NEURAL NETWORK BASED AUTONOMOUS CAR

SAHIL CHAUDHARY[1], RISHABH JAIN[2], MUKUL JAISWAL[3]

[1,2,3] Department of Computer Science & Engg, Inderprastha Engineering College, UP, INDIA

---***---

Abstract - *The project builds a neural network based autonomous car prototype using Raspberry Pi, ultrasonic sensor and Pi camera. Raspberry Pi is used as a processing chip. A Pi camera along with an ultrasonic sensor is used to provide necessary data from the real-world environment to the car. The car handle three tasks: self-driving on the lane, stop sign detection and front collision avoidance and therefore, capable of reaching the given destination safely and intelligently.*

Key Words: *Raspberry-pi, Pi-Camera, Neural Network, Open CV.*

1. INTRODUCTION

A self-driving car is a vehicle that must be able to navigate without human input to a pre-defined destination over a path that have not been adapted for its use. It makes its own decisions while driving, able to cope with all situations and continuously keep learning from the environment and decisions on that environment to maximize its output and accuracy over time. A fully autonomous car is able to handle all the driving tasks in all driving modes under every environmental condition, just like human drivers do.

The development of autonomous cars is uprising rapidly in automotive industry. They are increasingly catching attention worldwide because prospective of this technology is clear as it will dramatically change transportation by minimizing traffic jam and accidents, increasing efficiency and allowing faster speed. Autonomous cars are predicted to increase traffic flow and lower fuel consumption which will reduce contamination in urban areas by improving driving and significantly reduce needs for parking space. In addition, autonomous cars will speed up people and freight transportation, as well as increase the security, specifically a significant reduction in traffic collisions by reducing the human error.

The car would drive itself from one place to the other on its own it would possess integrated features like lane-detection, obstacle-detection and stop sign detection.

2. LITERATURE SURVEY

A driverless car is a self-driving vehicle that can drive itself from source to destination without assistance from a driver. According to urban designer and futurist Michael E. Arth, driverless electric vehicles—in conjunction with the increased use of virtual reality for work, travel, and pleasure— could reduce the world's 800,000,000 vehicles to a fraction of that number within a few decades. Arth claims that this would be possible if almost all private cars requiring drivers, which are not in use and parked 90% of the time, would be traded for public self-driving taxis that would be in near constant use. Driverless passenger car programs include the 800 million EC EUREKA Prometheus Project on autonomous vehicles (1987-1995), the 2getthere passenger vehicles (using the FROG-navigation technology) from the Netherlands, the ARGO research project from Italy and the DARPA Grand Challenge from USA [9].

The control mechanism of an autonomous car consists of three main blocks as shown below:

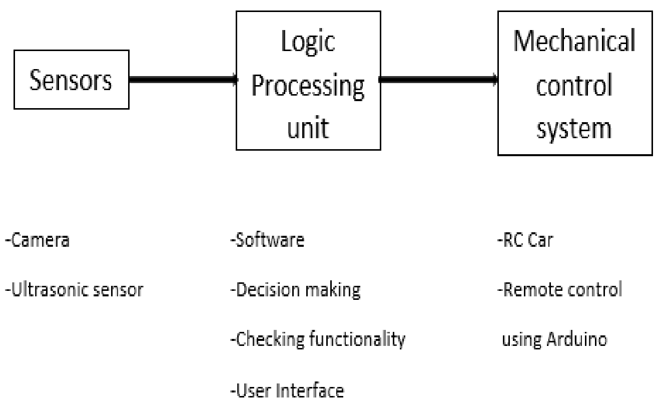

Fig -1: Control mechanism of Car

Most autonomous vehicle projects made use of stock cars and modified them, adding "smart" hardware to create automated cars. The advantage of using stock cars is the ease of obtaining the car through sponsors. The stocks cars help convey the message autonomous vehicles are not science fiction anymore and these systems can be implemented on normal cars.

3. METHODOLOGY

In order to make the car autonomous in working environment, Artificial Intelligence, Image Processing and Machine learning techniques are used. There are various approaches to gain knowledge of environment and that by Image Processing or simple line tracking. Learning of car is done by basically two approaches, supervised and reinforcement, which are discussed in detail in coming section.

In both approaches, we focused on deep CNN model. We are more interested to know what kind of features CNN extract for decision.

3.1 Supervised learning

The external feedback is used by learning functions in supervised learning for mapping input to output. This external feedback acts as a teacher for the learning function. Convolutional Neural Network (CNN)is used which is one of the deep learning architectures, for the supervised learning.

The big thing of any deep learning model is getting feature representations through back-propagation. Unlike other classifiers which need hand-engineered features, CNN knows which information like color or shape is important to do the task. The motivation of using the CNN over another neural network will be discussed later.

3.1.1 Image Pre-processing

In total we have 1100 samples from which training data has shape (819,30,45) because it has 840 training sample images and test data has shape (398,59,40) because it has testing 376 sample images. The length of output class in 3 which are the actions the car can take i.e. include forward, left and right. The images collected are in RGB form that possess three layer and the size of the images are 320x240.

Resize Each 320x240 image of test and train sets is converted into matrix of size 1x24x32 which is sent to neural network. This resize is done in order to minimize the learning time of hyper parameters.

Rescaling is used to convert the data to type float32 before sending to the network. Image pixel values are now re-scaled to 0-1 range.

Image equalization because, we have collected the images by driving the car in real track which would have possibility of low contrast places or different illumination conditions which can deviate the results. Hence to normalize under different illumination conditions, Image histogram equalization is used. After that, image is smoothed to reduce noise introduced by histogram equalization.

Image labeling The Fig. 2 displays the image labeling scheme of for each action over three images. When forward action is taken, the center image is labeled as forward, left image is labeled as right and right image is labeled as left (Fig. 2a).

The given pattern of image labeling is followed because value is found on the basis of center camera image, so when a car reaches at a state, where the center camera image is like we have left camera image the appropriate action at that stage would be left. Labeling scheme for left and right actions is shown in (Fig. 2b) and (Fig. 2c)

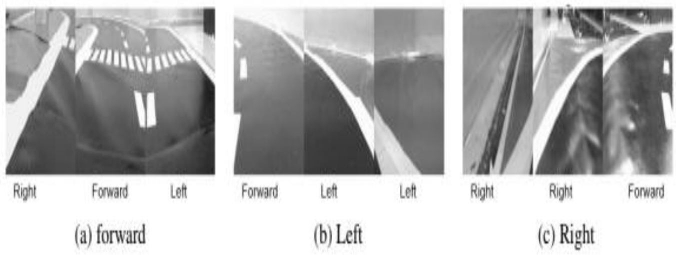

Fig -2: Image Labeling

Class labeling because our data is categorized into various fields, we need to convert it to a vector of numbers. Using One-hot encoding, we get a Boolean row vector of 1x3 for each image which consists of all zero values expect for the class it represents. For example, for forward class, it will be [1 0 0].

Class weights for each class, we have huge number of samples in dataset e.g. forward class could have 600 while right and left class have 400 and 300. In such condition, we need to assign class weights to each class and there is a urgent need to standardize the data so that each class has equal distribution of samples while training.

3.1.2 Deep Learning for Autonomous Driving

we need to classify the state information because we want to predict the best actions, on a given state of the car in environment, i.e. cameras images with respect to actions. There are many classification algorithms are their but we are using Convolutional Neural Networks (CNN) model for our problem.

The reason for using the CNN over other classification techniques is that it makes an efficient use of patterns and structural information in an image. As we know, in RNN, output dependency on all previous values results very bad for images. Moreover, less parameter and memory requirements. As we are targeting our model to our car with low computing power, CNN is the best option that we can opt.

CNN Architecture (DMNet) Figure 3 shows the best CNN model architecture that fits our data with maximum accuracy. We have named it DrivingMatterNet (DMNet). The architecture has 18 layers, of which 4 are convolutional layers. It has around 21,000 parameters.

The accuracy graph is shown in fig 3. We have got 89% training accuracy and 73% validation accuracy.

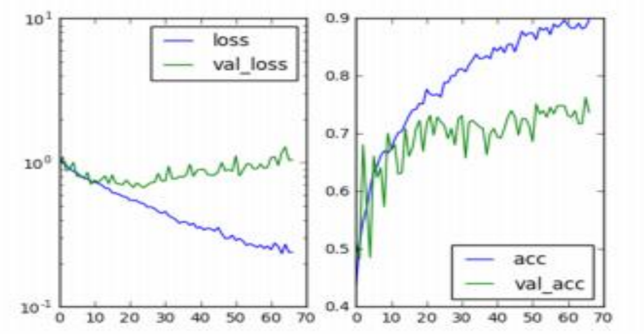

Fig -3: Accuracy and Loss on Training and Testing Data Points

3.2 Deep Reinforcement learning

Deep reinforcement learning is a mixture of reinforcement learning and deep learning. In reinforcement learning, there is no external feedback or answer key to our problem but agent (the car in our case) still has to decide how to act in a certain situation or task. So, to perform its task, agent learns from its experience. Still, some knowledge is provided about the goodness or badness of an action on a state, known as reward (or punishment in case of negative reward).

Reinforcement models employs different dynamics unlike supervised learning i.e. rewards or punishment to reinforce unprecedented types of knowledge. So, the car has a current state s of the environment E, the car can perform an action a which will transition it to next state s′. Each action is evaluated by a reward r.

The set of action car takes, define the policy Π and reward it returns, defines the value v. The goal is to select the correct policy Π to maximize the rewards. This just defines our solution in reinforcement learning scenario in Markov Decision Process framework, so we have to maximize the value of

$$E(r_t | \Pi_t, s_t)$$

for all possible states s at time t.

4. IMPLEMENTATION

The raspberry-pi which is the central controller would be mounted on the car. The ultrasonic sensors would be placed on the front bumper of the car, while the pi-camera module would be placed on the roof of the car. The motor-driver ICs are responsible for the operation of motors and thus the motion of the car.

The ultrasonic sensors placed at the bumper of the vehicle would be used to detect any obstacle in front of the car and take according actions. Whenever there is any obstacle in front of the car and lies within the pre-determined distance from the car, the raspberry-pi orders the motor driver ICs to stop supplying power to the wheels and hence stops the motion of the car depending upon the proximity of the obstacle.

The distance measured is also displayed on the output window of the program. The next step is detection of lane and stops sign. For these, we use the principles of image processing. For detection of the stop sign, HAAR cascade classifier was used in Open CV. As we know, Open CV provides classifier a well as a detector.

To initiate the HAAR-cascade successfully, we uploaded positive and negative images of the data (stop sign). The positive image would be the image of the target data to be recognized and the negative image would be any image without the target data. On observing and obtaining matches with the region of interest of the images, the signals and signs are identified.

The region of interest is mainly the bounding box for the signs and signals. On feeding the necessary and appropriate data, the module is trained using the fundamentals of machine learning. Once the training is done, during the run, the decisions of lane observation, stop sign detection is carried by the use of neural networks to provide undisrupted motion of the vehicle. For lane detection, monocular vision method is used as shown in the figure below.

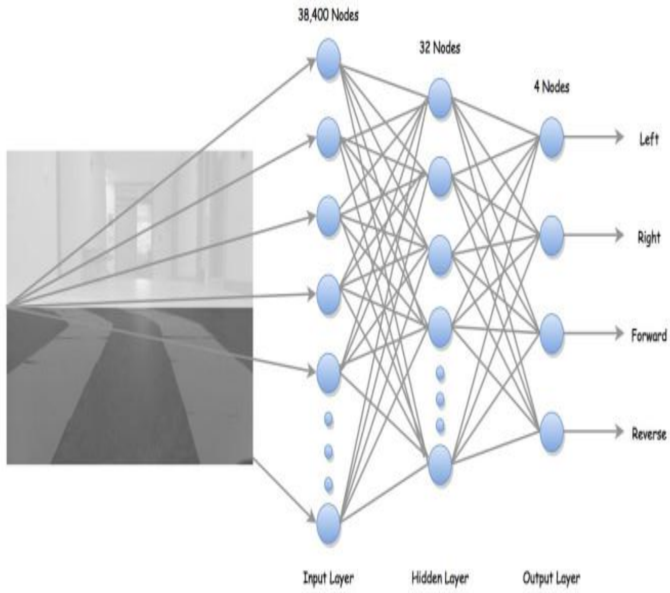

Fig -4: Monocular Vision method for Lane Detection

5. EXPERIMENT AND ANALYSIS

Prediction on the testing data gives an accuracy of 73% compared to the accuracy of 89% that the training sample data returns.

Haar features are rotation sensitive by nature. However, in this project, rotation is not a concern as the stop sign is fixed object, which is also a general case in real world environment.

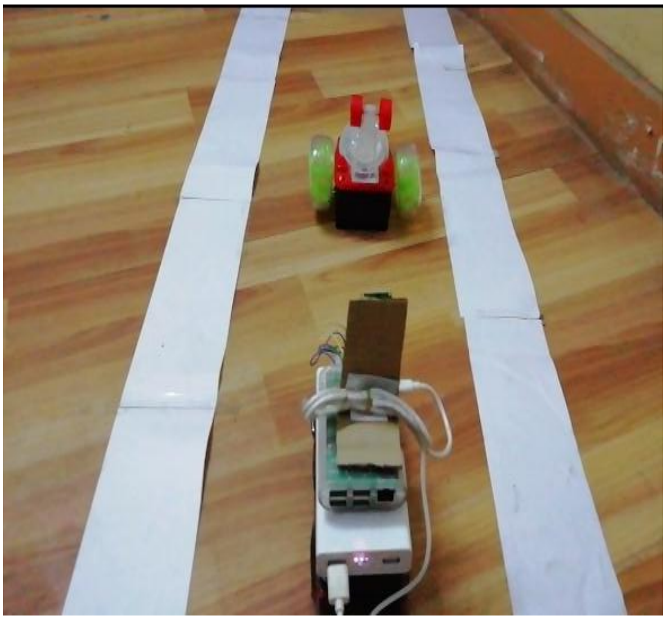

Fig -5: Front Collision Avoidance

For distance measurement purpose, the ultrasonic sensor is used to calculate the distance to an obstacle coming in front of the RC car and gives accurate results when taking proper sensing angle and surface conditions into considerations.

Fig -6: Stop Sign Detection

On the other hand, Pi camera provides "good enough" measurement results. As we know the actual distance, we know when to stop the RC car.

We have tested for various threshold values. Finally, we have set threshold value to 30 centimeters. As soon as the camera detects threshold value of 30 centimeters, it stops the car instantly.

Experimental results of distance measured using pi camera are shown as below:

Order	1	2	3	4	5	6
Actual Value (cm)	15	20	25	30	35	40
Measured value (cm)	16.0	20.9	23.8	26.3	35.9	39.9

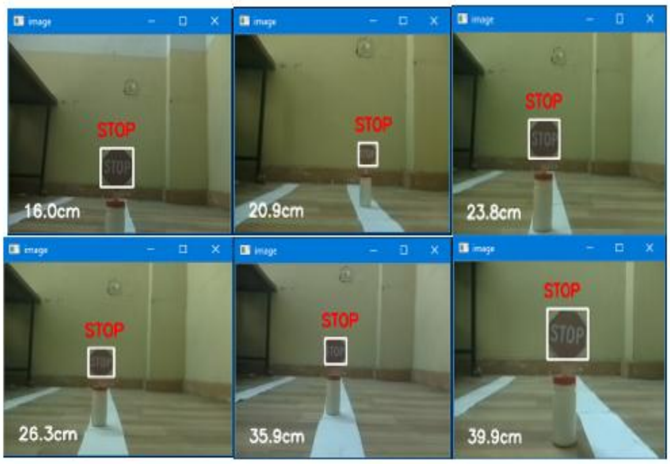

Fig -7: Experimental results of distance measured by Pi camera

Fig -8: Lane Detection

Overall, the RC car could successfully navigate on the lane with the ability to avoid front collision, and respond stop sign accordingly.

6. CONCLUSIONS

The autonomous car would surely prove out to be a boon in the automation industry and would be preferred over many traditional techniques. They could be used for patrolling and capturing the images of the offender. As they won't require any drivers, the accidents caused by the carelessness of the goods carrier vehicles would be reduced and would ensure better logistic flow.

Buses for public transport would be more regulated due to minimal errors. Hence, due to greater autonomous nature and efficiency, an autonomous car of this nature can be practical and is highly beneficial for better regulation in the goods and people mover's section.

7. FUTURE SCOPE

To enhance the performance and ensure practicality of the car, the efficiency and processor speed need to be raised. A camera of better resolution would also be required as the scenes keep changing rapidly in the real world. Also, the speed of the car should decrease gradually so that the passengers aren't hurt and the goods aren't damaged.

REFERENCES

[1] Raspberry pi 3 b model data sheet [Online], available at http://docseurope.electrocomponents.com/webdocs/14ba/0900766b 814ba5fd.pdf

[2] Pi camera module datasheet [Online], available at http://docseurope.electrocomponents.com/webdocs/127d/ 0900766b 8127db0a.pdf

[3] Ultrasonic module Hc- sr04 model datasheet [Online], available at http://www.micropik.com/PDF/HCSR04.pdf

[4] Haar Feature-based Cascade Classifier for Object Detection, available at http://www.swarthmore.edu/NatSci/mzucker1/opencv-2.4.10-docs/modules/objdetect/doc/cascade_classification.html

[5] zhengludwig.wordpress.com/projects/self-driving-rc-car

[6] https://coding-robin.de/2013/07/22/train-your-own-opencv-haar-classifier.html

[7] www.efxkits.com/blog/raspberry-pi-technology-with-applications

[8] https://medium.com/software-engineering-for-the-real-world/simultaneous-video-processing- and-storing-using-a-raspberry-pi-3-4d4edf6d4778

[9] www.123seminarsonly.com/Seminar-Reports/013/49379095-DRIVERLESS-CAR-DOCUMENTATION.pdf

BIOGRAPHIES

Sahil Chaudhary
Student at Inderprastha Engineering College, Ghaziabad.

Rishabh Jain
Student at Inderprastha Engineering College, Ghaziabad.

Mukul Jaiswal
Student at Inderprastha Engineering College, Ghaziabad.