Traffic Sign Classification

Colin Howes
University of Waterloo

Abstract

Efficient and accurate classification of traffic signs constitutes an interesting computer vision problem with a high degree of real-world relevance. Traffic sign recognition is necessary for the development of autonomous vehicles and driver assistance systems, which must be able to correctly detect and classify traffic signs in real-time under varying conditions. As such, traffic sign classification has enjoyed significant recent attention, and current techniques are able to meet or surpass human performance in publicly available data sets. Here, I compare successful approaches to traffic sign classification and present the performance of a traffic sign classifier based on a convolutional neural network.

1 Introduction

Traffic sign classification is a computer vision problem with a great deal of relevance to current advancements in autonomous vehicles and driver assistance systems. A viable traffic sign classifier must not only detect and categorize signs with a high degree of accuracy, it must perform efficiently, tolerate potentially noisy environments, and perform correctly in the presence of defects or variations in sign appearance [12, 13].

While many techniques have been applied to address the traffic sign classification problem, here I focus on a representative subset of state-of-the-art techniques that exhibit near-human performance in empirical testing: linear discriminant analysis, support vector machines, convolutional neural networks, and committees made up of multiple classifiers.

2 Benchmarks

The techniques discussed here were selected from high performing approaches entered into the German Traffic Sign Recognition Benchmark (GTSRB) competition, or developed and tested on this dataset after the conclusion of the competition [12, 13]. This competition provides a high quality data set of traffic sign images split into training and test sets that can be used to design, validate, and compare approaches to traffic sign classification. The data set consists of 34799 training images and 12630 test images comprised of 43 classes. The data set is unbalanced, with class representation ranging from 0.5 % and 5.5 % of images [12].

The German Traffic Sign Recognition Benchmark was originally formulated as a competition in two stages. Following the completion of the final stage, the GTSRB data set was made available online as a public benchmark to drive the development and empirical analysis of new classification algorithms [12, 13]. Here, I discuss the performance of classifiers entered at each stage of the competition, as well as other classifiers developed following the conclusion of the competition that provide an analysis based on the GTSRB data set.[1]

3 Comparison of Techniques

3.1 Features

Traffic signs are designed to be easy for humans to detect and recognize quickly. In order to accomplish this, signs tend to be made up of distinct shapes, have high contrast lettering or pictograms, and strong colouration for high-importance sign classes (e.g. stop signs). The techniques discussed here make use of different feature sets to drive training and classification.

While contrast and colour are fairly easy to define based on RBG and greyscale pixel values, other features are more difficult to represent. In particular, histograms of oriented gradient (HOG) are commonly used

---

[1]It is worth noting that all of the published classifiers developed after 2012 that I have found as of this writing have provided an analysis using the GTSRB data set, so limiting discussion to approaches using this benchmark does not limit the scope of this paper.

by image classification algorithms as a means of determining shape and orientation information. Briefly, the orientation of pixel values across image subregions are computed and combined hierarchically to produce feature vectors representing geometric information about an image [12, 13]. A number of other features were preprocessed and made available along with the GTSRB data set, but they were not used in any of the classifiers compared here, so they are not discussed.

3.2 Linear Discriminant Analysis

Linear discriminant analysis (LDA) ...

3.3 Support Vector Machines

3.4 Convolutional Neural Networks

Convolutional neural networks (CNNs) ...

3.5 Committees

Multi-column deep neural networks (MCDNNs) ...

3.6 Discussion

In general, methods based on convolutional neural networks have been shown to offer extremely high accuracy, matching or even exceeding human performance [2, 3, 10, 11]. This high accuracy comes at the expense of both training and classification time, with large networks requiring days to train. Additionally, access to high-end hardware is necessary in order to achieve real-time classification with a trained model, making it difficult to implement high performant CNNs in embedded hardware that may need to support other computationally expensive functionality such as lane detection [2, 3, 4].

Linear discriminant analysis and support vector machines offer a significant performance advantage over neural networks, but at the cost of some accuracy. Although both LDA and SVM based techniques boast higher than 95 % accuracy, this may not be acceptable for real-world sitations where recognizing and reacting to a sign is critical for passenger safety []. Moreover, the designers of the GTSRB competition note that some of the human control participants found the interface used to gather human responses was uninituitive and difficult to use [12, 13]. As a result, human performance may have been artificially low due to testing conditions, while the nature of the constructed data set was likely advantagous to the classifiers since each test case has a well defined class with none of the ambiguity associated with real-world driving scenarios. Thus, implementations that fail to match human performance in an artificial environment

are likely not good candidates for use in real autonomous vehicles or other safety-critical applications.

As hardware continues to become more powerful, computationally expensive classifiers based on committees of neural networks will likely become an even more attractive approach due to their robustness and accuracy. Indeed, advances in powerful, highly parallel embedded components, such as FPGA, are already seeing use in the automotive industry, are already being used in autonomous driving research [5, 7].

Finally, the emergence of "standard" benchmarks for traffic sign classification is something that bears consideration. The motivating idea behind benchmarks is that they give researchers and developers a way to assess how well a system performs. The consequence of this is that systems end up being tailored towards high performance on a benchmark rather than on the task they are designed to perform. In the context of machine learning, this could lead to overfitting, such that classifiers perform well on benchmarks but not on real-world examples.

One solution to this is to adopt many different benchmarks, and use classifiers that perform well on all of them. This is solution suits the traffic sign classification problem well, given that different regions have adopted different sets of traffic signs and classifiers need to be trained on data from every region where it is expected to be used. Indeed, a number of public data sets have been released since the GTSRB data set was made public, including the LISA (United States) and BelgiumTS (Belgium) data sets [8, 9]. By benchmarking traffic sign classifiers on a diverse set of benchmarks will prevent overfitting and improve generalizability.

4 Methodology

In order to validate previous results demonstrating the effectiveness of convolutional neural networks for traffic sign classification, I implemented a simple traffic sign classifier based on a convolutional neural network and evaluated it using the German Traffic Sign Detection Benchmark data set [12, 13].

4.1 Architecture

My approach was similar to the approaches outlined in [10, 11], though the architecture I present here used all colour channels and is purely feed-forward, with the classifier layer connecting only to the final convolutional layer. Similar models have shown high performance in previous replications, including [1] and [14], and the simplicity of this architecture makes it an attractive choice for experimentation.

The model is made up of three pairs of feed-forward convolutional layers seperated by 2×2 max-pooling lay-
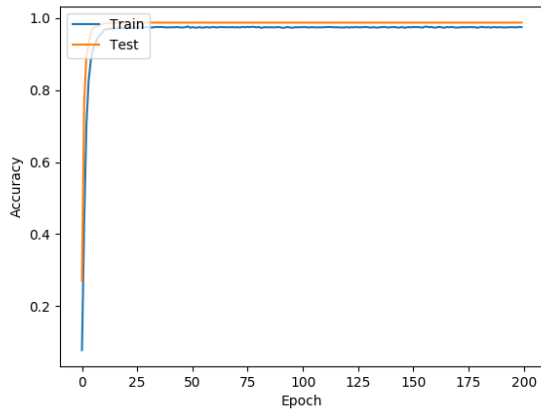
Figure 1: Accuracy over 200 epochs on the base image set. Data for "test" instances are taken from performance on a validation set at the end of each epoch
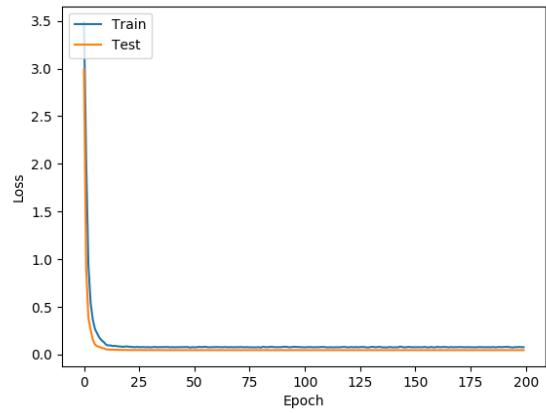

Figure 2: Loss over 200 epochs on the base image set. Data for "test" instances are taken from performance on a validation set at the end of each epoch

ers. Each of these pooling layers feeds into a dropout layer which drops $1/4^{th}$ of its inputs. The final set of convolutional, pooling, and dropout layers are flattened and passed into a classifier layer, which feeds into a softmax output layer that returns a probability distribution over the classes for a given input. The prediction for a given input is simply the maximum over this probability distribution.

4.2 Training

I trained the models over 200 epochs using the default batch size of 32 and a learning rate of 0.001. I used categorical cross-entropy as the loss function and stochastic gradient descent as the optimizer, which is suggested for multi-label classification problems [6].

5 Results

When training with only unaltered images provided in the GTSRB data set, my model had an accuracy of 95.42 % on the test set. This is reasonably high considering the simplicity of the model and lack of data set augmentation, though accuracy is considerably lower than both human performance (98.84 %) and the performance of other classifiers implemented using CNNs (98.31 %) [10, 11, 12, 13]. Figure 1 and Figure 2 show the accuracy and loss of the network over 200 training epochs when trained only on unaugmented images provided in the GTSRB data set.

In order to assess the benefit of augmenting the data set with simple transformations of training images, I trained the model again on the same architecture, randomly ap-

plying a gaussian blur to half of the training examples. I observed a marginal improvement over the unaugmented data set, with the new model correctly classifying 96.00 % of test images. Due to time limitations, it was not possible to experiment with other transformations, though I suspect that applying a random mix of transformations to generate the augmented data set would produce better results, since this technique has been used with considerable success in the past [2]. Figure 3 and Figure 4 show the accuracy and loss of the network over 200 training epochs when trained the augmented data set.

6 Conclusions

7 Summary

I presented an analysis and comparison of a set of traffic sign recognition techniques that performed well on a competition data set. In addition, I presented a simple implementation of a traffic sign classifier using a convolutional neural network and compared the effectiveness of this implementation with the performance of past techniques.

References

[1] CHILAMKURTHY, S. Keras tutorial - traffic sign recognition.

[2] CIRESAN, D., MEIER, U., MASCI, J., AND SCHMIDHUBER, J. A committee of neural networks for traffic sign classification. In *The 2011 International Joint Conference on Neural Networks* (July 2011), pp. 1918–1921.

[3] CIRESAN, D., MEIER, U., MASCI, J., AND SCHMIDHUBER, J. Multi-column deep neural network for traffic sign classification. *Neural Networks 32*, Supplement C (Aug. 2012), 333–338.
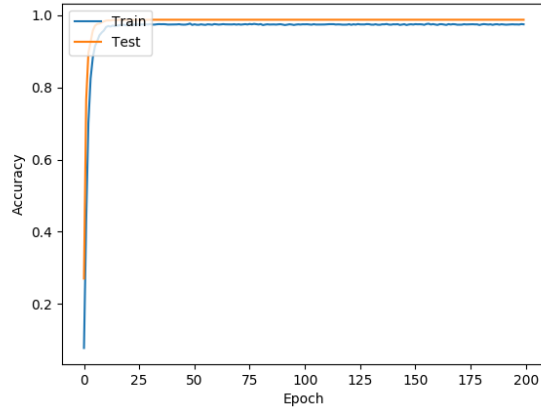
Figure 3: Accuracy over 200 epochs using data augmentation. Data for "test" instances are taken from performance on a validation set at the end of each epoch
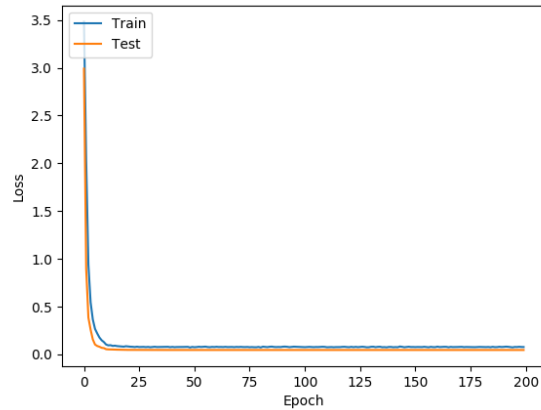

Figure 4: Loss over 200 epochs using data augmentation. Data for "test" instances are taken from performance on a validation set at the end of each epoch

[4] HILLEL, A. B., LERNER, R., LEVI, D., AND RAZ, G. Recent progress in road and lane detection: a survey. *Machine vision and applications 25*, 3 (2014), 727–745.

[5] INTEL. Autonomous driving is powered by intel.

[6] LASAGNE. lasagne.objectives - documentation.

[7] LIU, S., TANG, J., ZHANG, Z., AND GAUDIOT, J.-L. Caad: Computer architecture for autonomous driving. *arXiv preprint arXiv:1702.01894* (2017).

[8] MATHIAS, M., TIMOFTE, R., BENENSON, R., AND GOOL, L. V. Traffic sign recognition #x2014; How far are we from the solution? In *The 2013 International Joint Conference on Neural Networks (IJCNN)* (Aug. 2013), pp. 1–8.

[9] MOGELMOSE, A., TRIVEDI, M. M., AND MOESLUND, T. B. Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey. *IEEE Transactions on Intelligent Transportation Systems 13*, 4 (Dec. 2012), 1484–1497.

[10] SERMANET, P., CHINTALA, S., AND LECUN, Y. Convolutional neural networks applied to house numbers digit classification. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* (Nov. 2012), pp. 3288–3291.

[11] SERMANET, P., AND LECUN, Y. Traffic sign recognition with multi-scale Convolutional Networks. In *The 2011 International Joint Conference on Neural Networks* (July 2011), pp. 2809–2813.

[12] STALLKAMP, J., SCHLIPSING, M., SALMEN, J., AND IGEL, C. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *The 2011 International Joint Conference on Neural Networks* (July 2011), pp. 1453–1460.

[13] STALLKAMP, J., SCHLIPSING, M., SALMEN, J., AND IGEL, C. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks 32*, Supplement C (Aug. 2012), 323–332.

[14] STARAVOITAU, A. Traffic signs classification with a convolutional network.