Machine Learning Nanodegree

Capstone Project Proposal

Suresh Aswathnarayana

April 3, 2017

Classify Traffic Signs Using Deep learning

1. Domain Background

Autonomous vehicles have been gaining speed over the last couple of years. The Darpa Grand Challenge a prized competition for autonomous vehicles a decade ago sowed the seeds for developing autonomous vehicles. The aim of this competition was to bring out the research work that was happening in only the academia and big research labs into main stream implementation. Autonomous vehicles utilizes a combination of advanced hardware and software in order to perform the driving task. The wiki page to the Darpa grand Challenge is shown here https://en.wikipedia.org/wiki/DARPA_Grand_Challenge.

It involves the usage of mechatronics a combination of mechanical and electronic components. Some of the mechatronics components are integrated cameras, laser sensors, radars, lidars, and feedback loop control systems for braking, acceleration and steering mechanisms. In order to integrate all these hardware, software techniques involving computer vision, Natural Language Processing, robotics and advanced Artificial Intelligence is required. Artificial intelligence involves Machine Learning, Deep Learning and Reinforcement learning where in the vehicle learns based on the existing data and also from the new data that the vehicle gathers in operation.

2. Problem Statement

A requirement for an autonomous vehicle is that the vehicle responds to traffic signs posted on the road so that it can adjust to speed, brake if needed, steer automatically or perform other necessary tasks that a human driver would perform. This involved the techniques of computer vision where the cameras of the vehicle detects the features of the images in the traffic sign and Machine Learning, Deep Learning in order to let the vehicle know what the traffic sign means and respond accordingly. In this project we only develop the deep learning model to detect the traffic signs and any other implementation such as response to the traffic signs by autonomous vehicle is beyond the scope of this project.

The task of classification of traffic signs is basically a multi class classification where our machine learning model classifies different traffic signs based on the images provided. Images of different traffic signs are provided as input to the machine learning model. Each of the images have a label associated with it. For example if the image is a stop sign, it will have a particular label (example 14) associated with it. Similarly if the image is a 30 km/hr it will have a particular label (1) associated with it. Based on these image inputs the task of our machine learning classification model is to classify these images as the correct associated labels. So if we provide a large dataset of images to our machine learning model, the task is to classify these images into different labels that are associated with each of these images. Also the input images that we provide contains clear traffic sign images. We do not need to implement any other preprocessing steps like applying a bounding box onto the images for detecting the traffic signs in

the images. Below I show examples of the traffic signs from the German Traffic Sign Recognition Benchmark (GTSRB).

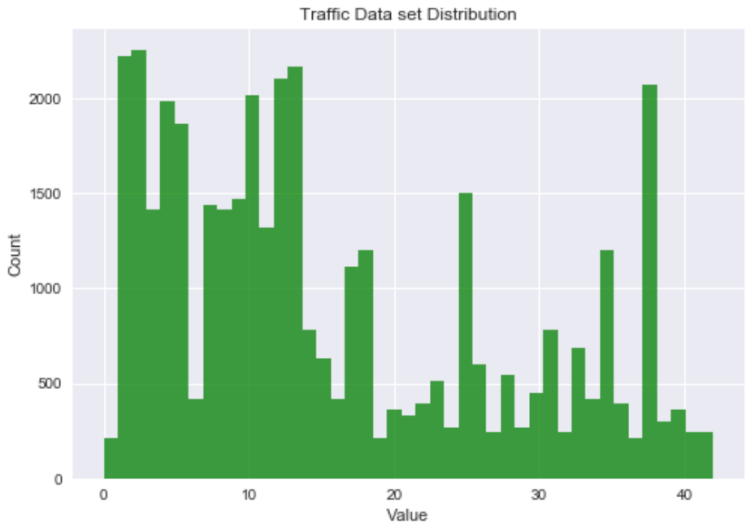
3. Datasets and Inputs

I plan to use the German Traffic sign recognition benchmark (GTSRB) data set for my final capstone project. This data set is a benchmark for computer vision and machine learning. This data set is a large multi- category classification data set. Below is the link to the GTSRB data set.

http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset

The data set contains approximately 39209 training data, 12569 test data and 43 classes as labels.

The images provide in the data set are in the .ppm format. Python provides libraries for reading the images in this format. Not all the images in the data set are of uniform size and a preprocessing step needs to be done to get all images to a uniform size (32x32). Images are in color format and plan to convert it to gray scale for training purposes. The data set is also imbalanced as there is no uniform distribution among all classes. This imbalance might prove to be challenging to train. A solution would be under sample the dominating classes or over sample the minority classes through data augmentation. A histogram of the distribution of the classes is shown below.



4. Solution Statement

I plan to use deep learning methods in order to classify the multi –class problem of the traffic signs. Deep learning models are effective at determining the different and important features of a given image. I plan on using the convolutional neural network or short for CNN. CNN is an effective deep learning algorithm for finding patters in images. My aim is to detect as many traffic signs as possible with my deep learning algorithm. The data set provided by GTSRB the images contain traffic sign images. I do need to implement any computer vision algorithms for traffic sign detection in the images. The preprocessing step I need is to resize the images to a uniform size, convert the images to gray scale and normalize the data. The images in the data set do not have much noise indicating that we might need to generate synthetic data for incorporating noise so that our CNN model generalizes data better. The task of generating data will be performed only if the CNN model overfits the existing data or does not generalize well on the test and un seen data.

5. Benchmark Model

I will compare my results using CNN vs the results listed under the results section of the German benchmark. The url is listed below.
http://benchmark.ini.rub.de/index.php?section=gtsrb&subsection=results

A snap shot of the results from the GTSRB is shown below.

TEAM	METHOD	TOTAL	SUBSET (All signs)
[3] IDSIA	Committee of CNNs	99.46%	99.46%
[155] COSFIRE	Color-blob-based COSFIRE filters for object recogn	98.97%	98.97%
[1] INI-RTCV	Human Performance	98.84%	98.84%
[4] sermanet	Multi-Scale CNNs	98.31%	98.31%
[2] CAOR	Random Forests	96.14%	96.14%
[6] INI-RTCV	LDA on HOG 2	95.68%	95.68%
[5] INI-RTCV	LDA on HOG 1	93.18%	93.18%
[7] INI-RTCV	LDA on HOG 3	92.34%	92.34%

6. Evaluation Metrics

The CNN trained on the data set will be compared for accuracy. The model will be trained to soft-max regression type multi class problem and the accuracy will be compared to the results given in the website listed above. Since our data set is unbalanced we also use the confusion matrix so as to find how accurate our model classifies the different classes of the traffic signs. Confusion Matrix indicates which off the classes is accurately classified and which not. This helps us in improving and tuning the model as to increase the accuracy of the model.

7. Project Design

- Collection of data here involves the already data set provided. I will try to utilize one or all the data sets regular RGB representations, HOG features, Haar-like features or the Hue histogram to see which one or the combination of the features would give me the best accuracy.
- Basic data exploration.
- Resize the images to one size (32x32) as the all the images in the data set might not be the same size.
- Convert to gray scale if needed.
- Scale the images between 0 and 1.
- Build the Convolutional Neural Network model. The choice of layers, sub sampling, pooling I would do based on the accuracy of my training.
- Calculate the accuracy of the models trained and compare it to the available benchmark.