International Journal of Computational Intelligence Systems
Vol. **12(2)**; 2019, *pp.* 476–484
DOI: https://doi.org/10.2991/ijcis.d.190318.001; ISSN: 1875-6891; eISSN: 1875-6883
https://www.atlantis-press.com/journals/ijcis/

Using a Deep Learning Model on Images to Obtain a 2D Laser People Detector for a Mobile Robot

Eugenio Aguirre* , Miguel García-Silvente

Department of Computer Science and A.I., CITIC-UGR, E.T.S. Ingenierías en Informática y en Telecomunicaciones, University of Granada, 18071 - Granada, Spain

ARTICLE INFO

Article History
Received 10 Oct 2018
Accepted 12 Feb 2019

Keywords
People detection
2D laser
Machine learning
Deep learning
Mobile robots

ABSTRACT

Recent improvements in deep learning techniques applied to images allow the detection of people with a high success rate. However, other types of sensors, such as laser rangefinders, are still useful due to their wide field of vision and their ability to operate in different environments and lighting conditions. In this work we use an interesting computational intelligence technique such as the deep learning method to detect people in images taken by a mobile robot. The masks of the people in the images are used to automatically label a set of samples formed by 2D laser range data that will allow us to detect the legs of people present in the scene. The samples are geometric characteristics of the clusters built from the laser data. The machine learning algorithms are used to learn a classifier that is capable of detecting people from only 2D laser range data. Our people detector is compared to a state-of-the-art classifier. Our proposal achieves a higher value of F_1 in the test set using an unbalanced dataset. To improve accuracy, the final classifier has been generated from a balanced training set. This final classifier has also been evaluated using a test set in which we have obtained very high accuracy values in each class. The contribution of this work is 2-fold. On the one hand, our proposal performs an automatic labeling of the samples so that the dataset can be collected under real operating conditions. On the other hand, the robot can detect people in a wider field of view than if we only used a camera, and in this way can help build more robust behaviors.

1. INTRODUCTION

In mobile robots, interaction with humans is a very relevant aspect that needs to be improved. Detecting people using robots is a key aspect that makes human–robot interaction (HRI) more natural and flexible. In recent years several proposals have been made to resolve the detection of people and different sensors have been used. It is evident that the ideas for solving the problem of detecting people depend on the robot's sensor system and the features present in the working environment.

In indoor environments, sensors are mainly cameras and laser rangefinders (LRFs). Vision-based approaches to detecting people are very popular and several types of vision devices have been used to solve this task. Mono, stereo, and RGB-depth (Red, Green, Blue plus depth) cameras are some of the vision devices used. Stereo and RGB-depth devices provide color and depth information. For example, Ref. [1] proposes a fuzzy algorithm for detecting and tracking people in the vicinity of a robot using stereoscopic vision. Kinect sensor [2] is used in Ref. [3] to propose people detectors that make use of the data provided by depth sensors and red-green-blue images to deal with the characteristics of HRI scenarios. In addition, skeleton-based approaches using RGB depth have been proposed, see Refs. [4] and [5]. However, vision-only approaches are not the perfect solution for all work situations and environments. Light conditions can affect these methods, depth information is not always reliable and false positives (FPs) in skeleton detection methods are possible.

As for laser-based approaches, some approaches are based on movement features [6], but these methods fail when people do not move, for example, standing or sitting. Such situations can be detected by approaches based on geometric characteristics. In Ref. [7] a set of 14 geometric features is used for leg detection. An AdaBoost learning algorithm is then used for feature selection and to train a classifier. The legs are detected individually and some problems arise when one leg is partially occluded by the other. In Ref. [8] the authors propose schemes for detecting and tracking human legs using fewer features than in Ref. [7]. In Ref. [9], the authors propose a human detection method that uses only a single laser range scanner to detect the waist of the target person. Also, a human-following algorithm is proposed and tested in a two-wheel mobile robot. Ref. [10] proposes an algorithm for people detection, tracking and following from laser data. The authors apply their approach to manage an intelligent power wheelchair. The considered state of the art is the work of Spinello and Siegwart [11] that uses geometric characteristics and an AdaBoost classifier that combines 50 weak classifiers. On these previous works, some authors take the laser data from scenarios with or without people, while other authors label manually

* *Corresponding author. Email:* eaguirre@decsai.ugr.es

the clusters obtained from the laser range data. In our proposal laser data are automatically labelled.

Some conclusions can be drawn from these works. Compared to purely vision-based approaches, the use of a LRF is an advantage, as they are robust against lighting changes and tracking algorithms are faster and more efficient. In addition, the fields of view of LRFs are usually wider than the fields of view of cameras, giving the robot the possibility to detect the presence of people earlier.

With the intention of merging the strengths of cameras and LRFs, some multisensorial solutions have been proposed within the area of mobile robots. The idea is to use data fusion techniques to mix the information supplied by the vision system and the laser device. In Ref. [12], the authors propose a ROS-based multimodal people detection and tracking framework. Their proposal is applied to a mildly humanized robot platform equipped with an array of RGB-D, stereo, and 2D laser range sensors. Ref. [13] combines three types of devices, Kinect, laser, and a thermal sensor to perform the detection of people. In these systems, the authors state that multisensory approaches show that the combination of different sensory cues increases the reliability of their people detection and tracking systems.

In this paper, we focus our attention on the role of LRF in detecting people. These algorithms will be useful both when the person is within and outside the camera's range of vision. Thanks to the 2D LRF, the robot can scan a wide range of the environment and obtain valuable information about the angle and distance of detected objects with good accuracy. In addition, the computational demand for LRF is low due to the relatively low amount of data to be processed. Normally, LRFs are located on the robot close to the ground, so the scan plane allows the robot to detect people's legs. First, a Kinect camera [2] is located near the robot's LRF and both are calibrated and synchronized to capture videos and perform laser scans of people and backgrounds in indoor environments. The videos and scans are stored on a disk while the robot navigates through office-like environments and traverses different types of offices, corridors, and hallways. Second, this dataset will not be manually labelled, but will be labelled with the new method proposed in this paper. This new method uses the power of a computational intelligence technique such as neural networks applied to computer vision to detect and locate the position of people in the images in the dataset. For this purpose, various techniques and models of deep learning about images in our own dataset are studied and evaluated. People's positions in the images are used to automatically label the samples taken by the laser scan. The corresponding coordinate transformation has been carried out to correctly link both sensors. The laser measurements are analyzed and clustered using the jumping distance algorithm. A process of feature extraction of each cluster is carried out taking into account the geometric information. Again computational intelligence is applied. In this phase, the machine learning is used to classify the clusters in people legs or background. Thus, the set of features of each cluster and the corresponding label are used as input to various methods of supervised machine learning in order to identify the machine learning algorithm that is most interesting. Thirdly, our proposal is compared with the work of Spinello and Siegwart [11], considered state-of-the-art classifier in the field of 2D detection from simple frames of laser data. Since the set of samples is very unbalanced, the score of F_1 is the most appropriate measure of comparison between the two classifiers. With the same data, our proposal obtains a value score of F_1 higher than that of Spinello's work. Finally, a new dataset is constructed from the saved data to generate a balanced set of leg and background samples. The new dataset increases our method's ability to detect people's legs. The final classifier is obtained again using machine learning and is evaluated in a test set formed by range data not previously considered by the learning algorithm and obtaining high precision values of around 96%.

The rest of this paper is organized as follows. Section 2 describes the hardware, some software components, and the methods for calibrating both the camera and the laser. From there, the Section 3 analyzes different deep learning techniques for detecting people in images and shows the experimental comparative study conducted on our own dataset. Section 4.1 explains the new approach to leg detection proposed in this paper and the comparison with Spinello's work. In Section 4.2 the final classifier for the detection of people is obtained and evaluated on the test set. Finally, some conclusions and ideas on future works are shown in the Section 5.

2. SYSTEM DESCRIPTION

Our hardware system is composed by a PeopleBot mobile robot [14] equipped with a LRF SICK LMS200 [15] and a Kinect sensor [2] version 1. The LRF has a 180° field of view and it operates at 75 Hz. In the current operation mode the maximum range of distance is 8 m. The systematic error given by the manufacturer is ±15 mm at range 1–8 m. Please do notice that the range of error is low and the measures can be considered accurate enough for the usual tasks of mobile robots. The LRF is mounted at a height of 30 cm above the floor and the Kinect device is located above the LRF. Kinect sensor has both a colour and depth camera [16]. In both cases the resolution of images is 640 × 480 pixels at 30 fps. The standard range of distances of the depth camera is from a minimum of 800 mm to a maximum of 4000 mm although there exists a near mode to allow distances from 400 mm to 3000 mm. In this work the standard mode was chosen. Because it uses infrared technology for the depth camera, Kinect does not work under direct sunlight, for example, outdoors. In Ref. [17] a detailed analysis of Kinect can be found. The robot has an embedded board computer but it is not very powerful thus a laptop is used to run the part of the system which performs the video processing. This laptop is wired linked to the onboard computer and features an Intel Core i5 with 8 GB DDR3 RAM. The laser data are sent by the onboard computer to the laptop to be collected while the Kinect sensor is connected to the USB port of the laptop. Figure 1 shows the robot with the LRF and the vision system.

Regarding the software architecture of our system, this has been implemented using c++ and the resources of the libraries of the robot manufacturer on linux (Ubuntu distribution). The manufacturer supplies this robot with the Aria and ArNetworking libraries. The former is used to execute the program within the robot and the latter to execute the client-server-based program. In order to execute our approach in the laptop, the resources of ArNetworking library have been used. A new server module has been implemented to supply new services to the control program used in this work. The new service allows the client program to obtain the raw data of the sensor laser. Also a new module has been implemented in the client side to receive the laser data using multi-thread programming. So

Figure 1 | Peoplebot robot equipped with laser rangefinder SICK LMS200 and a Kinect camera.

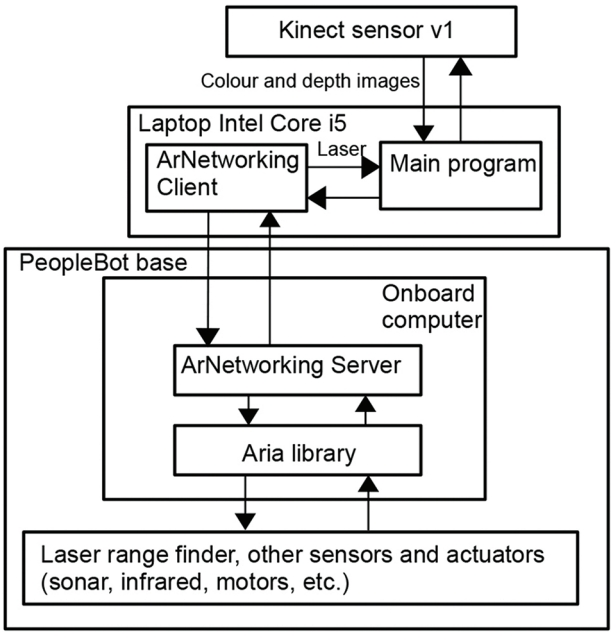

Figure 2 | Architecture of our system.

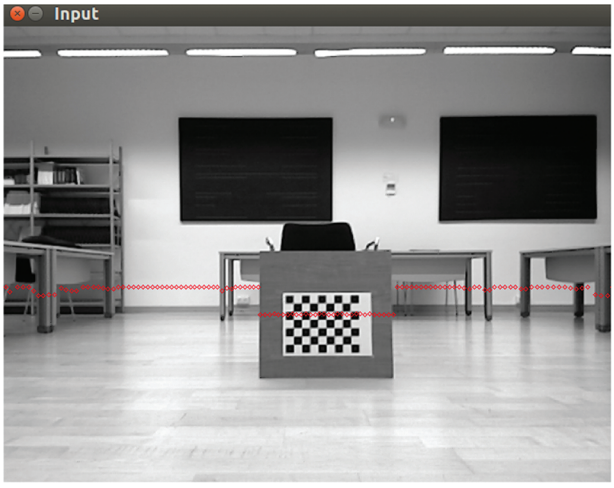

Figure 3 | Results of camera-laser calibration.

the laser data are supplied to the laptop at 10 ms frequency. As we mentioned earlier the Kinect camera is connected to the laptop and it is managed using the OpenCV and OpenNI libraries. OpenNI allows to obtain the BGR color image and the corresponding depth image. Figure 2 shows the architecture of our system.

The camera has been calibrated using the Robust Automatic Detection of Calibration Chessboards approach [18]. This method allows us to obtain the intrinsic parameters of the camera. Using these parameters and an own method by the authors, similar to the Robust Automatic Detection in Laser of Calibration Chessboards (RADLOCC) approach [19], the laser has been extrinsically calibrated to the camera. Thus, a point detected in the plane of the laser can be translated to the 3D coordinates of the camera and projected on the color image. Results of the calibration process can be shown in Figure 3. The red points are the laser measures translated to image coordinates.

3. DEEP LEARNING TECHNIQUES FOR PEOPLE DETECTION ON IMAGES

In order to detect people in images, different deep learning-based techniques have been proposed in the specialized literature. In particular, Faster R-CNN, Region-based Fully Convolutional Networks (R-FCNs), Single Shot Multibox Detector (SSD), and Mask R-CNN have been analysed to be used in this work. In regards to the deep learning frameworks, we have used both TensorFlow and Caffe2, which implement these detectors.

3.1. Faster Region–Based CNN

Before Faster R-CNN networks, external region proposal algorithms were used by the detectors to hypothesise objects locations. However, these algorithms produced a bottleneck that increased the

running time. To solve this, in Ref. [20], it is proposed a fully convolutional network called Region Proposal Network (RPN), end to end trainable. Thus, Faster R-CNN is composed by two modules: a RPN that proposes regions and a Fast R-CNN that classifies them into categories.

RPN networks take an image and generate a set of rectangular regions with an objectness score using a sequence of convolution layers which are shared with the next module. Several models, such as ZFNet or VGGNet, have been studied to obtain the optimal feature map. To generate the proposed region, a sliding window with different sizes (anchors) is slid over this feature map (last convolutional layer output). Finally, each proposed region feeds a fully connected layer, a box regression, and a box classification layer which refine and classify it.

Although nowadays there are models that have improved Faster R-CNN speed, few models improve its accuracy being Faster R-CNN with Resnet one of the most accurate object detection models.

3.2. Region-Based Fully Convolutional Networks

R-FCNs [21] is a simple but accurate and efficient model for object detection. The main goal is to share as much as possible the computational cost. To achieve this, it is necessary to use fully convolutional layers. R-FCN uses the same object detection strategy that Faster R-CNN (Faster Region-Based Convolutional Neural Networks), meaning that it is also based on two stages: A subnetwork propose regions and the next one classifies them.

Firstly, a fully convolutional network (typically convolutional layers of a ResNet) is used as backbone to obtain a features map. Secondly, a set of proposal regions (Region Of Interest, RoI) is extracted with a RPN. This subnetwork is also fully connected and share weights with the next module. After that, each RoI is passed to a range of convolutional layers and finally to a layer made up from a bank of maps called "position-sensitive scores maps." This bank consist of $k^2\,(C+1)$ maps where k is the spatial grid size which describes relative positions and C is the number of classes (+1 background). The model ends with a position-sensitive RoI pooling layer. This layer aggregates the outputs of the previous layer and generates the score for each RoI.

This model is able to reduce the computational time and reaches similar accuracy thanks to the convolutional layers and the position-sensitive score maps.

3.3. Single Shot Multibox Detector

SSD [22] uses only one deep neural network for object detection. It is based on a feed-forward convolutional network which output is a set of default bounding boxes associated with each feature map and scores for object detection. At prediction time, bounding boxes are refined to adjust them with the shape of the object. Moreover, this model is easy to train because the region proposal is removed.

The first part of the network, called base network, consists of a standard architecture (such as VGG16) used for image classification. Then, it is added a set of extra convolutional layers (features map) whose size decreases progressively and allows to predict detections at multiple scales. After that, default bounding boxes are associated with each feature map to compute offsets relative to the default box and scores regarding the presence of objects. A lot of bounding boxes are generated and compared to other and it may be very likely that they do not contain any object. To improve results, SDD uses nonmaximum suppression to eliminate overlap boxes and hard negative mining to balance classes during training.

SSD is not very different to other models since it only skips the region proposal step. The prediction of the bounding box and the classification is done in "one shot." Because of that, SSD is one of the fastest detectors.

3.4. Mask R-CNN

Mask R-CNN [23] is a simple and flexible model which is able to efficiently detect objects and generate a segmentation mask in one image. It is an R-CNN extension that use a new branch to predict objects masks in parallel with the Faster R-CNN for object detection.

Mask R-CNN performance is described as follows: Faster R-CNN generates a class label and a bounding box offset for each candidate object, then a new branch outputs the object mask. However, segmentation task requires a much finer extraction of the spatial layout. This is possible thanks to several techniques such as pixel to pixel alignment.

In the same way as Faster R-CNN, segmentation branch generates a binary mask for each RoI proposed by the RPN. Specifically, a $m \times m$ mask is predicted for each RoI using a fully connected network. Moreover, pixel precision requires a good alignment to preserve spatial correspondence. To do this, a RoIAlign layer is used in this model.

3.5. Experimental Comparative Study

In order to choose a deep learning technique for our system, different models and architectures have been tested. To evaluate each case, we have used an own dataset that consist of 624 images (312 positives and 312 negatives) that have both people and backgrounds. These images have been collected navigating with the robot at our office-like environment. These samples have been manually labelled indicating the region of the image where people are located. Furthermore, a sample is labelled as positive if some person is shown in the image and negative if no one is shown.

Table 1 shows the list of models that have been checked using our own test set. The "Network name" consists of the name of the model, the standard or base architecture used to generate features maps, and the standard dataset that was used to train each model. A code has been included to link this table with Table 2 which shows the results of each model on our own dataset.

In order to evaluate the different deep learning techniques, we have used the typical measures of binary classification. Those are, number of true positive (TP), number of FP, number of true negative (TN), number of false negative (FN). Also the computation time by image is taken into account. In this work, these measures are defined in the following manner:

Table 1 List of models, including the identification code, deep learning architecture, and standard dataset.

Code	Network Name: Model + Architecture + Dataset
	TensorFlow
1	ssd_mobilenet_coco
2	ssd_mobilenet_coco
3	ssd_mobilenet_coco
4	ssd_inception_v2_coco
5	ssd_inception_coco
6	ssd_inception_v2_coco
7	rfcn_resnet101_coco
8	rfcn_resnet101_coco
9	rfcn_resnet101_coco
10	faster_rcnn_resnet101_coco
11	faster_rcnn_resnet101_coco
12	faster_rcnn_resnet101_coco
13	faster_rcnn_inception_resnet_v2_coco
14	faster_rcnn_inception_resnet_v2_coco
15	faster_rcnn_inception_resnet_v2_coco
	Caffe2
16	mask_rcnn_resnet101_coco (detectron)
17	mask_rcnn_resnet101_coco (detectron)
18	mask_rcnn_resnet101_coco (detectron)

Table 2 Results of detectors listed in Table 1.

Code	P	Acc	Sens	Spec	Time
			TensorFlow		
1	0.3	0.955	0.955	0.955	0.081
2	0.5	0.91	0.833	0.987	0.127
3	0.7	0.721	0.442	1	0.07
4	0.3	0.957	0.974	0.93	0.088
5	0.5	0.926	0.865	0.987	0.09
6	0.7	0.697	0.394	1	0.09
7	0.3	0.918	0.981	0.856	0.142
8	0.5	0.943	0.981	0.907	0.141
9	0.7	0.958	0.977	0.939	0.143
10	0.3	0.99	0.785	0.785	0.182
11	0.5	0.926	0.987	0.865	0.195
12	0.7	0.947	0.987	0.907	0.175
13	0.3	0.945	0.987	0.904	0.452
14	0.5	0.962	0.987	0.936	0.44
15	0.7	0.976	0.987	0.965	0.488
			Caffe2		
16	0.3	0.95	0.974	0.926	0.117
17	0.5	0.96	0.978	0.962	0.118
18	**0.7**	**0.982**	**0.99**	**0.974**	**0.118**

Acc, Accuracy; Sens, Sensitivity; Spec, Specificity. The model chosen for our proposal is listed in bold type.

- **TP**. For positive images, if the greater detection exceeds a certain overlap threshold *IoU* and a certain probability threshold *P*.
- **FN**. For positive images, if the greater detection does not exceed both thresholds *IoU* and *P*.
- **TN**. For negative images, if no one is detected with a probability greater than the threshold *P*.
- **FP**. For negative images, if a person is detected with a probability greater than the threshold *P*.

Using these measures, the Accuracy (Acc), Sensitivity (Sens), or TP rate, the Specificity (Spec) or TN rate are defined by Eq. (1), Eq. (2), and Eq. (3) respectively.

$$Acc = \frac{TP + TN}{TP + FP + FN + TN} \tag{1}$$

$$Sens = \frac{TP}{TP + FN} \tag{2}$$

$$Spec = \frac{TN}{TN + FP} \tag{3}$$

The overlap threshold *IoU* has been established in 0.65 and the results have been obtained with different probability thresholds *P* as it is shown in Table 2.

Depending on the task, there should be a compromise between time and precision. In general, the best result has been obtained using a Mask-RCNN with ResNet101 and trained with COCO. Moreover, we have to take into account that this model not only detects objects but also generates their masks. The masks of the detected people are useful for us since they will allow to properly label the laser data, therefore the model number 18 is chosen for our proposal. Details on the network architecture of this model can be seen in Ref. [24].

4. PEOPLE DETECTION IN THE LASER SCANS

4.1. Leg Detection

Previous approaches for people detection based on 2D laser measures, process the laser data in a similar way using the algorithm of jump distance to generate clusters which will be classified depending on certain geometrical features. The problem of such approaches is how to properly label the laser data to identify the scans corresponding to people legs or to some object of the environment or that are part of the background. The samples are usually manually labelled with the help of videos recorded while the laser was gathering the data. This manual method could generate samples with the wrong labels that could affect the supervised machine learning. And therefore, we could obtain biased classifiers and with a poor behaviour in the real world. In our proposal, we use the masks of people in the images to automatically label the clusters obtained from the laser data. These masks are generated by the Mask-RCNN model which has obtained good results in the study of subsection 3.5. This automatic process to label the samples allows to collect the data in a more natural way. Thus the robot can navigate by the environment recording laser data and images while the people are moving or stay static. Thereafter both laser data and images are processed to generate the dataset needed to train the machine learning algorithms or to test the learned classifiers. Below our proposal for people detection is explained in detail.

First, as it was mentioned above, color and depth images from Kinect and 2D laser range data are recorded while people are freely walking at the proximity of the robot and when it is navigating by the environment. Second, the laser data are translated to camera coordinates using the extrinsic calibration data. Thereafter, laser data are clustered using a jump distance algorithm, a minimum number of points and a maximum length for the cluster. Third, the Mask-RCNN is used on this frame to obtain the masks of the people present in the frame (see Figure 4). Fourth, every cluster is analysed to test whether their points, translated to image coordinates, are on the legs of some people.

In order to do this, please note that depth information has to be used, that is, both laser and pixel points have to be translated to 3D coordinates of the camera to compute the real distance between

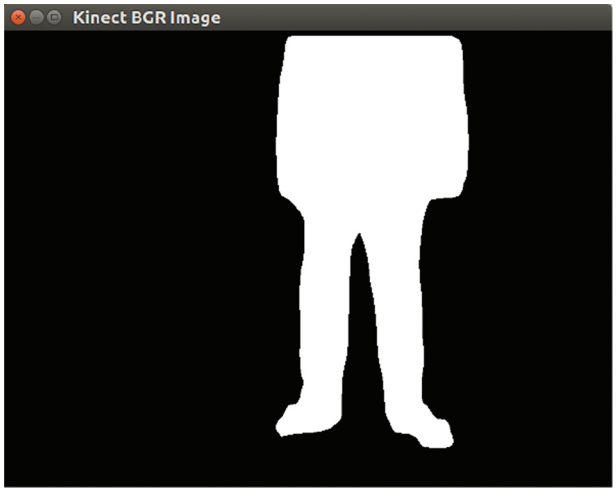

Figure 4 | Mask of people obtained from Mask-RCNN.

them. If the points of a cluster are on a leg of the mask, then the cluster is labelled as leg. If the people are walking fast it is possible that the situation of laser points in the legs were not perfect, so that a threshold of distance is taken into account. If the cluster is not on any leg in the image, then the cluster is labelled as no-leg. Figure 5 summarizes the process for automatic labelling of the 2D laser measures.

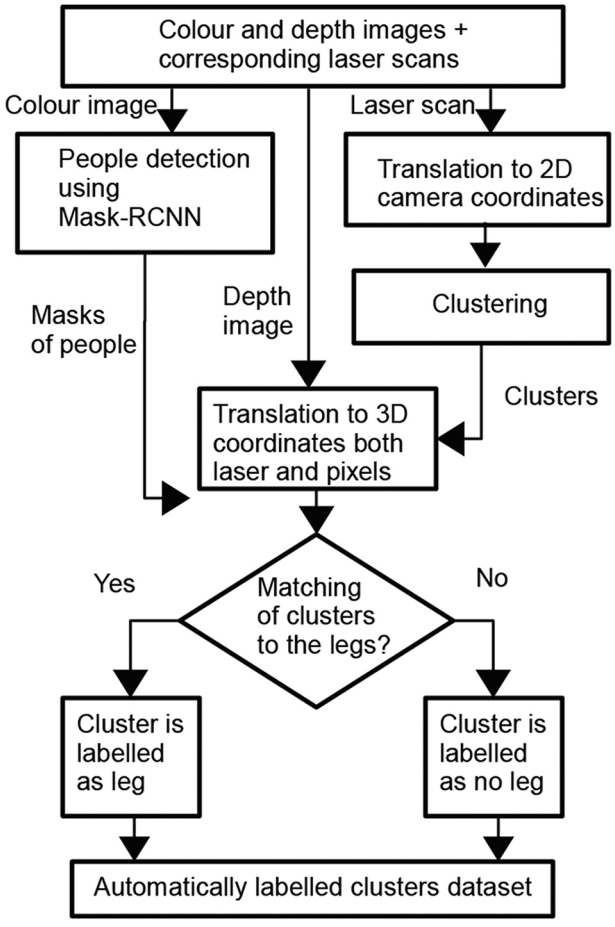

Figure 5 | Process for automatic labelling of the 2D laser data.

Figure 6 shows the result of the automatic labelling process. The blue and red points are laser data clustered in different clusters. The green points are laser points which have not been assigned to any cluster. Finally, the yelow color is used for laser points that are considered near the legs thus those clusters will be labelled as legs.

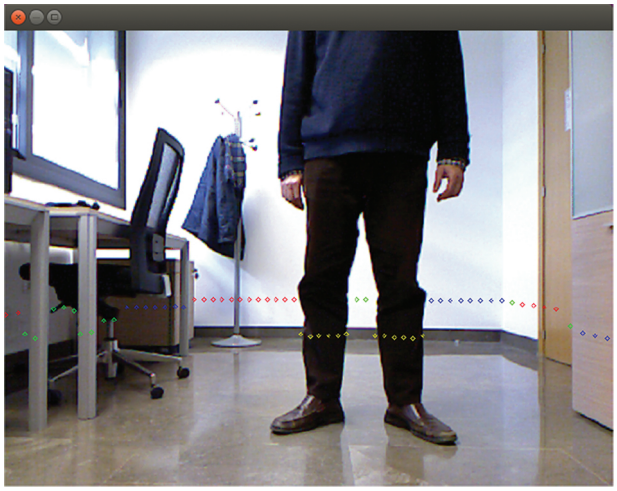

Figure 6 | Automatic labelling of the clusters using the mask.

The clusters will be classified by supervised machine learning algorithms. Previously it is needed to compute some geometrical properties to represent such clusters. Different possibilities exist in the specialized literature to compute the geometrical properties. In our proposal we use as geometrical features: the *contour* of the neighbour points in a cluster from P_1 to P_n, the *width* defined as the distance from P_1 to P_n, and the *depth* as the maximum distance between a point P_i and the line $\overline{P_1P_n}$.

These attributes have also been used in Ref. [8] with good results. An important difference with our approach it is that in Ref. [8] the samples of legs are taken in a controlled environment placing one leg in front of a vertical board in various leg configurations in order to be scanned by the LRF. These geometrical properties are shown by Figure 7.

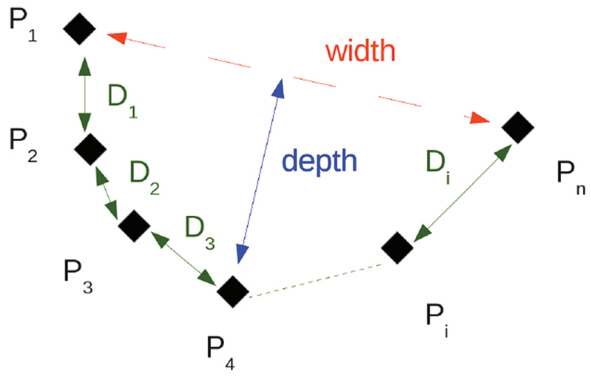

Figure 7 | Geometrical properties used in the clusters feature extraction.

Once the clusters are represented by the three geometrical properties, this information along with the label of the class, leg or no leg, is supplied to the machine learning algorithms. This is a problem of binary classification with unbalanced classes because the number of samples of no legs is greater than the samples of legs. This is because in every scan, LRF scans with an angle of 180° and the number of people in every frame, at the same time, is one or two so most of the clusters for every frame correspond to the background or other objects of the environment. The labelled dataset of geometrical properties is divided into training and test sets. A sample is considered positive if its label is leg and negative on the contrary. The training set contains 2100 positive and 19337 negative samples. Test set contains 696 positive and 7554 negative samples. To train several machine learning algorithms, the machine learning platform Weka [25] was used. The experiments are done 10 times and a 10-fold cross validation is carried out. The algorithms checked are PART which is a rule-based algorithm, J48 which uses a decision tree C4.5, a Multilayer perceptron of neural networks, and Random Forest algorithm which is based on a forest of random trees. The results of average accuracy for each algorithm in the training set are shown by Table 3, taking into account the accuracy as the percentage of correctly classified instances (both positives and negatives).

Table 3 Accuracy average of several machine learning algorithms using a 10-fold cross validation.

PART	J48	Multilayer Perceptron	Random Forest
95.40	96.34	90.20	96.75

The algorithm with the best result in this experiment was the Random Forest so that this algorithm is chosen to build a classifier which can be assessed in the test set. Furthermore another interesting measures have been computed to check the behaviour of this algorithm in the training and test sets. In this sense, due that the training and test sets are unbalanced, additional measures as *Precision*, *Recall*, and F_1 score must be analysed. Let *TP*, *TN*, *FP*, *FN* the number of TP, TN, FP, and FN classified instances respectively then *Precision*, *Recall*, and F_1 score are defined by Eq. (4), Eq. (5), and Eq. (6), respectively.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{6}$$

Results of the Random Forest algorithm on the training and test sets are shown by Table 4.

Table 4 Results of the random forest algorithm on the training and test sets.

Set	Accuracy	Precision	Recall	F_1
Training	0.968	0.967	0.968	0.967
Test	0.933	0.947	0.933	0.939

The results of Random Forest are good enough both on the training and test sets therefore this algorithm is chosen to classify the clusters in our proposal. Figure 8 summarizes the process to obtain the classifier from the automatically labelled clusters.

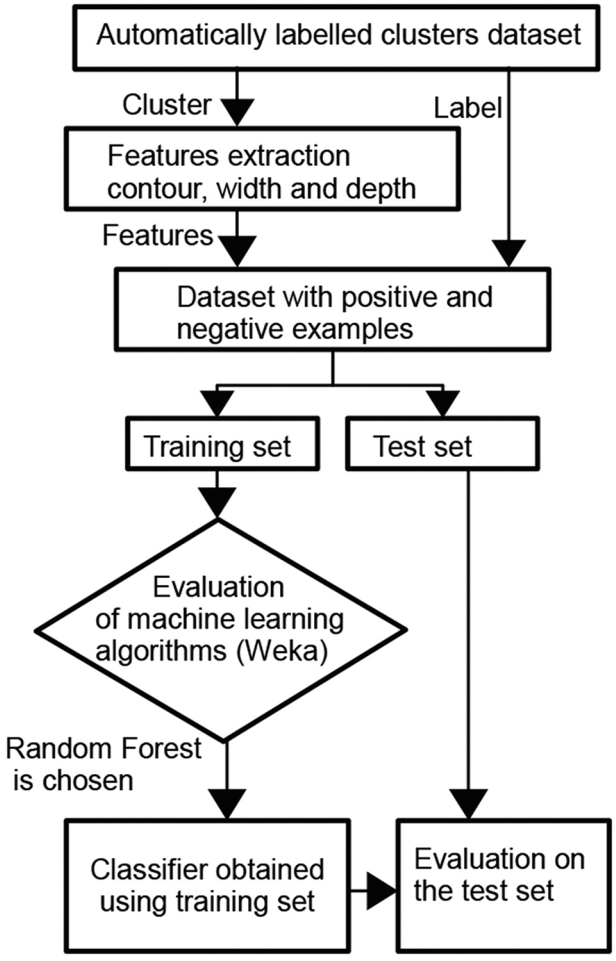

Figure 8 Process to obtain the classifier from the automatically labelled clusters.

To compare our approach with the state-of-the-art classifier, the open source implementation of Spinello [11] has been download from http://www2.informatik.uni-freiburg.de/~spinello/people2D.html. The approach of Spinello takes as input every scan of the LRF as coordinates x, y of the sensed point and a label indicating if the point belongs to people or to the background. Thereafter the points are segmented in clusters and several features are computed to classify the clusters by an AdaBoost classifier which combines 50 weak classifiers. In this comparison, the dataset is the same used to train the Random Forest algorithm establishing the same training and test sets for both approaches. To measure the quality of the evaluation for both classifiers we are going to use the same evaluation measure of Ref. [11]. This measure is computed by using the precision-recall curve, generated by the variation of the AdaBoost classification threshold θ. The curve can be summarized in a single quality measure for the classifier C using the maximum F_1 score over the detector's AdaBoost threshold θ as it is computed by Eq. (7).

$$\max_{\theta} F_1 = \max_{\theta} 2 \cdot \frac{Precision\,C(\theta) \cdot Recall\,C(\theta)}{Precision\,C(\theta) + Recall\,C(\theta)} \quad (7)$$

To test the approach of Spinello in our test set, first we train the detector of Spinello using the dataset which is available for training in the web page of this classifier. Thus a trained classifier is obtained and it is evaluated on our test set giving the results shown in Figure 9. In this first case, the maximum value for $\max_\theta F_1$ is 0.776 It is because Precision and Recall are not high at the same time. That is, when Precision is high then Recall achieves only on 0.64 since there are a high value of FNs. This result is reasonable since the detector has been trained with the original data.

After that, the detector is trained on our training set and the obtained classifier is evaluated on our test set giving the results shown by Figure 10.

In this second case, the maximum value for $\max_\theta F_1$ is 0.908 This result is good due that the detector has been trained on our dataset which contains data taken from our kind of environment and therefore it is better fitted for the test set. Comparing against our approach, as it is shown on Table 4, our proposal gives for $F_1 = 0.939$, which overcomes the results of the state-of-art classifier in our test set.

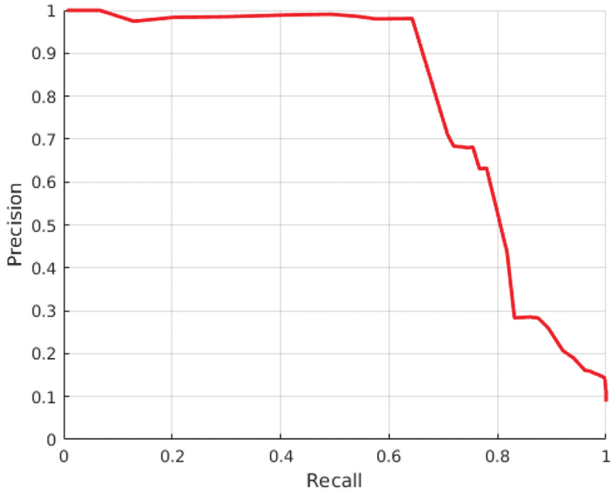

Figure 9 Precision-recall curve for the Spinello's classifier trained on the original dataset and applied to our test set.

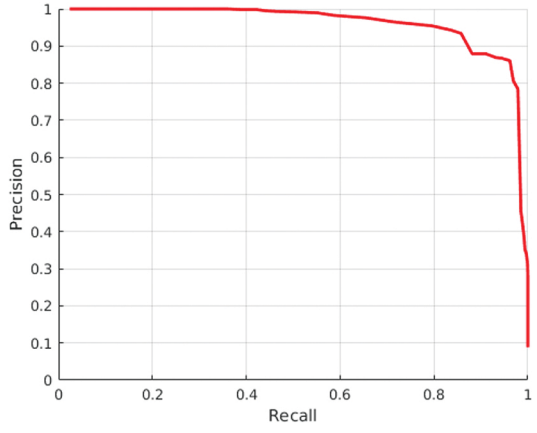

Figure 10 Precision-recall curve for the Spinello's classifier trained on our dataset and applied to our test set.

4.2. Results Using Balanced Training and Test Sets

Once our proposal has been explained and compared to the Spinello's classifier, the idea is to improve the *Accuracy* of our detector. To achieve this, a new dataset is taken from the laser and images recorded by the robot. In this case, the clusters are chosen by removing clusters with very similar features in order to obtain a balanced dataset. Thus, in order to generate a final classifier, a balanced dataset containing both positive (2068) and negative (2068) samples is used for training. Also a balanced test set is generated with 701 positive and 701 negative samples, not previously used by the machine learning algorithm. The results on the test set are shown by Table 5.

Table 5 Results of random forest algorithm using a balanced test set.

Set	Accuracy	Precision	Recall	F_1
Test	0.96	0.96	0.96	0.96

In order to check the values of accuracy in each class, Table 6 shows the contingency table for the test set where it can be seen that both classes are properly classified.

Table 6 Contingency table of random forest algorithm on the test set.

		Observation Class	
		Positive (%)	Negative (%)
Predicted class	Positive	96.7	3.3
	Negative	4.7	95.3

5. CONCLUSIONS

In this paper a LRF-based system using a mobile robot in order to detect people has been proposed. The approach tries to use the strengths of LRF sensors and the advances in people detection obtained by the deep learning models. A comparative study of different models and architectures of deep learning techniques has been carried out in order to obtain a model capable to detect the masks of people in the images. By using the masks of people, and matching them with the laser data, we can automatically label numerous samples of clusters relative to people legs. The clusters are described by certain geometric features. Thereafter several classifiers are trained using Weka in order to choose the best classifier. The results of the developed classifier are good not only in the training set but also in the test set. The approach has been compared to the considered state-of-the-art classifier using the measure F_1 score, due that the classes are unbalanced, obtaining our approach a higher value of F_1 score. Finally, the accuracy of our classifier can be improved using a balanced dataset achieving high rates of classification for every class.

The contribution of this work is double, on one hand, we show how a tool based on computational intelligence, such as people detectors based on deep learning, can be used to automatically label a set of samples formed by 2D laser range data. In this work, this idea has been applied to people detection but other interesting objects of the environment, or behaviours of pedestrians, could be characterized using our approach. On the other hand, by using our detector, mobile robots can detect people in a wider field of view than only by using the camera, since the learn classifier is only based on 2D laser range data. This can be interesting to avoid collisions with people that is out of the field of view of the camera, or to be aware of human presence, or to complement the visual information in people following task or other kinds of HRIs obtaining more robust behaviours.

As future work, this approach can be used to analyse the way the people walk to detect possible problems in the elderly people, which can be very useful for a service robot.

ACKNOWLEDGMENTS

This work has been supported by the Spanish Government TIN2016-76515-R Grant, supported with Feder funds.

REFERENCES

[1] R. Paúl, E. Aguirre, M. García-Silvente, R. Muñoz-Salinas, A new fuzzy based algorithm for solving stereo vagueness in detecting and tracking people, Int. J. Approx. Reason. 53 (2012), 693–708.
[2] Microsoft, Kinect for X-BOX 360, 2010, [Online], Available: http://www.xbox.com/en-US/kinect.
[3] A. Ramey, Á. Castro-González, M. Malfaz, F. Alonso-Martin, M.A. Salichs, Vision-based people detection using depth information for social robots: an experimental evaluation, Int. J. Adv. Robot. Syst. 14(3) (2017), 1–15.
[4] G.Th. Papadopoulos, A. Axenopoulos, P. Daras, Real-time skeleton-tracking-based human action recognition using kinect data, in: C. Gurrin, F. Hopfgartner, W. Hürst, H.D. Johansen, H. Lee, N.E. O'Connor (Eds.), MMM (1), Lecture Notes in Computer Science, vol. 8325, Springer, Cham, Switzerland, 2014, pp. 473–483.
[5] S. Prabhu, J.K. Bhuchhada, A. Dabhi, P. Shetty, Real time skeleton tracking based human recognition system using kinect and arduino, inIJCA Proceedings on National Conference on Role of Engineers in Nation Building, NCRENB, Mumbai, Maharashtra, India, 2015, vol. 2, pp. 1–6.
[6] D. Schulz, W. Burgard, D. Fox, A.B. Cremers, People tracking with mobile robots using sample-based joint probabilistic data association filters, Int. J. Robot. Res. 22(2) (2003), 99–116.
[7] K.O. Arras, O.M. Mozos, W. Burgard, Using boosted features for the detection of people in 2D range data, in Proceedings 2007 IEEE International Conference on Robotics and Automation, Roma, Italy, 2007, pp. 3402–3407.
[8] W. Chung, H. Kim, Y. Yoo, C.-B. Moon, J. Park, The detection and following of human legs through inductive approaches for a mobile robot with a single laser range finder, IEEE Trans. Ind. Electron. 59(8) (2012), 3156–3166.
[9] J. Cai, T. Matsumaru, Human detecting and following mobile robot using a laser range sensor, J. Robot. Mechatronics. 26 (2014), 718–734.
[10] A. Leigh, J. Pineau, N. Olmedo, H. Zhang, Person tracking and following with 2D laser scanners, in Proceedings of 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, 2015, pp. 726–733.
[11] L. Spinello, R. Siegwart, Human detection using multimodal and multidimensional features, in IEEE International Conference on Robotics and Automation, Pasadena, CA, 2008, pp. 3264–3269.
[12] T. Linder, K.O. Arras, People detection, tracking and visualization using ROS on a mobile service robot, in: A. Koubaa (Ed.), Robot Operating System (ROS), Studies in Computational Intelligence, vol. 625, Springer, Cham, 2016, pp. 187–213.
[13] L. Susperregi, J.M. Martínez-Otzeta, A. Ansuategui, A. Ibarguren, B. Sierra, RGB-D, laser and thermal sensor fusion for people following in a mobile robot, Int. J. Adv. Robot. Syst. 10 (2013), 271.
[14] Adept Mobilerobots, Performance Peoplebot Robot Operations Manual v.8.2., Amherst, NH, USA, 2011.
[15] Sick - Industrial Sensors, LMS200-30106 - technical data, 2008, [Online], Available: http://www.sick.com.
[16] B. Freedman, A. Shpunt, M. Machline, Y. Arieli, Depth mapping using projected patterns, Patent Application, 2008. WO 2008/120217 A2.
[17] Z. Zhang, Microsoft kinect sensor and its effect, MultiMedia IEEE. 19(2) (2012), 4–10.
[18] J.Y. Bouguet, A Release of a Camera Calibration Toolbox for Matlab, 2008, [Online], Available: http://www.vision.caltech.edu/bouguetj/calib_doc.
[19] A. Kassir, T. Peynot, Reliable automatic camera-laser calibration, in Australasian Conference on Robotics and Automation, Brisbane, Queensland, Australia, 2010.
[20] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks,in Advances in Neural Information Processing Systems 28, (NIPS 2015), Montreal, Canada, 2015, pp. 91–99.
[21] J. Dai, Y. Li, K. He, J. Sun, R-FCN: object detection via region-based fully convolutional networks, in Advances in Neural Information Processing Systems 29, (NIPS 2016), Barcelona, Spain, 2016, pp. 379–387.
[22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in European Conference on Computer Vision, Springer, Amsterdam, The Netherlands, 2016, pp. 21–37.
[23] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in Computer Vision (ICCV), 2017 IEEE International Conference, IEEE, Venice, Italy, 2017, pp. 2980–2988.
[24] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, K. He, Detectron, 2018, https://github.com/facebookresearch/detectron
[25] E. Frank, M.A. Hall, I.H. Witten, The weka workbench, in: M. Kaufmann (Ed.), Data Mining: Practical Machine Learning Tools and Techniques, fourth ed., Elsevier, Morgan Kaufmann Publishers, Cambridge, MA, 2016.