****

**Cleaning Robot with Automatic Docking System**

**Internship Report**

On 2-month internship conducted by

**Robocoupler Pvt. Ltd.**

**Jointly Submitted by:**

1. NAMAN GUPTA (Team Leader)
2. SHRUTI AGRAWAL
3. SAKSHI KOKARDEKAR
4. AAKANSHA KUMARI
5. DHANESH SHUKLA
6. YASH BAHETY
7. MEDA NARENDRA
8. SRIDHAR VEMPATI

# **Introduction**

A robotic vacuum cleaner, sometimes called a robovac or a Roomba as a generic trademark, is an autonomous robotic vacuum cleaner which has a limited vacuum floor cleaning system combined with sensors and robotic drives with programmable controllers and cleaning routines.

**What is a Cleaning Robot?**

A robotic vacuum cleaner, sometimes called a robovac or a Roomba as a generic trademark, is an autonomous robotic vacuum cleaner which has a limited vacuum floor cleaning system combined with sensors and robotic drives with programmable controllers and cleaning routines.

Early designs included manual operation via remote control and a "self-drive" mode which allowed the machine to clean autonomously without human control. Some designs use spinning brushes to reach tight corners, and some include a number of cleaning features along with the vacuuming feature (mopping, UV sterilization, etc.).

More recent models use artificial intelligence and deep learning for better mapping, object identification and event-based cleaning. Marketing materials for robotic vacuums frequently cite low noise, ease of use, and autonomous cleaning as main advantages. The perception that these devices are set-and-forget solutions is widespread but not always correct. Robotic vacuums are usually smaller than traditional upright vacuums, and weigh significantly less than even the lightest canister models. However, a downside to a robotic vacuum cleaner is that it takes an extended amount of time to vacuum an area due to its size. They are also relatively expensive, and replacement parts and batteries can contribute significantly to their operating cost.



Figure 1: A TCM Reinigungsroboter (German cleaning robot)

# **Literature Survey**

In 1956, the American science fiction author Robert A. Heinlein described the concept of a robotic vacuum cleaner with a recharging dock in his novel The Door into Summer: "Basically it was just a better vacuum cleaner .... It went quietly looking for dirt all day long, in search curves that could miss nothing .... Around dinner time it would go to its stall and soak up a quick charge."

In 1969 on 2 April an episode of The Avengers was broadcast in which the character Inge Tilson played by Dora Reisser says "...I saw a demonstration once. A robot vacuum cleaner. It swept around the house, went back into its cupboard, automatically plugged in and recharged itself...". The teleplay for this episode which was entitled "Thingumajig" was written by Terry Nation. It was episode 27 of Season 7.

n 1990, three roboticists, Colin Angle, Helen Greiner, and Rodney Brooks, founded iRobot.It was originally dedicated to making robots for military and domestic use. It launched the Roomba in 2002, which was able to change direction when it encountered an obstacle, detect dirty spots on the floor, and identify steep drops to keep it from falling down stairs. The Roomba proved to be the first commercially successful robot vacuum. In 2005, iRobot introduced the Scooba, which scrubbed hard floors.

In 1996, Electrolux introduced the first “Robotic Vacuum Cleaner”, the Electrolux Trilobite. It worked well but had frequent problems with colliding with objects and stopping short of walls and other objects, as well as leaving small areas not cleaned.As a result, it failed in the market and was discontinued. In 1997, one of Electrolux's first versions of the Trilobite vacuum was featured on the BBC's science program, Tomorrow's World.

In 2001, Dyson built and demonstrated a robot vacuum known as the DC06. However, due to its high price, it was never released to the market. Electrolux Released the Trilobite robotic vacuum cleaner. The Robotic vacuum cleaner launched at a price of $1,800.00. There were two models: The ZA1 and the ZA2

In 2001, Dyson built and demonstrated a robot vacuum known as the DC06. However, due to its high price, it was never released to the market. Electrolux Released the Trilobite robotic vacuum cleaner. The Robotic vacuum cleaner launched at a price of $1,800.00. There were two models: The ZA1 and the ZA2.

In 2010, the Neato Robotics XV-11 robotic vacuum introduced laser-based mapping, allowing navigation in straight lines rather than the traditional random navigation.

In 2015, Dyson and iRobot both introduced camera-based mapping.

In 2016, iRobot CEO claimed that 20% of vacuum cleaner’s sales worldwide were robots.

In 2017, Roborock was launched and sold $1.4 in hours. Later it was Voted Best Robot Vacuum Award 2019 of Trusted Reviews and Best Robot Vacuum you can buy by T3 Magazine UK. Based.

As of 2018, obstacles such as dog faces, cables and shoes remain very difficult for robots to navigate around.

# **Features of Cleaning Robot**

**Cleaning modes**

Robotic vacuum has different types of cleaning modes, usually include the following:

**Auto:** This mode is helpful for general cleaning. Usually, the mode cleans a space until the battery runs out.

* **Spot:** with the help of this mode, the vacuum focus on a particular dirty zone.
* **Turbo:** This mode is used to clean and pick up the most dirt and dust, but it may create noise.
* **Edge:** This mode helps to clean edges & corners.
* **Quiet:** The mode helps to reduce noise levels while cleaning. It's helpful when you are at home.
* **Remote control:** It allows the user to control the direction of the vacuum.

**Wet mopping**

* Some models can also mop for wet cleaning, autonomously vacuuming and wet-mopping a floor in one pass (sweep and mop combo).
* The mop is either manually wetted before attachment to the bottom of the robot or the robot maybe able to automatically spray water on to the floor before running over it.

**Antidrop**

* Most robots include anti-drop and anti-bump [IR sensors](https://en.wikipedia.org/wiki/Passive_infrared_sensor).
* Anti-winding
* When approaching obstacles, will automatically turn away.

**Antitwining**

* Prevents the robot getting twined by wires

**Mapping**

* The first robovac used random navigation. This sometimes caused the unit to miss spots when cleaning or be unable to locate its base station to recharge, and did not provide the user a history of which spaces were cleaned.
* More sophisticated models include mapping ability. The unit can use gyro, camera, radar, and laser ([laser distance sensor](https://en.wikipedia.org/wiki/Laser_distance_sensor) or LDS) guided systems to create a floor plan, which can be permanently stored for more efficiency, and updated with information on areas which have been (or have not been) cleaned. Thus, the cleaning efficiency is greatly improved and the repetition rate is reduced significantly.
* Models with a multiple floor plan feature can store several floor plans.

**Virtual No-Go lines**

* Virtual No-Go lines set boundaries, to restrict the unit's movements to desired cleaning areas.

**Quick recharge**

* Most robot vacuums come with a Lithium-ion battery of around 2000 mAh that will last long enough to handle approximately 200 m2 of floor space (about 100 minutes).
* Regular charge time is 5 to 6 hours.

**Schedule**

* Scheduled daily cleaning. All-Timetable means a full week of different daily schedules can be programmed.

**Connected app**

* Some models allow control of the unit using an app over a WiFi connection, from your smartphone or connected home automation device, e.g. [Amazon Alexa](https://en.wikipedia.org/wiki/Amazon_Alexa) and the [Google Assistant](https://en.wikipedia.org/wiki/Google_Assistant).

**Software upgrades**

* Some units are able to receive [over the air](https://en.wikipedia.org/wiki/Over-the-air_programming) (OTA) firmware updates.

**HEPA Filters**

* The [HEPA](https://en.wikipedia.org/wiki/HEPA) Filters are industry standard now for robot vacuum cleaners.

# **Project**

**What is automatic docking system & how it works?**

If the cleaning robot wants to recharge by itself whenever the battery voltage is low, it must be able to navigate back to the docking area and connect with the docking station automatically. Some key techniques include self-localization, global and local path planning, docking and charging status detection, and fault-tolerant processing. Before reaching the docking area, the robot mainly depends on its own locomotion capabilities to work. In the final docking process, the robot and the docking station work cooperatively to complete the task.

A docking method based on the self-localization of the robot and the infrared detectors of the docking station is proposed. The robot can navigate back to the docking station for recharging operations when the on-board battery is too low.

**Our Approaches: -**

1. **Approach – 1:**

This is an approach to achieve our aim using a Machine Learning Concept. Machine Learning is incorporating human like intelligence in machines, training them to act accordingly by learning from past experiences with the help of various algorithms & mathematical functions. It allows the machine to learn from big data without being explicitly programmed.

In this approach, we tried to train our bot virtually by using a software MS LOBE. It trains the model using the concept of reinforcement learning, where we as a user provides feedback to every action or result predicted by it. We provided the model a set of images - of a blank room and other of charging points, and trained it to differentiate between them.

We also used various test cases and when we got 100% accuracy of our model, we derived the code in Python for it (Code is written in the previous slide).

**Code:**

import base64

import requests

img\_filepath = "path/to/image.jpg"

with open(img\_filepath, "rb") as f:

image\_string = base64.b64encode(f.read()).decode("utf-8")

response = requests.post(

url="http://localhost:38101/v1/predict/6ebce86e-72ef-48af-b1b2-444063aac049",

json={"image": image\_string}, )

data = response.json()

top\_prediction = data["predictions"][0]

print("predicted label:\t{}\nconfidence:\t\t{}"

.format(top\_prediction["label"], top\_prediction["confidence"]))

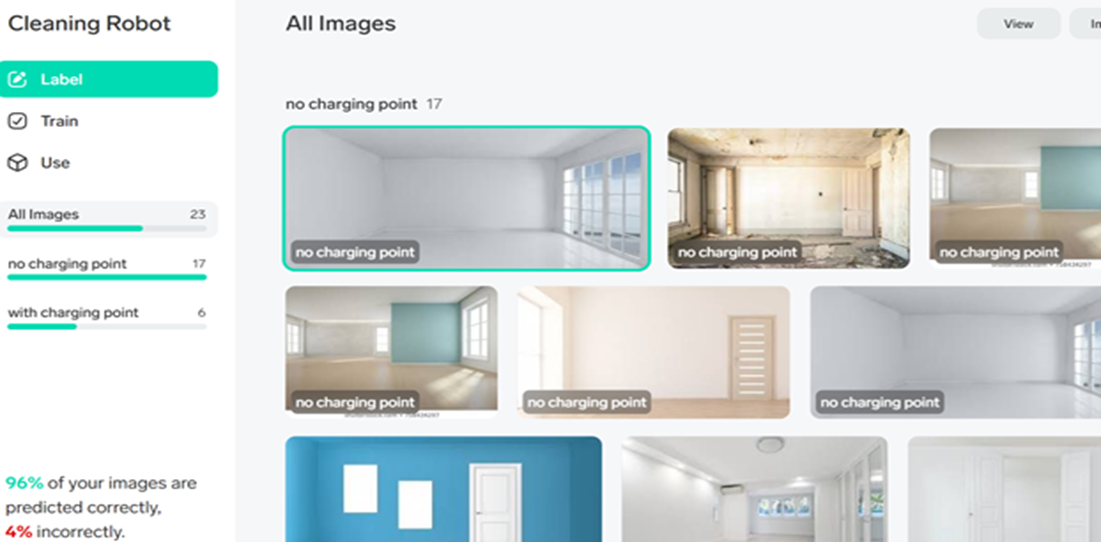


Figure 2: Dataset Given

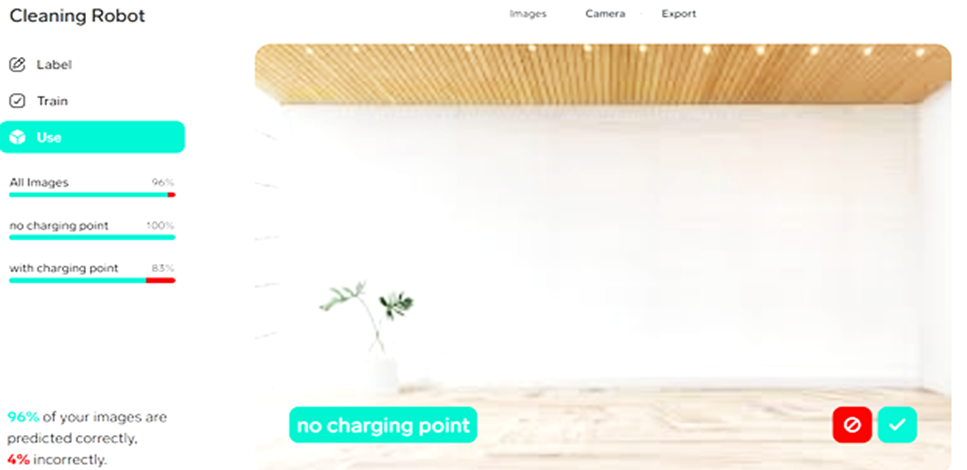


Figure 3: Output of Approach 1

1. **Approach – 2: -**

**Algorithmic Approach:**

Algorithms can be used to find and remember the path to the docking system.

The bot will follow a specific pattern to avoid obstacle and memorise the path in its memory.

There are various searching algorithms available such as A\*, Breadth First Search, Depth First Search

Best Algorithm here is Depth First Search.

In Depth first Search is an algorithm for traversing or searching tree or graph data structures. The algorithm starts at the root node (selecting some arbitrary node as the root node in the case of a graph) and explores as far as possible along each branch before backtracking.

**Code:**

**from PIL import Image, ImageDraw**

**images = []**

**a = [**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**[1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0 ,0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0 ,0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 0 ,0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0 ,0, 0, 0, 1, 0, 1, 1, 1, 1],**

**[1, 0, 1, 1, 1, 1, 1, 1, 0, 1, 0 ,0, 0, 0, 1, 0, 0, 0, 0, 0],**

**[1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0 ,0, 0, 0, 1, 1, 1, 1, 1, 1],**

**[1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0 ,0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 0 ,0, 0, 0, 0, 0, 0, 0, 0, 1],**

**[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1],**

**]**

**zoom = 80**

**borders = 6**

**start\_i, start\_j = 1,1**

**end\_i, end\_j = 5,19**

**def draw\_matrix(a, the\_path=[]):**

**im = Image.new('RGB', (zoom \* len(a[0]), zoom \* len(a)), (255, 255, 255))**

**draw = ImageDraw.Draw(im)**

**for i in range(len(a)):**

**for j in range(len(a[i])):**

**color = (255, 255, 255)**

**r = 0**

**if a[i][j] == 1:**

**color = (0, 0, 0)**

**if i == start\_j and j == start\_j:**

**color = (0, 255, 0)**

**r = borders**

**if i == end\_i and j == end\_j:**

**color = (0, 255, 0)**

**r = borders**

**draw.rectangle((j\*zoom+r, i\*zoom+r, j\*zoom+zoom-r-1, i\*zoom+zoom-r-1), fill=color)**

**if a[i][j] == 2:**

**r = borders**

**draw.ellipse((j \* zoom + r, i \* zoom + r, j \* zoom + zoom - r - 1, i \* zoom + zoom - r - 1),**

**fill=(128, 128, 128))**

**for u in range(len(the\_path)-1):**

**y = the\_path[u][0]\*zoom + int(zoom/2)**

**x = the\_path[u][1]\*zoom + int(zoom/2)**

**y1 = the\_path[u+1][0]\*zoom + int(zoom/2)**

**x1 = the\_path[u+1][1]\*zoom + int(zoom/2)**

**draw.line((x,y,x1,y1), fill=(255, 0, 0), width=5)**

**draw.rectangle((0, 0, zoom \* len(a[0]), zoom \* len(a)), outline=(0, 255, 0), width=2)**

**images.append(im)**

**path\_so\_far = []**

**def go\_to(i, j):**

**global path\_so\_far, end\_i, end\_j, a, m**

**if i < 0 or j < 0 or i > len(a)-1 or j > len(a[0])-1:**

**return**

**if (i, j) in path\_so\_far or a[i][j] > 0:**

**return**

**path\_so\_far.append((i, j))**

**a[i][j] = 2**

**draw\_matrix(a, path\_so\_far)**

**if (i, j) == (end\_i, end\_j):**

**print("Found!", path\_so\_far)**

**for animate in range(10):**

**if animate % 2 == 0:**

**draw\_matrix(a, path\_so\_far)**

**else:**

**draw\_matrix(a)**

**path\_so\_far.pop()**

**return**

**else:**

**go\_to(i - 1, j) # check top**

**go\_to(i + 1, j) # check bottom**

**go\_to(i, j + 1) # check right**

**go\_to(i, j - 1) # check left**

**path\_so\_far.pop()**

**draw\_matrix(a, path\_so\_far)**

**return**

**go\_to(start\_i, start\_j)**

**images[0].save('result.gif',**

**save\_all=True, append\_images=images[2:],**

**optimize=True, duration=200, loop=0)**

1. **Approach – 3: -**

**Sensors and Bluetooth:**

LIDAR Sensor: LIDAR (Light Detection and Ranging) is an optical remote sensing system which can measure the distance of a target by illuminating it with light. LIDAR technology is being used in Robotics for the perception of the environment as well as object classification.



Figure 4: LIDAR Sensor

**Code:**

import threading

import PyLidar3

import matplotlib.pyplot as plt

import math

import time

def draw():

global is\_plot

while is\_plot:

plt.figure(1)

plt.cla()

plt.ylim(-9000, 9000)

plt.xlim(-9000, 9000)

plt.scatter(x, y, c='r', s=8)

plt.pause(0.001)

plt.close("all")

is\_plot = True

x = []

y = []

for \_ in range(360):

x.append(0)

y.append(0)

port = input("Enter port, which lidar is connected:")

Obj = PyLidar3.YdLidarX4(port)

threading.Thread(target=draw).start()

if(Obj.Connect()):

print(Obj.GetDeviceInfo())

gen = Obj.StartScanning()

t = time.time() # starting time

while (time.time() - t) < 30: # scan for 30 seconds

data = next(gen)

for angle in range(0, 360):

if(data[angle] > 1000):

x[angle] = data[angle] \* math.cos(math.radians(angle))

y[angle] = data[angle] \* math.sin(math.radians(angle))

is\_plot = False

Obj.StopScanning()

Obj.Disconnect()

else:

print("Error connecting to device")

1. **Approach – 4: -**

**RFID Technology:**

Radio-frequency identification uses electromagnetic fields to automatically identify and track tags attached to objects. An RFID system consists of a tiny radio transponder, a radio receiver and transmitter. This was one of our initial approaches, but as this was not economical i.e., cost effective, this technology was not used.

* RFID Tags can be used at the entrance of the room in which the robot is entering.
* Robot at first scans the RFID tag and accordingly remembers its location.
* This RFID tag acts like a checkpoint for the robot.
* Whenever there is low battery, the robot may trace its path back to the entrance of the room in which it entered last.
* After returning to the checkpoint, the robot may trace its path back to docking station.

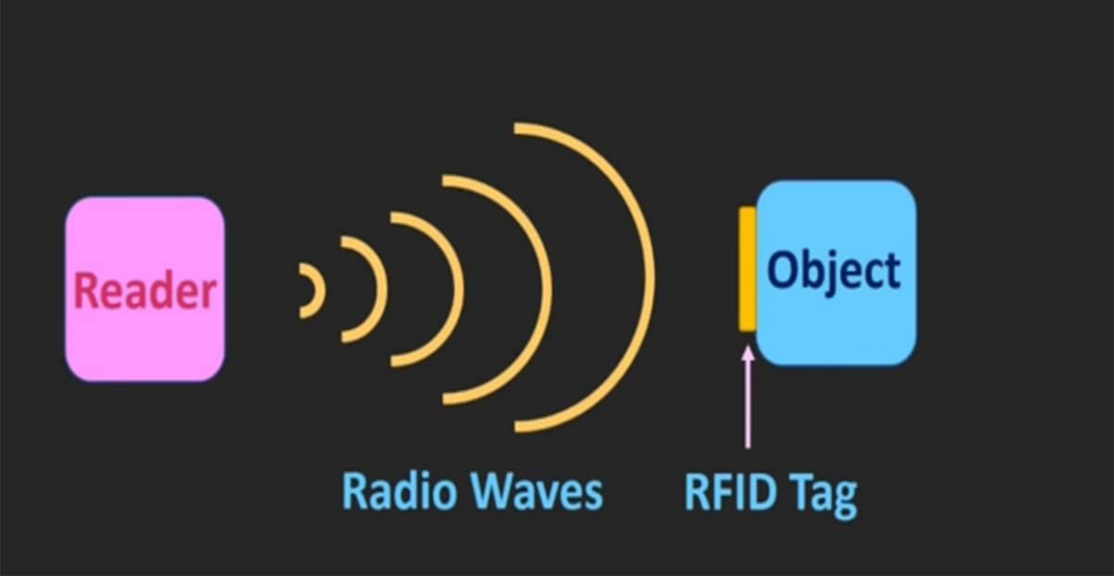


Figure 5: RFID Sensor Working

**How will our robot navigate?**

Our robot has previously mapped the entire area of home and now has details(data) of positioning of various objects in house and positioning of charging stations among various rooms.

**What can be used in our Docking System?**

* Cameras with built-in object recognition technology for a 360-degree view of the surroundings
* Motion sensors that identify nearby objects and can alert the system to adjust or change course
* GPS units to assist in navigating the bot into the appropriate docking space based on coordinates or a specific location
* Newer systems include FLIR (Forward-Looking Infrared) cameras that analyze real-time images and incorporate that data into the propulsion system to guarantee seamless manoeuvring, even in the tightest spaces.

**HOW CLEANING BOTS ARE SAVIOURS DURING COVID-19 PANDEMIC?**

As the pandemic highlights the importance of cleaning and disinfecting, demand for autonomous cleaning robot has increased.

Society is waking up to the fact that cleaning is a dangerous job right now. It is not just unpleasant; it's actually dangerous. Heads of governments are referring to cleaning staff in the same sentence as nurses and doctors. Cleaning robots are being deployed in hospitals around the world and since the COVID-19 crisis demand for disinfection has increased.

Users can add disinfectant to the robot’s clean water tank. Institutions and businesses want cleaning and disinfecting to happen three times a day, rather than once every three days.

Autonomous cleaning has gone from being an efficiency issue to a safety issue. Robots are keeping frontline cleaners safe from exposure.



Figure 6: Automatic Cleaning Robots in Hospitals

These robots are increasingly advocated as a simple solution for the immediate disinfection of rooms and spaces of all surfaces in one process and as such they seem attractive to hospital management, also because of automation and apparent cost savings by reducing cleaning staff. Presently, disinfection robots do not replace routine (manual) cleaning but may complement it.

# **CONCLUSION**

In the near future the cleaning Robot will be a must have appliance for the home. The bots will become more advance and the fittest will survive the market. The self-charging system is the primary step to this advancement. Automatic Docking System is must for the cleaning Robots. Now the robots are designed to think like a Human. This is Possible with the help of Machine Learning and AI. We have created here an AI model which can easily differentiate between rooms and find its docker. This will also help robot to identify the various parts of the room. With this the bot can be easily directed to operate in a particular area of the house.

# **Acknowledgement**

We would like to thank Robocoupler Pvt. Ltd for giving us this golden opportunity to serve our candidature as interns. We have learnt a lot in these 45-days of internship.

We would also like to thank our mentor for giving us this opportunity to work on such an innovative project. Your support, insights and guidance has helped our team in the completion of this project. Thank you for being such an amazing mentor and for always motivating us.