PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)
100-ft Ring Road, Bengaluru – 560 085, Karnataka, India

Report on

DESIGN AND IMPLEMENTATION OF DESKTOP PET

Submitted by

Nesha Kavya Madder (PES1UG19EC181)

Nesara K V (PES1UG19EC180)

Navya N Udupa (PES1UG19EE069)

Keshika Shankar (PES1UG19EE089)

JAN-MAY 2022 & AUG-DEC 2022

Under the guidance of

Prof. Swetha G
Assistant Professor
Department of Electronics and Communication Engineering
PES University
Bengaluru - 560085

FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING
PROGRAM B. TECH

CERTIFICATE

This is to certify that the Project entitled

DESIGN AND IMPLEMENTATION OF DESKTOP PET

is a bona fide work carried out by

Nesha Kavya Madder (PES1UG19EC181)

Nesara K V (PES1UG19EC180)

Navya N Udupa (PES1UG19EE069)

Keshika Shankar (PES1UG19EE089)

In partial fulfillment for the completion of the Program of Study B. Tech in Electronics and Communication Engineering under rules and regulations of PES University, Bengaluru during the period January–December 2022. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report. The project report has been approved as it satisfies the in respect of project work.

Signature with date &Seal	*Signature with date & Seal*	*Signature wit date & Seal*
Prof. Swetha G	*Dr. Anuradha M*	*Dr. Keshavan B K*
Internal Guide	*Chairperson*	*Dean of Faculty*

Name of the students
Nesha Kavya Madder (PES1UG19EC181)
Nesara K V (PES1UG19EC180)
Navya N Udupa (PES1UG19EE069)
Keshika Shankar (PES1UG19EE089)

Name / Signature of the Examiners:
Name Signature

DECLARATION

We, Nesha Kavya Madder, Nesara K V, Navya N Udupa, Keshika Shankar hereby declare that the project entitled, " DESIGN AND IMPLEMENTATION OF DESKTOP PET", is an original work done by us under the guidance of **Prof. Swetha G**, **Assistant Professor, Department of Electronic and Communication Engineering, PES UNIVERSITY**, and is being submitted in partial fulfilment of the requirements for completion of 6th semester course work in the Program of Study B.Tech in **Electronics and Communication Engineering.**

PLACE: BENGALURU

DATE:

NAME AND SIGNATURE OF THE CANDIDATE

Nesha Kavya Madder	**(PES1UG19EC181)**
Nesara K V	**(PES1UG19EC180)**
Navya N Udupa	**(PES1UG19EE069)**
Keshika Shankar	**(PES1UG19EE089)**

ACKNOWLEDGEMENT

We would like to thank this opportunity to thank **Prof. Swetha G**, Assistant Professor, Department of Electronics and Communication Engineering, PES University, for his persistent guidance, suggestions, assistance and encouragement throughout the development of this project. It was an honor for me to work on the project under his supervision and understand the importance of the project at various stages.

We are grateful to the project coordinator **Prof. Rajasekar M**, Department of Electronics and Communication Engineering for, organizing, managing, and helping with the entire process.

We would like to thank **Dr. ANURADHA M**, Professor and Chairperson, Department of Electronics and Communication Engineering, PES University, for her invaluable support and thankful for the continuous support given by the department. I am also very grateful to all the professors and non-teaching staff of the Department who have directly or indirectly contributed to our research towards enriching work.

Chancellor **Dr. M R DORESWAMY**, Pro-Chancellor **Prof. JAWAHAR DORESWAMY**, Vice Chancellor **Dr. SURYAPRASAD J**, the Registrar **Dr. K S SRIDHAR** and Dean of Engineering **Dr. KESHAVAN B K** of PES University for giving me this wonderful opportunity to complete this project by providing all the necessaries required to do so. It has motivated us and supported our work thoroughly.

We are grateful and obliged to our family and friend for providing the energy and encouragement when we need them the most.

Nesha Kavya Madder (PES1UG19EC181)
Nesara K V (PES1UG19EC180)
Navya N Udupa (PES1UG19EE069)
Keshika Shankar (PES1UG19EE089)

ABSTRACT

Mobile Robots play a major role in robotics. They also act as companion in this fast running word .The goal of this project is to generate a indoor mobile robot which can navigate around the Desktop .The project also aims at getting the stability information about the surrounding and move around desktop with detection of edge and obstacle .

This paper proposes a robust approach and cost-effective approach for obstacle detection and avoidance along with simultaneous edge detection along with voice assistant . It utilizes the USB camera, motor driver and ultrasonic sensor data for obstacle avoidance . IR sensor is used for edge detection of the desktop. It also ultisizes the integration of USB camera with ROS for object detection . It uses python api integration for providing an voice assistant .

This paper aims at building a desktop pet in a effective way using both camera and sensor data. The robot was successfully able to walk forward, turn left and right without any support. It was also able to perform operations like detect the obstacle and name the obstacle present infront of it . The robot can successfully save data from its force sensors and navigate with detecting objects and also provides a voice assistant.

.

Table of Contents

LIST OF FIGURES

1.INTRODUCTION

Outdoor autonomous robot navigation is a topic of active research. Finding safe, passable routes for navigation that enable the robot to go forward while avoiding obstacles is necessary for the mission. Any autonomous system must have the capacity to sense its surroundings using external sensors, such as vision, distance, or proximity sensors. Ultrasound and lidar are two examples of distance sensors that primarily provide us with perspective in the sensor's plane. These sensors are positioned on moveable, rotatable platforms that may be manipulated in order to get it in all directions. The lowest and greatest distances that can be measured by these sensors are likewise restricted. Therefore, using numerous sensors is necessary to obtain better data. 3D-lidar sensors can scan the surrounding area very well.

The educational robots are designed to be economical, with low-cost sensors and reproducible motor drives. These machines' embedded sensors are quite sensitive to background noise. These robots are designed to graphically illustrate theoretical ideas. These robots' repeated use is compromised if they are utilized in settings where the impact of indicating sources increases the noise in sensing. Additionally, when advanced robotics concepts are tested, the robots frequently fail.

The main task of this research is to use the open source software platform robot operating system (ROS) as the core to design a set of mobile robot systems with low cost, high performance, and scalability

Recently, researchers have shown particular interest in low-cost educational robots. Recent works (S. Wang and Magnenat, 2016; S. Bazeille and Filliat, 2015) show that there has been a growing trend of trying to accommodate accessible low-budget robots in the academic environment.

1.1 MOTIVATION

Service robots must be capable of self-localization in order to carry out the duty efficiently. Numerous researchers have previously developed the robot localization method utilising a variety of sensors. These plans, however, frequently raise the system's cost. In order to service robots that are widely used in the real world, the major problem is how to reduce the cost of the robot system. In this study, we present a reliable technique that would help for mobile robots that uses cameras and reasonably priced sensors to identify the environment.

1.2 PROBLEM STATEMENT

Design and implementation of desktop pet

Goal:

1. Object detection and edge detection using the fusion of USB camera and IR sensor

2.Obstacle avoidance using ultrasonic sensors

3. To develop a voice assistant like alexa,siri which gives us the update time, tells us jokes, and gives us information from Wikipedia

2.LITERATURE SURVEY

2.1 "ROS-based Stereo Vision System for Autonomous Vehicle"

Objective:

Designing an autonomous vehicle that is cost-effective and maintains a constant speed and distance for surveillance.

Key points:

- The vehicle is made with acrylic sheets to accommodate a maximum payload of 2kg.
- The model has basically three layers that accommodate the camera, single board computers, motor drives, motor, battery, and microcontroller.
- A gap of 80mm is left between each stage for the feasibility of the model which is done by machined clamps.
- The wheels are made to turn from left to right, thus rotating in the opposite direction using the differential drive system.
- A specific control system is used for position lock and autonomous navigation.
- The accelerometer is used to obtain the position of the vehicle by providing the position feedback in terms of x and y components.
- The feedback is compared to the set point and an error signal is generated which is given to the PID controller.
- ROS is used i.e it is an open-source platform that is used for communication, data processing, and data acquisition,
- The ROS in the base station acts as a publisher and acts as a subscriber on the onboard computer.

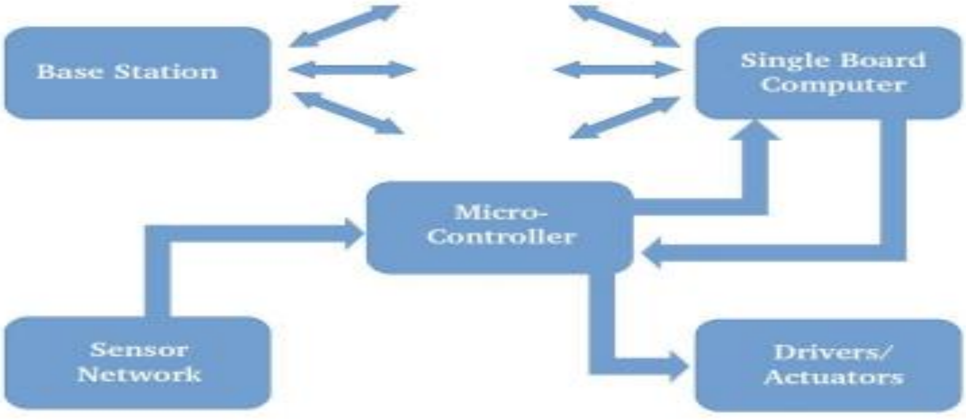


Figure 2.1.1 Block diagram for electronic integration.

Takeaways:

- ROS is being used for effective communication between the system.
- The control system ideology is being utilized well
- Sensor data was accurate and is well used for position lock and autonomous navigation.

2.2 “Obstacle Detection and Avoidance Robot”

Objective: The obstacle is being detected at a distance of 15 cm when it senses the right path to move with the level of distance to the next obstacle

Key Points:

- The microcontroller used in the robotic platform is ATmega328P. Programmed by arduino, it contains 14 digital pins and 6 analog pins.
- Three ways to power the arduino board is, by connecting type b usb, 9v battery, and by connecting the positive end of the 9V to Vin and negative to ground.
- As the 5v supplied to the arduino board is not enough to rotate the motor, L293D Motor driver is used.
- Bluetooth HC-05 module, is used which acts as an interface between mobile and arduino. It basically works as a master/slave module.
- For rotation servo motors are used, SG-90 servo motor is being used here. It rotates between 0 to 180 degrees.
- The servo used here is of 1.6kg/cm and a power supply ranging from 4.8-6V.
- HC-SR04 Ultra sonic sensor is being used as it has frequencies above 20Khz and can detect an object from 2cm upto 400cm.
- The pins of the ultrasonic sensor i.e Vcc, ECHO, TRIG and GND.
- SD card module is used to transfer the data from arduino to sd card.
- The servo is used to mount the sensor onto the top of the robot, which in turn helps the sensor to detect the obstacle within the reach of 15cm.

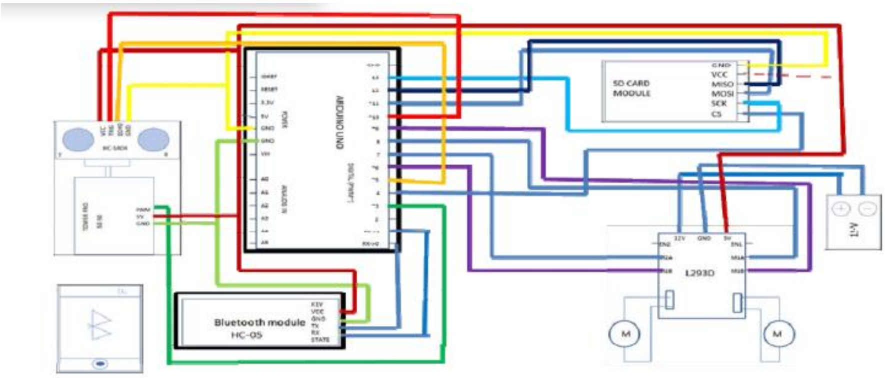


Figure 2.2.1 Pin Diagram

Takeaway:

- The ultrasonic sensor is used for very effective obstacle detection.
- It is much cheaper as compared to other devices.
- Servo motor can be utilized to increase the area coverage of the ultrasonic sensor.

2.3 Object Detection Based on YOLO Network

Objective :

While capturing are many problems with images in real-world shooting such as noise, blurring and rotating jitter, etc. All these problems have an crucial impact on object detection. Using traffic signs as an example, we established image degradation models which are based on YOLO network and combined traditional image processing methods to simulate the problems existing in real-world shooting. After establishing the different degradation models, we compared the effects of different degradation models on object detection. We used the YOLO network to train a robust model to improve the average precision (AP) of traffic signs detection in real scenes.

Key Points :

They have selected 1,652 images containing traffic signs as the data set. Then, 1,318 images were selected which form the standard training set, and the remaining 334 images were used as the test set. All traffic signs in the images are labelled. Furthermore, they chose 334 images which were taken in the real world to form the test set that analyzed the model's performance. All these images have degeneration processing. The background of these images is complex, the light of the images has obviously changed and they have different degrees of occlusion.

We use the YOLOv3 neural network on an Intel CPU i7 -4790 16G memory, NVIDIA GTX1080,Ubuntu16.04, a 64-bit operating system for model training and testing.
We use 0.9 momenta and 0.0005 decay in order to speed up training and prevent overfitting. According to the training set, the learning rate will get adjusted . All models are based on the Darknet-53 neural network and use the same initial weight values for training, while it is different for the training sets.

They made some rules for ease of description. At the start , we assumed the coordinates origin (0,0) to be the lower left corner of the image. Second, checking whether the images were all in the first quadrant or not . The x-axis was the width of the image and the y-axis was the height of the image.

Takeaways :

The YOLO neural network integrates the candidate boxes extraction, feature extraction and objects classification methods into a neural network. The YOLO neural network extracts candidate boxes from images and objects are detected through the entire image features.

In the YOLO network, the images are divided into S×S grids. Candidate boxes are equally distributed on the X-axis and Y-axis. The candidate boxes have object detection and predict the confidence of the existence of the object in each candidate box. Confidence reflects whether the images include the object or not, as well as the accuracy of the object's Position.

The propose using the YOLO network model for object detection. We train the degenerative model which is fed with the degraded image. Through experiments we find that the network which is trained with the degraded images learns more features and that the model can cope with more complex scenes. The results show that the model improves the average precision of the object detection. The model which is trained with the degraded training sets has better generalizing ability and higher robustness.

2.4 Raspberry pi based voice operated personal assistant

Objective:

The purpose of this research project is to develop a personal assistant using the Raspberry Pi as the processor chip and supporting infrastructure. It stresses the replacement of screen-based interaction with ambient technologies like robotics and the Internet of Things, which means the user interface is built into the actual device. It is made up of parts like IR sensors, a Pi camera, a microphone, and a motor driver. It is a voice-controlled personal assistant with the ability to read text from photos and then articulate the equivalent to the client using the built-in speaker. Its movements are controlled by voice commands. By providing them with access to educational resources like Wikipedia, calculators, and other tools, it can assist those who are physically impaired in communicating with the rest of the world.

Keypoints:

- It is made up of parts like IR sensors, a Pi camera, a microphone, and a motor driver.
- It is a voice-controlled personal assistant with the ability to read text from photos and then articulate the equivalent to the client using the built-in speaker.
- By providing them with access to educational resources like Wikipedia, calculators, and other tools, it can assist those who are physically impaired in communicating with the rest of the world.

Takeaway:

- It uses raspberry pi as its processor chip.
- Use of microphone and motor driver.

3. METHODOLOGY

3.1 WORKFLOW OF OBSTACLE DETECTION

Methodology for the proposed work has been shown in the below flowchart. As per methodology, targets detected from ultrasonic sensor input data are denoted by tracks with information like the range associated with them. This information is extracted from the sensor. For a camera, the characteristics of the camera are determined by the calibration process. Subsequently, data is collected and processed frame-wise in real time to extract the region of interest accordingly by the YOLOv3 algorithm. Identified Obstacles are bounded with boxes and information on each obstacle is also gathered through it.

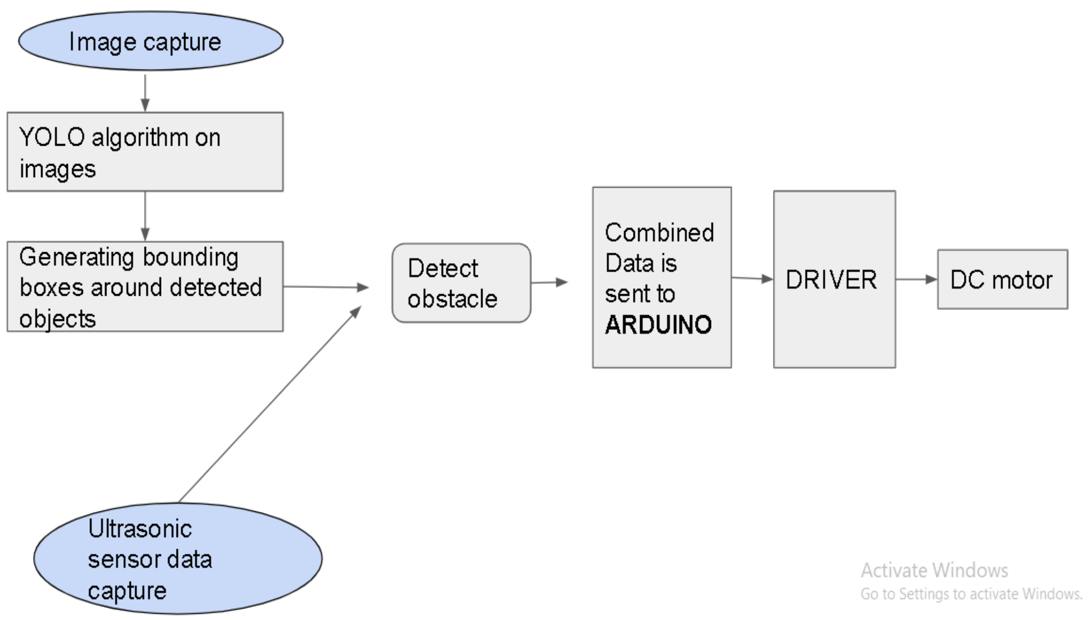


Figure 3.1.1 Proposed methodology for vision and radar data fusion for obstacle identification

For proper fusion of camera and sensor data, internal and external characteristics of the camera need to be determined for obtaining a transformation matrix. The camera calibration is done using ROS packages while determining intrinsic parameters and extrinsic camera parameters.

Conceptual representation of the coordinate system used by the camera and sensor for a target point is illustrated below. (Xc, Yc, Zc) stands for camera coordinates, (Xr, Yr, Zr) for sensor coordinates, and (Xr, Yr, Zr) for image coordinates (u, v). The coordinates of the target are Xr=rsin and Yr=rcos, and the sensor only provides us with the range(r) and angle, which only aids in determining the position of a potential target in a 2D plane. For the intended use, these coordinates must be converted to camera coordinates. As radar returns occur in a 2D plane with the value in the third dimension of coordinates always being 0, this transformation of the sensor's and camera's coordinates is now quite challenging. As seen in the above picture, camera data is obtained in the XrZr plane, whereas sensor returns are obtained in the YrZr plane.

After registration of camera data with radar data, the local position of objects like humans and other obstacles are measured, relative to the position of a vehicle. Simultaneously, an object classification task has been performed by YOLOv3 to give precise information which includes a class of objects along with range and angle. Then we let the radar points be the reference centers and sequentially process the image through the YOLOv3 algorithm with pre-trained weights from the COCO dataset which contains undistorted color images and contains 80 object classes to get the classified object further through radar we get some information on it which helps in determining the exact location of a particular classified object in the view of a vehicle.

3.2 WORKFLOW OF EDGE DETECTION:

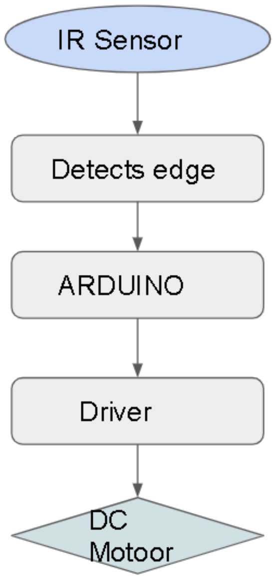


Figure 3.2.2 Workflow of edge detection

Here we have incoorporated two sensors:

1. When both the sensors reaches edge it moves backward .
2. When right sensor senses an edge it will try to move left.
3. In the same way,When left sensor senses an edge it will try to move right.

3.3 Control System of Mobile Robot Based using

The top robotics development environment today is ROS. Software developers can design robot applications with the use of the libraries and tools provided by ROS. It offers a variety of features, including hardware abstraction, device drivers, libraries, visualizers, message-passing, and package management. Distributed process framework ROS is contained in usable packages that are simple to share and release. To free us from laborious programming, ROS offers a full suite of robot communication techniques.

Two components make up the entire architecture:

1. When the camera picks up an object.
2. Ultrasonic sensor detects an object or edge.

We can determine if the object is present at a specific location or not by combining these two pieces of information.

Next, command and maneuver the robot to

The bot must now be moved and controlled using UART communication to avoid these items and edges.

The ros architecture:

The roscore:

To begin with, let's make sure Roscore is running.

A ROS-based system must have a number of nodes and programs, collectively known as roscore. In order for ROS nodes to communicate, a roscore must be operating. Using the roscore command, it is started.

Roscore will begin:

- a rosout logging node
- a rosout master
- a ros parameter server

Master: The ROS Master offers name and registration services to all other ROS system nodes. It keeps tabs on both topic and service publishers and subscribers. The Master's job is to make it possible for different ROS nodes to locate one another. Peer-to-peer communication takes place between these nodes once they have located one another.

The Parameter Server is likewise supplied by the Master.

The ROS Master and other crucial components are loaded by the roscore command, which is the most popular way to operate the Master. Let's take the example of two nodes, a camera, and an image viewer. A typical chain of events would begin with the camera informing the master that it wants to publish photographs on the topic.

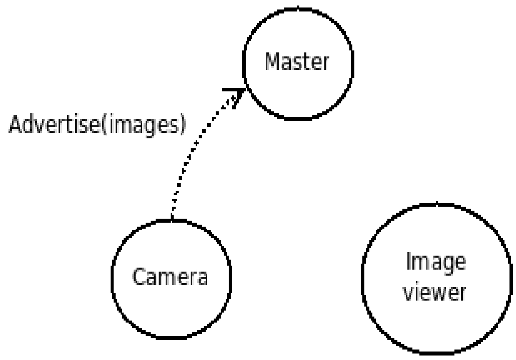


Figure 3.3.1 Communication of camera with master

Currently, Camera publishes photographs to the "images" topic, but as no one has subscribed to that subject yet, no data is actually transmitted. Now, in order to see whether there are perhaps any photos, the Image viewer wants to subscribe to the topic "images":

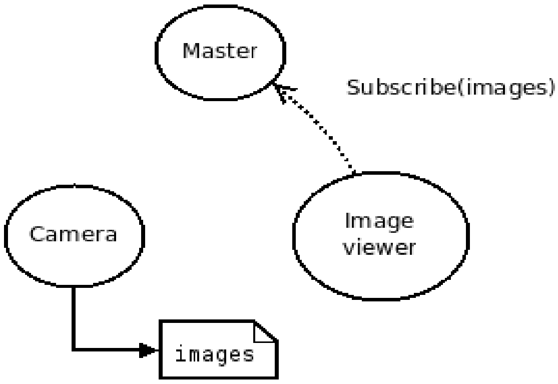


Figure 3.3.2 Communication of image viewer with master

The master node alerts the Camera and Image viewer of each other's existence so they can begin exchanging photos now that the topic "images" has a publisher and a subscriber:

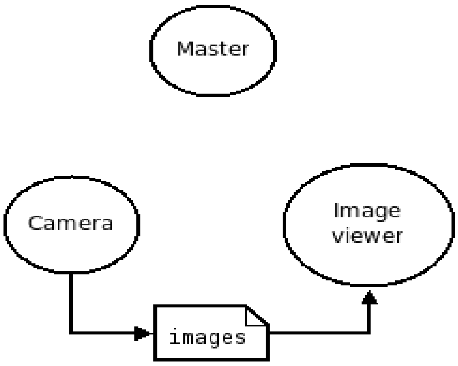


Figure 3.3.3 peer to peer communication

Topic: Nodes exchange messages across topics referred to as buses. Information creation and consumption are separated by topics' anonymous publish/subscribe semantics. Typically, nodes are unaware of their communication partners. Instead, nodes interested in data subscribe to the pertinent subject, and nodes producing data publish to the pertinent topic. A topic may have several publishers as well as subscribers. Each subject has its own name in the ROS network, and messages are transmitted on those topics.

Each topic is strongly typed by the ROS message type used to publish to it, and nodes can only accept messages of the same kind. The subscribers will not establish message transport unless the types match, even though the Master does not enforce type consistency among the publishers.

Topic Transports: TCP/IP-based and UDP-based message transmission are both supported by ROS at the moment. Over persistent TCP/IP connections, the TCP/IP-based transport, also known as TCPROS, streams message data. The sole transport that client libraries must support is TCPROS, which is the default transport used by ROS. The UDPROS UDP-based transport divides messages into UDP packets and is only presently supported by roscpp. Since UDPROS is a lossy, low-latency transport, teleoperation is one of its ideal use cases.
The desired transit is negotiated by ROS nodes in real-time. A node can fall back on TCPROS transport, for instance, if it prefers UDPROS transport but the other node does not support it. As compelling use cases emerge, this negotiation methodology permits the gradual addition of new transports

A command-line tool for engaging with ROS topics is called rostopic.
For example: $ rostopic listrosrun:
Without needing to provide its entire path or cd/roscd there first, rosrun enables you to run an executable in any package from anywhere.
Usage: rosrun <package> <executable>

Node: Actually, a node is nothing more than an executable program contained in a ROS package. A ROS client library is used by ROS nodes to connect with one another. A Topic can be published or subscribed to by Nodes. Nodes may also offer or make use of Services.

Publisher: It is a node that communicates with a topic continually.

Subscriber:This node receives information on a topic to which it has subscribed.

Server and Client: Although the publish/subscribe paradigm is quite versatile, RPC request/reply interactions, which are frequently needed in distributed systems, cannot be handled by its many-to-many one-way transport. Requests and responses are made via a service, which is represented by two messages: one for the request and one for the response. A client requests a service from a ROS node by sending a request message and then waiting for a response. The service is provided under a string name. The programmer is typically shown this interaction by client libraries as if it were a call to a distant operation.

Rosserial_arduino:
This package includes the extensions unique to Arduino that are necessary for rosserial client to function on an Arduino. It is intended to show how simple it is to use an Arduino to incorporate inexpensive sensors and custom hardware into your ROS project.

The two key features of this publisher and subscriber communication:
1.Many-to-many communication uses topics. Numerous subscribers can receive communications from numerous publishers on the same subject.
2.Publishers and subscribers can be added and deleted in any order and are decoupled through topics. Even though a subject has no active subscribers, a message can still be published there.

Messages: When ROS nodes broadcast data values (also known as messages), they do so using a streamlined message description language. A service is described in a SRV file. There are two elements to it: a request and an answer. A package's msg directory houses msg files, while the SRV directory houses srv files. Msgs are essentially plain text files with one line per field type and field name. You may utilize the following field types:

- nt8, int16, int32, int64 (plus uint*)
- float32, float64
- string
- time, duration
- other msg files
- variable-length array[] and fixed-length array[C]

Different programming languages are supported by the ROS client libraries, enabling communication between nodes.

rospy = python client library

Ros-Serial:rosserial is a protocol for multiplexing numerous subjects and services across a character device, like a serial port or network socket, and enveloping regular ROS serialized messages. It is employed in ros and Arduino communication.

Subscribing to ros topic: Numerous subscribers can receive communications from numerous publishers on the same subject. Through topics, publishers and subscribers can be created and terminated in any order and are divorced from one another.

The two key features of this publisher and subscriber communication are

1. Many-to-many communication uses topics. Numerous subscribers can receive communications from numerous publishers on the same subject.

2. Publishers and subscribers can be added and deleted in any order and are decoupled through topics. Even though a subject has no active subscribers, a message can still be published there.

Step 1: To launch Arduino, open a terminal and type Arduino

Step 2: Select the ros lib - Blink sketch from the Arduino examples menu to launch it.

Step 3: Launch a another terminal and enter the following command roscore

Step 4: Launch a another terminal and enter the following command.

rosrun rosserial_python serial_node.py /dev/ttyUSB0

Step 5: Launch a another terminal and enter the following command.

rostopic pub toggle_led std_msgs/Empty --once

Here,the USB camera captures the live image(video) continuously and publishes to the topic”this”.This node publishes frames of the video.Another node subscribes to this topic and processes the image using yolo and detects the object.

When object is detected this node sends a binary message 1 to arudino using serial communication .The arduino on receiving this stops and moves backwards.

This communication is done using messages.Here,’compressed messages ’ is used.

Raw Message Definition

 # This message contains a compressed image

Header header	# Header timestamp should be acquisition time of image # Header frame_id should be optical frame of camera # origin of frame should be optical center of camera # +x should point to the right in the image # +y should point down in the image # +z should point into to plane of the image
string format	# Specifies the format of the data # Acceptable values: # jpeg, png
uint8[] data	# Compressed image buffer

Compact Message Definition

std_msgs/Header header

string format

uint8[] data

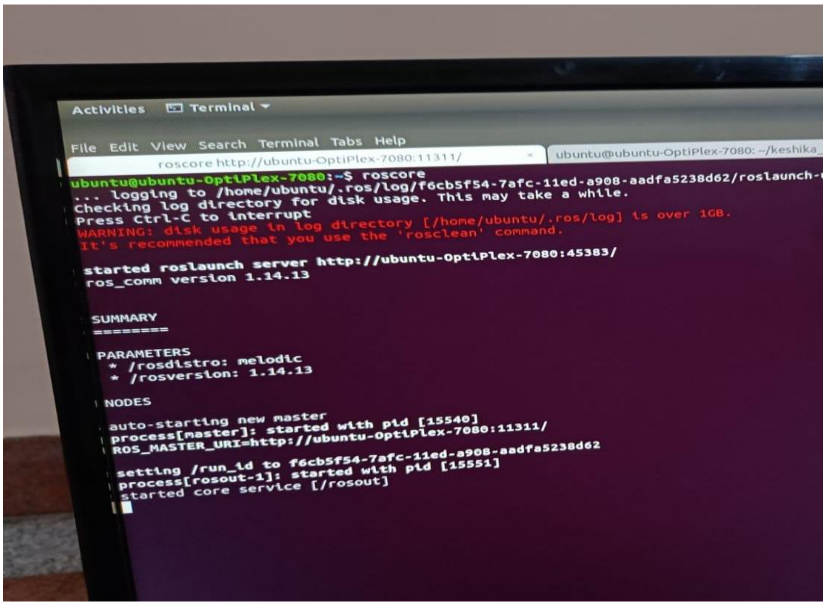


Figure 3.3.4 Starting of the rosscore

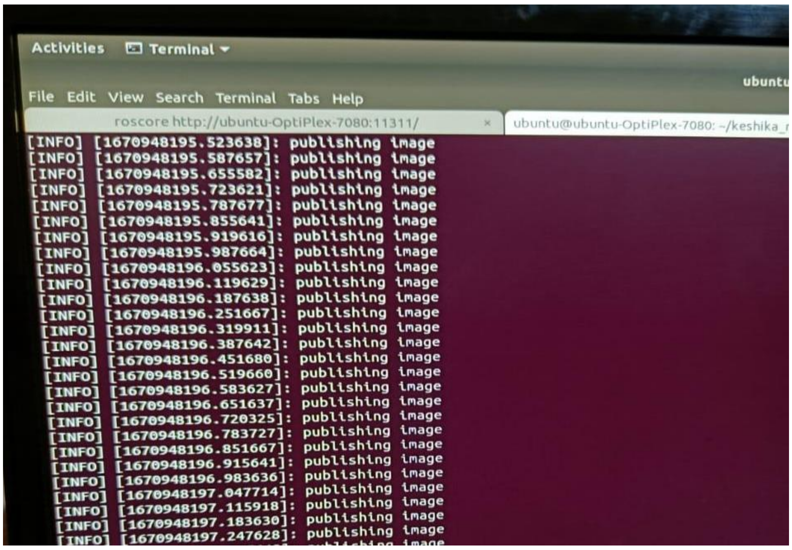


Figure 3.3.5 Publisher

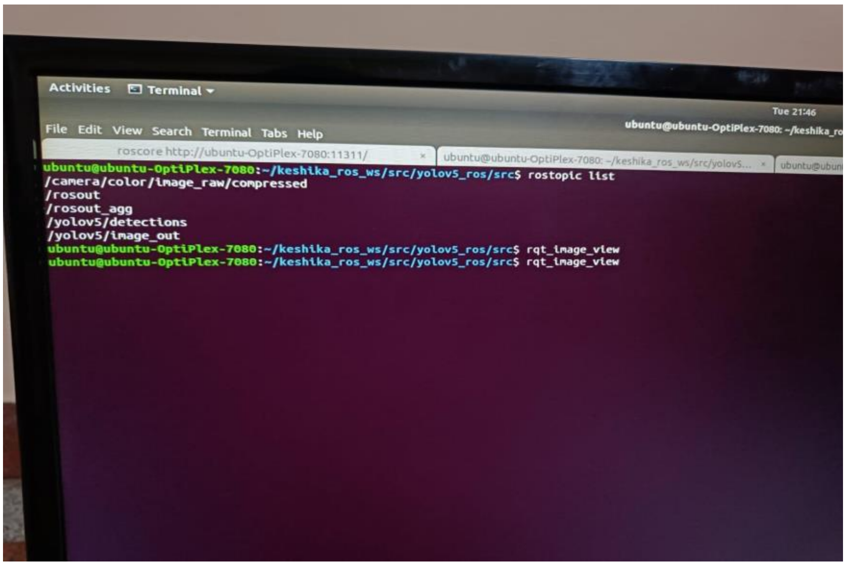


Figure 3.3.6 rqt_imgview for output

3.4 DESCRIPTION AND IMPLEMENTATION OF OBJECT DETECTING ALGORITHM

Introduction

One of the common and difficult issues in computer vision is object detection. Researchers have widely experimented and contributed to the performance improvement of object detection and associated tasks including object classification, localization, and segmentation over the course of the last decade thanks to deep learning's rapid evolution. Object detectors can be broadly divided into two groups: single stage and two stage object detectors. Single stage detectors concentrate on all feasible spatial region proposals for object detection via relatively easier architecture in one go, whereas two stage detectors primarily focus on selected region proposals approach via sophisticated architecture. Any object detector's using inference time and detection accuracy.

In general, two stage detectors perform better in terms of detection accuracy.

1.Obtaining the region of interrest

2.Obtaining the perspective transformation

3.YOLO Algorithm detection

YOLO is a method that provides real-time object detection using neural networks. The popularity of this algorithm is due to its accuracy and quickness. It has been applied in a variety of ways to identify animals, humans, parking metres, and traffic lights.The phenomenon of object detection in computer vision includes identifying different things in digital photos or movies. Among the objects found are people, vehicles, chairs, stones, structures, and animals.

The YOLO algorithm employs the following three methods:

• Remaining blocks

• Box regression bounding

• Union Over Intersection (IOU)

Remaining blocks

The image is first separated into several grids. S x S are the dimensions of each grid. The grids created from an input image are displayed in the following image.

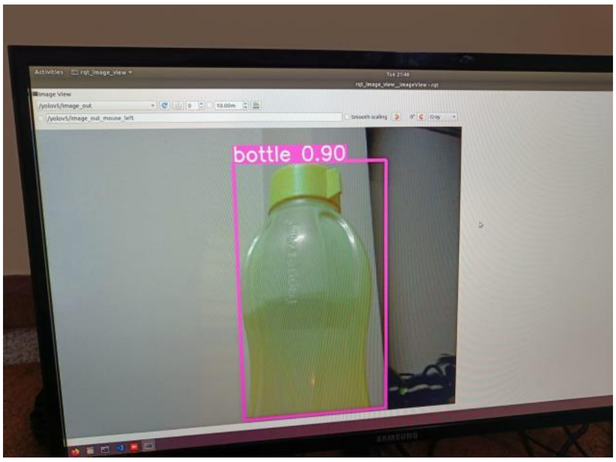


Figure 3.4.1 Bottle detection by YOLO algorithm

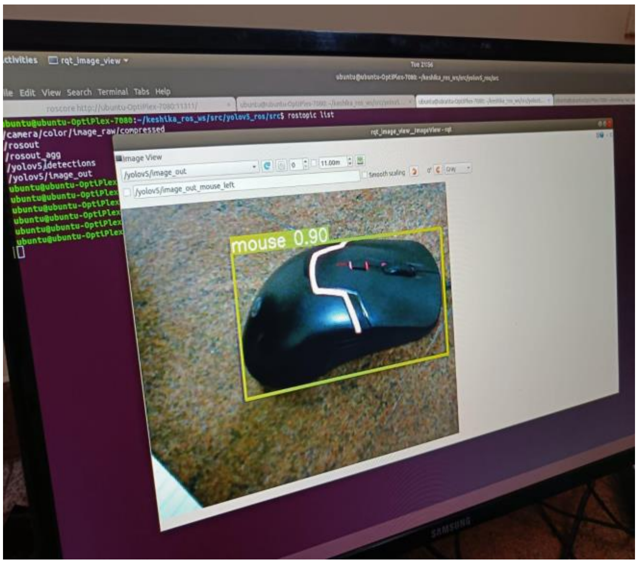


Figure 3.4.2 mouse detection by YOLO algorithm

There are numerous grid cells with equal dimensions in the image above. Every grid cell will be able to detect items that enter it. For instance, a grid cell will be in charge of detecting an object if its centre appears within that cell.

Regression with bounding boxes

An outline that draws attention to an object in a picture is called a bounding box.

The following characteristics are present in each bounding box in the image:

- Height (bh)
- Width (bw)
- Class
- Bounding center(bx,by)

COCO DATASET EXPLANATION

The widespread Common objects in context (COCO) is used In order to facilitate future research for object detection, instance segmentation, image captioning, and person keypoints localization, Common Objects in Context (COCO) is a database. A sizable object detection, segmentation, and captioning dataset is called COCO.

COMPLETE ANALYSIS

The Robot would detect object that is present and turn to other direction . The objects are detected using coco dataset and yolo algorithm .

3.5 OBSTACLE AVOIDANCE USING ULTRASONIC

Objective:

To detect the desktop edge while moving using the ultrasonic sensors and prevent the robot from falling off the desk

Methodology:

The most commonly observed problem in the desktop robot is the detection of the low-level object which is present on the surface. The robot finds it hard to classify the difference between the ground of the table and the low-lying objects, this often leads to stalling the robot during autonomous navigation in the indoor environment.

It is very hard to spot the few centimeters difference present between the low-lying obstacle and the non-obstacle. Here the robot makes physical contact with the low-lying obstacle and spots the subtle difference. We use one or more ultrasonic sensors in order to avoid the drop-off. The robot here as it moves climbs over the low-lying obstacle and uses the ultrasonic sensors' time series data to distinguish whether the object which is present is an obstacle or a non-obstacle and backs up before getting stuck on it. Over the decade, people have developed various techniques to detect the obstacle present which mainly focuses on using either lidars, IR sensors, sonars, radar, etc. This project tends to eliminate the use of such costly elements by making use of ultrasonic sensors. Here ultrasonic sensors are placed on the bottom of the robot to avoid the fall from the cliffs and in order to measure the distance from the ground it transmits the signal.

Working:

Using an ultrasonic sensor, which emits an unclear heartbeat every 0.3 seconds to detect any obstacles within a 100cm radius, collision avoidance is made possible. Taking into account the system's 30 cm width, a servo controls the indication at various locations from 0 to 180. The servo motor, which operates the ultrasonic sensor previously described, rotates it from 0 to 90 degrees. The servo drives the sensor to scan the region from 90 to 180 if there are any objects between 0 and 90, and moves left if there are no obstacles, and vice versa. Similar to this, the robot detects objects at a perfect 90-degree angle, and if any obstacles are found between 0 and 180 degrees, it travels back

3.6 VOICE ASSISTANT

BLOCK DIAGRAM:

The Raspberry Pi is the main part or element of the system as it is necessary to connect various parts or components and to process the requests .Peripherals such as a microphone ,and speaker for the working of the voice

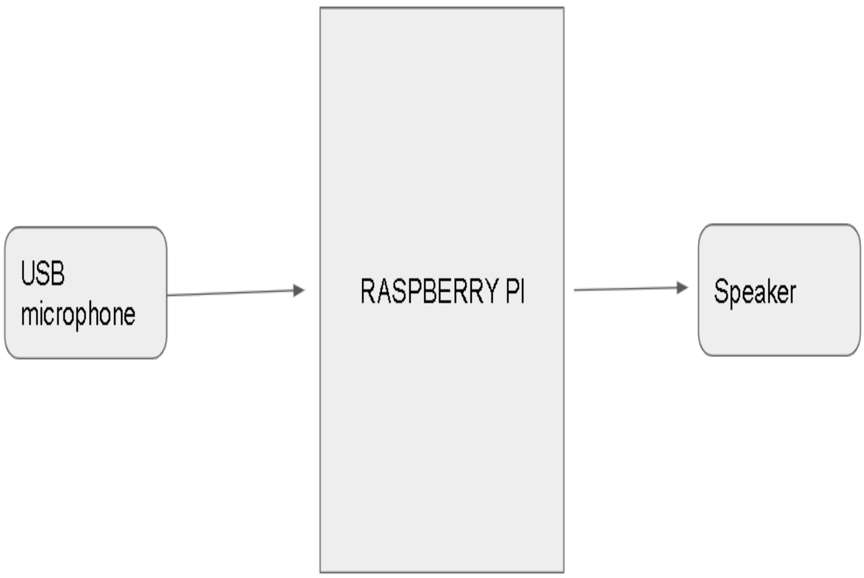


Figure 3.6.1 Voice assistant integration

PYTHON LIBRARIES

Python libraries are inbuilt functions available

RPIO.GPIO

This package offers classes or functions that give the Raspberry Pi's GPIO pins controllability. This package allows us to write/change state as well as read the devices connected to the GPIO pins.

Datetime

When the DateTime object is called by this module, it returns the date and time in the appropriate format.

Pyttsx3

In a programme, this module is used to convert text to speech and operates in the background.

Wikipedia

We have utilised the Wikipedia module to seek or retrieve information from Wikipedia.

Pyjokes

The online collection of Python jokes is done with Pyjokes.

WORKING METHODOLOGY

Using Raspberry as the foundation the system operates. The Raspberry Pi welcomes the user as soon as it is powered on by the user. The programme code-containing Python script is run automatically (auto-start) when the system boots up.

The system receives input through the microphone iI records until a person indicates to stop and is noticed to indicate that the user has concluded the request, and then converts it to text form . The technique is set up so that after processing the request and looking up the keyword, the system returns the proper textual output, which the speech-recognition module then converts to speech.The speaker connected gives speech output.

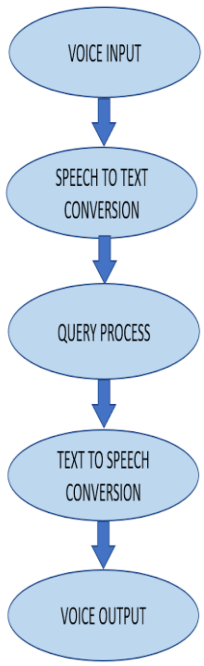


Figure 3.6.2 Working of voice assistant

4. Hardware and Software Requirements

4.1 Software Architecture:

i)Arduino IDE:

A cross-platform program built with Java and Processing is the Arduino IDE. The code editor comes with tools like automatic indentation, brace matching, and syntax highlighting. It only takes one click to compile and submit the code to a board. It does not require the use of makefiles or command-line interfaces to run programs.

ii)Ros:

ROS, known as ros, is a collection of open-source robotics middleware. While not an operating system (OS), ROS is a collection of software frameworks for the creation of robot software. It therefore provides services like dedicated hardware, low-level device control, the implementation of frequently used functionality, message-passing between processes, and package management that are designed for a heterogeneous computer cluster. A graph architecture is utilised to describe the running sets of ROS-based processes, with processing taking place in nodes that can receive, post, and multiplex messages on control, state, planning, actuators, and other issues. Although responsiveness and low latency are necessary for controlling robots, ROS is not a real-time operating system (RTOS). However, ROS can incorporate real-time coding.

4.2 Hardware Used

Raspberry Pi

Raspberry is Small single-board computers (SBCs). The Raspberry pi model is used for applications like robotics because it was more widely used than planned. Its been used mainly for its low cost and open design. It is used widely in numerous fields, including weather monitoring and robotics Project . It supports the HDMI and USB standards.

The code is feed into the raspberry pi , where raspberry pi works according to the code.

Figure 4.2.1 Raspberry pi 3b+

USB camera

A USB camera is one that attaches to a computer by being plugged into a USB port on the robot machine. The nearby items are captured by the USB camera. The photos are taken by the USB camera and transmitted into the computer programme. Real-time photos are captured via a USB camera, and this function aids in real-time working robots.

Figure 4.2.2 USB camera

Arduino UNO

Arduino UNO is a microcontroller kit with ATmega328P. Its features are 14 digital input/output pins with six PWM dual operation pins, six analog inputs with 10 bit ADC, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header, and a reset button. It connects through an adaptor or directly to a computer/laptop via a USB cable, which serves the dual purpose of power supply and acting as a serial port to interface the Arduino and the computer. To avoid power fluctuations, a 9 V-12 V AC can power it through a DC adapter

Figure 4.2.3 Arduino UNO

IR Sensor

IR reflectance sensors operate by counting how much light is reflected in the receiver. IR sensors are made up of two diodes, one of which emits light and the other of which receives it.The robot detects a white surface when the receiver receives a reflection ray, and it detects a black surface when the receiver cannot receive a reflection ray. The office robot uses IR sensors to sense the line.

We used 2 sensors for the designed robot

It is a piece of technology that detects and measures infrared radiation from its surroundings. Despite being on the same electromagnetic spectrum as visible light, IR is not visible to the human eye because its wavelength is longer.

Figure 4.2.4 IR sensor

Ultrasonic Sensor

An ultrasonic sensor is an instrument that measures the distance to an object using ultrasonic sound waves.An ultrasonic sensor transmits and receives ultrasonic pulses from a transducer to determine the proximity of an item. Boundaries reflect high-frequency sound waves, creating distinctive echo patterns.High-frequency sound waves reflect from boundaries to produce distinct echo patterns.

The principle behind the working of ultrasonic sensor is that it sends an ultrasonic pulse which travels through the air and if there is an obstacle or object, it will bounce back to the sensor.

Figure 4.2.5 Ultrasonic Sensor

Lead Acid Battery :

The lead Acid Batter that is being used is of 12 volts (1.7A).

Lead acid Battery is used because a gel electrolyte is utilised rather than a liquid, the battery can be used in various situations without leaking. It is rechargeable and offers continuous power supply for a longer period of time.

Figure 4.2.6 Lead Acid Battery

Sensor Placement:

IR sensor are placed below the robot so that when the distance goes beyond 15cm it indicates that is has encountered the edge and it has to move back.

Here we have incoorporated two sensors:

1. When both the sensors reaches edge it moves backward .
2. When right sensor senses an edge it will try to move left.
3. In the same way,When left sensor senses an edge it will try to move right.

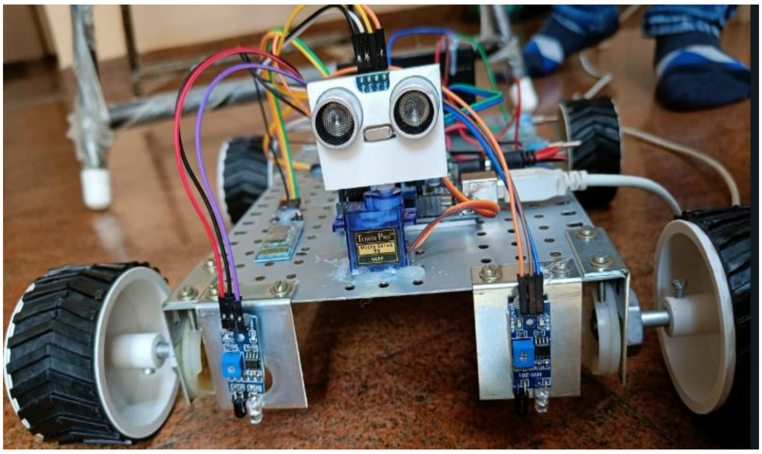


Figure 4.2.7 Sensor placement

5.1 CONCLUSION:

In our project, we created an inexpensive indoor robot that navigates by itself while avoiding obstacles. We have incorporated a number of ideas, including voice assistant, edge detection, obstacle recognition, and leveraging sensor and camera data.

With a 76–80% accuracy rate, we implemented object detection utilising the camera by yolo method and the COCO dataset.

Utilising the idea of a change in sensor data using multiple sensors when an is identified, we have carried out edge detection.

We were able to efficiently control the robot using arduino and simultaneously run obstacle detection using a camera and sensors using the g ROS framework through a literature survey, a comparative study of state-of-the-art implementations and we picked the best method out of them without the necessity of using LIDARS and CLIFF sensors.

To implement voice assistant, we used RPIO.GPIO, Datetime, Pyttsx3, Pyjokes, Wikipedia, and other python libraries.

Our project was completed in a really collaborative approach by researching, analysing, and implementing our answer and our originality on the researched method using the Arduino app and several open source resources.

We gained a better understanding of the functionality and utility of creating a robot that would assist users by concentrating more on the localization and movement of the robot in indoor situations. We also saw how our project might be utilised to alter the direction of localization and 2-Dmapping. Using ROS, Arduino, and a machine learning framework, we have created an effective robot model that serves as a companion for people.

5.2 FUTURE SCOPE:

This robot can be extended to a variety of academic fields. Based on information from the sensors and camera, the robot moves. Every time an object is recognized by the camera, the binary output is recorded in a file.

We will be able to use this data and assist the robot in navigating without any 2-d maps that are built using expensive sensors by integrating capabilities like autonomously moving the robot along the edge, acquiring odometry data and the location of objects, or using QR codes.

The focus of further research will be on using the aforementioned techniques to determine the robot's precise location in order to create a 2-dimensional map that the robot may use to navigate around enclosed spaces. integrating the voice assistant to operate concurrently. establishing a framework to save the information on obstacles that have been observed in relation to where the robot is located utilising distance information from sensors.

6.References:

[1] Zhang Zhaohui, Mei Xuesong, Bian Xu, Cai Hanghang and Ti Jian, "Development of an intelligent interaction service robot using ROS," 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016, pp. 1738-1742, doi: 10.1109/IMCEC.2016.7867516.

[2] R. Huang, J. Pedoeem and C. Chen, "YOLO-LITE: A Real-Time Object Detection Algorithm Optimised for Non-GPU Computers," 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 2503-2510, doi: 10.1109/BigData.2018.8621865.

[3] X. Ruan and X. Xing, "Application of Autonomous Mapping Algorithm on a Desktop Robot System," 2009 Fifth International Conference on Natural Computation, 2009, pp. 448-453, doi: 10.1109/ICNC.2009.156.

[4] Y. Dai, Z. Li and D. Wang, "Design of an indoor map construction and target detection technology platform based on ROS," 2021 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2021, pp. 136-140, doi: 10.1109/YAC53711.2021.9486637

[5] .E. V. Polyakov, M. S. Mazhanov, A. Y. Rolich, L. S. Voskov, M. V. Kachalova and S. V. Polyakov, "Investigation and development of the intelligent voice assistant for the Internet of Things using machine

[6] learning," 2018 Moscow Workshop on Electronic and Networking Technologies (MWENT), 2018, pp. 1-5, doi: 10.1109/MWENT.2018.8337236.

[7] B. Abhishek, S. Gautham, D. Varun Rufus Raj Samuel, K. Keshav, U. P. Vignesh and S. R. Nair, "ROS based stereo vision system for autonomous vehicle," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017, pp. 2269-2273, doi: 10.1109/ICPCSI.2017.8392121.

[8] R. Chinmayi et al., "Obstacle Detection and Avoidance Robot," 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2018, pp. 1-6, doi: 10.1109/ICCIC.2018.8782344.

[9] P. Vashistha, J. P. Singh, P. Jain and J. Kumar, "Raspberry Pi based voice-operated personal assistant (Neobot)," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 974-978, doi: 10.1109/ICECA.2019.8821892.

final_presentaation

ORIGINALITY REPORT

9%	1%	9%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Chengji Liu, Yufan Tao, Jiawei Liang, Kai Li, Yihang Chen. "Object Detection Based on YOLO Network", 2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC), 2018 Publication	4%
2	Harimohan Jha, Vaibhav Lodhi, Debashish Chakravarty. "Object Detection and Identification Using Vision and Radar Data Fusion System for Ground-Based Navigation", 2019 6th International Conference on Signal Processing and Integrated Networks (SPIN), 2019 Publication	3%
3	wiki.ros.org Internet Source	1%
4	Submitted to Cranfield University Student Paper	1%

Design and Implementation of Desktop Pet

Nesara K V
Dept. of Electronics and Communication
PES University
Bangalore, India
nesarakv@gmail.com

Navya N Udupa
Dept. of Electrical and Electronics
PES University
Bangalore, India
udupanavya19@gmail.com

Nesha Kavya Madder
Dept. of Electronics and Communication
PES University
Bangalore, India
neshamadder666@gmail.com

Keshika Shankar
Dept. of Electrical and Electronics
PES University
Bangalore, India
keshikashankar@gmail.com

Shwetha G
Dept. of Electronics and Communication
PES University
Bangalore, India
swethag@pes.edu

***Abstract*—This paper proposes a robust approach and cost-effective approach for obstacle detection and avoidance along with simultaneous edge detection. It utilizes the camera and ultrasonic sensor data. IR sensor is used for edge detection of the desktop. It comprises components, IR sensors, an ultrasonic sensor, a USB camera, and a Motor driver.**

***Keywords*—-ultrasonic sensor,arduino, raspberry pi, Yolo algorithm,**

I. INTRODUCTION

Autonomous robot navigation in outdoor environments is an area of active research. The task requires identifying safe, traversable paths for navigation which allow the robot to progress while avoiding obstacles. The ability to sense the environment through external sensors, such as vision, distance, or proximity sensors is a fundamental requirement of any autonomous system. Distance sensors such as ultrasound and lidar primarily give us the perspective in the plane of the sensor. In order to get it in all directions, these sensors are mounted on moving and rotating platforms that can be controlled. These sensors also have limitations in minimum and maximum distance measurement. Thus, multiple sensors need to be used to get better information. 3D-lidar sensors are capable of scanning the environment in great detail; however, they are expensive and require significant computing resources

Camera sensors, on the other hand, are increasingly used because of their capability to provide more information. A single camera can detect obstacles with a wide field of view. Depth measurement can be achieved with supervised machine learning algorithms. Cameras with IR capabilities allow for obstacle detection even in low light. With improvements in camera sensors, high-resolution sensors are available at reasonable costs. Several concepts have been researched using color images for obstacle avoidance problems, such as edge detection, pixel values, and optical flow.

In vision-based systems, different approaches have been presented to solve the problem of obstacle detection and avoidance. Building a 3D map of the environment using the stereo camera has been researched. Although this method is able to create a robust system, it's computationally intensive and it might require user inputs to guide the system while it's mapping the environment. Nature-inspired approaches, like that of insects being able to detect obstacles, can be implemented efficiently. These models use optical flow, key points, and perspective cues.

This method does not require prior assumptions about the operating environment and it's able to adapt dynamically. Therefore it is suitable for both indoor and outdoor environments.

II. MOTIVATION

1.

The robots have the potential for widespread usage and can be used in various real-world applications such as keeping updated with the information to the user, and companionship. Mimo enables customers to interact with technology in the most natural and intuitive way with their voice. To check time and date and climatic conditions we need to look regularly into mobile instead ask which would respond more quickly than the time consumed in looking into mobile.

To avoid the usage of mobile for small things which usually cause eye effects due to mobile light radiation. Mobile can be a distraction sometimes instead use emo only for important information as it allows you to answer the call and draw other information.

III. METHODOLOGY

Commonly, sensors like laser scanners, sonar sensors, and cameras are used mostly for building a map of the environment.Laser scans can detect the location of a target, environment, or obstacles by transmitting laser beams, but it is very expensive.Our approach to solving the issue is to make use of the ultrasonic sensors and a normal raspberry pi camera to autonomously navigate the robot on the desktop and avoid obstacles and detect edges which helps it from not falling off the desktop edge.

A. WORKFLOW OF OBSTACLE DETECTION USING FUSION OF CAMERA AND ULTRASONIC SENSOR

Methodology for the proposed work has been shown in the below flowchart. As per methodology, targets detected from ultrasonic sensor input data are denoted by tracks with information like the range associated with them. This information is extracted from the sensor. For a camera, the characteristics of the camera are determined by the calibration process. Subsequently, data is collected and processed frame-wise in real time to extract the region of interest accordingly by the YOLOv3 algorithm. Identified Obstacles are bounded with boxes and information on each obstacle is also gathered through it.

For proper fusion of camera and sensor data, internal and external characteristics of the camera need to be determined for obtaining a transformation matrix. The camera calibration is done using ROS packages while determining intrinsic parameters and extrinsic camera parameters.

A conceptual representation of the coordinate system used by the camera and sensor for a target point is illustrated below. (Xc, Yc, Zc) stands for camera coordinates, (Xr, Yr, Zr) for sensor coordinates, and (Xr, Yr, Zr) for image coordinates (u, v). The coordinates of the target are Xr=rsin and Yr=rcos, and the sensor only provides us with the range(r) and angle, which only aids in determining the position of a potential target in a 2D plane. For the intended use, these coordinates must be converted to camera coordinates. As radar returns occur in a 2D plane with the value in the third dimension of coordinates always being 0, this transformation of the sensor's and camera's coordinates is now quite challenging.As seen in the above picture, camera data is obtained in the XrZr plane, whereas sensor returns are obtained in the YrZr plane.

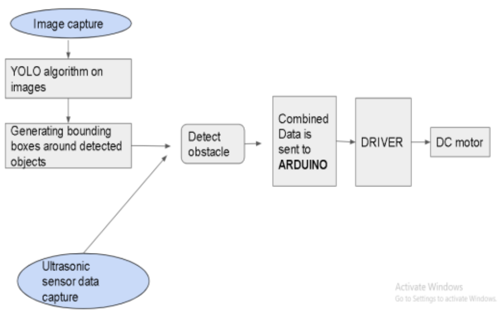


Fig 1. Proposed methodology for vision and radar data fusion for obstacle identification

B. WORKFLOW OF EDGE DETECTION

An active IR sensor is used in this project to sense the edges. It consists of two elements: The IR transmitter and the IR receiver. Basically, the IR transmitter emits Infrared rays, the rays hit an obstacle and bounce back. Based on the intensity of the received signal the object distance is found. The IR sensor we are using in this robot senses the edge. and transfer this information to the microcontroller which is working as a brain in this Arduino robot. The motor is associated with the sensor if there is an edge at the left sensor then the robot will move a little back and move in left or right in any direction. A robot that can automatically balance itself and not fall down from any table or surface can avoid damage. The working of this robot is very simple it simply detects the edges and avoids them. if there is an edge where the robot can fall then from there the robot will. There are two sensors using one sensor at each front corner and another at each cheek. We are using two IR sensors. which sense the edges of both the sensors attached facing down the ground. so they can easily find the edge in the edge-avoiding robot. Ir sensors transmit the infrared light and also received the same. so in this case the IR sensor transmits the light and receives the same from the robot.

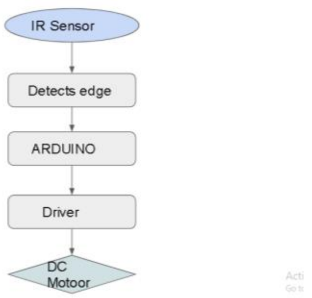


Fig 2. Block diagram of Edge detection

An active IR sensor is used in this project to sense the edges. It consists of two elements: The IR transmitter and the IR receiver. Basically, the IR transmitter emits Infrared rays, the rays hit an obstacle and bounce back. Based on the intensity of the received signal the object distance is found. The IR sensor we are using in this robot senses the edge. and transfer this information to the microcontroller which is working as a brain in this Arduino robot. The motor is associated with the sensor if there is an edge at the left sensor then the robot will move a little back and move in left or right in any direction. A robot that can automatically balance itself and not fall down from any table or surface can avoid damage. The working of this robot is very simple it simply detects the edges and avoids them. if there is an edge where the robot can fall then from there the robot will. There are two sensors using one sensor at each front corner and another at each cheek. We are using two IR sensors. which sense the edges of both the sensors attached facing

down the ground. so they can easily find the edge in the edge-avoiding robot. Ir sensors transmit the infrared light and also received the same. so in this case the IR sensor transmits the light and receives the same from the robot..

C. Control System of Mobile Robot Based using ROS

The top robotics development environment today is ROS. Software developers can design robot applications with the use of the libraries and tools provided by ROS. It offers a variety of features, including hardware abstraction, device drivers, libraries, visualizers, message-passing, and package management. Distributed process framework ROS is contained in usable packages that are simple to share and release. To free us from laborious programming, ROS offers a full suite of robot communication techniques.
Two components make up the entire architecture:
2. When the camera picks up an object.
3. Ultrasonic sensor detects an object or edge.
We can determine if the object is present at a specific location or not by combining these two pieces of information.
Next, command and maneuver the robot to
The bot must now be moved and controlled using UART communication to avoid these items and edges.

D. The ros architecture:
The roscore:

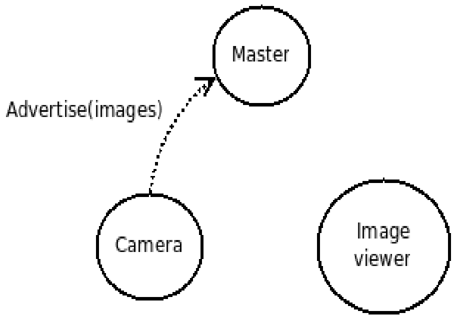
To begin with, let's make sure Roscore is running.
A ROS-based system must have a number of nodes and programs, collectively known as roscore. In order for ROS nodes to communicate, a roscore must be operating. Using the roscore command, it is started.

Roscore will begin:
- a rosout logging node
- a rosout master
- a ros parameter server

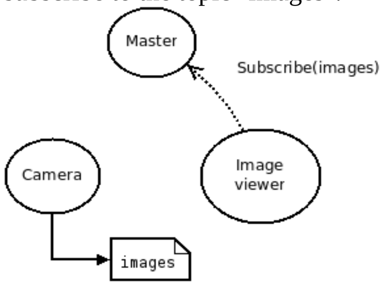
E. *Identify the Headings*

Master: The ROS Master offers name and registration services to all other ROS system nodes. It keeps tabs on both topic and service publishers and subscribers. The Master's job is to make it possible for different ROS nodes to locate one another. Peer-to-peer communication takes place between these nodes once they have located one another.The Parameter Server is likewise supplied by the Master.

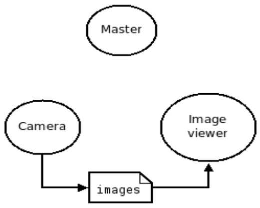
The ROS Master and other crucial components are loaded by the roscore command, which is the most popular way to operate the Master.
Let's take the example of two nodes, a camera, and an image viewer. A typical chain of events would begin with the camera informing the master that it wants to publish photographs on the topic.



Currently, Camera publishes photographs to the "images" topic, but as no one has subscribed to that subject yet, no data is actually transmitted. Now, in order to see whether there are perhaps any photos, the Image viewer wants to subscribe to the topic "images":



The master node alerts the Camera and Image viewer of each other's existence so they can begin exchanging photos now that the topic "images" has a publisher and a subscriber:



Topic: Nodes exchange messages across topics referred to as buses. Information creation and consumption are separated by topics' anonymous publish/subscribe semantics. Typically, nodes are unaware of their communication partners. Instead, nodes interested in data subscribe to the pertinent subject, and nodes producing data publish to the pertinent topic. A topic may have several publishers as well as subscribers. Each subject has its own name in the ROS network, and messages are transmitted on those topics.

Each topic is strongly typed by the ROS message type used to publish to it, and nodes can only accept messages of the same kind. The subscribers will not establish message transport unless the types match, even though the Master does not enforce type consistency among the publishers.
Topic Transports: TCP/IP-based and UDP-based message transmission are both supported by ROS at the moment. Over persistent TCP/IP connections, the TCP/IP-based transport, also known as TCPROS, streams message data. The sole transport that client libraries must support is TCPROS, which is the default transport used by ROS. The UDPROS UDP-based transport divides messages into UDP packets and is only presently supported by roscpp. Since

UDPROS is a lossy, low-latency transport, teleoperation is one of its ideal use cases.
The desired transit is negotiated by ROS nodes in real-time. A node can fall back on TCPROS transport, for instance, if it prefers UDPROS transport but the other node does not support it. As compelling use cases emerge, this negotiation methodology permits the gradual addition of new transports.

F. DESCRIPTION AND IMPLEMENTATION OF OBJECT DETECTING ALGORITHM

One of the common and difficult issues in computer vision is object detection. Researchers have widely experimented and contributed to the performance improvement of object detection and associated tasks including object classification, localization, and segmentation over the course of the last decade thanks to deep learning's rapid evolution. Object detectors can be broadly divided into two groups: single-stage and two-stage object detectors. Single-stage detectors concentrate on all feasible spatial region proposals for object detection via relatively easier architecture in one go, whereas two-stage detectors primarily focus on selected region proposals approach via sophisticated architecture. Any object detector's performance is assessed using inference time and detection accuracy. In general, two-stage detectors perform better in terms of detection accuracy.

G. OBTAINING THE REGION OF INTEREST

ROI pooling takes every ROI from the input and takes a section of the input feature map which corresponds to that ROI and converts that feature-map section into a fixed dimension map. The output fixed dimension of the ROI pooling for every ROI neither depends on the input feature map nor on the proposal sizes, It solely depends on the layer parameters.

H. VOICE ASSISTANT

Using speech recognition as its foundation, the system operates. The Raspberry Pi welcomes the user as soon as it is powered on by the user. The program code-containing Python script is run automatically (auto-start) when the system boots up.

The system receives input through the microphone in, records it until a person indicates to stop and is noticed to indicate that the user has concluded the request, and then converts it to text form. The technique is set up so that after processing the request and looking up the keyword, the system returns the proper textual output, which the speech-recognition module then converts to speech.The speaker connected gives speech output.

IV. CONCLUSION

In this project, we have created an inexpensive indoor robot that navigates by itself while avoiding obstacles. We have incorporated a number of ideas, including voice assistant, edge detection, obstacle recognition, and leveraging sensor and camera data.
With a 76–80% accuracy rate, we implemented object detection utilizing the camera by the Yolo method and the COCO dataset.
Utilizing the idea of a change in sensor data using multiple sensors when an object is identified, we have carried out edge detection.
We were able to efficiently control the robot using Arduino and simultaneously run obstacle detection using a camera and sensors using the g ROS framework through a literature survey, a comparative study of state-of-the-art implementations and we picked the best method out of them without the necessity of using LIDARS and CLIFF sensors.

To implement voice assistant, we used RPIO.GPIO, Datetime, Pyttsx3, Pyjokes, Wikipedia, and other python libraries.

Our project was completed in a really collaborative approach by researching, analyzing, and implementing our answer and our originality in the researched method using the Arduino app and several open-source resources.
We gained a better understanding of the functionality and utility of creating a robot that would assist users by concentrating more on the localization and movement of the robot in indoor situations. We also saw how our project might be utilized to alter the direction of localization and 2-D mapping. Using ROS, Arduino, and a machine learning framework, we have created an effective robot model that serves as a companion for people.

V. FUTURE SCOPE

This robot can be extended to a variety of academic fields. Based on information from the sensors and camera, the robot moves. Every time an object is recognized by the camera, the binary output is recorded in a file.

We will be able to use this data and assist the robot in navigating without any 2-d maps that are built using expensive sensors by integrating capabilities like autonomously moving the robot along the edge, acquiring odometry data and the location of objects, or using QR codes.
The focus of further research will be on using the aforementioned techniques to determine the robot's precise location in order to create a 2-dimensional map that the robot may use to navigate around enclosed spaces.
integrating the voice assistant to operate concurrently.
establishing a framework to save the information on obstacles that have been observed in relation to where the robot is located utilizing distance information from sensors.

VI. ACKNOWLEDGMENT

We would also like to take this opportunity to thank our guide, Prof. Swetha G, for helping us throughout. Also, to our review committee member Mr. Rajshekhar Mohan for invaluable advice and critique. Furthermore, we would like to express gratitude and earnestly acknowledge our parents for supporting us and our friends for encouraging us.

VII. REFERENCES

[1] Zhang Zhaohui, Mei Xuesong, Bian Xu, Cai Hanghang and Ti Jian, "Development of an intelligent interaction service robot using ROS," 2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), 2016, pp. 1738-1742, doi: 10.1109/IMCEC.2016.7867516.

[2] R. Huang, J. Pedoeem and C. Chen, "YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers," 2018 IEEE International Conference on Big Data (Big Data), 2018, pp. 2503-2510, doi: 10.1109/BigData.2018.8621865.

[3] X. Ruan and X. Xing, "Application of Autonomous Mapping Algorithm on a Desktop Robot System," 2009 Fifth International Conference on Natural Computation, 2009, pp. 448-453, doi: 10.1109/ICNC.2009.156.

[4] Y. Dai, Z. Li and D. Wang, "Design of an indoor map construction and target detection technology platform based on ROS," 2021 36th Youth Academic Annual Conference of Chinese Association of Automation (YAC), 2021, pp. 136-140, doi: 10.1109/YAC53711.2021.9486637

[5] .E. V. Polyakov, M. S. Mazhanov, A. Y. Rolich, L. S. Voskov, M. V. Kachalova and S. V. Polyakov, "Investigation and development of the intelligent voice assistant for the Internet of Things using machine

[6] learning," 2018 Moscow Workshop on Electronic and Networking Technologies (MWENT), 2018, pp. 1-5, doi: 10.1109/MWENT.2018.8337236.

[7] B. Abhishek, S. Gautham, D. Varun Rufus Raj Samuel, K. Keshav, U. P. Vignesh and S. R. Nair, "ROS based stereo vision system for autonomous vehicle," 2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI), 2017, pp. 2269-2273, doi: 10.1109/ICPCSI.2017.8392121.

[8] R. Chinmayi et al., "Obstacle Detection and Avoidance Robot," 2018 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), 2018, pp. 1-6, doi: 10.1109/ICCIC.2018.8782344.

[9] P. Vashistha, J. P. Singh, P. Jain and J. Kumar, "Raspberry Pi based voice-operated personal assistant (Neobot)," 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 2019, pp. 974-978, doi: 10.1109/ICECA.2019.8821892.

Design and Implementation of Desktop Pet

Nesha Kavya Madder ,Nesara K V , Navya N Udupa , Keshika Shankar
Project Guide : Asst. Prof. Shwetha G

PROBLEM STATEMENT

The project aims to build a desktop pet which acts like a companion and provides a way of integrating both edge and obstacle using sensor and usb camera data which also acts as an voice assistant .

MOTIVATION

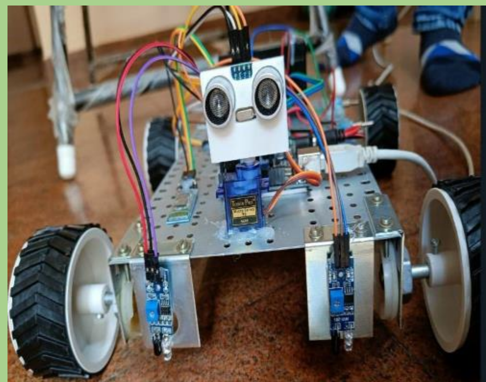
- Autonomous robot navigation in indoor environments is an area of active research.
- The ability to sense the environment through external sensors, such as vision, distance or proximity sensors is a fundamental requirement of autonomous system.

DATASETS AND SOFTWARE REQUIREMENTS

- Coco dataset and Yolo algorithm
- ROS and Arduino IDE

DESIGN APPROACH /METHODS

- Obstacle detection using integration of both ultrasonic sensor and USB camera data.
- Edge detection using IR sensor
- Using ros and yolo algorithm for object detection and avoiding collision.
- Using python for implementation of voice assistant.



RESULTS AND DISCUSSION

- Obstacle avoidance using ultrasonic sensors lets the robot move autonomously
- Robot detects edge and does’nt
  fall off edge
- Object detection using camera was done and data was stored
- Voice assistant interacts with the user.

CONCLUSION AND FUTURE SCOPE

- In this we have created a inexpensive indoor robot that navigates autonomously .using obstacle avoidance and edge detection .
- As it stores the data of the object detection it can be used for the localization of robot