapping for Tracking People in a Mobile Robot

Claudia Álvarez-Aparicio *, Ángel Manuel Guerrero-Higueras, Luis V. Calderita, Francisco J. Rodríguez-Lera, Vicente Matellán and Camino Fernández-Llamas

Department of Mechanical, Computer Science and Aerospace Engineering, Campus de Vegazana s/n, University of León, 24071 León, Spain; am.guerrero@unileon.es (Á.M.G.-H.); lv.calderita@unileon.es (L.V.C.); fjrodl@unileon.es (F.J.R.-L.); vicente.matellan@unileon.es (V.M.); camino.fernandez@unileon.es (C.F.-L.)

* Correspondence: calvaa@unileon.es

Abstract: Convolutional Neural Networks are usually fitted with manually labelled data. The labelling process is very time-consuming since large datasets are required. The use of external hardware may help in some cases, but it also introduces noise to the labelled data. In this paper, we pose a new data labelling approach by using bootstrapping to increase the accuracy of the PeTra tool. PeTra allows a mobile robot to estimate people's location in its environment by using a LIDAR sensor and a Convolutional Neural Network. PeTra has some limitations in specific situations, such as scenarios where there are not any people. We propose to use the actual PeTra release to label the LIDAR data used to fit the Convolutional Neural Network. We have evaluated the resulting system by comparing it with the previous one—where LIDAR data were labelled with a Real Time Location System. The new release increases the *MCC*-score by 65.97%.

Keywords: bootstrapping; convolutional neural networks; LIDAR; PeTra; re-training; robotics

check for updates

Citation: Álvarez-Aparicio, C.; Guerrero-Higueras, Á.M.; Calderita, L.V.; Rodríguez-Lera, F.J.; Matellán, V.; Fernández-Llamas, C. Convolutional Neural Networks Refitting by Bootstrapping for Tracking People in a Mobile Robot. *Appl. Sci.* **2021**, *11*, 10043. https://doi.org/10.3390/app112110043

Academic Editor: Alessandro Gasparetto

Received: 8 October 2021
Accepted: 24 October 2021
Published: 27 October 2021

1. Introduction

Mobile robots, especially social or assistive robots, coexist with people in the environment where they are deployed. Such robots need to be able to carry out some basic tasks. First, they need to know their position in the environment. Second, they have to move from one point to another autonomously, avoiding obstacles and without damaging people or objects. Finally, they interact with people and they even work with them on specific tasks. The first two skills have been extensively studied and developed in the literature, thus, today there are quite robust solutions. The third one is a slightly more complex skill and many studies are currently focusing on it.

The autonomous-behaviour generation in a robot faces several challenges. The robot not only has to be able to "survive" in any environment where it is deployed but also human–robot interaction has to be as similar as possible to human–human interaction. Interaction not only refers to communication, but it also has to do with navigation or obstacle avoidance. All these aspects relate to one single basic skill, people tracking, i.e., it is necessary to know where the people are every time.

Tracking people is not only useful to improve navigation skills in mobile robots but also to encourage socially acceptable robots. Many solutions in the literature attempt to solve this problem, typically using both vision and range sensors as shown in [1]. Some researchers focus on Convolutional Neural Networks (CNNs) since they provide a better generalization compared to traditional methods that use geometric features. It should be pointed out that most of the proposals use Red Green Blue Depth (RGB-D) cameras to detect people in the environment. For instance, the authors in [2] propose a solution based on a RGB-D camera combining RGB and depth data to gather input data for a segmentation CNN. Other researchers combine data from several sensors. For instance, in [3,4], authors propose a method to train a CNN with data provided from both Laser Imaging Detection and Ranging (LIDAR) sensors and cameras. In [5], the authors propose to combine 2D and 3D LIDAR data to train a Support Vector Machine (SVM) to detect pedestrians for autopilot systems.

However, the above approaches are both computing-demanding when running on-board a robot. Different solutions have been proposed in the literature to deal with the problem of tracking people using 2D LIDAR sensors despite their being traditionally used for autonomous navigation. Different methods for robot navigation in crowded indoor environments have been reviewed in [6]. The usage of the geometric features of the human legs, as well as the rate and phase of gait, was proposed in [7] to enhance people tracking systems. Furthermore, in [8], the clustering and centre point estimation combined with the walking centre line estimation, the speed, and the step length separation are used to detect people with or without a walker. However, these approaches are not robust enough when dealing with occlusions or changes in gait speed. To address such issues, we presented People Tracking (PeTra) in [9], a tool that allows to locate people within the robot surroundings using the information provided by a LIDAR and a CNN.

PeTra is used in this work to validate our proposal. The first release was presented in [10]. The system builds an occupancy map through a LIDAR sensor's readings. The sensor is located 20 cm above the floor. LIDAR's readings are processed by a CNN which returns a second occupancy map segmenting the readings belonging to people close to the robot. From the second occupancy map, a centre-of-mass calculation provides the people's location estimates. A new PeTra release was presented in [11] including not only a correlation method of location estimates for tracking people in time but also an optimized model for the CNN which allows the system to work in real time.

PeTra provides a good performance in the scenarios where it has been evaluated. However, its performance is worse in some specific locations, such as empty rooms without furniture; corridors; or scenarios with more than two people. In such cases, PeTra sometimes detects people where there are none, for instance, close to the walls. Thus, it may be convenient to refit the PeTra's CNN to achieve better performance in such environments.

To obtain a supervised learning model with high accuracy, it is necessary to provide a large volume of data at the training phase. Generally, CNNs have been fitted by using manually labelled data. The labelling process is very time-consuming. Some researchers automatically label data by using external hardware. External hardware may report positive results but it also introduces some constraints related to the number of required devices or their measurement error. Despite the above issues, a CNNs fitted with such labelled data usually provides acceptable results. A popular alternative is using bootstrapping techniques, which is somehow similar to self-training.

The use of CNNs has increased in recent years because of the increasing computing capabilities. They are applied in every environment, from research to industry. Depending on the task to be solved, supervised or unsupervised learning techniques may be used. We focus on supervised learning since it is the most popular approach. The training process requires a dataset gathering input data, as well as their expected output.

A fully functional network model requires a large volume of data to train it. The main issue has to do with gathering such a volume of data because a fussy labelling process is needed. Labelling is usually done manually, supervising every piece of input data and labelling their corresponding output. This process is very time-consuming. Some researchers propose to use external hardware to automatically label output data. These proposals have some constraints, on the one hand, it is very common that some "noise" is introduced because of the device's mean error. On the other hand, the hardware availability may be limited—for instance, if mobile transceivers are required for every people in the scene—and so, it could be impossible to label every piece of data correctly. To deal with the above issues, bootstrapping techniques can be useful.

Bootstrapping was first proposed in [12] applied to word-sense disambiguation using unlabeled samples and a few labelled samples. An initial classifying model was built using the labelled samples. Then the unlabeled samples were classified extracting new patterns that were used to build an enhanced classifier.

Bootstrapping has been applied in different research areas, for instance, in document analysis and recognition. The authors in [13] used bootstrapping to resolve segmentation problems in the processing of music scores getting a 99.2% classification accuracy. The authors in [14] proposed a scene text detection technique using bootstrapping and text border semantics for accurate localization of texts in scenes, getting an 80.1% f-score for the MSRA-TD500 dataset.

In healthcare, the authors in [15] point out issues related to the segmentation of craniofacial cartilage images. Labelling such images is very challenging since only experts can differentiate cartilages. The authors proposed to use self-training to fit a CNN and therefore achieve high segmentation accuracy. The authors in [16] present a new prediction approach for imbalanced DNA-protein binding data, they use a bootstrap strategy to under-sample the negative data to balance the number of binding and non-binding samples. Results demonstrate that the method achieves a high prediction performance and outperforms the state-of-the-art sequence-based DNA-protein binding predictors.

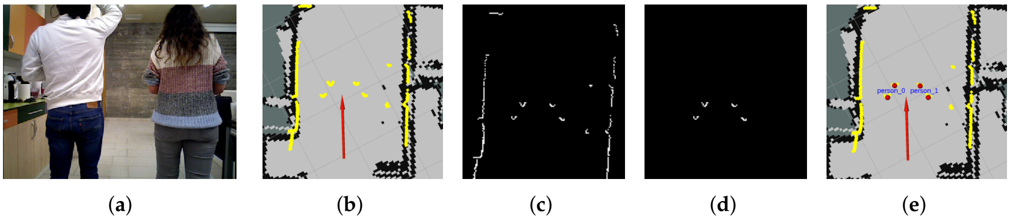
In agricultural engineering, the authors in [17] apply bootstrapping methods to refit a CNN which allows for segmenting plant sections. The proposed CNN was initially fitted with only 30 images manually labelled.

In industry, the study presented in [18] uses bootstrapping to predict the useful life of a rolling bearing. The performance increased significantly by testing through different datasets and compared to MSCNN-based, BLSTM-based, and MLP-based models.

In the robotics field, bootstrapping techniques have been applied in some research. The authors in [19] propose a novel pipeline for object detection using bootstrapping that improves 60-fold the training speed. They assess the effectiveness of the approach on a standard Computer Vision dataset (PASCAL VOC 2007 [20]) and demonstrate its applicability to a real robotic scenario with the iCubWorld Transformations [21] dataset. The authors of [22] use bootstrapping to create a method of teaching a Haru's robot its empathic behavioural response from its interaction with people. The results show that this technique is an efficient tool to speed up the robot's learning compared to the online learning method initially used.

In this work, we propose to use bootstrapping for refitting PeTra. It is important to point out that to fit PeTra's CNN, it is necessary to provide pairs of images. The first image shows a first occupancy map built from LIDAR readings (raw data), see Figure 1c. The second image shows a second occupancy map of the same location but built from the LIDAR readings only belonging to people's legs (labelled data), see Figure 1d. To obtain labelled data, a beacon-based Real Time Location System (RTLS) was used in the first version of PeTra [9]. The people in the scene carried a mobile transceiver that gathers the beacons' signal to estimate their location. The main drawback of using a RTLS to label data is related to the measurement error of the device. However, KIO has a ±30 cm average error, which is related to the identification problems of PeTra on specific locations. This paper poses a new data labelling method for refitting PeTra's CNN by bootstrapping. The CNN has been refitted with data labelled by PeTra itself. As a result, the CNN achieves higher accuracy.

The remainder of the paper is organized as follows: Section 2 describes the materials and evaluation methods used to carry out the research; results are presented and discussed in Sections 3 and 4, respectively; Finally, conclusions and future works are proposed in Section 5.



(a) (b) (c) (d) (e)

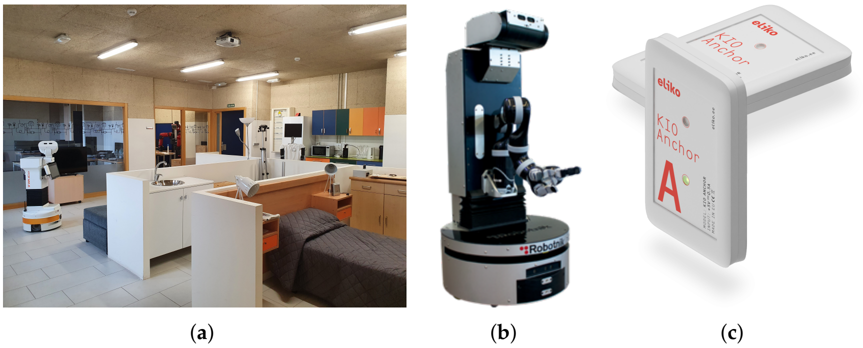
Figure 1. (**a**) Snapshot from the robot's camera. (**b**) LIDAR Readings visualized on Rviz. (**c**) Occupancy map built from LIDAR's readings (raw data). (**d**) Occupancy map returned by the PeTra's CNN (labelled data). (**e**) Location estimates calculated by PeTra.

2. Materials and Methods

A set of experiments has been carried out to evaluate the accuracy of the PeTra's refitted CNN. In this section, the main elements of the experiment are described in depth, as well as the methodology used to evaluate the accuracy of the refitted CNN.

2.1. Leon@Home Testbed

The experiments were conducted in the mock-up apartment known as Leon@Home Testbed [23], shown in Figure 2a, a certified testbed [24] of the European Robotics League (ERL) located in the Robotics Group's lab at the University of León. Its main purpose is to benchmark service robots in a realistic environment. The apartment is a single bedroom mock-up home built in an 8 m × 7 m space. Walls of 60 cm in height divide it into a kitchen, living room, bedroom, and bathroom.



(a) (b) (c)

Figure 2. (**a**) General view of the apartment. (**b**) Orbi-One robot. (**c**) KIO RTLS.

2.2. Orbi-One Robot

Orbi-One, shown in Figure 2b, is a service robot manufactured by Robotnik [25]. It accommodates several sensors such a RGB-D camera in the head, and a Hokuyo LIDAR sensor in its mobile base. It also has a six-degrees-of-freedom arm attached to its torso. Inside, an Intel Core i7 CPU with 8 GB of RAM allows it to run the Robot Operating System (ROS) framework [26] in charge of managing the robot hardware.

2.3. KIO RTLS

KIO, a commercial RTLS manufactured by Eliko [27] is shown in Figure 2c. It has been used for labelling data to later fit the PeTra's CNN for the first time. This RTLS system calculates the location of a mobile transceiver, usually called *tag*, in a two- or three-dimensional space. KIO uses Radio Frequency Identification (RFID) beacons, usually known as *anchors*, placed in known locations in the mock-up apartment of Leon@home Testbed. The anchor locations have been placed according to the method described in [28]. To fit PeTra's CNN, people in the scene carried a mobile transceiver that gathers the beacons' signal to estimate their location. Then, the second occupancy map was built from the first one by cropping the LIDAR readings away from people's legs, obtaining pairs of images similar to the ones shown in Figure 1c,d. However, location estimates provided by KIO have an average ±30 cm error according to the manufacturer's specifications. The evaluation carried out in [28] shows that the measurement error is higher in some areas and lower in others; however, on average, the claims of the manufacturer are correct.

2.4. PeTra

Figure 1 illustrates Petra's running, from the time the robot gathers data from its LIDAR sensor until PeTra locates people around the robot. PeTra first builds a 2D occupancy map from all the readings of the LIDAR, see Figure 1c. This first occupancy map is processed by a CNN that returns a new occupancy map including only the readings belonging to people leg-like patterns, see Figure 1d. The CNN used by PeTra is based on the U-net architecture [29]. The architecture was originally designed to perform the biomedical image segmentation [30]. Postprocessing the output of PeTra's CNN, a centre-of-mass calculation provides people location estimates. Correlating the location estimates by using a Kalman filter allows for tracking each person in the scene over time. A video showing PeTra's operation is available online [31].

PeTra has shown good performance in the scenarios where it has been evaluated. However, its performance is worse in some specific locations, such as corridors. In these cases, PeTra sometimes detects people where there are not. Thus, it may be necessary to refit the PeTra's CNN to get better performance in such environments.

2.5. Data Gathering

The fitting of PeTra's CNN was carried out by using a public dataset known as RRID:SCR_015743 [32]. Data are available at the Robotics Group of the University of León's website [33]. The data were gathered on 14 different scenes [10]. In all of them, the Orbi-One robot stood still while one or more people moved around it. Three different environments of the mock-up apartment of Leon@home Tesbed were considered: the kitchen, the bedroom, and the living room.

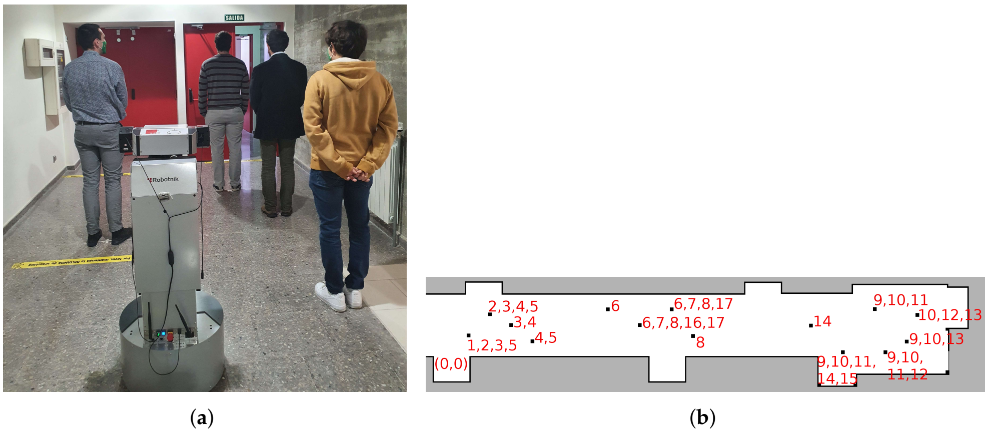
This RRID:SCR_015743 dataset is composed of two releases: the first release (v1) was released in November 2017, and the second one (v2) was released in February 2018. Both releases consist of *Rosbag* files, a ROS feature that allows for capturing the information gathered by the robot and recording it for further processing. The data for fitting PeTra's CNN have been gathered in the mock-up apartment of Leon@home Testbed.

The first release of RRID:SCR_015743 dataset contains 81 Rosbag files. It has been used to fit PeTra's CNN for the first time. In this release, the data contained in the Rosbag files were labelled by using KIO.

The second release of RRID:SCR_015743 dataset contains 42 Rosbag files. It has been used to refit PeTra's CNN. In this release, the data contained in the Rosbag files were labelled by using PeTra's CNN fitted with the first release of RRID:SCR_015743 dataset.

Moreover, a new dataset was created to evaluate PeTra's performance using both the CNN fitted with the first release of RRID:SCR_015743 dataset (labelled by KIO), and the CNN refitted with the second release of RRID:SCR_015743 dataset (labelled by PeTra).

The data for this dataset was gathered in the corridor of Leon@home Testbed, data are available online (DOI: 10.5281/zenodo.4541258) [34]. This dataset contains 25 Rosbag files, numbered 1–25, recorded in different locations with the Orbi-One robot standing still. Two types of Rosbag files were recorded. In 17 Rosbag files, numbered 1–17, people were standing still in the scene. They were placed in known locations to obtain ground-truth data. The locations where people were placed for each Rosbag file are shown in Figure 3b. An example of a real scene at gathering-data time is shown in Figure 3a. The remaining 8 Rosbag files, numbered 18–25, were recorded without people in the scene to evaluate the True Negatives rate.



(**a**) (**b**)

Figure 3. (**a**) The real scene at gather data time. (**b**) Map and people locations at dataset creation. The squares indicate the people's locations, and the numbers indicate the Rosbags where they appear.

2.6. Data Labelling by Bootstrapping

Supervised learning techniques require labelled datasets for fitting models. We can label data either manually, which is time-consuming; or automatically, which reduces time but may introduce measurement errors. So far, the KIO system was used to automatically label the training data; however, this method features several drawbacks described below. This paper proposes a new data labelling method, using the PeTra itself to label the data that will be later used to refit its CNN.

The main drawback we found when labelling with KIO has to do with the number of available tags. We have just two KIO tags. Thus, we can record Rosbag files with two people at most in the scene. In contrast, PeTra allows for locating all the people in the scene.

As mentioned in Section 2.5, PeTra was used to locate people in the scenes on the second release of RRID:SCR_015743 dataset. PeTra's location estimates are used to label occupancy maps. However, we have not used the entire dataset to refit the CNN. We have selected the scenes where PeTra showed the best performance. We discarded some Rosbag files for each scenario (see [10] for details): 2, 11, and 14 for the kitchen; 2, 7, 11, and 13 for the bedroom; and 2, 5, 9, 11, and 14 for the living room. The remaining Rosbag files were used to refit the CNN.

As mentioned above, raw occupancy maps are built from the points detected by the LIDAR sensor, see Figure 1c. PeTra's estimates contain three points, as shown in Figure 1e: two for the location of the people's legs and a third one for the people's centres. To label the occupancy maps, we "draw" a 15 cm circle around the people legs. We assume that the LIDAR readings inside those circles belong to people. These readings are used to build the second occupancy map, see Figure 1d.

2.7. Refitting Process

The CNN is the main component of PeTra allowing to locate people at the scene. The experiment proposed consists of fitting twice the CNN model used by the tool. The first fitting was done with the data labelled using the KIO RTLS devices. Thus, the first PeTra version is available and ready to use. This version has shown good performance but it is worse in some specific locations, such as empty rooms without furniture or corridors. In such cases, PeTra sometimes detects people where there are not, usually close to the walls.

Once the first version is available, PeTra is used to label data of a different dataset (v2), as is described in Section 2.5. The resulting dataset is used to perform the second fitting, getting in a new neural network model for PeTra.

2.8. Convolutional Neural Network Fitting

PeTra's CNN was fitted on Caléndula, the High Performance Computing (HPC) cluster located in Supercomputación Castilla y León (SCAYLE), which provides HPC services to the research centres and companies in Castilla y León, Spain. Caléndula has 345 servers (+7000 cores), 18.8 TB of memory and an overall computation performance of 397 TFlops (Rpeak).

Specifically, the fitting was carried out in a server with 2 Xeon E5-2695 v4 processors with 36 cores, 384 GB RAM, 2 hard drives of 200 GB each, Infiniband FDR 56 GB/s, and 8 Nvidia V100 GPUs.

The PeTra's CNN was developed by using the Keras API for Python and Tensorflow as backend. To fit the CNN, we need pairs of images. Each pair is composed of a raw occupancy map and its corresponding labelled occupancy map. In this case, 80% of the dataset was used to fit the CNN and the remaining 20% to test it. The CNN was trained for 30 epochs, with a *batch_size* value (number of images processed per iteration) of 128. The fitting process reports a precision score, consisting of the accuracy, and the loss, a scalar value that the fitting tries to minimize: the lower the loss, the higher the True Positive rate.

2.9. Evaluation

The evaluation was carried out in two ways. First, the accuracy score and loss value of both models were compared. The accuracy is a measure of how accurate the model's predictions are compared to the true data. The loss value is the sum of errors made for each example in training or validation sets. The models were trained with 80% of the total images that comprise the dataset and were tested through the 20% remaining.

Then a comparison of PeTra's performance was carried out using both the CNN fitted with data labelled by the KIO device, and the CNN refitted by bootstrapping. A new specific dataset was gathered for this purpose, as described in Section 2.5. Ground-truth data to evaluate the performance of both CNNs were obtained by locating people in the scenes in known positions, see Figure 3b.

To evaluate the precision, each Rosbag file is played twice: the first one by using PeTra with the CNN fitted with data labelled by the KIO device; and the second one by using PeTra with the CNN refitted by bootstrapping. For each running, we need to know whether or not all the people in the scene have been recognized properly. Specifically, we need to know how many people are in the scene, the number of people detected by PeTra, the number of people correctly detected, the number of people not detected by PeTra, and the number of people wrongly detected. Such data allow us to obtain the confusion matrix that allows visualization of the performance of our algorithm.

Moreover, to evaluate the overall performance of both CNNs the following Key Performance Indicators (KPIs) obtained through the confusion matrix are considered: Sensitivity (*Sen*), Specificity (*Spe*), Precision (*Pre*), Accuracy (*Acc*), and Matthews Correlation Coefficient (MCC). The *Sen* score—see Equation (1)—shows the rate of positive cases that were correctly identified by the algorithm. The *Spe*—Equation (2)—measures the proportion of negative cases that were correctly identified by the algorithm. The *Pre* score—see Equation (3)—shows the fraction of relevant instances among the retrieved instances. The *Acc*—see Equation (4)—measures the proportion of correct predictions both positives and negatives cases among the total number of cases examined. Finally, the *MCC*—see Equation (5)—is used as a measure of the quality, considering both true and false positives and negatives cases. It takes into account true and false positives and negative cases.

$$Sen = \frac{T_P}{T_P + F_N} \tag{1}$$

$$Spe = \frac{T_N}{T_N + F_P} \tag{2}$$

$$Pre = \frac{T_P}{T_P + F_P} \tag{3}$$

$$Acc = \frac{T_P + T_N}{T_P + F_N + T_N + F_P} \tag{4}$$

$$MCC = \frac{T_P \times T_N - F_N \times F_P}{\sqrt{(T_P + F_N) \times (T_P + F_P) \times (T_N + F_N) \times (T_N + F_P)}} \tag{5}$$

3. Results

PeTra's CNN was fitted according to Section 2.7. First, PeTra's CNN was fitted with data labelled by using the KIO device. The fitting dataset has 57,827 pairs of images. Of these images, 80% (46,261) were used to train the model, and the remaining 20% (11,566) to test it. The training process took 37 min and 8 s, yielding an accuracy score of 0.7668 and a loss value of −0.7678.

Next, PeTra's CNN was refitted by bootstrapping with data labelled by PeTra itself. The fitting dataset has 16,408 pairs of images. Out of the images, 80% (13,126) were used to train the model, and the remaining 20% (3282) to test it. The training process took about 10 min and 40 s, yielding an accuracy score of 0.8306 and a loss value of −0.8287.

The CNNs evaluation was carried out as explained in Section 2.9. Results are reported in Figures 4 and 5. Each data column represents the information of a Rosbag file. The blue bar represents the number of people that were in the scene (Actual people). The orange bar shows the number of people correctly detected (People correctly detected), the number of people not detected (People not detected) is presented in grey, and the number of people wrongly detected (People wrongly detected) is presented in yellow. From such data, Tables 1 and 2 show the confusion matrix for each CNN.

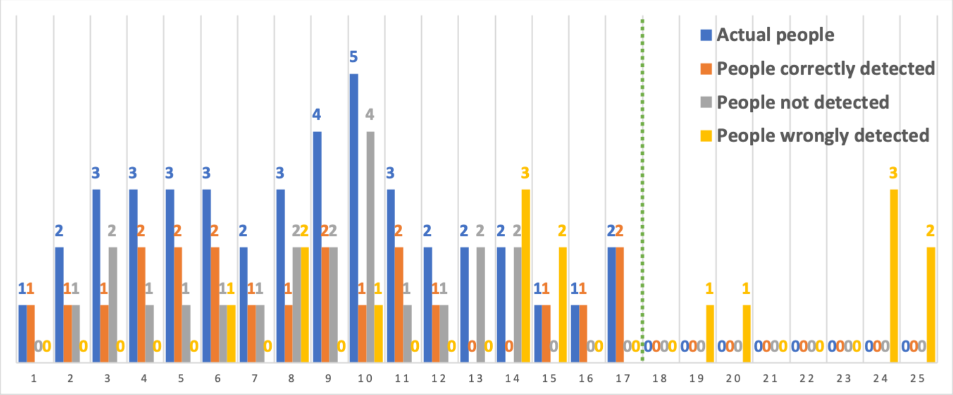


Figure 4. Performance of CNN fitted with labelled data by KIO (CNN$_{KIO}$).

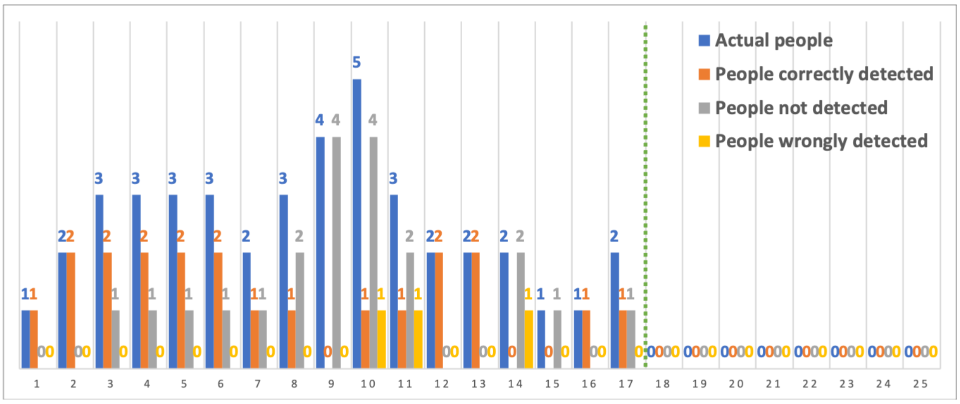


Figure 5. Performance of CNN refitted by bootstrapping (CNN$_{bootstrapping}$).

According to the values posed in Tables 1–3 show the Sensitivity (Sen), Specificity (Spe), Precision (Pre) Accuracy (Acc), and MCC for both the CNN fitted with labeled data by KIO and the CNN refitted by bootstrapping.

Table 1. Confusion matrix of the CNN fitted with labeled data by KIO (CNN$_{KIO}$).

		Prediction	
		No People	**People**
Actual	**No people**	4	16
	People	21	21

Table 2. Confusion matrix of the CNN refitted by bootstrapping (CNN$_{bootstrapping}$).

		Prediction	
		No People	**People**
Actual	**No people**	8	3
	People	21	21

Table 3. Sensitivity (*Sen*), Specificity (*Spe*), Precision (*Pre*), Accuracy (*Acc*), and MCC for both the CNN fitted with labelled data by KIO (CNN$_{KIO}$) and the CNN refitted by bootstrapping (CNN$_{bootstrapping}$).

Model	***Sen***	***Spe***	***Pre***	***Acc***	***MCC***
CNN$_{KIO}$	0.5676	0.1600	0.5000	0.4032	−0.2859
CNN$_{bootstrapping}$	0.8750	0.2759	0.5000	0.5472	0.1852

4. Discussion

The accuracy score of the CNN trained by applying the bootstrapping technique increased by 8.32% compared with the CNN trained with data labelled by KIO. In addition, the loss value has also increased by 7.93% compared to the training done with bootstrapping and the KIO.

The confusion matrices presented in Section 3 provide an overall view of the differences between the two CNNs. As mentioned in Section 2.9, the T_P values match the number of people correctly detected by both CNNs, specifically, 21 and 21, respectively. The F_P values match the number of people wrongly detected, there are not any people at the scene but the model detects someone, by both CNNs, specifically, 16 and 3, respectively. The F_N values match the number of people not detected by both CNNs, specifically, 21 and 21, respectively. This field has not improved because people are far away from the robot or the LIDAR data have partial occlusions from the scene. On the other hand, the T_N value matches the number of cases where PeTra does not detect any people and actually, there are no people in the scene. Such cases correspond to the scenes recorded in Rosbag files 18–25. According to Figure 4, there are 4 cases where the CNN fitted with KIO does not return people's location estimates on scenes where there are no people. Specifically, in the scenes recorded in Rosbag files 18, 21, 22, and 23. In the other four cases, the CNN fitted with KIO return people location estimates where there are none. For the CNN refitted by bootstrapping, Figure 5 shows that in all cases (8), the system does not return people location estimates for scenes where there are no people.

According to Table 3 both CNNs have the same Precision value ($Pre = 0.5$). However, the rest of the metrics are higher with the CNN refitted by bootstrapping. The sensitivity value—an important measure for imbalanced data—is considerably lower for the CNN fitted with labelled data by using the KIO device ($Sen = 0.5676$) compared to the CNN refitted by bootstrapping ($Sen = 0.8750$), this represents an increase of 54.16%. The Specificity value for the CNN refitted by bootstrapping ($Spe = 0.2759$) is 72.43% higher compared to the CNN fitted with labelled data by using the KIO device ($Spe = 0.1600$). The accuracy value is also lower for the CNN fitted with labelled data by using the KIO device ($Acc = 0.4032$) compared to the CNN refitted by bootstrapping ($Acc = 0.5472$), thus representing an increment of 35.71%.

Finally, the *MCC* score is the most important KPI to consider since it engages the rate of True Positives correctly detected with a low rate of False Positives. For the CNN fitted with data labeled by the KIO device the value is lower ($MCC = -0.2859$) than the CNN refitted by bootstrapping ($MCC = 0.1852$). The MCC ranges in the interval $[-1, +1]$, with extreme -1 and $+1$ values reached in case of fatal and perfect classification, respectively. This value is 65.97% higher for the CNN refitted by bootstrapping.

5. Conclusions

This paper presents a comparative study of fitting a CNN with data labelled by bootstrapping versus data labelled with an external ground-truth system. Specifically, this work compares the performance of the CNN used by the PeTra tool to track people close to a mobile service robot. CNN performance was evaluated with a new dataset gathered for such purpose. The CNN has been fitted twice: the first time by using training data labelled with the KIO RTLS device, and the second time by using the PeTra tool itself to label the training data.

We analysed the KPIs of both models. Results show that labelling with PeTra improves four out of the five KPIs described. Sensitivity in the CNN trained with bootstrapping increases by 54.16% compared with CNN trained by KIO. The Specificity is 72.43% higher in the CNN trained with by bootstrapping compared to the CNN trained with KIO. Furthermore, the Accuracy of the CNN trained with bootstrapping is 35.71% higher than the CNN trained with KIO. Finally, the MCC KPI is 65.97% higher in the CNN trained by bootstrapping compared to the CNN trained with KIO. Thus, this work allows to assert that bootstrapping increases the accuracy of a CNN-based people tracking tool. The improvement is significant in the cases where there are no people in the scene (Rosbags 8–25). In such cases, the CNN trained by bootstrapping does not detect any people where there are none. Therefore, using bootstrapping to label data is a good alternative to get a more accurate CNN. Moreover, this method is an especially interesting alternative in environments where a RTLS or a ground-truth system is not available.

Moreover, it is important to point out that PeTra allows to label data of all people in the scene, whereas the KIO device—as the most commercial RTLS—has a limited number of mobile transceivers. In addition, PeTra's CNN refitted by bootstrapping performs especially well in scenarios without people, thus avoiding false positives and any possible incorrect robot behaviour. Similarly, in scenarios where there are a larger number of people, it also improves significantly. Furthermore, our solution provides a labelling method suitable to use in scenarios where a ground-truth system is not available.

The source code of PeTra is available online under an open-source license [35]. A docker image with all required software to test PeTra and to double-check the overall evaluation posed in this paper is also available online under an open-source license [36].

Author Contributions: Conceptualization, C.Á.-A. and Á.M.G.-H.; methodology, Á.M.G.-H. and V.M.; software, C.Á.-A.; validation, C.Á.-A., Á.M.G.-H. and L.V.C.; formal analysis, F.J.R.-L. and V.M.; investigation, C.Á-A., Á.M.G.-H., L.V.C., F.J.R.-L. and V.M.; resources, V.M.; data curation, C.Á.-A.; writing—original draft preparation, C.Á.-A. and Á.M.G.-H.; writing—review and editing, L.V.C., F.J.R.-L., V.M. and C.F.-L.; visualization, C.Á.-A. and Á.M.G.-H.; supervision, V.M.; project administration, C.F.-L.; funding acquisition, C.F.-L. All authors have read and agreed to the published version of the manuscript.

Funding: The research described in this article has been partially funded by the grant RTI2018-100683-B-I00 funded by MCIN/AEI/ 10.13039/501100011033 and by "ERDF A way of making Europe"; Instituto Nacional de Ciberseguridad de España (INCIBE), under the grant "ADENDA 4: Detección de nuevas amenazas y patrones desconocidos (Red Regional de Ciencia y Tecnología)", addendum to the framework agreement INCIBE–Universidad de León, 2019–2021; and the regional Government of Castilla y León under under the grant BDNS (487971).

Institutional Review Board Statement: Ethical review and approval were waived for this study, due to the fact that subjects involved in the experimentation can not be identified directly or indirectly with the data collected for this work. No personal data was recorded according to GDPR Art. 4.1.

Informed Consent Statement: Not applicable.

Data Availability Statement: PeTra's new model [35], the Rosbag files used in the experiment [32], and the Rosbag files used in the evaluation [34] are available online.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
ERL	European Robotics league
HPC	High-Performance Computing
KPI	Key Performance Indicator
LIDAR	Laser Imaging Detection and Ranging
MCC	Matthews Correlation Coefficient
PeTra	People Tracking
RFID	Radio Frequency Identification
RGB-D	Red Green Blue Depth
ROS	Robot Operating System
RTLS	Real Time Location System
SCAYLE	Supercomputación Castilla y León
SVM	Support Vector Machine

References

1. Arras, K.O.; Lau, B.; Grzonka, S.; Luber, M.; Mozos, O.M.; Meyer, D.; Burgard, W. *Towards Service Robots for Everyday Environments*; STAR 76; Springer: Berlin/Heidelberg, Germany, 2012; Chapter Range-Based People Detection and Tracking for Socially Enabled Service Robots, pp. 235–280.
2. Wang, Y.; Wei, X.; Shen, H.; Ding, L.; Wan, J. Robust fusion for RGB-D tracking using CNN features. *Appl. Soft Comput.* **2020**, *92*, 106302. [CrossRef]
3. Matti, D.; Ekenel, H.K.; Thiran, J.P. Combining LiDAR space clustering and convolutional neural networks for pedestrian detection. In Proceedings of the 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–6.
4. Bu, F.; Le, T.; Du, X.; Vasudevan, R.; Johnson-Roberson, M. Pedestrian planar LiDAR pose (PPLP) network for oriented pedestrian detection based on planar LiDAR and monocular images. *IEEE Robot. Autom. Lett.* **2019**, *5*, 1626–1633. [CrossRef]
5. Lin, T.C.; Tan, D.S.; Tang, H.L.; Chien, S.C.; Chang, F.C.; Chen, Y.Y.; Cheng, W.H.; Hua, K.L. Pedestrian detection from lidar data via cooperative deep and hand-crafted features. In Proceedings of the 2018 25th IEEE International Conference on Image Processing (ICIP), Athens, Greece, 7–10 October 2018; pp. 1922–1926.
6. Rios-Martinez, J.; Spalanzani, A.; Laugier, C. From Proxemics Theory to Socially-Aware Navigation: A Survey. *Int. J. Soc. Robot.* **2015**, 137–153. doi:10.1007/s12369-014-0251-1. [CrossRef]
7. Lee, J.H.; Tsubouchi, T.; Yamamoto, K.; Egawa, S. People tracking using a robot in motion with laser range finder. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 2936–2942.
8. Duong, H.T.; Suh, Y.S. Human Gait Tracking for Normal People and Walker Users Using a 2D LiDAR. *IEEE Sens. J.* **2020**, *20*, 6191–6199. [CrossRef]
9. Guerrero-Higueras, Á.M.; Álvarez-Aparicio, C.; Olivera, M.C.C.; Rodríguez-Lera, F.J.; Fernández-Llamas, C.; Rico, F.M.; Matellán, V. Tracking People in a Mobile Robot From 2D LIDAR Scans Using Full Convolutional Neural Networks for Security in Cluttered Environments. *Front. Neurorobot.* **2018**, *12*, 85. [CrossRef]
10. Álvarez-Aparicio, C.; Guerrero-Higueras, Á.M.; Olivera, M.C.C.; Rodríguez-Lera, F.J.; Martín, F.; Matellán, V. Benchmark dataset for evaluation of range-based people tracker classifiers in mobile robots. *Front. Neurorobot.* **2018**, *11*, 72. [CrossRef] [PubMed]
11. Álvarez-Aparicio, C.; Guerrero-Higueras, Á.M.; Javier, F.; Clavero, J.G.; Rico, F.M.; Matellán, V. People Detection and Tracking Using LIDAR Sensors. *Robotics* **2019**, *8*, 75. [CrossRef]
12. Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics, Cambridge, MA, USA, 26–30 June 1995; pp. 189–196.
13. Choi, K.Y.; Coüasnon, B.; Ricquebourg, Y.; Zanibbi, R. Bootstrapping samples of accidentals in dense piano scores for cnn-based detection. In Proceedings of the 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), Kyoto, Japan, 9–15 November 2017; Volume 2, pp. 19–20.

14. Xue, C.; Lu, S.; Zhan, F. Accurate scene text detection through border semantics awareness and bootstrapping. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 355–372.
15. Zheng, H.; Perrine, S.M.M.; Pitirri, M.K.; Kawasaki, K.; Wang, C.; Richtsmeier, J.T.; Chen, D.Z. Cartilage Segmentation in High-Resolution 3D Micro-CT Images via Uncertainty-Guided Self-training with Very Sparse Annotation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Lima, Peru, 4–8 October 2020; pp. 802–812.
16. Zhang, Y.; Qiao, S.; Ji, S.; Han, N.; Liu, D.; Zhou, J. Identification of DNA–protein binding sites by bootstrap multiple convolutional neural networks on sequence information. *Eng. Appl. Artif. Intell.* **2019**, *79*, 58–66. [CrossRef]
17. Barth, R.; IJsselmuiden, J.; Hemming, J.; Van Henten, E.J. Synthetic bootstrapping of convolutional neural networks for semantic plant part segmentation. *Comput. Electron. Agric.* **2019**, *161*, 291–304. [CrossRef]
18. Huang, C.G.; Huang, H.Z.; Li, Y.F.; Peng, W. A novel deep convolutional neural network-bootstrap integrated method for RUL prediction of rolling bearing. *J. Manuf. Syst.* **2021**, in press. [CrossRef]
19. Maiettini, E.; Pasquale, G.; Rosasco, L.; Natale, L. Speeding-up object detection training for robotics with falkon. In Proceedings of the 2018 IEEE/RSJ international conference on intelligent robots and systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 5770–5776.
20. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
21. Pasquale, G.; Ciliberto, C.; Rosasco, L.; Natale, L. Object identification from few examples by improving the invariance of a deep convolutional neural network. In Proceedings of the 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4904–4911.
22. Vasylkiv, Y.; Ma, Z.; Li, G.; Brock, H.; Nakamura, K.; Pourang, I.; Gomezv, R. Shaping Affective Robot Haru's Reactive Response. In Proceedings of the 2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN), Vancouver, BC, Canada, 8–12 August 2021; pp. 989–996.
23. Robotics Group of Universidad de León. Leon@home Testbed. Available online: https://robotica.unileon.es/index.php?title=Testbed (accessed on 12 February 2021).
24. EU Robotics. ERL Certified Test Beds. Available online: https://www.eu-robotics.net/robotics_league/erl-service/certified-test-beds/index.html (accessed on 12 February 2021).
25. Robotnik. Robotnik Homepage. Available online: https://robotnik.eu/es/ (accessed on 12 February 2021).
26. Quigley, M.; Conley, K.; Gerkey, B.; Faust, J.; Foote, T.; Leibs, J.; Wheeler, R.; Ng, A.Y. ROS: An open-source Robot Operating System. In Proceedings of the ICRA Workshop on Open Source Software, Kobe, Japan, 12–17 May 2009; Volume 3, p. 5.
27. Eliko. KIO RTLS—A UWB-based Indoor Positioning System. Available online: https://www.eliko.ee/products/kio-rtls/ (accessed on 12 February 2021).
28. Guerrero-Higueras, Á.M.; DeCastro-García, N.; Rodríguez-Lera, F.J.; Matellán, V. Empirical analysis of cyber-attacks to an indoor real time localization system for autonomous robots. *Comput. Secur.* **2017**, *70*, 422–435. [CrossRef]
29. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention (MICCAI), Munich, Germany, 5–9 October 2015; Volume 9351, pp. 234–241.
30. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [CrossRef]
31. Álvarez-Aparicio, C. PeTra's New Release: Tracking People by Using a 2D LIDAR Sensor. Available online: https://youtu.be/GCI7lDXQLAM (accessed on 12 February 2021).
32. Dataset RRID:SCR_015743. 2019. Available online: http://robotica.unileon.es/index.php/Benchmark_dataset_for_evaluation_of_range-based_people_tracker_classifiers_in_mobile_robots (accessed on 1 June 2020).
33. Robotics Group of Universidad de Léon. Robotics Group Homepage. Available online: https://robotica.unileon.es/ (accessed on 12 February 2021).
34. Dataset 10.5281/zenodo.4541258. 2021. Available online: https://zenodo.org/record/4541259#.YWAAqNMzY1I (accessed on 7 October 2021).
35. Álvarez Aparicio, C. PeTra (People Tracking). Available online: https://github.com/ClaudiaAlvarezAparicio/petra (accessed on 12 February 2021).
36. Álvarez-Aparicio, C. PeTra Docker Image. Available online: https://hub.docker.com/r/claudiaalvarezaparicio/petra (accessed on 12 February 2021).