

# Auto-Parking of vehicle by using Supervised Learning Approach

## Group member:

Mustaqeem Ashraf

Mazahir Hussain

Shamsa Kanwal

Tooba Anwar

## Approach:

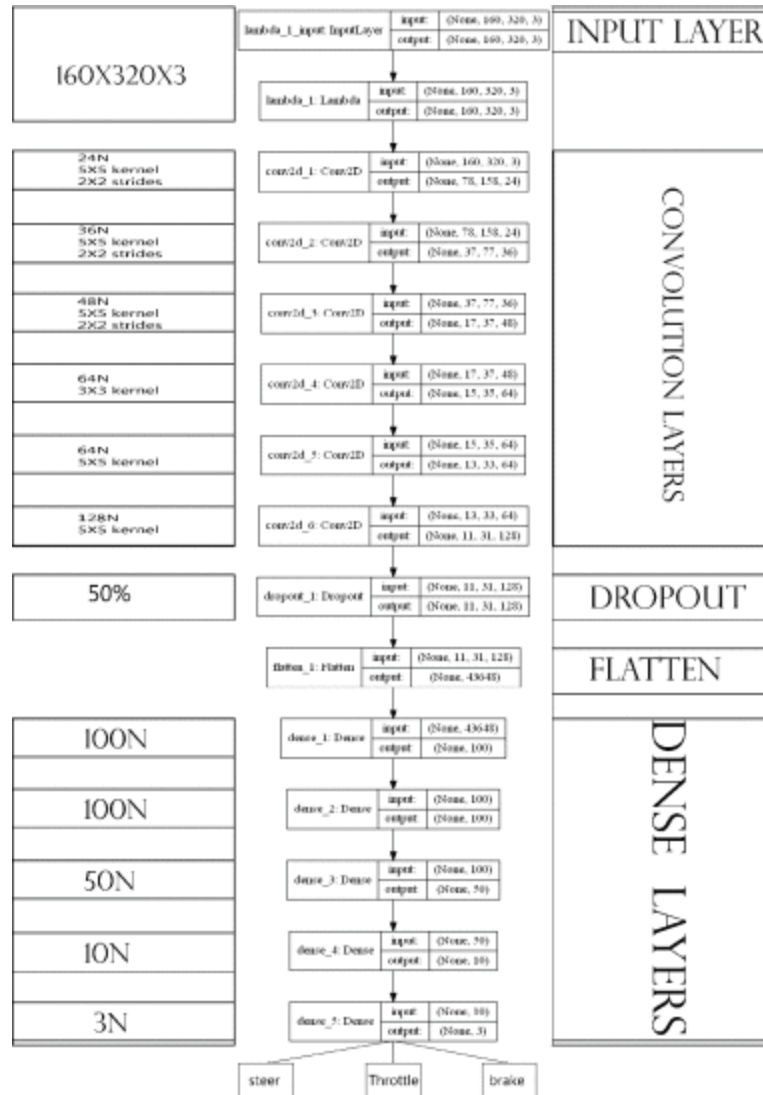
For the obtained model to be able to clone the behavior of humans and park the car in different environments and conditions without making mistakes, we used supervised learning as the approach for training the model.

## Neural network:

The neural networks used in this project is a regression one to predict the output of the steering angle showing in **Model architecture**.

## Architecture:

The NN architecture types used in our experiments are feed forward neural networks. Since we are working with inputs that are images, we use networks with convolutional layers as the first layers. The first layer as the input layer takes images with 160X320X3 where 160X320 are the height and width of the images and 3 is the number of channels (Red-Green-Blue) then there is 5 convolution layers with 24, 36, 48, 64, 64 neurons with the first three having filter sizes of 5X5 and a jump at each pass of the filter by 2X2 pixels (strides of sizes 2X2), and the final 2 layers having filter sizes of 3X3 with no strides. The approach of increasing the number of neurons at each layer helps with extracting details and features from the image easily. All the convolution layers have an elu activation function because it tends to converge the cost to zero faster than relu and produces more accurate results. A dropout layer is added after the convolution layers to prevent overfitting, the dropout layer as the name implements it drops randomly choosing a predefined percentage of neurons (50% in our case) and it drops out those neurons by disabling them. Then there is a flatten layer so that the multidimensional output of the convolution layers can be converted to a one-dimensional output. The last part of the network is 5 dense layers with 100, 100, 50, 10, 3 neurons respectively and all with elu activations except the output layer which has a linear activation because this is a regression problem, all the above layers are saved as a model of type Sequential.



Model architecture

[Show All](#)

The images are fed to the model as input data and the steering angel, throttle and brake as labels in the format of Numpy arrays. The hyper parameters (number of layers, number of neurons, steps per epoch, number of epochs, learning rate and batch size) are chosen empirically and after a number of trials the optimal values found are: Learning rate =0.0001 batch size =500 –600. Number of epochs =20. Steps-per-epoch =1000