

Design of A Data Transmission Unit for An Autonomous Vehicle

Prof. Sunayana Jadhav

Department of Electronics and Telecommunications Engineering, Vidyavardhini's college of Engineering & Technology, Vasai Road(W).
Palghar – 401202, India

Alok Dubey

Department of Electronics and Telecommunications Engineering, Vidyavardhini's college of Engineering & Technology, Vasai Road(W).
Palghar – 401202, India

Amit Bhagat

Department of Electronics and Telecommunications Engineering, Vidyavardhini's college of Engineering & Technology, Vasai Road(W).
Palghar – 401202, India

Manoj Thakur

Department of Electronics and Telecommunications Engineering, Vidyavardhini's college of Engineering & Technology, Vasai Road(W).
Palghar – 401202, India

Vibha Parab

Department of Electronics and Telecommunications Engineering,
Vidyavardhini's college of Engineering & Technology,
Vasai Road(W). Palghar – 401202, India

Abstract- The vehicle equipped with autonomous driving capability detects the environment, locates its position, and operates appropriately to go to the specified destination safely without human input. For developing such autonomous vehicles many complex algorithms are used which involve various computer science fields such as Machine Learning, Artificial Intelligence, Neural Networks and Image Processing. The autonomous vehicle needs to provide accurate output in real time environment which in turn increases the computational power of the system. This paper proposes to build an end-to-end autonomous driving system based on most recently published system designs from academic research and industry practitioners. The proposed Data Transmission Unit comprises of a central wide angle camera mounted on the dashboard of a vehicle. The live video stream from the camera is fed to the Convolutional neural network (CNN) as input which in turn predicts the steering angle for every frame. The predicted steering angle is calibrated based on the designed steering and brake mechanism of the vehicle.

Keywords— Autonomous vehicle, Deep learning, Convolutional neural network, Steering mechanism.

I. INTRODUCTION

Autonomous Driving systems have attracted significant interest from many industry experts as well as in the field of research. Many automotive companies such as Google, Tesla, and Mobileye are investing heavily in research and development of autonomous driving systems. The amount of accidents occurring daily are in millions, which is due to reckless and careless driving. Use of efficient Self Driving cars will help in reducing the number of accidents occurring and improve the quality of road life. The concept of autonomous driving is also used in transportation with door step food delivery which is under development by many organizations such as Uber, Ola, etc. Constructing or building an autonomous vehicle is a tough job because the autonomous

vehicles need to make critical decision in real life environment and if the results are not accurate the outcomes can be critical or sometimes even dangerous for human life.

II. RELATED WORK

The authors of [1] describe a CNN that is very different than the traditional pattern recognition. The CNN model is trained in such a way that it learns to compute steering angle by itself in different atmospheric conditions. The model is trained for hours from the dataset collected driving a vehicle in different atmospheric conditions. The vehicle used for collecting the dataset had three cameras mounted on the car and for every frame it is recording the steering angle. This dataset is then trained on the CNN model that consists of five convolutional layers, three fully connected layers and an output layer which is the predicted steering angle. Based on the simulation tests conducted in real environment, the trained model achieved an autonomy of 90%. This model is able to drive on roads having no lane markings or a proper path defined.

Shih-Chieh Lin, et al, discuss in [2] about the architectural implications faced in designing an autonomous vehicle. This paper discusses about the design constraints faced such as performance, processing and computational power, latency, decision making abilities and hardware design constraints. It is identified that for autonomous vehicle to react to a constantly changing traffic it is very critical for the system to make decisions with a processing latency less than 100ms. Object detection, tracking and localization are the three process that amount to 94% of the computational power. The proposed system consists of three major components namely scene recognition which is used to track nearby objects, path planning to define the path of the vehicle from source to destination. It is found out that the tail latency should be used to evaluate the performance of an autonomous vehicle rather than the mean latency. Based on the results obtained the

authors conclude that specialized hardware like FPGA and ASIC have more efficiency and also latency is less than 1.8ms as compared to CPU's and GPU's.

The ALVINN system [6] is a 3-layer back-propagation system built by a group at CMU to complete the undertaking of lane following. It trains on images from a camera and a distance measure from a laser range discoverer to yield the direction in which the vehicle should move.

The MIT University[3] has collected and analyzed the naturalistic dataset of semi-autonomous vehicle to characterize the human-machine interaction across a range of different ecosystem. The datasets are used to extract the feature of the environment. Deep learning aims to address detection, labeling, generation and planning of the surrounding environment of an autonomous vehicle. They have a used face-camera for driver face gesture and position monitoring. Instrument cluster camera for vehicle state. 122 drivers have driven in 23 different cars up to 5.1 million miles. 7.1 billion video-frames are processed in neural computations.

Though there are ample of theoretical information available regarding autonomous vehicle but lack of practical information. Taking into account this limitation, we aim to build a deep learning model and design the hardware mechanisms of an autonomous vehicle.

III. PROPOSED METHODOLOGY

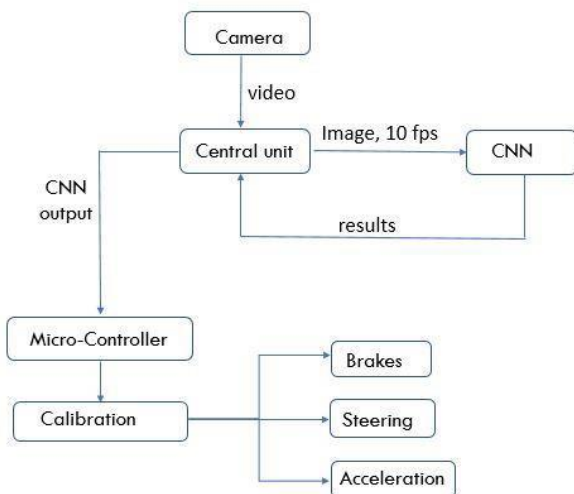


Fig. 1. System Architecture

The proposed system comprises of a central wide angle camera mounted on the dashboard of a vehicle. The camera captures video and will be feeding it to a central unit as frame. Raspberry Pi 3b+ is used as a central unit in our system. The live video stream from the camera will be converted to images at a rate of 10 frames per second. These images will then be pre-processed and fed as input to the CNN model. The CNN model will predict steering angle for every frame processed. The output of the CNN model will be calibrated based on the steering mechanism designed. The system has mainly three parts namely Software, Interfacing and Hardware designing. The software design unit comprises of data augmentation, data pre-processing and designing a CNN model. Hardware

design comprises of steering mechanism design and defines the parameters of an autonomous car. The interfacing part consists of installing the central unit, defining communication protocols and sending commands to hardware components.

A. DATA AUGMENTATION

The dataset consists of around 45,000 frames along with the steering angle value for each frame. The dataset acquired is static and not enough to successfully train the model and accurately predict the steering angle. There is a need to collect data for different environmental conditions or augment the current dataset to emulate different driving conditions using image processing techniques. The various data augmentation methods of the dataset are described below.

1. Image Blur

The blurring effect also known as smoothing an image, removes outlier pixels from the image or noise present in the image. In this process low-pass filter is used to remove noise from an image while leaving the majority of the image same as before.

2. Image Zoom

Zooming refers to expansion of an image and this is achieved by adding pixels to an image. The major task is interpolation of the new pixels from the surrounding original image. Zooming can be done by either pixel replication or bilinear interpolation and doesn't affect picture qualities or pixels.

3. Image Brightness

To adjust the brightness of an image the pixel values of the image are changed. Brightness of the image is reduced by subtracting a constant value from the pixel and increased by adding a constant value to the pixel.

B. DATA PRE-PROCESSING

After successfully augmenting the dataset, the next step is to pre-process the dataset in order to reduce unwanted features, this procedure is also called as a dataset normalization. As per the Deep learning model [1], it requires an image of 66x200 resolution and the image should be in YUV color space. The reason of preferring YUV color space over RGB color space is due to its perceptual similarities to human vision [5]. Every batch of training as well as validation data undergoes image preprocessing. The original and pre-processed image can be observed in Fig. 2 and Fig. 3 respectively.



Fig. 2. Original image



Fig. 3. Pre-processed image

C. Convolutional Neural Network(CNN) CNN is the brain of an autonomous vehicle. The Deep learning model is used to successfully predict steering angle, acceleration and deceleration command. It needs to be trained on vast and diverse dataset. A human eye captures every frame and then the brain sends commands to whether it should turn right or left and also to accelerate or decelerate. A human brain needs hours of practice to accurately make decisions. A Deep learning model is a virtual emulation of a human brain. It requires a dataset to be trained in order to predict the output, which in our case is the steering angle.

Deep learning is a subset of Machine learning. Machine learning requires a set of structured data with labels and features defined properly, while Deep learning does not require structured data. It consists of many layers which are used to extract features of the dataset making the model more robust to any changes in the dataset. CNN is a type of Deep Learning Neural network is especially used for extracting features from images and videos [4]. The proposed model [1] consists of five convolutional layers, three fully connected layers, one output layer and a flatten layer. The output of the model is the predicted steering angle. All the three layers are explained in detail below.

1. Convolutional layer

An image consists of many features such as edges, contrast and brightness. Convolutional layer is used to extract different features from the images by applying convolutional function of different kernel size, image strides and an activation function. Stride is the amount of pixels the kernel would skip during convolutional operation. As shown in Fig. 4, a convolutional layer is defined by number of kernels, size of kernel and strides.

Kernels of different sizes are used to extract different features. All these convolved images are stacked upon each other in such a way that each section of an image consists of different features which helps in reducing computational complexities.

2. Flatten layer

The flatten layer converts the output of the fully connected layer which is a 3-dimensional matrix to a 1-dimensional array in order to feed it to the fully connected layer. Neurons are generated from this fully connected layer. As shown in Fig. 4, after flattening the output of convolutional layer, the features of input image is represented in 1164 neurons.

3. Fully connected layer

It is a simple feed forward neural network. It forms the last layer of the CNN. To create a fully connected layer we need to define number of neurons present in the layer as well as the

activation function. We have used ELU (exponential linear unit) as an activation function in our model.

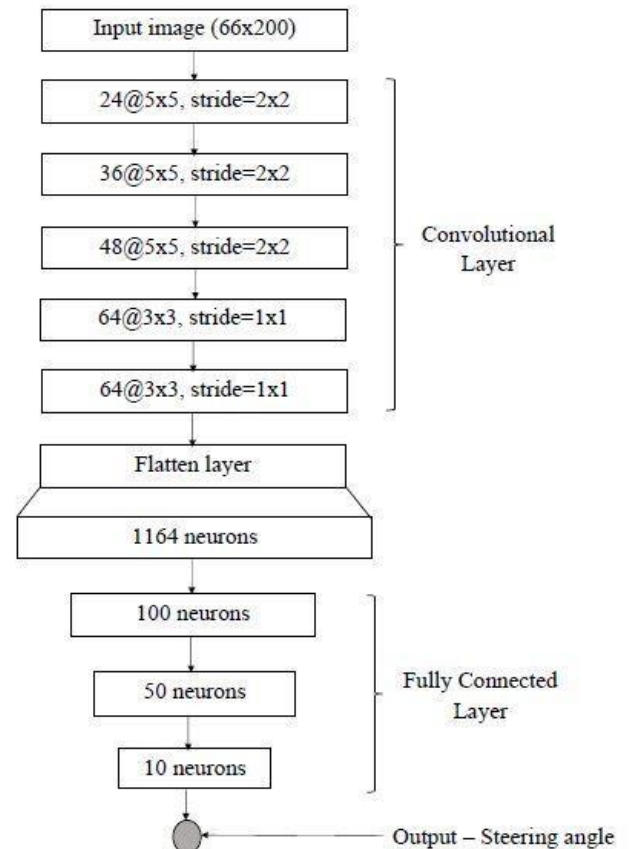


Fig. 4. CNN model architecture

D. INTERFACING UNIT

A central wide angle camera is mounted on the dashboard of the vehicle which continuously records the live video stream on the Raspberry Pi 3b+. This video stream must be converted to images at a rate of 10 fps in order to provide it as an input to the CNN and OpenCV[9] is used for this process. UART protocol is used to capture video from the camera. For communication with the central unit, Secure Shell(SSH) protocol is used in this system. Our CNN model outputs a steering angle command for each frame which is given to the micro-controller. As shown in Fig. 5, micro-controller gives steering angle and acceleration or deceleration command to DC motor driver and stepper motor driver respectively.

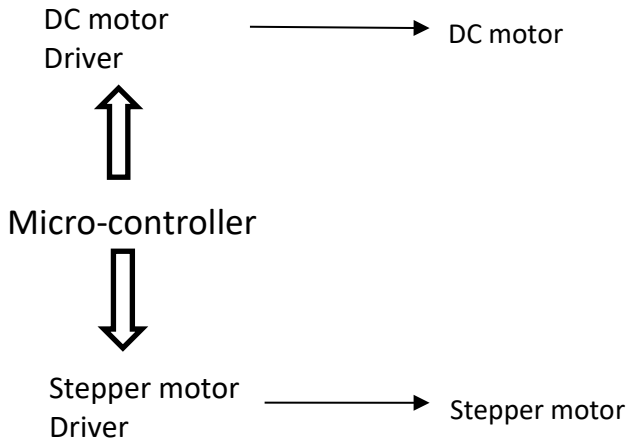


Fig. 5. Interfacing architecture

E. HARDWARE DESIGN

Steering angle design, acceleration and brake control are the most essential components in designing an electric vehicle. DC motor is used to drive the vehicle and gear mechanism is used to increase the torque. To control the steering, stepper motor is used and steering angle is designed by using gear and pinion. Gear is designed according to the required torque and a pulse is provided to control steering angle. The gear mechanism can be seen in Fig. 6.

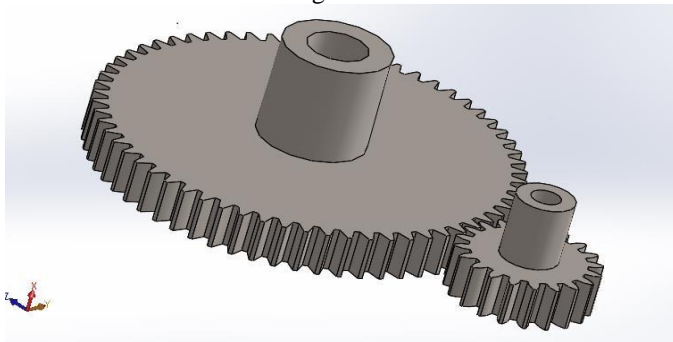


Fig. 6. Gear mechanism

Power analysis is done to find the required power to control the speed of the vehicle. Further, DC motor is controlled by L298D driver to drive the motor. Speed is controlled by sending PWM signal to the driver. Braking is done by switching off the power supply and applying reverse jerk. The design specifications and calculations are explained in the next section.

IV. RESULTS AND DISCUSSION

The following parameters are assumed for calculations:

Parameters	Value
Mass of vehicle (M)	8 kg
Speed of vehicle	5 km/hr
Slope factor	0.1
Wheel diameter	25 cm
Acceleration due to gravity	9.8 m/sec ²
Wheel radius	12.5 cm
Air drag coefficient (Cd)	0.338
Air density	1.145 kg/m ³
Cross sectional area	2.5 m ²

Angular velocity:-

Linear Speed = 1.38 m/s
 Angular Speed = 5.5 rad/s
 Frequency = 53 rpm

Peak Torque and power:-

Peak Torque = (Mass of the vehicle) x (acceleration due to gravity) x (Wheel radius) x (slope factor)
 = 1.96 N-m
 Power = (Peak torque) x (angular speed)
 = 10.78 Watt

Continuous Power and Torque:-

Air resistance = (0.5) x (Cd) x (Air density) x (Cross sectional area) x (Linear speed)²
 = 1.01 Watt
 Rolling resistance = 0.092 x (Mass of vehicle) x (linear speed)
 = 3.68 Watt
 Continuous Power = (Air resistance) + (Rolling resistance)
 = 3.71 Watt
 Continuous Torque = (Continuous Power) x (Wheel radius)
 = 0.668 N-m

Calculation for steering mechanism:-

Gear1 teeth = 55
 Gear2 teeth = 10
 Torque of stepper motor (T2) = 7.2 N-cm
 Gear1 teeth / T1 = Gear2 teeth / T2
 T1 = 39.6 N-cm
 where, T1 - Torque required to move steering

The acquired dataset is divided in two batches namely, training and validation batch. Training batch consists of 80% of the dataset while the validation batch consists of 20% of the dataset. The training batch is used to train the defined CNN model for 30 epochs with 300 steps per epoch and a batch size of 100 images from the dataset.

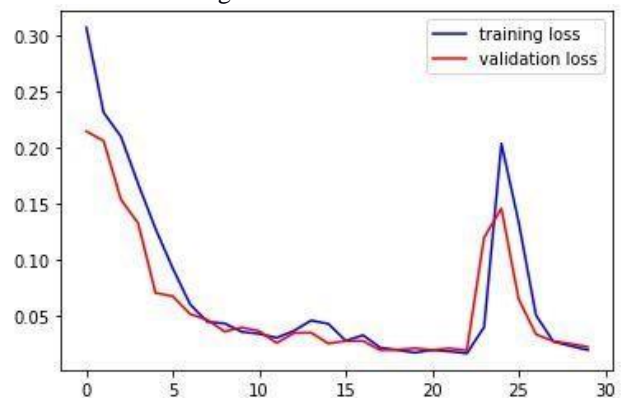


Fig. 7. Training and Validation loss

Fig. 7 shows the training and validation batch loss with number of epochs on x-axis and loss value on y-axis is defined. It is observed that both training and validation loss approximate to zero after training the model for 30 epochs. A

high level deep learning library, keras[7], is used to train the model which uses tensorflow[8] as it's backend deep learning library. For loss visualization pickle package is used. Mean squared error is used in calculation of the loss component and to define the accuracy of the CNN model R^2 accuracy is used.

$$mse = \sum_i^n (y - y^{\wedge})^2 \quad IV.I$$

$$R^2 = 1 - \frac{(y - y^{\wedge})^2}{(y - y')^2} \quad IV.II$$

where, y – actual
output y^{\wedge} –
predicted output
 y' – mean of all actual output

The mean squared error obtained after training the model is approximately 0.132 and the R^2 accuracy is greater than 90%.

V. CONCLUSION

The Deep learning model successfully predicts accurate steering angle for a given input video stream. It is observed that there is no need to specifically extract features from the input and provide it to the CNN model. The model itself identifies vital features from the input and adjusts the weights of the neurons to predict accurate output. The designed steering mechanism accurately responds to the change in steering angle output of the CNN model which is possible due to accurate hardware design and calibration. Thus the Data transmission unit of the autonomous vehicle provides accurate output in real time environment which in turn increases the computational power of the system.

VI. ACKNOWLEDGEMENT

We would like to thank our Principal, Dr. Harish Vankudre for providing the necessary facilities required for completion of our project. We take the opportunity to thank the University of Mumbai for giving us the chance to do this project. We take this opportunity to thank the Head of the Department, Dr. Vikas Gupta and staff members of Department of EXTC for their moral support and guidance. We would also like to express our sincere gratitude towards our Project Guide Prof. Sunayana Jadhav whose guidance and care made the project successful.

VII. REFERENCES

- [1] Mariusz Bojarski, Davide Del Testa, et al. "End to End Learning for Self-Driving Cars". arXiv:1604.07316v1 [cs.CV] 25 Apr 2016.
- [2] Shih-Chieh Lin, Yunqi Zhang, et al. "The Architectural Implications of Autonomous Driving: Constraints and Acceleration". ACM ISBN 978-1-4503-4911-6/18/03.
- [3] Lex Fridman, Daniel E. Brown, et al. "MIT Advanced Vehicle Technology Study: Large-Scale Naturalistic Driving Study of Driver Behavior and Interaction with Automation". arXiv:1711.06976v4 [cs.CY] 14 Aug 2019.
- [4] Yann LeCun, Leon Bottou, et al. "Gradient-based learning applied to regional document". Nov-1998.
- [5] Michal Podpora, Grzegorz Pawel Korbas, et al. "YUV vs RGB – Choosing a Color Space for Human-Machine Interaction". Position papers of the 2014 Federated Conference on Computer Science and Information Systems pp. 29–34, DOI: 10.15439/2014F206, ACSIS vol. 3.
- [6] Chi, Lu, and Yadong Mu. "Deep steering: Learning Endto-End Driving Model from Spatial and Temporal Visual Cues" arXiv preprint arXiv:1708.03798 (2017). M. Young, The technical writer's notebook. Mill Valley, CA: University Science, 1989.
- [7] <https://keras.io/>
- [8] <https://www.tensorflow.org/>
- [9] <https://opencv.org/>