Machine Learning Engineer Nanodegree

Computer Vision Capstone Project Proposal

Thirupathi Pattipaka

May 5th, 2017

Domain Background

There are various companies are designing autonomous vehicles or self-driving cars. For autonomous vehicles, it is essential paramount to automatically detecting/classifying the traffic signs. It is very important to classify the traffic signs to avoid any accidents. Traditionally, computer vision and machine learning algorithms are used to classify these signs. Since the availability of GPU, deep neural networks able to produce better accuracy within limited time.

Problem Statement

This project aimed to classify German traffic sign data by using deep convolutional neural networks using Keras and TensorFlow. I will develop a new classification algorithm based on deep learning algorithms (CNN) and compare it with pre-exited classification algorithms.

Datasets and Inputs

The German Traffic Sign Dataset (http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset) consists of 39,209 32×32 pixel color images that we are supposed to use for training, and 12,630 images that we will use for testing. Each image is a photo of a traffic sign belonging to one of 43 classes, e.g. traffic sign types.

Each image is a 32×32×3 array of pixel intensities, represented as [0, 255] integer values in RGB color space. Class of each image is encoded as an integer in a 0 to 42 range. Let's check if the training dataset is balanced across classes.

The images contain one traffic sign. Images contain a border of 10 % around the actual traffic sign (at least 5 pixels) to allow for edge-based approaches. Images are stored in PPM format (Portable Pixmap, P6) and size varies between 15x15 to 250x250 pixels and they are not necessarily squared. The actual traffic sign is not necessarily centered within the image. This is true for images that were close to the image border in the full camera image

Solution Statement

The main aim of this project is to achieve the better accuracy than what was achieved in the competition by trying different architecture of the CNN models.

Benchmark Model

A Convolutional neural networks (CNNs) showed particularly high classification accuracies in the competition which was reported in the published paper (http://www.sciencedirect.com/science/article/pii/S0893608012000457)

Evaluation Metrics

These are following metrics I am planning to use evaluate performance of algorithms.

Accuracy = (True positives + True negatives)/ (True Positives + False Positives + False Negatives + True Negatives).
F1 = 2 * (precision * recall) / (precision + recall)
precision = True positives / (True positives + False positives)
recall = True positives / (True positives + False negatives)

Accuracy is defined as total number of correct predictions (True positives and true negatives) divided by total number of predictions made (TP + FP+FN+TN). Precision is the number of True Positives divided by the number of True Positives and False Positives. Put another way, it is the number of positive predictions divided by the total number of positive class values predicted. It is also called the Positive Predictive Value (PPV). Recall is the number of True Positives divided by the number of True Positives and the number of False Negatives. Put another way it is the number of positive predictions divided by the number of positive class values in the test data. It is also called Sensitivity or the True Positive Rate. The F1 Score is the 2*((precision*recall)/ (precision + recall)). It is also called the F Score or the F Measure. Put another way, the F1 score conveys the balance between the precision and the recall.
It is hard to judge that different traffic signs may have importance relative to other traffic signs. Therefore, it is a non-trivial judgment of which metric is the best, so I am using all metrics in this project.

Project Design

1. Import and Explore the data

First, I will plan to explore the data. I want to see the distribution of images per class to understand the distribution is even across all classes.

2. Preprocessing the data

I will resize all images to have same size for images and will scale images as well.
http://benchmark.ini.rub.de/?section=gtsrb&subsection=dataset

3. Train the model

For my first attempt, I want to try the Convolutional neural networks. As I iterate on the design I might try to fine tune the parameters of CNN algorithm.

4. Evaluate/compare models

Using confusion matrices, accuracy, precision, F1 scores I will compare the models and try to improve the accuracy of model.