Cleaning Robot - cleaning the world with the support of a mobile artificial intelligence platform

Ahmad Shehada
Gdansk University of Technology,
Faculty of Electronics,
Telecommunication, and Informatics
Email: s197063@student.pg.edu.pl

Nirav Vaghasiya
Gdansk University of Technology,
Faculty of Electronics,
Telecommunication, and Informatics
Email: s196738@student.pg.edu.pl

Murad Abdullayev
Gdansk University of Technology,
Faculty of Electronics,
Telecommunication, and Informatics
Email: s196353@student.pg.edu.pl

Johan Carcamo
Gdansk University of Technology,
Faculty of Electronics,
Telecommunication, and Informatics
Email: s196763@student.pg.edu.pl

Arda Candaş
Gdansk University of Technology,
Faculty of Electronics,
Telecommunication, and Informatics
Email: s196696@student.pg.edu.pl

Andrii Melnyk
Gdansk University of Technology,
Faculty of Electronics,
Telecommunication, and Informatics
Email: s196900@student.pg.edu.pl

***Abstract*—Cleaning Robot is an autonomous vehicle programmed with Nvidia Jetson Nano for its precise arm movement which picks the waste/garbage when detected by the camera with its Computer vision capability for Object detection, YOLO(You Only Look Once), one of the popular frameworks enabling efficient detection of an object. The design of our robot operates on a singular-object focus, identifying waste and providing real-time distance and direction information. The design of our robot is such that it navigates towards detected waste, stopping at a certain distance before collecting and disposing the thrash into a bin.**

I. Introduction

In this ever evolving world, think about a friend that exists in our cities to tackle a common issue which we face in the urban world - waste. The Cleaning Robot is not just a mechanical machine, but serves as a valuable companion in our streets, in our cities among us aiming to make our surrounding cleaner and more sustainable.

In this research paper, we will tell you our motivation for creating this Robot which is designed to adapt to the challenges of urban life and one of the current environmental problems, that is, waste management or waste collection. Imagine it as the most active entity with camera as eyes and robotic arm as our helping hand with artificial intelligence in our neighbourhood not only collecting waste, but also recognizing and separating it into recyclable and non-recyclable categories, such as metals, plastic, glass, and paper wastes, lending a hand in our collective effort to keep our environment clean and green.

With urbanization, our traditional waste management burden also increases and hence our Garbage Collector Robot comes into play with its ability to navigate and detect garbage using eyes(its camera) for collection and to think and learn like a brain.

As we said before, our Cleaning Robot is not just a machine but another way of approaching the environmental problems we face together. We are introducing it to play a vital role in society and with a bigger picture as in our near future our cities will not only be efficient but also conscious environmentally, preserving the balance of our ecosystems.

II. Components

A. MOBOT-EXPLORER-A1: Platform Movement

We control platform movement through UART communication with platform controller ATmega128. We managed to build ROS for ARM and integrate UART communication.

B. MYCOBOT: Robotic Arm

Robotic Arm, which will pickup the garbage once detected using the gripper.

C. NVIDIA Jetson Nano System-on-Module Maxwell GPU + ARM Cortex-A57 + 4GB LPDDR4 + 16GB eMMC

We use Jetson nano for running the object detection and sending signal to the Robotic arm to work on collecting waste, once there is garbage detected.

D. The Intel® RealSense™ D435i

With its wide range view and with depth technology for measuring distance and direction of the object, it is preferred for Computer Vision applications.

E. Power bank Mi 50W Power Bank 20000mAh

We use power bank to supply and energize Nvidia nano jetson module and Raspberry pi 4.

F. BATTERY 6V 3.4Ah MW POWER MW 3.4-6

Two rechargeable batteries are used to power robotic arm and platform.

III. System Architecture

The system architecture of the robot is based on integration of Raspberry pi 4 and Nvidia jetson nano module. Raspberry pi 4 acts as a communicator which facilitates and links the Nvidia jetson nano module responsible for object detection, movement of the robot and platforms arm for movement and picking up trash to the trash bin. Raspberry pi 4 sort of fills the gap, making it as bridge for communication. Raspberry pi 4 ensures every working components is running smoothly. So, When an object is detected, the information is sent to the Mobile robot via Raspberry pi 4 to make the robot move towards the detected object to the certain distance for collecting the garbage and an actionable command is sent to the platforms arm to pick the the trash, putting it into the bin and returning to its rest position.

IV. Hardware Setup and Fabrication

A. Raspberry Pi 4 and Nvidia Nano Setup

Nvidia jetson nano is set up for object detection by flashing the Nvidia jetson nano developer kit SD card image, installing SDK and installing dependencies for object detection and connecting IntelRealsense Camera to it. On the Raspberry Pi 4, Raspberry OS and necessary communication libraries are installed. A script for communication is developed on Raspberry Pi 4 to fetch object detection results which is then transmitted or sent to arm and platform for its movement via command/program. Both Raspberry Pi 4 and Nvidia jetson nano are connected to the same network which facilitates the communication and a real-time system of object detection is implemented for sending and receiving of the data including the distance and direction of the object. For example, When a specific object is detected by Nvidia jetson nano, it sends a message to Raspberry pi 4 indicating the type and location of the object. The Raspberry Pi 4 interprets the information and trigger commands and sends them to arm and platform for the movement i.e. moving towards the detected object, picking up the waste and throwing into the trash bin.

B. Trash Bin Fabrication

3D view of our trash can which is fixed on the side of MOBOT-Explorer A1:

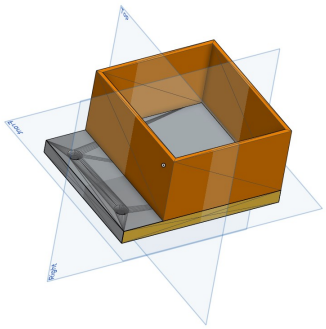

Fig. 1: 3D view of trashcan

V. Object Detection

A. Model Training

For the project, we used YOLOv5s model. The ”s” stands for small for faster inference and considering the suitability for the Nvidia Jetson Nano module which we used for object detection. We trained our model with 9799 images with a single class ”Garbage” split accordingly as shown in the table:

TABLE I: Data Split

Dataset	Number of Images	Percentage
Train	6695	70%
Validation	2072	20%
Test	1032	10%

With the following parameters:

TABLE II: Training Configuration

Parameter	Value
Image Size	640
Batch Size	32
Epochs	1000
Model Weights	yolov5s
Initial Training Rate	0.01
Final Training Rate	0.001
Optimizer	Stochastic Gradient Descent (SGD)

We achieved the following results with our training:

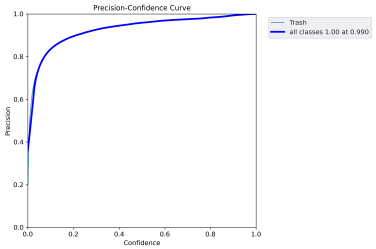

Fig. 2: Precision-confidence curve

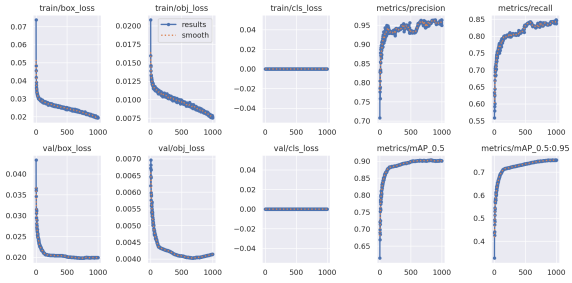

Fig. 3: Accuracy and Recall

Key points from the graph:

- The graph shows various loss and performance metrics for both training and validation data.

- Loss metrics (box_loss, obj_loss, cls_loss) generally decrease over time, indicating model improvement.
- Performance metrics (precision, recall, mAP) generally increase over time, also indicating improvement.

B. Camera SDK and Remote Connection

A camera SDK is built that streamlines interaction of camera enabling both depth and RGB vision with suitable resolution. In this case, the least resolution is taken for quick execution to make sure the flow of video data is smooth for real-time processing.

C. Object Parameters

Listing 1: Distance and Direction

```
x = int((xyxy[0] + xyxy[2])/2)
y = int((xyxy[1] + xyxy[3])/2)
if x != [] or y != []:
    dist = depth_frame.get_distance(x, y)*100
    Xtarget = dist*(x  - intr.ppx)/intr.fx
    lines.append(Xtarget)
    Ytarget = dist*(y  - intr.ppy)/intr.fy
    lines.append(Ytarget)
    Ztarget = dist
    lines.append(Ztarget)
    hfov = 2*degrees(atan(424/(2*intr.fx)))
    theta = (((x-intr.ppx)/(424))
             *(hfov*((424-intr.ppx)/intr.ppx)))
    lines.append(theta)
```

The provided code (see Listing 1) is about extracting information about the position of the object and the direction from the center of the camera. x and y are the average of x and y coordinates calculated from the list xyxy containing four values representing bounding box coordinates around the object. Then, a condition is checked, whether x or y has an value, which is always going to be true. To obtain the distance of the object at the x and y coordinates, `get_distance` on the `depth_frame` is used. It is multiplied by 100 to get in centimeters instead of meters. Xtarget, Ytarget and Ztarget are the 3-Dimensional space calculated cameras parameter to represent 3D coordinates of the object in the cameras coordinates system which is then appended to the list called lines and for the direction, horizontal field of view(hfov) of the camera is calculated using 'atan' i.e. arc tangent function. Then, the angle 'theta' is calculated based on the x coordinates and parameters of the camera to give the direction of the object.

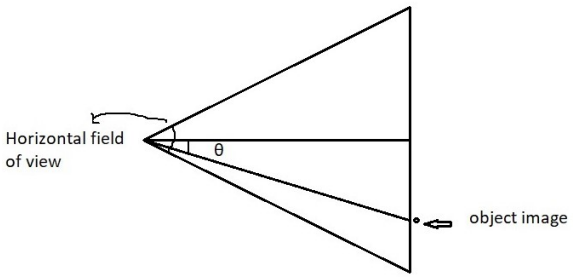

Fig. 4: Horizontal field of view

As shown in Figure 4 , the vertical line is the whole field of view i.e. the image width which in this case is 424px.

$$\theta = \frac{\frac{x-\text{imagewidth}}{2}}{424} * Horizontal field of view$$

where,

$$\frac{\frac{x-\text{imagewidth}}{2}}{424}$$

is the distance of the object image from the center of the frame.

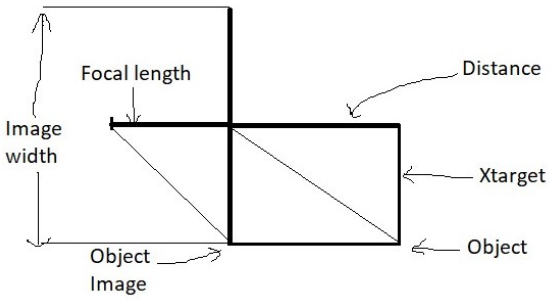

Fig. 5: Xtarget and Ytarget

As shown in Figure 5, we have two identical triangles. So,

$$\frac{\text{Xtarget}}{\text{distance}} = \frac{x - \text{imagewidth}}{\text{focal length}}$$

and Xtarget can be calculated by,

$$\text{Xtarget} = (x - \text{imagewidth}) \cdot \left(\frac{\text{distance}}{\text{focal length}}\right)$$

To calculate Ytarget is similar. the only difference is, we use image height and y coordinate.

VI. Platform and Arm Control

A. Platform Micro-controller Program

Listing 2: Platform movement

```

import socket
import serial
from time import sleep

command_list_forward_slow = [
    b'x',
```

```
    b'x',
    b'v -120     ',
    b'b'
]
command_list_forward_med = [
    b'x',
    b'x',
    b'v -135     ',
    b'b'
]
...
...
...
```

The provided code (see Listing 2) defines the list of commands for controlling platform movement. The commands are categorized into forward movement, backward movement, and turning. The command structure includes start, indicating a movement mode, direction commands(forward, right, left) , end/stop commands represented by `b'x'`, `b'm'`, `b'w'`, `b'd'`, `b'a'`, `b'b'` respectively with speed values `(v -120, v -135, v -150, v -255, v 140, v 150, v 185, v 250)`.

B. Arm Movement

Listing 3: Arm movement

```
def arm_move(theta):
    print(theta)
    command_list = [
        b'power on\r\n',
        b'get_position is\r\n',
        b'gripper_set state 1 100 \r\n',
        b'set_angles_sync -90 0 0 0 0 0 20
            20\r\n',
        b'set_angles_sync 0 0 0 0 0 0 20
            20\r\n',
        b'gripper_set state 0 50 \r\n',
        b'set_angles_sync %d -90 -20 35 0 50
            20 20\r\n' % (theta + 15),
        b'power is\r\n',
        b'gripper_set state 1 50 \r\n',
        b'set_angles_sync %d -10 0 -75 0 45
            20 22\r\n' % (theta + 15),
        b'set_angles_sync 140 0 0 -75 -10 45
            20 20\r\n',
        b'gripper_set state 0 100 \r\n',
        b'power is\r\n',
        b'power is\r\n',
        b'gripper_set state 1 100 \r\n',
        b'set_angles_sync -40 0 0 0 0 0 20
            20\r\n',
        b'power is\r\n',
        b'power is\r\n',
        b'gripper_status \r\n',
        b'gripper_set state 1 100 \r\n',
        b'set_angles_sync -70 -130 155 -125
            90 20 20 20\r\n',
        b'power is\r\n',
        b'power is\r\n',
        b'power is\r\n',
    ]
```

The provided code (see Listing 3) defines the synchronized angles of each joint of the arm for its movement for grabbing the object with gripper and its power status. It also takes the parameter "theta" from Raspberry Pi 4 i.e. the direction of the the object detected. 15 is added to the theta for the calibration.

Command	Description
`b'power on\r\n'`	Turn on power.
`b'get_position is\r\n'`	current position.
`b'gripper_set state 1 100 \r\n'`	gripper state to 1 (open) with a force of 100.
`b'set_angles_sync -90 0 0 0 0 0 20 20\r\n'`	angles for the joints.
`b'set_angles_sync %d -90 -20 35 0 50 20 20\r\n' % (theta + 15)`	dynamically calculated angle for joint 1 using theta
`b'gripper_status \r\n'`	Gripper status.
`b'power is\r\n'`	Power status.

TABLE III: Robotic Arm Commands

VII. Problem Solving and Collaboration

One of the biggest challenge faced during the project was operating the cleaning robot on batteries. It could only run for shorter duration when fully operated on batteries. For now, we are using batteries to power robotic arm and platform movement one at a time meaning once the platform has been moved towards the object at a certain distance, it stops and then robotic arm uses the battery for its movement of picking up trash.

VIII. Cleaning Robot

Here is the end result of the Cleaning Robot.

Fig. 6: Cleaning Robot.

IX. Conclusion

In conclusion, We have achieved following milestone with Cleaning robot:

- It can navigate efficiently with good accuracy of object detection and pickup waste successfully.

- It is entirely mobile and can be control or commands can be given remotely.
- It currently handles and detect one object at a time, focusing on systematic approach for cleaning/gathering trash.

However, There is room for improvements and future scope:

- As mentioned earlier, Battery issue needs to be fixed to enhance operational capabilities of robot, ensuring sustained performance without draining much power during platform movement and arm movement and carrying it out smoothly.
- Detection results can be further used to analyze type of waste in the neighbourhood differentiating recyclable and Non-recyclable waste for more cleaning efforts.
- Its navigation capabilities can be improved for diverse terrain, increasing its overall versatility and adapting to surroundings in handling environmental challenges.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude our supervisor for this project, Prof. Dr. Hab. Inż. Jacek Rumiński who supported, encouraged, and appreciated our efforts collectively, guiding us on every step with necessary inputs, providing necessary resources we needed for the completion of this project, whose expertise and mentorship is invaluable. We would like to thank Andrzej Galikowski and Adam Bujnowski for helping us out with required 3D printing we needed to accomplish this project and solving the battery issues we faced during the project. We are so grateful to all of you for your contribution to this project and to the Gdansk University of Technology for providing us with the ideal learning and conductive environment for this study.

REFERENCES

[1] X.-R. Huang, W.-H. Chen, W.-Y. Pai, G.-Z. Huang, W.-C. Hu and L.-B. Chen, "An AI Edge Computing-Based Robotic Automatic Guided Vehicle System for Cleaning Garbage," *2022 IEEE 4th Global Conference on Life Sciences and Technologies (LifeTech)*, Osaka, Japan, 2022, pp. 446-447, doi: 10.1109/LifeTech53646.2022.9754931.

[2] K. Lakshmi, I. k, N. S. Abdullah, and K. R. Monica Bharathi, "Deep Learning based Garbage Detection for Smart City Applications," *2022 6th International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India, 2022, pp. 1464-1468, doi: 10.1109/ICICCS53718.2022.9788161.

[3] J. Bai, S. Lian, Z. Liu, K. Wang, and D. Liu, "Deep Learning Based Robot for Automatically Picking Up Garbage," *IEEE Transactions on Consumer Electronics*, vol. 64, no. 3, pp. 382-389, Aug. 2018, doi: 10.1109/TCE.2018.2859629.

[4] Z. Chang, L. Hao, H. Tan, and W. Li, "Design of mobile garbage collection robot based on visual recognition," *2020 IEEE 3rd International Conference on Automation, Electronics and Electrical Engineering (AUTEEE)*, Shenyang, China, 2020, pp. 448-451, doi: 10.1109/AUTEEE50969.2020.9315545.

[5] H. S. Pandita, V. Vaidya, M. Doifode, P. Bhavthankar, and A. Venkatesh, "Garbage Classification using Machine Learning to Aid Recycling," *2022 3rd International Conference for Emerging Technology (INCET)*, Belgaum, India, 2022, pp. 1-6, doi: 10.1109/INCET54531.2022.9824215.

[6] L. Yu, G. Pan, and M. Li, "Garbage Detection Algorithm Based on Deep Learning," *2021 IEEE 5th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, Xi'an, China, 2021, pp. 469-473, doi: 10.1109/ITNEC52019.2021.9586894.

[7] H. Hu, X. Jiang, X. Liu, R. Ding, S. Ma, and B. Wang, "Summary of Domestic Garbage Classification and Detection Based on Deep Learning," *2021 7th International Conference on Computer and Communications (ICCC)*, Chengdu, China, 2021, pp. 858-862, doi: 10.1109/ICCC54389.2021.9674502.

[8] B. Xiao, J. Guo, and Z. He, "Real-Time Object Detection Algorithm of Autonomous Vehicles Based on Improved YOLOv5s," *2021 5th CAA International Conference on Vehicular Control and Intelligence (CVCI)*, Tianjin, China, 2021, pp. 1-6, doi: 10.1109/CVCI54083.2021.9661149.

[9] C. Mayorga et al., "GABOT: Garbage Autonomous Collector for Indoors at Low Cost," *2019 International Conference on Mechatronics, Electronics and Automotive Engineering (ICMEAE)*, Cuernavaca, Mexico, 2019, pp. 56-61, doi: 10.1109/ICMEAE.2019.00018.