Research Inventy: International Journal of Engineering And Science
Vol.13, Issue 6 (June 2023), PP 79-87
Issn (e): 2278-4721, Issn (p):2319-6483, www.researchinventy.com

Research, design, and experiment the autonomous car using AI

Phan Lan Anh
[1] Phan Lan Anh, Vietnam Korea university of information and communication technology, Vietnam.
Corresponding Author: ptlanh@vku.udn.vn

Abstract: *Today, automatic technology is being researched for application in all aspects of life from the applications of the home to the industry. In this trend, self-driving cars are also one of the fields that many researchers and companies are interested in and develop in both theoretical and experimental fields. Current studies have applied image processing to detect lanes and obstacles, but have not provided a solution when facing problems such as blurred, broken lane lines, tree shadows, and building shadows. They are the main reasons for reducing the accuracy of the algorithm. In this paper, we study, design, and do experiments based on hardware, which is Raspberry Pi, a Pi camera using Opencv and Neural Network to build self-driving cars capable of recognizing lanes and recognizing lanes. notice boards.*

Keywords: *Self-driver car, image processing, convolution nerural network, Raspberry Pi, camera Pi, Arduino uno…*

Date of Submission: 08-06-2023 Date of Acceptance: 22-06-2023

I. INTRODUCTION

In recent years, along with the development of information technology, artificial intelligence, the field of automation is also developing very strongly. The demand for driverless cars and self-driving devices is growing. In particular, lane identification and identification for Autonomous vehicles as well as Driver Assistance Systems is a particularly important problem, requiring increasingly high accuracy, and safe even in many different conditions such as light, weather, and a variety of operating environments. For industrial self-propelled devices in the past, input data is usually taken from sensors such as infrared sensors, magnetic sensors, etc., along with fixed designs for the operating environment such as light, line, etc., which limits the range of operation as well as the versatility of the devices. Modern self-driving devices such as driverless cars, drones, cargo robots, and automated delivery drones often use camera systems to collect data for fund identification. moving direction. Data from the camera sensor can be processed by conventional image processing algorithms such as color filtering, edge detection, image rotation, noise filtering, etc. to extract the road lines from which to determine the lane. and the center of the road. However, those algorithms are often unstable with noise due to brightness changes, lanes with tree shadows, lanes with wet rain, and even unable to identify lanes when lanes are lost... However, in practice, the road markings are not always as clear as in the example above, the line can be blurred or obscured by other vehicles. The algorithm cannot be extracted. This is a huge limitation for conventional image processing algorithms.

Recently, the field of artificial intelligence in general and deep learning, in particular, has achieved many breakthroughs achievements in many fields of life and technology. Deep learning is a branch of machine learning that relies on a set of algorithms to attempt to model highly abstracted data using multiple processing layers with complex structures, or by Another way involves many nonlinear transformations to approximate any function between output and input [1-2]. Deep learning is a powerful set of machine learning techniques that use multi-layered artificial neural networks. Various deep learning architectures such as Multi Layers Perceptron (MLP), Convolution Neutral Network (CNN), Deep Belief Network (DBN), and Deep Belief Network (DBN) Recurrent Neutral Network (RNN) has been applied to fields such as computer vision, automatic speech recognition, natural language processing, speech sound recognition, and bioinformatics. proven to produce very good results for a variety of tasks. There have been many studies aimed at applying deep learning to the problem of lane identification and giving very good results [3-9]. With algorithms using deep learning models, models are often built according to the same architecture as Segmentation networks [10], from the input image, the model will classify pixels as lanes or not. Desiring to learn and develop new technology for self-driving cars, I made this research to propose an application model of IoT and AI (deep learning) with a good ability in predicting the path of vehicle devices. self-propelled with small processing power, easy to build training data. In addition, I use the Haar cascade to recognize signs, traffic lights, and obstacles.

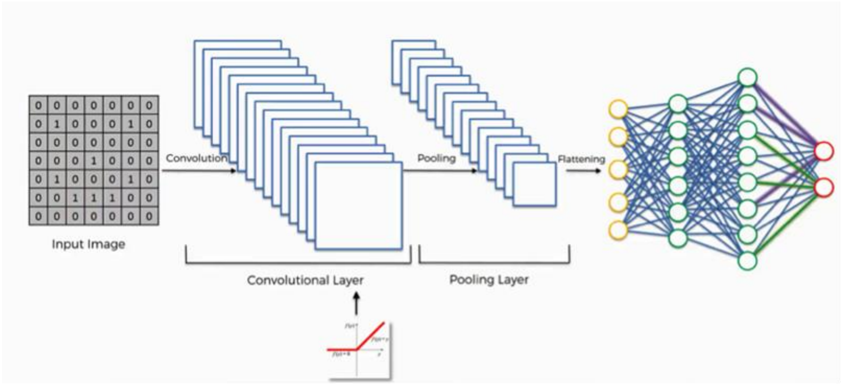
II. CNN AND IMAGE PROCESSING

2.1 Convolution neural network.

Convolutional Neural Network is one of the most popular and influential deep learning models in the Computer Vision community. CNN is used in many problems such as image recognition [11-15], video analysis, MRI images, or problems in the field of natural language processing, and most of them solve these problems well.

CNN is a neural network architecture that is well suited for problems where the data is images or video. There are two main types of layers in CNN: The convolutional layer and the Pooling layer.

Figure 1. Convolution Neural Network (CNN)

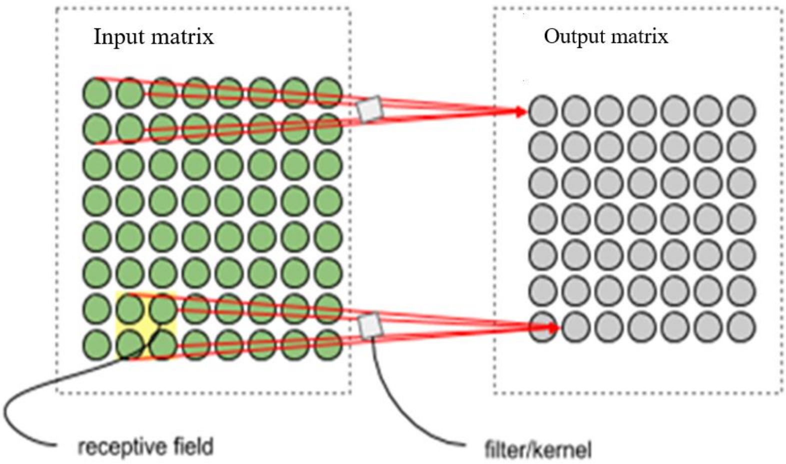


Convolutional layers

The convolution layer is the most important and also the first layer of the CNN model. This class has the main function of detecting spatially efficient features. In this layer, there are 4 main objects: input matrix, filters, and receptive field, feature map. The convolution layer takes as input a 3-dimensional matrix and a set of filters to learn. This filter will slide through each position on the image to calculate the convolution between the filter and the corresponding part of the image. This corresponding part of the image is called the receptive field, that is, the area that a neuron can see to make a decision, and the battle produced by this process is called the feature map. To imagine, you can imagine, the filters are like the watchtowers in a prison that scan the surrounding space, in turn, to look for the escaped prisoner. When an escaped prisoner is detected, an alarm will sound, just like filters looking for a certain feature the convolution will give a large value.

For the example below, the input data is a matrix of size 8x8x1, a filter set of size 2x2x1, and a feature map of size 7x7x1. Each value in the feature map is calculated as the sum of the product of the corresponding elements of the 2x2x1 filter with the receptive field on the image. And to calculate all the values for the feature map, you need to slide the filter from left to right, from top to bottom. Therefore, it can be seen that convolution preserves the spatial order of the pixels. For example, the left corner point of the input data will correspond to the side of a point to the left corner of the feature map.

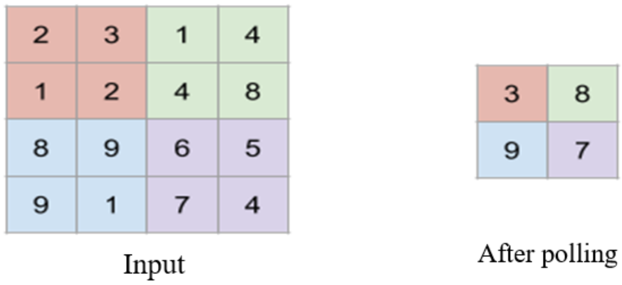
Figure 2. Feature map



Pooling layer

After the activation function, we normally use the pooling layer. Some common types of pooling layers are max-pooling, and average pooling, whose main function is to reduce the dimension of the previous layer. With a 2x2 pooling, you need to slide this 2x2 filter over areas of similar size and then calculate the max, or average, for that area.

Figure 3. Pooling

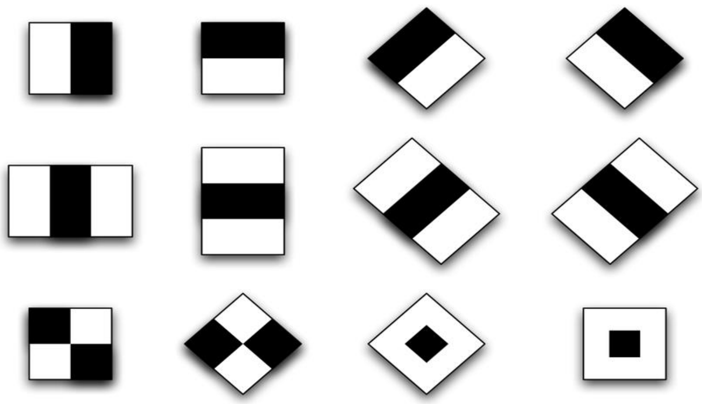


Keras currently supports many different types of pooling layers such as max pooling, average pooling, and global average pooling... To use we do the same as with convolutional layers.

2.2 Harr Cascade

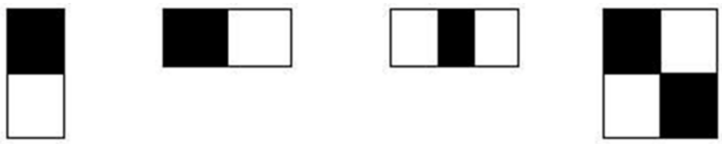
Haar-Like features [16] are rectangles classified into different regions as shown:

Figure 4. Haar - Like



Includes 4 basic features to identify the object you want to identify. Each Haar-Like feature is a combination of two or three black or white rectangles as shown in the following figure:

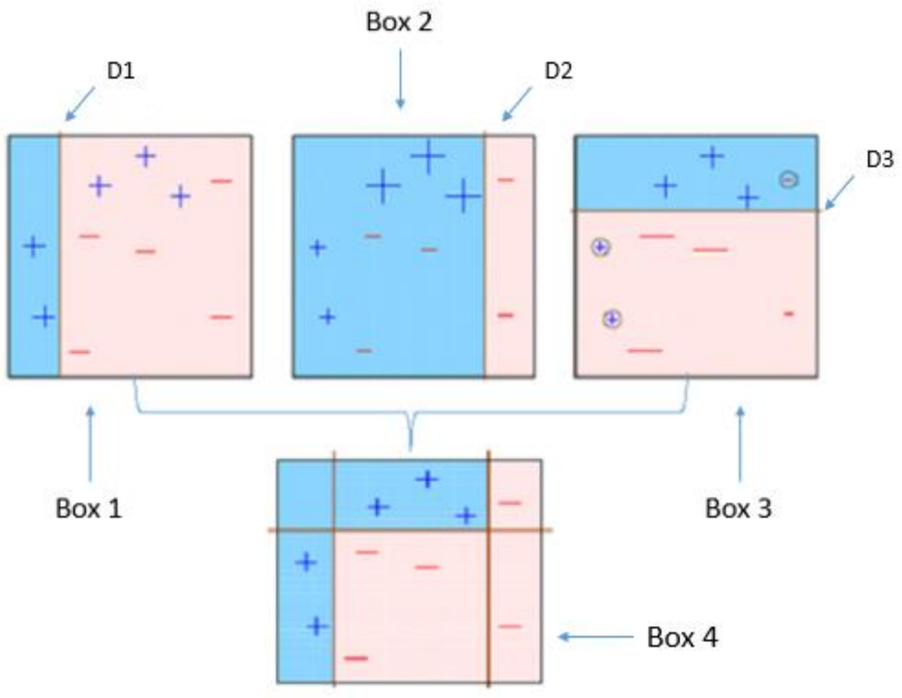
Figure 5. Four feartures of Haar - Like



AdaBoost

AdaBoost is a complex nonlinear strong classifier based on the boosting approach introduced by Freund and Schapire in 1995. Adaboost also works on the principle of linearly combining weak classifiers to form one of the classifiers.

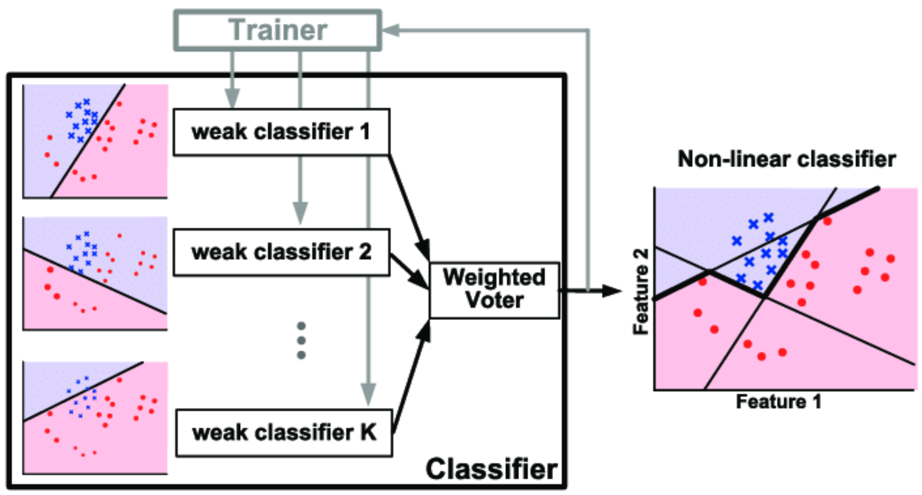
Figure 6. Boosting



When there are different weak classifiers, combine them to form a strong classifier. The combination is pretty simple, just logical AND/OR blocks depending on the result given. A simple linear combination can already compute the combination of the above decision outputs.

Adaptive
In normal boosting, the classifiers all have an equal role, but after going through Adaboost, the classifiers will combine the weak classifier into a strong classifier.

Figure 7. Result of filter

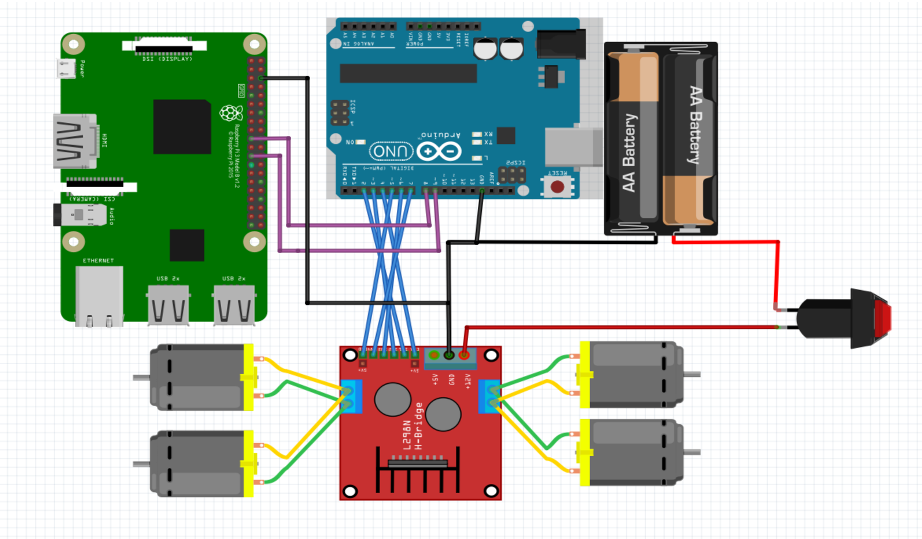


III. DESIGN

3.1. Hardware

The vehicle is designed and connected components as shown in the figure 8. The image signal obtained from the Pi camera will be transferred to the Raspberry and analyzed here. The Raspberry Pi will then make the decisions for the car needs to go, such as turn right, turn left, stop... These decisions will be sent to the Arduino Uno for the Arduino to control the motors to make.

Figure 8. The Schematic of car



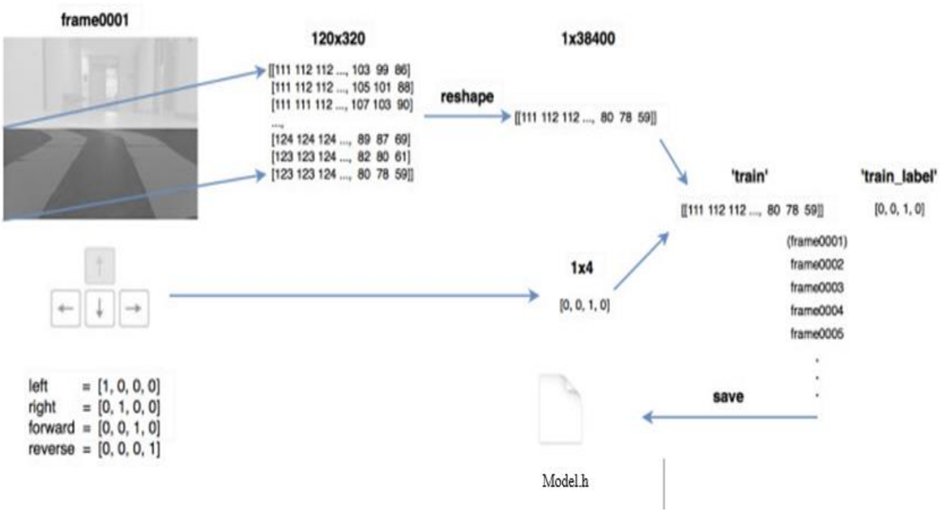
3.2 Soft ware

Step 1: Collect data large enough during manual vehicle control.

Use the Vehicle Control Keyboard to collect data. The data after being received will be saved to the folder (DataCollected) to serve the data training step.

Step 2: Conduct training from previously collected data.

Figure 9. Data trainning



Previously collected data including images path and steering angle will be fed into a pandas data frame, each frame is cropped and converted into a complex array. The images are then visualized and data balanced. Next, prepare for processing the pandas data frame that will output the path list and steering list data. At the Split dataset step, the Path list will split into xTrain and xVal, and the steering List into yTrain and yVal. Model creation and training. Finally, all the concatenated image data is saved to the Model creating a model.h file.

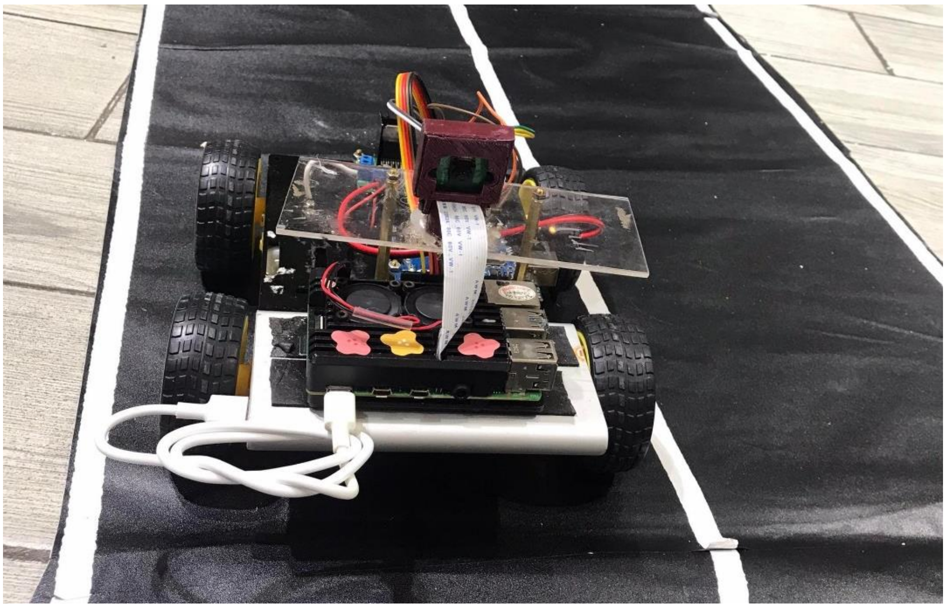
Step 3: Execute
When the car is running, the Webcam will acquire images with the new model (model.h file) that has been trained before, and the motor will predict the steering angle to determine the lane.

IV. EXPERIMENT AND RESULT

4.1 Hardware

The hardware result is an autonomous vehicle that can go in the right lane, meet a dead end, can turn around and recognize the signs and control the vehicle according to the requirements of the signal.

Figure 10. Hardware of the car



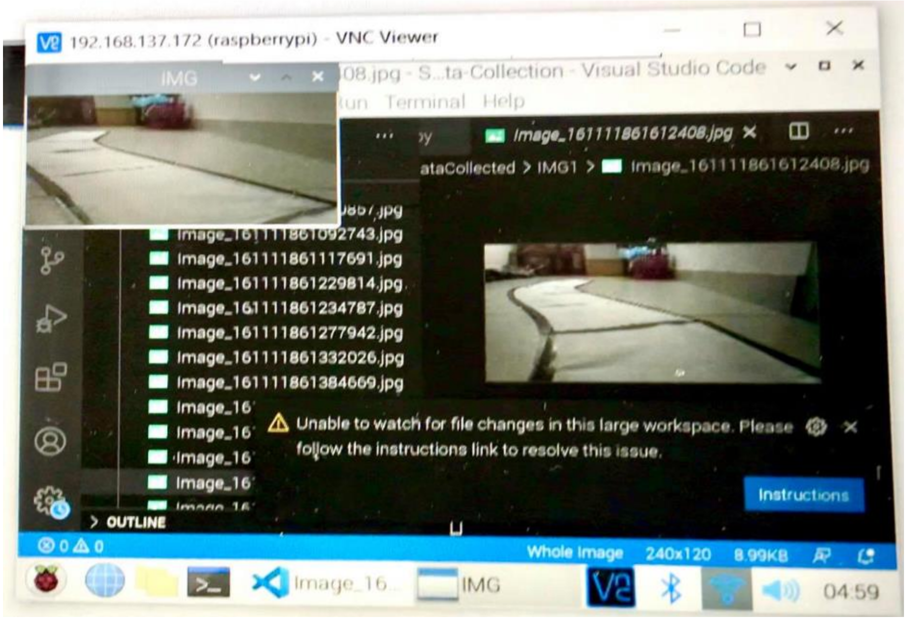
4.2 Software

4.2.1 Lan detection

Step 1: Manually control the vehicle and collect data.

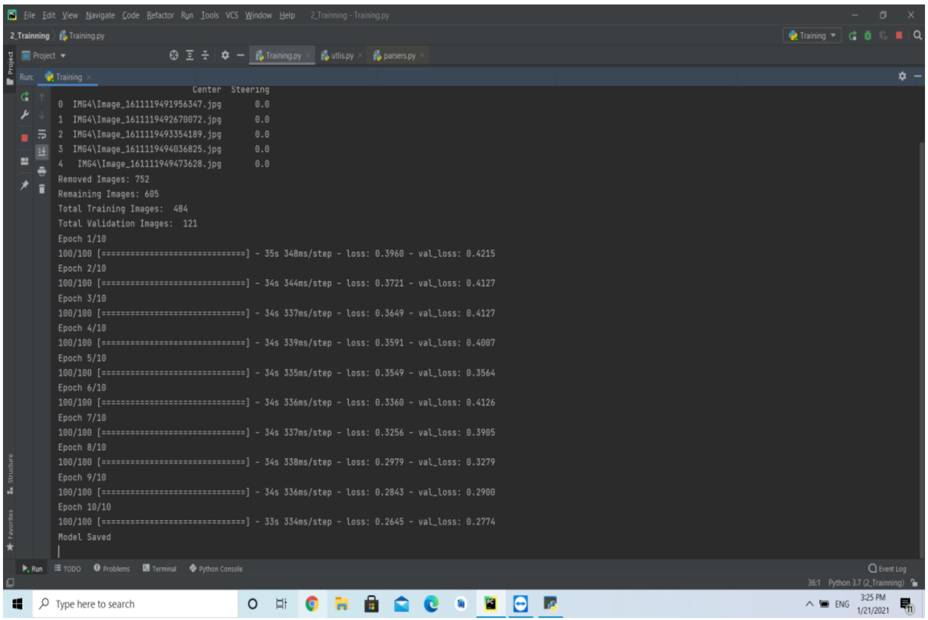
Figure 11. The car follows the lane

Figure 12. Collect the data from the images



After driving the vehicle and collecting data obtained folders of images and logfiles. Prepare for the training step.
Step 2: Conduct training from previously collected data.
Add the data information from Step 1 to the pandas data frame, visualize and balance the data.
And perform data separation, take arguments, process data, create Tensorflow model and then conduct training

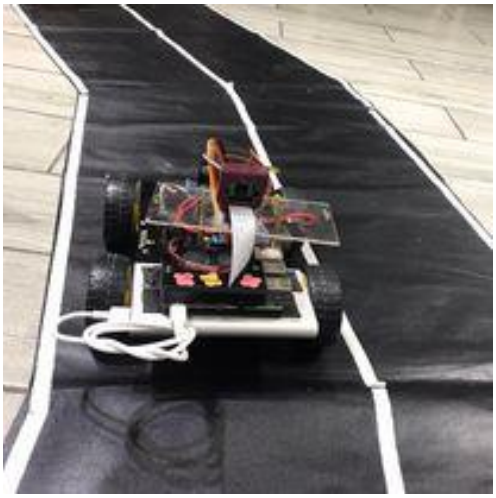
Figure 13. Data trainning



When the training is completed, a model.h file will be obtained to store the steering angle to perform road recognition and self-driving processing for the vehicle in Step 3.

Step 3: Execute

Figure 14. The result



4.2.2 Object detection

Figure 15. The car detect the traffic light

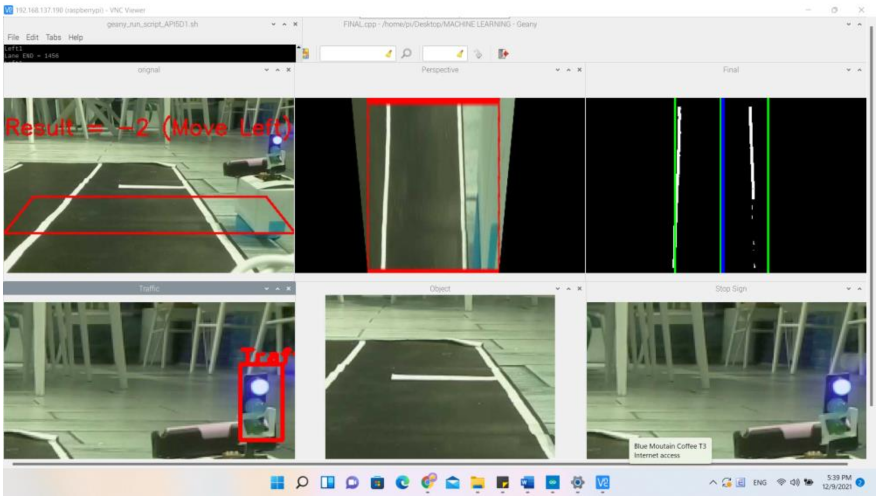
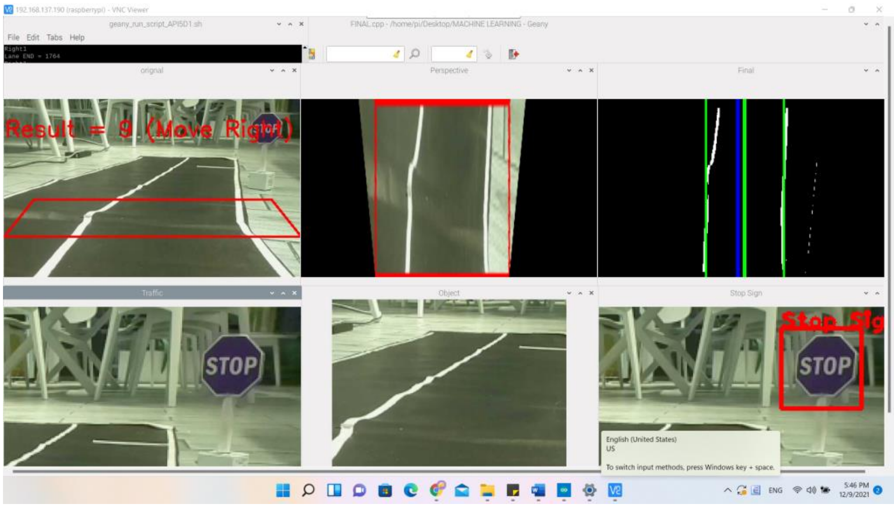


Figure 16. The car detects the Stop Sign



V. CONCLUSION

In this article, we have studied, designed and built a successful self-driving car model using Raspberry Pi, Camera Pi, Arduino uno, motor, and other electromagnetic components. Arduino uno controls the motor and Raspberry pi 4 is used for the training process. The camera is used in the process of collecting vehicle data and predicting that the car can steer itself at corners in the lane without human assistance.

REFERENCES

[1]. M. Nielsen, "Neural Networks and Deep Learning," in Neural Networks and Deep Learning, US, Springer, 2019, pp. 2-4.

[2]. Jurgen Schimidhuber, " Deep learning in neural networks: An overview", Elsevier, Volume 61, 2015, pp 85-117.

[3]. Ian Goodfellow, Yoshua Bengio, Aaron Courville, "Deep learning," in Deep learning, US, The MIT Press US, 2016, pp. 326-365.

[4]. Malte Oeljeklaus, Frank Hoffmann, Torsten Bertram, "A Fast Multi-Task CNN for Spatial Understanding of Traffic Scenes," Maui, 2018 21st International Conference on Intelligent Transportation Systems(ITSC), 2018, pp. 2825-2830.

[5]. Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Ad-am, "Encoder-Decoder with Atrous Separable Convolution for Semantic Image," Munich, 2018 European Conference on Computer Vision (ECCV), 2018, pp. 833-851.

[6]. Muhammad Jefri Muril; Nor Hidayati Abdul Aziz, HA Ghani, Nor AA Aziz, "A review on deep learning and nondeep learning approach for lane detection system", IEEE Conference on Systems, Process & Control (ICSPC), 2020.

[7]. Jigang Tang, Songbin Li, Peng Liu, "A review of lane detection methods based on deep learning", Elsevier, Volume 111, 2021.

[8]. Noor Jannal Zakaria, Mohd Ibrahim Shapiai,..., " Lane detection in autonomous vehovles: A systematic review", IEEE access, Volume 11, 2023, pp 3729 - 3765.

[9]. Sunil Kumar, Manisha Jailia, Sudeep Varshney, "A comparative study of deep learning based lane detection methods", IEEE conference, 2022,

[10]. Yuenan Hou, Zheng Ma, Chunxiao Liu, and Chen Change Loy, "Learning Lightweight Lane Detection CNNs by Self Attention Distillation," Seoul, 2019 International Confer-ence on Computer Vision (ICCV), 2019, pp. 1013-1021.

[11]. Kamran, Sharif Amit, "Efficient Yet Deep Convolutional Neural Networks for Seman-tic Segmentation," Yogyakarta, 2018 International Symposium on Advanced Intelligent Informatics (SAIN), 2018, pp. 123-130.

[12]. Myeongsuk Pak, Sanghoon Kim, "A review of deep learning in image recognition", IEEE Xplore conference, 2017,.

[13]. Qing Liu, Ningyu Zhang, Wenzhu Yang, Sile Wang, Zhenchao Cui, Xiangyang Chen, Liping Chen, "A review of image recognition with deep convolutional neural network", ICIC 2017.

[14]. Hironobu Fujiyoshi, Tsubasa Hirakawa, Takayoshi Yamashita, "Deep learning - based image recognition for autonomous driving", Science direct, 2019, volume 43, pp 244 - 252.

[15]. A. Moujahid, M.E Tantaoui, M.D Hina, A. Soukane, A. Ortalda, A. Elkhadimi, A. Ramdane Cherif, "Machine learning techniques in ADAS: a review", International conference on advances in computing and communication engineering 2018.

[16]. T. Kurita," Improvement of learning for CNN with ReLU activation by sparse regulari-zation" , Anchorage, 2017 International Joint Conference on Neural Networks (IJCNN), 2017, pp. 2684-2691.