JASAE
Journal of Applied Science and Advanced Engineering
Vol. 1 No. 2 (2023)
e-ISSN: 2985-7252
https://jasae.org/

Developing an Object Detection and Gripping Mechanism Algorithm using Machine Learning

Hazem Mohammad[1]*, Sameer Kishore[2], and Judhi Prasetyo[3]
[1,2,3]Middlesex University, Dubai, United Arab Emirates
*email: hh777@live.mdx.ac.uk, Hazem7mohammad@gmail.com

ARTICLE HISTORY
Received: 24 August 2023
Revised: 15 September 2023
Accepted: 20 September 2023

KEYWORDS
Object detection
Localizing
YOLO
Gripping
Robot

ABSTRACT– Localizing and recognition of objects are critical problems for indoor manipulation tasks. This paper describes an algorithm based on computer vision and machine learning that does object detection and gripping tasks. Detection of objects is carried out using a combination of a camera and depth sensor using a Kinect v1 depth sensor. Moreover, machine learning algorithms (YOLO) are used for computer vision. The project presents a method that allows the Kinect sensor to detect objects' 3D location. At the same time, it is attached to any robotic arm base, allowing for a more versatile and compact solution to be used in stationary places using industrial robot arms or mobile robots. The results show an error of locating an object to be 5 mm. and more than 70% confidence in detecting objects correctly. There are many possibilities in which this project can be used, such as in industrial fields, to sort, load, and unload different kinds of objects based on their type, size, and shape. In agriculture fields, to collect or sort different kinds of fruits, in kitchens and cafes where sorting objects like cups, bottles, and cans can occur. Also, this project can be added to mobile robots to do indoor human services or collect trash from different places.

Introduction

A. Background

Some early scholars used manual grasping systems for object grasping, having the robot go to a predefined location to pick an object and then to another predefined location to place the object; these methods are straightforward, simple, and suitable for fixed scene operation tasks. However, external factors can easily affect these systems or methods because they do not visually detect the object's color, size, or shape, and they only follow preprogrammed coordinates.

Machine vision has changed the game for industrial robots. The demand for intelligent autonomous robots has increased, and robots that can be used in complex scenes that are not easily affected by external factors are very much needed.

For object-grabbing robots, there are three steps to get the task done: (A) detecting the object, (B) estimating the coordinates of the object, and (C) grabbing the object. The ability to manipulate and grasp a desired object for picking or placing is a challenging problem, and it requires an accurate calculation of the object's position relative to the arm and an accurate kinematics model to achieve the task precisely.

The problem for visually guided robots is that a camera detects the objects' location, shape, and dimensions, as shown in Figure 1. In Figure 1, the size and the distance of the object are entirely dependent on the area calculations of the objects being viewed from the top as a 2D image.

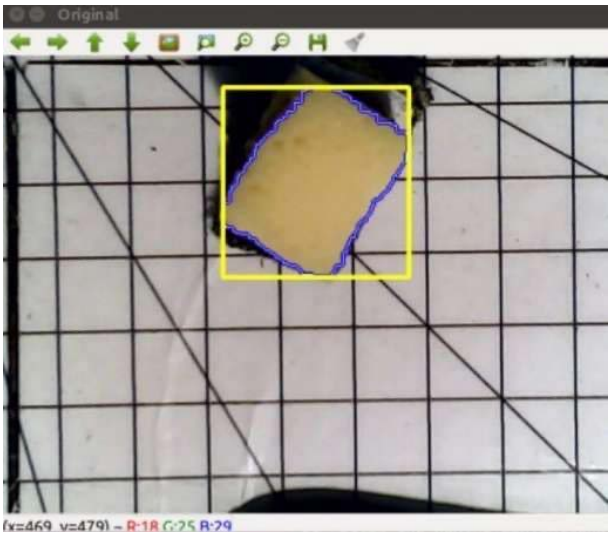

Figure 1. Top view of workspace showing object's 2D location

The position of the camera sensor is crucial in such robots. Having the camera attached to a fixed top view of the workspace area separated from the robotic arm, like in Figure (2), can limit the flexibility as well as the accuracy of the robot because of the need for having a specific fixed setup or workspace for the robot to work, mostly in 2D. Also, the robotic arm can block the view of the camera while moving.

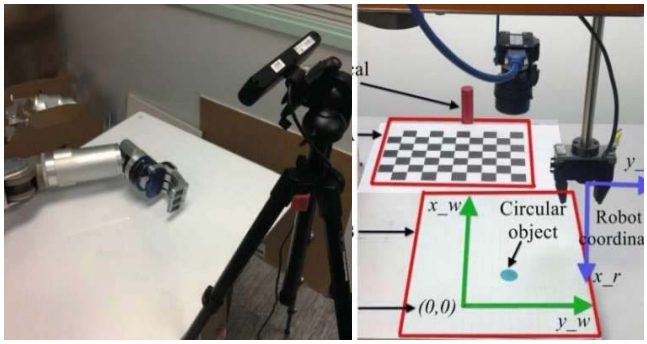

Figure (2): Top view camera for workspace area [5,11]

There are many applications where a top view of the 2D camera is practical and can work well, like picking good fruits from damaged fruits or sorting color-coded boxes or objects. However, the detection is only based on color, which can put the robot at risk of false detection because there is no accurate recognition of the type of object being manipulated. Also, a 2D camera can be limited in some scenarios where a depth camera is needed. It can also be difficult for a 2D camera attached to the base of a robotic arm viewing a front field of view to detect the size and location of the objects in front of it if it is entirely dependent on the area calculation of the object detected because the distance can vary, Figure 3.

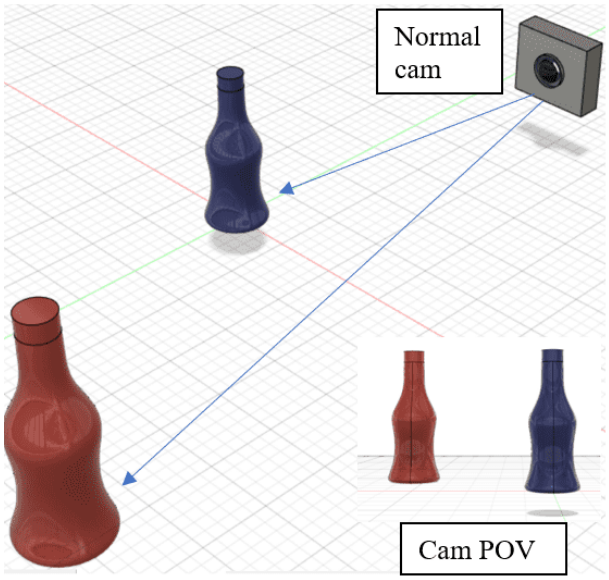

Figure 3. The 2D camera can detect two objects' distances and sizes falsely if only it depends on the area calculation for the object's distance.

As shown in Figure 3, the two objects are of two different sizes and locations, where from the camera's point of view, they are in the exact same location and distance and are the same in size. In Figure 3, the camera is assumed to be attached to the front of the point of view of the base of a robotic arm and not the top view of a workspace.

For grasping an object, it is required to recognize and locate the object in 3D space, which can be difficult for 2D image-based detection to fulfill. Therefore, the 3D point cloud-based target detection approach is more appropriate.

The Kinect vision sensor by Microsoft [13] is low-cost and used for robotic systems research for its ability to provide 2D and depth sensor images using point clouds.

This project has an objective of whether knowing what shape and size the object is, as well as the actual location of that object, can help tremendously in accurately grasping it more easily. So, with a depth sensor, it can be easier to find out the size and location of an object relative to the sensor more accurately for the pick-and-drop task to take place.

The research of this paper aims to design an algorithm that has a perception ability capable of detecting and recognizing objects as well as locating objects in a 3D environment and then grasping them. Grabbing objects using a robotic arm based on 3D coordinates estimation is attempted with the help of the Kinect v1 camera [13]. In order to know how accurate and effective this method can be in locating and grasping objects.

B. Related Research

Ten papers or projects have been explored to highlight the works on locating objects in 3D space using a depth sensor. These papers were published from the year (2015 to 2022) in addition to one paper published in 2012 [5]. All the papers share the need to use a Microsoft Kinect depth camera sensor in order to be able to make use of laser data as well as camera data.

In this project, the goal is to develop a vision-based object detection and a gripping mechanism by estimating the 3D location of an object and grasping it using a robotic arm to place it in a different location. The first five projects [1-5] share almost the same goal: having the robot locate objects in 3D space and then do the relocating or repositioning tasks. In contrast, the second half of the papers [6-10] focus on other goals, purposes, and applications. Additionally, the approach of this project is to have the depth sensor connected to the base of the robotic arm in a way that shows a front field of view for the objects around it (the reason for this approach is explained in the introduction section of this report), this approach found to be followed in papers [1-2]. Furthermore, papers [1 and 3] use the YOLO method and algorithms for object recognition and detection. This method has proved its effectiveness and is explored for this project to be used for object recognition.

The project developed in this paper is mainly similar to the "Object Detection and Grabbing Based on Machine Vision for Service Robot" paper [1]. The similarities are the approach where object grabbing is based on estimating the 3D coordinates in which an object is located, the function of grasping the object and relocating it to a different location, and having the depth sensor attached in a way where it can view objects from the front at the base of the robotic arm as in Figure (4), not from the side or the top and not set up in a way where it is not attached to the robotic arm completely, Figure (2), and not attached to the end effector.

The differences are the following: in paper [1], the object detection is based on a deep convolutional neural network and depth-first algorithm to achieve object grabbing. The sensor being used is a Kinect v2 camera from Microsoft [16], and a six-degree-of-freedom robotic arm to grab objects is used.

However, this project uses Darknet [19] for ROS combined with YOLO to match the camera and the depth sensor (Kinect v1) readings for recognition and position detection.

Also, in this project, the challenges attempted are to create a robot that can do the tasks that an RGB camera color detection powered robot can do and do more using a robust object recognition method and a depth sensor to fulfill the needs and demands of other applications. In addition to a compact solution consisting of the sensor and any robotic arm that is easy to calibrate, that can be attached to stationary or mobile robots. It starts with accurately locating bottles and cups and adding more objects in the future. The challenges faced and solved in the paper [1] were first to match the color and depth received by the sensor to restore the object edge information. The second is to calibrate the transformation matrices of the world-coordinate system, the robot coordinate system, and the target coordinate system for the grabbing task. The third challenge that was solved was to match the bounding box while detecting objects to the edges of the objects. To make the grasping task more efficient. That was solved using an object segmentation method based on depth information using an RGB-D camera to ensure target objects are entirely presented in the region of interest.

The paper [1] showcases the ability to detect objects using RGB based on the highest confidence result. Then, the location of the object is determined using the depth sensor to map the position for the robotic arm to be able to grasp the object. It has been confirmed the effectiveness of the proposed method.

Materials and Methods

A. Scope of work

Keras [17], TensorFlow [18], and Darknet [19] are examples of available deep-learning libraries. In this project, Darknet is used, and YOLOv2 [20,21] is implemented within it.

YOLO is an object detection model used for real-time detections, which is very fast when done on CUDA-capable systems [22], which is helpful for this project and for robotics in general.

For 2D and 3D object detection and location with YOLO, darknet_ros, in conjunction with point-cloud data, used by Darknet_ROS_3D [23], which was created by Francisco Martin [24], to match the 3D space to the detections and then locate where the detected object is in a 3D space.

The object's location is then calculated for a transformation matrix, starting with the Kinect sensor and ending with the end effector for a robotic arm to do the gripping task.

This attempt is approached by building a structure consisting of a Kinect depth camera sensor attached to a robotic arm's base, in which this layout, Figure 4, can be used or added on top of a stationary object such as a desk or top of a customized mobile robot such as a rover (Figure 5), for future projects.

As seen in Figure 5, this structure adds more flexibility, mobility, and simplicity to the design to attach the robot to many setups and workspaces. This will eliminate the need to set up the robot in a specific stationary method, where the camera needs to be placed on the top of a workspace to calculate the position of the objects from a top field of view,

Figure 4. Approached setup

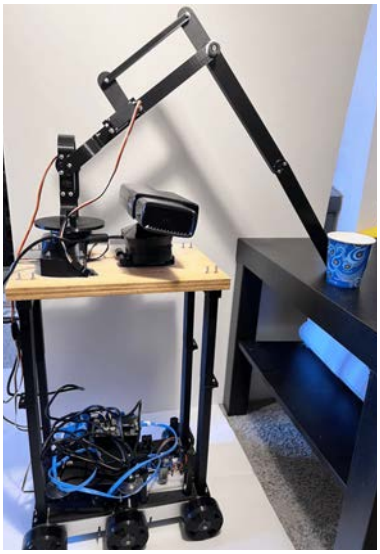

Figure 5. Future potential application.

as shown in Figure 1 and Figure 2. Additionally, not attaching the camera to the arm's end effector is applied to eliminate extra weight and to have a better stable field of view without shaking or any vibration.

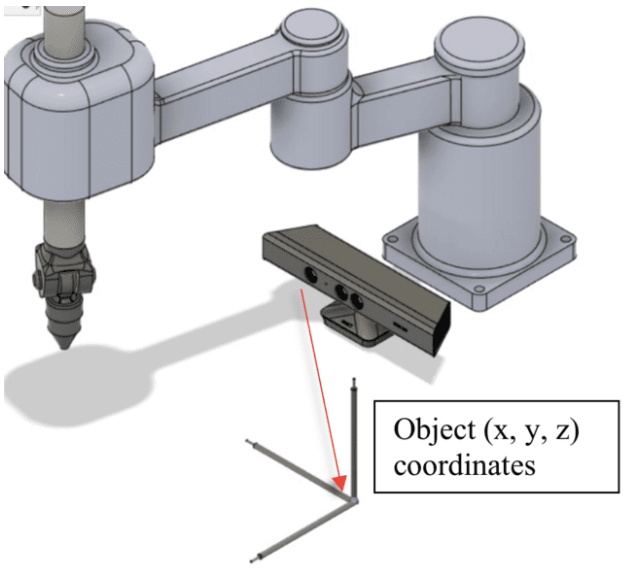

Figure 6. 3D location of the object in the environment.

This solution can increase the accuracy of locating and finding the transformation of each object relative to the camera link, Figure 6. The center point of each object from the point cloud is found and provided by the Kinect's depth sensor.

B. Implementation

The implementation has three main objectives:

1. Algorithm to estimate 3D position (using depth sensor)

This project uses a version of YOLO adapted for ROS; the project starts by creating a ROS package on an Ubuntu system called my_object_recognition_pkg, which contains a launch file (Kinect.launch) that covers the 2D and 3D detection.

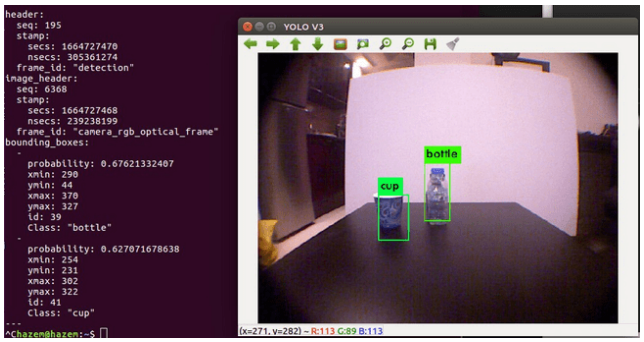

Figure 7. It publishes the name of the object occupying the space detected by the depth sensor.

Another file is yolov2-tiny.yaml: it is fast and efficient for seeing the point clouds in RViz, which notably affects the system load. point_cloud_topic: a topic publishing the point cloud data of the Kinect, which matches the RGB 2D detection and 3D cloud data to generate bounding boxes in 3D. A python file (yolo_3d_data_extraction.py) is developed to do the tasks, as shown in Figure 8.

Figure 8 explains the code flow, which does the following:

Extracts the marker data generated by the darknet_ros_3d.launch from the topic /darknet_ros_3d/bounding_boxes (shown in Figure 7) and then filters the required object, such as a cup or a bottle. Then, it stores multiple objects of the same type (cup) and calculates the center of the bounding box based on the max and min values. The center is presented as the object's coordinates of (x, y, z) relative to the Kinect sensor. The coordinates are then sent to the robotic arm for the gripping task.

2. Grabbing objects (EPSON robotic arm).

EPSON [25] A Robotic arm was used in this project to achieve precise and reliable picking and dropping tasks and focus on the algorithm and not on the kinematics of the arm. This arm is easy to calibrate with the Kinect sensor for the operation to start.

A Python file is made to receive the data of the coordinates and send it to the IP address of the robotic arm so it can go to the required position and start the pick-and-place operation. In the future, any robotic arm can be calibrated and used to achieve the goal of this project.

3. Measuring the effectiveness.

Testing must be done to measure the accuracy of the detection. Many factors play a role in the detection performance, such as external factors like lighting and internal code parameters.

Figure 8. Showing the algorithm's overall flow.

The following is a quantification of the accuracy and performance when the factors are changing. Factors affecting the detection are divided into external and internal as follows:

A. External factors:

(i) Height of camera

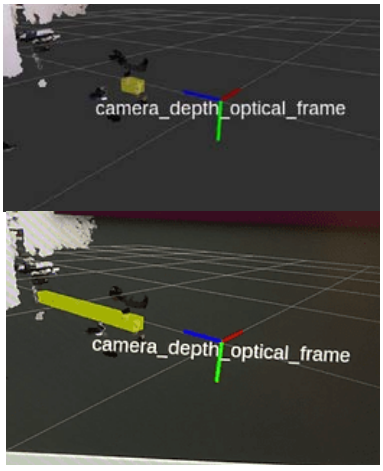

Figure 9. Difference of sensor height relative to the object.

Figure (9) demonstrates the difference in the height of the camera. When the sensor directly faces the object on the lower image, the bounding box reading is incorrect (represented by the yellow box). The bounding box reading is correct when the sensor is higher than the object it detects (as seen from the upper image).

Figure 10. Showing testing different heights to pick the optimal one.

(ii) Tilt of camera: Different results are given for the height measurements of an object relative to the sensor, as the object's height measurement is less as the object moves far from the sensor. In addition, the tilt angle degree can improve the view for better object recognition rather than a side view of an object.

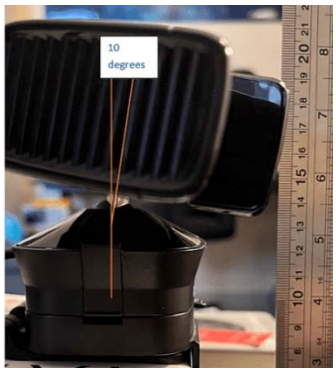

Figure 11. Showing testing different angles to pick the optimal one.

(iii) Background of the field of view: White and noisy background were tested (different kinds of objects or colors).
(iv) Lighting: Different lighting conditions were examined.
(v) The reach-ability of the arm: is 55 cm for this project.
(vi) Zoom lens:

One of the limitations of the depth sensor is that it can detect and form objects' bounding boxes only beyond 40 cm from its lens. It can be chosen to attach a zoom lens to the sensor.

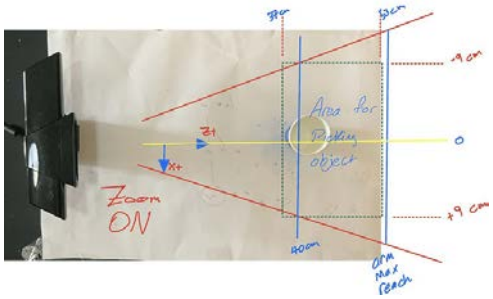

Figure 12. When attaching a zoom lens, the sensor can detect and form a bounding box in the green area shown. From 37 cm to 53 (arm max reachable distance) in the Z direction, and from -9 cm to 9 cm in the X direction.

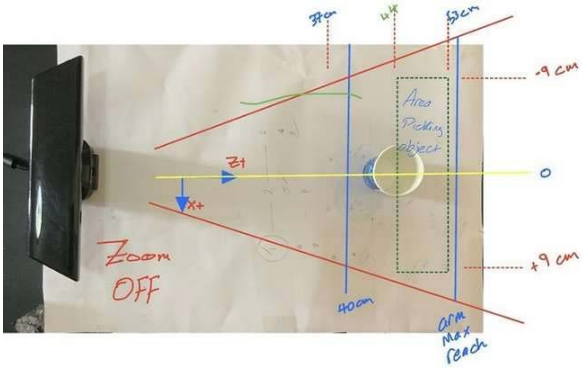

Figure 13. When detaching a zoom lens, the green area is much smaller.

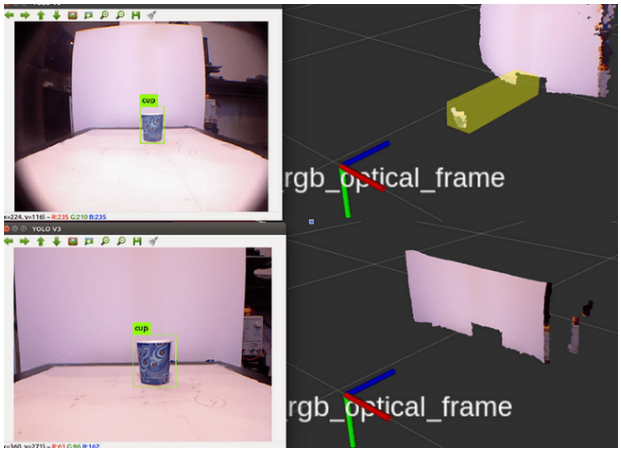

Figure 14. Attaching/detaching the zoom lens

When the object is at almost 36 cm distance from the sensor, and the zoom lens is on, like the top image, the sensor picks up the object, and the bounding box is generated. When the zoom lens is removed, as shown in the bottom image, the LiDar sensor does not see the object, and no 3D bounding box is formed or generated. In this case, the object must be farther than 44 cm from the sensor to be detected, making the pick-up area smaller, as shown in Figure 14, and requiring a giant robotic arm to reach the object.

Figure 15. Zoom lens difference. The image or view of the camera without the zoom lens is much clearer and with much less distortion (left picture) than when the zoom lens is on the camera (right picture).

It is more important to have the robot close to the object than to have less distortion, and that is why, in most of the iterations, the choice was made to have the zoom lens on the camera.

B. Internal factors:

(i) System specifications: The processing power of the computer system is limited to the following specifications (Figure 16).

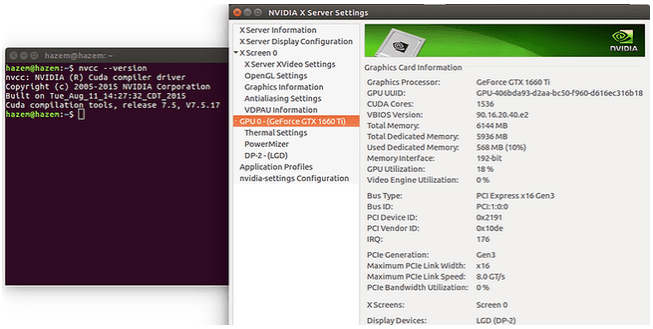

Figure 16. CUDA specifications and GPU.

(ii) YAML files parameters.
(iii) The number of objects (classes) required to detect form the library
(iv) The frame rate of the camera
(v) Minimum probability for darknet_3d.yaml: 0.3 by default
(vi) Minimum detection threshold for darknet_3d.yaml: 0.3 by default
(vii) Minimum detection threshold for yolov2_tiny.yaml: 0.3 by default

Testing the detection task:

1. The reach-ability of the arm: not changeable for this project
2. The frame rate of the camera: not changeable for this project
3. Zoom lens
4. Height of camera
5. Tilt of camera
6. Background of the field of view
7. Lighting
8. The number of objects (classes) to detect form the library
9. Minimum probability for darknet_3d.yaml
10. Minimum detection threshold for darknet_3d.yaml
11. Minimum detection threshold for yolov2_tiny.yaml

Results and Discussion

For the results to be considered, a valid detection some metrics needed to be set. A score from (0) to (3), and (3) being the best score was developed.

Table 1. Example of different results of different iterations.

	Iteration 1	*Iteration 2*
If the **2D** *bounding box is generated*	*3*	*3*
If the **3D** *bounding box is generated*	*1*	*3*
accuracy of detecting the types of bottles and cups correctly	*1*	*3*
Percentage of cup confidence	*2*	*2*
Percentage of Bottle confidence	*0*	*3*
How quick to detect cup or bottle (Time)	*3*	*3*
Total score	*1.6*	*2.83*

The following is an explanation of the score system:

1. If the 2D bounding box is generated:
2. If the 3D bounding box is generated:
3. Accuracy of detecting the types of bottles and cups correctly

   For all the above, if:
   Most of the time score (3),
   Half the time score (2),
   Small amount of time (1),
   Not at all (0)

4. Percentage of cup confidence:
5. Percentage of Bottle confidence:
   Percentage is more than 80 score (3),
   Percentage (60-80) score (2),

The following is an explanation of the score system:

1. If the 2D bounding box is generated:
2. If the 3D bounding box is generated:
3. Accuracy of detecting the types of bottles and cups correctly

   For all the above, if:
   Most of the time score (3),
   Half the time score (2),
   Small amount of time (1),
   Not at all (0)

4. Percentage of cup confidence:
5. Percentage of Bottle confidence:
   Percentage is more than 80 score (3),
   Percentage (60-80) score (2),
   Less than (60) score (1),
   Not at all (0)
6. Time to detect cup or bottle:
   Every 1 second score (3),
   Between (2-5) score (2),
   More than 6 seconds (1),
   System hangs (0)

Based on the system created:

If score > 2.6 = the detection is valid. The confidence of detecting an object reached more 90% and the error of picking up objects was 5 mm.

This project focused on detecting cups and bottles and adding more objects in the future, as shown in Figures 17 and 18.

Many iterations for detection were developed, and the best iterations were picked based on the best performance. The project worked as expected, and many developments will be done in the future, such as adding more objects to the detection classes and attaching the robot to a mobile robot for SLAM applications.

The system detects and recognizes objects such as cups and bottles with almost 80% accuracy. Objects' coordinates in 3D space are extracted without any problems. The robotic arm can do the gripping task smoothly with an error of 5mm.

Figure 17. Sorting robot for kitchens and cafes, dish loading and recycling plastic and cans.

Figure 18. Industrial fields using EPSON robotic arm, repetitive sorting of different kinds of objects, loading and unloading heavy objects.

Proof of concept design for the future applications:

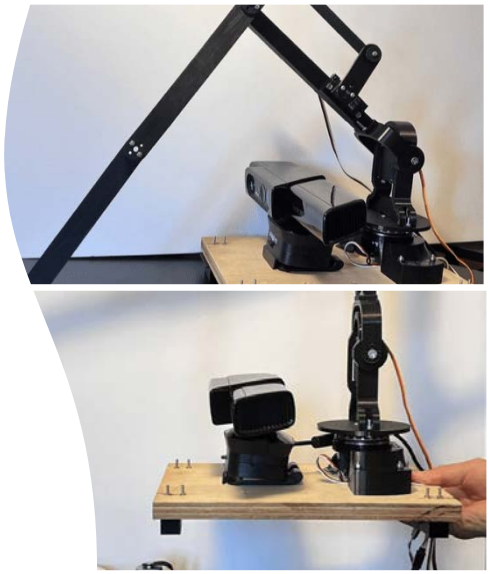

Figure 19. Compact removable standalone solution, can be attached to mobile robots for more dynamic applications

Advantages over projects with color detection RGB camera sensor:

- Compact design
- More dynamic solution for different applications and situations
- Can be used with different manipulators
- True 3D location and detection

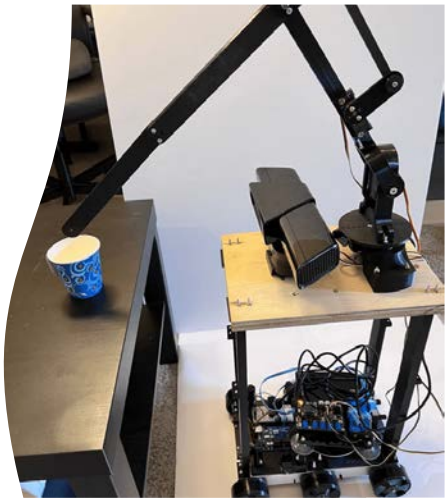

Figure 20. Mobile robot design advantages

Conclusion

The goal of this project was achieved; the goal was to build a compact solution consisting of a depth sensor and any robotic arm that is not only capable of doing the tasks that are being done using an RGB camera but also able to do more tasks like being fixed to a stationary base or on top of a mobile robot, making the robot more capable of doing more dynamic tasks in different situations and applications.

References

[1] Zhang G, Jia S, Zeng D, Zheng Z. Object detection and grabbing based on machine vision for Service Robot. 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON).2018; https://doi.org/10.1109/iemcon.2018.8615062.

[2] Jia S, Ju Z, Xu T, Zhang H, Li X. Mechanical arm grasping and target recognition strategy based on the coarse-to-fine algorithm. 2016 IEEE International Conference on Information and Automation (ICIA), 2016;1812-1817. https://doi.org/10.1109/ICInfA.2016.7832112.

[3] Han M, Wang P, Wang X, Li X. Mobile Grabbing Robot with Target Detection Using Quadric Fitting. 2021 IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA). 2021; 199-200, https://doi.org/10.1109/ICTA53157.2021.9661895.

[4] Ouyang W, Huang W, Min H. Robot Grasp with Multi-object Detection based on RGB-D Image. 2021 China Automation Congress (CAC). 2021;6543-6548. https://doi.org/10.1109/CAC53003.2021.9728678.

[5] Chi W, Meng MQ, Chen X. Robot aided object segmentation based on kinect without prior knowledge. 2012 IEEE International Conference on Robotics and Biomimetics (ROBIO). 2012;1784-1788. https://doi.org/10.1109/ROBIO.2012.6491226.

[6] Ali MM, Liu H, Stoll R, Thurow K. Arm grasping for mobile robot transportation using Kinect sensor and kinematic analysis. 2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings 2015; 516-521. https://doi.org/10.1109/I2MTC.2015.7151321.

[7] Ali MM, Liu H, Stoll N, Thurow K. Intelligent arm manipulation system in life science labs using H20 mobile robot and Kinect sensor. 2016 IEEE 8th International Conference on Intelligent Systems (IS). 2016;382-387. https://doi.org/10.1109/IS.2016.7737449.

[8] Behzadi-Khormouji H, Derhami V, Rezaeian H. Adaptive Visual Servoing Control of robot Manipulator for Trajectory Tracking tasks in 3D Space. 2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM).2017;376-382, https://doi.org/10.1109/ICRoM.2017.8466231.

[9] Shim KH, Jeong JH, Kwon BH, Lee BH, Lee SW. Assistive Robotic Arm Control based on Brain-Machine Interface with Vision Guidance using Convolution Neural Network. 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC), 2019;2785-2790. https://doi.org/10.1109/SMC.2019.8914058.

[10] Ma Q, Niu J, Ouyang Z, Li M, Ren T, Li Q. Edge Computing-based 3D Pose Estimation and Calibration for Robot Arms. 2020 7th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2020 6th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom). 2020; 246-251. https://doi.org/10.1109/CSCloud-EdgeCom49738.2020.00050.

[11] Farag M, Abd Ghafar AN, Alsibai MH. Grasping and Positioning Tasks for Selective Compliant Articulated Robotic Arm Using Object Detection and Localization: Preliminary Results. 2019 6th International Conference on Electrical and Electronics Engineering (ICEEE), 2019;284-288. https://doi.org/10.1109/ICEEE2019.2019.00061.

[12] Mars 2020 Perseverance Rover Wallpaper [1080x2297] : r/MobileWallpaper n.d. https://www.reddit.com/r/MobileWallpaper/comments/ltjuio/mars_2020_perseverance_rover_wallpaper_1080x2297/

[13] https://www.researchgate.net/figure/Kinect-v1-uses-the-structured-light-triangulation-method-for-depth-sensing_fig2_277637546

[14] ros-visualization/rviz. GitHub. https://github.com/ros-visualization/rviz (accessed Feb. 02, 2022).

[15] AmandaDattalo. ROS/Introduction - ROS Wiki 2019. http://wiki.ros.org/ROS/Introduction (accessed February 2, 2022).

[16] Karl-Bridge-Microsoft. Kinect for Windows - Windows apps 2022. https://learn.microsoft.com/en-us/windows/apps/design/devices/kinect-for-windows (accessed February 27, 2023).

[17] Keras Team. Keras documentation: Getting started n.d. https://keras.io/getting_started/ (accessed February 2, 2022).

[18] TensorFlow Developers. TensorFlow 2023. https://doi.org/10.5281/ZENODO.4724125.

[19] MarkoBjelonic. darknet_ros - ROS Wiki 2018. https://wiki.ros.org/darknet_ros (accessed February 2, 2022).

[20] Tossy. YOLO V4 for darknet_ros 2023.

[21] Bjelonic M. YOLO ROS: Real-Time Object Detection for ROS 2018.

[22] Nvidia. CUDA Instalation Guide Windows. CUDA Instalation Guide Windows 2022. https://docs.nvidia.com/cuda/cuda-installation-guide-microsoft-windows/index.html (accessed February 2, 2022).

[23] Rico FM, Ramos FG. gb_visual_detection_3d 2023.

[24] Rico FM. Personal Blog Francisco Martín Rico 2022. https://gsyc.urjc.es/~fmartin/ (accessed February 2, 2022).

[25] Epson America, Inc. 6-Axis Robots | Explore by Product Series | Epson US n.d. https://epson.com/6-axis-robots-product-family (accessed February 2, 2022).