Machine Learning Engineer Nanodegree

Capstone Proposal

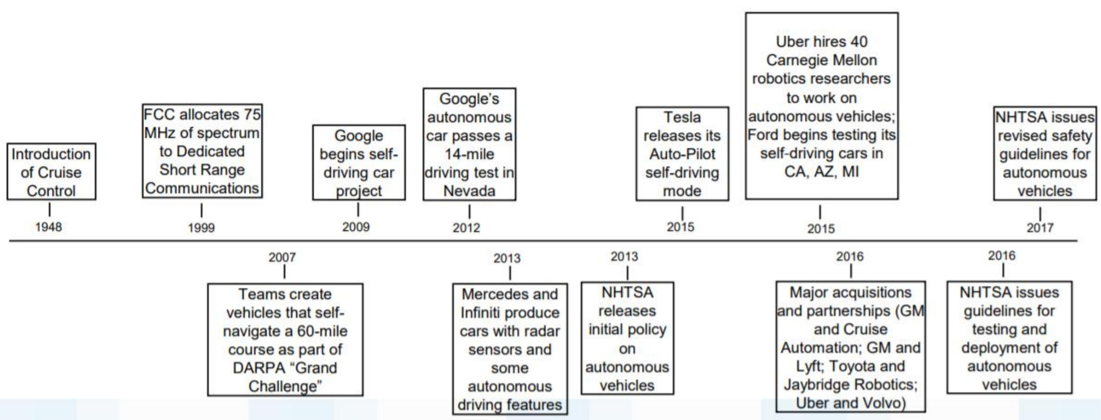
Ahmed Awad
September 20th, 2019-09-20

Proposal

Domain Background

Autonomous vehicles had always been a dream. Imagining a car driving itself without any human interference was science-fiction and was first introduced in movies. Today, as we talk there are many companies out there building autonomous cars.

Experiments have been conducted on self-driving cars since at least the 1920s. Promising trials took place in the 1950s and work has proceeded since then. numerous major companies and research organizations have developed working prototype autonomous vehicles including Mercedes-Benz, General Motors, Continental Automotive Systems, Autoliv Inc., Bosch, Nissan, Toyota, Audi, Volvo, Vislab from University of Parma, Oxford University and Google.



References:

https://www.mcca.com/wp-content/uploads/2018/04/Autonomous-Vehicles.pdf

https://en.wikipedia.org/wiki/History_of_self-driving_cars

Problem Statement

The objective of this project is in-lane driving, where the car should learn to identify the lane and stay driving within the lane. This will be viewed as a supervised learning problem.

Datasets and Inputs

This will be viewed and treated as a classification problem instead of regression and the reason is that I will use my self-collected dataset which is collected using a monocam mounted on an RC toy car which doesn't steer with angle. Also, I made my own lanes using white paper sheets.

Note that: my RC car has only one angle of rotation to the left and to the right which justifies the classification approach with 3 classes.

The dataset will be in the form of an input RGB image and the target is the class (Right, Left, Forward) which is used to control the car. All images are consistent in dimentions (256, 455, 3).

The sampling rate that is used for collecting the images/dataset is 5 fps to avoid redundant frames. Also, tried to make the three classes balanced and avoid biasing towards the forward class. I collected 10k examples in my dataset.

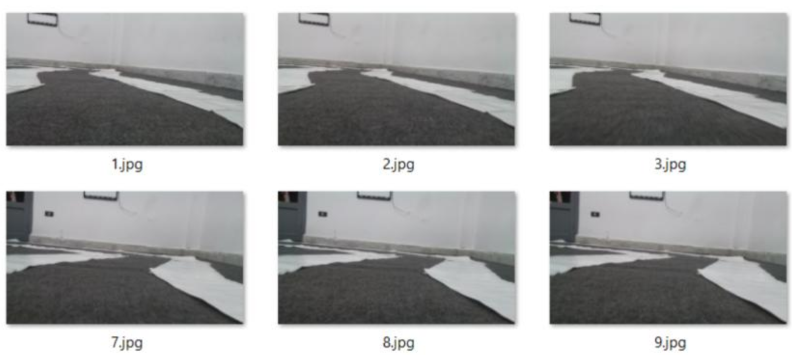


Fig (1)-Samples from my dataset

Solution Statement

Training a convolutional neural network architecture (CNN) on the dataset, after which the network should be able to predict the appropriate class (control action) given an input live image during driving.

Benchmark Model

For this task a simple benchmarking assumption will be used where all images are classified as forward class.

Evaluation Metrics

Accuracy score will be used as evaluation metric. Accuracy is calculated as the number of correctly classified examples over all the examples. For the naïve benchmark we are using, accuracy is 33% assuming the dataset is balanced. Of course, for accuracy to be a good metric, I will be careful balancing the three classes during collecting the dataset.

Project Design

Preprocessing:

- EDA on the classes and downsample or upsample to achieve class balance which is essential for our accuracy metric.
- Data augmentation techniques like brightness/contrast and horizontal flipping.
- Cropping the region of interest (the lane) to be the input to the network (200,66,3).
- Normalizing all the images by dividing pixels/255 before being fed to the network.

Data Splitting:

- Data will be split to train and test sets with ratio 80:20 respectively.
- EDA on the test set and downsample or upsample to achieve class balance which is essential for our accuracy metric to be fair.

Model Architectures:

- Will try simple architectures with stack of conv and pooling layers then a fc layer and training them from scratch using my dataset.
- Maybe using transfer-learning on some famous models like vgg-16.

Model Training and Evaluation:

- Training the proposed models and evaluating the models until we reach a good accuracy, hopefully.

References:

[1] Nvidea's paper "End to End learning for self-driving cars":

https://images.nvidia.com/content/tegra/automotive/images/2016/solutions/pdf/end-to-end-dl-using-px.pdf