Detection and location of domestic waste for planning its collection using an autonomous robot

1st Pascual Tornero
AUROVA Lab
Computer Science Research Institute
University of Alicante
Alicante, Spain
Email: pascual.tornero@ua.es

2nd Santiago Puente
AUROVA Lab
Computer Science Research Institute
University of Alicante
Alicante, Spain
Email: santiago.puente@ua.es
ORCID: 0000-0002-6175-600X

3rd Pablo Gil
AUROVA Lab
Computer Science Research Institute
University of Alicante
Alicante, Spain
Email: pablo.gil@ua.es
ORCID: 0000-0001-9288-0161

***Abstract*—This paper presents an approach of a detection and location system for waste recognition in outdoor environments that can be usable on an autonomous robot for garbage collection. It is composed of a camera and a LiDAR. For the detection task, some YOLO models were trained and tested for classification of waste by using a own dataset acquired from the camera. The image coordinates predicted by the best detector are used in order to compute the location relative to the camera. Then, we used the LiDAR to get a global waste location relative to the robot, transforming the coordinates of the center of each trash instance. Our detection approach was tested in outdoor environments obtaining a mAP@.5 around 0.99 and a mAP@.95 over 0.84, and an average time of detection less than 40 ms., being able to make it in real time. The location method was also tested in presence of objects at a maximum distance of 8 m., obtaining an average error smaller than 0.25 m.**

***Index Terms*—waste detection, robot location, deep learning, RGBD**

I. Introduction

Over the last few years, one of the most important areas to improve is the waste management, which is an area that is being developed in the last years and supported by the EU. In the period from 1995 to 2019, although the generation of waste increased, the total amount of municipal waste landfilled diminished [1]. According to the consulted information, in 1995 the municipal waste landfilled were 121 million tonnes (286 kg per capita) while in 2019 this quantity fell to 53 million tonnes (119 kg per person), corresponding to an average annual decline of 3.3% which is a great fact. This was possible due to the effort made to increase the quantity of waste that was being recycled (recycling and composting) from 18% in 1995 to 48% in 2019, but although the governments have managed to achieve these results, they need to keep improving them.

This research was made with the objective of providing a solution to waste accumulation in outdoor environments, which is one of the most common cases of bad waste management. In this paper, we propose a sensory system that makes use of color images, depth images and other sensors signals. It is able to detect instances of waste, like plastic bottles, cans, boxes, etc. classifying them in four categories (plastic bottle, box, glass bottle, can). It obtains the 3D coordinates of each waste instance respect to a fixed reference system. Our sensory system is mounted at a robotic mobile platform which has also been installed a robotic arm. The whole robotic system is able to navigate as well as detect and locate throughout navigation each waste and pick it up. In this paper, only the sensory system for waste detection and location in the environment is shown. The pick up process was already described in [2].

First of all, to get a slight idea of how to deal with the problem, similar works from another authors were seen. In [3], authors made use of a MLP neural network using images with a resolution of 150x150 pixels to train a system, providing for each image the bounding boxes of the detected waste. In another previous work [4], authors also used a neural network to detect garbage, which was a pre-trained network called MobileNet [5]. They also performed a local location method of the waste detected relative to a robot position instead of a global location relative to a world reference frame. To do this, they used an alignment process using as information, the center of the color image, the bounding box center of each detection and the arrangement of the camera. In the same line, in [6], it was proposed a deep learning framework with different architectures, such as Densenet, InceptionResnet-V2, MobileNet and Xception, which were tested on Trashnet dataset [7] to provide a most efficient approach for garbage detection and classification. Besides, the authors of [8] used an algorithm based on deep learning convolutional neural network called Efficientnet in order to classify garbage in four categories (recyclables, kitchen garbage, hazardous garbage and other garbage). Also in this research, data augmentation and normalization are carried out to solve the problem of small amount of data sets and different sizes of pictures.

As seen before, the majority of state-of-the-art proposals are based on 2D images. However, some recent deep learning approaches are based on 3D data. One of these 3D-based techniques that should be highlighted is PointNet [9], which tries to solve the problems of irregularities and messy set of

This research was funded by Spanish Government through the project RTI2018-094279-B-I00. Besides, computer facilities were provided by Valencian Government and FEDER through the IDIFEFER/2020/003

information typical of point clouds in order to detect, classify and segment scenes without using any convolution. Other example of an approach based on 3D data is A-CNN [10], which introduces a model to perform annular convolutions to a set of 3D points, solving the problems of common convolutions applied to 3D sets of information.

The rest of this paper is distributed as follow. In section II is described the proposed system for detection and location waste. Later, we show experiments in a realistic environment to test our system in section III and we discuss the achieved results in section IV. Finally, a section of conclusions is presented.

II. System description

Our sensory system is implemented on a robotic mobile platform whose name is BLUE [11] which is shown in Fig. 1. This robot has multiple sensors to work in different kind of conditions, but in this work, only two sensors were used: a RGBD camera and a LiDAR. It's used the signal from the camera Intel RealSense D435i to acquire color and depth images, and also the transformed signal from a 3D LiDAR to be able to get the global position and orientation of the robot in each moment. The camera obtains both types of images, color and depth, with a resolution of 1280x720 pixels, works at a frequency of 15 FPS and its ideal measurement range is between 0.3 and 3 meters. From the other part, the LiDAR, model Velodyne VLP16, works at a frequency of 10 Hz., uses a total of 16 measurement channels and can measure distances up to 100 meters.

The first task that our system performs is the detection and classification of instances of waste that appear in the color images acquired from the camera. To do this, we selected YOLO (You Only Look Once) architectures, a 2D-based method that returns the bounding box of the detection and its classification. YOLO family is a well-known single-stage object detector that works in real-time. This family is composed of several models, each new version tries to improve the performance of the previous one. Two representative models are the third and fifth version. On the one hand, YOLOv3 [12] was selected because it is a hybrid neural approach which combines the networks used in YOLOv2 [13] and a residual networks instead of anchors blocks. The backbone is called Darknet-53. Unlike previous versions of YOLO which predicts the output at the last layer, YOLOv3 predicts bounding boxes at 3 different

Fig. 1. BLUE robot

Fig. 2. Examples of the images taken from our own dataset

scales. On the other hand, the election of YOLOv5 [14] was motivated because it is smaller than YOLOv4 [15] and faster comparing inference times. YOLOv5 is very different from all other previous versions since it is not based on Darknet. Both models, YOLOv3 and YOLOv5 were trained with our own dataset. This consists of images taken in outdoors environments at our university. The images were taken by the camera in a wide variety of conditions: different places and scenarios to perform a good training, for example in environments with different surfaces like ground, with and without grass, pavements of various colors and textures, and under sunny and cloudy conditions, as shown in Fig. 2.

Each image presents different types of garbage, which can be classified in four categories. The first class is "glass bottle", which is the class given to all the waste made with glass, like wine bottles, beer bottles etc. The second is "box", which groups all the trash that is in box-shaped or a cube. In third place, in the category called "can" all the cylinder-shaped garbage that isn't a bottle is contained, like soft drink cans among others examples. The last class is "plastic bottle", which includes the biggest part of the trash made with plastic, like bottles, recipients etc. that it's not included in the previous categories. The number of categories had been selected to made an easy initial classification by a mobile robot without the necessity of great number of stores in the mobile platform. In Fig. 3 are shown an amount of waste that was used to take the example images of our dataset.

Once the images were acquired, we did a manual labeling of all the waste instances in each photo. Specifically, we annotated, on a text file, the values of center on x and y axis, width, height and category of the bounding box of each instance on the photo. It should be noticed that, with the objective of making a good dataset, we decided to include some few photos were no trash instance appear, because this helps later to reduce the quantity of false positives (detection made by the neural network that are not a garbage instance). After completing this work, our dataset is composed of a total of 1201 scene images, where few of them are images with no instances. Besides, the number of instances in our dataset is 8652 and its distribution by categories is shown in Table I. As you can see, the number of instances for each category

TABLE I
INFORMATION ABOUT OUR DATASET

	Captured	Augmented	% of total
Total of instances	8652	69216	100.0
Glass bottle instances	1842	14736	21.2
Plastic bottle instances	2306	18448	26.7
Box instances	2335	18680	27.0
Can instances	2169	17352	25.1

is balanced, less in the case of ”glass bottle”, which is lower because it was hard to maintain a balance between all the categories when the photos of the environment were taken, where objects of all classes were deployed randomly.

Let me note that we also applied the data augmentation technique to increase sharply the number of example images and instances without having to label them again. The data augmentation is a commonly used process to make bigger an amount of data when it isn’t so large or we want to have more examples [16]. On this occasion, seven new images were obtained from each original photo, multiplying by eight the number of examples. This is achieved applying to each image transformations in two steps, the first consists in flip vertically, horizontally and both flips to get 3 new images, and then, in the second step, used a convolutional filter of 5x5 kernel, to introduce noise, to the last 4 photos to obtain another 4. It also needs to create new text files with the new labels, that are earned by an automatic process made with a Python script. After applying the data augmentation, our dataset is composed of 7192 scene images.

The next step is completing the location task, where the objective is to locate the trash detected previously on a 3D space, which will be always referenced relative to a fixed reference frame in the world. To perform this task, two values are needed, the global position of the robot, which is its 3D position and orientation in all axis, and also the position of the waste relative to the camera.

Fig. 3. Examples of the trash used in the experimentation scenario

In this case, the global position and orientation of the robot will be the global position and orientation of the LiDAR L, and it is achieved using the information given by the LiDAR measures, which is used to calculate the movement that the robot has made between two frames and return the 3D position and the orientation, in a quaternion shape [17]. Besides, the 3D location of each trash instance, relative to the camera frame C, is estimated using both, the central point of each detected bounding box $[u, v]$ in the image and the camera parameters used to capture it. This is calculated with according to equation 1. There, $[x_c, y_c, z_c]$ are the coordinates of the point in 3D in meters relative to camera, u is the pixel of the selected point in the 2D image for the X axis, v is the same but in this case for the Y axis, d is the depth of this point in meters, and finally f_u, f_v and u_0, v_0 are the focal length and optical center from intrinsic parameters of the camera, respectively.

$$\begin{cases} x_c = d \\ y_c = -\frac{(u-u_0)d}{f_u} \\ z_c = \frac{(v-v_0)d}{f_v} \end{cases} \tag{1}$$

Once the waste coordinates $[x_c, y_c, z_c]$ relative to the camera C were computed, it’s necessary to transform them to a global location respect to the fixed 3D LiDAR frame L by applying equation 2, where $[x_c, y_c, z_c]$ are 3D coordinates relative to camera and $[x_l, y_l, z_l]$ the new ones relative to LiDAR, in meters. The 4x4 matrix got its values from the positioning in the BLUE robot of the RGBD camera relative to the LIDAR, they are empirical values. Later, it’s necessary to transform this location from LiDAR frame L to the global frame of the robot R. This is achieved applying the rotation of the LiDAR contained in a quaternion format, and then, adding up the obtained values to the global 3D position of the LiDAR. This math process is performed with the equation 3, where $[x_l, y_l, z_l]$ are the previous 3D values and $[x_r, y_r, z_r]$ the new ones, in meters, and Q° and Q^* are the quaternion values and their conjugated, respectively.

$$\begin{bmatrix} x_l \\ y_l \\ z_l \\ 0 \end{bmatrix} = \begin{bmatrix} 1 & -0.024 & -0.02 & -0.01 \\ 0.029 & 0.935 & 0.354 & 0.003 \\ 0.011 & -0.353 & 0.935 & 0.079 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 0 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} 0 \\ x_r \\ y_r \\ z_r \end{bmatrix} = Q^\circ(0, x_l, y_l, z_l)^\circ Q^* \tag{3}$$

Once this transformation process is done then the detection and location can be shown in the image, and the locations of the waste can be saved so that the robot is able to use them when it’s necessary. It should be noted that all this process is controlled by ROS (Robot Operating System), which is used to send and receive the messages from the sensors and the partial and total results. A scheme of the whole process carried out by our sensory system can be seen in Fig. 4.

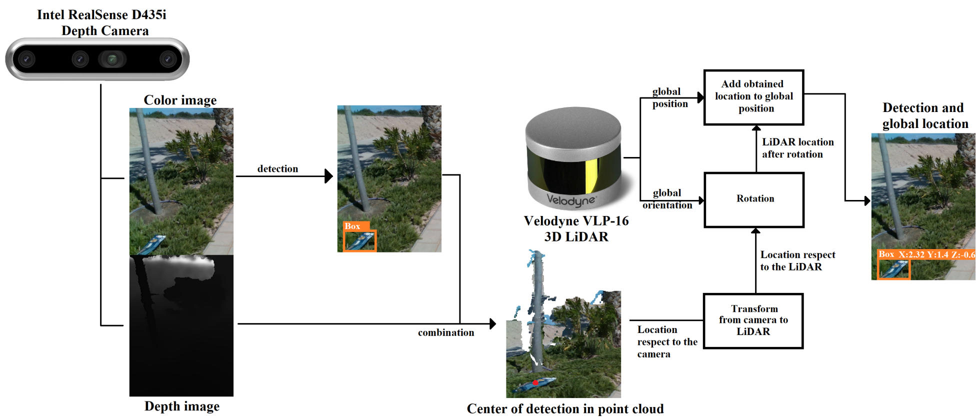

Fig. 4. Workflow of our proposal

III. Experiments

In this section, we tested the proposed approach in real operating conditions. The experiments are divided in two parts. The first part, is to measure the performance for the detection task. To do this, we firstly perform the training and obtain a parametrically optimised model for the selected YOLO versions. We used a NVIDIA DGX-A100 GPU to accomplish it successfully. Specifically, we mainly carried out the hyper-parametric adjustment of the several implementations of YOLOv3 and YOLOv5. The learning rate was fixed on 0.005 for all the cases. Then, the second part of the experimentation section consists in testing the 3D location task of the previously detected trash from the YOLO models.

For the detection task, we proceeded as follows. First, we trained with a little dataset (a third part of the original) and for a number of epochs that doesn't exceed the quantity of 50. After, we decided to increase the size of our dataset to the actual one, and establish the number of epochs to train in 100. Then, the results improved respect to the previous. At this moment, it was decided to augment the number of training epochs to 150. We made use of the dataset without the data augmentation and selected the train parameter called multi-scale. This option makes that the training is done using the example images on different proportions in order to make it more invariant to the size of the trash, what is expected to help to detect the waste located far from the robot. Apart from that new parameters usage, it was determined to use different available sizes of the neural network for both versions, because YOLO lets to select a training model that can be tiny (YOLOv3-tiny [18] and YOLOv5s [14]) or heavier, with a bigger structure and with more adjustable weights (YOLOv3-SPP [19] and YOLOv5l [14]). Finally, after training with the two YOLO versions and with a dataset where the data augmentation is not applied, it was opted to train now with only the YOLOv5, more specifically with the YOLOv5l model, with the multi-scale parameter activated, with the dataset after using the data augmentation and for a total of 250 epochs.

Besides, to test the task of locating the garbage previously detected by the neural detector, some trials were made in which the waste was disposed on known positions, so it can be seen if the system returns the same or similar values and then we can study the variability of them, like a position error. One thing to keep in mind is that the RGBD camera that is being used doesn't return very confident values at a distance higher than 3 meters. This fact provides a limitation of the practical experiment without compromising the global location algorithm. In Fig. 5 is shown the real locations for waste objects detected in our experiments. In particular, in our experimentation, we used 3 objects from the category glass bottle which were labeled as GB1, GB2 and GB3, and 5 objects from the category can which were labeled as C1, C2, C3, C4 and C5. In section IV, these waste objects are also used to estimate their 3D spatial locations on different moments.

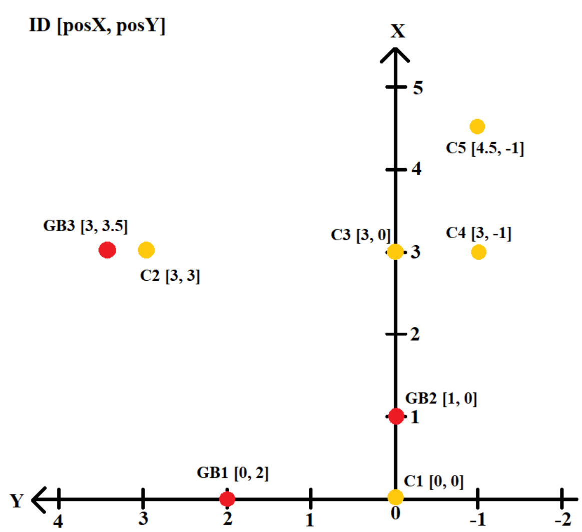

Fig. 5. Location results in meters for waste objects in our dataset

IV. Results

In this section, the experimental results obtained for detection and location are shown and discussed. To analyze the

TABLE II
INFORMATION ABOUT THE PERFORMED TESTS TO DETECT OBJECTS IN THE SCENE

Model	Epochs	Dataset	Precision	Recall	mAP@.5	mAP@.5:.95	Speed (ms)
YOLOv3-tiny	150	Not augmented	0.861	0.828	0.862	0.496	9
YOLOv3-SPP	150	Not augmented	0.935	0.928	0.958	0.646	64
YOLOv5s	150	Not augmented	0.956	0.908	0.954	0.643	11
YOLOv5l	150	Not augmented	0.947	0.939	0.97	0.671	35
YOLOv5l	250	Augmented	0.991	0.992	0.9951	0.8424	36

different tests that were performed, we presented the Table II. This table shows the detection algorithm results. In this table, there are some different scores to evaluate the performance of the detector models. All these values are within range $[0, 1]$ and the best possible result is 1. The precision indicates the relation between correct detections and the total number of detections, and it is calculated dividing the number of true positives between the total number of positives. The recall shows the capacity of finding the positives in the samples, without taking in care the false positives, and it is calculated dividing the number of true positives between the number of true positives plus false negatives. The mAP is the average of the relationship between precision and recall on its curve that is the same than the area under the curve precision-recall. The value of mAP@.5 shows the relationship when the recall values are under 0.5, and mAP@.95 when the recall value is between 0.5 and 0.95. The mAP@.95 is the most important value, because if it is high, it indicates that the system is detecting the biggest part of the samples (high percentage of true positives) without detecting as waste things that aren't waste (low number of false positives). Finally, the speed is the time needed to make the detection in one image. In the case of the table, all the values are the average of these values for all the samples of the test set.

Comparing the YOLOv3-tiny and YOLOv5s, we obtained that YOLOv5s gives better results (0.958 mAP@.5 versus 0.862 mAP@.5) in practically the same time, since it only spent 2 milliseconds more. A similar situation happens when we used two largest models as YOLOv3-SPP and YOLOv5l, both using the same dataset and number of training epochs. In this case, we obtained similar results with both versions (around 0.95-0.97 mAP@.5), respectively, but YOLOv5l detected the trash twice as fast as YOLOv3-SPP (35 ms. versus 64 ms.). Therefore, for our task, we can say that YOLOv5l has a better performance than the rest of models. we can also use YOLOv5s instead of YOLOv5l when we need better temporal efficiency versus detection effectiveness.

Once the best YOLO version was chosen, we readjusted the model by training throughout 250 epochs and applying the whole dataset with augmented data in order to improve its performance for detection of trash. Thereby, we accomplished an improvement of between 3.5% and 8.4% detection depending on the used evaluation metric without requiring more time, only 1 ms. more according to our experimentation.

To see if those good results are reflected on the proposed system when it is being tested on a real situation, a true proof was developed, and the qualitative results can be seen in Fig. 6. Note that to better see the made detection process, the information about trash location is tiny to avoid overlap with other bounding boxes and not lose any detail of each image. As it can be seen in Fig. 6 a) and c), the system had a very good performance, detecting almost all or all instances of waste. There were few cases where exists false negatives and some waste is not detected, as the bottle that is on the grass in Fig. 6 b). The non-detection of trash happens commonly only in far waste instances, so there isn't a big problem with that because, as it will be commented later, in the location tests. Moreover, in a very little number of cases, the system detects false positives as it can be seen in Fig. 6 d), and in order to prevent this issue, the confidence threshold for the final system was increased. In summary, the best model performs really well although there are some cases in which appears some problem, as in far o very far waste instances, or very little instances, but they can be reduce or erased, and this performance is the same for all type of conditions and backgrounds, although these cases are not shown in the paper.

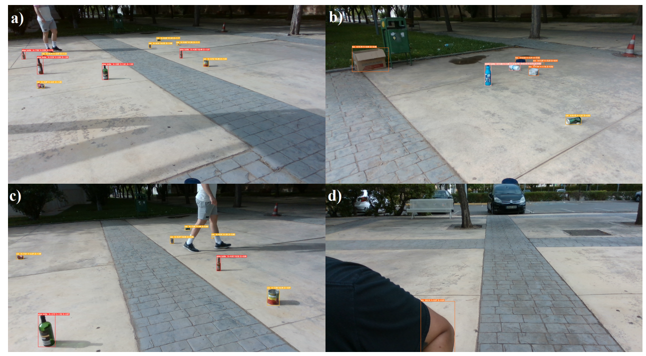

Fig. 6. Qualitative results of detection task with YOLOv5l

Emphasizing location tests, the obtained results were positives because the determined locations were similar to the real ones, existing a little variability around it, especially when the camera was close to the object, around 3 meters or less. However, when the camera was at a greater distance, the estimated locations caused more error and it can also induce false values. This behavior is because the used RGBD camera has a maximum meausuring range around 3 metres, so this is something that has to be assumed because the only possible solution is using a better sensor to obtain depth values. An example of this problem can be seen in Table III and IV which show the location errors that our system estimated for the waste instances (C1, GB1 etc.). The real location was previously shown in Fig. 5. It can be noticed that when the measures are out of the range of 0.3-3 meters from the camera the location error is greater and its variation more pronounced.

An undesirable case is when the robot is quickly turning itself, fact that causes worst results because the global location of the LiDAR and the camera have different publication frequencies. In these cases, it is possible that the location values for one image have to be calculated with a previous global location value (the difference is only of tenths of seconds) and this fact causes little differences between the true location value and the obtained one by the system. This error can be limited if the LiDAR and the camera work at the same frequency, but this is a bit difficult due to the hardware restrictions and by the fact that the LiDAR measurements need to be processed after be received, to be able to publish them with ROS. In order to mitigate the cause of this problem, we added a synchronization control on the system code, to make that the global location and depth values are taken from the closest possible instants, reducing the impact of the difference of publication of readings from both sensors.

In summary, the location task was passed thanks to the use of a synchronization control for the ROS messages and for the fact of taking in care only the location values of waste instances that were in the range of 0.3-3 meters. The results show that the system had a good behavior, obtaining values with an average absolute error of 0.14 meters and an average of 0.07 meters deviation. However, the seen outcome can be improved if the global location of the LiDAR increases its precision, per example with an accelerometer, or if the RGBD camera is capable of get good depth measures in a larger range. It also needs to be noticed that our sensory system obtains the coordinate in Z axis, but for this experiment, the values estimated for this dimension don't vary more than 0.02 or 0.03 m, so we decided not to show its information on the experiments and results, although our system obtains it without problems.

TABLE III
TEST FOR THE LOCATION TASK, AVERAGE ERROR AND DEVIATION IN METERS. USING VALUES IN RANGE 0-8 M

Values in range 0-8 m				
ID	Avg. Error X	Deviation X	Avg. Error Y	Deviation Y
GB1	0.19	0.16	0.06	0.08
GB2	0.11	0.21	0.05	0.07
GB3	0.23	0.13	0.30	0.13
C1	0.09	0.13	0.06	0.10
C2	0.26	0.08	0.32	0.17
C3	0.17	0.27	0.13	0.11
C4	0.16	0.12	0.11	0.17
C5	0.22	0.38	0.24	0.17
AVG	0.18	0.19	0.16	0.12

TABLE IV
TEST FOR THE LOCATION TASK, AVERAGE ERROR AND DEVIATION IN METERS. USING ONLY VALUES IN RANGE 0.3-3 M

Values in range 0.3-3 m				
ID	Avg. Error X	Deviation X	Avg. Error Y	Deviation Y
GB1	0.15	0.12	0.04	0.06
GB2	0.06	0.07	0.05	0.07
GB3	0.23	0.13	0.30	0.13
C1	0.09	0.13	0.06	0.07
C2	0.25	0.07	0.25	0.09
C3	0.09	0.02	0.07	0.05
C4	0.13	-	0.00	-
C5	-	-	-	-
AVG	0.14	0.07	0.11	0.06

V. CONCLUSIONS

This paper presents an approach of a detection and location system for waste in outdoor environments. One of the main contributions of the paper is that it was tested in an autonomous robot for garbage collection. The system was tested using different neural models, and creating a dataset of garbage for outdoor environment, obtaining for the best training a mAP@.5 of 0.9951 and a mAP@.95 of 0.8424, and detecting the instances in the image in an mean time of 36 ms. Another of the main approaches of the system is that it provides a 3D coordinates of the objects in the world coordinates to allow a robotic system to schedule the navigation in order to pick up the set of garbage in the environment. Finally, they need to be highlighted that it provides a location precision that has an average of absolute errors of 0.14 m. and an average of standard deviations of 0.07 m. around the true value.

REFERENCES

[1] Eurostat. "Municipal waste statistics," Accessed on: Feb, 2021. [Online]. Available: https://ec.europa.eu/eurostat/statistics-explained/SEPDF/cache/10360.pdf

[2] V. De Gea, S.T. Puente, P. Gil, "Domestic waste detection and grasping points for robotic picking up," IEEE International Conference on Robotics and Automation. Workshop: Emerging paradigms for robotic manipulation: from the lab to the productive world, 2021.

[3] J. Monroy de Jesús, A. Reyes Nava, C. Ambriz Polo, E. López González, "Desarrollo de un algoritmo de aprendizaje basado en deep learning para su implementación en un robot recolector de basura (identificación de objetos)," 2019. [Online]. Available: https://www.ipomex.org.mx/recursos/ipo/files_ipo3/2019/44020/6/65b06350f0126e90d0f2b2bd51029ee6.pdf

[4] S. Bansal, S. Patel, I. Shah, A. Patel, J. Makwana, R. Thakker, "AGDC: Automatic Garbage Detection and Collection," 2019, arXiv:1908.05849

[5] Andrew G. Howard et al. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," 2017. arXiv:1704.04861

[6] L. Jianghai, "An intelligent garbage classifier based on deep learning models," SPIE proc. MIPPR 2019: Remote Sensing Image Processing, Geographic Information Systems, and Other Applications, pp. 341-347, 2020. doi:10.1117/12.2541782

[7] M. Yang, G. Thung, G. "Classification of trash for recyclability status," CS229 Project Report, 2016.

[8] G. Qiang, S. Yuliang, W. Shikai, "Research on deep learning image recognition technology in garbage classification," 2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS), 2021, pp. 92-96, doi: 10.1109/ACCTCS52002.2021.00027.

[9] C.R. Qi, H. Su, K. Mo, L.J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification an Segmentation," 2017. arXiv:1612.00593v2

[10] A. Komarichev, Z. Zhong, J. Hua, J. "A-CNN: Annularly convolutional Neural Networks on Point Clouds," 2019 arXiv:1904.08017v1

[11] I. del Pino, S. Cova, "Presenting BLUE: A robot for localization in unstructured environments." IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC), pp 130-135. 2018, doi:10.1109/ICARSC.2018.8374172

[12] R. Redmon, A. Farhadi, ”YOLOv3: An incremental Improvement,” 2018. arXiv:1804.02767v1

[13] A. Bochkovskiy, C.Y. Wang, H.Y.M. Liao, “Yolov4: Optimal speed and accuracy of object detection,”. 2020 arXiv:2004.10934

[14] J. Glenn, C. Liu, H. Adam, L. Yn, changyu98, P. Rai, T. Sullian, ”Ultralytics/yolov5:Initial Release (Version v1.0)”, Zenodo. Available online: http://doi.org/10.5281/zenodo.3908560.

[15] J. Redmon, A. Farhadi, “Yolo9000: better, faster, stronger,” in IEEE Conf. on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.

[16] C. Shorten, T.M. Khoshgoftaar, ”A survey on Image Data Augmentation for Deep Learning,”. J Big Data 6, 60, 2019. doi:10.1186/s40537-019-0197-0

[17] H. Wang, C. Wang, C. Chen, L. Kie, ”F-LOAM : Fast LiDAR Odometry and Mapping,” IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020. arXiv:2107.00822

[18] W. He, Z. Huang, Z. Wei, C. Li, B. Guo, ”TF-YOLO: An Improved Incremental Network for Real-Time Object Detection,”. Applied Sciences. 2019; 9(16):3225. https://doi.org/10.3390/app9163225

[19] H. Zhanchao, W. Jianlin, ”DC-SPP-YOLO: Dense Connection and Spatial Pyramid Pooling Based YOLO for Object Detection,” 2019, arxiv:1903.08589