

Personal Assistant Robot

Ziany Alpholicy X.^{1,*}, Sagar S. Bhandari¹, Praveen P. Dsouza¹,
Divanshu C. Raina¹

¹*Xavier Institute of Engineering, Opposite S.L. Raheja Hospital,
Mahim (West), Mumbai 400037, Maharashtra, India*

**Corresponding Author: ziaxavier@gmail.com*

(Received 02-08-2020; Revised 29-12-2021; Accepted 29-12-2021)

Abstract

Since the boom in science and technology, humans have been trying to invent machines that could reduce their efforts in day to day activities. In this paper, we develop a personal assistant robot that could pick up objects and return it to the user. The robot is controlled using an android application in mobile phones. The robot can listen to user's command and then respond in the best way possible. The user can command the robot to move to given location, capture images and pick objects. The robot is equipped with ultrasonic sensor and web camera that helps it to move to different location effectively. It is also equipped with sleds that play important role in object picking process. The robot uses a tiny YOLOv3 model which is rigorously trained on several images of the object. There are some possible improvements that can be achieved which could help this robot to be used in several other fields as well.

Keywords: actuators and sensors, TCP socket, object detection

1 Introduction

In recent years, humans have been creating various machines to help the physically challenged people. As people age they tend to face challenges in their every-day life and hence they require assistance from others to carry out their routine work. They are challenged to move physically and if they need to pick up objects they may require someone's assistant. Also, there are workers in factories that might face difficulties while working as they might be surrounded with hazardous chemicals or dangerous machines. People working in hospitals might also be exposed to several diseases while handling the relocation of different medical equipment. In this paper, we develop a personal assistant robot that a user can operate without physically moving from their location. There exists many such robots that can carry out routine jobs automatically. Although many of them are still incapable of eradicating the above mentioned problem.

2 Research Methodology

A technical research paper in 2015, published by the students of Indian Institute of Information Technology, Chittoor, described the development of an assistant robot that can be operated using speech commands and that can be used in hospitals, homes, industries and educational institute. They developed a robot that can be controlled using human voice. The robot could move to different locations and relocate an object from one place to another. They implemented a robotic arm by calculating several parameters such as variation of angle and distance between the robotic hands with time, angular velocity of the robotic arm [1].

In 2005, another research paper was published by some students of National Chiao Tung University, which described the development of develop a personal assistant robot which should be able to assist the user physically with real movement and actions. They developed the robot that had its own intelligent sensors and actuators. They also implemented the face tracking function which was achieved by radial basis function type neural network (RBFNN). They implemented several features like home care, remote monitoring, security, etc [2].

Recently, in 2018, another research paper was published where an interactive personal assistant robot was developed using the Raspberry Pi computing engine. The robot that they developed was self-balancing which was implemented with the help of principle of dynamic balancing. They used Google Text To Speech (GTTS) to convert the voice commands into texts which can be recognized by the computing engine. They used bluetooth services to maintain connection between the robot and the mobile application. They also implemented multiple face recognition system so that they could replace the robot with the security guard [3]. More information can be found in the literature [4], [5], [6].

3 Results and Discussion

The object detection model used here is YOLO V3. The raspberry pi, due to its limited computing power, cannot be used to implement the original YOLO model. Hence, we use YOLO Tiny, a tiny and yet efficient version of YOLO. For the datasets we used Google's open images dataset V6. To train the model, we have to set the number of batches for the datasets so as to determine the iteration of training the model. The ideal number of iterations should be 2000 batches per number of objects. In our case we are dealing with 7 objects so the batches should be 14,000. While the model is training the prime objective of the algorithm is to decrease the average loss. We started with average loss of 4.5 and reached to an average loss of 1.08. Figure 1 shows the graph depicting the decrease in average loss as the number of iteration increases.

The robot and android are connected through TCP socket connection. The Raspberry Pi is cond to work as a standalone network and whenever booted up starts its hotspot. The android application is required to connect to the pi hotspot. A server socket is created by raspberry pi upon boot up which waits for the client to connect to the server. As soon as the android application is connected to pi hotspot through Wi-Fi network, the user can create a client socket and connect to the server listening on pi. The client-server socket connection then can be used to transfer data between the robot and the android application.

In order to execute the movement of robot from its starting place to the desired location, the paths need to be set within robot. The robot uses this pre-defined path to

reach to a particular location. There can be several paths for different location within an area. To set a path, the user is supposed to move the robot from its home location to that particular spot. While the robot is moving it records the command given to it by the user and save it in the dynamic python list. When the user completes the moving job, the robot saves the path from the list in the form of text file and saves it into its path directory. The robot also computes the returning path just by reversing the commands given it by the user. So, two files are saved as one complete path and these files are used to execute the movement of robot from home location to path and vice versa.

When the user commands the robot to pick an object, the robot uses the path text files saved within the path directory to reach to the desired location. As soon as the robot reaches the location, it uses its webcam to capture the images and then uses those images to search for the object which user has decided to pick. It uses YOLO algorithm to detect the object in the captured image. As soon as the object is detected, the robot uses the ultrasonic sensor to measure the distance between itself and the object. According to the readings of sensor, it then relocates itself as much close to the object such that it can pick it up. After picking the object it then gets back to its home location using the path files. Figure 2 and Figure 3 show the object detection result on two different objects. Figure 4 shows the robot with sled opened while Figure 5 shows the robot with sled closed.

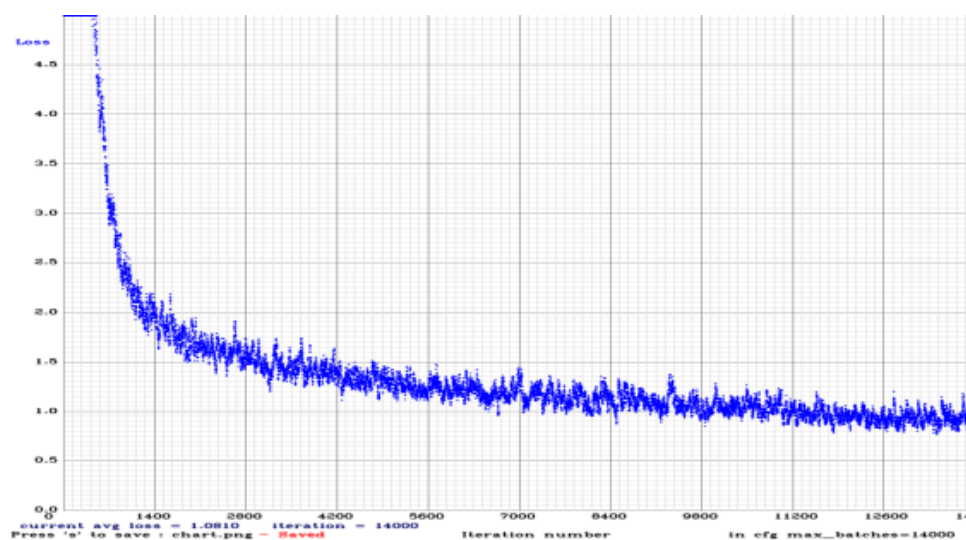


Figure 1. Graph depicting the decrease in average loss as the number of iteration increases



Figure 2. The object detection result on two different objects

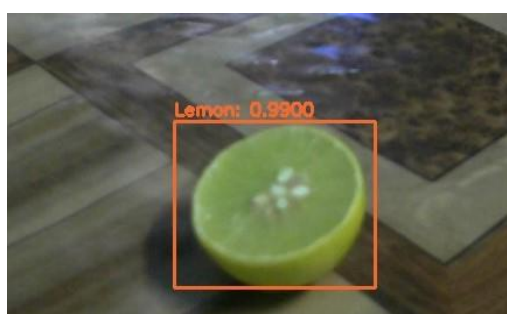


Figure 3. The object detection result on two different objects

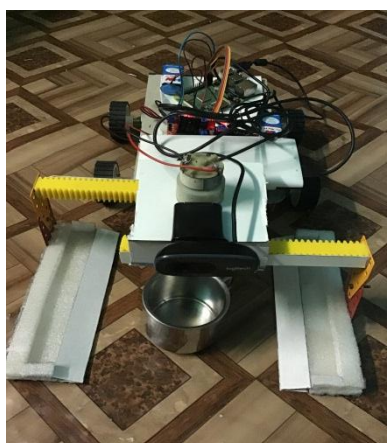


Figure 4. The robot with sled opened

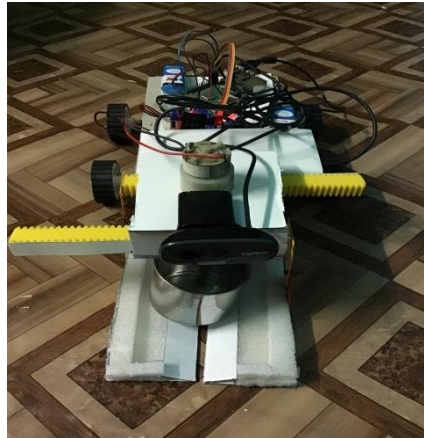


Figure 5. The robot with sled closed

4 Conclusion

The robot developed can be used in different scenario ranging from normal home use to hospital or chemical industries. It is equipped with intelligent features that can be used for efficiently picking several objects. The path storing and object detection via ultrasonic sensor helps the robot to precisely locate from one place to another. The robot can be equipped with voice recognition feature enabling its usage effectively and directly to physically challenged people. The robot can also be equipped with speakers which can speak out information regarding the position of the robot relative to the path saved in its memory. It can also speak out list of object available, paths set within its memory. The robot can also be equipped with Google Coral in order to increase its speed and allow the user to get live stream video from robot's webcam. It will also increase its speed in detecting object. The number of ultrasonic sensor can also be doubled so as to prevent the repeated checking of object by turning left. There's an alternate solution to this solution i.e. using stepper motor to use one ultrasonic sensor for both the direction. The robot sled can also be converted to robotic hands or design inspired from claw to increase its capabilities of picking objects.

References

- [1] A. Mishra, P. Makula, A. Kumar, K. Karan and V. K. Mittal, “A Voice-Controlled Personal Assistant Robot.” *International Conference on Industrial Instrumentation and Control (ICIC)*, 2015.
- [2] I.H. Shanavas, P.B. Reddy and M.C. Doddegowda, “A Personal Assistant Robot Using Raspberry Pi.” *International Conference on Design Innovations for 3Cs Compute Communicate Contro*, 2018.
- [3] C.H. Lin, H. Andrian, Y.Q. Wang, and K.T. Song, “Personal Assistant Robot.” *Proceedings of the 2005 IEEE International Conference on Mcchafronics*, 2005.
- [4] Real-time object detection with deep learning and OpenCV, <https://www.pyimagesearch.com/2017/09/18/real-time-object-detection-with-deep-learning-and-opencv/>
- [5] Train your own tiny YOLO v3 on Google colaboratory with the custom dataset, <https://medium.com/@today.rafi/train-your-own-tiny-yolo-v3-on-google-colaboratory-withthe-custom-dataset-2e35db02bf8f>
- [6] The AI Guy. (2020, January 28). YOLOv3 in the CLOUD: Install and Train Custom Object Detector (FREE GPU) [Video file]. Retrieved from <https://www.youtube.com/watch?v=10joRJt39Ns&t=1440s>

This page intetntionally left blank