Vision-Based Steering Control and Speed Assistance for Vehicles

Name - Chinmay kumar Das

Branch - I & E (Sec A)

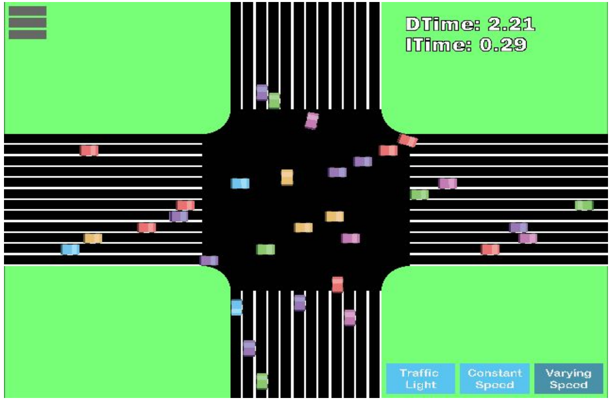
Regd no - 1401106259

Content

- ❖ Introduction
- ❖ Objective
- ❖ System design
- ❖ Input unit
- ❖ Processing unit
- ❖ Block diagram
- ❖ Circuit diagram
- ❖ Result
- ❖ Future work
- ❖ Reference
- ❖ Q&A

Introduction

- Lack of autonomy in Indian Transportation system
- Lots of accident happen due to human errors
- These accident can be avoid using autonomous car
- Stanley - The first Self Driving Car



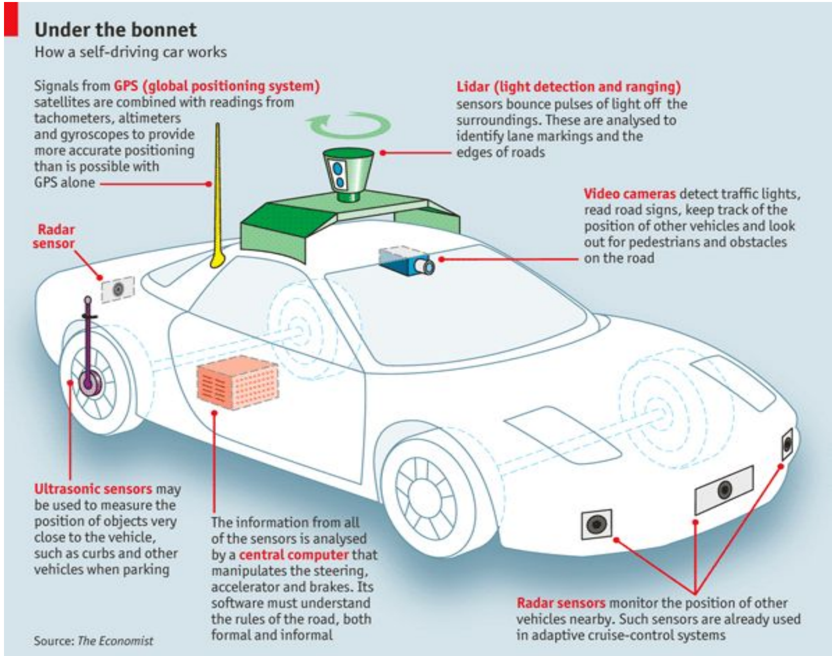
Source: http://imgur.com/gallery/h1RZT

Source: robots.stanford.edu/talks/stanley/

Autonomous Car

Key components of Self Driving Car -

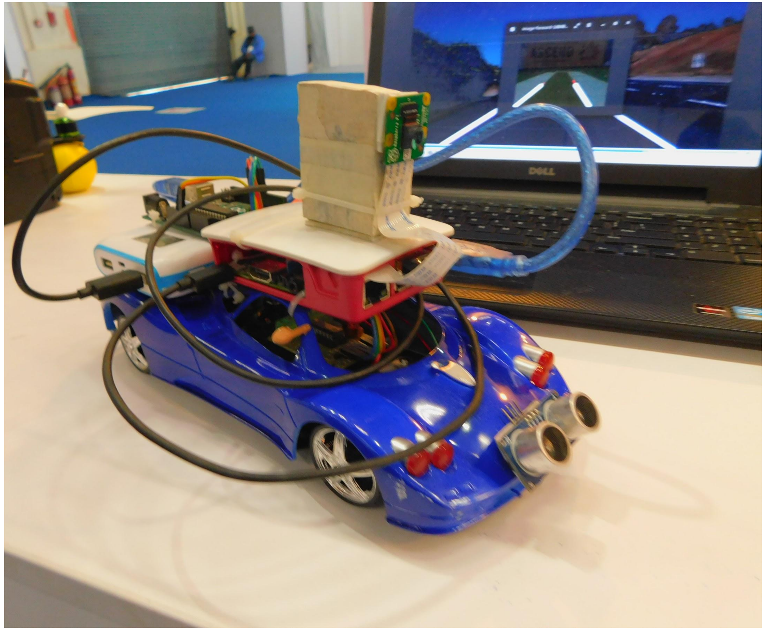
- Radio Detection And Ranging(RADAR)
- Ultrasonic Sensor
- Laser Detection And Ranging(LIDAR)
- GPS
- Video Camera
- Central Computer



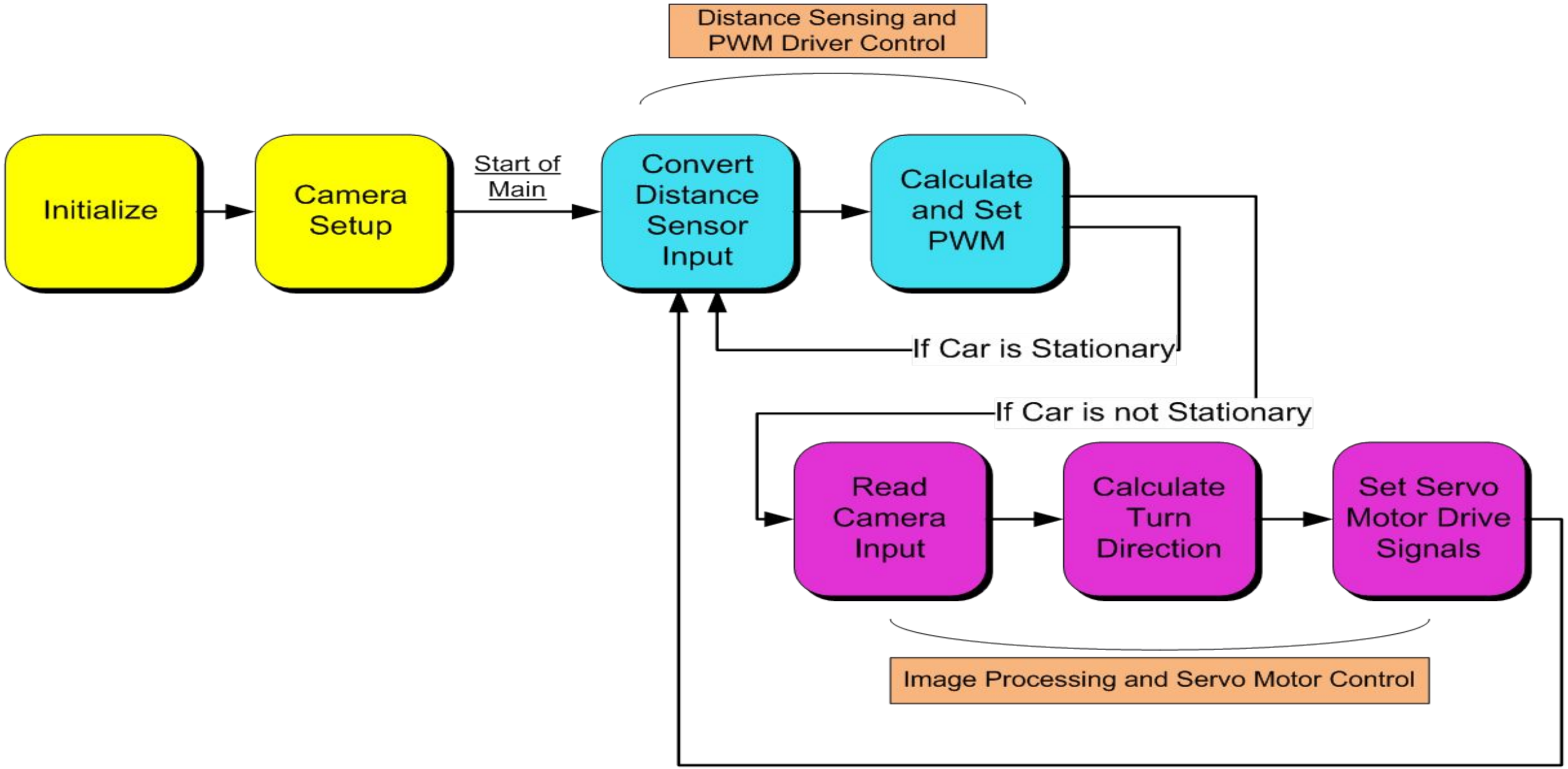
Objective

<u>Modify a RC car to handle four tasks</u>:

- Self-driving on a lane track
- Stop sign and traffic light detection
- Front collision avoidance
- Automatic speed control in the track



Block Diagram of Self Driving RC car



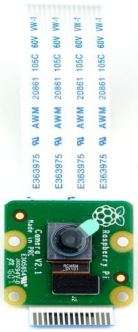
System Design

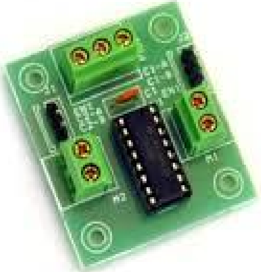
The system consists of three subsystems:

- Input unit
  - Camera
  - Ultrasonic sensor
  - Motor driver circuit
  - Power supply
- Processing unit
  - Raspberry Pi
- RC car unit.

Input Units

- Raspberry pi
- Raspberry pi camera
- Arduino
- Motor Driver circuit
- Ultrasonic sensor
- RC car
- Power bank





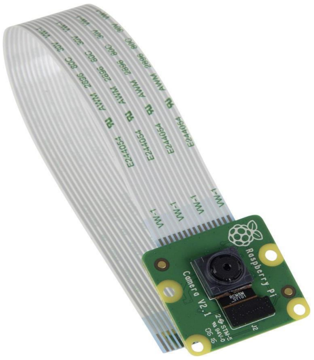
Raspberry Pi -

The **Raspberry Pi** is a tiny and affordable micro computer.

- SoC: Broadcom BCM2837 (System on a chip)
- CPU: 4× ARM Cortex-A53, 1.2GHz
- RAM: 1GB
- Storage: microSD
- GPIO: 40-pin header

Power Supply -

- Input Voltage - DC 5V
- Input Current - 500 / 1000 mA
- Output Voltage - 5V
- Output Current - 2A(Max)



Raspberry Pi Camera -

- Resolution - 8 MP
- Video modes - 1080p30, 720p60 and 640 × 480p60/90
- Sensor - Sony IMX219
- Focal length - 3.04 mm

Arduino -

- Microcontroller board based on the ATmega328P
- 14 digital input/output pins
- 16 MHz quartz crystal
- USB connection
- Power jack

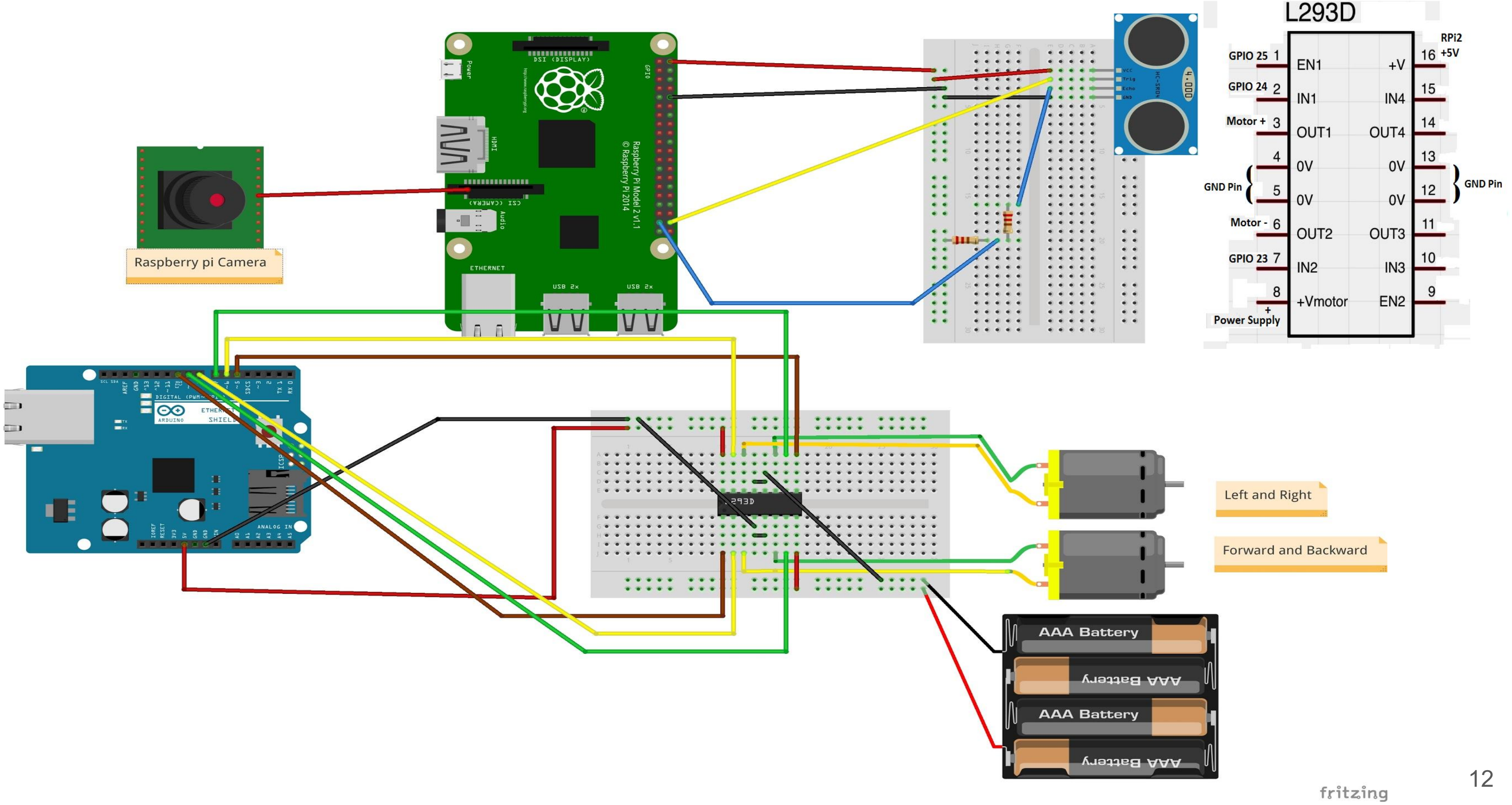
Motor Driver Circuit -

- Driver: L293D
- power supply: +5V ~ +46V
- Io: 2A
- Max power: 25W (Temperature 75 celsius)

Ultrasonic Sensor (HC-SRO4)-

- Supply voltage 5V
- Ultrasonic Frequency 40 kHz
- Maximal Range 400 cm
- Minimal Range 2 cm
- Trigger Pulse Width 10 μs
- Ultrasonic Sound velocity = 343 m/s

Circuit Diagram



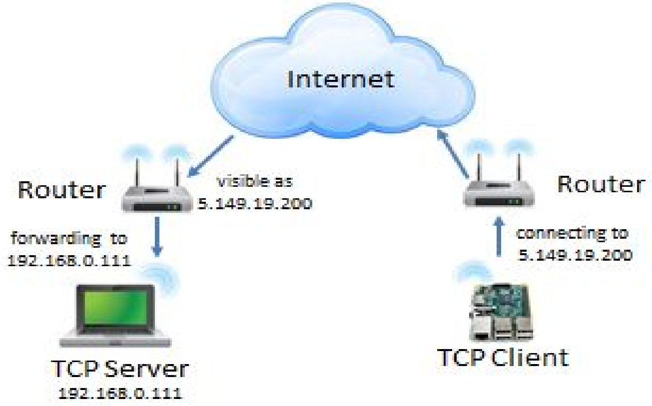
Processing Unit

<u>The processing unit handles multiple tasks</u>:

- ➢ Send data from to computer for training process.
- ➢ Neural network training and prediction(steering).
- ➢ Object detection(stop sign and traffic light).
- ➢ Distance measurement(collision avoidance).
- ➢ Sending instructions to Arduino through USB connection.

TCP Server

- A server program that runs on the computer.
- Receive streamed image frames and ultrasonic data from the Raspberry Pi.
- Image frames are converted to grayscale and are decoded into matrix.
- Train the neural network and then send the model to Raspberry Pi.



Object Detection

- Detect stop sign and traffic light (Red and Green)
- Used Haar cascade classifiers for object detection.
- Calibrate the camera for distance measurement.

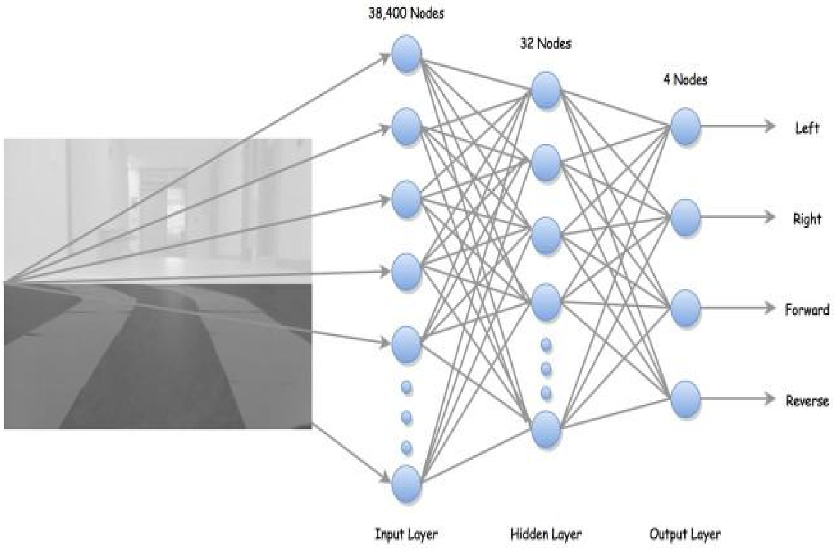
source:https://zhengludwig.wordpress.com/

Driving on a lane track by Automatic steering control -

- Create path by A4 size paper.
- Define region of interest of the track by camera angle
- Capture the image in grayscale and store it while manually driving
- Label the image according to the steering action
- Define the parameter for neural network
- Feed the input parameters to the neural network algorithm
- Save the model
- Run the model in Raspberry Pi in Lane track

Neural Network architecture and it parameters -

- Input layer has **38,400** (320×120) nodes and **32** nodes (chosen arbitrarily) in the hidden layer, **4** nodes in the output layer (left, right, forward and reverse).
- Using Back Propagation algorithm to get the result.
- Once the network is trained, it only needs to load trained model afterwards, thus prediction can be very fast.
- It happens in real time with less latency.



Workflow of Back Propagation Algorithm -

- Set the input parameters - 1.Input layer size 2.Hidden layer size 3.Numbers of lebel
- Set the training parameters -
  - Image array
  - Labels(Forward,Backward,Left,Right)
- Taking initial weight by random with extra bias 1 in each input layer.
- For each layer calculate Output of individual neuron = $a_j^i = \sigma(\sum_k (w_{jk}^i \cdot a_k^{i-1}) + b_j^i)$
- Cost function = $J(\Theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} y_k^{(i)} \log h_\theta(x^{(i)})_k + (1-y_k^{(i)}) \log(1-h_\theta(x^{(i)})_k)\right]$
- For optimisation of cost function we need to derivative w.r.t weight. $\frac{\partial J_\Theta}{\partial \Theta} = 0$
- Then save the model with optimize parameters for test.

Notation used

- x - Input
- y - Output
- theta - weight
- m - No of Training example
- i - index of units in a layer
- j - index of layers
- k - index of labels
- σ = Activation function (Sigmoid function)

w^i_{jk} is the weight from the k^{th} neuron in the $(i-1)^{th}$ layer to the j^{th} neuron in the i^{th} layer,

b^i_j is the bias of the j^{th} neuron in the i^{th} layer, and

a^i_j represents the activation value of the j^{th} neuron in the $i^t h$ layer.

How to apply this in Real life

- Using image processing technique for better accuracy.
- Then apply the same algorithm with after processing the images or video.
- Then control the steering according to it.

Result

- Prediction on the testing samples returns high accuracy if more amount of image collected.
- In actual driving situation, predictions are generated about 10 times a second (streaming rate roughly **10 frames/sec**).
- Haar cascaded classifier is rotation sensitive. In this project, however, rotation is not a concern as both the stop sign and the traffic light are fixed objects, which is also a general case in real world environment.
- Ultrasonic sensor is only used to determine the distance to an obstacle in front of the RC car and provides accurate results , On the other hand, Pi camera provides good enough measurement results after calibration of camera.

Application

- Autonomous Transportation Systems

  inside industry
- Autonomy in Agricultural field
  - Automatic Weeder Machine
- Reduce human error
  - Prevent accident

Source: google.com

Future Work

- Implement the set up in the real automobile or on any robot.
- Integrate with powerful hardware like NVIDIA.
- Integrating with Robot Operating System (ROS) for control the hardware.
- Integrating GPS and Compass module to be added for long distance travel.

Reference

1. “End to End Learning for Self-Driving Cars” by Mariusz Bojarski , Davide Del Testa, Daniel Dworakowski, Bernhard Firner. URL: https://arxiv.org/pdf/1604.07316.pdf
2. “Speed Assistance and Localization for Inner-City Vehicles” by Miguel Angel, Olivares-Mendez, Jose Luis, Sanchez-Lopez 1 , Felipe Jimenez. URL: www.mdpi.com/1424-8220/16/3/362/pdf
3. “Visualizing and Understanding Convolutional Networks” by Matthew D. Zeiler and Rob Fergus. URL:https://www.cs.nyu.edu/~fergus/papers/zeilerECCV2014.pdf
4. “A Camera Calibration Method for Obstacle Distance Measurement Based on Monocular Vision” by Lin Chenchen. URL: http://ieeexplore.ieee.org/abstract/document/6821579/authors
5. “CS231n Convolutional Neural Networks for Visual Recognition”. URL: cs231n.stanford.edu/
6. https://www.raspberrypi.org/
7. https://www.udacity.com/course/artificial-intelligence-for-robotics--cs373
8. https://www.coursera.org/learn/neural-networks
9. https://create.arduino.cc/projecthub/onyx/ultrasonic-sensor-alarm-1ec0f3
10. opencv.org/

Q&A

Thank you