International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249 – 8958 (Online), Volume-9 Issue-5, June 2020

Virtual Self Driving Car using the Techniques of Image Processing and Deep Neural Networks

R. Poonkuzhali, Vineet Kumar Singh

Abstract - Self-driving cars come with both confronts and openings. Many tech gigantic companies like Google, Tesla, Apple and many more are funding billions of dollars for the implementation of a driverless car. In this modern era of automation, every human need has been driven towards things to be automated. From automated traffic control to automated home, everything comes up to the rescue of human to provide a comfortable and relaxing lifestyle. After almost automating everything now mankind has moved to automate the transportation, starting with automating the vehicles. With this the first step taken is to devise a self-driving car or a driverless car, with an aim to provide human with relaxed driving. Ever since the idea immersed, every year Google redefines the model to meet the need. In this paper, an open source simulator by Udacity, known as Self driving Car Engineer has been used for collecting the dataset and executing the neural net implemented using Python in association with packages like Keras, OpenCV etc.

Keywords - Neural Networks, Feature Extraction, Image Augmentation, Behavioral Cloning

I. INTRODUCTION

Self-driving car implementation in real time is a herculean task and requires a lot of funding and many other infrastructures. The proposed work is based on the concept of Image Processing, Data Preprocessing, Deep Neural Networks, Polynomial Regression and Behavioral Cloning. There are many challenges that are being faced in the real time implementation of driverless cars. From technical to legal issues. The dream for self-driving cars is no far a dream. The day is not so far that driverless car will be on roads within five years. A recent survey of U.S. drivers states that, 1) The driver feels reasonably safe while having a ride in a driverless car. 2) Cost of self-driving car at the end comes same as a normal car with a driver and 3) Allow driverless car to be on public road. As mentioned earlier many tech giants have invested billion dollars, one such company is Google, which has been working with self-driving car since 2010 [1].

II. LITERATURE SURVEY

Krasniqi and Hajrizi reported the use of IoT Technology to drive the automotive industry from connected to full autonomous vehicles [2]. They proposed ideas in which the autonomous car can be aided with Internet of Things (IoT). The role of new communication technologies like 5G can play a vital role in the success of self-driving cars. The paper analysed IoT as the driver of the car and the requirements to meet the market demand for the same.

Revised Manuscript Received on May 15, 2020.
* Correspondence Author
R. Poonkuzhali*, School of Electroncis Engineering, VIT university, Vellore, Tamilnadu, India . Email: rpoonkuzhali@vit.ac.in
Vineet Kumar Singh, B.Tech School of Electroncis Engineering, VIT university, Vellore, Tamilnadu, India. Email: vineetkumar.singh@vitstudent.ac.in

Jorge Villagra, Leopoldo Acosta and Rosa Blanco reported an automated driving, intelligent vehicle which compares the human driving skill with complex driving skills developed by artificial neural network [3]. Human drivers are equipped with skills to take decision based on situation at the moment but the driverless car doesn't possess the sense to take that sort of situation posing legal issues. Eby, Molnar, and St. Louis, discussed different in-vehicle and self-driving technologies for crash avoidance like Lane departure warning/mitigation, curve speed warning, forward collision warning/mitigation, blind spot warning and cross-traffic alert. These are some of the challenges to be overcome [4]. This paper produces some of the solutions to the problem but are not met to the real world. A novel system is proposed in [5] where, each and every car was integrated with GPS and one wireless antenna such as WiFi antenna or DSRC antenna. The GPS localization data of vehicles is sent to nearby stations through wireless medium of communication. This model proposes 3 types of infrastructure V2V, V2I and I2V.

Gora and Rub proposed a mathematical traffic model for autonomous vehicle. The autonomous car is simulated assuming it to belong to a microscopic model, that is at each step of simulation the vehicle is considered a separate unit [6]. The security challenges which are faced by different technologies associated with driverless vehicle are addressed and pose a solution to them [7]. Based on function of technologies they divided them into four subparts: network ,vision, position , and sensing technologies and a detail study is proposed. Buluswar and Draper incorporated the image processing technique to identify the real time objects and based on identification of image trains the model to take decision [8]. Kuderer, Gulati, and Burgard presented the idea of having a camera on top of the car so as the captured image can be used to make real time decision in order to facilitate the working of a self-driving car. The simulation part is done using Udacity's Self-Driving car engineer, an open source simulator [9].

III. BLOCK DIAGRAM

The model devised is based on Deep Learning and Fig.1 shows the flow diagram. Basically it consists a series of nodes, arranged in layers (known as hidden layers), take input from the input node and adjust the weight and biases in order to match the known output and then using the same weight and biases, it predicts for the newly fed input values. Hence, to train such model at first dataset is collected. After collecting the dataset, task of data preprocessing is achieved so as to make the dataset unbiased towards a particular value and suitable for easier computation by the neural network.

Next the neural net is formed using the model proposed by Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao and Karol Zieba [10]. The coded neural net model is then linked with the simulator for testing.

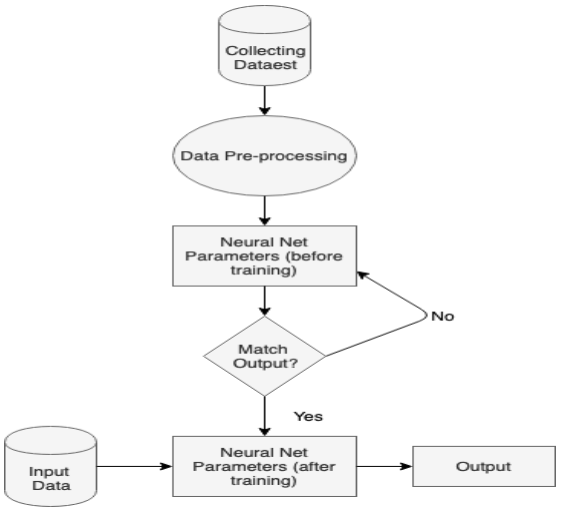

Fig.1 Flow diagram of the model

IV. TRAINING DATA COLLECTION

The simulator operates in two mode, one mode is the training mode and the other mode is the autonomous mode. Along with the two modes of operation it has two different region of operation one being for normal city rode and one for hilly regions. The training mode is used to collect the dataset used for training the model and the autonomous mode to verify the model being devised.

The simulator has a virtual car, which has three of its cameras attached on left, right and center, making it capable for capturing images from three of the view. Along with capturing the images it also keeps records of steering angle, throttle value, reverse data and speed in a csv form. Fig.2 shows the format of dataset collected. Fig.3 shows the image for an instance of captured value.

	center	left	right	steering	throttle	reverse	sp
0	center_2018_07_16_17_11_43_382.jpg	left_2018_07_16_17_11_43_382.jpg	right_2018_07_16_17_11_43_382.jpg	0.0	0.0	0.0	0.64
1	center_2018_07_16_17_11_43_670.jpg	left_2018_07_16_17_11_43_670.jpg	right_2018_07_16_17_11_43_670.jpg	0.0	0.0	0.0	0.62
2	center_2018_07_16_17_11_43_724.jpg	left_2018_07_16_17_11_43_724.jpg	right_2018_07_16_17_11_43_724.jpg	0.0	0.0	0.0	0.62

Fig.2 Dataset of the center, left, right image name with steering angle, throttle value, reverse data and speed.

Fig.3 Left, Center and Right image for particular instance

The collection of datasets is highly dependent on how well the virtual car is driven on the car. Many a times situation like, driving at a steering angle of 0^o can occur and recorded dataset has more of its value aligned towards 0^o. This might distract the model by exposing it to a biased angle of 0^o. Moreover, the image collected via this method is limited, hence this limitation might lead to a less accurate model. To overcome these problems, data preprocessing is necessary.

V. DATA PREPROCESSING

Data preprocessing is one of the most important step, that should be done before feeding the data to model either for training or testing. This step makes the input data suitable for processing and ensures that model is not biased towards a particular value. Fig.4 shows histogram plot of steering angles (in radians).

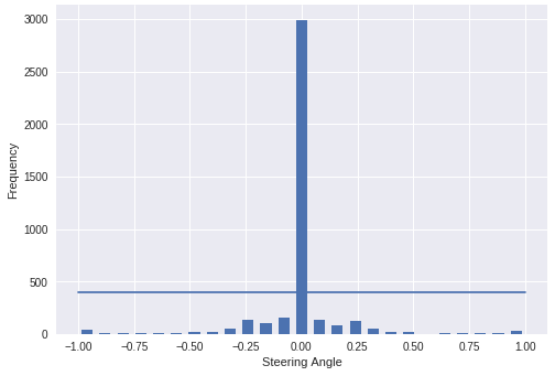

Fig.4 Histogram plot of steering angle in radian

As it can be seen from the plot that frequency of 0 radian is much high compared to that of the other angles and this may cause the model to be biased more towards 0 radian rather than other angles. This happens because the model is exposed to 0 radian many more times, forcing the model parameter to 0 radian. Hence, to handle this the frequency above 400 is cut and the final distribution is shown in Fig.5.

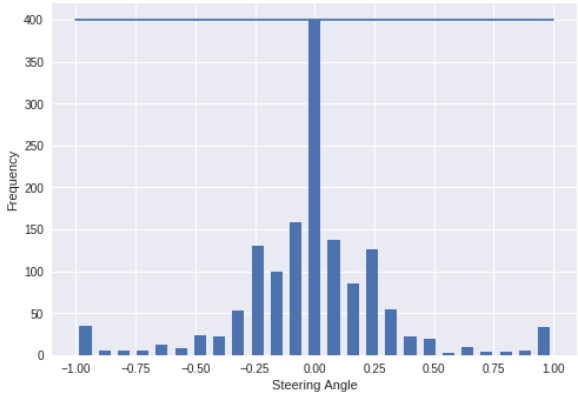

Fig.5 Histogram plot of steering angle in radian after preprocessing

As it can be seen after preprocessing, the model will be equally exposed to all the values of angles. After removing certain part of data for unbiasing the model, we may lose certain important information.

This also implies that we have deleted some of the images from our dataset. So, to overcome this, a method of Image Augmentation is used which is proposed by Li , Hatley and Kim [11] . This method takes one image and performing some transformations to it, in order to achieve more images. Augmentation on images reproduces more image dataset from limited image dataset [12]. More dataset means more exposure of model to different case leading to higher accuracy reported by . Scaramuzza, Fraundorfer, Siegwart and Pollefeys[13]-[14].

Some of the image augmentation performed on the images like zooming, panning, altering brightness, flipping and they are shown in Fig.6 - Fig.9

A. Zooming Image

A given image is taken and zoomed to create a new image. Fig.6 shows a comparison between original image and zoomed image.

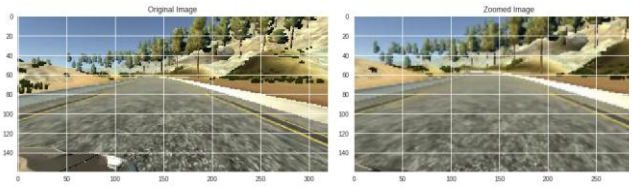

Fig.6 Left half of image is the original image and zoomed image on the right side

B. Panning an Image

This method creates a sense of speed around moving object. Creating an image with the object of interest to be focused while the background to be blurred. Fig.7 shows a comparison between original image and panned image.

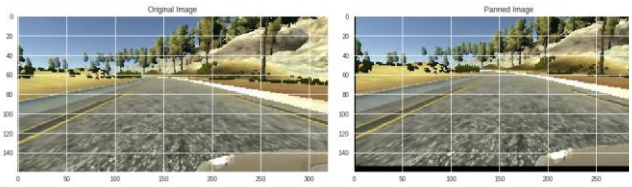

Fig.7 Left half of image is the original image and panned image on the right side

C. Brightness Altering

In this method the original's image brightness is increased or decreased to create a new image. Fig.8 shows a comparison between original image and brightness altered image.

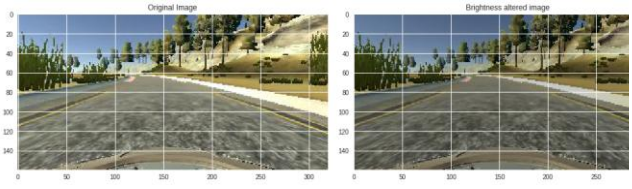

Fig.8 Left half of image is the original image and brightness altered image on the right side

D. Flipping the image

Original image is flipped by 180^0 along the vertical axis. In this case flipping changes the angle from positive to negative radian. Fig.9 shows a comparison between original image and flipped image.

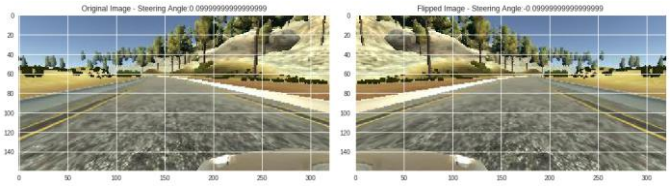

Fig.9 Left half of image is the original image and brightness altered image on the right side

The above-mentioned techniques are some of the methods to overcome the scarce of data points problem. Now to make these datasets suitable for being processed by the neural net model proposed by Nvidia we need to convert the RGB (Red, Green and Blue) to YUV (Y is luminescence, U&V are chrominance) [15].

After preprocessing the data from the above steps, we are there with ample of datasets to go. Many a time it's redundant to use the entire image, as it adds to the computational overhead. Hence, we take a certain portion of the image which is called the Region of Interest. As later we are going to use the Nvidia Neural net framework, so to make dataset adaptable to it we convert the image from RGB scale to YUV scale. After these two steps (choosing region of interest and converting from RGB to YUV), the image appears as shown in Fig.10.

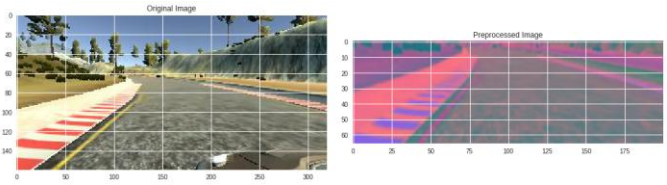

Fig.10 Left half of image is the original image and two step preprocessed image on the right side

The final stage of preprocessing is dividing the dataset into 3 sets, known as Training Dataset, Validation Dataset and Testing Dataset. Training dataset is the dataset used by the neural net to find the weight and bias, with the final aim to minimize the mean square error [13], Validation dataset is used to give an estimate of model skill while tuning model's parameter. The third one, testing dataset is used to test the accuracy of model designed. Fig.11 shows the histogram distribution of the Training and Validation dataset.

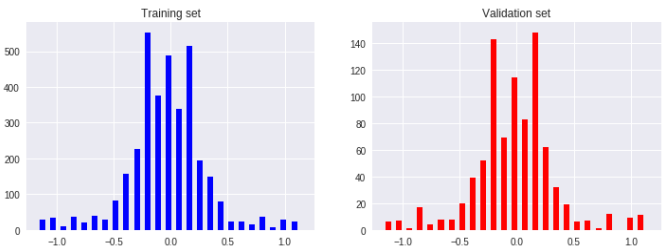

Fig.11 Left half of image is the distribution of training dataset and distribution of the validation dataset on the right side

It can be seen from the image that almost both the Training and Validation set has equal distribution, hence, showing the data is preprocessed properly without being biased.

VI. NEURAL NETWORK ARCHITECTURE

In general, neural network is a set of nodes forming interconnected layer, with each node have an activation function like ReLu or Exponential. The in between layers are called hidden layers. There are input and output nodes too, which take input data and pass through the hidden layers to produce the output. These hidden layers have parameters associated with it called weight and bias. Fig. 12 is the structure of general neural network.

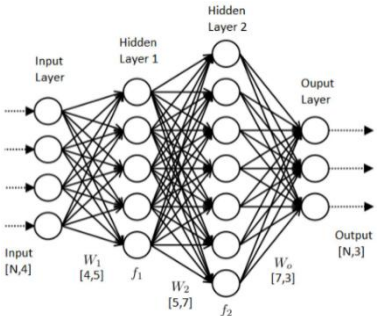

Fig.12 General Overview of a Neural Network

Fig. 12 has N data points with 4 input nodes, 2 hidden layers with, first hidden layer having 5 nodes and second having 7 nodes and finally connected to 3 output nodes. This is a fully connected neural network. The input layer is a matrix(X) of [N,4] and the dimensions of first hidden layer is reported by Cox and Shea [16]. Let this layer's matrix be considered as W_1 and the same way second hidden layer weighted matrix be W_2, then output matrix(Y) of [N,3] is reported by Nister [17].

$$Y = X * W_1 * W_2 + B \ldots \quad (1)$$

Where B is [N,3] bias matrix. This Y is compared with Y' (actual output) and mean square error is calculated as:

$$MSE = \sqrt{\sum (Y - Y')^2} \ldots \quad (2)$$

The goal is to minimize the mean square error in order to achieve a model with higher accuracy. After each time the model is trained two losses are obtained and each time the loss value is decreased significantly. Hence making the model better each time. Fig. 13 shows a glimpse of a neuron of a neural network.

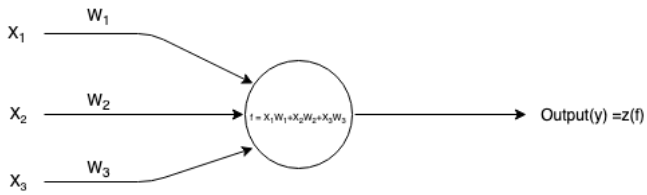

Fig. 13. Neuron Level Processing of data

Here X_1, X_2 and X_3 are input variables and W_1,W_2 and W_3 are the weight of the branches and Z is any activation function like ReLu, Exponential etc.

Now, the proposed neural network Nvidia architecture is shown in Fig. 14. The main aim remains the same, which is to minimize the mean square error between the expected output and the resultant output of the neural net architecture.

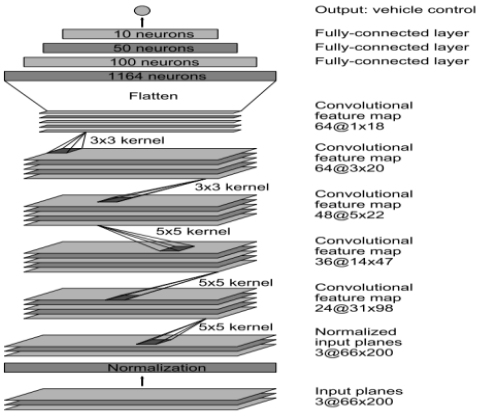

Fig.14 Different layers of Nvidia Proposed Neural Network Architecture for self-driving cars

The neural network consists of in total 9 layers, having a normalization layer, 5 convolutional layers and 3 fully connected layers. The input images are split into YUV planes and passed to the neural network. The first layer of the neural network performs image normalization. The normalizer is hard-coded, and it can't be adjusted in the learning process. Performing normalization in the network allows the normalization scheme to be altered with the network architecture and this is aided by GPU processing. The convolutional layers are structured to perform feature extraction and are chosen empirically through a series of experiments that varied layer configurations. Stride convolutions are used in the first three convolutional layers with a 2×2 stride and a 5×5 kernel and a non-stride convolution with a 3×3 kernel size in the last two convolutional layers.

Let's look at each layer in details:

- Input Layer – The input layer consists of three planes. With each plane denoting Y-Luminance, U, V – Chrominance. This is done to ensure the model gets the input image in accordance to its requirement, for efficient processing of the image.
- Normalized Plane – This plane performs the normalization of the three input image planes. Normalization in general terms to make the values get centered around (-1,1), following the Gaussian Distribution. This is done to ensure less time taken for the processing.
- Convolutional Plane- This is a concept that combines the concept of Neural Network and Convolution. In this method first the input plane of the image is convoluted with a mask and the resultant image is given as input to a set of layers, forming the neural network. Five such layers are used for rigorous training of the model.

The five convolutional layers are followed with three fully connected layers leading to an output control value which is the steering angle of the car. The fully connected layers are designed to function as a controller for steering angle.

VII. RESULTS AND DISCUSSIONS

After the neural network has been trained, it's time to test it on the simulator. To test it on the simulator, the code was integrated with the simulator using flask and socket. After the successful integration of the code and simulator, the model is tested in the Autonomous mode. Fig.15 shows the successful simulation of the Driverless car. On the top left corner of the image steering angle can be seen clearly and, on the bottom, right we have the speed of the car. The graphic simulation incorporates almost all the real time challenges that a driver faces while driving on a road with no traffic, like turns, keeping the car in middle of the road etc.

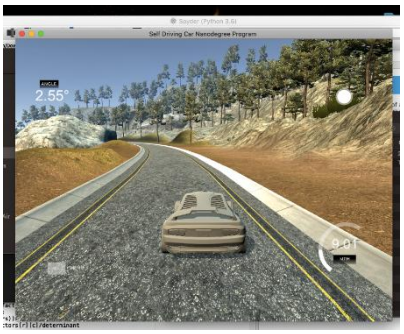

Fig. 15 Successful simulation of the Self Driving car

This might seem obvious to a human as it is blessed with a brain to take decisions but might be complex for machines who are built to compute but not to take decisions [18-19].

Fig.16 shows the summary of the architecture of Nvidia. neural network used for this simulation

Layer (type)	Output Shape	Param #
conv2d_6 (Conv2D)	(None, 31, 98, 24)	1824
conv2d_7 (Conv2D)	(None, 14, 47, 36)	21636
conv2d_8 (Conv2D)	(None, 5, 22, 48)	43248
conv2d_9 (Conv2D)	(None, 3, 20, 64)	27712
conv2d_10 (Conv2D)	(None, 1, 18, 64)	36928
flatten_2 (Flatten)	(None, 1152)	0
dense_5 (Dense)	(None, 100)	115300
dense_6 (Dense)	(None, 50)	5050
dense_7 (Dense)	(None, 10)	510
dense_8 (Dense)	(None, 1)	11

Fig. 16 Summary of the neural network architecture

From the above summary it can be summed that, total number of parameters =252,219, total number of trainable parameters = 252,219, total number of non-trainable parameters = 0. Fig. 17 shows the comparison between the loss of validation dataset and training data set after each epoch, the less the error the more accurate the model is. The model was trained with 10 epochs with 300 steps per epoch.

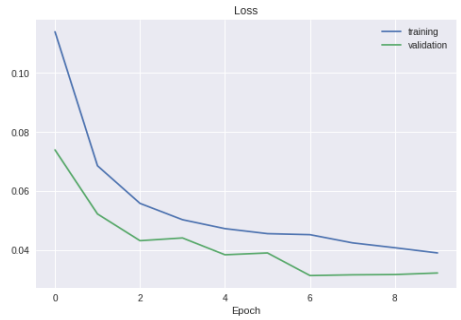

Fig.17 Loss of training and validation after each epoch

VIII. CONCLUSION

The self-driving car is successfully implemented and the objective for the same is met. The model was achieved with 10 epochs, with each epoch having 300 steps. Adam optimizer with coefficient of $1e^{-4}$ was used. And the activation function at each node of the layer was "elu"or exponential linear unit defined as:

$$R(z) = \begin{cases} z & , z > 0 \\ \alpha(e^z - 1), & z \leq 0 \end{cases} \quad (.3)$$

The proposed model in this project poses both challenges and opportunities. Legal laws and safety measures need to be framed for proper functional implementation of such complex model in real time. Many countries have tested these models but have not implemented in public. But for sure the next decade will witness the driverless car.

REFERENCES

1. Eric R. Teoh and David G. Kidd, "Rage against the machine? Google's self-driving cars versus human driver", Journal of Safety Research, Vol. 63, 2017, pp. 57-60.
2. X. Krasniqi and Hajrizi E., Use of IoT Technology to Drive the Automotive Industry from Connected to Full Autonomous Vehicles, IFAC, 2016. pp. 269-274.
3. Jorge Villagra, Leopoldo Acosta and Rosa Blanco, Automated Driving, Intelligent Vehicle, 2018, pp.275-342.
4. D. W Eby, Molnar, L. J., and St. Louis, R. M , "In-vehicle and self-driving technologies. Perspectives and Strategies for Promoting Safe Transportation Among Older Adults, 2019, pp. 197–223.
5. Wang, X., Wei, T., Kong, L., He, L., Wu, F., & Chen, G. "ECASS: Edge computing based auxiliary sensing system for self-driving vehicles", Journal of Systems Architecture, 2019, pp.
6. Gora, P., and Rüb, I., "Traffic Models for Self-driving Connected Cars". Transportation Research Procedia, vol. 14, 2016, pp. 2207–2216.
7. De La Torre, G., Rad, P., and Choo, K.K. R, "Driverless vehicle security: Challenges and future research opportunities". Future Generation Computer Systems. 2018
8. Buluswar, S. D., and Draper, B. A., "Color machine vision for autonomous vehicles", Engineering Applications of Artificial Intelligence, vol. 11, no. 2, 2004, pp.245–256
9. Kuderer, M., Gulati, S., and Burgard, W., Learning driving styles for autonomous vehicles from demonstration. 2015 IEEE International Conference on Robotics and Automation (ICRA).
10. Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to End Learning for Self-Driving Cars. 2016.
11. H. Li, R. Hartley and J. Kim, A linear approach to motion estimation using generalized camera models. In Computer Vision and Pattern Recognition, June 2008, pp. 1–8.

12. D. Scaramuzza, F. Fraundorfer, and R. Siegwart, May 2009 Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. In International Conference on Robotics and Automation, pp. 4293 –4299.
13. D. Scaramuzza, F. Fraundorfer, R. Siegwart, and M. Pollefeys, September 2009, Absolute scale in structure from motion from a single vehicle mounted camera by exploiting nonholonomic constraints. In International Conference on Computer Vision, pp. 1–7.
14. R. Siegwart, I. Nourbakhsh, and D. Scaramuzza,et. 2011 Introduction to Autonomous Mobile Robots. MIT Press, 2nd edition.
15. C. Mei, P. Rives. Calibrage non biaisé d'un capteur central catadiotrique. In *RFIA*, Tours, France, January 2006.
16. D. A. Cox and D. O'Shea, 2007, Ideals, varieties, and algorithms - An introduction to computational algebraic geometry (2nd. ed.). Springer.
17. D. Nister, An efficient solution to the five-point relative pose problem. In Pattern Analysis and Machine Intelligence, vol. 26, June 2004, pp. 756–777.
18. T. Kazik, L. Kneip, J. Nikolic, M. Pollefeys, and R. Siegwart, Real-time 6d stereo visual odometry with non-overlapping fields of view. In Computer Vision and Pattern Recognition, June 2012, pp. 1529–1536.
19. M. A. Fischler and R. Bolles, Random consensus: a paradigm for model with applications to image analysis. In Communications of the ACM, vol. 24, 2018, pp. 381–390.

AUTHORS PROFILE

R.Poonkuzhali, working as Associate Professor in School of Electronics Engineering, VIT University, Vellore, Tamilnadu, India. Her research area includes wearable antennas, antennas for military applications. Several papers have been published in MIMO antennas, fractal antennas in VHF/UHF range.

Vineet Kumar Singh is studying B,Tech, with Specialization in Internet of Things and Sensors. His research area includes Image Processing and machine learning techniques.