Cloud Based End to End Self Driving Car Prototype

by

Shiladitya Biswas, BE/10081/2014

Srivatsa Sinha, BE/10327/2014

Gajal Vasita, BE/10308/2014

Submitted in partial fulfillment of

Requirement for the award in the degree of

Bachelor of Engineering

in

Electronics and Communication Engineering

Supervised by

Dr. Sanjaya Shankar Tripathy

Assistant Professor

Department of Electronics and Communication Engineering

Mr. Subrat Kumar Swain

Assistant Professor

Department of Electrical and Electronics Engineering

Department of Electronics and Communication Engineering

Birla Institute of Technology, Mesra, Ranchi – 835215

May 2018

<u>DECLARATION CERTIFICATE</u>

This is to certify that the work presented in this project entitled "**Cloud Based End to End Self-Driving Car Prototype**", in partial fulfillment of the requirement for the award of Degree of **Bachelor of Engineering in Electronics & Communication Engineering**, submitted to the Department of Electronics & Communication Engineering of Birla Institute of Technology, Mesra, Ranchi, Jharkhand is a bona fide work carried out by **Shiladitya Biswas, Srivatsa Sinha** and **Gajal Vasita** under our supervision and guidance.

To the best of my knowledge, the content of this project, either partially or fully, has not been submitted to any other institution for the award of any other degree.

Date:

Dr. Sanjaya Shankar Tripathy
Department of ECE
Birla Institute of Technology
Mesra, Ranchi.

Mr. Subrat Kumar Swain
Department of EEE
Birla Institute of Technology
Mesra, Ranchi.

Department of Electronics and Communication Engineering

Birla Institute of Technology, Mesra, Ranchi – 835215

CERTIFICATE OF APPROVAL

This is to certify that the project entitled "**Self-Driven Car using Convolutional Neural Network (CNN)**" is hereby approved as a suitable design of an engineering subject, carried out and presented in satisfactory manner to warrant its acceptance as prerequisite to the degree for which it has been submitted.

It is understood that by this approval, the undersigned do not necessarily endorse any conclusion drawn or opinion expressed therein, but approve the project for which it is submitted.

Internal Examiner

Date:

External Examiner

Date:

Dr. Srikanta Pal
Head of Department
Department of Electronics and Communication Engineering
Birla Institute of Technology
Mesra, Ranchi

ACKNOWLEDGEMENT

We owe our deepest gratitude to our advisor **Dr. Sanjaya Shankar Tripathy** and co-advisor **Mr. Subrat Kumar Swain**, for their constant support and motivation, despite their extremely busy schedule. Our interactions with them always resulted in new ideas and proved beneficial towards our work. Without their constant presence and supervision our work would not have been successful.

We are very grateful to **Dr. Srikanta Pal, Professor and Head, Department of Electronics and Communication Engineering, Birla Institute of Technology, Mesra** for extending all the facilities at all times for pursing this course.

And lastly our batch mates who have always been there with valuable suggestions and support in our endeavors.

Shiladitya Biswas	**Srivatsa Sinha**	**Gajal Vasita**
BE/10081/2014	**BE/10327/2014**	**BE/10308/2014**

ABSTRACT

In this project we propose to develop a cloud based self-driving car prototype. Self-Driving Car are the next generation transportation which promises to provide high level of on road security and safety by taking control of our vehicles and promoting us smother and fast travel experience.

There are basically two types of models that can be designed. One is perception based model where we use array of sensors to capture and hand craft features for localization and mapping purpose. But current work in A.I based Computer Vision System like Convolution Neural Network enable us the reduce lot of these manual effort as CNNs itself act as feature extractor from raw images and put them on in the classifier to give direct result. This model is mainly inspired from Nvidia's research is called end to end model. Still processing would require high hardware cost. We therefore try to abstract processing into cloud and aim to control various cars in parallel.

CONTENTS

LIST OF FIGURES

INTRODUCTION

The rise of the automobile has transformed our cities and culture. As the populations in and around cities have grown, so have traffic and car crashes. Improved structural designs and features such as anti-lock brakes have helped to reduce the number of accidents and the severity of injuries caused by those accidents in the last three decades. The number of crashes fell from 6.7m in 1988 to 5.7m in 2013. But the economic losses of car crashes still amounted to $424bn, according to a government report on 2010 data. This has increased the demand for self-driving cars. There is a large scale Research and Development currently going on in this field. Advocates of self-driven cars say these can substantially reduce the 1.24 million deaths that happen in road accidents every year .With the recent availability of High performance computers equipped with Graphics processing units (GPU) we are able to focus in this field

The Solutions are:

a) Sensor-Based Driving:

The automotive industry is currently developing sensor-based solutions to increase vehicle safety in speed zones where driver error is most common: at lower speeds, when the driver is stuck in traffic, and at higher speeds, when the driver is cruising on a long stretch of highway. These systems, known as Advanced Driver Assist Systems (ADAS), use a combination of advanced sensors, such as stereo cameras and long- and short-range RADAR, combined with actuators, control units, and integrating software, to enable cars to monitor and respond to their surroundings. Some ADAS solutions, such as lane-keeping and warning systems, adaptive cruise control, back-up alerts, and parking assistance, are available now. Many others are in the pipeline.

b) Connectivity-Based Solutions

Connected-vehicle systems use wireless technologies to communicate in real time from vehicle to vehicle (V2V) and from vehicle to infrastructure (V2I), and vice versa. (Note that we use the expression V2X as shorthand for communication between vehicles and any other object.)According to the USDOT, as many as 80 percent of all crashes—excluding those in which the driver is impaired—could be mitigated using connected-vehicle technology. Dedicated Short-Range Communication (DSRC), which uses radio waves, is currently the leading wireless medium for V2V communication. It operates at 5.9 GHz frequency, using standards such as SAE J2735 and the IEEE 1609 suite (protocols that establish what messages are sent, what the messages mean, and how they are structured),and is being tested rigorously to see if it can fully support V2V cooperative safety applications. Currently, DSRC offers the greatest promise.

c) Convergence-Based Solution

The convergence of communication- and sensor-based technologies could deliver better safety, mobility, and self-driving capability than either approach could deliver on its own. As Pri Mudalige, staff researcher for General Motors' Global R&D, puts it, "V2V technology...may simplify the all-sensor-based automotive advanced driver-assist systems, enhance their performance, and make them more cost effective.

Big companies like Google, Tesla, Uber, BMW, Volvo etc. are giving high emphasis on building and testing Self-Driven cars. Tesla Motor has already launched its third Model of Autonomous Electric Vehicle namely Tesla Model S 3. BMW is overhauling its research and development activities to focus on these vehicles. The company is updating its zero-emission vehicles after a lackluster response to its only fully battery-powered car, the i3, which recorded only 25,000 sales last year. By contrast, Tesla already has more than 370,000 orders for its Model S 3. Uber is also developing self-driving truck technology to move goods more safely and cost effectively around the world.

Meanwhile, in India self-driven cars are comparatively less due to challenging road conditions and several other factors. Still companies like Ola and Myles have recently started self-driven car services in Delhi, Bangalore and Kolkata.

DESIGN AND IMPLEMENTATION OF PROTOTYPE

1. Wooden Frame

In this section the basic structure and model of the car is described. Light weight and sturdy wooden ply was used to build the frame of the car. The car is fitted with two rear wheel motors and a front servo motor. We have used the Fifth Wheel mechanism in order to steer the car. The front wheels are completely free to rotate about its axis of rotation. The angle of rotation of the front axle is controlled by the servo motor. In the upcoming sections we will discuss the variation of the rear wheel speeds w.r.t the angle of rotation of the front axle.

2. Brief History of Fifth-Wheel Mechanism

The term Fifth Wheel comes from a similar coupling used on four-wheel horse-drawn carriages and wagons. The device allowed the front axle assembly to pivot in the horizontal plane, to facilitate turning. Basically a wheel was placed on the rear frame section of the truck, which back then only had four wheels; this wheel that was placed on the frame was the "fifth wheel", hence the name. The trailer needed to be raised so that the trailer's pin would be able to drop into the central hole of the fifth wheel.

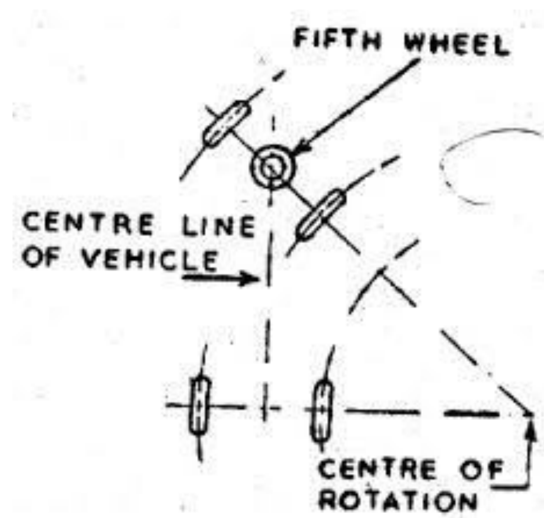

Fig 1: Fifth Wheel Principle

The invention of the fifth wheel for motorized trucks is often credited to US inventor Charles H. Martin of the Martin Rocking Fifth Wheel Co. who invented the device in 1915. It was submitted for patent in 1915, and finalized in 1916. Herman Farr invented it, and Martin hooked up with him and became the assignee. When they formed the Martin Fifth Wheel company Martin was president and Farr was named secretary. The 5th wheel was Farr's invention. Martin capitalized on it.

3. Component selection and installation

a) Rear Motor Selection:

In this section we primarily deal with component selection. We have assumed the payload of the car to be about 0.5 kg (the real payload weight is less than 0.5 Kg; we chose higher payload value just to be on the safer side). In order to make the car move successfully we need to calculate the minimum torque that each rear motor should apply.

Payload weight, $W = 0.5$ kg

Coefficient Of Friction, $\mu = 0.6$ to 0.85 (for rubber-dry concrete contact)

Number of motors, $N = 2$

Radius of Wheel, $R = 3.5$ cm (Approx.)

Maximum Frictional force that each motor has to overcome, $F = (W * \mu)/N$

$F_1 = 0.15$ N , when $\mu = 0.6$

$F_2 = 0.2125$ N, when $\mu = 0.85$

Range of frictional force $F_1 < F < F_2$ ------------------------(1)

This is the maximum range of frictional force that the floor exerts on the two wheels.

As a result the maximum Torque exerted will be in the range, $T = F * R$

$T_1 = 0.525$ N-cm

$T_2 = 0.744$ N-cm

Range of frictional Torque $T_1 < T < T_2$ ------------------------(2)

From equation (2) we see that in order to make the car move properly we need a motor whose torque is greater than 0.744 N-cm. We have selected the IG-32 Planetary gear DC Motor that has a rated Torque of 11 N-cm. These motors were fixed using an L bracket to the rear axle.

b) Servo motor selection for steering:

As already calculated in equation (2) any servo with rated torque greater than T_2 will be able to rotate the front axle about its pivot point. Tower pro MG995 having a rated torque of 9.4 N-cm is used to steer the car.

c) Other Electronics components:

A Raspberry Pi 3B minicomputer was used to perform the necessary computations required for the car to move. The Raspberry Pi acts as a server for the android app. The app sends the angle data to the minicomputer. A mathematical formula is derived in the next section that takes steering angle as input and gives the ratio of the Speeds of the left and right rear wheel of the car as output. The complete circuit box is shown below

d) The power circuit design:

A simple circuit was designed and hand soldered on a Vero-board that provides power to both the DC motors and the front servo motor. The circuit block diagram and finished power circuit is shown below.

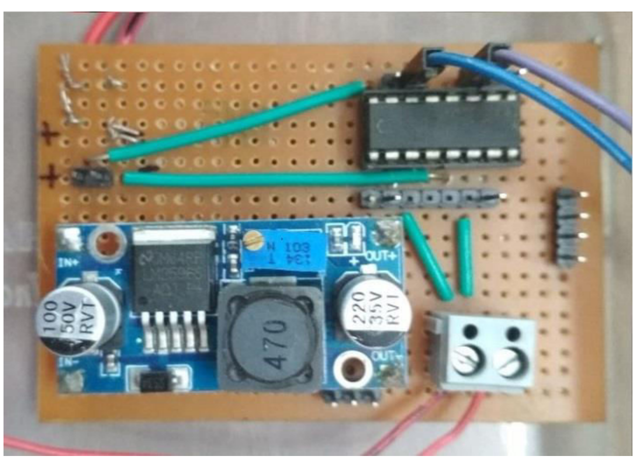

Fig 2: Power Circuit Board

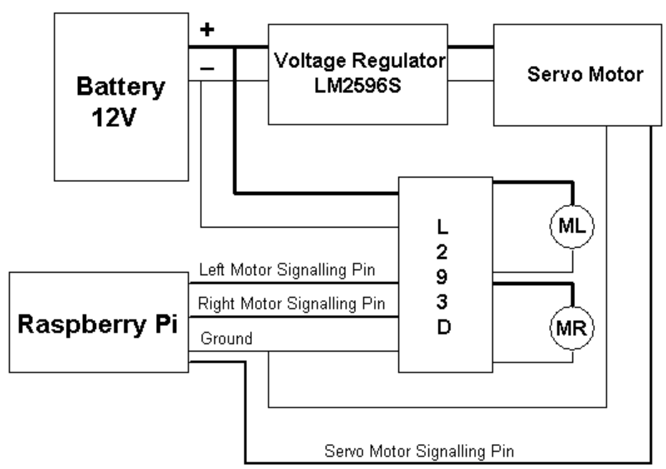

Fig 3: Power Circuit Board Block Diagram

4. Car Control Equations

As our main aim in this project was to build a Convolutional Neural Network, less attention was given to the hardware design of the car. More emphasis was given to the basic control equations that are necessary for the smooth locomotion of the car. These equations are derived in this section. See [9] and [10]

a) Servo Control:

It is a well-known fact that the position of a servo motor is dependent on the duty cycle of the pulse applied to the signaling pin of the servo as shown in figure below.

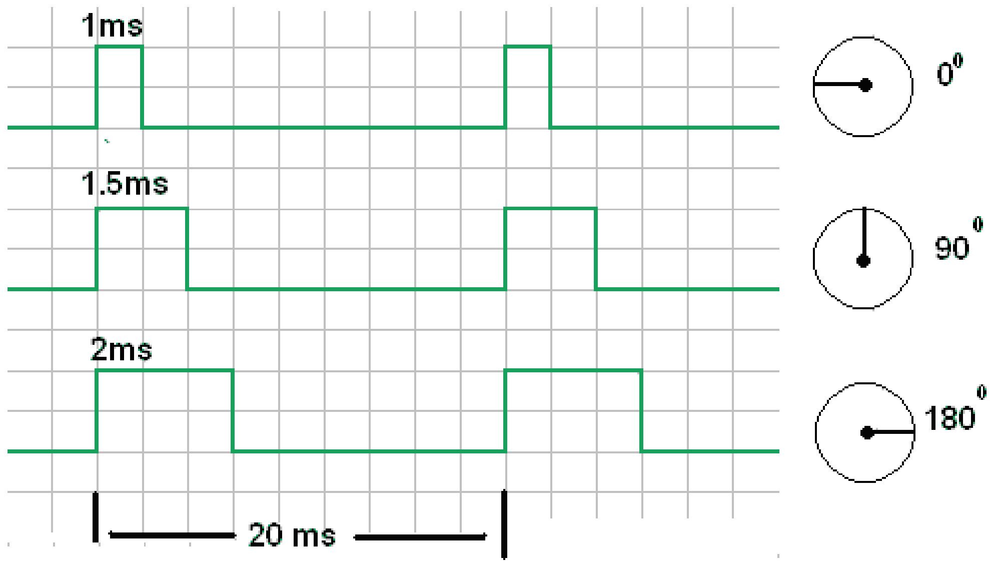

Fig 4: Servo control using PWM

We constrained the angle sent from the App to the car in between -30 to +30 (i.e. Left turn and Right turn). The sign convention is such that when the front axle turns left we have negative angle and when the front axle turns right we have positive angle. The percentage duty cycle of the servo for -30 and +30 was found out to be 20% and 80% respectively by trial and error method. Solving for a linear equation of the form,

$$Y = (M * X) + C$$

Substituting for Y= DutyCycle and X= Angle from App We get,

DutyCycle= (M * (Angle from App)) + C …………………………….. (3)

Solving equation (3) for M and C by putting DutyCycle= (20 , 80) and Angle from App= (-30, +30) respectively we get,

$$M=1 \text{ and } C=50$$

Thus the final servo control equation is:

DutyCycle=(1*(Angle from App)) + 50

b) Mathematical model of rear motor speeds:

As we all now in a four wheeled vehicle the speeds of the rear wheels are different from each other when it moves in a curvilinear motion. A dedicated system of gear mechanism fondly known as the Differential gear box is used for the smooth traversal of curvilinear paths by the vehicle. In this project we tried to achieve this functionality mathematically. The speeds of both the DC motors are controlled by the Hardware PWM pins [3] of the Raspberry Pi. More the duty cycle more is the speed of the DC motor and vice-versa. It is also known that when a vehicle traverses a curvy path the rear wheel closer to the center of the curve has low speed and the rear wheel away from the curve center has proportionally high speed. Thus a mathematical formula can be obtained that takes that takes steering angle as input and gives the ratio of the Speeds of the left and right rear wheel of the car as output.

From Figure below we get:

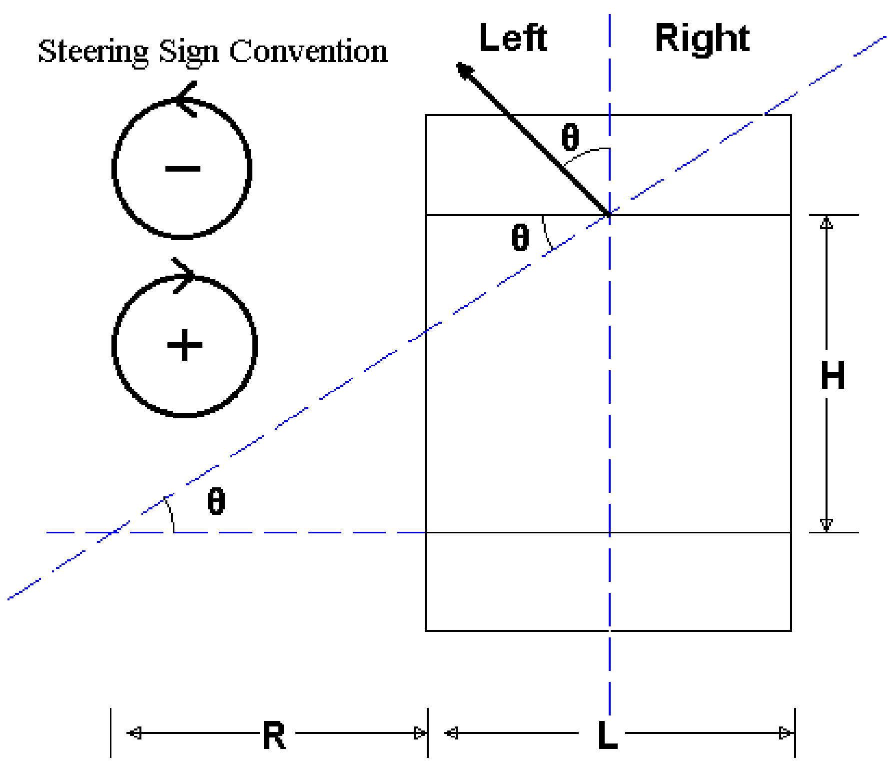

Fig 5: Simplified diagram to find speed ratios of rear wheel motors. The steering sign convention is shown on top left

$$\frac{H}{\left(R+\left(\frac{L}{2}\right)\right)} = tan\theta \text{ .... (4)}$$

Instantaneous Radius of curvature of left wheel, $\mathbf{R_L} = \mathbf{R} = \frac{H}{tan\theta} - \frac{L}{2}$ ……. (5)

Instantaneous Radius of curvature of right wheel, $\mathbf{R_R} = \frac{H}{tan\theta} + \frac{L}{2}$ …….......... (6)

It is evident from basic geometry that,

$$\frac{Instan.Radius\ of\ Curvature\ of\ Left\ Wheel/Motor}{Instan.Radius\ of\ Curvature\ of\ Right\ Wheel/Motor} = \frac{Speed\ of\ Left\ Wheel}{Speed\ of\ Right\ Wheel}$$

Thus,

$$\frac{Speed/DutyCycle\ of\ Left\ Motor}{Speed/DutyCycle\ of\ Right\ Motor} = \frac{\frac{H}{tan\theta}-\frac{L}{2}}{\frac{H}{tan\theta}+\frac{L}{2}} \text{ .... (7)}$$

To keep things simple we assumed that the rear motors rotated with half its maximum speed when the angle sent by the app is 0 degree or the car is following a rectilinear path. The motor speed is half its maximum when the PWM duty cycle is 50%.

Rearranging equation (7) we get:

$$\frac{Speed/DutyCycle\ of\ Left\ Motor}{Speed/DutyCycle\ of\ Right\ Motor} = \frac{2H - Ltan\theta}{2H + Ltan\theta}$$

Thus,

Speed /DutyCycle of Left Wheel= $K * (2H - (L * tan\theta))$ …………(8)

Speed/DutyCycle of Right Wheel= $K * (2H + (L * tan\theta))$ ……..…(9)

On a trial and error basis method we can select determine the value of K. The value of K can be approximate by putting the value of $\theta = 0$ and DutyCycle = 50 in equations (8) or (9).

5. Data Acquisition

Data acquisition system is used to acquire and store data. In our case we store image data and get the angle associated with this image. We used Pygame a Python library to capture and store the data using a webcam connected to the minicomputer. We label the image with a serial number and the angle of steering separated by an underscore. Eg. 3_ +20.jpg, 2_-19.jpg, etc.

These labeled images are used to train the Convolutional Neural Network and hence predict the steering angle only then the whole car is Self-Driven in nature. The image or data is captured and saved at a regular time interval. During this phase the car is controlled by an Android App. Both the Android Mobile Phone and the minicomputer are connected to a Wi-Fi network. The User has to enter the IP address and the port number of the Car/ Minicomputer into the Android App. As soon as START is pressed the car starts to move and acquire data. The user just has to guide the car between the obstacles.

6. Self-Driving Algorithm

This is the last part of the code or algorithm. Pygame is again used to get a snapshot of the front view of the car. This image is then uploaded to a cloud. A High performance computer (Here: Nvidia Jetson TK1) connected to this cloud downloads it. This image is fed to a pre-trained Convolutional Neural Network that predicts the steering angle of the car. As per the equations derived in section 4 above the rear wheel motor speeds are automatically adjusted. **Thus a Self-Driven Car is born.**

The completed car is shown beside:

Fig 6: Completed Car

ARTIFICIAL INTELLIGENCE BASED ALGORITHM

1) Overview of Artificial Intelligence

Artificial Intelligence is the intelligence demonstrated by the machine, or computers in contrast to the natural intelligence which is characteristics of living being mainly humans. As a human we are capable of doing lot of task which seems trivial to us but is actually of great difficulty for machines. Consider recognizing an object in an image its highly trivial task and even a year or two old baby can do this simple task but same task when given to computer, whatsoever be its computational speed and accuracy it fails at some point simply because computer needs to be instructed each and everything and it's more like memorizing then actual learning which takes place. Artificial Intelligent system is coded to do simple stuffs of human intelligence but in much faster and accurate way. Once an AI system is trained, over continuous improvement it easily tends to surpass human level performance.

Self-Driven Car

As object recognition is a problem of intelligence so is it to control the car, except that former is not trivial than latter. If simple object recognition is so difficult, one can imagine the complexity and processing required to drive the car. When we see a road in front of us we see this fig 7, but when computer sees the same image it sees fig 8.

Fig 7: How Humans see a Road

81	80	79	75	74	74	75	75	72	70	68	66	64	63	63
81	80	78	75	74	74	74	75	72	70	68	66	64	63	63
80	79	78	74	73	73	74	74	71	69	67	65	64	64	64
79	78	77	73	73	73	73	73	71	68	67	65	64	64	64
79	78	76	73	72	72	72	72	70	68	66	65	64	64	65
78	77	75	72	71	71	71	72	69	67	66	65	65	65	66
77	76	75	71	70	70	71	71	68	67	66	65	65	65	66
77	76	74	71	70	70	70	71	68	66	65	65	65	65	66
78	76	74	72	72	72	70	67	68	67	66	65	64	62	62
81	78	76	73	73	73	71	68	68	67	66	65	64	62	62
85	82	79	75	75	75	73	71	69	67	66	65	64	63	62
91	87	83	78	77	77	75	73	70	69	68	67	66	65	64
98	93	87	81	80	79	78	76	74	72	71	70	69	68	67

Fig 8: How a computer sees a Road

By this description one can easily understand the problems that lie behind simple image recognition task. We therefore turn towards algorithms proposed in AI to solve this problem.

When we drive the car we track and process lot of information, the distance of the car in front of us, the width of the road, pedestrian walking around, the traffic condition, our speed, various driving rules, etc. And we see all this through our eyes and we tend to make designs in fraction of a second. But how to make computer do this task?

2) Overview of Neural Networks

Neural networks are the classes of Artificial Algorithms, also known as Cognitive Algorithms, are computing systems which are vaguely inspired by biological neural network which are part of human brains. As a human we “learn” to do certain task by considering lots of examples. Say when we need to teach a kid English Letters we start by showing them the English alphabets and over the time they learn to recognize the characters and then words and so on. On similar lines a mathematical model of neuron was designed, which is basically takes input as certain numbers then take its weighted linear combination followed by a non-linearity to give the output. Shown below is the first ever proposed artificial neuron, the McCulloch Pitts model :

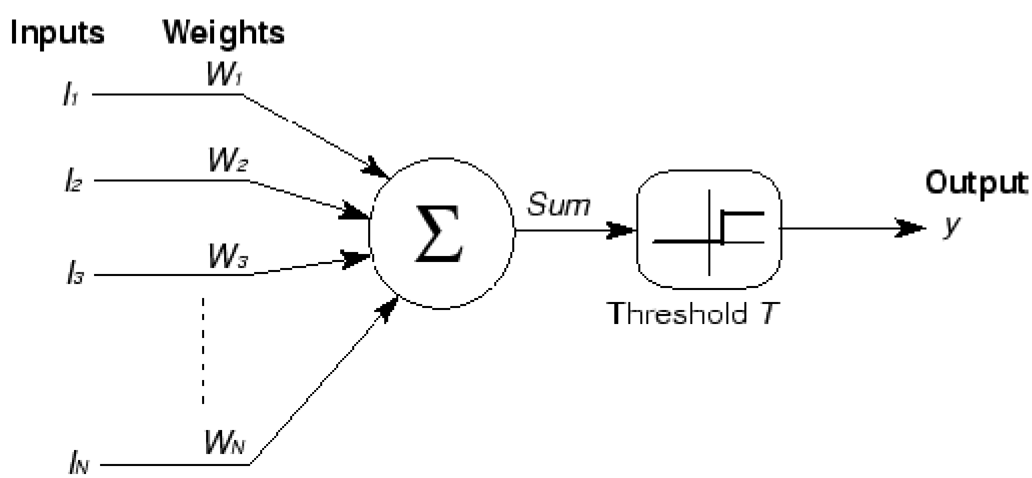

Fig 9: McCulloch Pitts Model

An ANN is basically based on a collection of connected units or nodes such artificial neurons (a simplified version of biological neurons in an animal brain). Each connection (a simplified version of a synapse) between artificial neurons can transmit a signal from one to another. The artificial neuron that receives the signal can process it and then signal artificial neurons connected to it. It consists of input layer where we pass our data, the hidden layer which processes the data and acts as a feature extractor and output layer which acts classification unit.

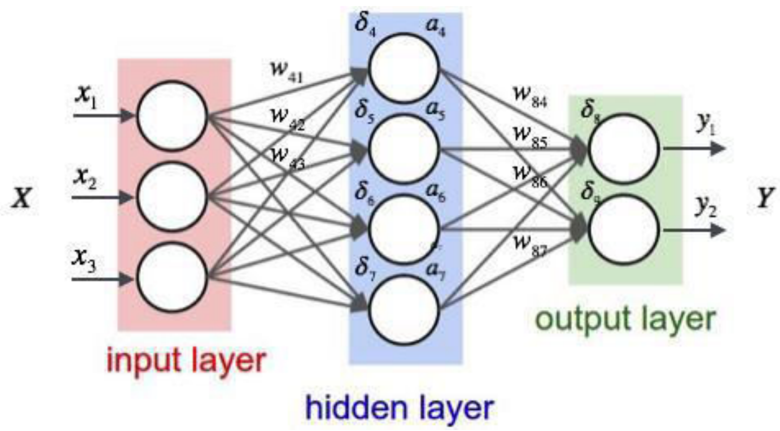

Fig 10: Multilayer Artificial Neural Network

3) Overview of Convolution Neural Network

Convolution neural network are special class of neural network which was invented by Yann LeCun which specializes in the field of Computer Vision tasks like image recognition system, image captioning system etc.

CNNs are also vaguely inspired from working of human eye. Work by Hubel and Wiesel in the 1950s and 1960s showed that cat and monkey visual cortexes contain neurons that individually respond to small regions of the visual field. Provided the eyes are not moving, the region of visual space within which visual stimuli affect the firing of a single neuron is known as its receptive field. Neighboring cells have similar and overlapping receptive fields. Receptive field size and location varies systematically across the cortex to form a complete map of visual space. The cortex in each hemisphere represents the contralateral visual field. On similar lines a Convolution network performs series of convolution operation over image with the filters which are trained to labeled examples to give the result. A general architecture of CNN is shown below.

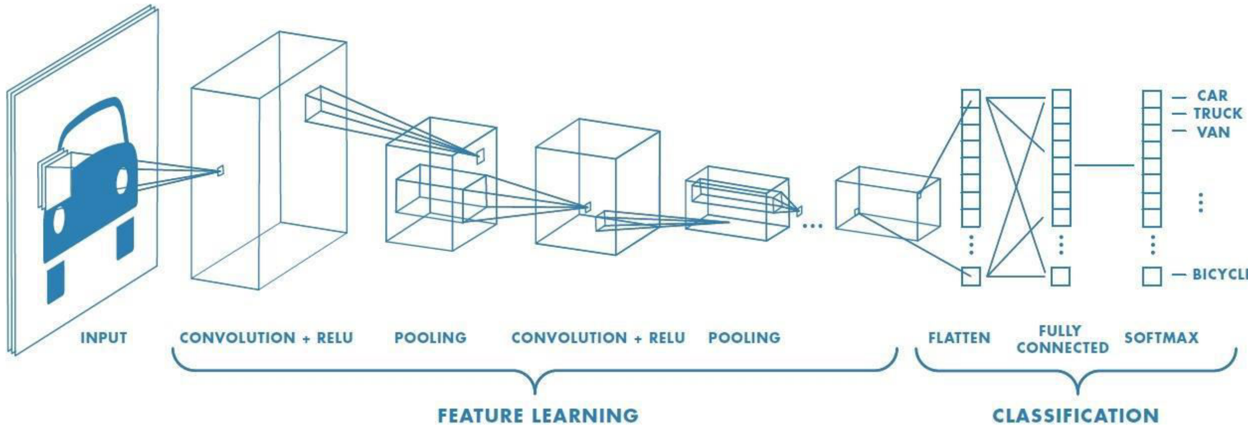

Fig 11: General Architecture of Convolutional Neural Network

First we input our image has ***Height (H) X Width (W) X Number of Channels(C)***. We than perform convolution operation with ***K*** number of filters of dimension ***DxD*** which results in a feature map of dimensions ***(H-D+1) X (W - D + 1) X K.***

Few choices which we make here are the number of filters K, the dimension D of the filter and the stride of the operation.

Another kind of operation which we perform is polling operation whereby over a region we forget all the features except with the one having maximum value. This operations gives computer ability to be invariant of the location of the object.

CNNs have proved itself in solving lot of task but still remain a black box to the scientist considering the fact that no one actually knows why it works. Various hypothesis have been given for same though

4) Traditional Computer Vision Algorithm vs Convolution Neural Network

Computer vision can be succinctly described as finding telling features from images to help discriminate objects and/or classes of objects. While humans are innately endowed with incredible vision faculties; it is not clear which features humans use to get such strong performance on vision tasks. Computer vision algorithms in general work by extracting feature vectors from images and using these feature vectors to classify images.

In traditional systems we used to generally using mathematical models and formulas to ourselves generate features that would be distinctive and help computer classify and generalize image over a broad domain of data. This method though worked fine didn't had any intelligent component and drove lot of effort into handcrafting synthetic features with no guarantee to work, in fact most of them failed to generalize. These techniques would work only on the specific task, say distinguishing between cat and dog and would generally not extend to other animals. With the advancement of A.I techniques and breakthrough into the realms of CNNs we got some really good results. Back in 2012, Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever designed a popular CNN architecture AlexNet which broke all records of previous accuracy on Image Recognition task and got top-5 error of 15.3 %. This result sowed the seeds for another path of Computer Vision tasks solved mainly through Convolution Neural Network

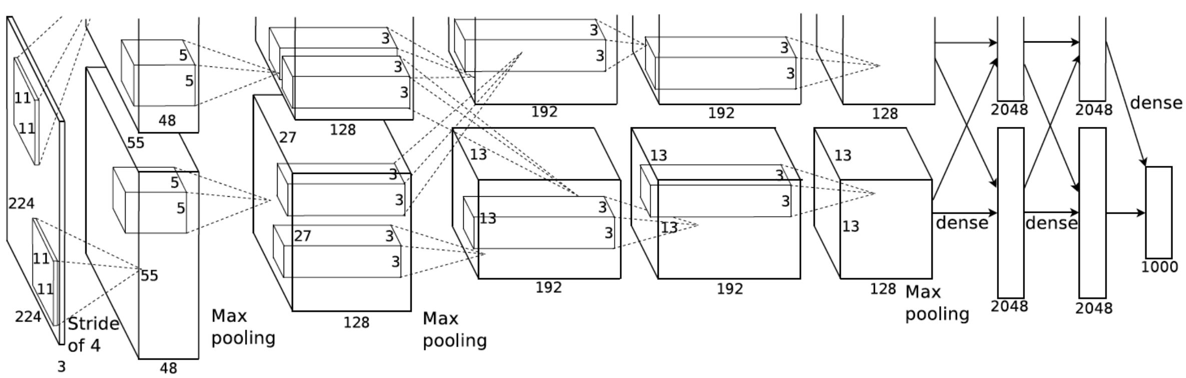

Fig 12: AlexNet

5) Self-Driving Car Models: Perception Based vs End to End Based

Designing a self-driving car involves lot of processing and even before that we need lot of information about the surrounding of car as discussed above. We can solve this problem using two approaches:

a) *Perception Based Model*:

In perception based model we tend to gather information about our surrounding through array of sensor data, this mainly consist of LIDAR, Distance sensors, Cameras, Speed sensors etc. Each of these information are then combined to various algorithms of computer vision and A.I to produce desired commands for driving a car. These commands are mainly in form of instruction for steering, speed control etc. These solutions are similar to traditional CV problem where we need to hard code lot of instructions to make car work and is generally costly due to expense of lot of sensors that are required to acquire various information. This is in contrast to how human behaves, where its mostly our eyes which we use and ears to drive cars. This encourages us to look alternatives into more economical and efficient alternative.

b) End to End Based Model:

With breakthrough of CNNs in various computer vision problems, another dimension towards solving of Self Driving Car problem is an End to End approach whereby we feed only the camera images and let CNNs itself learn the required features and produce the steering angle. This method has been described in Nvidia's work on End to End self-driving car model. This method drastically reduces the cost of equipment although still requires highly sophisticated machinery to process the images. Our Aim is to even reduce this cost by abstracting out all the common processes linked with self-driving car to public cloud datacenter like Amazon Web Services or Google Cloud Platform, thereby reducing the cost only to implement the hardware to control the mechanism automatically. [1],[2],[4].

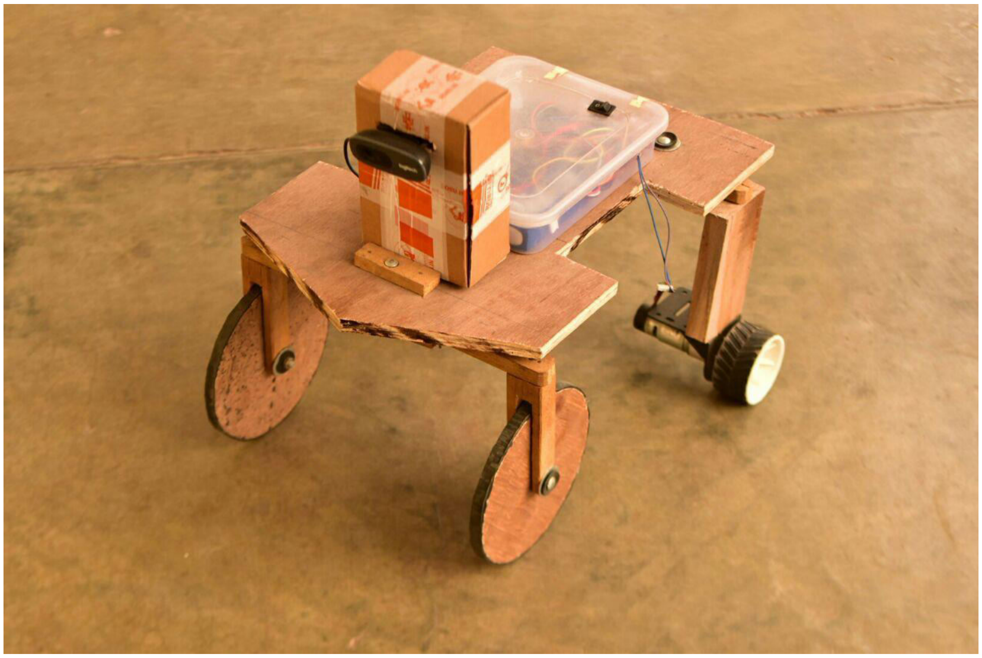

Fig 13: End to End Self-Driven Car

6) Our implementation of End to End Self Driving Car

a) Proof of Concept Design:

Instead of directly making a prototype we decided to first get some assurance as to how all these ideas will work in reality. Therefor as a proof of concept we built a small robotic car with our mobile camera mounted on it which continuously sent image to computer over Wi-Fi which acted as a cloud server for us.

We initially drove our car manually and recorded images worth of 5 hours of driving over a simple black and white path. We labeled each image with our control direction which was Left, Right or Forward. Next we trained a simple 3 layer neural network model.[6],[7],[8].

Specification of model is as follows:

Parameter	Value
No of Layers in Neural Network	3
Input Image Dimension	160 x120

Activation Function	Sigmoid
Dropout	0.3
Learning Rate	0.001
Decay	1e-06
Momentum	0.9
Loss	Categorical Cross entropy
Number of Epoch	100
Batch Size	30

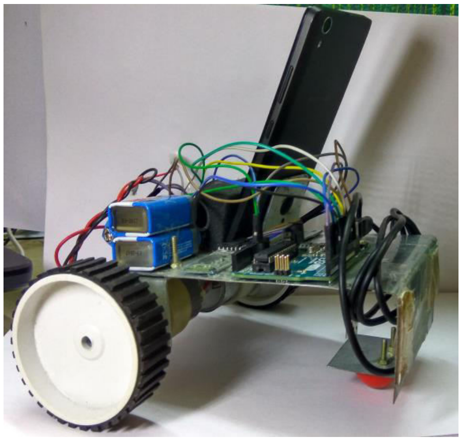

Fig 14: Simple Differential car to prove our concept

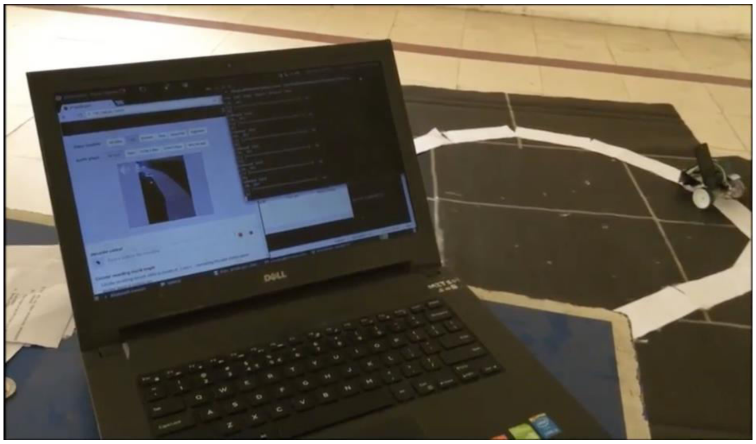

Fig 15: Self-Driven car following whit line. Direction commands sent from laptop to the car via TCP Socket

b) Proposed Architecture for Actual Design of Prototype

Successful Implementation of Prototype gave us confidence to pursue actual implementation of the prototype. For this we proposed to follow the work of Nvidia's Research into End to End Self Driving Car Model. The architecture consists of series of convolution and max polling operation that outputs the angle at the end of its output. Shown below is the architecture.

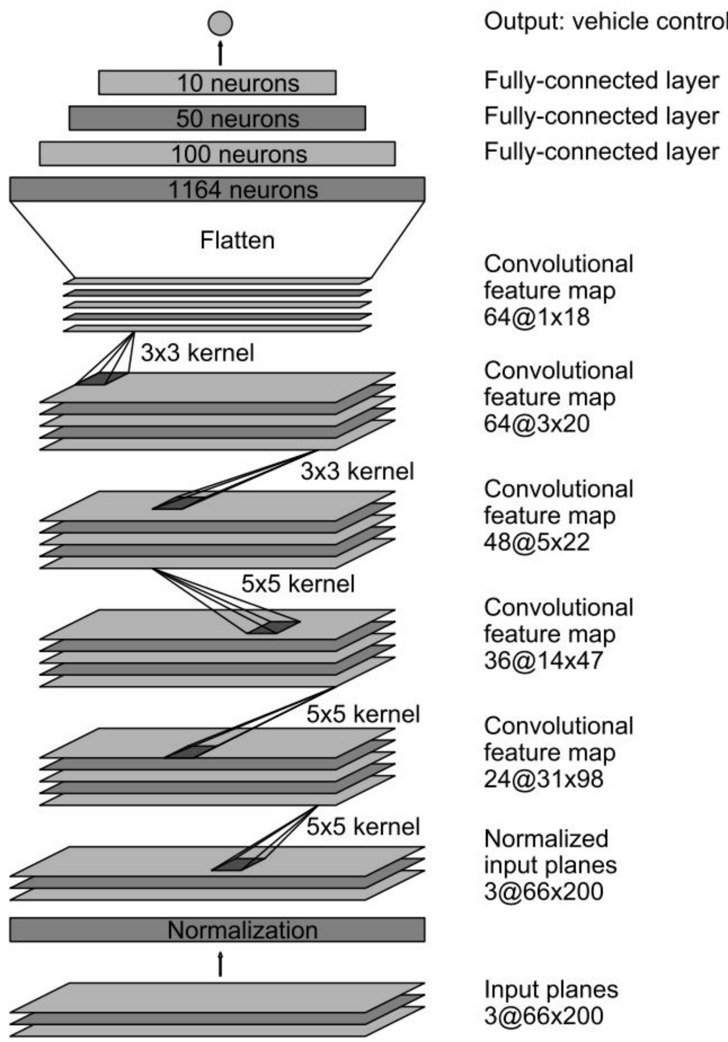

Fig 16: CNN Architecture implemented in this project

We tried to implement this model on our data collected from the robotic car, it gave slight improvement but not something we expected. We kept other parameters same. This was attributed to high noise in the image and various lightning conditions. Further above architecture gave way more degree of freedom to our car than required. This would generally result in over fitting and failure of generalization

RESULT

This section shows the result we obtained from our proof of concept project. The images obtained from the camera were fed into a 3 layer Artificial Neural Network. Our network was able to predict the direction successfully and the car traversed a white line of any curvature.

The table below shows the predicted and actual directions corresponding to the image captured by the camera:

INPUT IMAGE	PREDICTED DIRECTION	ACTUAL DIRECTION
	FORWARD	FORWARD
	RIGHT	RIGHT

	FORWARD	LEFT
	FORWARD	RIGHT

As far as our main Car is concerned it is able to acquire and store real time images captured by the camera. The images are stored in a folder named “Data” within the Mini-computer i.e. Raspberry Pi. Eighty percent of the total data is used to train the Convolutional Neural Network (CNN) and the remaining twenty percent is used to validate the CNN. We were able to process the images with AWS GPU (Nvidia tesla K80) servers at the rate of 10 FPS for raw road data. Unfortunately due to delay in fund sanction and time shortage we weren’t able to procure the processing unit capable to run a CNN live, on the go. Hence our real Car Prototype wasn’t able to navigate a path on its own on its own.

CONCLUSION

We were easily able to design our proof of concept model for self-driving car using a small robotic device steered over Wi-Fi. We achieved a decent accuracy of 0.75 on this model. This was due to the fact that we had lot noise in the background, although we performed some manual image processing like thresholding and morphological operation. Further we tried to run same data on the Convolution Neural Network but we did not get much improvement. This was attributed to the fact that CNNs allowed much more degree of freedom than simple 3 layer neural network. This amount of freedom would have given rise to over fitting rather than generalizing over all the Images.

In actual prototype implementation we were able to design a working car. This was a simple wooden model and utilized fifth wheel mechanism to steer. Further to control the car we were also able to develop a Smart Phone (Android) Application. The Application relied on the Socket Connection to send data to the public server who's IP and Port was entered by us. It used the mobile sensors to identify the inclination of the mobile and correspondingly turn the front wheels of the car. Here we faced a little glitch in synchronization of fast incoming data and sending same to hardware but it was overcome through Hardware PWM implementation. We were finally able to write codes to collect the data.

FUTURE WORK

Self-Driving Car as said before are the next generation travelling experience which promises smooth and wonderful travelling experience. This technology if accepted by masses has potential to eradicate traffic problems and enforce traffic rules which are some of the basic problems in Developing country like India. Further I believe that we can better our model, and try to work on some actual electrical based cars in closed environment setup. Our idea have potential to reduce cost of the self-driving cars drastically, this would also have environmental impacts as electrical cars won't create pollution in the environment. We can connect a bunch of cars and synchronize them to coordinate and control their speed to avoid collision thereby eliminating need to traffic lights, polices etc. thereby reducing traffic conditions. This idea has potential market in India and can be turned into a good business model. Big firms like Mahindra have already started experimenting technologies by hosting contest across premier colleges in India to develop same.

FUNDING TIMELINE AND CURRENT STATUS

Date of Progress Presentation **:** 09/05/2018

Date of Project Sanctioned **:** 27/09/2017

Amount Sanctioned **:** Rs. 49,614

Amount Cheque Received from Accounts Office **:** 09/02/2018

Items Order on 21st Feb 2018:

S.No	Items Procured	Qty	Cost(Rs.)	Total (Rs.)
1	Planetary Gear DC Motor	2	Rs 3664/-	Rs 7328/-
2	L Bracket Clamp	2	Rs 350/-	Rs 700/-
3	Turnigy 2200mAh 3S 30C Lipo Pack	1	Rs 3527/-	Rs 3527/-
4	Turnigy Accucel-6 50W 6A Balancer Charger	1	Rs 5781	Rs 5781/-
5	Circular Polarizer Filter	1	Rs 449	Rs 449/-
				Rs 17785/-

Items Received on **:** 14/03/2018

Sanctioned Amount left **:** Rs. 31,829/-

PUBLICATIONS

1) Srivatsa Sinha "**Review of Architecture and Optimization on Intel Xeon Scalable Processors in context of Intel Optimization TensorFlow on Intel AI DevCloud.**"
2) Poster Presentation at Intel Student Ambassador Summit to be held from 22nd - 25th May, 2018

REFERENCES

[1] N. S. Yeshodara, N. S. Nagojappa, and N. Kishore, "**Cloud based self-driving cars,**" in Cloud Computing in Emerging Markets (CCEM), 2014 IEEE International Conference on. IEEE, 2014, pp. 1–7.

[2] Y. Ukidave, D. Kaeli, U. Gupta, and K. Keville, "**Performance of the nvidia jetson tk1 in hpc in Cluster Computing (CLUSTER)**", 2015 IEEE International Conference on. IEEE, 2015, pp. 533–534.

[3] Raspberry Pi Pigpio Libraray: "http://abyz.me.uk/rpi/pigpio/".

[4] Sangamesh Ragate and Ryo Asai**, "OPTIMIZATION OF REAL-TIME OBJECT DETECTION ON INTEL® XEON ® SCALABLE PROCESSORS"** a tutorial draft by Colfax International

[5] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski et al, "**End to End Learning for Self-Driving Cars**" , arXiv:1604.07316, 25 April 2016

[6] Martin Abadi, Ashish Agarwal et al., "**TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems",**

[7] Karen Simonyan and Andrew Zisserman "**VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION**" , arXiv:1409.1556v6, 10 April 2015

[8] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. "**You Only Look Once: Unified, Real-Time Object Detection**" arXiv:1506.02640, 9 May 2016

[9] Cherry Myint , Nu Nu Win, "**Position and Velocity control for Two-Wheel Differential Drive Mobile Robot**", IJSETR, Vol-5, Issue-9, September 2016

[10] Farhan A. Salem, "**Dynamic and Kinematic Models and Control for Differential Drive Mobile Robots**", International Journal of Current Engineering and Technology ISSN 2277 - 4106