People detection on 2D laser range finder data using deep learning and machine learning[⋆]

José Abrego-González, Eugenio Aguirre, and Miguel García-Silvente

Department of Computer Science and A.I. (DECSAI).
Andalusian Research Institute in Data Science and Computational Intelligence
(DaSCI). CITIC-UGR., University of Granada (UGR), 18071 Granada, Spain
jabrego@correo.ugr.es, eaguirre@decsai.ugr.es,
m.garcia-silvente@decsai.ugr.es

Abstract. This work presents a machine learning based study on people detection using 2D Laser Range Finders (LRFs) combined with deep learning methodologies, aimed at enhancing mobile robot capabilities in various environmental conditions. The study introduces a novel integration of a monocular camera with an LRF on a mobile robot to improve the accuracy and efficiency of detecting and tracking people. By employing deep learning models such as CenterNet, the system leverages both image and 2D range data to facilitate automatic labeling of datasets, crucial for training robust classification algorithms. In order to achieve the best classifier, two experimental studies are introduced in this work. The former is carried out in a simulated environment and the latter in real-world, office-like environments. In simulations, various machine learning models are trained and evaluated, showing significant results in distinguishing human legs from other objects. The transition to real-world testing underscores the challenges and adaptations necessary to achieve high accuracy and reliability in dynamic settings. The XGBoost model emerged as the most effective classifier in our study, achieving the highest scores in accuracy, precision, recall, and F1-score, outperforming other methods across these key metrics. This work aims to advance the field of 2D LRF based people detection and also proposes a solution for real-time applications, balancing precision and computational efficiency. Experimental results from both simulated and real-world environments demonstrate the system's effectiveness.

Keywords: People detection · Deep learning · Machine learning · 2D LRF

1 Introduction

Mobile robots rely on detecting and tracking people for applications like Human-Robot Interaction (HRI), navigating crowded spaces, and safety in shared environments [20]. Various computer vision techniques using Monocular, Stereo, and

[⋆] This work was made possible thanks to the support of Senacyt Panamá (Scholarship No. 270-2022-164) and Grant PID2022-138453OB-I00 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe".

RGB-Depth cameras, including deep learning methods like YOLO, have proven effective for these tasks [15]. Despite the advantages of vision systems, 2D Laser Range Finders (LRFs) are favored in social and service robots for their reliability and wide field of view, overcoming the limitations of vision sensors in adverse conditions [4].

Detecting people using 2D laser technologies can be achieved through various approaches, as outlined in the survey conducted by M. Sharif [22]. Some techniques directly process laser measurements individually as inputs for supervised machine learning algorithms. Conversely, alternative methods first cluster these measurements and then derive features to characterize these clusters. In this work, we will adopt the latter approach and propose a set of innovative features compared to those described in the specialized literature.

Emerging studies utilize deep learning for enhanced detection from sensor data, offering significant improvements in reliability [21]. Given the absence of automatic labeling tools for 2D laser data, this work explores the potential of deep learning to automate the labeling of such datasets and the use of machine learning approaches to generate efficient leg detectors, aiming to enhance efficiency and accuracy in diverse applications [14].

This work is organized as follows: Section 2 outlines the proposed system's hardware and software, emphasizing camera and LRF integration. In Section 3, deep learning methods for object detection are detailed, with a focus on the CenterNet model. Section 4 describes the mobile robot's hardware and software components. Section 5 discusses the simulated environment for leg detection using 2D LRF and the associated machine learning training. Experimental evaluations in simulated and real-world conditions are covered in Sections 5 and 6, respectively. The conclusions and some ideas on future work are commented in Section 7.

2 Description of the proposal

In the initial phase of this study, we employ a simulated environment using CoppeliaSim to rigorously test our experimental setup. This approach ensures that it performs effectively under controlled conditions designed to mirror real-world scenarios. This simulation allows us to evaluate the detection capabilities and overall system reliability without the complexity and variability inherent in physical environments. Through this simulation, we gain insights into the system's performance, providing a foundation for further development and refinement before real-world application.

The next step involves using real-world data to validate the findings from our simulations. Building on prior work in people detection, tracking [2], and automatic labeling of 2D range data [1], this proposal introduces a refined method by employing a monocular camera instead of the Kinect 1.0 sensor, avoiding the Kinect's infrared drawbacks under certain lighting conditions. The system incorporates a LRF sensor mounted 30 cm above the floor on a mobile robot, paired with a Jetson TX2 Developer Kit, which manages the camera and LRF integration. This setup captures both image and 2D range data alongside odometry and velocity data, stored on an onboard SSD during navigation through an office-like environment.

This data supports offline deep learning analysis on powerful machines using TensorFlow 2 Object Detection API [12]. A specific focus is on the CenterNet HourGlass104 Keypoints 512x512 model [27], trained on the COCO 2017 dataset [17], to detect people in images. This model helps in keypoint detection, aiding in the identification of human legs through images, which are then correlated with 2D range data for automatic labeling. The labeled data facilitates the development of a binary classifier for detecting people's legs via machine learning algorithms.

3 Deep learning based object detection

3.1 Two and one stage approaches

Object detection methods, essential for identifying categories like people and animals in images, have advanced significantly with deep learning, particularly through Convolutional Neural Networks (CNN) [10]. These methods fall into two primary categories: two-stage and one-stage approaches.

Two-stage approaches, such as R-CNN [10] and its successors, Fast-RCNN [9] and Faster-RCNN [19], involve first extracting Regions of Interest (RoIs) and then classifying them. Innovations include Mask-RCNN [11] for simultaneous object and mask detection, and R-FCN [8], which employs position-sensitive score maps. Cascade R-CNN [5] addresses overfitting by training sequential detectors with increasing Intersection over Union (IoU) thresholds.

One-stage approaches, exemplified by YOLO [25] and SSD [18], streamline the process by directly classifying and regressing anchor boxes, eliminating the need for separate RoI extraction. Keypoint-based methods like CenterNet [28] represent objects using keypoints, which simplifies bounding box determination and avoids traditional anchor box disadvantages.

3.2 CenterNet

CenterNet utilizes a single central point in an object's bounding box for representation, regressing other properties like size and pose from image features [27]. A keypoint heatmap generated by a fully convolutional network aids in detecting object centers, and bounding boxes are predicted from these peaks. The model uses dense supervised learning for training and operates in real-time without non-maximal suppression during inference.

CenterNet not only provides excellent speed-accuracy trade-offs on the COCO dataset but also allows for multi-person pose estimation, identifying human joints as offsets from the center [6]. Various applications have demonstrated CenterNet's efficacy, such as fault diagnosis in train catenaries [7], biometric recognition [26], vehicle detection [23], and real-time person detection in surveillance [3].

For this project, a CenterNet HourGlass104 Keypoints 512x512 model pre-trained on the COCO 2017 dataset [17] was selected based on its optimal balance between speed and accuracy. This model is integral to detecting people and their keypoints within our collected dataset, demonstrating its versatility and robustness in object detection.

4 System overview

The system consists of three primary hardware components. The first is a PeopleBot mobile robot equipped with an LRF SICK LMS200 [13], which has a 180º field of view and is capable of accurate measurements up to 8 meters. The second component is an NVIDIA Jetson TX2 Developer Kit, mounted on top of the LRF. This integrated unit features both GPU and CPU, optimized for high efficiency and power, supporting an onboard camera with a resolution of 640 x 480 at 30 frames per second. The Jetson TX2 operates on Ubuntu 18.04 and connects to the robot's sensory system via USB. The third component is a laptop, positioned on the robot and connected to the Jetson TX2 via Ethernet, serving as the user interface.

The software architecture of the system is developed in C++, utilizing Aria and ArNetworking libraries provided by the robot manufacturer for programming and network communication, although only Aria is needed due to the direct wired connection of the Jetson TX2 which replaces the robot's original onboard computer. The Jetson TX2 is powered by a LiPo battery, similar to those used in drones, while the robot operates on a standard plumb battery system. OpenCV library manages the image processing, and both laser measurements and images are saved directly to an SSD connected to the Jetson TX2.

5 Leg detection using 2D LRF in simulated environments

To evaluate the effectiveness of various techniques for detecting people using 2D LRF data, a simulated environment was utilized. CoppeliaSim was chosen for its straightforward integration with several programming languages, including Python, which facilitated the simulation process.

Python was the primary programming language employed in this study due to its widespread adoption in research and data science. Its clear syntax and a comprehensive array of specialized libraries support efficient, complex data analysis and model development. Key libraries used include NumPy for numerical computations, Pandas for data manipulation, scikit-learn for machine learning, and OpenCV for computer vision. This combination of accessibility and robust functionality makes Python especially suitable for conducting data-driven research efficiently.

5.1 Data Collection Process

The data collection process for leg detection using a 2D LRF was meticulously designed within the CoppeliaSim simulation environment. Three unique scenarios were crafted to closely replicate real-world dynamics and interactions. These scenarios include a single person moving along a predefined path, a static group of people, and an assortment of non-human objects that mimic the shape of human legs.

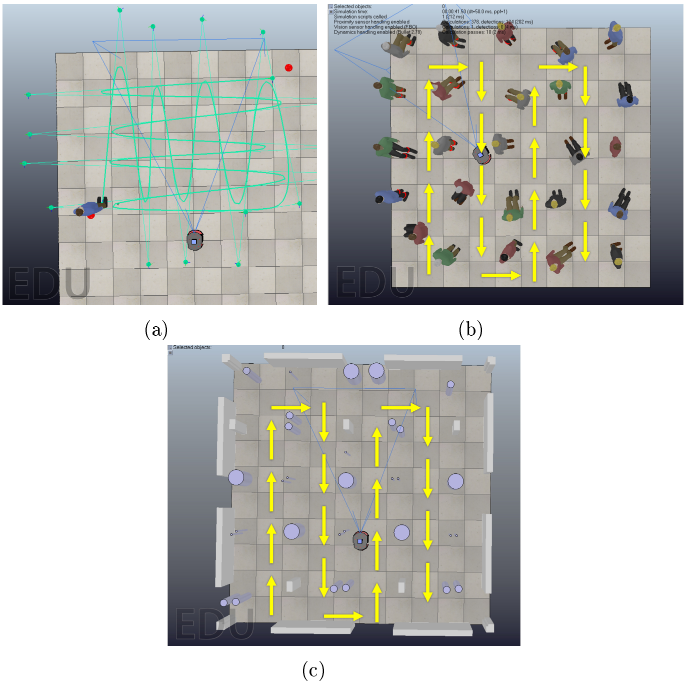

(a) (b) (c)

Fig. 1: Multiple scenarios for data collection: (a) Single person following a path; (b) Multiple people in stationary positions; (c) Objects-only scenario.

The detailed process for collecting the 2D LRF data is as follows:

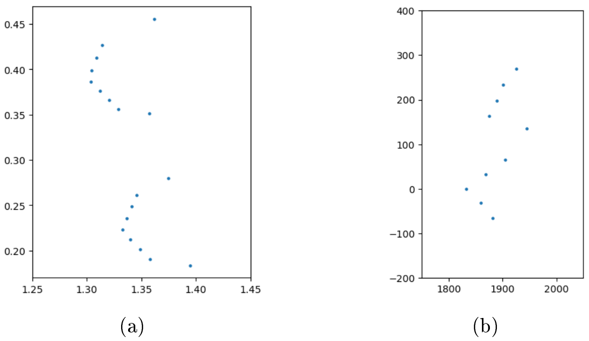
- **Single Person Scenario**: The scene incorporates only the Pioneer robot model and a person model that follows a predefined path (Bill on track). The robot remains stationary while the person navigates a path composed of consecutive S shapes, initially in a perpendicular direction, which then shifts to a parallel orientation. The exact path is illustrated in figure 1a. This scene is designed to capture samples of legs in motion. This specific scenario is crucial as leg detection of a moving person appears slightly distorted due to the sampling speed of the 2D LRF.

- **Multiple Person Scenario**: In this scene, 25 person models are used, comprising 13 standing models (Standing Bill) and 12 sitting models (Sitting Bill). Each model is randomly rotated to ensure a variety of perspectives before data collection begins. The robot moves along a predefined path, marked by yellow arrows in figure 1b, while the person models remain stationary.
- **Objects Only Scenario**: This setup is filled with diverse objects including walls, rectangles, and cylinders of various sizes, arranged to simulate a complex environment. The goal is to gather non-leg samples, enhancing the dataset's diversity. The robot navigates a predefined path, indicated by yellow arrows in figure 1c, during data collection.

After a comprehensive data collection process, we now turn our attention to the next section where we will delve into the specifics of how we have applied clustering to our collected data.

5.2 Clustering

In our simulated environment, we chose to use settings that replicate those of the physical LRF SICK LMS200 laser sensor. This decision ensures that the simulated robot's operational characteristics align closely with those of its real-world counterpart. Specifically, the simulated Pioneer robot employs a 2D LRF that provides a 180-degree field of view with an angular resolution of 0.5 degrees, positioned approximately 30 cm above the ground.



(a)

(b)

Fig. 2: Outline of a person's leg in both simulated (a) and real world (b) environments.

When a standing person is scanned by this LRF, the image appears as two closely spaced semi-circumferences, as illustrated in Figure 2. Each sensor scan generates 361 polar coordinates, which are then transformed into Cartesian coordinates for further analysis.

One of our primary objectives is to discern features within the sensor's field of view, particularly focusing on the lower portion of a person, such as legs. To approach this challenge, we adopted a binary classification framework. The goal is to categorize the data into two groups: representations of human legs and all other objects. While this method enhances our ability to distinguish between various shapes, it is important to recognize the limitations inherent in using laser-based technology. For instance, objects like table legs or other cylindrical items that mimic the dimensions of human legs can pose challenges for accurate differentiation. Thus, while our approach provides a structured way to analyze the sensor data, the precision of distinguishing between very similar shapes solely through this method can vary.

Given the variable nature of the cluster sizes in our dataset, we incorporated a variety of features for cluster characterization. These features are drawn from established practices in related research and general techniques employed for polygon characterization.

5.3 Features extraction

To prepare data samples for machine learning tasks, we extract a comprehensive set of geometric features from the clusters, expanding upon successful methods like the Leigh's detector [16]. The comprehensive set of features utilized to describe the clusters includes:

- **Depth:** Measurement of the cluster's extent from the front to the back.
- **Width:** Measurement across the widest part of the cluster perpendicular to the depth.
- **Perimeter:** The total length around the boundary of the detected cluster.
- **Radius:** The radius of the circle that best fits the point cloud as determined by the Taubin fit.
- **Sigma:** The mean squared error (MSE) between the fitted circle and the actual points in the cloud, indicating the fit's accuracy.
- **Area:** Area of the polygon calculated using the shoelace formula.
- **Distance:** Distance from the centroid of the point cloud to the 2D LRF, providing a spatial reference.
- **Number of points:** Number of points comprising the cluster, reflecting its density and complexity.
- **Angles:** Sum of the internal angles of the polygon, which helps in understanding the geometric structure.

Given the typical appearance of a leg in laser-based imagery resembling semicircles, the algebraic circle fit method developed by G. Taubin [24] is applied. This technique allows us to identify the circle that most accurately represents the shape of the point cloud.

With a comprehensive set of features defined for each cluster, we them proceed to the model training phase.

5.4 Model Training

The next phase of our research involves training various machine learning models, primarily selected from the scikit-learn library. Our selection encompasses a range of robust algorithms including Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Random Forest, and AdaBoost. Additionally, XG-Boost, though not part of scikit-learn, has been included due to its efficacy in handling large and complex datasets.

The simulated environment dataset comprised 18,890 positive samples (60%) and 12,536 negative samples (40%), revealing an initial class imbalance. To address this, we balanced the dataset by randomly sampling from the positive class to match the size of the negative class, using a fixed seed for consistent results. This adjustment ensures equal representation of both classes, facilitating more effective and unbiased model training.

We allocated 75% of this balanced dataset for training and the remaining 25% for testing. The splitting process was conducted through a stratified random selection using a predefined seed to ensure the reproducibility of our results. To optimize the models, we utilized a grid search coupled with 5-fold cross-validation to determine the best hyperparameters for each algorithm. This structured approach aims to maximize the predictive accuracy and reliability of our classification models.

With the model training complete, we now move on to evaluate their performance. In the next section, we will present key metrics such as accuracy, precision, recall, and the f1-score, providing a clear overview of the different models' effectiveness.

5.5 Performance metrics

Table 1: Performance metrics of various classification models evaluated on the test split of the dataset, sorted by f1-score. The highest scores for each metric are highlighted in bold.

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	**0.994**	**0.994**	0.993	**0.994**
Random Forest	0.992	0.990	**0.995**	0.992
KNN	0.992	0.991	0.993	0.992
SVM (kernel=rbf)	0.990	0.990	0.990	0.990
AdaBoost	0.981	0.977	0.985	0.981
SVM (kernel=linear)	0.966	0.946	0.989	0.967
SVM (kernel=poly)	0.963	0.957	0.969	0.963

The analysis of various machine learning algorithms on the dataset highlights distinct performances, suitable for different practical applications. The XGBoost exhibited superior metrics (accuracy, f1-score, precision each at 0.994, and recall at 0.993), suggesting its robustness in achieving high predictive accuracy. It is followed closely by the KNN and Random Forest, both showcasing high accuracy and f1-score around 0.992, with Random Forest slightly outperforming in recall (0.995). For detailed performance metrics of each model, see Table 1.

The SVM with an RBF kernel also performed commendably, maintaining balanced metrics around 0.990, although other kernel types showed reduced effectiveness. The AdaBoost and SVM with a linear kernel demonstrated moderate success.

Moving forward from our simulation results, the next phase of our study involves applying the best models to a real-world setting to validate their practical efficacy. This step is crucial for transitioning from a controlled experimental environment to actual operational scenarios, where variables and conditions are more dynamic and unpredictable.

6 Real World Environment

In the real-world phase, we leverage an automatic labeling process facilitated by deep learning models specializing in pose estimation. This approach aims to enhance the accuracy of data annotation, which is pivotal for training and validating our machine learning models under real-world conditions.

6.1 Data Collection Process

Data was gathered using the PeopleBot mobile robot equipped with an LRF SICK LMS200. The robot captured multiple sequences, collecting both 2D laser data and visual images via an onboard camera on the Jetson TX2 development board. The scenarios involved both static and dynamic elements: the robot remained stationary while multiple individuals walked in its vicinity, and in other tests, the robot moved, simulating the task of following a designated person. This movement was managed using an industrial joystick. Although the focus of this study does not include robot control—which will be explored in future work—the recorded scenarios were designed to reflect the expected operational conditions of the mobile robot in real-world settings.

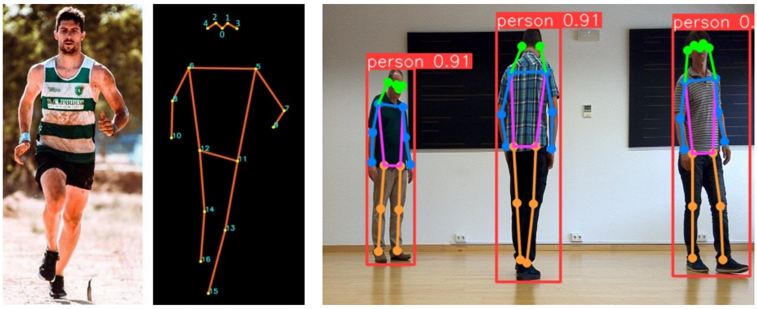
The primary environments for these recordings were office-like spaces, including corridors, office rooms, and larger indoor areas such as conference rooms and hallways. This setup aimed to simulate typical interactions and navigational challenges the robot would face in a working environment.

In total, 9,241 frames across three different sequences were captured with the robot in stationary positions, and 6,320 frames in five sequences where the robot was manually controlled to mimic the behavior of following a person.

Having described the data collection process and the operational settings, we now turn our attention to the methodology for processing this data. The next section delves into the automatic data labeling process, which utilizes pose estimation techniques.

6.2 Automatic labeling using bounding box and keypoints

Leveraging 2D range data for leg detection involves machine learning techniques applied to datasets that ideally represent realistic operational scenarios, as discussed in Section 1. Our approach uses a pre-trained CenterNet HourGlass104 Keypoints (CHK) model to identify bounding boxes and keypoints for legs, facilitating the automatic labeling of 2D range data. The process integrates the detection of people in images from an onboard camera with the localization of corresponding laser points.



(a)

(b)

Fig. 3: (a) Keypoints diagram of CenterNet HourGlass104 Keypoints. (b) Bounding box and keypoints supplied by CenterNet Hour Glass104 Keypoints detector.

Bounding boxes and keypoints for human legs are defined by specific keypoints: kp^i_{13} and kp^i_{15} for the left leg, and kp^i_{14} and kp^i_{16} for the right leg, as illustrated in Fig. 3. 2D laser points are clustered using the jump distance algorithm, with valid clusters transformed and projected onto image coordinates using the robot's camera calibration parameters.

The calibration process involves capturing images of a checkerboard pattern from various angles to compute the camera's intrinsic (focal length, optical centers) and extrinsic (position and orientation in relation to the robot) parameters. This alignment is essential for enabling a seamless overlay of laser data onto the visual images and facilitates the accurate identification and labeling of leg clusters based on the proximity of laser points to the keypoints on the images.

The process begins by projecting the 2D LRF data onto the image captured by the RGB camera using a transformation matrix. This alignment allows us to overlay the LRF data onto the corresponding visual content accurately. Once aligned, the CenterNet model identifies key points on the person, specifically kp^i_{13}, kp^i_{15}, kp^i_{14}, and kp^i_{16}, which correspond to the left knee, left foot, right knee, and right foot, respectively.

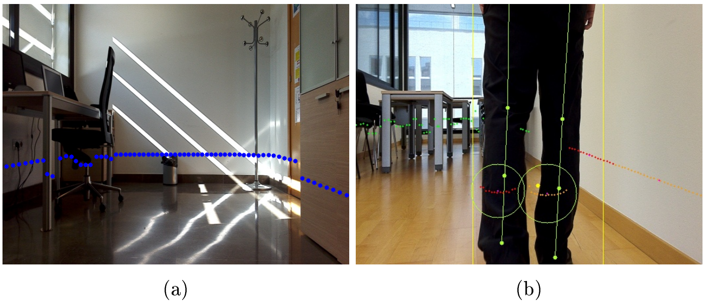

(a)
(b)

Fig. 4: (a) Projection of 2D laser range data to colour image. (b) Assignment of the projections of clusters to detect legs.

Considering the LRF primarily captures data at shin height and there are no direct equivalent keypoints, a new reference point for each leg is created by calculating the midpoint between the knee and foot keypoints. The distance between this newly established midpoint and the nearest LRF point cluster is then measured. If this distance is within a predefined threshold and the rest of the points in that cluster are nearby, then the cluster is marked as a positive leg detection, indicating the presence of a person's leg.

Clusters not identified as legs are labeled as negative samples, representing background objects. This selective filtering aims to ensure the data used for training and validation is both relevant and accurately labeled.

The process is visualized in Fig. 4a which illustrate the integration of image and laser data, and Fig. 4b showing the assignment of clusters to detected leg keypoints.

With our data now accurately labeled using the automated process, we proceed to characterize the data using the techniques outlined in section 5.2 before moving to the next phase where once again we will train a selection of machine learning models. This stage will leverage adjustments and enhancements, informed by the insights gathered during the prior phases.

An experimental study, outlined in the following section, selects the optimal machine learning algorithm for classifying these features, ensuring robust leg detection using 2D range data.

6.3 Model training

Following the evaluation criteria outlined in section 5.5, we narrowed down the selection to models that achieved an F1-score of 99% or higher. This criterion ensured that only the most accurate models were considered for training on real-world data. This select group includes: XGBoost, Random Forest, KNN and SVM (with RBF kernel).

Given the close performance results of these models, a new metric, inference time, has been introduced to find the best trade-off between computing time and performance. This metric measures the speed at which a model can process input data and return a result, an essential factor for real-time applications. The inclusion of inference time helps assess the practicality of deploying these models in scenarios where response speed is critical.

After characterization, we obtained a dataset consisting of 19,632 samples, balanced evenly across both classes. The same methodology outlined in section 5.4 was used to split the dataset, and also for selecting the best hyperparameters for the different models.

With our dataset prepared and models optimized, we then proceed to a detailed evaluation. In the upcoming section, we will analyze model effectiveness using both traditional accuracy metrics and the newly introduced inference time, ensuring a better understanding of their potential for real-world applications.

6.4 Performance metrics

To comprehensively assess the performance of our models, we focused on calculating the mean inference time. This metric represents the average processing time required for 1,000 samples across 19 distinct splits of our complete balanced dataset. Additionally, we calculated the standard deviation to evaluate the variability in processing times across these splits.

These tests were conducted using a high-performance laptop equipped with an Intel Core i7-12650H processor, featuring a 24 MB L3 cache, speeds up to 4.70 GHz, 10 cores, and 16 threads. The system also includes 16 GB of DDR5 RAM operating at 4800 MHz across two modules. The laptop also features an NVIDIA GeForce RTX 4070 mobile GPU; however, this component was not utilized in the current stage of testing as our analysis was confined to machine learning models that did not leverage GPU acceleration.

Table 2: Performance metrics of various classification models evaluated on the test split of a real world dataset, sorted by f1-score. The highest scores for each metric are highlighted in bold.

Model	**F1-Score**	**Accuracy**	**Precision**	**Recall**	**Mean (ms)**	**STD (ms)**
XGB	**0.9944**	**0.9944**	0.9946	**0.9943**	**2.0**	**0.4**
RandomForest	0.9941	0.9941	0.9953	0.9928	23.1	1.6
KNeighbors	0.9930	0.9930	0.9946	0.9914	57.4	4.8
SVC (kernel=rbf)	0.9913	0.9914	**0.9960**	0.9867	343.1	12.7

In the real-world dataset, the evaluation of machine learning algorithms (Table 2) has yielded results that closely align with those obtained in the simulated environment, demonstrating a strong consistency across different experimental setups. The XGBoost model has continued to exhibit exceptional performance, achieving the highest f1-score of 0.9944, which correlates precisely with its accuracy. This high degree of accuracy and precision (0.9946) coupled with a nearly identical recall (0.9943) underscores its robustness in handling real-world data. Additionally, the algorithm benefits from a notably efficient execution time, averaging just 2.0 milliseconds per instance with a minimal standard deviation of 0.4 milliseconds, highlighting its suitability for applications requiring high-speed data processing.

Random Forest also shows excellent performance, slightly surpassing XGBoost in precision but with a longer execution time. However, its capacity for easy explainability may render it particularly valuable in scenarios where understanding model decisions is crucial, despite the slower performance.

K-Nearest Neighbors (KNN) and the Support Vector Machine (SVM) with an RBF kernel, while robust, are slower and slightly less effective in terms of recall and overall f1-score compared to XGBoost and Random Forest. The significant processing time of the SVM, in particular, could be a drawback in rapid decision-making environments. These findings confirm the suitability of XGBoost and Random Forest for practical applications, emphasizing the need to balance performance metrics and operational requirements when selecting algorithms.

For an in-depth understanding of the models configuration settings, we present the specific parameters used for the three top-performing algorithms in our study:

- **Random Forest** is set with an entropy criterion, a maximum depth of 12, and 100 estimators. The maximum features are determined by the square root of the total number of features, which optimizes the diversity and computational efficiency of the model.
- **XGBoost** a maximum depth of 6 with 200 estimators to balance learning capacity and prevent overfitting, ensuring high precision and recall in its performance metrics.
- **KNN** employs the ball tree algorithm with 11 neighbors and weights by distance, using the Manhattan distance metric (p=1). This configuration enhances sensitivity to local data structures, beneficial for our complex dataset.

7 Conclusions

The integration of deep learning-based pose estimation with traditional 2D Laser Range Finders (LRFs) has significantly enhanced mobile robots' ability to detect human presence. The automated fusion of image and 2D range data streamlines the creation of accurate training datasets, crucially reducing the manual effort and time involved in labeling.

Experimental results validate the effectiveness of the proposed system, particularly highlighting the performance of the XGBoost model, which demonstrates high accuracy and reliability in both simulated and real-world settings. This model proficiently distinguishes between human figures and other objects, ensuring practical utility in operational environments.

Future work will focus on developing people tracking capabilities using the predictions generated by models trained in this study. This advancement aims to enhance the real-time tracking and interaction capabilities of mobile robots in complex environments, further bridging the gap between laboratory settings and real-world applications.

References

1. Aguirre, E., García-Silvente, M.: Using a deep learning model on images to obtain a 2d laser people detector for a mobile robot. International Journal of Computational Intelligence Systems **12**(2), 476–484 (Jan 2019). https://doi.org/10.2991/ijcis.d.190318.001
2. Aguirre Molina, E., García Silvente, M., Pascual, D.: A multisensor based approach using supervised learning and particle filtering for people detection and tracking (2016). https://doi.org/10.1007/978-3-319-27149-1_50
3. Ahmed, I., Ahmad, M., Rodrigues, J., Jeon, G.: Edge computing-based person detection system for top view surveillance: Using CenterNet with transfer learning. Applied Soft Computing **107**, 107489 (05 2021). https://doi.org/10.1016/j.asoc.2021.107489
4. Benedek, C.: 3D people surveillance on range data sequences of a rotating lidar. Pattern Recognition Letters **50**, 149–158 (2014). https://doi.org/10.1016/j.patrec.2014.04.010
5. Cai, Z., Vasconcelos, N.: Cascade R-CNN: Delving into high quality object detection. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6154–6162 (2018). https://doi.org/10.1109/CVPR.2018.00644
6. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1302–1310 (2017). https://doi.org/10.1109/CVPR.2017.143
7. Chen, Y., Song, B., Zeng, Y., Du, X., Guizani, M.: A deep learning-based approach for fault diagnosis of current-carrying ring in catenary system. Neural Computing and Applications pp. 1–13 (07 2021). https://doi.org/10.1007/s00521-021-06280-4
8. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: Object detection via region-based fully convolutional networks. In: Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 29 (2016)
9. Girshick, R.: Fast R-CNN. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1440–1448 (2015). https://doi.org/10.1109/ICCV.2015.169
10. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014). https://doi.org/10.1109/CVPR.2014.81
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2980–2988 (2017). https://doi.org/10.1109/ICCV.2017.322
12. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., Murphy, K.: Speed/accuracy trade-offs for modern convolutional object detectors. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3296–3297. IEEE Computer Society, Los Alamitos, CA, USA (2017). https://doi.org/10.1109/CVPR.2017.351

13. Intelligence, S.S.: Sick sensor intelligence, LMS200 (2002), http://www.mysick.com
14. Kim, J.J., On, B.W., Lee, I.: High-quality train data generation for deep learning-based web page classification models. IEEE Access **9**, 85240–85254 (2021). https://doi.org/10.1109/ACCESS.2021.3086586
15. Lafuente-Arroyo, S., Martin-Martin, P., Iglesias-Iglesias, C., Maldonado-Bascon, S., Acevedo-Rodriguez, F.J.: RGB camera-based fallen person detection system embedded on a mobile platform. Expert Systems with Applications p. 116715 (2022). https://doi.org/10.1016/j.eswa.2022.116715
16. Leigh, A., Pineau, J., Olmedo, N., Zhang, H.: Person tracking and following with 2d laser scanners. In: 2015 IEEE International Conference on Robotics and Automation (ICRA). pp. 726–733 (2015). https://doi.org/10.1109/ICRA.2015.7139259
17. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014)
18. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
19. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 28 (2015)
20. Rubagotti, M., Tusseyeva, I., Baltabayeva, S., Summers, D., Sandygulova, A.: Perceived safety in physical human-robot interaction-a survey. Robotics and Autonomous Systems **151** (2022). https://doi.org/10.1016/j.robot.2022.104047
21. S. Mohamed, I., Capitanelli, A., Mastrogiovanni, F., Rovetta, S., Zaccaria, R.: Detection, localisation and tracking of pallets using machine learning techniques and 2d range data. Neural Computing and Applications **32**, 8811–8828 (07 2020). https://doi.org/10.1007/s00521-019-04352-0
22. Sharif, M.H.: Laser-based algorithms meeting privacy in surveillance: A survey. IEEE Access **9**, 92394–92419 (2021). https://doi.org/10.1109/ACCESS.2021.3092687
23. Sun, Y., Li, Z., Wang, L., Zuo, J., Xu, L., Li, M.: Automatic detection of vehicle targets based on centernet model. In: 2021 IEEE International Conference on Consumer Electronics and Computer Engineering (ICCECE). pp. 375–378 (2021). https://doi.org/10.1109/ICCECE51280.2021.9342498
24. Taubin, G.: Estimation of planar curves, surfaces and nonplanar space curves defined by implicit equations, with applications to edge and range image segmentation. IEEE Trans. PAMI **13**, 1115–1138 (1991)
25. Vijayakumar, A., Vairavasundaram, S.: Yolo-based object detection models: A review and its applications. Multimedia Tools and Applications (Mar 2024). https://doi.org/10.1007/s11042-024-18872-y
26. Yuan, L., Mao, J., Zheng, H.: Ear detection based on CenterNet. In: 2020 IEEE 2nd International Conference on Civil Aviation Safety and Information Technology (ICCASIT. pp. 349–353 (2020). https://doi.org/10.1109/ICCASIT50869.2020.9368856
27. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points. In: arXiv preprint arXiv:1904.07850 (2019)
28. Zhou, X., Wang, D., Krähenbühl, P.: Objects as points (2019). https://doi.org/10.48550/arXiv.1904.07850