Proceedings of the 2nd International conference on Electronics, Communication and Aerospace Technology (ICECA 2018)
IEEE Conference Record # 42487; IEEE Xplore ISBN:978-1-5386-0965-1

Working model of Self-driving car using Convolutional Neural Network, Raspberry Pi and Arduino

Aditya Kumar Jain
Electronics and Communication Department
Dharmsinh Desai University
Gujarat, India
adityame999@hotmail.com

***Abstract*— The evolution of Artificial Intelligence has served as the catalyst in the field of technology. We can now develop things which was once just an imagination. One of such creation is the birth of self-driving car. Days have come where one can do their work or even sleep in the car and without even touching the steering wheel, accelerator you will still be able to reach your target destination safely. This paper proposes a working model of self-driving car which is capable of driving from one location to the other or to say on different types of tracks such as curved tracks, straight tracks and straight followed by curved tracks. A camera module is mounted over the top of the car along with Raspberry Pi sends the images from real world to the Convolutional Neural Network which then predicts one of the following directions .i.e. right, left, forward or stop which is then followed by sending a signal from the Arduino to the controller of the remote controlled car and as a result of it the car moves in the desired direction without any human intervention.**

Keywords— Artificial Intelligence, Self-driving car, Convolutional Neural Network, Raspberry PI, Arduino

I. INTRODUCTION

Globally speaking, nearly 1.3 million people die in road crashes each year, on average 3,287 deaths a day [1].And talking about India the number of people who were killed in a road accidents in 2013 alone were 1, 37,000 [2].Speeding, talking over phone, drunk driving and breaking traffic rules are the root causes behind these accidents and the statistics are rising day by day which is now becoming a major concern. No matter how hard we try to create awareness regarding traffic rules and safety that has to be followed while driving, accidents are still occurring and aren't showing a sign to stop. Though human errors can never be eliminated, but accidents can definitely be stopped. And in this case technology has surely come to our rescue. Starting from the very early radar-based collision detection to present day's technology, the advancement and improvement in this technology had seen an exponential growth in recent years.

Self-driving cars is the one of the most discussed technology of current scenario. What was once imagined is a reality now. The definition Self-driving cars is a car which promises to take the traveler to their destination with minimal human control while taking safety as its first priority. Many companies throughout the world are making a serious and continuous efforts to make driving a safe and risk free process and have started building prototypes for the same. Amongst these companies are Google, Tesla Mercedes and many more who have built a successful and functioning prototype and are planning to release a model in the upcoming years.

Self-driving cars are expected to have faster reflexes than humans, make more reliable judgments, thus avoiding mere faults which causes accidents at the first place. Apart from saving precious lives, other advantages these technology gives is better traffic flow regulation because unlike humans these cars ride with proper traffic rules, making rides smooth and congestion free. Self-driving cars can also help in tackling parking space issues by allowing to create a taxi/pooling service for the unused cars and by unused car we mean to say the car that is either staying for few hours while the owner is at work or the car that is in the garage while the owner is out for a vacation. Thus we could make better use of land instead of using it for parking space. The basic model of any Autopilot system involves radar, a front-facing camera, a digitally-controlled digital braking system, and long range ultrasonic sensors located around the car. Radars enables the detection of vehicles and other moving objects around the car, front facing camera helps to detect and recognize objects like cars, trees, driving lane, humans, traffic signals and other important data. All these information are taken in real-time environment and are fused into a learning network which then predicts the car's response accordingly.

II. RELATED WORK

The work done by Chun-Che Wang, Shih-Shinh Huang, Li-Chen Fu and Pei-Yung Hsiao [3] aims to improve driving, by creating an assistance system. To enhance driver's safety at nighttime the algorithm includes lane detection along with vehicle recognition system. Lane detection helps to localize the markers so that the lane can be detected while vehicle recognition involves taillight extraction along with taillight paring algorithm. Another research work [4] done by

Xiaodong Miao, Shunming Li and Huan Shen models to locate the positions of road lane in real-time. Operation like canny edge extraction is done to extract edge map to which matching technique is applied followed by the selection of potentials edge points. Finally linking is done to localize the lane lines. In [5] Anik Saha, Dipanjan Das Roy, Tauhidul Alam and Kaushik Deb aims to convert the image from RGB format to gray-scale format. Then flood-filling algorithm was applied to label the connected components. Then the largest connected component is extracted which is nothing but the lane. The model proposed by Gurjashan Singh Panna, Mohammad Dawud Ansari and Pritha Gupta [6] in developing a prototype of autonomous car involves implementation of lane detection algorithm along with obstacle detection. Their project aims to build a monocular vision autonomous car prototype which is capable of reaching at a particular destination safely. Another work model proposed by R.Mohanapriya, L.K.Hema, Dipeshwarkumar Yadav, and Vivek Kumar Verma [7] and Ms. D.D Jadhav, Komal Jadhav , Kajal Shinde , Anjali Sonawane [8] are similar which involves equipping GPS and GSM system on a 4 wheeled robot. The GPS system steers the robot and is capable of reaching from one point to another without any human intervention. While in the former one with the help of GSM system they promise to report theft in case is there is any. An SMS alert is sent to the vehicle owner reporting about the issue and as a result of it, the owner of the car can switch the ignition off and in the latter one the project states that vehicle can only be turned on if the authorized person sends a predefined location to the car. In [9] Dhanasingaraja R, Kalaimagal S, Muralidharan G developed a system that takes the current position and gets the user destination. Then the system finds the shortest path to the destination and also extracts features like latitude, longitude from the graph. So in a nutshell it helps in navigation as well as monitoring the car.

In above mentioned models, the art of taking decision (left, right forward or stop) was merely based on Image Processing techniques while in some of them, it relies on GPS system to take the action.

With all the work done before and after analyzing and studying thoroughly, the paper presents a new model which involves Machine Learning as well as Image Processing. Image Processing helps in preparing the input image. The image is resized, converted into grayscale. Once it is completed the image is fed into the Convolutional Neural Network. The output of Neural Network helps in taking the directional decision. Once it is done, the controller simple sends the corresponding signal and the car moves in that particular direction.

III. Proposed Model

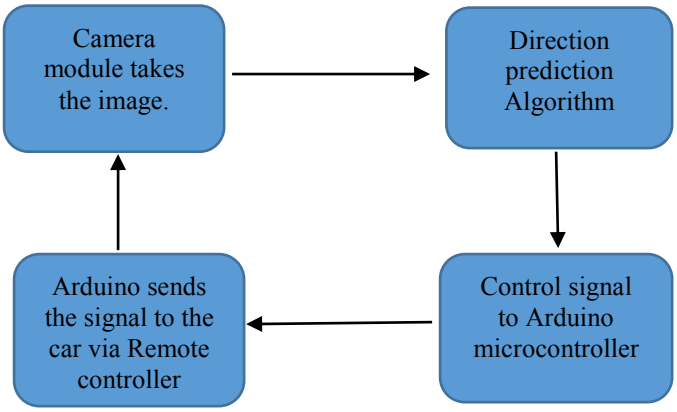


Fig.1 Proposed self-driving model

The proposed model takes an image with the help of Pi cam attached with Raspberry Pi on the car. The Raspberry-Pi and the laptop is connected to the same network, the Raspberry Pi sends the image captured which serves as the input image to the Convolutional Neural Network. The image is gray-scaled before passing it to the Neural Network. Upon prediction the model gives one of the four output i.e. left, right, forward or stop. When the result is predicted corresponding Arduino signal is triggered which in turn helps the car to move in a particular direction with the help of its controller. Below is the pictorial view of the model for better understanding.

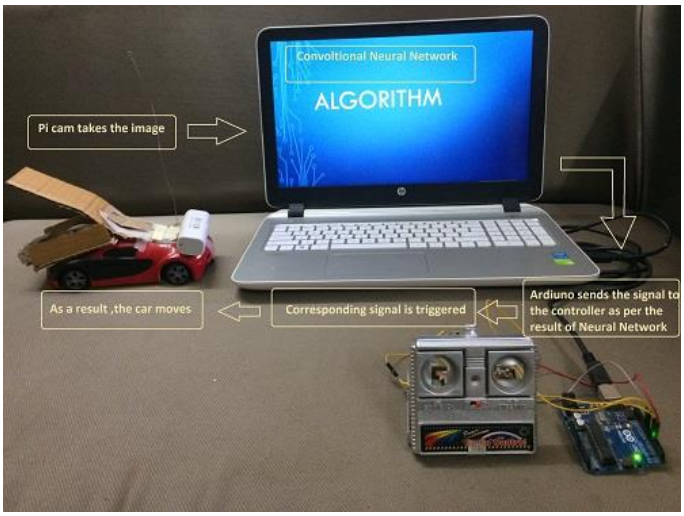


Fig 2. Pictorial view of the proposed model

IV. Hardware and Software Requirnment

The Hardware involved in this project are Raspberry Pi, Pi camera, Arduino microcontroller and of course a toy remote controlled car. Let's discuss each hardware in brief.

A. Raspberry Pi

The Raspberry Pi is a small low cost single board computer having a processor speed ranging from 700 MHz to 1.2 GHz

for the Pi 3.The on-board memory ranges from 256 MB to 1 GB RAM. The boards supports up to 4 USB ports along with HDMI port. Along from all this it has number of GPIO pins which support protocols like I²C.Moreover it also supports Wi-Fi and Bluetooth facility which makes device very compatible with other devices. It supports Scratch and Python programming languages [10]. It supports many operating systems like Ubuntu MATE, Snappy Ubuntu, Pidora, Linutop and many more out of which Raspbian is specifically designed to support Raspberry Pi's hardware [11].

Fig 3. Raspberry Pi

B. Pi Camera

Pi camera is great gadget to capture time-lapse, slow motion with great video clarity. The dimensions of camera are 25mm to 24mm by 9mm, which connects to Raspberry Pi via a flexible elastic cord which supports serial interface. The camera image sensor has a resolution of five megapixels and has a focused lens. The camera provides a great support for security purpose. Various characteristics of the camera are it supports 5MP sensor, Wide image, capable of 2592x1944 stills, 1080p30 video on Camera module v1 [12].

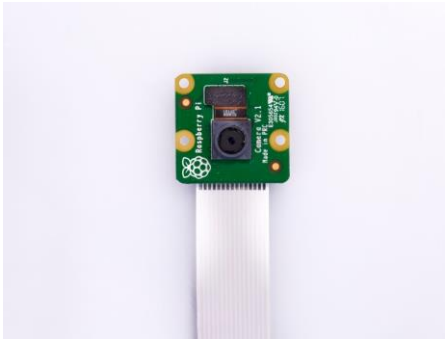


Fig 4. Pi Camera

C. Arduino Microcontroller

This microcontroller is based on ATmega329P.There are 14 digital input/output pins available out of which 6 can be used a PWM outputs. It also supports 6 analog inputs. It has 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It has 32 kb of flash memory and 2 kb of SRAM and weighs around 25g [13].Apart from all these features Arduino IDE is very user friendly and uses basic c as its programming language

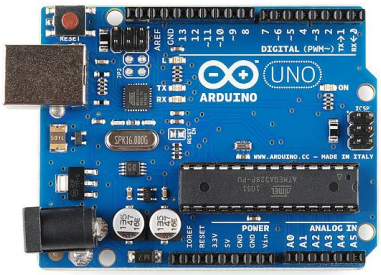
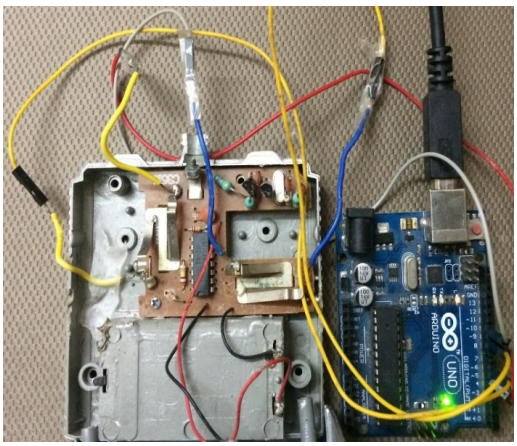


Fig 5. Arduino Uno

After attaching these hardware on the car and connecting Arduino with the controller. The setup looks something like shown in the figure.

(a)



(b)

Fig 6. Overall hardware setup. (a) Raspberry Pi attached with Pi camera mounted over remote controller and (b) Arduino connected with the car's controller.

These were all about the hardware needed to build a working prototype. Along with these hardware the software used in this

project. Are Raspberry Pi cam interface, Arduino IDE, OpenCV and Spyder environment for developing machine learning model .Let's discuss each of these briefly.

D. Arduino IDE

Ardino IDE is the platform were the programs are written for Arduino board.It has compile button which helps in compiling the code along with the upload tab which helps to upload the code on the board.Programs written on Arduino IDE are often called Sketches and are saved as .ino extension.The editor has numerous other features like verify, save, upload , include library and serial monitor.Apart from this ,the developers have made easy to use functions, which makes coding easy and fun.Moreover there are number of examples provided for each and every interface which helps the user learn more about functions and hardware as well.

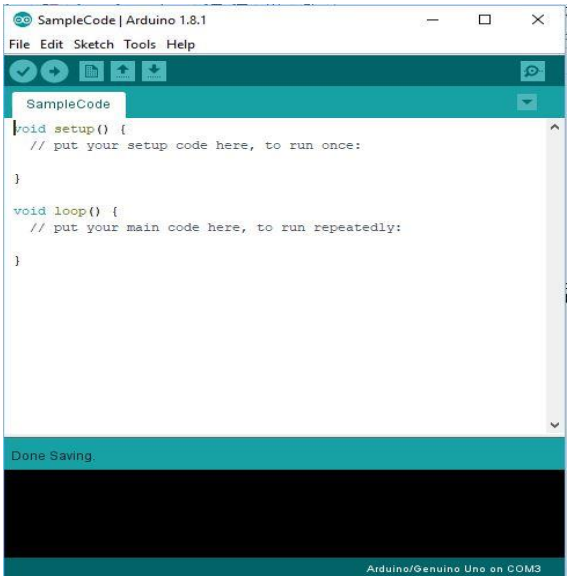


Fig 7. Arduino IDE

E. OpenCV and Spyder environment

OpenCV is an open source computer vision library which is capable of handling images/videos from fairly basic tasks to utter complex tasks like facial recognition. It supports C++, C, Python and Java programming languages and supports Windows, Linux, Mac OS, iOS and Android. Written in optimized C/C++, the library can take advantage of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform [14]. In this project it is serving a major support, it helps to crop out the section of the video from the Raspberry-Pi cam interface as shown above and converts it to the grayscale, resize it and then passes it to the Convolutional Neural Network. Spyder is a powerful interactive development environment for the Python language which has advanced editing, interactive testing, debugging and introspection features and a numerical computing environment. It has a matplotlib as plotting library which helps to plot 2D/3D graphs [15].

F. Raspberry Pi Cam Interface

To remotely capture the live feed from the camera to the laptop we need to develop an interface which would serve this purpose.This is were the software Rpi-cam interface comes into the picture. It's the program which helps you capture the live feed by just letting the ip address of the Raspberry pi.One can record and download video/image in various resolutions with different number of settings. Below is the view of the software under action.

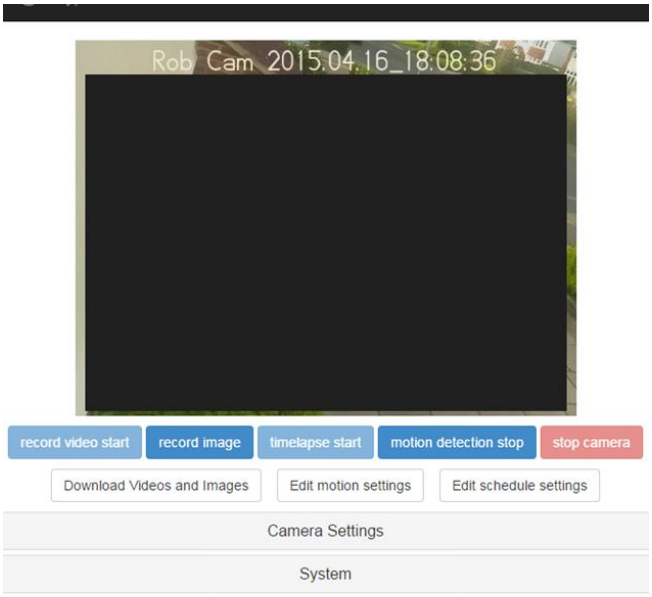
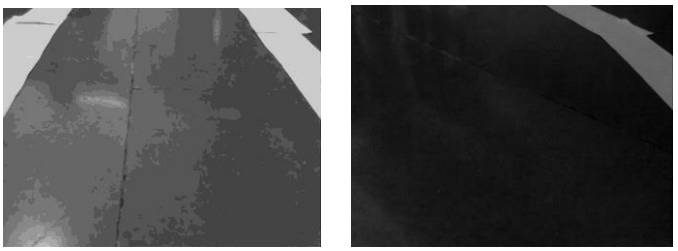


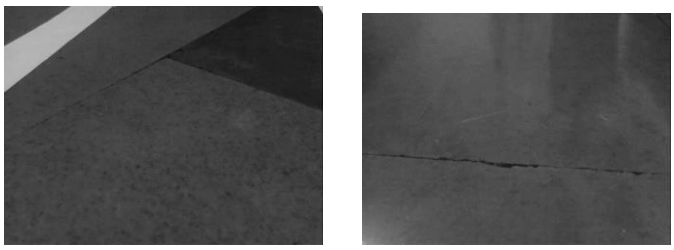
Fig 8. Raspberry-Pi camera interface

V. RESULTS

The car was trained under different combinations of the track i.e. straight, curved, combination of straight and curved and etc. Total of 24 videos were recorded out of which images were extracted.10868 images were extracted and was categorically placed in different folders like left, right, straight and stop. Below is the sample image of each of the scenario in its gray scaled version.



(a) (b)



(c) (d)

Fig 9. Sample images of the track on which the car is trained. (a) shows straight track, (b) shows left turn, (c) shows right turn and (d) shows when to stop

These images were resized to 320 x 240 and on which the network was trained on. The Convolutional Neural Network had 128 input nodes, 2 hidden layers of 32 nodes each and finally the output layer consisting of 4 nodes for each of the 4 output. To avoid overlearning of the network dropout of 0.5 was considered. 'Relu' activation function was used between the input and hidden layers and 'Softmax' activation function was used in the output layer. The Batch size was set to 10 and number of epochs was set to 3 and it took 5-6 hours to train on the GPU mode. This was all about the network configuration that was used to train the model. Let's now see how well the car performed on each track.

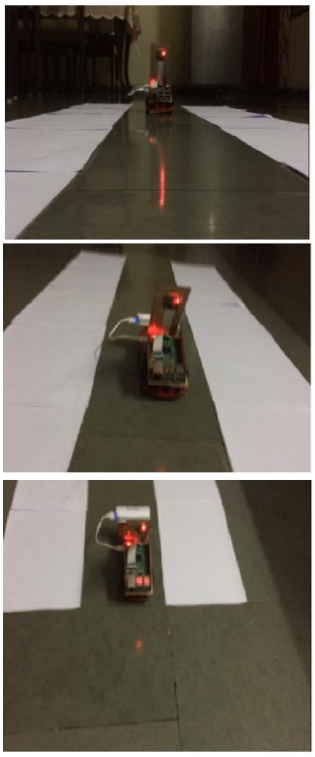


Fig 10. Car moving on straight track (Top to bottom sequence)

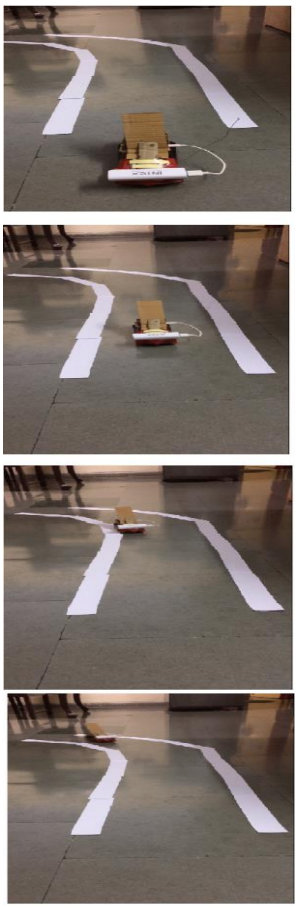
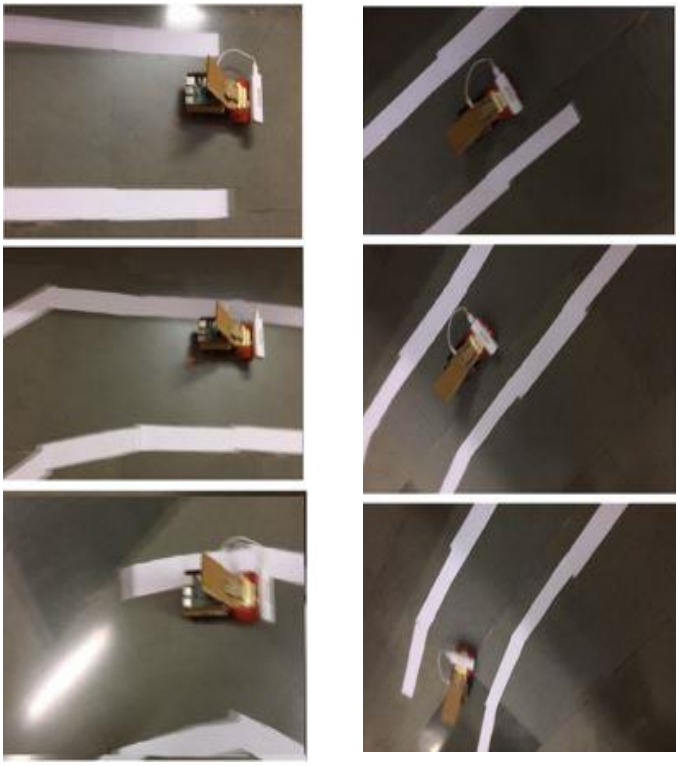
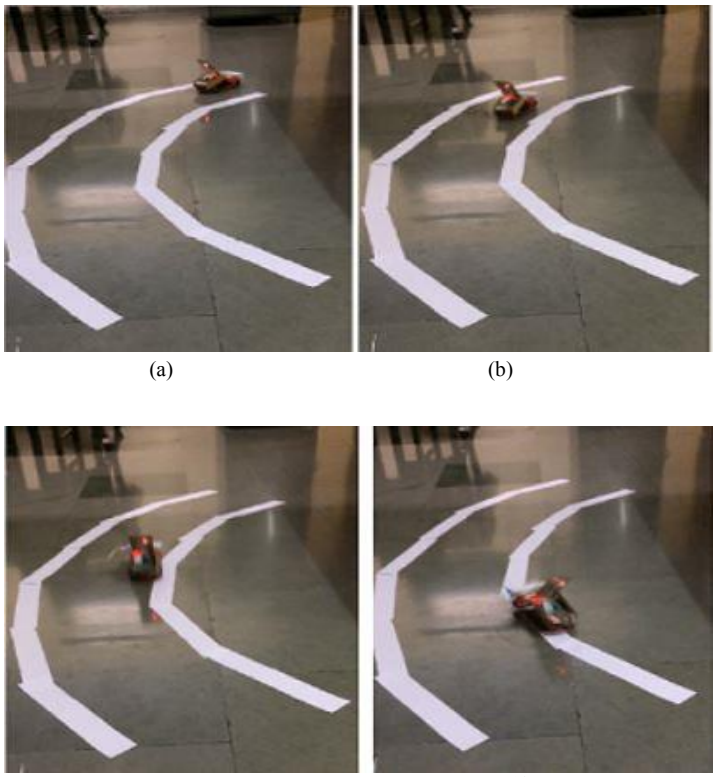


Fig 11. Car moving on straight followed by curved track (Top to bottom sequence)



(a) (b)

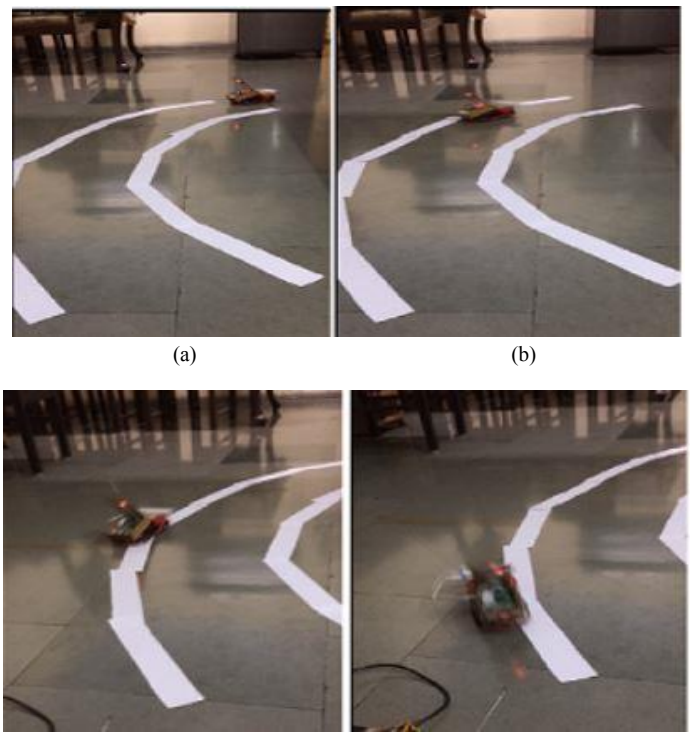
Fig 12.Car moving on straight followed by curved track (a) and (b) (top to bottom sequence)



(a) (b)

(c) (d)

Fig 13.Car moving on curved track (Alphabetical sequence)



(a) (b)

(c) (d)

Fig 14.Car moving on curved track (Alphabetical sequence)

VI. Conclusion and Future Work

A. Conclusion

In this paper, a method to make a model of self-driving car is presented. The different hardware components along with software and neural network configuration are clearly described. With the help of Image Processing and Machine Learning a successful model was developed which worked as per expectation. Thus the model was successfully designed, implemented and tested.

B. Future Work

Scope and reliability of this model can be improved. As seen in Fig. 13th and in Fig 14th, the car slightly moves out of the track which can be a serious issue if it hits nearby objects if we consider a real car. If we could develop an advanced system which would take care of this issue, it would not only make the system reliable but at the same time it would make the overall design attractive and risk-free from accidents.

Acknowledgment

This work wouldn't have been possible without my family's support. They were the source of constant motivation for me. I would also like to thank my mentor, Prof. Shital Thakkar would took her valuable time to guide me throughout the course of this project. Last but not the least I would like to mention my colleague, Chintan Acharya would helped me managing resources and ultimately helping me to achieve my goal.

References

[1] http://asirt.org/initiatives/informing-road-users/road-safety-facts/road-crash-statistics

[2] http://sites.ndtv.com/roadsafety/important-feature-to-you-in-your-car-5/

[3] Wang, Chun-Che, Shih-Shinh Huang, and Li-Chen Fu. "Driver assistance system for lane detection and vehicle recognition with night vision." *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*. IEEE, 2005.

[4] Miao, Xiaodong, Shunming Li, and Huan Shen. "ON-BOARD LANE DETECTION SYSTEM FOR INTELLIGENT VEHICLE BASED ON MONOCULAR VISION." *International Journal on Smart Sensing & Intelligent Systems* 5.4 (2012).

[5] Saha, Anik, et al. "Automated road lane detection for intelligent vehicles." *Global Journal of Computer Science and Technology* (2012).

[6] Pannu, Gurjashan Singh, Mohammad Dawud Ansari, and Pritha Gupta. "Design and implementation of autonomous car using Raspberry Pi." *International Journal of Computer Applications* 113.9 (2015)

[7] Mohanapriya, R., Hema, L. K., Yadav, D., & Verma, V. K. (2014). Driverless Intelligent Vehicle for Future Public Transport Based On GPS. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, *3*.

[8] Ms. D.D Jadhav.,Komal Jadhav,Kajal Shinde,Anjali Sonawane, , , , ."Autonomous Vehicle with Obstacle Avoidance and Tracking", Volume 4, Issue VIII, International Journal for Research in Applied Science and Engineering Technology (IJRASET) Page No: , ISSN : 2321-9653

[9] Dhanasingaraja, R., S. Kalaimagal, and G. Muralidharan. "Autonomous Vehicle Navigation and Mapping System." *Division of Mechatronics, 2014 International Conference on Innovations in Engineering and Technology*. Vol. 3. No. 3. 2014.

[10] https://en.wikipedia.org/wiki/Raspberry_Pi

[11] https://www.hongkiat.com/blog/pi-operating-systems/

[12] https://www.raspberrypi.org/documentation/hardware/camera/

[13] https://www.arduino.cc/en/Reference/Board?from=Guide.Board

[14] https://opencv.org/

[15] https://pythonhosted.org/spyder/