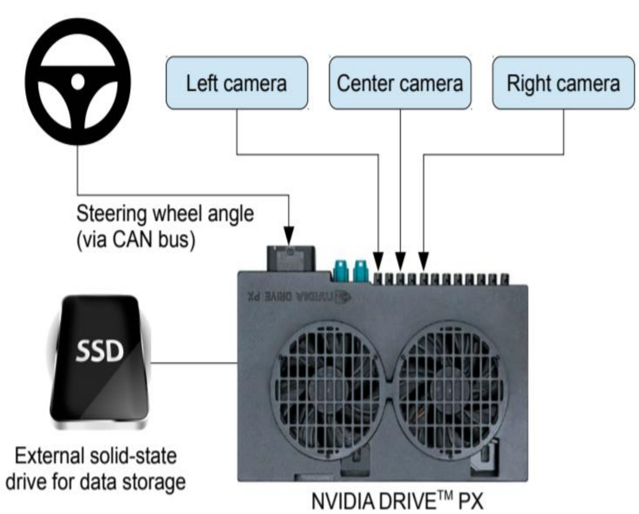
Team-4
Members: Ashwanth Ramji, Srujani Elango

# SELF DRIVING CARS – DEEP LEARNING

## PROPOSAL

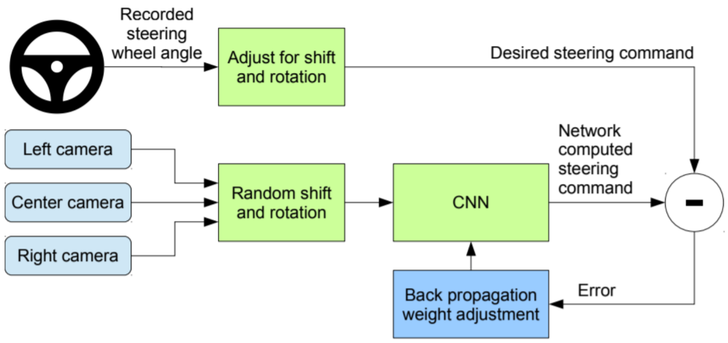
Autonomous vehicles are a very interesting and complicated topic which combine computer vision, deep learning and sensor fusion. All self-driving startups are working towards Level 4 autonomy. Renault has said to have achieved this, but it has still not been approved. The paper that we are interested in is End to End Learning for self-driving cars which was proposed by NVIDIA. Here they have trained a convolutional neural network(CNN) where a front facing camera is used directly to give steering commands. This method proves to be successful because it works well with minimum training data. This method works well on local roads and highways with or without lane markings. The system performance is much better when compared to traditional computer vision algorithms where lane marking detection, path planning, etc. are considered. The performance is better because there is no human selected intermediate criteria like lane detection. CNN has been in commercial use for over 20 years but only in the recent years has its use been maximized. This is because there much more data available like ILSVRC (Large Scale Visual Recognition Challenge) and the usage of high performance GPU's. The paper says that it is based on a seedling project known as DAVE where an RC car was trained on hours of human driving in similar, but not identical environments. Now there is much more data and computational power to take it to the next level and steer cars in public roads. CNN's are used so as to avoid human designated features like lane marking, guard rails and other cars.

## Working:




Three cameras (Left, Right and Centre cameras) are used to collect video input. The time stamped video along with the steering angle input by the driver at that time are recorded. The steering command is got by tapping into the car's controller area network and the steering command is represented by 1/r where r is the radius in meters. Training data contains images from the video along with the corresponding steering command. Steering wheel combined with all three camera inputs are fed to the NVIDIA Drive and the data is stored in a SSD.

To prevent the car from going off the road, data that show the car in different shifts from the center of the lane and rotations from the direction of the road is collected. All points below the horizon are assumed to be on flat ground and all points above the horizon are assumed to be far away. This technique might not work well for non-flat terrains and causes distortion. Objects like trees above the ground may cause distortion. Images are fed into the CNN which computed the steering command. The predicted command is compared to the original command and weights are adjusted.




This is how training of the neural network takes place. Once trained the network can generate steering commands form video images of the center camera alone. The researchers have collected data by drving cars in various cities on roads varying from local roads to highways and also varying climatic conditions.

**<u>Proposed Network architecture:</u>**

Number of layers: 9 (1 normalization layer, 5 convolutional layers and 3 fully connected layers). First three convolutional layers are strided and 2x2 stride and 5x5 kernel. The last two convolutional layers are not strided ones and have a 3x3 kernel size.

**<u>Method proposed for implementation:</u>**

We have to select what frames are useful for our problem and the data we have has labels of whether the driver was staying in lane or not. All data with labels of staying in lane are used as input for the CNN. Curved road frames are inserted more to remove bias for straight road frames. After the final set of frames are decided, data augumentation is done by adding artificial shifts and rotations so that the network knows how to recover from poor position or orientation. The trained network is put to test on a simulator before testing it out in the real world with an actual car. To understand how self-driving cars leverage deep learning and run an implementation of it, we will use the Udacity self-driving cars dataset. We are proposing to recreate the network and traiing and testing on a simulator. But for training instead of using data from the real world that are collected by driving we are proposing to collect data from a driving simulator itself. We will train the neurall network by driving the car in the simulator and collecting video data of the camera as one input. Keyboard arrows directions are used instead of steering commands. We can either use our keyboard here or we can use a simulator steering wheel that can be connected to our system. This method is called behavioural cloning. We are going to use the neural network architecture mentioned in the paper and use an unity car simulator application to record data and test our trained model with.

**<u>References:</u>**

- https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf
- https://www.youtube.com/watch?v=NJU9ULQUwng
- https://spectrum.ieee.org/cars-that-think/transportation/self-driving/how-driveai-is-mastering-autonomous-driving-with-deep-learning
- https://github.com/udacity/self-driving-car/tree/master/datasets