Design of autonomous vehicle using Machine learning

R.Kiruthika[1], K.Rajeswari[2], P.Manjula[3] and M.Surya[4]

[1,2,3]Assistant Professor, CSE Department, Mahendra Engineering College, India
[4]UG Scholar, CSE Department, Mahendra Engineering College, India
[1]kiruthikar@mahendra.info, [2]rajeswarik@mahendra.info,
[3]manjulap@mahendra.info, [4]smsunsurya@gmail.com

Abstract: *A partially autonomous vehicle system is presented in this paper. A miniature car model including the above features has been developed which showed optimum performance in a simulated environment. The proposed model is developed using machine learning technology. The system mainly consists of a Raspberry Pi, an Arduino, a Picamera or webcam, a sonar module, a Web interface and Internet modem. The Raspberry Pi and python programming is used for automation.*

Keywords: **Raspberry Pi, Machine learning, Python, autonomous vehicle**

1. INTRODUCTION

Autonomous vehicles (AVs) are increasingly gaining attention worldwide. Although autonomous driving is getting more and more important, this technology is still far from being mature. Our cities and roads are very unpredictable dynamic environments where multiple actors such as pedestrians, animals, street furniture or other vehicles coexist together. Hence it is needed to provide autonomous vehicles with robust perception systems in order to correctly understand the environment and be able to interpret what is happening in the surroundings to act in consequence. The main method for an AV to get information from the environment is through a variety of sensors [1].

Object detection is one of the main common tasks in computer vision. The main purpose is to detect an object which belongs to a particular set of predefined classes. Machine learning extends its focus nowadays in various fields of research as well as human welfare such as healthcare improvement [2], etc, this paper concentrate towards the designing of autonomous vehicle design for object detection and self-driving capability using machine learning techniques.

2. RELATED BACKGROUND

The fundamental building blocks of artificial intelligence are object detection and classification. Without the development and implementation of artificial intelligence, the concept of autonomous vehicle cannot become success. A major challenge is the integration of artificial intelligence with machine learning with autonomous operations. One solution is implemented using deep learning based software that uses a convolutional neural network algorithm for object tracking, detecting and classifying objects from raw data in real time [3]. Using convolutional neural network implementation objects can be tracked, detected and classified from video feeds supplied by UAVs (Unmanned Aerial Vehicle) in real time.

A new approach to detect and track Unmanned Aerial Vehicles (UAV) from a single camera is proposed in [4]. By using perspective transformation model background motion is estimated and then the moving object candidates are identified in the background subtracted image through deep learning classifier trained on manually labeled data sets.

Unlike geometric figures, objects in the real world are always irregular figures and the same object appears in various shapes when capturing from different angles or the object itself is changing its shape. Besides, images of objects in the real world environment are variants to illumination, rotation, scale and occlusion, which make object detection task more challenging. In recent years a large improvement in image recognition was made by series of CNN based solutions [5].

To gain a complete image understanding, it is not only concentrate on classifying different images but also try to precisely estimate the concepts and locations of objects contained in each image. This task is referred as object detection. A classifier is needed to distinguish a target object from all categories and to make the representations more hierarchical, semantic and informative for visual recognition. Support Vector Machine (SVM), AdaBoost and Deformable Part-based Model (DPM) are good choices [6].

3. SYSTEM ARCHITECTURE AND ESSENTIAL COMPONENTS

The block diagram for the hardware architecture of the car model is shown in figure below.

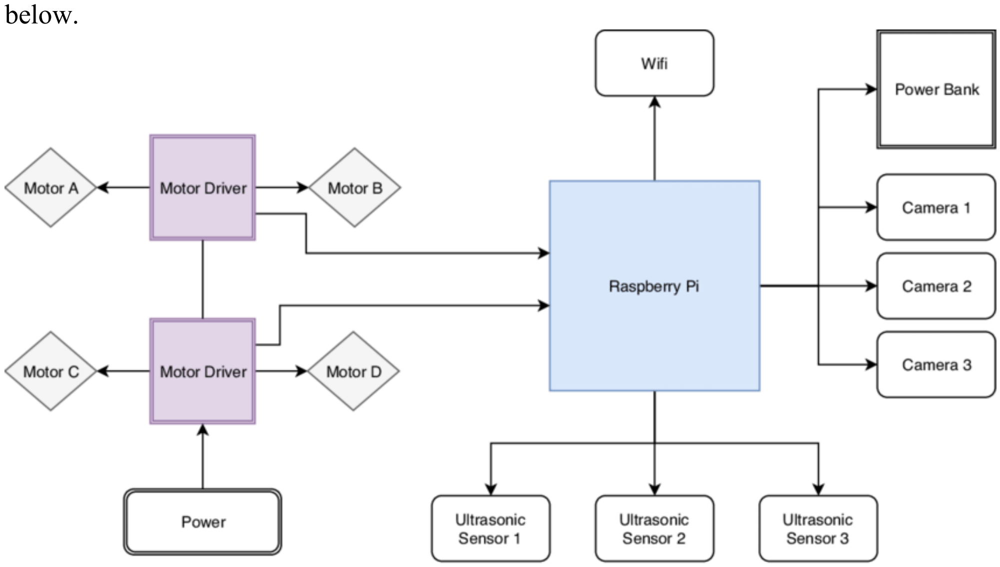

Figure 1. Abstract diagram of the hardware architecture of the car [7]

3.1. General purpose I/O GPIO pins

GPIO pins on Raspberry Pi enable us to interface physical devices like Motor Drivers, Sensors, and Cameras, etc., with the Linux processor and allow us to control their input through Python program in Raspberry Pi operating system. RPi.GPIO library is used to handle interfacing with pins. For GPIO pins, there are two different numbering systems. The Board uses the pins exactly as they are laid out on Raspberry Pi, and the BCM (Broadcom SoC numbering) differs for different versions of Raspberry Pi. Pins mode is set to BCM.

3.2. Motor driver

To control motors, two Dual H-Bridge L298 motor drivers are used. Motors are controlling wheels of car and are connected to Raspberry Pi through motor drivers. Motor driver controls the motor by two input pin, one to move forward and another for backward for a single tyre or motor. PWM pins of motor driver are used to control the speed of car.

3.3. Ultrasonic distance sensor

Collision detection and distance measurement of upcoming collision is a requisite for an autonomous car. There are three HC-SR04 ultrasonic distance sensors (sensorL, sensorC, sensorR) are used in Driving car model for collision detection in left, center and right direction respectively. There are four pins on HC-SR04 ultrasonic sensor that are connected to Raspberry Pi GPIO pins. VCC and GND are connected to voltage and ground respectively. An input signal is sent to TRIG pin from one of the GPIO pins, which triggers to send an ultrasonic pulse. The pulse waves bounce off objects in their path and some are bounce back to the sensor. If the sensor detects these return waves, a 5V signal is sent to the ECHO pin. The duration of the pulse is measured and thus distance is calculated from it.

3.4. Camera

The camera is the eye of the car. One Pi Camera module and two USB cameras are connected to Raspberry Pi's camera and USB ports.

3.5. Power management

A 12V 1.2A battery along with a power bank is attached as a power source which is charged by 5V 2.5A micro USB charge power adapter. Motor drivers and Raspberry Pi are powered separately. For testing purpose, when battery is low, 12V transformer is used to supply power to the components. Each of the above devices has specific pin configuration. These pin inputs are controlled through python code placed in Raspbian using RPi.GPIO, a module that controls Raspberry Pi GPIO channels. The center ultrasonic sensor is attached to pin 21 and pin 20. The right ultrasonic sensor is attached to pin 16 and pin 26. The left ultrasonic sensor is attached to pin 6 and pin 5. For tyre control, the pin connectivity is shown as per the table 1.

Table 1. Pin configuration for Motor drivers

Tyre	Forward pin	Backward pin	PWM pin
Front right	24	25	19
Front left	11	9	13
Back right	15	14	12
Back left	23	17	18

4. SYSTEM SET UP AND CAR MODEL ASSEMBLY

There are a few steps needed to set up Raspberry Pi before we can start working with it. The operating system is installed for Raspberry Pi. It is installed through loading NOOBS in microSD card. NOOBS (New Out Of the Box Software) is OS install manager for the Raspberry Pi.

Initially, Raspberry Pi is connected to monitor through HDMI cable and other devices through USB ports to complete all configuration. Later, Automatic Service Discovery can be introduced. Once everything is connected, since the Raspberry Pi does not have a power switch, on plugging the power adapter, it starts up by itself. When Raspbian boot process is completed, Raspbian home screen appears. Raspberry Pi's configuration is set from Menu- Preferences- Raspberry Pi configuration and then rebooted.

Two smart acrylic car chassis are used to for the assembling of car. The chassis is designed to fit various devices and compatible boards like Raspberry Pi, Motor Drivers, bread board etc. and contains holes to easily mount various sensors and cameras etc. It is capable of mounting 4 motors which are controlling 4 tyres of the car.

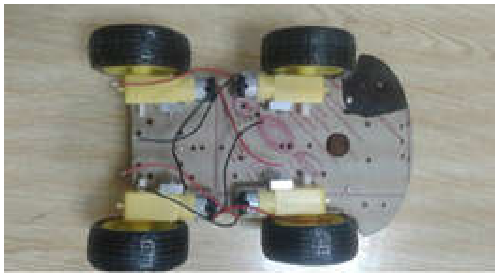

Figure 2. Tyres and motors on bottom chassis

Sensors are attached on the bottom front of one chassis with motors and motor drivers behind while right above on top front, there are three cameras with Raspberry Pi and bread board behind it. The second chassis is mounted above, two keep all the mess of wires hidden, with the help of copper pillars whose height can be differentiate to change the height between two chassis. On the top of second chassis, power bank and batteries are attached for power supply to Raspberry Pi and motor drivers.

Figure 3. Car model completely assembled

5. IMAGE PRE-PROCESSING AND DATASET ACQUISITION

The dataset is collected by driving car over track. The dataset consists of images taken from three cameras of the car which are stored in a folder and dataset.csv file which contains action taken on a state, path to the three images of left, right and center camera stored in images folder. The actions are 0 for forward, 1 for left, 2 for right and are stored against images of three cameras (left, right and center) of each state at certain time along with the images of next state.

There are different classification algorithms but here convolutional neural networks (CNN) is used for the proposed work to learn the datasets. The CNN make use of patterns and structural information in an image. The architecture has 18 layers, each of which four are convolutional layers. The car is controlled by PC or client. The client send actions to car server using web socket via network like move, forward, backward, right and left. For starting the server, the following steps are taken:

i. Clone the code for driving_action_car
ii. Configure config.json according to the car setting
iii. Run main.py
iv. Server is up and running

On the PC we have to follow the steps:

i. Clone driving_action
ii. Edit main.py and set the IP address
iii. Execute main.py.
iv. Now we can use W for forward, A for left, D for right and space for stop to control the car

Table 2. Observation of car model movement

Key used	Trial 1	Trial 2	Trial 3	Trail 4	Trial 5
W	Forward	Forward	Forward	Forward	Forward
A	Left	Left	Left	Left	Left
D	Right	Right	Right	Right	Right
Space	Stop	Stop	Stop	Stop	Stop

As though the car model movement is done with 100% accuracy, there are a few numbers of limitations in the model and are as follows:

i. Web-cam and Pi camera's images have different result. Pi camera images are found to have less contrast compared to web cam.

ii. Pi camera has a dedicated port and can give up to 30 fps. In Raspberry Pi there is 1 bus to handle USB port. Web cam can give 7fps at max for resolution of 160x120. Increasing the resolution impacts the fps.

iii. Ultrasonic sensor can give result at rate of 1 per 2 seconds. If we speed this to 1 per second, sensor will become unstable and it will start giving delay of 10+ seconds. Hence ultrasonic sensor is not considered in this work.

iv. Using three cameras on Raspberry Pi, heat up circuit after 3-5 minutes of streaming. Placing a fan on circuit can increase the streaming time.

6. CONCLUSION

The car model in this work is created by own environment from scratch that helps in understanding all the disadvantages one can face such as car on map with single light may perform different while when using four lights above it that is a brighter environment. Also supervised learning was done using convolutional neural network and achieves 100% accuracy. A few disadvantages listed have to be overcome in the future.

REFERENCES

[1] *Victor Vaquero Gomez, Object detection for Autonomous driving using deep learning, Dec 2015.*

[2] *Arul Srinivasan, A.Neela Madheswari, The Role of Smart Personal Assistant for improving personal Healthcare, International Journal of Advanced Engineering, Management and Science, vol.4, Issue.11, 2018.*

[3] *Matija Radovic, Offei Adarkwa, Qiaosong Wang, Object Recognition in Aerial images using convolutional neural networks, Journal of Imaging, June 2017.*

[4] *Dong Hye Ye, Jing Li, Qiulin Chen, Juan Wachs, Charles Bouman, Deep Learning for Moving object detection and tracking from a single camera in Unmanned Aerial Vehicles, Society for Imaging Science and Technology, 2018.*

[5] *Guangrui Liu, Real-time object detection for Autonomous driving based on Deep Learning, May 2017.*

[6] *Zhong-Qiu Zhao, Peng Zheng, Object Detection with Deep learning: A Review, IEEE Transactions on Neural networks and Learning systems, 2017.*

[7] *Syed Owais Ali Chishti, Hafiz M.Bilal Zaib, Sana Riaz, DrivingMatter: An Autonomous RC Car using Raspberry Pi, National University of Computer and Emerging Sciences, Pakistan, May 2018.*