Capstone Proposal: Traffic Sign Recognition

Tom Schwarzburg

Domain Background

Autonomous driving cars are often in the news. Be it Tesla with their Autopilot or similar solutions from more traditional carmakers, they must solve the same problems. One task in this huge field is recognizing traffic signs. Autonomous cars still must obey the traffic laws on the roads. As there currently is no good way to for example be able to tell the car which speed it is allowed to drive on a certain road, it has to get this information from traffic signs.
One big paper regarding recognition of traffic signs is the German Traffic Sign Recognition Benchmark[1](GTSRB), but due to its release in 2011, its shown solution does not make use of many improvements in Neural Networks that we have today.

Problem Statement

There are two problems that will be solved in this proposal. The first is in regard to the given Benchmark in the GTSRB paper: we have six classes (speed limit signs, other prohibitory signs, derestriction signs, mandatory signs, danger signs and unique signs) and for a given picture of a traffic sign, the created model should be able to classify it into these six classes.
The extended problem will have the same task, but with more classes. Instead of grouping similar signs into the six classes of the benchmark, we will use all 43 classes that are represented in the dataset.
The final solution should provide two models that are able to classify a given picture, once with grouped classes and once with the exact sign classification.

Solution Statement

To solve the given problem, I plan to use a ResNet-50 backbone pretrained on ImageNet, freeze the layers of the given backbone and amend it with two fully connected layers. Once the final layer has 6 neurons for the six combined classes and once it will have 43 Neurons, representing all the different traffic signs. As we have a classification problem with multiple classes, I will use Cross Entropy Loss as cost function for the model. In the next chapter I will introduce the dataset from the GTSRB that I am going to use for training the network. I will randomly split that dataset into a training and validation set with 90% training data and 10% validation. The trained model will be used to classify the data in the testing set, which will be used to calculate an evaluation metric to compare it to the benchmark model. The training data might have to be prepared by combining all the classes into the six combined classes.

Dataset and Inputs

The dataset used is from the aforementioned Benchmark, which is currently only available on Kaggle (https://www.kaggle.com/datasets/meowmeowmeowmeowmeow/gtsrb-german-traffic-sign), as the original source is down. It contains around 50000 images of 43 classes of traffic signs. The sign size varies between 15x15 and 222x193 (pixels). We also have a csv with the ground truth for the training images and the testing images. The ground truth does not contain the six combined classes, so that will have to be added later. It does not contain traffic signs which where not often observed, so the skew should not be that huge. But there will probably still be signs which are more represented than

---

[1] Stallkamp, Johannes & Schlipsing, Marc & Salmen, Jan & Igel, Christian. (2011). The German Traffic Sign Recognition Benchmark: A multi-class classification competition. Proceedings of the International Joint Conference on Neural Networks. 1453 - 1460. 10.1109/IJCNN.2011.6033395.

others. As they all represent traffic signs in Germany, they are perfect for my use case. Due to the low number of different classes, the resulting model will not be able to classify all possible traffic signs. Also, the main task that gets solved with this dataset is recognition. In the real world, we would still have to recognize the traffic signs in a picture taken by a camera to be then able to use the created model.

Benchmark Model

For comparison purposes I only use the model that was proposed originally in the paper with 6 classes. It will be interesting to compare my models with less optimization to state-of-the-art models in 2011 with huge efforts to optimize them. The chose models to compare against is “cnn_hog3” from team “IDSIA”. The other model for 43 classes that I will create, will simply show how good modern neural networks are, which does not necessitate a model to compare against.

Evaluation metric

The benchmark in the paper uses the correct classification rate (CCR), or often simply called accuracy, to judge the performance of the models. The CCR is calculated by adding the true positives (object is class 1 and detected as class 1) and true negatives (object is not class 1 and detected as not class 1) divided by the total number of all samples. This will be given for the all the classes combined and also as an individual metric for each class, to detect which classes perform the worst.

Project Design

The project will be developed step by step with a Jupyter notebook using AWS Sagemaker. I will start by uploading the data to a S3 bucket, loading it in the notebook and pre-processing the data and creating different metrics and visualizations to understand it. Then I will create, and train models as described beforehand and depending on the correct classification rate for the test data maybe implement a hyperparameter tuning job to improve the results. To further understand why some classes may have a low CRR, I will also create a confusion matrix of the results, to see which classes get misclassified as what class. The training will also be done on multiple instances to decrease training time. The final models will be deployed on an AWS endpoint and for testing purposes be queried using lambda functions.