Asian Journal of Applied Science and Technology (AJAST)
Volume 1, Issue 7, Pages 92-94, August 2017

Self Driving Car System Using (AI) Artificial Intelligence

Md Najmus Saquib[1], Mr.Javed Ashraf[2] and Col. (Dr) O.P.Malik[3]

[1]Department of Electrical Electronics & Communication Engineering, Alfalah University (Haryana), India.
[2]Associate Professor, Department of Electrical Electronics & Communication Engineering, Alfalah University (Haryana), India.
[3]Professor (HOD), Department of Electrical Electronics & Communication Engineering, Alfalah University (Haryana), India.

Article Received: 24 July 2017 *Article Accepted: 08 August 2017* *Article Published: 10 August 2017*

ABSTRACT

This research paper regarding the automatic vehicle driving system without driver. The purpose of this project to minimize the road accidents and cost effective transportation. According to google survey report every year thousands of people died in a car accident only due to human mistake like his behavior of vision, auditory perception and other road user also driving after taking alcohol. Few years ago google managed an automatic car but their cost is very high and training method is different but we using A 3D camera and radar which is mounted top of the car which can take 3D images of surrounding and two camera left and right side that takes image send the data to local server through wireless transmitter. By using GPS it navigate across the road. The steering angle is controlled by stepper motor which is connected through a computer CPU and GPU are already trained through millions of images taken from real road transportation in all traffic and weather conditions.

Keywords: Artificial Intelligence, Neural network, Hardware Sensors and Google Map.

1. INTRODUCTION

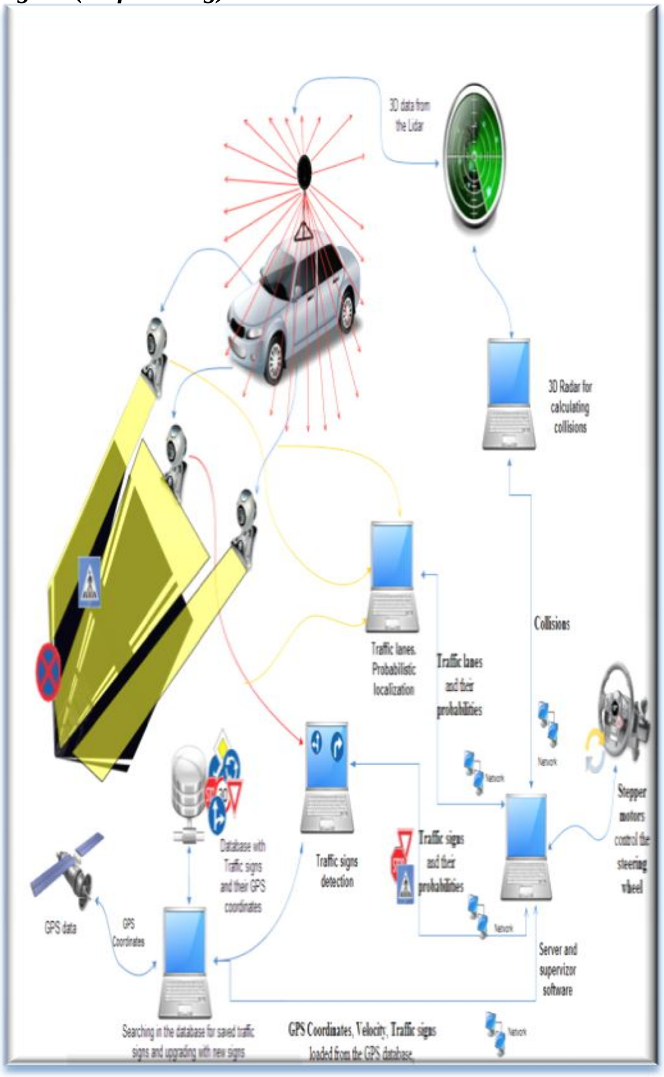
It will be no longer when driverless car would be running across the road when AI (Artificial Intelligence), Machine learning, deep learning and Neural technology become popular. We just tap on our smartphone APP and driverless car is parked on your door. By using voice command you will tell the car for desired destination. Car will use GPS, Camera and google map and Navigate itself across the road corner and traffic signal.

2. WORKING PRINCIPLE

The software is able to identify recognize the traffic signs and store them in a common database using Google Maps and GPS. The GPS software component records the signs and direction of travel from that area. Every car moving in the traffic using this software will register the new signs detected and will modify the degree of confidence and degree of recognition for other users. Another software component is able to recognize the demarcation lines between lanes. It uses three cameras to calculate exactly or using probabilities the position of the car on the road, where the roadsides are and to propose a new direction even in the absence of traffic signs for the next seconds. Another part of the software uses Artificial Intelligence to detect other car fingerprints from webcam images. The algorithms were implemented parallel and distributed. I developed a management software system based on semaphores that allows data processing and supervision from 3 different computers with multiple cores. This project contains also a home-made LIDAR – 3D radar and a software using OpenGLto create a 3D environment in which the car navigates. By using it, the car will take the decision of avoiding obstacles. The 3D radar helps the entire software system to increase the confidence of decision.

I also used a different training technique for the network that is drive car about 100 km with EEG sensor and took more than 1000 sample of human brain when he took any action like Acceleration, Break, left and Right turn, Every situation is mapped with the brain samples and supervised the neural network to took decision for the movement of steering and breaking.

Figure (Self Driving) Car


3D data from the Lidar
3D Radar for calculating collisions
Collisions
Traffic lanes. Probabilistic localization
Traffic lanes and their probabilities
Network
Network
Stepper motors control the steering wheel
Database with Traffic signs and their GPS coordinates
Traffic signs detection
Traffic signs and their probabilities
Network
GPS data
GPS Coordinates
Server and supervizor software
Searching in the database for saved trafic signs and upgrading with new signs
GPS Coordinates, Velocity, Traffic signs loaded from the GPS database
Network

Training equations.

The transition (State equation**)**

$X1 \sim \mu\ x1$ and $(Xk\ |Xk{-}1, (for\ k > 1)$

Observation equation

$(yk\ |Xk)$Posterior distribution (we want to find the posterior probability)$(X1{:}n\ |Y1{:}n)$Using Bayes:

$p\ X1{:}n\ Y1{:}n = (\boldsymbol{X1{:}n})\boldsymbol{p}(\boldsymbol{Y1{:}n})$

The joint probability can be written as:

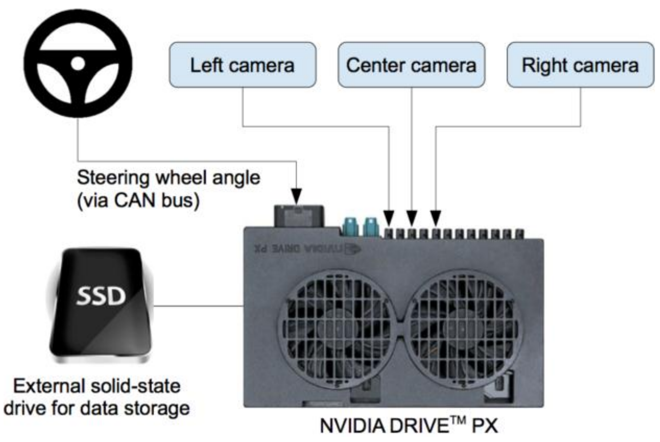
$p\ X1, Y1{:}n = p\ X1{:}n{-}1, Y1{:}n{-}1\ f\ Xn\ Xn{-}1\ g(Yn\ |Xn)$

Sop $X1{:}n\quad Y1{:}n) = p\quad X1{:}n{-}1,\quad Y1{:}n{-}1{,}n\quad Xn{-}1)(Yn|Xn)(Yn|Y1{:}n{-}1)$

Update step

$\boldsymbol{p\ Xn\ Y1{:}n = g\ Yn\ Xn\ p(Xn|Y1{:}n{-}1)}$=g$(\boldsymbol{Yn|Y1{:}n{-}1})$

Block Diagram


Left camera
Center camera
Right camera
Steering wheel angle (via CAN bus)
SSD
External solid-state drive for data storage
NVIDIA DRIVE™ PX

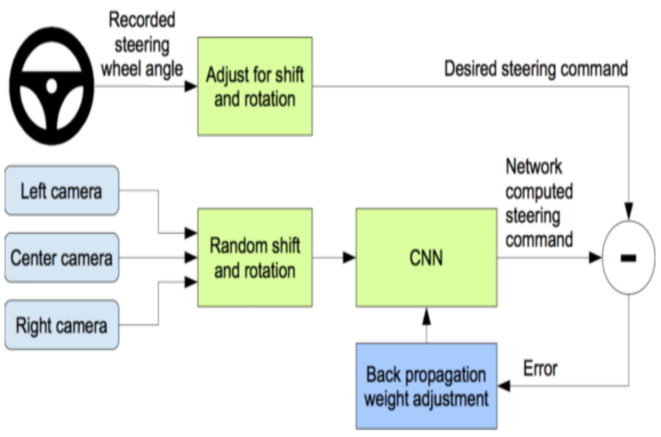
High-level view of the data collection system

Figure 2 shows a simplified block diagram of the collection system for training data of the car three cameras are mounted behind the windshield of the data-acquisition car, and time stamped video from the cameras is captured simultaneously with the steering angle applied by the human driver. The steering command is obtained by tapping into the vehicle's Controller Area Network (CAN) bus. In order to make our system independent of the car geometry, we represent the steering command as $1/r$, where r is the turning radius in meters. We use $1/r$ instead of r to prevent a singularity when driving straight (the turning radius for driving straight is infinity). $1/r$ smoothly transitions through zero from left turns (negative values) to right turn (positive values).

Training data contains single images sampled from the video, paired with the corresponding steering command (1/r). Training with data from only the human driver is not sufficient; the network must also learn how to recover from any mistakes, or the car will slowly drift off the road. The training data is therefore augmented with additional images that show the car in different shifts from the center of the lane and rotations from the direction of the road.

The images for two specific off-center shifts can be obtained from the left and the right cameras. Additional shifts between the cameras and all rotations are simulated through viewpoint transformation of the image from the nearest camera. Precise viewpoint transformation requires 3D scene knowledge which we don't have, so we approximate the transformation by assuming all points below the horizon are on flat ground, and all points above the horizon are infinitely far away. This works fine for flat terrain, but for a more complete rendering it introduces distortions for objects that stick above the ground, such as cars, poles, trees, and buildings. Fortunately these distortions don't pose a significant problem for network training. The steering label for the transformed images is quickly adjusted to one that correctly steers the vehicle back to the desired location and orientation in two seconds.

Figure 3 shows a block diagram of our training system. Images are fed into a CNN that then computes a proposed steering command. The proposed command is compared to the desired command for that image, and the weights of the CNN are adjusted to bring the CNN output closer to the desired output. The weight adjustment is accomplished using back propagation as implemented in the Torch 7 machine learning package and Unity simulator. Unity is an open source of online programming, designing and simulation of project. Images are cloned from GitHub a website which provide stored real time images taken by google self-driving car.


Recorded steering wheel angle
Adjust for shift and rotation
Desired steering command
Left camera
Center camera
Right camera
Random shift and rotation
CNN
Network computed steering command
-
Back propagation weight adjustment
Error

Once trained, the network is able to generate steering commands from the video images of a single center camera. Figure 4 shows this configuration.

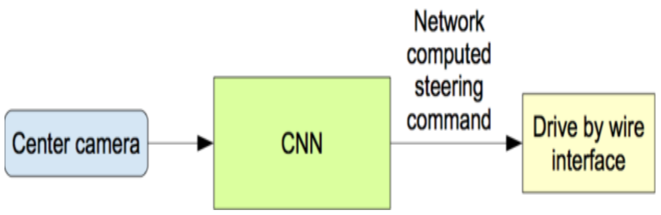


Figure 4: The trained network is used to generate steering commands from a single front-facing center camera

Sampling from a set of probability densities $\{\pi n\ X1{:}\}$ Sampling N independent random variables $X1{:}ni$ from each probability distribution and Estimating the distribution using the next formulas: $\varepsilon n\ X1{:}n = 1N\delta X1{:}ni\ (X1{:}n)\ Ni{=}1\delta X1{:}ni$ $(X1{:})$ is the difference at each sample. Sometimes these distributions are intractable in closed-form and are specially these cases in nonlinear and non-Gaussian models. The particle filter is based on the calculation of an importance density called weight of the model density. In the following formula I am using $(X1{:}n)$ to describe the weights of the

particles and (11:n) to describe the importance density. Zn is a normalization term.
$\varepsilon n\ X1{:}n = (X1{:}n)\ Zn\varepsilon n\ X1{:}n = (X1{:}n)\ qn(11{:}n)Zn\varepsilon n\ X1{:}n = Wn\ i\delta X1{:}ni\ X1{:}n\ Ni{=}1$ and $Wni = Wn\ X1{:}niWn\ X1{:}njNj{=}1$

So the selection importance distribution
$qn\ X1{:}n = qn -1\ X1{:}n-1\ (Xn\ |X1{:}n-1)$
$q1\ (x1)$ is the first step - a simple sample from the original/initial distribution
$(Xk\ |X1{:}k\ 1)$ – For other steps, it picks from the conditional probabilities. The variance of these estimations usually increases with n, so it is necessary a further refinement. In order to do resampling, a method to reduce variance is required- it will be based on sampling again from the newly created approximation distributions. All of the samples are associated with numbers to estimate the distribution.

Summary of the algorithm:
For the first time (n=1)
Sample $X1i\sim q1\ (X1)$
Compute the probabilities (weights) $W1\ (X1i)$ and $W1i$Resample $\{\ 1i\}$ to obtain N particles $\{1N,\ X1i\}$
For all other times $(n \geq 2)$
$X1{:}ni \sim qn\ Xn\ X1{:}n-1i$ and set $X1{:}ni \leftarrow (X1{:}n-1i,\ Xn\ i)$
Compute weights (:) and Wni
If is necessary, resampling the $\{\ 1{:}n\ \}$ in order to obtain $\{1N,\ X1{:}ni\}$
Print screens of the traffic lanes detection using particle filters.

Cascaded Neural Networks in order to recognize traffic signs
The traffic signs detection is based on Multi-Layer Perceptron Neural Networks. The neural Networks were trained using real images of traffic signs. The neural networks were trained using Back-Propagation with Momentum algorithm. There are two neural networks for traffic signs forms (blue forms and red forms).For each traffic sign form there is a neural network associated with. The networks were learned in this software.

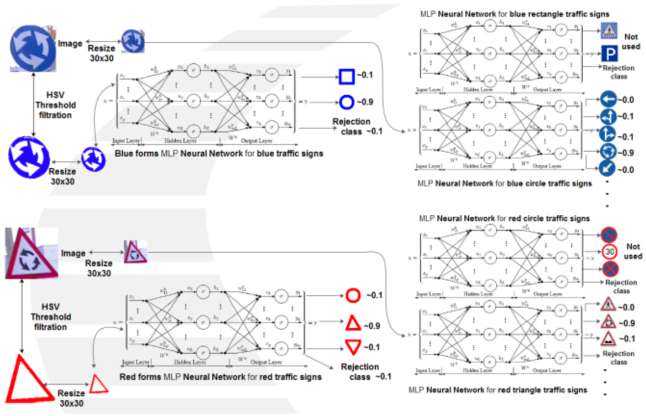


Image
Resize 30x30
HSV Threshold filtration
Resize 30x30
~0.1
~0.9
Rejection class ~0.1
Blue forms MLP Neural Network for blue traffic signs
MLP Neural Network for blue rectangle traffic signs
Not used
Rejection class
~0.0
~0.1
~0.1
~0.9
~0.0
MLP Neural Network for blue circle traffic signs
Image
Resize 30x30
HSV Threshold filtration
Resize 30x30
~0.1
~0.9
~0.1
Rejection class ~0.1
Red forms MLP Neural Network for red traffic signs
MLP Neural Network for red circle traffic signs
Not used
Rejection class
~0.0
~0.9
~0.1
Rejection class
MLP Neural Network for red triangle traffic signs

3. CONCLUSION AND FUTURE WORK

The reliable intelligent driver assistance systems and safety warning systems is still a long way to go. However, as computing power, sensing capacity, and wireless connectivity for vehicles rapidly increase, the concept of assisted driving and proactive safety warning is speeding towards reality. As technology improves, a vehicle will become just a computer with tires. Driving on roads will be just like surfing the Web: there will be traffic congestion but no injuries or fatalities. Advanced driver assistant systems and new sensing technologies can be highly beneficial, along with large body of work on automated vehicles. All the program and software is tested by Unity simulator and TORCS (The open racing car simulator) designed by NVidia corporation

REFERENCES

[1] *“DetailsGooglecar,”http://googlecarjames.weebly.com /details.html, (Visited on09/08/2016).*

[2] M.Frutiger and C.Kim, “Digital Autopilot Test Platform with Reference Design and Implementation of a 2-Axis Autopilot for Small Airplanes,” *Department of Electrical and Electronics Engineering*, pp.1–24, 2003.

[3] M.Weber, “Where to? A History of Autonomous Vehicles,” 2014.

[4] D. Helbing, “Traffic and related self-driven many-particle systems, *“Reviews of Modern Physics*, vol. 73, no. 4, pp. 1067–1141, Dec 2001.

[5] P. Rau, “Target Crash Population of Automated Vehicles,” in *In 24th International Technical Conference on the Enhanced Safety of Vehicles (ESV)*, 2015, pp.1–11.

[6] *http://googlecarjames.weebly.com/details.html*

[7] A. Fisher, “Inside Google’s Quest to Popularize Self- Driving Cars,” *Popular Science*, 2013. [Online]. *Available:http://www.popsci.com/cars/article/2013-09/google-self-driving-car*

[8] “Official google blog: Green lights for our self-driving vehicle prototypes,” *https://googleblog.blogspot.nl/2015/05/ self-driving-vehicle- prototypes-on-road.html, (Visited on02/16/2016).*

[9] K. R. Memon, S. Memon, B. Memon, A. R. Memon, and M. Z. A. S. Syed, “Real time Implementation of Path planning Algorithm with Obstacle Avoidance for Autonomous Vehicle,” in *3rd 2016International Conference on Computing for Sustainable Global Development”*, New Delhi, India,2016

[10] E. Frink, D. Flippo, and A. Sharda, “Invisible Leash: Object-Following Robot,” *Journal of Automation, Mobile Robotics & Intelligent Systems*, vol.10, no.1, pp. 3–7, Feb2016.

[11] Md Najmus Saquib, Mr. Javed Ashraf and Col. (Dr.) O.P Malik “ Self-Driving Car System Using AI (Artificial Intelligence)” *Asian Journal of Applied Science and Technology (AJAST)*, Volume 1, Issue 6, Pages 85-89, July 2017