Social Distancing and Monitoring Robot

Abhishek Saseendran[1], Jithu Subhash[2], Khushitha Anand[3], Neha Mohan Khade[4], Priyadarshini M[5]

[1, 2, 3, 4]*UG Student, Department of Computer Science and Engineering, Cambridge Institute of Technology, Bangalore, Karnataka, India*
[5]*Assistant Professor, Department of Computer Science and Engineering, Cambridge Institute of Technology, Bangalore, Karnataka, India*

Abstract - The coronavirus disease (COVID-19 pandemic) has become a global threat faced by people all around the world. Social distancing has proved to be an effective practice to curb the spread of COVID-19. In this context, we propose a 4-wheeled robot for queue monitoring and by which we can encourage social distancing. The robot automatically detects pairs of humans in a queue who are not adhering to the social distance constraint, i.e., about 2 feet distance between them by using an ultrasonic sensor to measure the distance between the two individuals. If social distancing constraints are violated, the robot alerts the individuals to maintain distance between them. In addition, we also equip the robot with a camera that wirelessly transmits images to a security/healthcare personnel who monitors if any two individuals are violating social distancing measures.

Key Words: IoT, Social Distancing, Queue monitoring.

1. INTRODUCTION

COVID-19 is a contagious disease caused by SARS-CoV-2. The main reason for the spread of COVID-19 is not maintaining social distancing. Social distancing or physical distancing is a set of non-pharmaceutical measures which intend to prevent the spread of a contagious disease by maintaining a physical distance of 2 to 3 feet between people and reducing the number of times people come into close contact with each other. Social distancing is a key factor during the current pandemic. It helps to limit the spread of COVID-19. Social distancing must be practiced by every individual in order to curb the spread of the disease. Now it is not possible to station a person and monitor social distancing at each queue. Banks, Public Offices, Malls, Schools, Theatres, and other public places usually see long queues for hours every day. To enforce and encourage social distancing we hereby design a social distancing and monitoring robot. The robot achieves this by observing the distance between disease-spreading individuals. The robot uses an ultrasonic sensor for detecting the distance between individuals in a queue. If any two individuals are found to have less than 2 feet of distance between them, the robot instantly sounds an alert to inform about the violation. The robot uses a line following algorithm to follow the queue. It has a speaker which is used to give alerts to the crowd. Also, it sends alerts of these violations along with a camera picture using Wi-Fi over IoT to inform the higher authorities/head office to update them about violations with proof, so that instant disciplinary action can be taken.

2. METHODOLOGY

2.1 Line following Algorithm

Line sensors emit infrared (IR) light and detect the presence of a black line underneath the robot. The line sensor detects the light levels that are reflected from the surface underneath it and returns them to the sensor. This is done by using two components, an emitter and a light sensor which is a receiver. These devices also have a component called a potentiometer which is a position sensor that adjusts the device's position based on its threshold. The outputs received are of two types for some line sensors, which are in analog and digital form. The analog output returns a constant reading of the light levels the sensor is detecting but is not always present. These outputs generated will be sent to Raspberry Pi and can be used only after the conversion of analog signals to digital form. The comparison of the light levels against a threshold level results in a digital output. Based on those results the potentiometer is adjusted by turning accordingly. The digital output will be high (1) if the sensor does not receive enough light to surpass the threshold value. The pin will be set to low (0) if enough light is received and the threshold value is surpassed. A black line won't reflect much light, so the output will be set to high (1) when a black surface is underneath.

The line-following robot[1] can be created using many sensors ranging from one to five or more and there are demerits too which could be overcome. A Line-following robot can have only one sensor, as this is enough to detect a line underneath the robot, but it is very difficult to find the line again and work out which way it needs to turn. To solve this problem, more than one sensor is used. If a second sensor is added to the robot and placed evenly on either side of the caster wheel, the information gathered from the sensors will allow the robot to find the line again. If the line beneath the robot is at the center, both sensors will return a (0) from their digital pins, as they are both over the white background. When the robot moves to the right, the left sensor will eventually cross onto the line, changing its output to a (1). When this problem occurs, the robot should correct its course and turn left to keep up with the line. In the same way, when the robot moves to the left, the right sensor will eventually cross onto the line, changing its output to a (1). When this happens, the robot should correct its course and turn right. Once the sensor returns the output value as (0), that determines that the line is back in the middle of the robot.

2.2 Ultrasonic Distance Measurement

The most common type of distance measuring sensor is the Ultrasonic Sensor, also known as the Sonar sensor, it detects the distance to objects by emitting high-frequency sound pulse which is ultrasonic waves, and calculates the distance based on the time taken by the echo signal to travel back after reflecting from the desired target. The basic principle is an echo for ultrasonic distance measurement[2]. High-frequency sound waves reflect from objects to produce distinct echo patterns. Ultrasonic sensors work by sending out a sound wave at a frequency greater than 330ms-1, which is above the range of human hearing. The transducer of the sensor acts as a microphone to receive and transmits ultrasonic sound waves that relay back information about an object's proximity. Ultrasonic sensors can measure and calculate the distance to a wide range of objects. The time difference between sending and receiving the sound pulse is used to determine the distance to an object by ultrasonic sensor. Distance is calculated by: Time x Speed of Sound divided by 2. The working principle of this module is simple. When the ultrasonic sound wave is sent out, it travels through the air and if it detects an obstacle or an object, it will bounce back to the sensor. The distance can be calculated by calculating the travel time and the speed of sound. Ultrasonic sensors can avoid collisions[3]. Ultrasonics are best used in the non-contact detection of presence, level, position, and distance and are so widely used as they are superior to infrared sensors because they aren't affected by smoke or black materials, light, dust, color; however, soft materials which don't reflect the sonar (ultrasonic) waves very well may cause issues but it's good and reliable.

2.3 IoT over Wifi

The Internet of Things (IoT) refers to millions and billions of devices, each having a unique identifier called the IP Address, that are connected via the Internet, all collecting and sharing data among each other. It is now possible to turn any electronic machine into a part of the IoT. The devices that are connected as part of the IoT, can transfer data without having a human-to-human or a human-to-computer interaction. The Raspberry Pi 3 Model B+[4] comes with an inbuilt 2.4GHz and 5GHz IEEE 802.11 b/g/n/ac Wireless LAN Module which specifies the implementation of wireless local area network (WLAN) for computer communication.

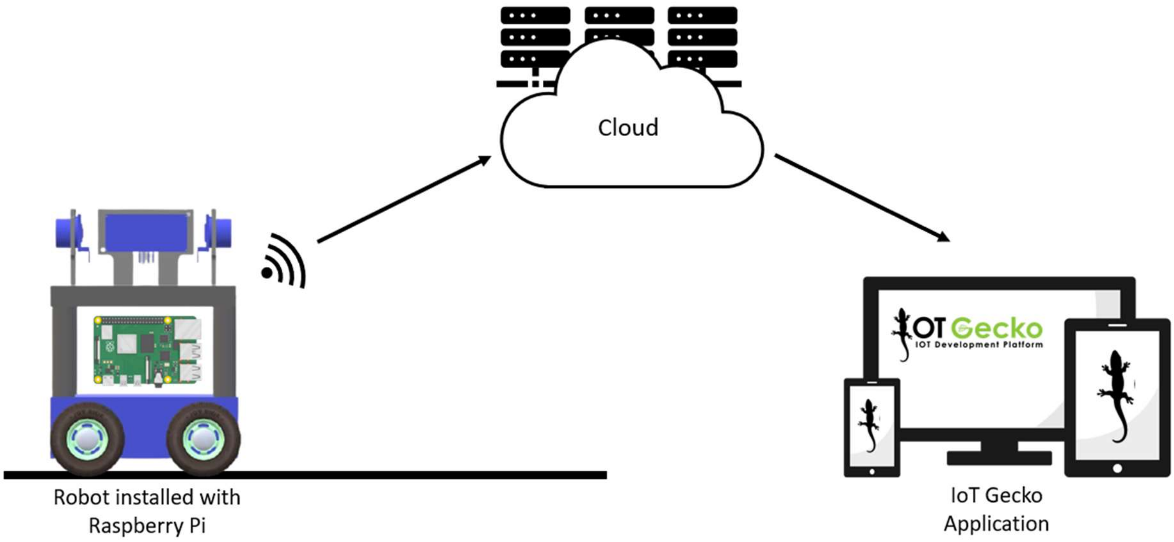


Fig- 1: Robot as part of the IoT

We make use of the cloud platform called IoT Gecko[5]. IoT Gecko is a development platform for the Internet of Things that provides easy-to-use GUI Builders and API Support over Arduino, Raspberry Pi, microcontrollers and other controller boards. It also provides templates to build our own IoT system to handle/debug and monitor it on the web using mobile or desktop devices that support a web browser. The social distancing violation images clicked by the robot are sent to the IoT Gecko cloud platform over Wifi where the concerned authorities can view the images.

3. MODELING AND ANALYSIS

3.1 COMPONENT DETAILS

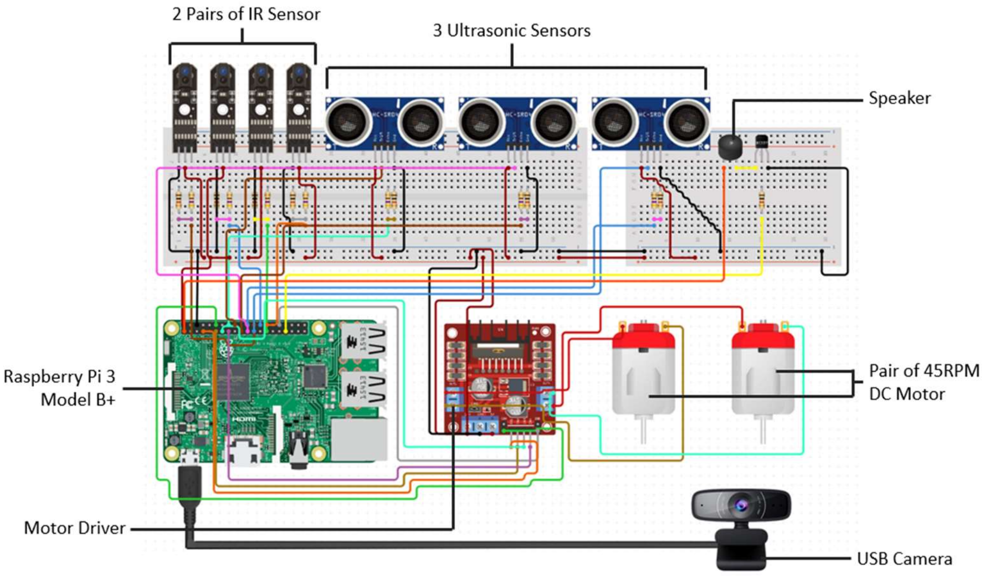


Fig- 2: Primary components and circuit design

3.1.1 Raspberry Pi 3 Model B+

Raspberry Pi is an ARM cortex-based popular single-board microcomputer. We use the Raspberry Pi 3 Model B+[4] which is the latest development board from Raspberry Pi. The microcomputer has a fast 64-bit 1.4GHz quad-core chip, 1GB of RAM, a 300Mbit/s ethernet, dual-band 802.11 wireless LAN, and Bluetooth 4.2 connectivity. It has improved thermals on the Pi 3 B+ meaning that the CPU on the BCM2837 SoC can run at 1.4GHz, which is a 17% increase from the previous Raspberry Pi model.

3.1.2 IR Sensor Pair

Infra-Red Sensor commonly known as the IR Sensor Pair is a combination of an Infrared source emitter and a light detector known as a photodiode and it works on the principle of transmitting light and receiving it and comparing the received amount of light with the help of a comparator LM358. Here we use 2 pairs of IR sensors on the front and back of the robot to give the ability to detect lines using a line following algorithm[1].

3.1.3 Ultrasonic Sensors

The Ultrasonic sensor is a popular sensor used in applications where measuring distance[2] or sensing objects are required. The HC-sr04 module has two components, the Ultrasonic transmitter, and Receiver. The Ultrasonic transmitter transmits an ultrasonic wave, this wave travels in medium, and then it gets reflected toward the sensor. This reflected wave is detected by the receiver module. The amount of time during which the Echo pin stays high gives the data about the time taken for the wave to return to the Sensor. Using this data and the speed of ultrasonic sound waves (330ms-1) the distance is measured. Here we use 3 ultrasonic sonic sensors, 2 of which are used for collision avoidance[3], and the other is used to measure the distance between two individuals in the queue.

3.1.4 USB Camera

We use ZEB-Crisp Pro which is an HD web camera with a 5MP lens having a resolution of 1280x720. The USB web camera comes with an auto white balance, a built-in microphone, a night vision feature using a switch for LED. We use OpenCV[6] Python module to click a picture of the two individuals in the queue where social distancing violation is detected. The picture is later sent to the cloud where the admin/healthcare personnel can view it and take necessary actions.

3.1.5 Speaker

A 12Ω speaker is installed into the robot to output digital audio files using the PyGame Python module. The speaker module is used to alert the crowd whenever the robot detects a breach in social distancing constraint by utilizing a digital audio file that outputs "Alert! Please maintain social distancing".

3.1.6 Motor Driver

Motor drivers behave as amplifiers since they supply a higher-current signal and also take a control signal. We use an L293D Motor Driver that contains two inbuilt driver circuits. Using this motor driver, two DC motors can be driven both in the forward and reverse direction simultaneously.

3.1.7 45RPM DC Motor

We use a 45 RPM 12V DC Geared Motor which is a high-quality low-cost DC geared motor. It provides a torque of about 2.7kgcm. These precision gear motors are extremely powerful and feature full metal gears to help drive gears, wheels, or almost anything else that needs to turn. We use a 45RPM motor because we require the robot to traverse the queue at a slow pace.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 06 | June - 2021 ISSN: 2582-3930

3.2 SYSTEM DESIGN

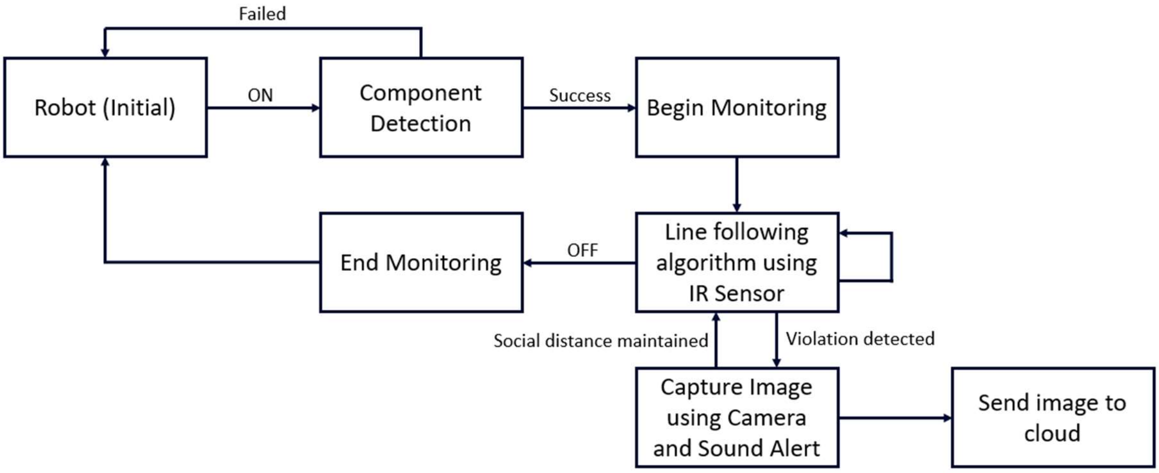
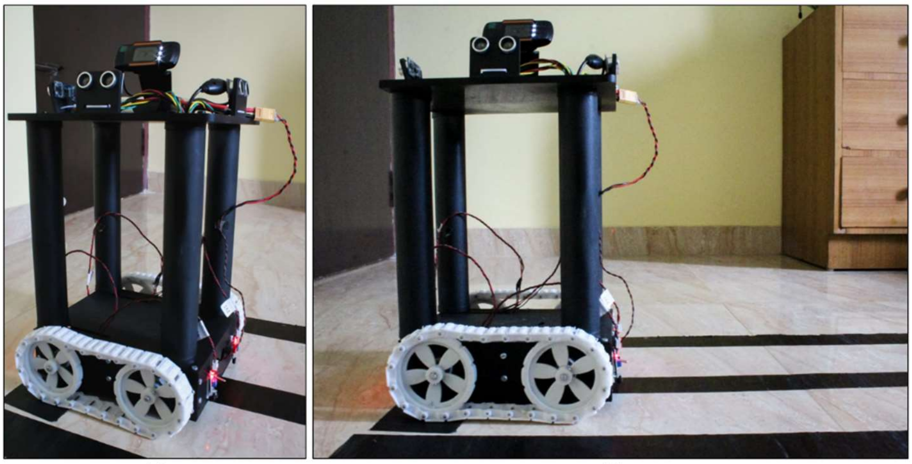


Fig- 3: System design

The system design (shown in Figure 3.) provides complete information about the various stages of the robot monitoring life cycle. When the robot is turned on, it checks whether the camera is enabled and if it has an active internet connection to the IoT Gecko cloud platform. If this stage of the system design is successful, the robot can begin monitoring with a press of a switch embedded in the robot. The robot then moves along the track using a line following algorithm[1] and monitors the queue for social distance violations. If a violation is detected, the robot alerts the constraint violating individuals to maintain social distance. It also clicks a picture of the individuals and sends it over the internet to the IoT Gecko platform[5] where concerned authorities can view it and take action. Once the individuals adhere to social distancing policies, the robot continues to monitor the queue and repeat the process until it is manually turned off.

3.3 IMPLEMENTATION



(a) (b)

Fig- 4: Robot as seen from perspective view (a) and side view (b)

The robot is installed with all the required components (from section 3.1). The Raspberry Pi 3 Model B+[4], the speaker, and the DC Motor Driver are safely installed inside the body of the robot. 4 wheels are fitted on 2 sides of the robot and 2 of the wheels are screwed to the pair of 45RPM DC Motors that are connected to the DC Motor Driver. The other two wheels are moved using belts that are linked to the wheels screwed to the motor. The track the robot moves on is made using 3 parallel black colored lines (as seen in Figure 4. (b)) that run along the length of the queue that is to be monitored. The start and end of the queue are marked by black colored lines, that are kept perpendicular to the queue. The 2 pairs of IR Sensors are placed on the front and back of the robot. The distance between the IR Sensors in each of the pairs should be greater than the width of the middle black colored track. The robot can now use the line following algorithm[1] to traverse along the queue. The 3 Ultrasonic sensors are fit on top of the robot. 2 of the ultrasonic sensors face the front and back of the robot, this enables the robot to detect obstacles[3] in the path the robot is traveling on. The third ultrasonic sensor is placed facing the queue to measure the distance between the individuals in the queue. A USB camera is mounted on top of the robot that faces the queue. The camera is triggered[6] to click a picture when the ultrasonic sensor that is used to measure the distance between individuals in the queue, calculates the distance to be less than 2 feet.

4. RESULTS AND DISCUSSION

Case 1: Line following algorithm

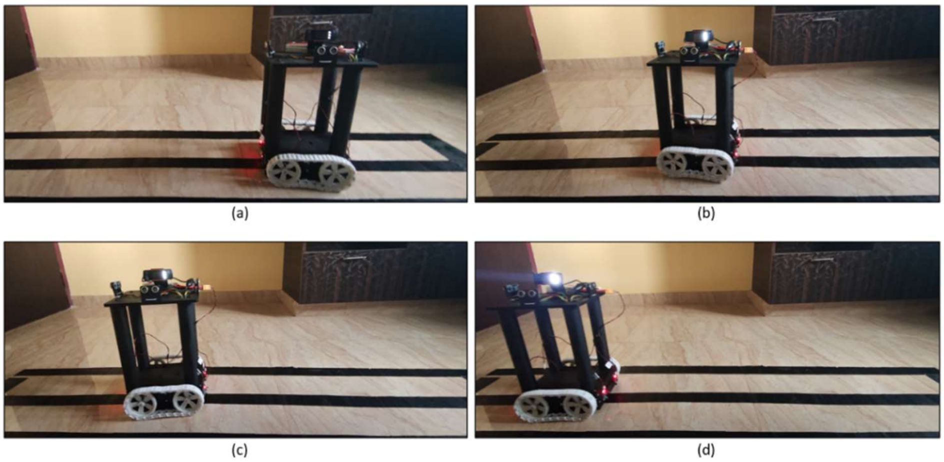


Fig- 5: Robot traversing the track using line following algorithm

The robot successfully incorporates the line following algorithm using the IR sensors to follow the queue on a black colored track, which detects the line and sends information to the Raspberry Pi. It takes appropriate action by changing the direction of the wheels by the motor drivers (as seen in Figure 5. (b) and (c)) in case the robot accidentally runs over the track. When the robot reaches one end of the queue which is marked by a perpendicular black colored line, one pair of the IR sensors detect no incoming signal, and the robot shifts its direction and travels to the other end of the queue marked by another perpendicular black colored line.

Case 2: Distance measurement between individuals in the queue

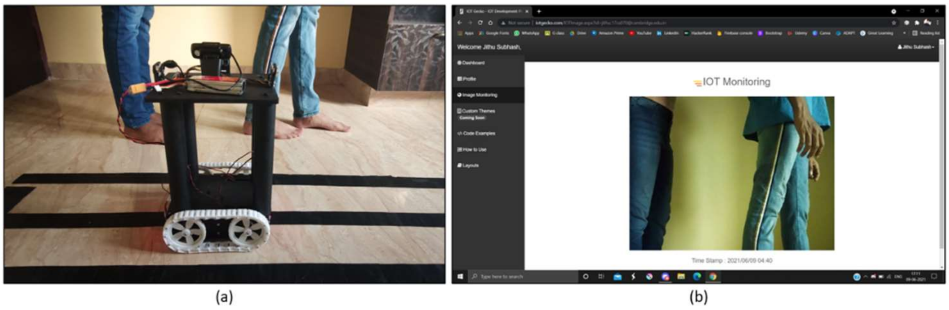


Fig- 6: Distance measurement and IoT Gecko platform

Successful detection of objects and social distancing is achieved by the use of an ultrasonic sensor based on the measured propagation time of the ultrasonic signal. If a violation occurs in maintaining social distance, i.e., 2 individuals who have less than 2 feet distance between them (as seen in Figure 6. (a)), then a speaker output is given as "Alert! Please maintain social distancing" and an image of the individuals violating social distancing is sent to admin by the means of the internet to the IoT Gecko cloud platform[5] (Figure 6. (b)).

Case 3: Detecting and avoiding obstacles in the path

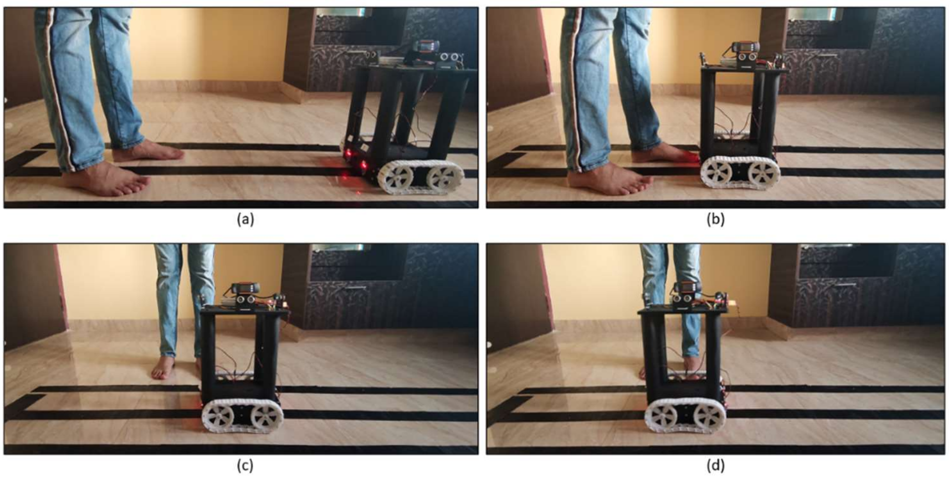


Fig- 7: Obstacle detection and avoidance

If an obstacle that is less than 25cm from the robot is detected in the track the robot traverses (as seen in Figure 7. (b)), the robot stops moving and speaker output is given as "Excuse me, please give me the way". Once the obstacle is removed (as seen in Figure 7. (c)), the robot continues monitoring the queue (Figure 7. (d)).

Discussion

We have implemented the robot with 2 pairs of IR sensors on either side of the robot and 3 ultrasonic sensors having a practical distance measurement of about 80cm each. For real-time implementation, further improvements can be added to the robot such as IR sensor arrays and better ultrasonic sensors to measure larger distances. An additional ultrasonic sensor can be used so that the robot will not be limited to monitoring only a single-lined queue, but will also be able to monitor double-lined queues. The upgraded robot can also be equipped with a 4-wheel drive mechanism using 2 motor drivers and taller robotic chassis so that the USB camera can be at a height of the average human.

5. CONCLUSION

In the context of the COVID-19 pandemic, we present a method to detect violations in social distancing norms in indoor scenes using a queue monitoring robot. The robot mainly uses a line following algorithm to move back and forth the length of the queue, and an ultrasonic-based obstacle detection and avoidance mechanism. The experimental results demonstrate that our robot successfully helps keep social distance between individuals in the queue and thereby curb the spread of the virus. As discussed in the previous section, further improvements can be made to the robot to better improve its accuracy and flexibility. Apart from the application in response to the COVID-19 pandemic, the robot can also be used for queue maintenance such as counting the number of people in a queue, or as a surveillance robot for theft monitoring around premises.

REFERENCES

[1] Nakib Hayat Chowdhurya, Deloara Khushi, Md. Mamunur Rashid, Algorithm for Line Follower Robots to Follow Critical Paths with Minimum Number of Sensors, International Journal of Computer (IJC) Volume 24, No 1, pp 13-22, 2017

[2] JohnGlendale Eden, Elijah Carl Telan, Jim Kyle Pancho, Ultrasonic Distance Measuring Device Study, ResearchGate, April 2019

[3] Mohanad Abdulhamid, Otieno Amondi, Collision Avoidance System Using Ultrasonic Sensor, ResearchGate, September 2020

[4] Anand Nair, Vikram Puri, Raspberry Pi-A Small, Powerful, Cost-Effective and Efficient Form Factor Computer: A Review, ResearchGate, December 2015

[5] IoT Gecko: IoT Development Platform http://iotgecko.com/

[6] Ivan Culjak, David Abram, Tomislav Pribanic, Hrvoje Dzapo, Mario Cifrek, A brief introduction to OpenCV, 2012 Proceedings of the 35th International Convention MIPRO, IEEE Conferences, 2012

BIOGRAPHIES

Abhishek Saseendran
Undergraduate Student,
Department of Computer Science and Engineering,
Cambridge Institute of Technology,
Bangalore

Jithu Subhash
Undergraduate Student,
Department of Computer Science and Engineering,
Cambridge Institute of Technology,
Bangalore

Khushitha Anand
Undergraduate Student,
Department of Computer Science and Engineering,
Cambridge Institute of Technology,
Bangalore

Neha Mohan Khade
Undergraduate Student,
Department of Computer Science and Engineering,
Cambridge Institute of Technology,
Bangalore

Priyadarshini M
Assistant Professor,
Department of Computer Science and Engineering,
Cambridge Institute of Technology,
Bangalore