Autonomous Obstacle Avoidance Robot Using Regression

Naveen Vakada, Aasish Chunduri, Kavya Manne,
Vidhya Lakshmi Meda and Kl Sailaja

June 9, 2020

AUTONOMOUS OBSTACLE AVOIDANCE ROBOT USING REGRESSION

Vakada Naveen[0000-0002-0274-3217] ,Chunduri Aasish[0000-0002-8363-3393], Manne Kavya[0000-0002-9198-5436], Meda Vidhyalakshmi[0000-0003-3397-2220], Kl Sailaja[0000-0001-9347-9142]

[0000-0002-0274-3217] VR SIDDHARTHA ENGINEERING COLLEGE, KANURU 520007, INDIA
vakadanaveen@gmail.com
[0000-0002-8363-3393] VR SIDDHARTHA ENGINEERING COLLEGE, KANURU 520007, INDIA
ch.aasish999@gmail.com
[0000-0002-9198-5436] VR SIDDHARTHA ENGINEERING COLLEGE, KANURU 520007, INDIA
kavyamanne69@gmail.com
[0000-0003-3397-2220] VR SIDDHARTHA ENGINEERING COLLEGE, KANURU 520007, INDIA
vidhyalakshmimeda@gmail.com
[0000-0001-9347-9142] VR SIDDHARTHA ENGINEERING COLLEGE, KANURU 520007, INDIA
sailaja0905@gmail.com

Abstract. Obstacle avoidance is considered as one of the main features of autonomous intelligent systems. There are various methods for obstacle avoidance. In this paper, obstacle avoidance is achieved by the difference between left wheel velocity and right wheel velocity of differential drive robot. The magnitude of difference between the wheel velocities is used to steer the robot in correct direction. Data is collected by driving the robot manually. Ultrasonic sensors are used for distance measurement and IR sensors are used to collect the data of wheel velocities. This data is used to build a linear machine learning model which uses sonar data as input features. The model is used to predict the wheel velocities of the differential drive robot. The model built is then programmed into Atmega328 microcontroller using Arduino IDE. This enables the mobile robot to steer itself to avoid the obstacles. Since all the components used for this robot are highly available and cost effective, the robot is economically affordable.

Keywords: obstacle avoidance, autonomous mobile robot, Arduino , Nodemcu, raspberry pi , machine learning, regression , stochastic gradient descent , pseudo inverse.

1 Introduction

Robotics have transformed and revolutionized industries in various fields. One of the recent trends in robotics is autonomous robots. They have gained much popularity because manual intervention is not required to control the robot. These robots can take decisions by themselves. They use a variety of sensors to sense the changes in their environment. Machine learning and artificial intelligence allows these robots to take decisions by themselves and maintain high degree of autonomy.

Autonomous robots have been useful in developing self driving cars, intelligent rovers, autonomous delivery systems, unmanned vehicles, surveillance robots etc. These wide arrays of applications led to the increase in research community of autonomous systems.

Obstacle avoidance is considered as one of the important features of autonomous mobile robots. Various algorithms have been successful in achieving obstacle avoidance for mobile robots autonomously. In the paper “The Obstacle Detection and Obstacle Avoidance Algorithm Based on 2-D Lidar”[1], Peng Yan et al, used a 2-D LIDAR sensor with measuring range of up to 80m. They proposed a simple solution using visibility graph method. Using raw data from LIDAR sensor, the position and shape of the obstacle is found out. Then, optimum direction is selected by using a cost function.

Another method to avoid obstacles is by using image processing. Cheng-Pei, et al. [2] proposed a solution for obstacle avoidance using single camera. They have used a camera and two laser projectors fixed on same base. They have used image based distance measurement system (IBDMS) to find the location of obstacles from the image. It consists of simple image processing steps. Path planning is also done to achieve autonomous patrol.

One of the easiest ways to detect obstacles is by using ultrasonic sensor. It gives the range of obstacles in front of it. Jin, Yun, et al. [3] developed an omnidirectional intelligent obstacle avoidance system. They used an ultrasonic sensor supported by pwm servo motor which helps it to rotate in any direction. The obstacle distances measured by the ultrasonic sensor are stored in an array for selecting an optimal path. The intelligent car will move in the direction obtained by minimizing the objective function [3].

Wu, Ter Feng, et al. [4] also used ultrasonic sensors to implement a real time object avoidance system for wheeled robots. They used six ultrasonic sensors to measure the distance between the robot and the obstacles. It uses wall following method to achieve optimal path design. But the robot may or may not reach the target, which is a drawback of that system.

In the paper, "Video surveillance robot control using smart phone and Raspberry pi.", Bokade et al[5]., has created a robot which can be controlled by using the mobile phone. They have used raspberry pi to control the robot through wireless connection. The streaming speed of the video is 15 frames per second.

Singh et al[6], proposed a wireless robot to live stream both video and audio. The robot can be used as a surveillance robot. A web application is developed to control the robot wirelessly. Arduino Uno R3 board is used to control the mobile robot.

Kadiam et al[7], developed a robot that can be used for video surveillance using wifi and raspberry pi. They have used ARM 11 processor and a USB camera to capture the video. The USB camera is connected to Raspberry pi. The video is then used for extracting useful information using data mining techniques and pattern recognition. This process can be often termed as smart surveillance [8].

Lei Tai [9] et al., in his paper proposed a deep network solution for obstacle avoidance of the robot. They used convolutional neural networks for capturing images as input and for the decision making process which is an output that gives commands to the robot in which direction it should move while avoiding the obstacles. The dataset that has been used for training the network has been collected by moving the robot manually by avoiding the obstacles. The robot is confined to avoid the obstacles in the indoor environment .The model shows high similarity between human decisions and robot decisions while avoiding the obstacles.

Shichao Yang [10] et al., predicted the trajectory of the robots to avoid the obstacles using deep networks. The detection of the obstacles and generating the commands to the robot is done from the monocular images. The dataset that is used for training the model is NYUv2 RGB-D. Convolutional neural networks are used to predict the depth and surface normal and also to predict the trajectory of the robot. The 3d cost functions are used which helps in choosing the best trajectory path.

Wilbert G [11] et al., proposed a system for obstacle avoidance for unmanned aerial vehicles. The input images are taken from the camera and compared to the images that are stored in the database. They include SURF which detects the obstacles and avoid them using the control law. The UAVs recovers the path after avoiding an obstacle. The proposed system is faster in detecting the obstacles and flexible.

Mihai Duguleana [12] et al., proposed a new methodology for solving the issue of autonomous development of robots that contain both static and dynamic obstructions. They proposed path planning with artificial intelligence approaches .They designed a trajectory planner algorithm so that robot can be maintained at any speed. They modelled the robot in VR and MATLAB and they also tested in both VR and in real environment.

Punarjay Chakravarty [13] et al., proposed CNN Architecture for predicting depth estimation from a single image. They proposed control algorithm for estimating depth for guiding a quadrotor away from obstacles. They collected data that is online images , trained the network and controlled the drone. They calculated the performance of the depth network in navigating the drone in different environments.

Wilbert G. Aguilar [14] et al., proposed an algorithm for obstacle avoidance system for unmanned aerial vehicles using one-eyed camera. They also tested the execution system including the obstacle detection and obstacle avoidance. They proposed Speeded up Robust Features (SURF) algorithm that is matching across the image from database and real time image at Unmanned Aerial Vehicles, this algorithm gives the high execution considering the accuracy.

2 Methodology

In this paper, ultrasonic sensors are used to collect data related to range of the obstacles and IR sensors to measure the wheel velocities. Unlike other methods we use the collected data to build a machine learning model, which takes ultrasonic sensor data as input feature and predicts the left wheel velocity and right wheel velocity. The entire process of building the autonomous obstacle avoidance robot is divided into three phases as shown in fig 1.

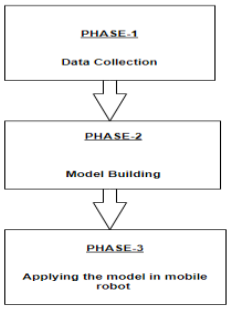

Fig 1. Overview of building the autonomous object avoidance robot

Data is required to build a machine learning model to achieve obstacle avoidance. We used two sensors for collecting the data. They are ultrasonic sensor and IR sensor. The two sensors are placed on the mobile robot. Raspberry pi is used to collect the data from these sensors.

The ultrasonic sensors are used to collect the range of obstacles in front direction. IR sensors are used to collect the rpm of left wheel and right wheel. Rpm values can be converted to into speed in metres per second using formula 1.

$$speed = 3.14159 * \frac{D}{100} * \frac{RPM}{60} \text{........ (1)}$$

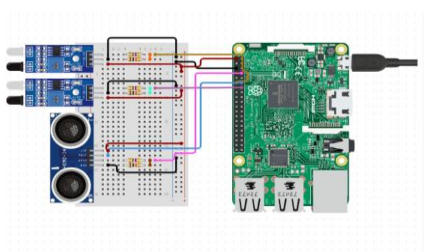

Fig 2. Raspberry pi and sensors connection.

The circuit diagram for the sensors and raspberry pi shown in the fig 2. Note that the Ultrasonic sensors is placed in the front side of the mobile robot and IR sensors are placed facing towards the wheels of the mobile robot as shown in the figure 3.

Fig 3. Position of sensors on the mobile robot

The mobile robot is controlled by NodeMCU [16]. The DC motors of the mobile robot are connected to NodeMCU as shown in fig 3. The circuit diagram for the DC motor connections with NodeMCU is shown in fig 4.

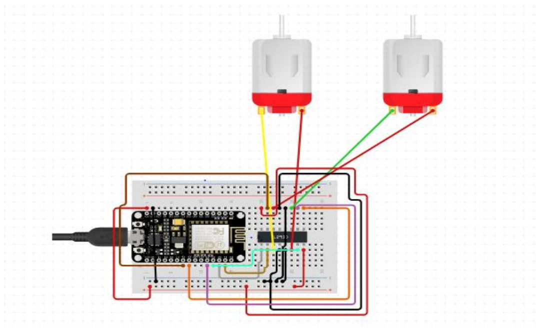

Fig 4. DC motor and NodeMCU connections

NodeMCU is connected to a mobile app which controls the mobile robot by changing the wheel velocity from the app. This helps in the manual training of the robot. The app which controls the mobile robot during training phase is should in figure 5.The android application can be built using MIT app inventor tool [17]. The connections of android app and Nodemcu are shown in figure 6.

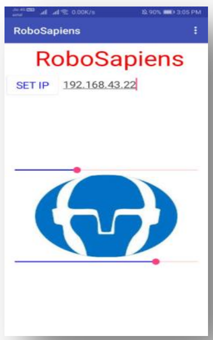

Fig 5. Android app to control the mobile robot manually.

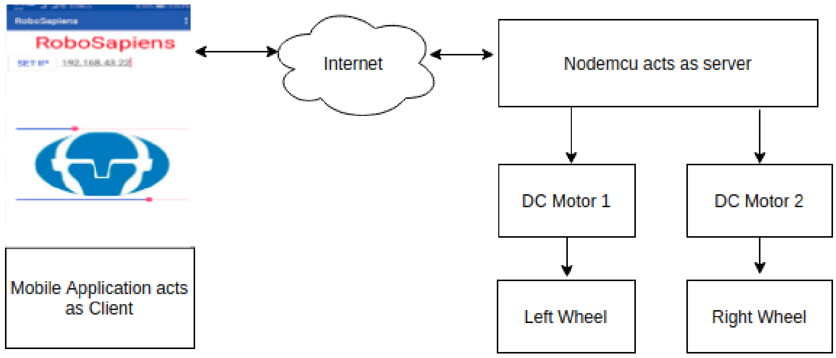

Fig 6. Connections between android application and nodemcu.

Algorithm to train the mobile robot manually and the collect the sensor data:

1. Turn on the mobile robot by powering it with a battery or a power bank.
2. Connect to the mobile robot from the android application by using the IP address of NodeMCU.
3. Control the movement of the mobile robot manually by using the android application.
4. Avoid obstacles by controlling the mobile robot only in one direction i.e. either left or right. Here let us choose right direction only.

The data collected during the training phase is stored as a dataset in raspberry pi. The features stored in the dataset include obstacle distance in centimetres, left wheel speed and right wheel speed in terms of rpm. The sample dataset is shown in table1.

Table 1. Sample dataset for the machine learning model

Sonar Distance (cm)	Left wheel velocity(rpm)	Right wheel velocity(rpm)
30	1000	1000
25	950	950
23	900	855
21	850	800
15	800	500
10	800	200

The flow chart for the data collection phase is shown in fig 7.

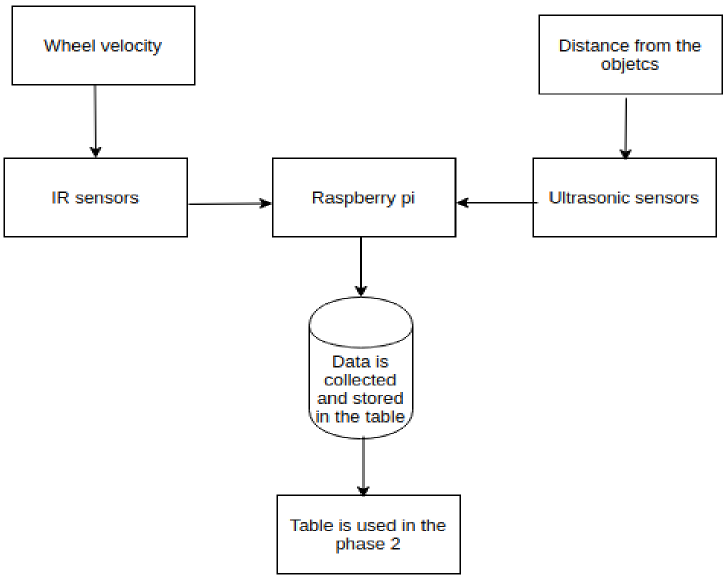

Fig 7. The analysis diagram for phase 1

The dataset collected from the training phase is used to train the regression model. The input features for the training model is object range. We use this single feature to predict the left wheel velocity and right wheel velocity of the mobile robot. The flow-chart for model building phase is shown in fig 8.

The machine learning model used for predictions is linear regression model. We use a simple linear model of degree ‘ d ‘ to predict the left wheel velocity and right wheel velocity. The model used for the velocity prediction is shown below.

Model equations for wheel velocity:

$$V[L] = W0 + W1 * x + W2 * x^2 + \cdots + Wd^d$$
$$V[L] = W'0 + W'1 * x + W'2 * x^2 + \cdots + W'd^d$$

V[l] = velocity of the left wheel

V[r] = velocity of the right wheel.

Where w0 , w1 ,w2 ... wd , w’0 , w’1 , w’2 ... w’d are the parameters of the model which are found out using stochastic gradient descent algorithm. Pseudo inverse algorithm can also be used to find out the weights ‘w’. The equation to find out w using stochastic gradient descent algorithms is shown below.

Update rule for weights in stochastic gradient descent:

Error function $E = \frac{\sum E_i}{n}$

$E_i = (y - yd)^2$

Y = predicted velocity

Yd = desired velocity

$w[i] = w[i] - \eta * dw[i]$

Where 'η' is a chosen parameter called learning rate which varies between 0 and 1 and dw[i] is the gradient of the error function corresponding to the weight w[i].

By iteratively updating the weights the error function is minimized and the optimal weights are found out for the model.

The final model obtained by the optimal weights can be used in the mobile robot to achieve the obstacle avoidance algorithm.

The other way to find the weights of the model is using the pseudo inverse algorithm shown in figure 8.

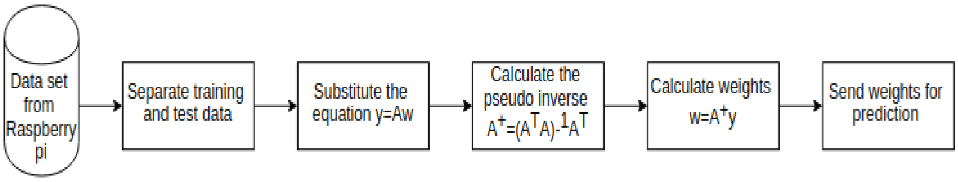

Fig 8. Pseudo inverse algorithm to find weights of the model

3 Obstacle Avoidance algorithm

There are three ultrasonic sensors placed on the mobile robot during testing phase on the front, left and right side.

The obstacle avoidance algorithm for the mobile robot is shown below:

1. Read the distance value from front side ultrasonic sensor of the mobile robot.
2. If the distance is greater than 25cm set the left wheel velocity and the right wheel velocity of the mobile robot as follows, otherwise go to step 3.

$V[l] = V[r] = 1000$

Here 1000 is the pwm value sent to the DC motors.

3. If the distance is less than 25cm then read the distances values of ultrasonic sensors from left and right side of the mobile robot.

dl = left side distance of the obstacle
dr = right side distance of the obstacle

(a) If dl < 10cm and dr > 10 cm then the robot should turn right side . We use the machine learning model to calculate the wheel velocities using right turn model.
(b) If dl > 10cm and dr < 10 cm then the robot should turn left side. We use the machine learning model to calculate the wheel velocities using the right turn model and interchange the left and right wheel velocities.

(c) If dl <10cm and dr < 10 cm the robot turns backwards. We set one of the wheel velocity to 0 and the other wheel velocity to 1000 for 2 seconds to turn the mobile robot in the backward direction.

The robot avoids the obstacle by changing its direction based on the magnitude of difference between the left wheel velocity and right wheel velocity. Thus obstacle avoidance is achieved using the given object avoidance algorithm.

4 Implementing obstacle avoidance

The circuit for implementing the autonomous mobile robot is shown in fig 9.

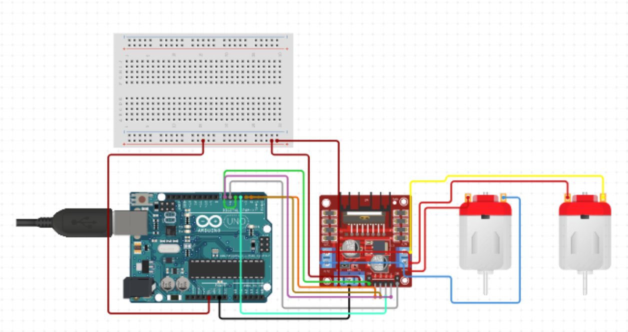

Fig 9. Circuit diagram for implementing autonomous mobile robot

The Arduino board [15] contains the code for implementing the linear regression model by substituting the distance value of the ultrasonic sensors. The arrangement of sensors over the mobile robot should as shown in fig 3.

5 Results and analysis

The accuracy of the obstacle avoidance model depends on the accuracy of training data collected during training phase. The model built using degree 3 linear equations performs better than the degree 2 model. The algorithm used is very simple to implement. The time complexity of the algorithm to predict the wheel velocities is O(1). We get the prediction instantaneously which makes it suitable for real time obstacle avoidance.

The testing phase of the robot is shown in fig 10 and fig 11.

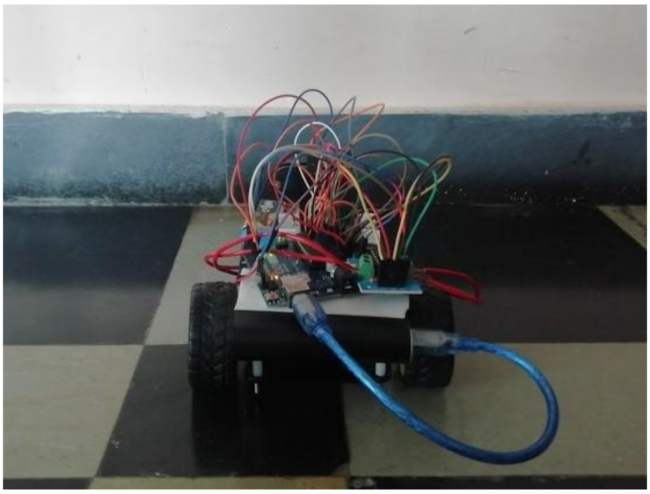

Fig 10. Robot approaching the obstacle.

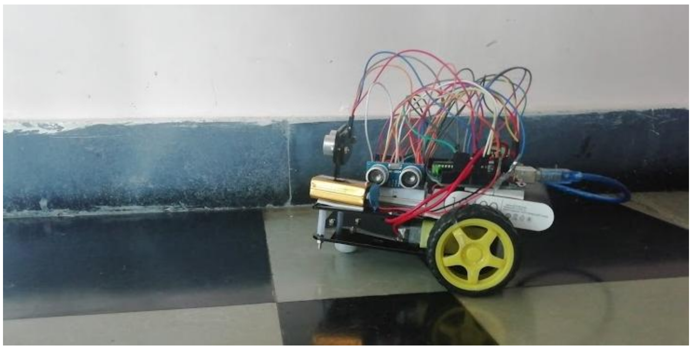

Fig 11. Robot avoiding the obstacle.

6 Conclusion and Future scope

In this paper an autonomous obstacle avoidance robot is developed which is suitable for real time applications. The robot developed serves as a prototype for implementation in real time applications. The robot can explore unknown environments and an IP cam can be placed on it for remote surveillance of the area. There are various applications for these types of robots in military vehicles, surveillance robots, unmanned vehicles and self driving cars.

An IP camera can be placed on the mobile robot which can be connected to a mobile application which enables live streaming of the surveillance area by connecting to the internet. The video data collected can be given as input to data mining algorithms and pattern recognition algorithms to extract useful information.

References

1. Peng, Yan, Dong Qu, Yuxuan Zhong, Shaorong Xie, Jun Luo, and Jason Gu. "The obstacle detection and obstacle avoidance algorithm based on 2-d lidar." In 2015 IEEE International Conference on Information and Automation, pp. 1648-1653. IEEE, 2015.
2. Tsai, Cheng-Pei, Chin-Tun Chuang, Ming-Chih Lu, Wei-Yen Wang, Shun-Feng Su, and Shyang-Lih Chang. "Machine-vision based obstacle avoidance system for robot system." In 2013 International Conference on System Science and Engineering (ICSSE), pp. 273-277. IEEE, 2013.
3. Jin, Yun, Shengquan Li, Juan Li, Hongbing Sun, and Yuanwang Wu. "Design of an Intelligent Active Obstacle Avoidance Car Based on Rotating Ultrasonic Sensors." In 2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER), pp. 753-757. IEEE, 2018.
4. Wu, Ter Feng, Pu Sheng Tsai, Nien Tsu Hu, and Jen Yang Chen. "Use of ultrasonic sensors to enable wheeled mobile robots to avoid obstacles." In 2014 Tenth International Conference on Intelligent Information Hiding and Multimedia Signal Processing, pp. 958-961. IEEE, 2014.
5. Bokade, Ashish U., and V. R. Ratnaparkhe. "Video surveillance robot control using smartphone and Raspberry pi." In 2016 International Conference on Communication and Signal Processing (ICCSP), pp. 2094-2097. IEEE, 2016.
6. Singh, Diksha, Pooja Zaware, and Anil Nandgaonkar. "Wi-Fi surveillance bot with real time audio & video streaming through Android mobile." In 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), pp. 746-750. IEEE, 2017.
7. Kadiam, Vineela, and G. Pavani. "Smart Phone Controlled Two Axes Robot for Video Surveillance Using Wireless Internet & Raspberry Pi Processor." International Journal of Research in Advent Technology 2, no. 10 (2014): 97-100.
8. Zhang, Lun, Stan Z. Li, Xiaotong Yuan, and Shiming Xiang. "Real-time object classification in video surveillance based on appearance learning." In 2007 IEEE Conference on Computer Vision and Pattern Recognition, pp. 1-8. IEEE, 2007.
9. Tai, Lei, Shaohua Li, and Ming Liu. "A deep-network solution towards model-less obstacle avoidance." In 2016 IEEE/RSJ international conference on intelligent robots and systems (IROS), pp. 2759-2764. IEEE, 2016.
10. Yang, S., Konam, S., Ma, C., Rosenthal, S., Veloso, M., & Scherer, S. (2017). Obstacle avoidance through deep networks based intermediate perception. arXiv preprint arXiv:1704.08759.
11. Aguilar, W.G., Casaliglla, V.P. and Pólit, J.L., 2017, January. Obstacle avoidance for low-cost UAVs. In 2017 IEEE 11th International Conference on Semantic Computing (ICSC) (pp. 503-508). IEEE.
12. Duguleana, M. and Mogan, G., 2016. Neural networks based reinforcement learning for mobile robots obstacle avoidance. Expert Systems with Applications, 62, pp.104-115.
13. Chakravarty, P., Kelchtermans, K., Roussel, T., Wellens, S., Tuytelaars, T. and Van Eycken, L., 2017, May. CNN-based single image obstacle avoidance on a quadrotor. In 2017 IEEE International Conference on Robotics and Automation (ICRA) (pp. 6369-6374). IEEE.
14. Aguilar, W.G., Casaliglla, V.P., Pólit, J.L., Abad, V. and Ruiz, H., 2017, June. Obstacle avoidance for flight safety on unmanned aerial vehicles. In International Work-Conference on Artificial Neural Networks (pp. 575-584). Springer
15. https://www.arduino.cc/en/Main/Software

16. https://www.nodemcu.com/index_en.html
17. https://appinventor.mit.edu/explore/ai-with-mit-app-inventor