

Development of Deep Learning Autonomous Car Using Raspberry Pi

Koon Ling Lee¹, Hong Yin Lam^{1*}

²Department of Electrical Engineering Technology, Faculty of Engineering Technology,
Universiti Tun Hussein Onn Malaysia, 84600 Pagoh, Johor, MALAYSIA

*Corresponding Author Designation

DOI: <https://doi.org/10.30880/peat.2021.02.01.053>

Received 09 January 2021; Accepted 01 March 2021; Available online 25 June 2021

Abstract: An autonomous vehicle is a vehicle that senses its environment and operates without human intervention. An autonomous car involves the extraction of surrounding information whereby such information is analysed by using the neural network. Machine learning algorithm requires the record the database of the use control for decision making and prediction. A major concern of such a driving method is to avoid the accident, which system can calculate the situation of the road and made the right decision to avoid the accident on the road. This work is concerned in modelling an autonomous car based on few aspects such as the distance between the obstacle and the view by the raspberry pi and the image classified. The image classifier is used to detect the sign board on the road. The aim of this work is to produce a prototype of an autonomous car which is capable of accurately and correctly to make the decision to launch the car fully automatic. The prototype is built with the sensor at the Raspberry Pi as the view of human, where system hardware is connected to remotely control via serial communication. This project consists of the auto mode and manual mode. The autonomous car is controlled by the user in via manual mode for the purpose of data collection and training in order to be used in the auto mode. Such a developed prototype further proves the real practical implementation of an autonomous self-driving car in the age of artificial intelligence.

Keywords: Autonomous Vehicle, Autonomous Car, Numpy, Machine Learning, Image Classified

1. Introduction

1.1 Overview

An autonomous car is a self-driving vehicle capable of perceiving the world around it and handling itself without human intervention. [1] Autonomous driving had to include many technologies, such as localisation planning and control, which need many heterogeneous sensors, actuators, and computers. According to [2], there are five levels of autonomous driving, the level four autonomous driving under some condition, but the level four is fully autonomous for all normal vehicles, all these levels are defined

by the Society of Automotive Engineers(SAE). Nowadays, we are still far from the fully autonomous driving car but autonomous car technology is making our driving safer and easier.

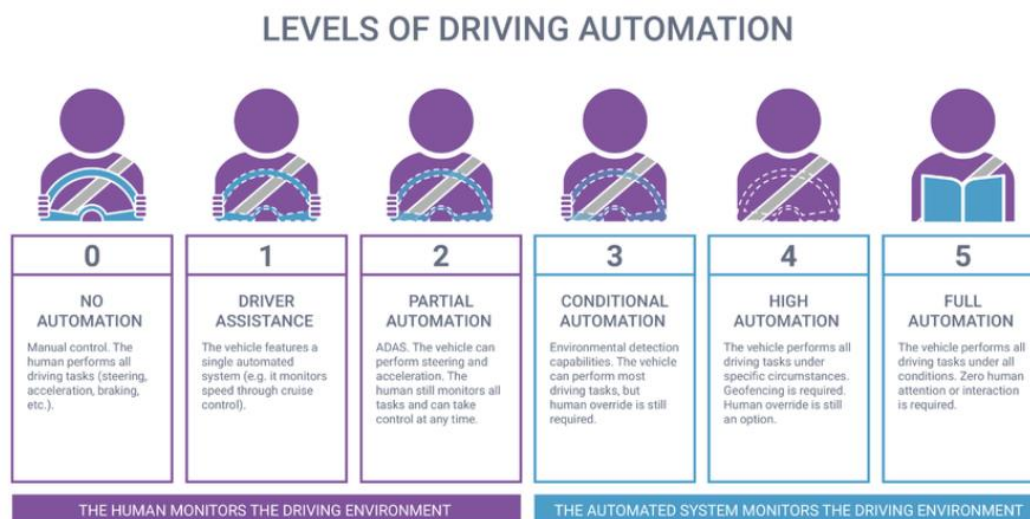


Figure 1: The automated driving level [3]

The autonomous vehicle needs to learn from a convolution neural network, using a large amount of data. Then, it needs to assume to get the effect of acknowledgment by inputting the value into the network in real-time. [4] A successful autonomous car must show out the stimulation results show high accuracy in learning to drive by observing the lanes and bypassing the obstacles. [5]

The autonomous car algorithm is based on the feature-based technique localises the lanes in the road images by combining the low-level features, such as painted lines or lane edges etc. [6] According to [7], the technique requires well-studies road having well-painted lines or strong lane edges, otherwise it will fail. The disadvantage is it does not impose any global constraints on the shape of the edges of the lane, and the technology may be troubled by occlusion or noise.

The autonomous car also have a localisation system to know the position of the vehicle and the perception of the car, because autonomous cars find optimal paths and monitor vehicle movement only when the ego-vehicle location is accessible from the localisation system. . [8]

The perception of the system provide important information on the surrounding environment depend on the use of several types of sensor components such as, cameras, radars, and laser scanner. For instance, a range sensor-based (consist of radar and a laser scanner) perception system detects and tracks static and dynamic obstacles, [9] [10] whereas a vision-based perception system recognises various visual objects.[11] [12] The data of the objects observed and recognised will be used for the autonomous driving device situation assessment.

This study focuses on developing and implement the machines learning to remote control(RC) car to make it self-driving, this system used the raspberry pi B+ to become the "brain" of the system and Arduino Uno is used to controlling the car direction. The car will equip with a 5 Mp camera to detect the road situation and the ultrasonic to detect the distance or the blind area of the vehicles.

2. Methodology

The methodology had covered the system operating principle, block diagram, flowchart, hardware set-up, proposed algorithm design and also bill of material.

2.1 Proposed system operating Principle & Block diagram

The autonomous car system will run used the neural network parameters to train the data had been collected by the computer. The raspberry pi acts as an interface that helps the car in collecting the surrounding data and send the data to the computer wirelessly; the neural network model will make a prediction for the steering based on the input images. The prediction will send to the command of the RC car by the Arduino.

Figure 2 shows the block diagram of the system. The autonomous car needs to make sure the first part is to make sure all the sensor and the RC car can work well, because the pi camera needs to live stream with the computer, and also the ultrasonic sensor is needs to continuously detect the obstacle in front, and brakes if deemed necessary. Next, the Arduino board is used to stimulate button-press action. The sign board can be detected by used the haar feature-based cascade classifier in Convolution Neural Network(CNN). The Haar feature-based cascade classifiers is an effective object detection proposed by Paul Viola and Michael Jones in their paper. [13] There are two mode for the autonomous car, one is manual mode and another one is the auto mode. In manual mode, a program is running on a computer that asks for input control. While the car is running, the pi camera will stream the real-time video to the laptop, when the user makes a decision, the frame will be saved into an array as the data for the output layer and all the files will be saved in npz file format. Then the npz file will be used for the training data, a large number of the image is split into train, validation, and test data. The training data is passed to CNN. Then hyper parameters are tuned with the help of validation data. In auto mode, the autonomous car can runs depend on the image taken by the Raspberry Pi camera. The convolution neural network will make the prediction output for each of the images that were streaming. At the same time, the ultrasonic sensor is run on the computer to detects an object and calculates the distance. If the object is a stop signal and a minimum distance is reached, then the stop signal will be sent to the RC car.

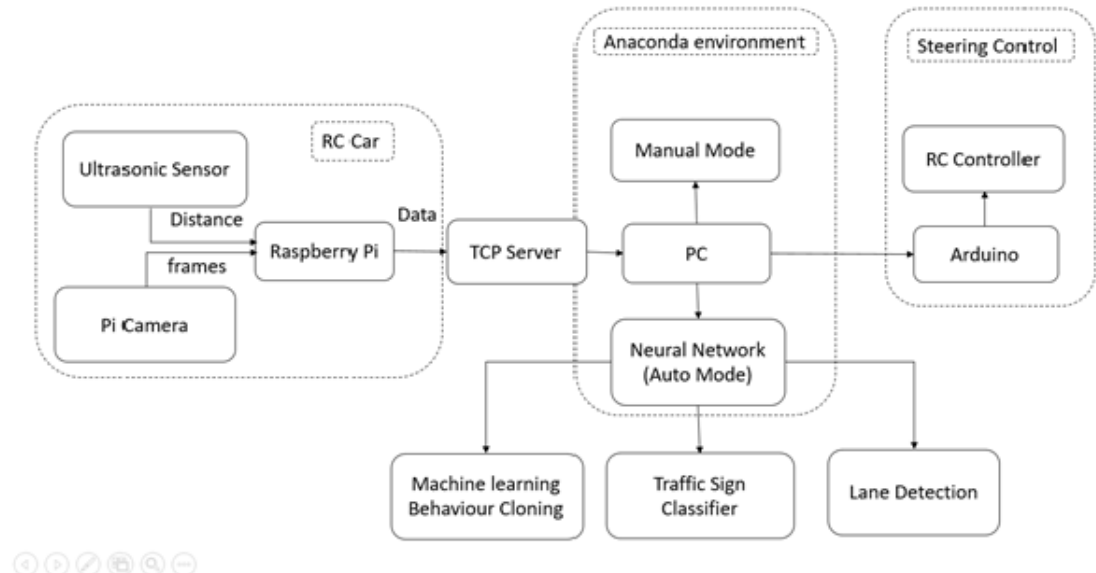


Figure 2: Block diagram system

2.2 System hardware & Set-up

Raspberry Pi model 4 and Camera Board V2 are chosen for the vision of the autonomous car and the ultrasonic sensor is used to detect the obstacle in the blind spot. The raspberry pi was start-up with a lithium battery, and its display by using mobile hotspot via VNC viewer as the internet-connected to both devices must be the same. Figure 3 show the hardware connection for the raspberry pi for the RC car, and there has a 1k Ω resistor between the echo pin in the ultrasonic sensor and raspberry pi to avoid

the high voltage directly through the sensor. The Arduino will act as a controller for the RC car. When the button is pressed, the resistance between the relevant chip pin and ground is zero. Thus, the Arduino needs to use to simulate button pressed action. Four Arduino pins are chosen to connect four chip pins on the controller, corresponding to forward, reverse, left and right actions respectively, Figure 4 shows the hardware system setup in Arduino.

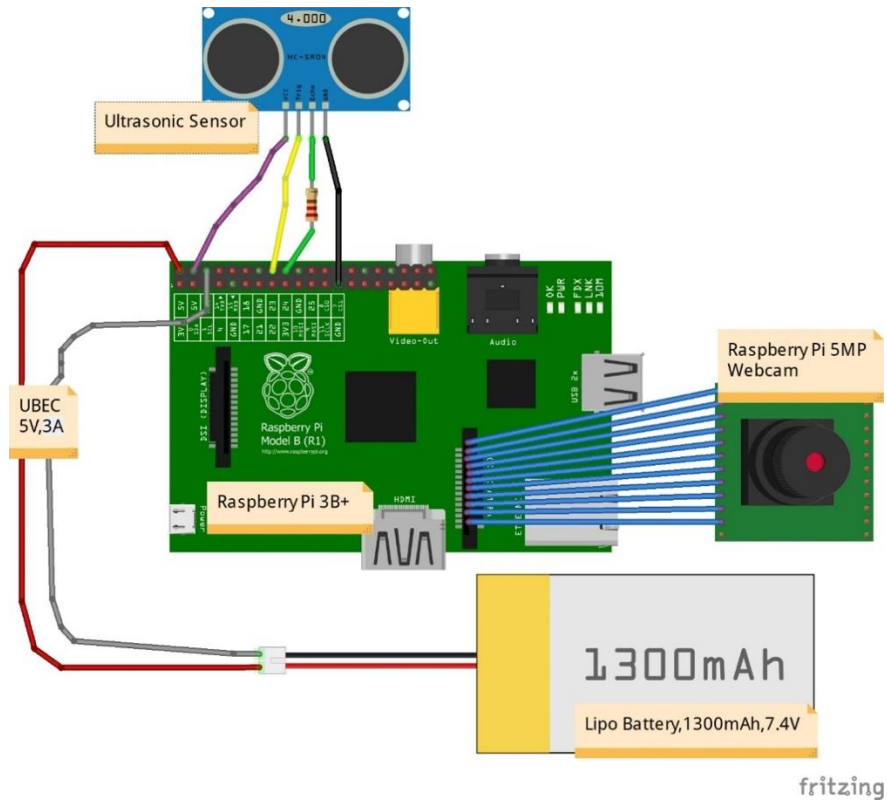


Figure 3:Raspberry Pi circuit diagram

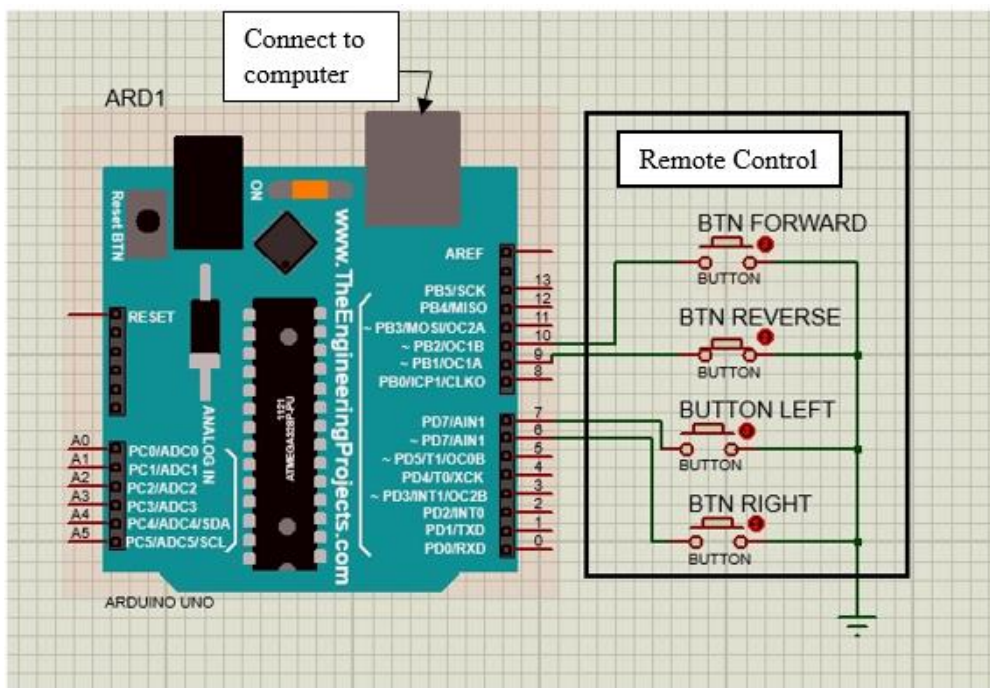


Figure 4: Arduino circuit diagram

2.3 Algorithm design and flowchart

The algorithm of the autonomous car is to be implemented with the data collected by the Raspberry Pi acts as a machine interface that helps in collecting image data for the neural network and realising algorithm decision on system output. The operation of the autonomous car of visual decision tasks can be distinguished into three stages: (1) Manual mode, (2) Train the data (3) Auto mode. The autonomous car must be collected the data in the manual mode first before it launches in auto mode.

The raspberry pi and Arduino is connected with Transmission Control Protocol (TCP) which it needs to enable the server and client, which is the computer is the server, and the raspberry pi is the client. In the manual mode program, the camera will stream the image frame from the raspberry pi camera to the computer by the socket. Next, the Arduino is used to control the RC car. It will use to send the LOW signal, indicates grounding the chip pins of the controller. The computer outputs command to Arduino using the serial interface, and then the Arduino reads the commands and writes out LOW or HIGH signals, simulating button-press actions to drive the RC car. In manual mode, the computer will save the image frame label data and the control label when there have sent the control data to the RC car. The computer will check it is the data is repeated. If repeated, it will delete the frame. If not, the data will save in the npz file as trained data. The sign board detection is needed to run by the haar cascade classifier. After that, the autonomous car detects the sign board automatically by the cascade classifier, and the npz file will be trained in the neural network. In autonomous mode, the raspberry pi will send the frame to the computer via socket, and compare the image frames label with the control label in the neural network. Then, the machine will predict the output based on the comparison in the neural network, and send the control data through the Arduino to the RC car and control the movement. Besides, the ultrasonic sensor will be used as the input control to avoid the obstacle in front of the car. The flowchart in Figure 5 shows the autonomous car in manual mode and Figure 6 is the flowchart for the auto mode.

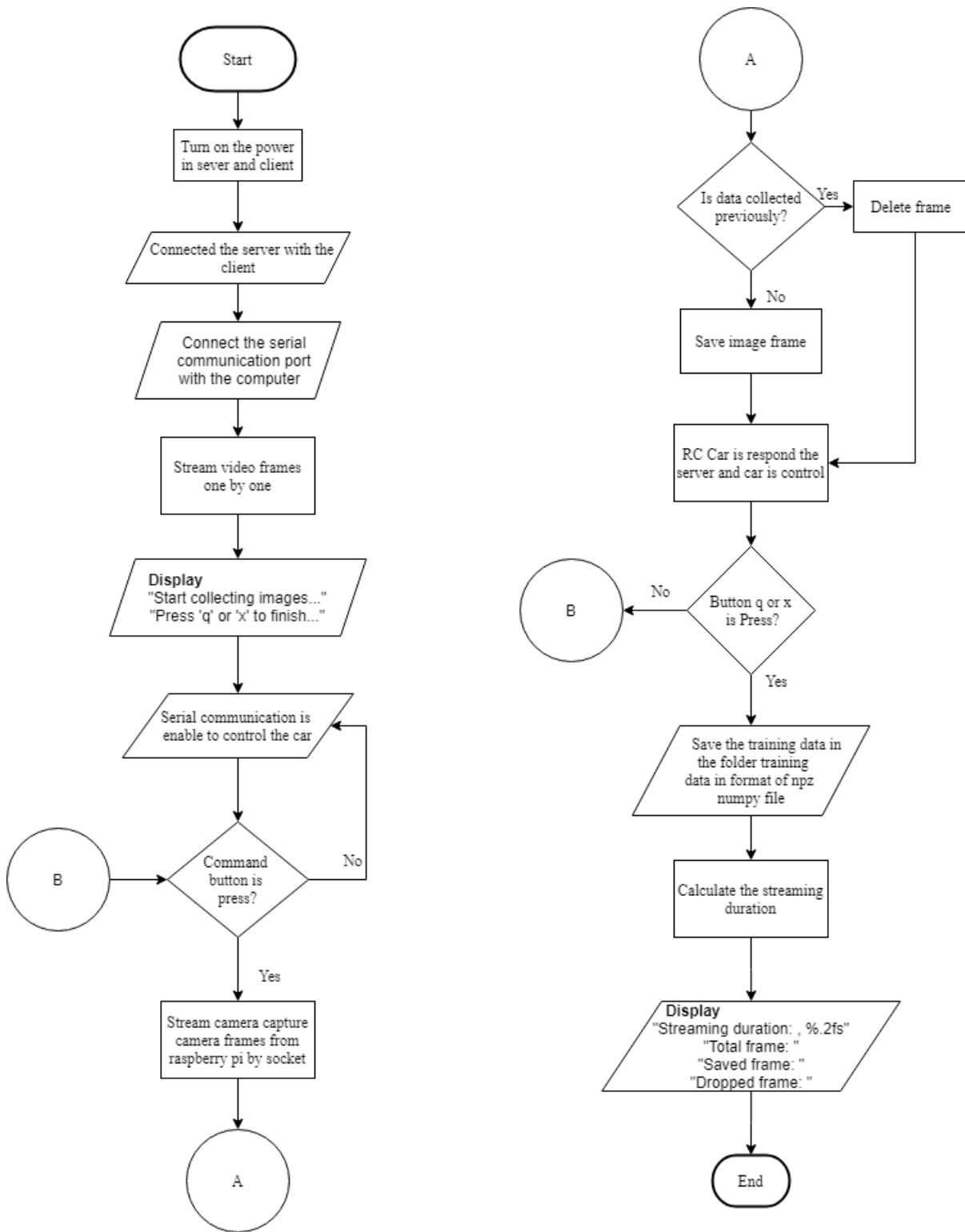


Figure 5: Flowchart in the manual mode

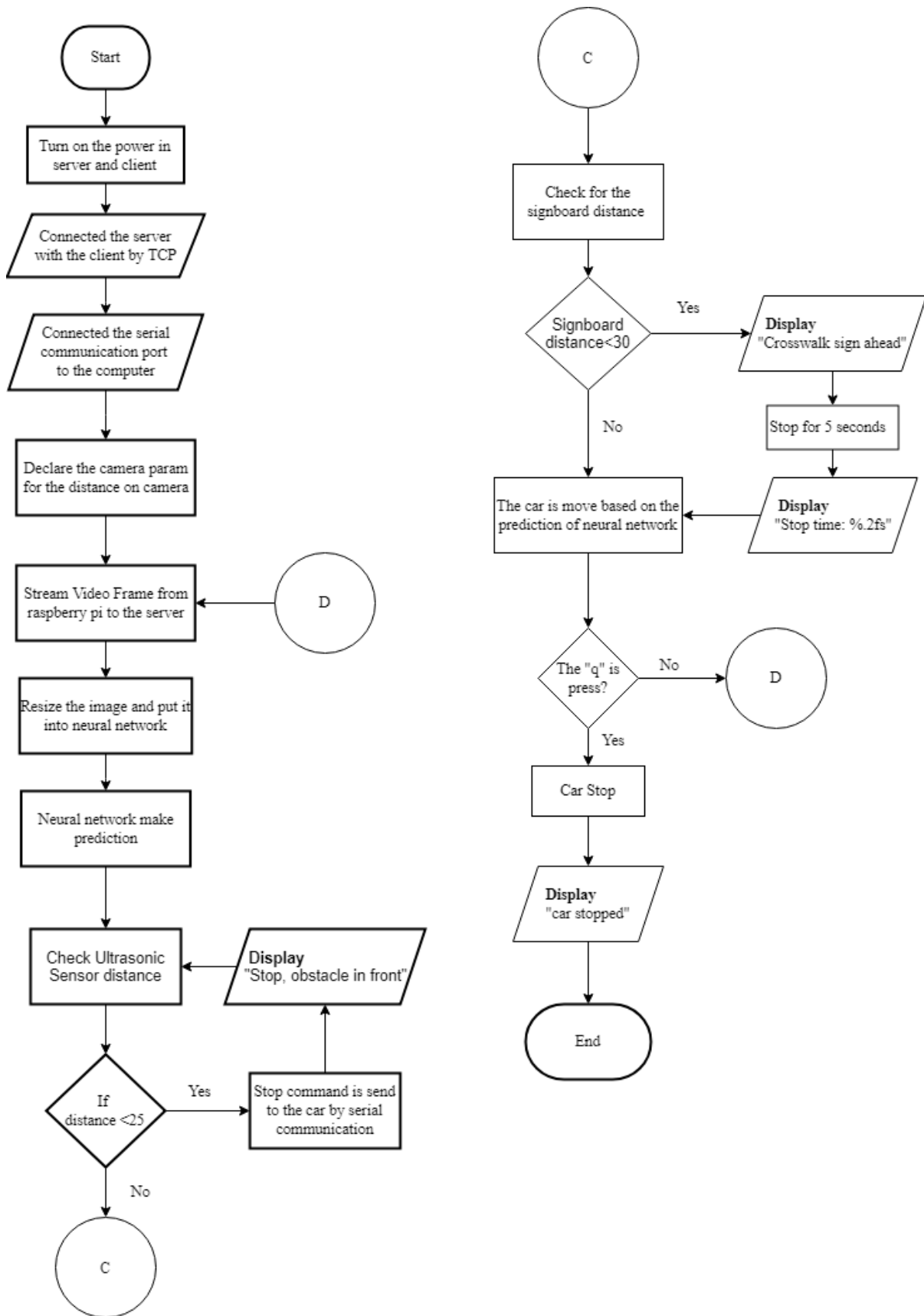


Figure 6: Flowchart in auto mode

2.4 Bill of Material and Project budget

Table 1 shows the list of hardware and component cost that used in this project.

Table 1: Bill of Material of Prototype Model

| Hardware/Components | Quantity | Cumulative Cost (RM) |
|----------------------------|----------|----------------------|
| 1. Raspberry Pi 4 | 1 | 175.00 |
| 2. Raspberry Pi 5MP Webcam | 1 | 15.50 |
| 3. Ultrasonic Sensor | 1 | 3.80 |
| 4. 3A switch mode UBEC 5A | 1 | 7.50 |
| 5. Resistor 1 k Ω | 3 | 0.50 |
| 6. RC Car | 1 | 19.00 |
| 7. Arduino UNO | 1 | 16.90 |
| 8. Lipo Battery 1400mAh | 1 | 29.00 |
| 9. Jumper Wire | 40 | 9.00 |
| GRAND TOTAL (RM) | | 276.20 |

3. Results and Discussion

The results and discussion of the deep learning autonomous car and the performance of the algorithm will be present in this section. The data had been collected and discuss based on the system used in the development of object detection function for the self-driving car.

3.1 Results

3.1.1 Autonomous Car in Manual Mode

To collect the data, the user should use the keyboard to control the RC car. When controlling the car, the car must slowdown to let the video streaming to keep up to save the frame. Next, the car must move to the centre of the lane, and make sure images are mostly labelled correctly and your data is mostly balanced, because it will affect the prediction of the output. In the condition that the car is not central, the collected some error correction data may also help for the training data. The frame will always run and when the command key is press by the user, the frame will be saved in the form of npz files. Figure7 show there have run 998 frame but only 469 number is saved in npz file.

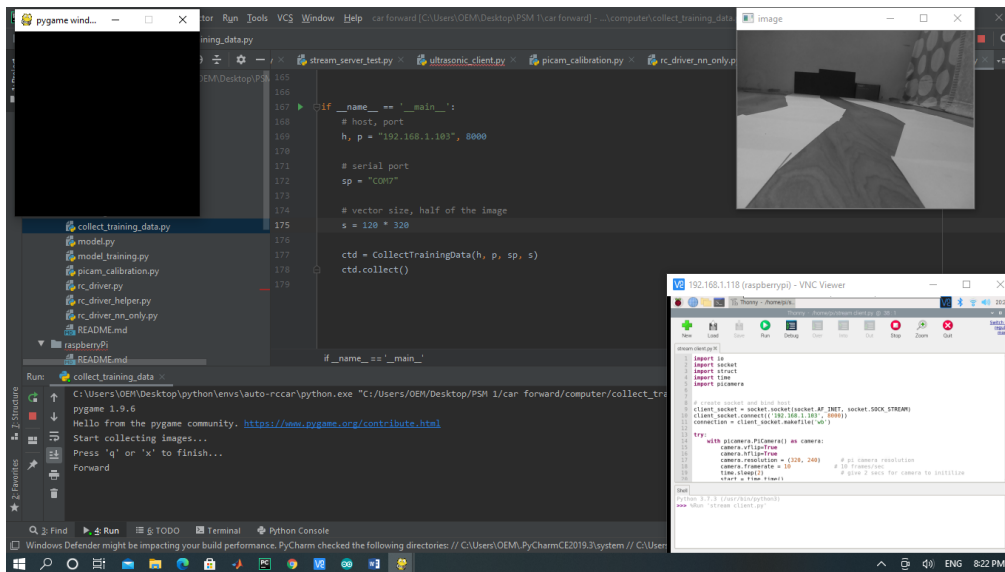


Figure 7: The interface for the collecting data

3.1.2 Autonomous Car in Neural Network Training

Neural network training of the data involve several proves. During training and prediction process, only half of the images is involved as this will reduce the significantly the training period. In this project, the training layer is used 38,400 (320×120) nodes in the input layer and 64 nodes in the hidden layer, which this is described in the coding "input_size = 120 * 320". The number of nodes in the hidden layer is chosen randomly. There have 4 output neuron for the four output prediction which is forward, left, right and reverse. Figure 8 shows the output for the training accuracy and also the validation accuracy and Figure 9 shows the train image and the original image.

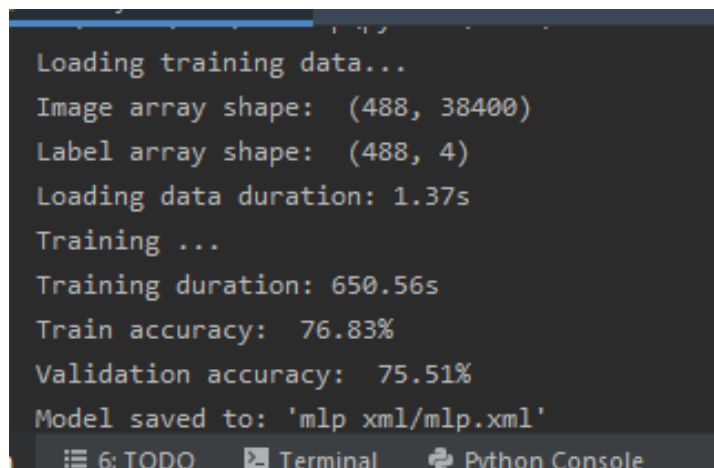


Figure 8: The accuracy after train by neural network

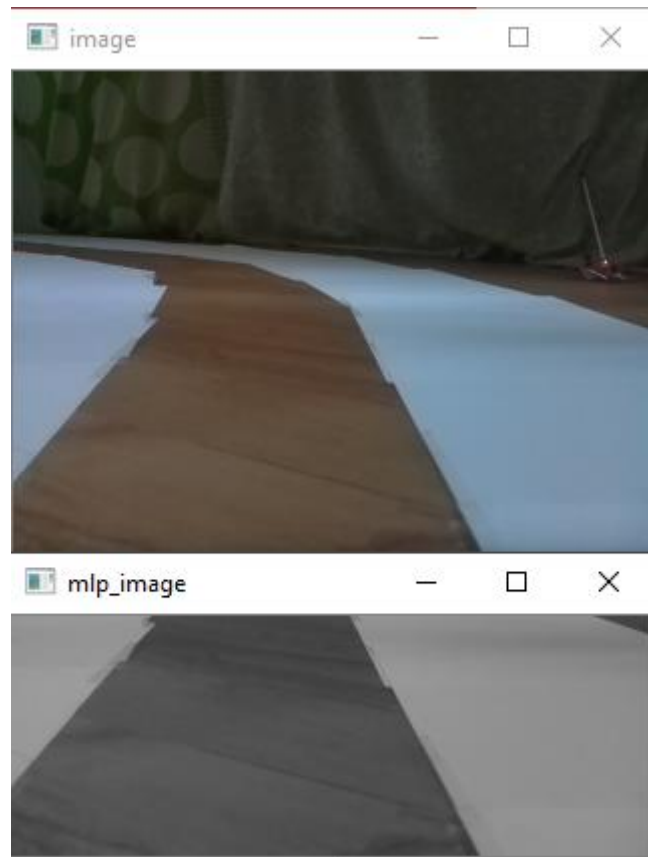


Figure 9: mlp image is train image (half of the original image)

The data collected will be converted to a numpy array, then the image of the train is combined with the train mark by the user input. Finally, all paired image data and labels are stored in the npz format. The neural network is trained in OpenCV3 using a back spreading process. Once training is finished, the weights will be saved to the Extensible Markup Language (XML) format. The same neural network is designed and loaded with a qualified XML file to produce predictions.

3.1.3 Self-driving Implement

The project's neural network output was divided into three neurons: forward, turn left and turn right .All the frame predictions are produced about 15 times a second in the actual driving situation (streaming rate of approximately 15 frames/s).Figure 10 show the result for the autonomous car with self-driving and the stop crosswalk sign is detected at the certain distance.

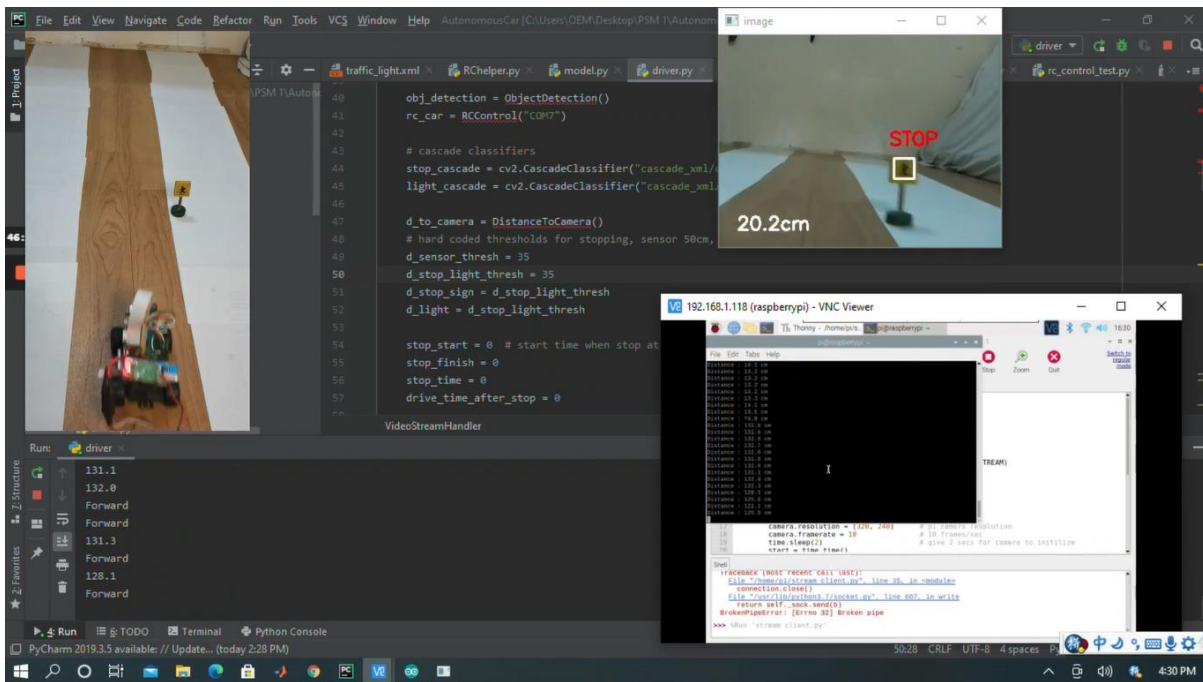


Figure 10: The autonomous car run during straight

3.2 Discussions

The training set is used to train the model, while the validation set is used only to test the efficiency of the model. When increasing the neural network’s size layer, it will affect the accuracy of the training data. In order to increase the number of layers, it took a much longer time to train and that the accuracy of the results was not much improved, so it was not a good decision to improve the accuracy, so keep the layer as 120*320 nodes in the first layer. For the hidden layer, it has different neurons such as 16, 32, and 64, the maximum efficiency is the 32 neuron, the 64 neuron was take too long too trained the data and the accuracy was close to 32 neuron. The training process will stop when it close to 80.00 % accuracy as the accuracy value will no longer increase. Table 2 and 3 shows the data collected in forward, left turn, right turn, and in the complex road. Figure 11 shows the train accuracy graph and Figure 12 validation accuracy graph.

Table 2: The training accuracy of the data with samples

| Sample(N)/ Direction | 0 | 200 | 300 | 400 | 500 | 600 | 700 |
|-------------------------|---|-------|-------|-------|-------|-------|-------|
| Forward | 0 | 63.10 | 68.04 | 75.34 | 79.74 | - | - |
| Left | 0 | 53.24 | 65.64 | 76.45 | 82.43 | - | - |
| Right | 0 | 41.54 | 60.29 | 70.78 | 78.31 | - | - |
| Complex | 0 | 37.83 | 47.15 | 55.86 | 69.36 | 82.14 | 87.41 |

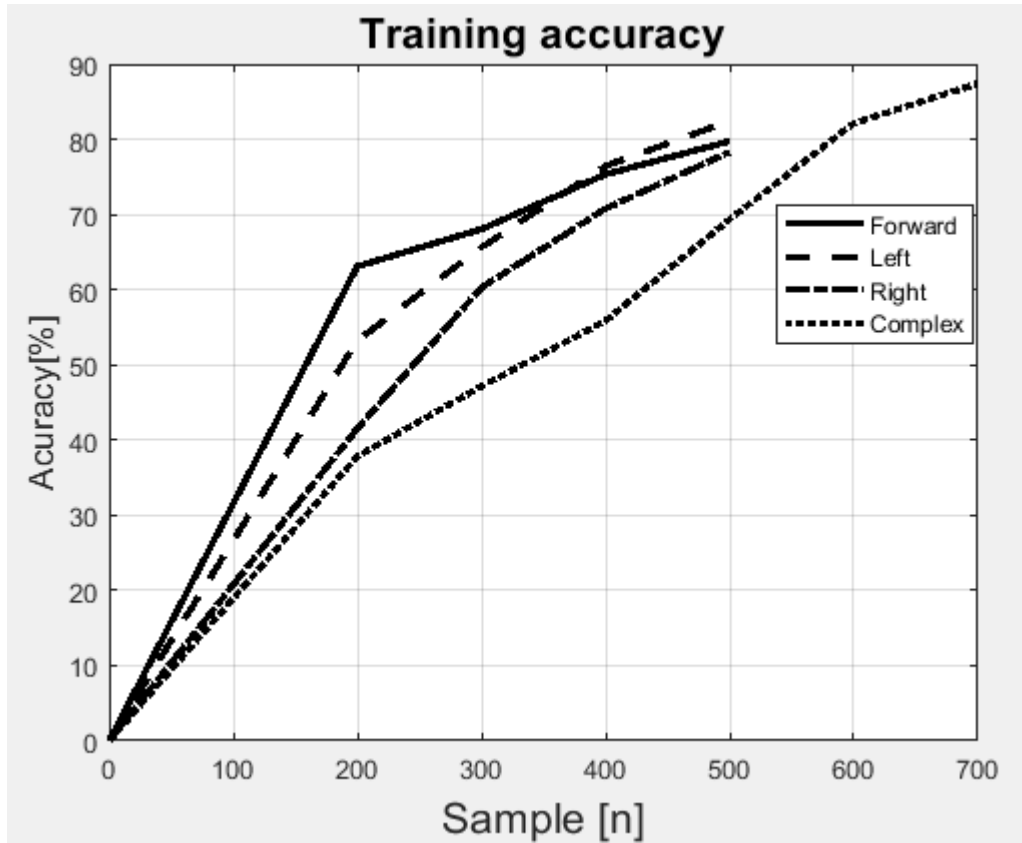


Figure 11: Training accuracy graph

Table 3 : The validation accuracy of the data with samples

| Sample(N)/ Road | 0 | 200 | 300 | 400 | 500 | 600 | 700 |
|--------------------|---|-------|-------|-------|-------|-------|-------|
| Forward | 0 | 60.1 | 67.44 | 74.21 | 79.42 | - | - |
| Left | 0 | 49.12 | 64.49 | 75.10 | 82.30 | - | - |
| Right | 0 | 37.78 | 57.32 | 69.63 | 77.78 | - | - |
| Complex | 0 | 34.52 | 40.68 | 54.32 | 68.37 | 82.03 | 86.77 |

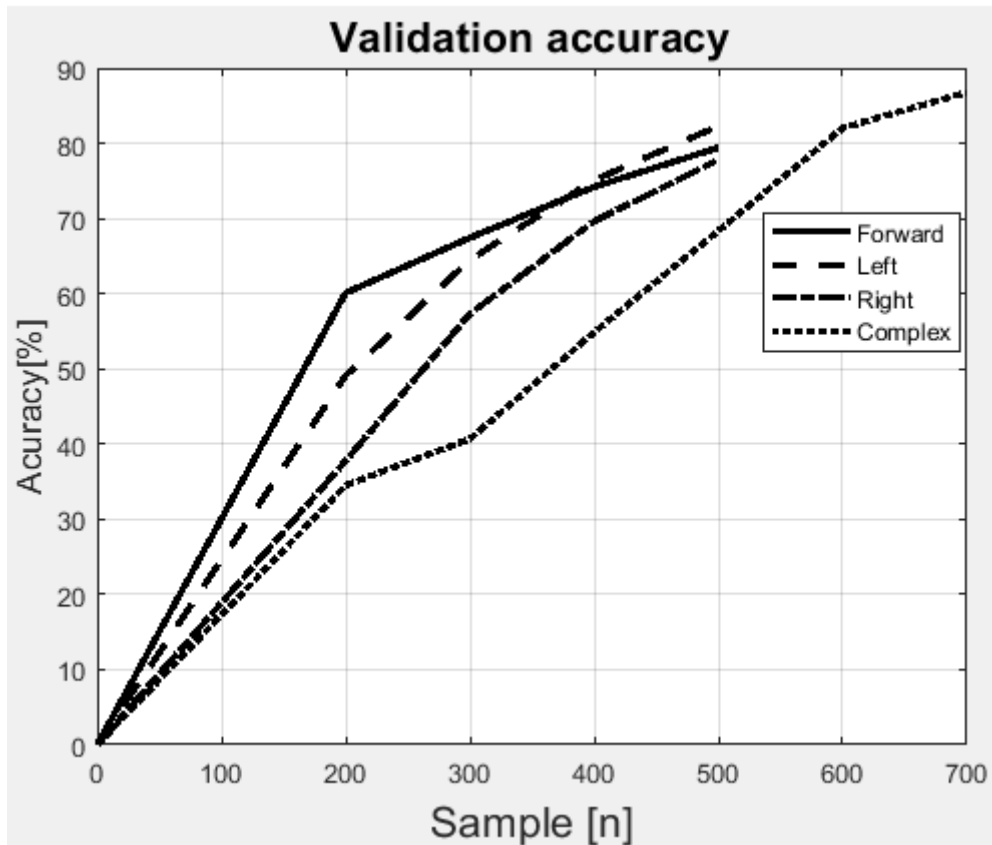


Figure 12: Validation accuracy graph

The distance measurement by the ultrasonic sensor is accurate when there have an obstacle in front, but the measurement by the pi camera is not accurate for the result. Table 4 show the detection distance by using the pi camera.

Table 4: The distance measure by pi camera

| | | | | | | |
|-----------------------|------|------|------|------|------|------|
| Actual Distance (cm) | 10 | 15 | 20 | 25 | 30 | 25 |
| Measure Distance (cm) | 11.7 | 17.5 | 22.6 | 28.0 | 33.8 | 42.8 |

The accuracy of the monocular vision approach could be affected by the object bounding box variations in the detecting process, error in the camera calibration process and also the nonlinear relationship between the distance and camera coordinate. Next, the surrounding effect will also affect the reading by the pi camera, for example, the different brightness of light and the different place that the signboard places will affect the reading. When the distance is closer, so the camera will rapidly change in the camera coordinate, so the value will more accurate and the greater the error.

3.3 Limitation

The project is limited because the five main issues will directly affect the performance of the accuracy of the autonomous car.

1. Battery issue the battery is used 2 AA batteries for the RC car, but if the capacity of the battery decrease, the speed and the steering of the RC car is affected.
2. The environment will affect the output of the neural network, if the light or colour of the road is differs from the one trained, it may cause the wrong prediction from the neural network.
3. Camera module also limited because during training the data, there have many pictures need to train from the different angle, if the camera angle has been moved or changed, the device might not be able to recognise the current track status.

4. The quality of the RC car will also affect the output for the neural network, if the car alignment has problem, it will cause the angle of the wheel can turn left and turn right are different, so the car can't work properly.
5. The network issue will also affect the performance of the autonomous car, as the project needs the raspberry pi to provide the live streaming to the computer via WiFi Link. The live streaming will let the neural network to decide the correct command to send to the controller, which is Arduino. However, if the network of the WiFi link was unstable, the video may lag and be stuck on a single picture frame. This will cause the system error which repeatedly performs detection. A more powerful machine is needed to replace the raspberry pi in order to perform rapid detection process.

4. Conclusion

This work has provided relevant real-world traffic scenarios. It is to avoid accidents over all kinds of representative situations that can occur on main roads, side streets or urban traffic. Autonomous vehicles are theoretically safer than human drivers. Millions of miles of road tests have already been launched for the self-driving car in others country, and experts say that the technology clearly has the potential to be safer than human drives. The Baidu company was planning to operate a fully autonomous car in 2025. In autonomous vehicles, the scenarios can be extrapolated globally to include the same road networks. This will contribute to the creation of subsequent testing data. Through this research, the autonomous car should be 100.00 % accuracy to run on the road, because from this research, even the accuracy is 80.00 %, it will also have the error prediction from the neural network. Besides, to implement the real autonomous car, the car should have a better sensor to avoid the wrong prediction from the neural network.

4.1 Recommendation

This prototype work revealed several critical problem that need to be addressed. In general, the work can be improved by the algorithm more accuracy for the sensors, as because the serious accidents sometimes result from the sensing function or computer system failures. The speed of the autonomous car can be control according to the road signboard. The present algorithm performs on the operations of all in the real-time frames. It is accurate, but if it begins learning by itself and avoids needless calculations of regions that are already known or familiar, its effectiveness could be further enhanced. Once the car starts to travel on the roads, it will determine the road situation, and the car will remember the characteristic features. Then the computer will generate the XML file every time after the car travels. So, the project can further improve with the availability of some additional equipment, which can be more autonomous as well as more realistic for a vehicle on the road.

Acknowledgement

The authors would like to thank the Faculty of Engineering Technology, Universiti Tun Hussein Onn Malaysia for its support.

References

- [1] Jo, K., Kim, J., Kim, D., Jang, C., & Sunwoo, M. (2014). Development of autonomous car— Part I: Distributed system architecture and development process. *IEEE Transactions on Industrial Electronics*, 61(12), 7131-7140.
- [2] Litman, T. (2020). Autonomous vehicle implementation predictions: Implications for transport planning.
- [3] SAE On-Road Automated Vehicle Standards Committee. (2014). Taxonomy and definitions for terms related to on-road motor vehicle automated driving systems. *SAE Standard J, 3016*, 1-16.

- [4] Oh, C. S., & Yoon, J. M. (2019, February). Hardware acceleration technology for deep-learning in autonomous vehicles. In *2019 IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 1-3). IEEE.
- [5] Okuyama, T., Gonsalves, T., & Upadhyay, J. (2018, March). Autonomous driving system based on deep q learning. In *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)* (pp. 201-205). IEEE.
- [6] EK, W. Y. T., & Sben, D. (2004). Lane detection and tracking using B-Snake, Image and Vision Computer. *Image and Vision computing*, 22, 269-280.
- [7] McCall, J. C., & Trivedi, M. M. (2006). Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation. *IEEE transactions on intelligent transportation systems*, 7(1), 20-37.
- [8] Jo, K., Kim, J., Kim, D., Jang, C., & Sunwoo, M. (2015). Development of autonomous car—Part II: A case study on the implementation of an autonomous driving system based on distributed architecture. *IEEE Transactions on Industrial Electronics*, 62(8), 5119-5132.
- [9] Kang, Y., Roh, C., Suh, S. B., & Song, B. (2012). A lidar-based decision-making method for road boundary detection using multiple kalman filters. *IEEE Transactions on Industrial Electronics*, 59(11), 4360-4368.
- [10] Zhao, H., Sha, J., Zhao, Y., Xi, J., Cui, J., Zha, H., & Shibasaki, R. (2011). Detection and tracking of moving objects at intersections using a network of laser scanners. *IEEE transactions on intelligent transportation systems*, 13(2), 655-670.
- [11] Hu, Y., Zhao, W., & Wang, L. (2009). Vision-based target tracking and collision avoidance for two autonomous robotic fish. *IEEE Transactions on Industrial Electronics*, 56(5), 1401-1410.
- [12] Lin, F., Dong, X., Chen, B. M., Lum, K. Y., & Lee, T. H. (2011). A robust real-time embedded vision system on an unmanned rotorcraft for ground target following. *IEEE Transactions on Industrial Electronics*, 59(2), 1038-1049.
- [13] Viola, P., & Jones, M. (2001, December). Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001 (Vol. 1, pp. I-I)*. IEEE.