iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** II **Month of publication:** February 2022

DOI: https://doi.org/10.22214/ijraset.2022.40505

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue II Feb 2022- Available at www.ijraset.com

Self-Driving Car Using Machine Learning and Neural Network

Haroon Ijaj Hublikar[1], Atharva Ajit Keny[2], Rahul Dilip Ghonge[3], Khan Sawood Iftekhar Ahmed[4]
[1, 2, 3, 4]*Department of Electronics Engineering, University of Mumbai*

Abstract: ***Self-driving cars are autonomous vehicles that can run on their own without human intervention and could represent a technological revolution in the next decade. This work presents the development of a low-cost prototype of a miniature self-driving car model using simple and out-of-the-box technology. The idea is to use only camera modules to create self-driving cars that move between destinations with minimal human intervention. The purpose of this project is to accelerate the process of driving a car through automation. The results of this project will definitely reduce the number of car accidents that occur today. This project uses NVIDIA Jetson Nano as the main controller and has a processing power of about 500 gigaflops. The controller is mounted on the car platform itself and performs all the calculations needed to drive the car on board. The car uses a trained convolutional neural network (CNN) that predicts all the parameters that the car needs to drive smoothly. These are directly connected to the main steering mechanism and the output of the deep learning model determines the steering angle of the vehicle.***

Keywords: CNN, NVIDIA Jetson Nano, Self-driving, automation, deep learning

I. INTRODUCTION

The topic of this project is building an autonomous vehicle from scratch, and more specifically, a self-driving RC car. The goal of the project is to build a model capable of autonomous driving on the track, while demonstrating the capability to perform behaviours such as lane changing. The project will go through the entire process of building such a vehicle, starting from the very RC car model and the embedded hardware platform, to the end-to-end machine learning pipeline necessary for automated data acquisition, labelling and model training. The main motivation behind the selected topic is the fast-moving progress of applied artificial intelligence (AI) and the predicted importance of autonomous vehicles on the future of humanity, from independent mobility for non-drivers and low-income individuals, reduced pollution, traffic and parking congestion to increased safety on the roads. Autonomous vehicles are also predicted to be relied on in some of the most complex human planned endeavours, such as space exploration. The meteoric rise of AI along with deep learning (DL) methods and frameworks, have made possible the creation of such an autonomous vehicle without expensive laboratories and years of research. Currently, there are a number of private companies as well as academic groups working on autonomous vehicles and their integration into existing regulations, laws and society itself.

The advantages and quality of life improvements autonomous vehicles offer range from safer and less congested roads, reduced parking and fewer vehicles per capita to up to several thousands of dollars saved per year in travel time reduction, fuel efficiency, parking benefits and crash costs. It's obvious, with everything stated above, as well as with the well-known rise of artificial intelligence, that the field of autonomous vehicles is at its very beginnings and that it will have an important long-term impact on society, through both financial and ethical aspects. The accessibility of such technology should be made more broadly available to researches and students if the field is to continue progressing through increased discussions on important topics and not stagnate in a winter of autonomous vehicles, which is one of the reasons behind the topic of this project.

II. AIM

To create a prototype car which will be able to lap around a track, after it has been trained using supervised learning.

III.OBEJECTIVE

A. The goal of Self driving vehicle is to make a completely practical mechanized vehicle that can decrease human exertion
B. Lessen the mishap rate, give better fuel utilization and better traffic stream.
C. Self-driving vehicles are made for giving advantages to the general public, for example, giving transportation to those individuals who can't drive due to being old or physical impairment.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue II Feb 2022- Available at www.ijraset.com

IV. LITERATURE SURVEY

A. Learning a Driving Simulator, Eder Santana, George Hotz, University of Florida.

Comma.ai's approach to Artificial Intelligence for self-driving car is based on an agent that learns to clone driver behaviours and plans manoeuvres by simulating future events in the road. This paper illustrates one of our research approaches for driving simulation. One where we learn to simulate. Here we investigate variational autoencoders with classical and learned cost functions using generative adversarial networks for embedding road frames.

B. End to End Learning for Self-Driving Cars, Mariusz Bojarski, Davide Del Testa, NVIDIA Co-operations.

They trained a convolutional neural network (CNN) to map raw pixels from a single front-facing camera directly to steering commands. This end-to-end approach proved surprisingly powerful. With minimum training data from humans the system learns to drive in traffic on local roads with or without lane markings and on highways. It also operates in areas with unclear visual guidance such as in parking lots and on unpaved roads.

C. 3D Visual Perception for Self-Driving Cars using a Multi-Camera System, Christian Häne, Lionel Heng, University of Trier, Germany.

Cameras are a crucial exteroceptive sensor for self-driving cars as they are low-cost and small, provide appearance information about the environment, and work in various weather conditions. They can be used for multiple purposes such as visual navigation and obstacle detection. We can use a surround multi-camera system to cover the full 360-degree field-of-view around the car. In this way, we avoid blind spots which can otherwise lead to accidents. To minimize the number of cameras needed for surround perception, we utilize fisheye cameras. Consequently, standard vision pipelines for 3D mapping, visual localization, obstacle detection, etc.

V. HARDWARE DESIGN

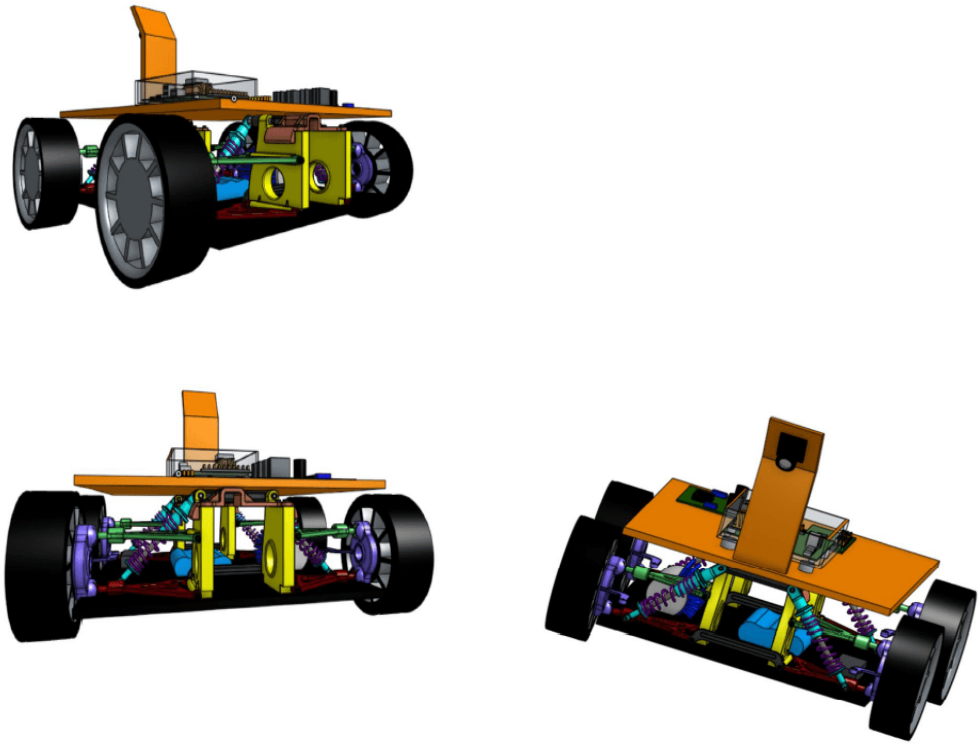


Figure 1. 3D-Model of the prototype car, Front, Back, Side view.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

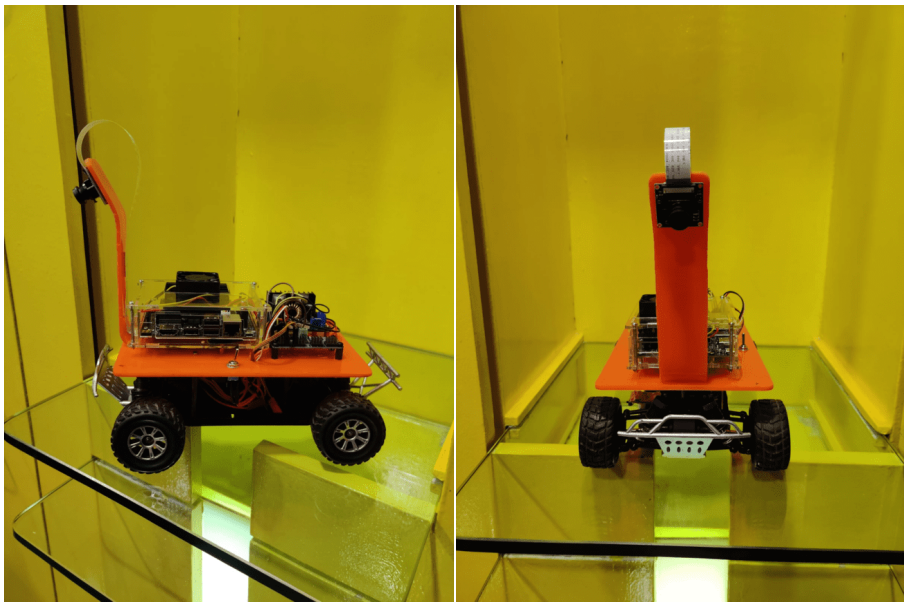


Figure 2. Prototype Model of The Self-Driving Car

Since our project requires constant parameter and code changes, we are running a virtual machine on our onboard computer (jetson nano), which is mounted on the car platform itself. When the main power switch is turned on, it powers the onboard computer (jetson nano). Once the Jetson nano is turned ON it loads the OS from the onboard micro-SD card. It initializes the Virtual machine environment, which can be accessed by the IP address, http://192.168.1.103:8888 and filling of appropriate password set by the user.

After successful login we can now run our program from the jupyter notebook. We then run the code block by block, to test its basic motion. After this, the car is ready to be placed on the track/road. Once the car is placed on the track, the process of supervised learning starts, the user is supposed to keep the car at various places and take images of the track while running the program that is responsible for capturing these images. These images will then be used to create a dataset, this dataset will be called when the car attempts to go around the track autonomously. The user can change the throttle gain, steering bias, steering gain live on the go.

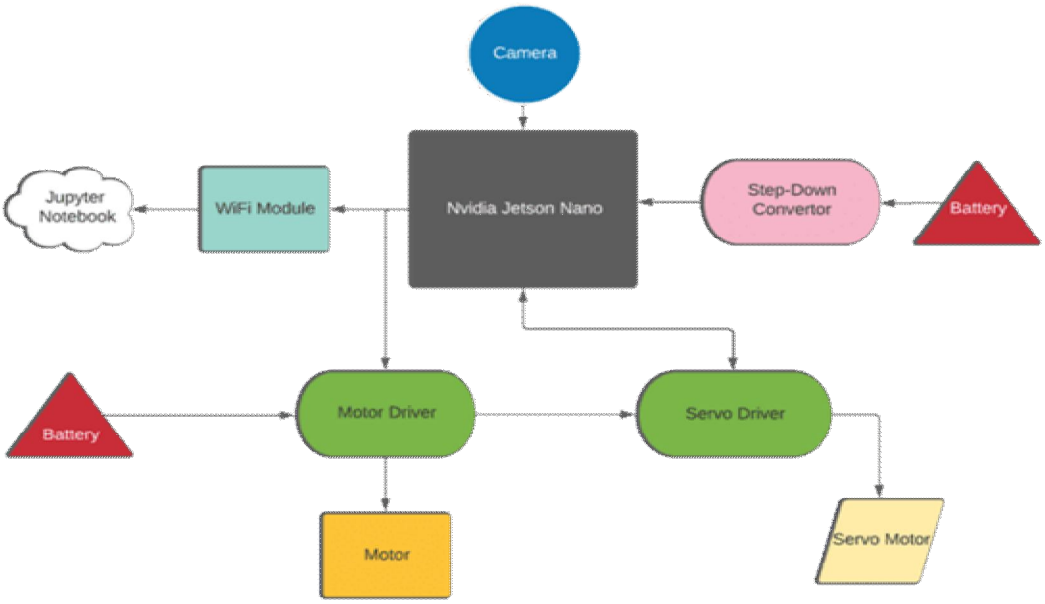


Figure 3. Block Diagram

VI. TECHNICAL SPECIFICATIONS

A. Micro-Controller (NVIDIA Jetson Nano)

GPU: NVIDIA Maxwell architecture with 128 NVIDIA CUDA® cores
CPU: Quad-core ARM Cortex-A57 MP Core processor
Memory: 4 GB 64-bit LPDDR4, 1600MHz 25.6 GB/s
Storage: 16 GB eMMC 5.1
Video Encode: 250MP/sec, 1x 4K @ 30 (HEVC), 2x 1080p @ 60 (HEVC),4x 1080p @ 30 (HEVC), 4x 720p @60 (HEVC), 9x 720p @ 30 (HEVC)
Video Decode: 500MP/sec, 1x 4K @ 60 (HEVC), 2x 4K @ 30 (HEVC), 4x 1080p @ 60 (HEVC), 8x 1080p @ 30 (HEVC), 9x 720p @ 60 (HEVC)
Camera: 12 lanes (3x4 or 4x2) MIPI CSI-2 D-PHY 1.1 (1.5 Gb/s per pair)
Connectivity: Gigabit Ethernet, M.2 Key E
Display: HDMI 2.0 and eDP 1.4
USB: 4x USB 3.0, USB 2.0 Micro-B
Others: GPIO, I2C, I2S, SPI, UART
Mechanical: 69.6 mm x 45 mm
260-pin edge connector

B. Camera

A robust 8MP IMX219 Camera for Nvidia Jetson Nano.
It's capable of 3280 x 2464-pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.
It is attached to Jetson by the dedicated standard CSi interface.
Has a FOV of 170°

C. Servo Controller

This board/chip uses I2C 7-bit address between 0x60-0x80, selectable with jumpers
Terminal block for power input (or you can use the 0.1" breakouts on the side)
Reverse polarity protection on the terminal block input
3 pin connectors in groups of 4 so you can plug in 16 servos at once (Servo plugs are slightly wider than 0.1" so you can only stack 4 next to each other on 0.1" header
"Chain-able" design
A spot to place a big capacitor on the V+ line (in case you need it)
220-ohm series resistors on all the output lines to protect them, and to make driving LEDs trivial
i2c-controlled PWM driver with a built-in clock. Unlike the TLC5940 family, you do not need to continuously send it a signal tying up your microcontroller, it's completely free running.
6 address select pins so you can wire up to 62 of these on a single i2c bus, a total of 992 outputs - that's a lot of servos or LEDs
Adjustable frequency PWM up to about 1.6 KHz
12-bit resolution for each output - for servos, that means about 4us resolution at 60Hz update rate

D. Step-Down Power Module

Adjustment method: first correct input power (between 4-40V) then multi-meter to monitor the output voltage and adjust potentiometer (usually clockwise turn boost, buck turn counter clockwise),
Input: IN+ input positive IN- input negative
Output: OUT+ output positive OUT- output negative
Static power consumption is only about 6mA.
Connection: Welding, plus pin can be directly soldered after the PCB.
Short circuit protection: current limiting, thermal protection, self-recovery.
Non-isolated step-down (BUCK)switching regulator.
Short circuit protection: Current limiting, self-recovery
Potentiometer adjustment direction is as Clockwise (increase) and Anti-clockwise (decrease)
Non-synchronous rectification

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

E. Electronic Speed Controller

Continuous Current :30A
Battery Input: 2S Lipo
Can be used with 130/180/260/380 brushed motor
Three Functions: Forward/Reverse/Breaking

F. LI-ION Battery

Nominal voltage: 7.4v
Capacity: 2500mah
Max discharge rate: 2c
Discharge current up to 5A
Dimensions: 65mm x 36mm x 18mm
Weight: 120g

G. Motor

Voltage: 7.2-8.4
Outer diameter: 27.5mm
Length: 46.5mm
Shaft length: 14.6mm
Shaft diameter: 2.3mm
Speed: 21000 rpm
Motor gear: 17T

H. Servo Motor

Voltage: 4.8-6.0 V
Torque: 3.0 / 3.5kgf.cm
Rate: 0.15 / 0.13 s / 60 degrees
Dimensions: 28.5 * 13.0 * 27.0 mm
Gears: Plastic Gears
Weight: 17.5 g

VII. SOFTWARE DESIGN

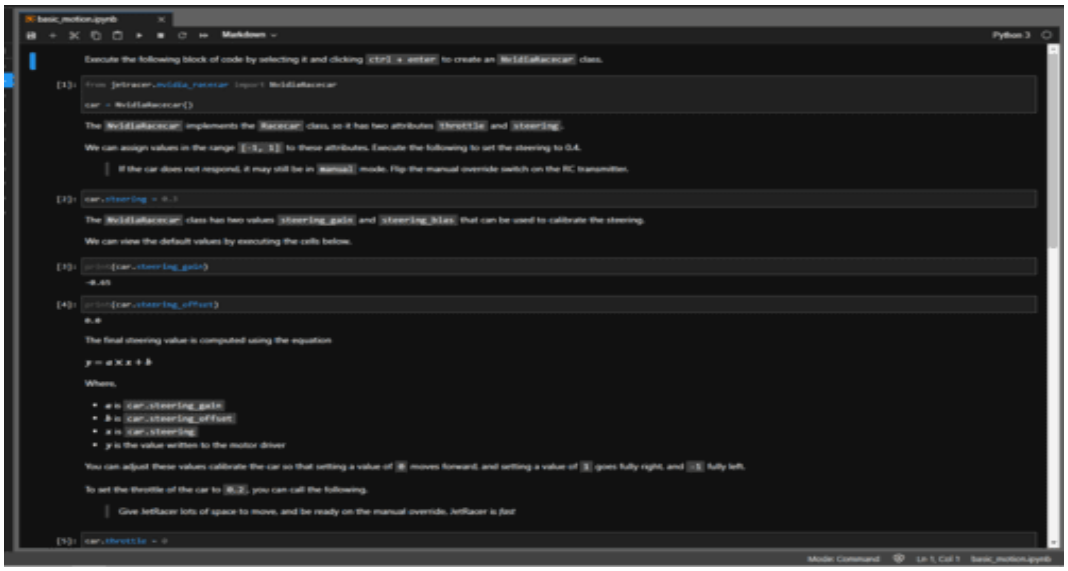


Figure 4. Basic Motion

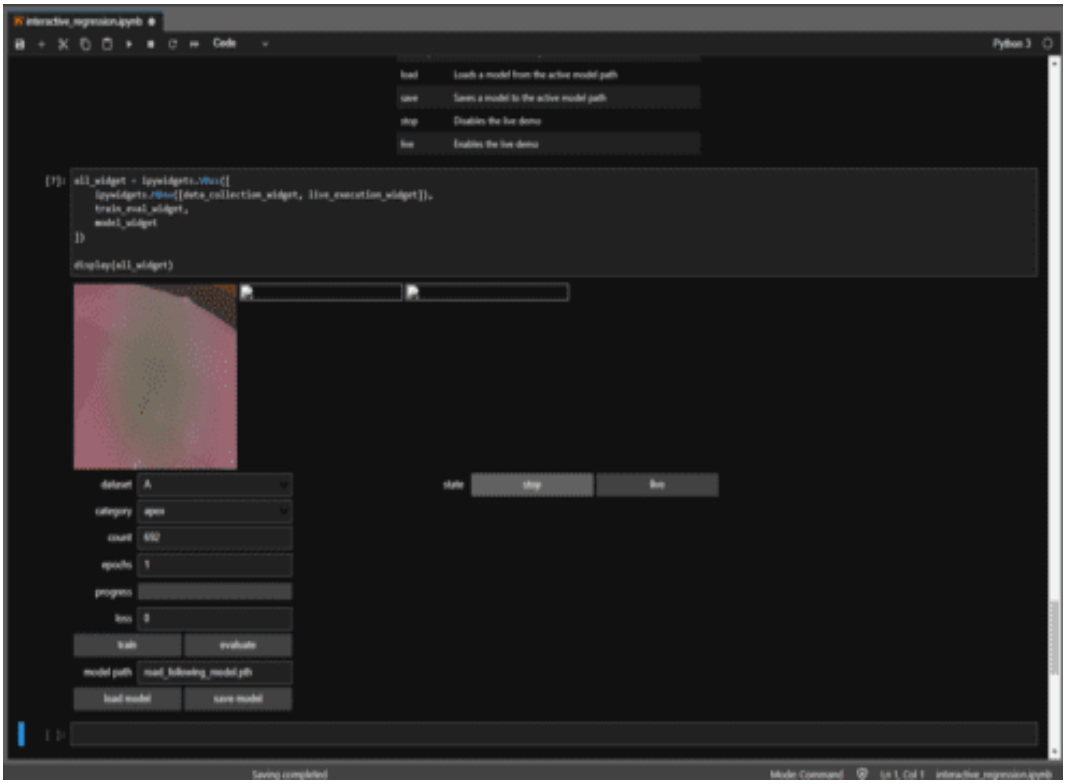


Figure 5. Interactive Regression

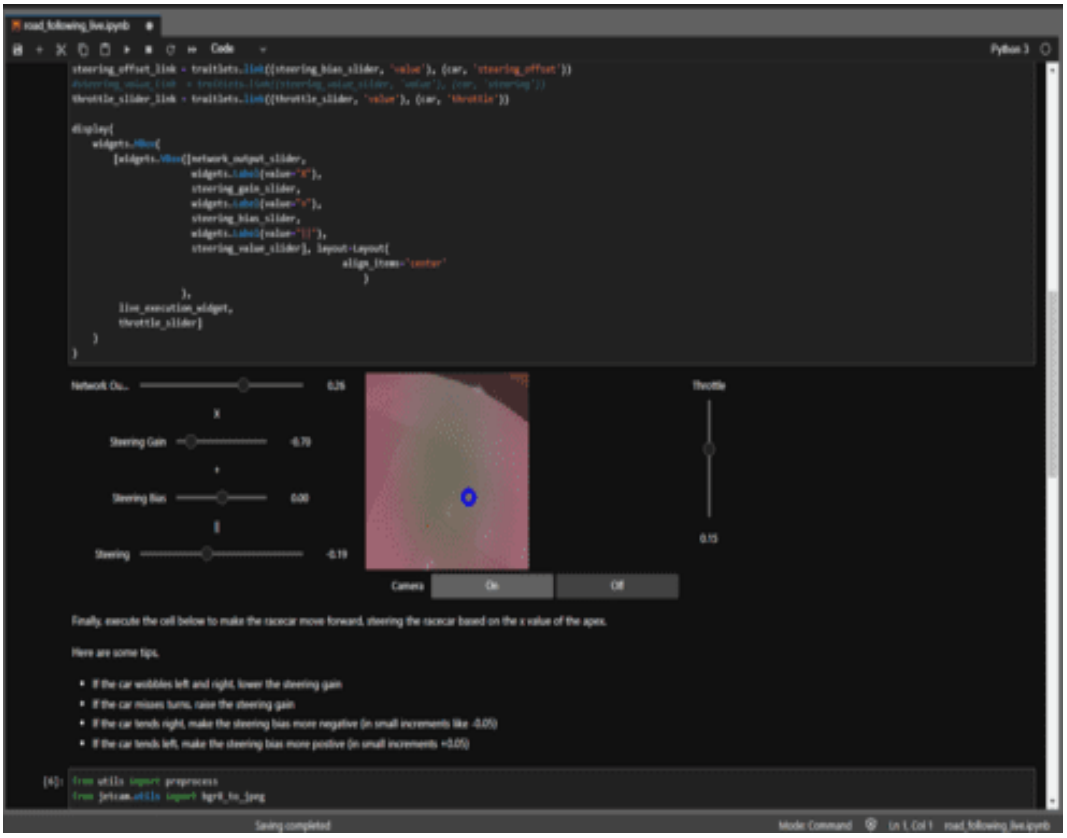


Figure 6. Road Following

The software implementation will be done in the next semester, some of the explanation here is theoretical.

1) The bot is accessed through Jupyter Notebook, on a Wi-Fi connection. You can do so using a browser on a computer connected to the same Wi-Fi network.
2) The bot runs Ubuntu (Linux) OS.
3) All the programming for the algorithm and the working of the bot will be done in Python script.
4) The program will have access to the camera, the motor driver and the servo motor using the library already made available by NVidia.
5) All the photos taken from the camera for a specific track will be saved in a folder as its own profile which the bot can revisit whenever it encounters the same track again.
6) The user will be able to see the live feed from the camera using a plugin.
7) The user will also be able to change certain parameters using the GUI of the Jupyter Notebook that gives us access to the Python Script, which is currently running as well as any errors that might occur.

VIII. RESULTS DISCUSSION

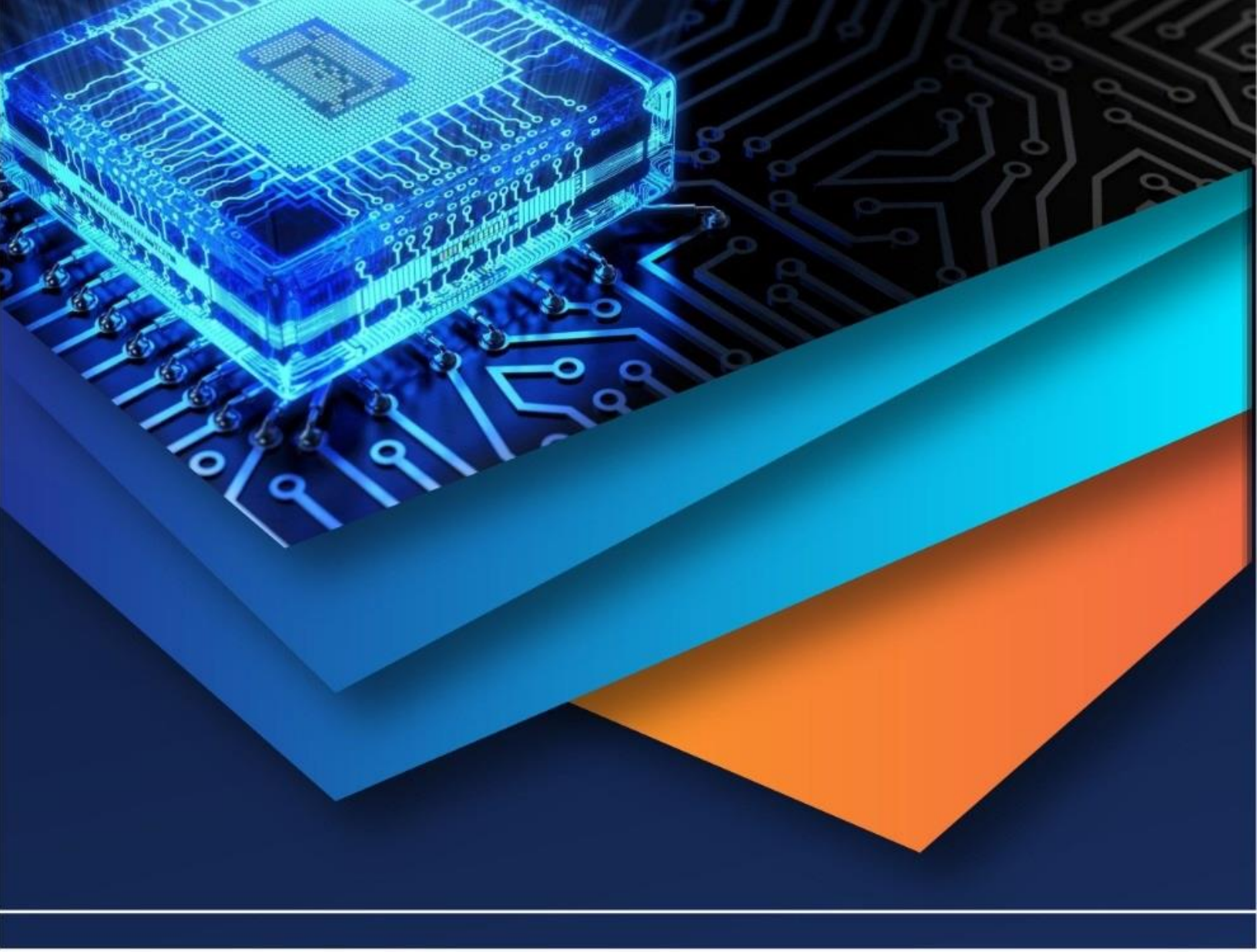
We were successfully able to build and implement a fully autonomous machine learning based vehicle model. It took several months of trial and error to get the car follow a pre-built test track. The project has proven to be a great way through which machine learning can be studied and applied through a hands-on approach. It can seem a bit overwhelming for unassuming beginners, but when taken in bite sized steps, it is more than manageable while being a lot of fun. It also proves that working on autonomous vehicles, albeit smaller scale models, does not need to involve a huge financial investment or obligation, the necessary equipment can be obtained for a couple hundred euros or dollars and with a savvy building mindset can be made into a great platform which actually performs well in the real world.

IX. CONCLUSION

With regards to generalizability, the hardware and software setup should be fully replicable, while the exact testing environment may not. In this sense, it is a major advantage that the project uses freely available software and cheaper hardware, improving access to similar setups. The vision system is vulnerable to noise, and various room layouts, colors, lighting conditions and times of the day seem to have a significant impact on its performance. The project resulted in a fully functional custom hardware platform with an end-to-end machine learning pipeline that allows automated data collection and labelling, with complex models such as the second iteration of the proposed architecture implementation which allows the car to autonomously navigate randomly generated, never before seen tracks and even perform behaviors. The rapid prototyping pipeline has also proved functional, with the trained models even being interchangeable between the simulated vehicle and the real one, which could be used as a valuable tool for pre-training real vehicles to mitigate the otherwise avoidable errors at the beginning stages of an autonomous vehicles training

REFERENCES

[1] Learning a Driving Simulator, Eder Santana, George Hotz, University of Florida.
[2] End to End Learning for Self-Driving Cars, Mariusz Bojarski, Davide Del Testa, NVIDIA Co-operations.
[3] 3D Visual Perception for Self-Driving Cars using a Multi-Camera System, Christian Häne, Lionel Heng, University of Trier, Germany.
[4] Artificial Intelligence: A Modern Approach Stuart J. Russel and Peter Norvig
[5] Deep Learning Ian Goodfellow, Yoshoua Bengio, and Aaron Courville
[6] Applied Predictive Modeling Max Kuhn and Kjell Johnson



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)