

ABSTRACT:

The Autonomous Car is an open source DIY self-driving platform for small scale cars based on a raspberry pi computer, a camera, and servo shield board to interface with the R/C car. We drive the vehicle around a lined car to capture images and steering angles, which trains a neural-network autopilot to drive itself around the track. Top speeds around corners are about 4-6mph.While collecting training data, the car itself doesn’t do all that much. It basically takes pictures and sends them to an amazon server and gets servo commands in return. The server is where magic happens.

First collects the images and driving information from the user manually driving the car around the track. The default way to do this on the donkey is via the mobile friendly web page delivered by the server. The mobile web page even has a live video view of what the car sees and a virtual joystick. The server records the data from a person driving the car, then uses those images and joystick positions to train a TensorFlow neural network model in software. This happens quickly-a full trip latency takes about 1/10 second once trained, model can be loaded on the car .This uses the supervised learning technique often referred to as behavioural cloning like HVI/F library and CAN library along with HV controller-[which includes velocity control, shift control, communication, mode management, angle control, steer torque control, exception, pedal control, DA/AD].

Next we will be training the Donkey car to drive on its own based on our driving style …….

INTRODUCTION:

Deep learning is an artificial intelligence function that imitates the workings of the human brain in processing data and creating patterns for use in decision making. As such, IOT is the extension of internet connectivity into physical devices and everyday objects. The internet of things is a system of interrelated computer devices, mechanical and digital machines, objects or people that are provided with unique identifiers (UID’s) and the ability to transfer data over a network without requiring human-human or human-computer interaction.

The combination of these technologies are used for building devices for consumer use, including connected vehicles, home automation, wearable technology, connected health and gadgets with remote monitoring appliances.

DEEP LEARNING KAR:

The autonomous car is one of the applications of Deep Learning & IOT (internet of things). Donkey is a high level self-driving library written in Python. It was developed with a focus on enabling fast experimentation and easy contribution.

The purpose of this car is to develop a cleaning robot by using Tensor flow SSD MobileNetV2 DNN on the Raspberry Pi +Pi camera to build an autonomous car capable of object detection through image capturing.

APPLICATIONS OF AUTONOMOUS KAR:

* Forset surveillance
* Indoor mapping
* Autonomous navigator

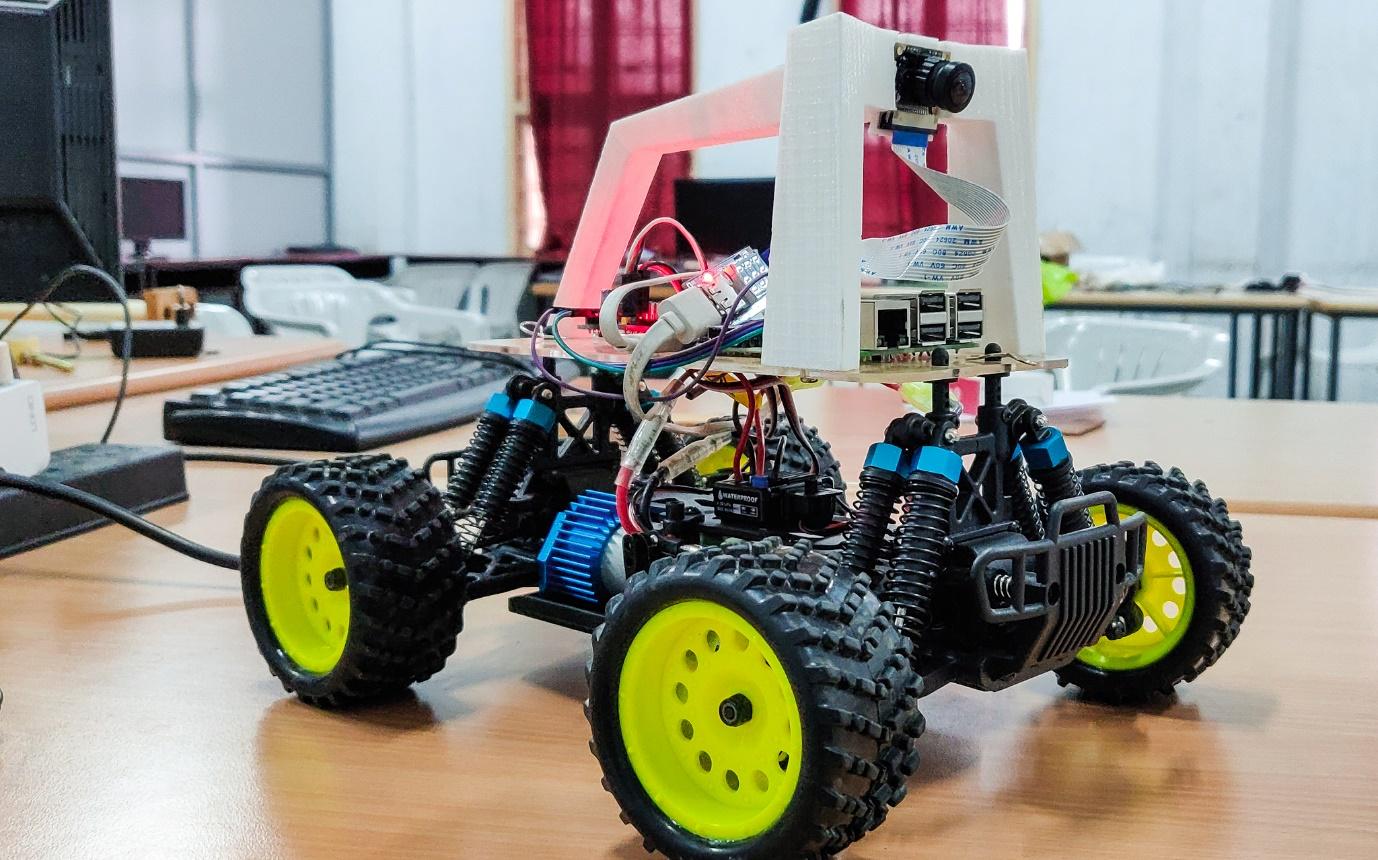
HARDWARE ASSEMBLY:

* We made use of “[1/16 2.4Ghz Exceed RC Blaze EP Electric RTR off Road Buggy (Fire Yellow)](https://docs.google.com/document/d/1iy9AoZpfV_qv_vF3wz4aaOt9yX-3GifyYcQPx6Xzvhg/edit)**”** fully supported[chassis](https://docs.google.com/document/d/19jGLcztW1KxmLwCOuTAcPgTDBrFuErqnGSJ7qjPjrfU/edit)in our project**.**
* However, the battery capacity of this buggy is around 45 minutes.
* Fix the M2.5 Hexagon [Copper Standoff spacer](https://docs.google.com/document/d/1Yq9OIHO3mKKLrdbnVxarYjCK-MidJ1Jyo_LVoKCcJa0/edit) to the base plate. Now, Connect the Raspberry Pi, Servo Motor controller and voltage converter on to the board through the hexagon copper standoff spacer on the base plate.
* USB and Ethernet ports are faced forward as it gives access to the SD card. Attach the Roll cage on top of the base plate by making use of 3 M3\*12mm Pentalobular screws from the bottom of the base plate.
* [Connecting pi to PCA9685](https://docs.google.com/document/d/1roQ4lpjl9ZeW3MXxeJfpRvDUSIkOzbmvAYE3MRvHPWY/edit): Usually, the PCA9685we’re talking about has the capacity to run 64 servo motors at a time. The power for the servo typically comes from the motor [ESC](https://docs.google.com/document/d/14PAN1X1qHeUyWChPrtErLqBXJOaHqYal_6epYB3OT_c/edit), which has a BEC (Battery Eliminator Circuit) built in.
* Connect the female end of the [mini Tamiya connector](https://docs.google.com/document/d/1yjstI7om5rXp6Oftu3Kt_hr8_uEL1NUx8t-EgdW1wJg/edit) to the Voltage converter. Bind the female mini Tamiya connector to the wire from the ESC of the car. The male Tamiya connector will be connected to the battery.
* While, Connect the Voltage converter and the Raspberry Pi using a USB cable.

ATTACHING THE CAMERA:

* [Here’s how to attach the camera...](https://docs.google.com/document/d/1p_eCAzEs_oo1CduAYDwukwV4jhMBwWSC0ohfPE4Md9w/edit)
* Later, Insert the already Flashed SD card in the Raspberry Pi and bench-test the Electronics.
* Now, clamp the Roll Cage on the bottom plate to the car by using pins.
* Throttle cable is attached to Channel 0.
* Steering is attached to channel 1.

Finally, done with the hardware!!



SOFTWARE INSTALLATIONS:

OVERVIEW:

Autonomous car, software components are needed to be installed on the robotic platform which are either Raspberry Pi or Jetson Nano which have setup docs. After installing, the Deep Learning car application will be created from the template. Next the car is trained to drive on its own. This uses a supervised learning technique often referred to as behavioural cloning.

INSTALLING SOFTWARE ON HOST PC:

When controlling autonomous car via behavioural cloning, you will need to setup a host PC to train the machine learned model, from the data collected on the robot.

* For installing autonomous car on windows:

1. Install anaconda python 3.7 64 bit.
2. Open the Anaconda prompt window via Start Menu | Anaconda 64bit | Anaconda Prompt.
3. Type git. If the command is not found, then install git 64 bit.
4. [Code](https://docs.google.com/document/d/1CcdpioH08j3Tz2rOMdhHeFbDs9-TPBUL9UMTZ60jPvk/edit).

INSTALLING SOFTWARE ON RASPBERRY PI:

1. Flashing of operating system:

---> Using a card reader connect SD card to your computer. Before flashing, format your SD card. A software named BALENA ETCHER is helpful in flashing OS. It should be first downloaded in our system.

---><https://www.raspberrypi.org/downloads>.

\*\*By using the above link to download Raspbian Buster with desktop and recommended software. Press on download zip. Open BALENA ETCHER and select the Raspbian image from downloads and flash it. Create a file named SSH to store the installation of software. Now SD card can be ejected first and then remove it. Insert the SD card in raspberry pi and connect it to your computer.

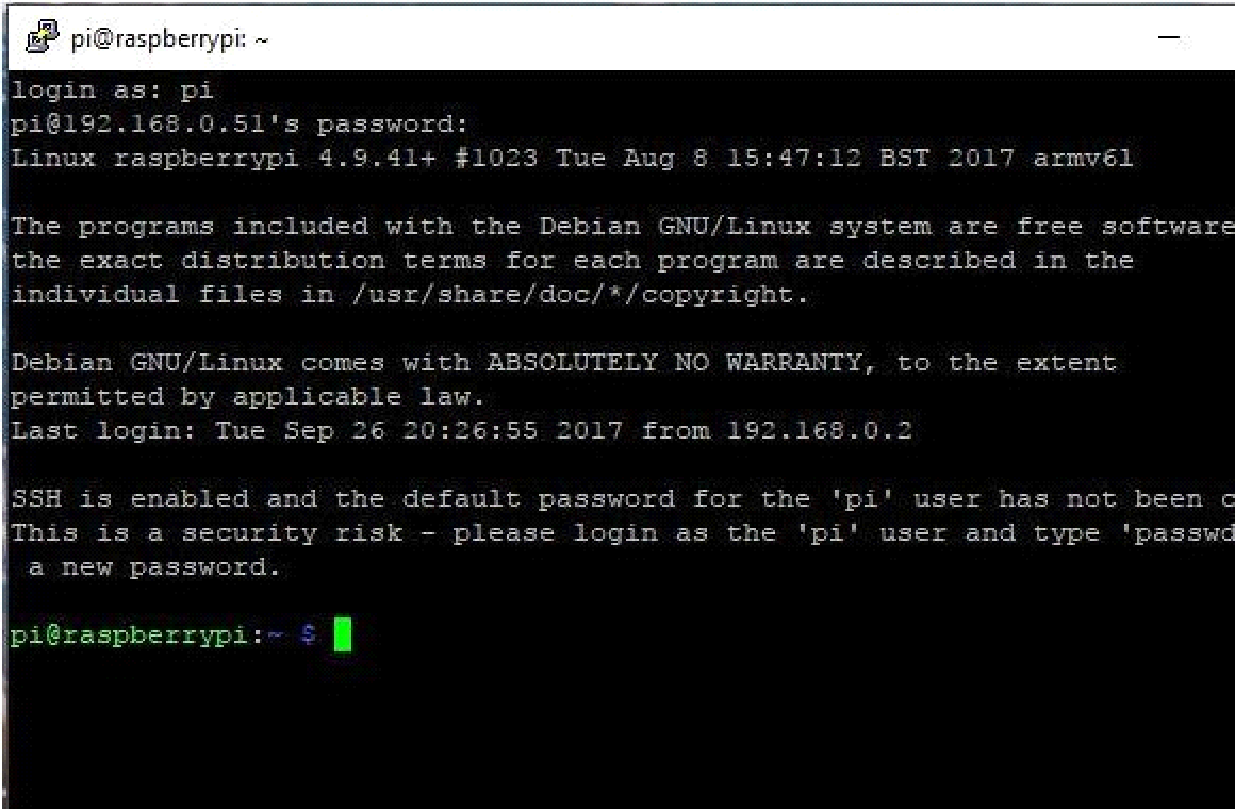


2. Connecting to the pi:

---> User name: pi

---> Password: raspberry

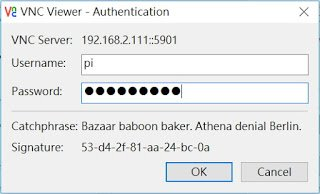
\*\*Connecting to the pi by using username and password via putty software:



3. The IP address of pi can be known by using the following command:

---> sudo if config

Connect to a particular Wi-Fi network can be performed by using the software called VNC viewer. Login with IP address of the raspberry pi.



4. Further manipulations of the hostname and password are performed by the below command.

sudo raspi-config

In fact, this command is also used to enable interfacing options with I2C and CAMERA.

5. Installations:

* [Dependencies](https://docs.google.com/document/d/1FktflNoUUY5Gti4XFiU0jWQCI7YZCM6JENzg84r67t0/edit)
* [Optional CV Dependencies](https://docs.google.com/document/d/1DYNK9rCc1u_tTu4fCtfKc4gcAr7v5YGASW68vgfpdZ8/edit)
* [Setup virtual environment](https://docs.google.com/document/d/1T3VV1G4a6vRKCMYdnqlveNt5pkUnMQPn_JB-2JmUdfA/edit)
* [Donkey Car Python code](https://docs.google.com/document/d/1rV8l_GZkXBwJQ5rLZjJ01g3j-MSZphnhGAbPstIxRSs/edit)
* [Optional open CV.](https://docs.google.com/document/d/1fzsxJkiNb0CjxRsCADNFRvWT9AuVOklAqb7GS4tMq0M/edit)

CALIBRATING THE CAR:

Calibration is the process of configuring an instrument to provide a result for a sample within an acceptable range.

ADJUSTMENTS OF CAR SETTINGS:

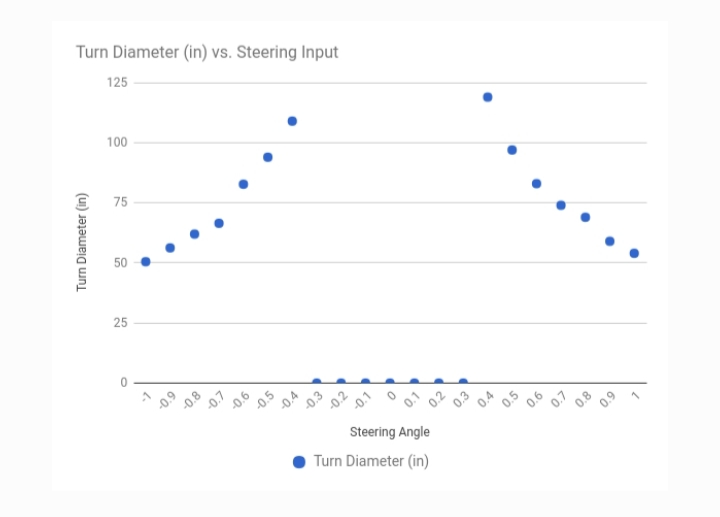
Config.py and myconfig.py are the scripts generated when we ran the donkey created car –path ~/mycar command. All of the car settings are in the above two scripts.

* [Steering calibration](https://docs.google.com/document/d/1VenbrSLN1_9ed5NUiyZNIo1blVwrwLoBLVJ8LC4qoSY/edit)
* [Throttle Calibration](https://docs.google.com/document/d/1KEEv8OFsNGJ1X400ZPHsU2OngI-Mkvci4q_gc_Z9eVs/edit)

Now, open your myconfig.py script and enter the PWM values for your car into the throttle \_controller part.

* THROTTLE\_FORWARD\_PWM=PWM value for full throttle forward
* THROTTLE\_STOPPED\_PWM=PWM value for zero throttle
* THROTTLE\_REVERSE\_PWM=PWM value at full reverse throttle

\*\* [Fine tuning your calibration](https://docs.google.com/document/d/1cxH2roRY-tECECmxY2byhvpVM1rgI0DOwwSbAYrjVSA/edit)



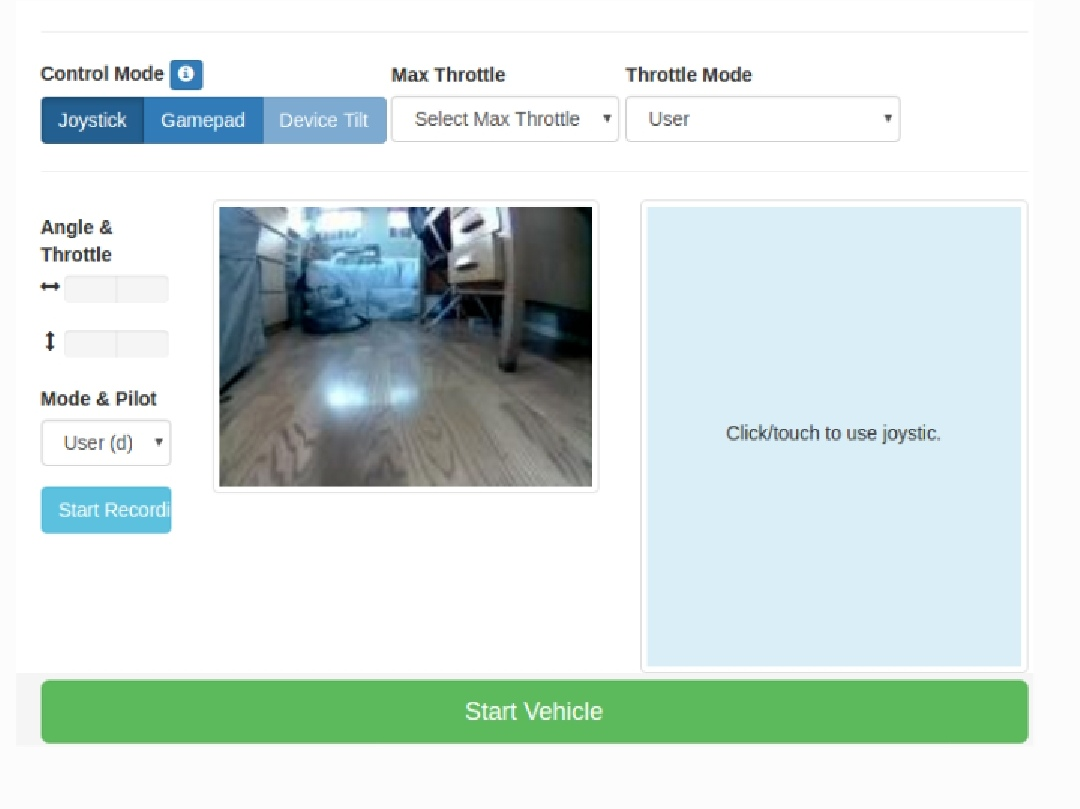
START DRIVING:

1. The further step after calibration is to start driving.  
   Open the car's folder…

cd ~/mycar  
 python manage.py drive

2. From the IP address of a car a page in the web browser is opened at URL :

<IP’s address>:8887  
  
 3. There are 3 ways to run the car:  
 ---> Gamepad.  
 ---> Device tilt.  
 ---> Joystick.



4. Gamepad: By the use of the key controls   
 space : stop car and stop recording  
 r : toggle recording  
 i : increase throttle Driving with the web controller  
 k : decrease throttle  
 j : turn left  
 l : turn right  
  
 5. Device Tilt:  
 Now tilting the phone forward will increase throttle and tilting it side

to side will turn the steering.  
  
 6. Joystick:  
 ---> Left analog stick - Left and right to adjust steering  
 ---> Right analog stick - Forward to increase forward throttle  
 ---> Pull back twice on right analog to reverse  
 # Joystick records the pictures automatically.

TRAINING AN AUTOPILOT:

Now, we are able to drive the car using keras to train a neural network.

1. COLLECTION OF DATA:

* Driving is practised around the track for a couple of times.
* Restart the manage.py process to create new-tub session. The joystick will record automatically.
* If crash or run off the track happens, the triangle button is pressed to erase the last 5 seconds of records.
* After collecting the 10-20 laps of data, the car can be stopped by using ctrl-c.
* The data is collected in the data folder in the most recent fold tub holder.

2. TRANSFER DATA FROM CAR TO COMPUTER:

* Raspberry pi is not very powerful. So transfer data to PC computer then to jetson Nano(As jetson nana is powerful).
* Use rsync to copy car folders from to copy cars folders from raspberry pi:-

    rsync –r [pi@192.1.168](mailto:pi@192.1.168)>  :~/mycar/data/

~mycar/data/ ~mycar/data

3. [Train a model](https://docs.google.com/document/d/10Xyd06n6CyXZW_PyonTvrN0HKOLAZCBTR4KehCSCuw8/edit)-To train the path model.

\*\* Tips to be followed while training:-

---> The options we look in the menu are user, local angle, local

Pilot (Local refers to the trained model which is locally hosted

on raspberry).

* User: By this option, we can control both the steering and throttle control.
* Local angle: Trained model controls the steering.
* Local pilot: Controls both steering and throttle.
* Build a simple track: Use the ribbons the tracks built and checked for various paths. The path is followed successfully.

CONCLUSION:

The ultimate goal of this project is to develop something useful from a self-driving autonomous car. Although, we made use of several libraries to make the car learn by models and trained paths, we are interested in developing an autobot which can clean and which can also be handy to supply the necessary materials to the users when ordered through commands.