Machine Learning Engineer Nanodegree

Capstone Proposal

Lang Su

July 28th, 2018

Proposal

Domain Background

Traffic sign recognition is one of the most important parts in self-driving car field. It enables vehicles to recognize modern traffic signs on the road during driving such as speed limit and stop sign.

On November 8th, 1968. The Vienna Convention on Road Signs and Signals was held to increase road safety and aid international road traffic by standardising the signing system for road traffic, such as road signs, traffic lights and road markings (Wikipedia, 2018).

The first TSR systems which recognized speed limits were developed appeared at the end of 2008, on the redesigned BMW 7 Series. Deep learning neural networks method is one of the major algorithms for pattern recognition problems currently. It is also widely used on Traffic signal recognition on self-driving cars.

Problem Statement

The problem to be solved is how to teach computers recognize the traffic sign when it receives a traffic sign image no matter what the scale is. For instance, we want the machine to know that the sign in the picture is a 40km/h speed limit sign even though the sign is relatively small and not perfectly clear in the picture, and output “40km/h Speed Limit”.

Datasets and Inputs

The dataset I used is a subset of the Belgium Traffic Signs database. There are 62 classes in total, and there is one directory for each class. Each directory contains the corresponding images of the sign in .ppm format (RGB color), as well as a csv file that has annotations. There are training dataset and testing dataset already separated in advance.

Note: the delimiter of the csv file is semicolon (;).

The csv file contains the following details,

Filename -- Image file the following information applies to

Width, **Height** -- Dimensions of the image

Roi.x1,Roi.y1, Roi.x2,Roi.y2 -- Location of the sign within the image

ClassId -- The class of the traffic sign

The input images are the keys to train our machine to learn how to recognize traffic signs. The csv file also helps us focus on the right spot on the image and classify different traffic signs.

Reference:

Radu Timofte*, Markus Mathias*, Rodrigo Benenson, and Luc Van Gool, Traffic Sign Recognition - How far are we from the solution?, International Joint Conference on Neural Networks (IJCNN 2013), August 2013, Dallas, USA.

Solution Statement

The solution to this traffic sign recognition system project is Convolutional Neural Network method in Deep Learning. Trough CNN and the input images, our machine is expected to: classify traffic signs in a video stream or in a picture.

CNN is an operation of filtering input data by setting up layers (filters). In image recognition, we use 2D filters to process data to generate the features. During training, the coefficients of these features are computed and optimized through backpropagation algorithm with gradient descent.

Benchmark Model

The benchmark model for this project is the 60% accuracy score when doing testings in the test dataset.

Evaluation Metrics

The evaluation metric that is used to quantify the performance of both the benchmark model and the solution model is given by the following mathematical formula:

Accuray score = # Correct predictions when comparing to correct labels in test dataset / # tests

For example, when doing a number of 100 tests, the model made 70 correct predictions and the remaining 30 predictions are incorrect, the accuracy score is:

70 / 100 = 70%

Project Design

Step 1: Import training dataset

Import the dataset at the first place, display and store the total number of data, and randomly display a few traffic sign images.

Step 2: Build a traffic sign detector using OpenCV CascadeClassifier, which helps detect objects in a video stream.

The final expected behavior is that, our model is able to detect the traffic sign in a video stream or in a picture. Building such a classifier is important and will boost our confidence even though it cannot tell what the sign means yet.

Step 3: Create a Convolutional Neural Network with transfer learning to classify traffic signs

By applying proper number of layers with certain parameters, we will have the first CNN to train our model.

Step 4: Compile and train the model

Compile and train our model with training dataset.

Step 5: Use the formula above to calculate the accuracy score running on the test dataset

Test our model using dataset for testing.

Step 6: Adjust parameters, layers to achieve higher test score