Deep learning based Traffic Sign Classification and Class Verification

Nicolas Six
School of Computer Science
Georgia Institute of Technology, Atlanta, USA
nicolas.six@gatech.edu

Shashank Dhar Burugula
School of Electrical and Computer Engineerring
Georgia Institute of Technology, Atlanta, USA
shashankdhar@gatech.edu

Yufeng Yang
School of Electrical and Computer Engineerring
Georgia Institute of Technology, Atlanta, USA
yfyang@gatech.edu

Abstract

Traffic signs exist in our lives every day. With the development of artificial intelligence (AI) and deep learning, more and more methods have been developed to do pattern detection and recognition tasks. In our project, we explore two different classification and verification architectures on TT100k dataset, first using a DARTS-based model and second using a MobileNet-based model, the later achieving an accuracy of almost 90%. *We also designed a class verification model which achieves* 98.9% *precision, based on a Siamese network with MobileNet as the backbone model. Our model was successfully able to generalize well on test data which was completely unseen while in the training phase, where our test data was taken from a German traffic signs dataset, hence being very different from the training data, which consisted of traffic sign images collected from China. Our project focuses on the important application of traffic sign classification and verification, which can be a major step in further development of self driving cars.*

1. Introduction

In modern day, Artificial Intelligence(AI) is in the forefront amongst various fields of research, both in academic and industrial settings. Image classification is a useful application of AI with and with currently available algorithms and compute power for machine learning and deep learning techniques, has proven to be an almost solved problem. In our project, we focused on the application of deep learning on traffic sign classification and verification, essentially a image classification and verification task.

In the initial years, Traffic sign detection and classification have been attempted by Md. Abdul Alim et al in [1] where they've used color features and a simple neural network for the classification task. As years progressed, and deep learning models have taken over representation and classification tasks with outstanding precision, Shao Femind et al [2] have utilized simplified Gabor filtering methods to pre-process grey-scale images, passing them as an eight channel input to a CNN to perform the necessary feature extraction and classification, in real time environments.

Further work has been done in [3] where they use ”highly possible regions proposal network” (HP-RPN) to a modified R-CNN based network, essentially filtering out most non-traffic sign areas in images. Domen Tabernik et al [4] have utilized a Mask R-CNN detector, alongside using novel data augmentation techniques based on the distribution of geometric and appearance distortions, hence improving the learning capability on the domain of traffic signs.

2. Preliminaries

In our project, we performed two separate tasks: traffic sign classification and verification. The classification task being given a traffic sign, we have to give an output referring to its original class, such as a speed limit sign (with its value) or a height limit sign. For the verification task, given two images, we give an output indicating whether the two images are from the same class or not. For example, if the system has one speed limit sign and one prohibition sign, then the output is “Not Same”; if the inputs are two prohibition sign then the output is “Same”. To reach our goals, the dataset and data pre-processing methods chosen have been laid out below, in the following sections.

2.1. TT100K Dataset

In our project, we used Tsinghua-Tencent 100K (TT100K) dataset [5] as training and testing data. TT100K is a benchmark for traffic-sign detection and classification. It provides 100000 images containing 30000 traffic-sign instances. It is focused on small object detection in real world. Examples of images can be found in Fig. 1. Signs in yellow, red and blue boxes correspond to warning, prohibitory and mandatory signs respectively. In the dataset, each traffic sign has its own label. Also, some classes are grouped as a family such as speed limits for different speeds are grouped as one as "pl*". For example, "pl60" means speed limit for 60 kmph speed limit sign.

2.2. Data Pre-processing

Original TT100K data is not enough to train a good model, thus data augmentation was done on the available images to prevent over-fitting and improve the robustness of the models. In our project, because of the specific way in which most typical traffic signs look, some of the traditional transformations on image recognition like upside down rotation may change the meaning of the sign. Hence, meaningful augmentation methods that we selected are as follows.

- Slight angular rotation: Here we use a small angle rotation to prevent changing the meaning of the image. For example, if we have a warning sign that shows road changing to the left, and after we do the $180°$ rotation, the sign would have no meaning because there is no sign which means the road is changing backward to the right. So we can only apply rotations less than $45°$
- Illumination level change: Because in TT100K dataset, images are collected from different illumination levels, the traffic signs in the dataset may be different from what are shown in Fig. 1. Thus we have to change the illumination level of the signs to make our model capable to recognize the traffic signs in daylight setting and evening settings.

3. Traffic Sign Classification

In this section we include the details of our implementation of traffic sign classification task and the details of the architecture chosen.

3.1. DARTS-Based Model

Differentiable architecture search (DARTS) [6] was proposed based on continuous relaxation and gradient descent in the architecture space, as shown in Fig. 2. This method has brought about a large improvement in most of the Reinforcement Learning based methods, such as [7]. ENAS [8] brings improvement to this method and allows it to reach an efficiency close to DARTS, but DARTS still possesses the advantage of being an easy run on a single consumer grade GPU, making it a better choice for our study.

3.2. MobileNet-Based Model

MobileNet [9] was proposed as an efficient network architecture and a set of two hyper-parameters in order to build very small, low latency models that can be easily matched to the design requirements for mobile and embedded vision applications. It has fewer number of parameters and fewer number of multiplications and additions compared to traditional convolution neural networks.

In MobileNet, it uses a depth-wise separable convolution which is shown in Fig. 3.

Also, in MobileNet it uses another architecture of convolutional layer instead of the traditional layer, which is shown in Fig. 4.

4. Traffic Sign Class Verification

In this section we give details of our approach to class verification and the proposed model to deal with the task. In the verification problem, the task at hand is about how to design a network to tell if two traffic signs are from the same class or not. Because we know that some traffic signs are very similar to each other like speed limit signs with the difference only in number. Thus this task is also important nowadays, in application related to autonomous vehicles which have to exactly identify what the speed limit exactly is in order to accelerate/decelerate the vehicle accordingly.

4.1. Siamese-Based Model

A Siamese neural network [10] is an artificial neural network that uses the same weights while working in tandem on two different input vectors to compute comparable output vectors. Architecture of a Siamese network is shown in Fig. 5 [11].

To make use of input images, we have to use a specific model to extract the embeddings of the image. Like what we learned in class to do the image captioning task, we use the network of image classification to extract the image embeddings. Here we have utilized a Siamese-based model to do our task of traffic sign class verification.

The backbone model we have used to extract the image embeddings is a MobileNet [9], which was mentioned in Section. 3.2.

5. Numerical Results

In this section we will include the results of our two tasks.

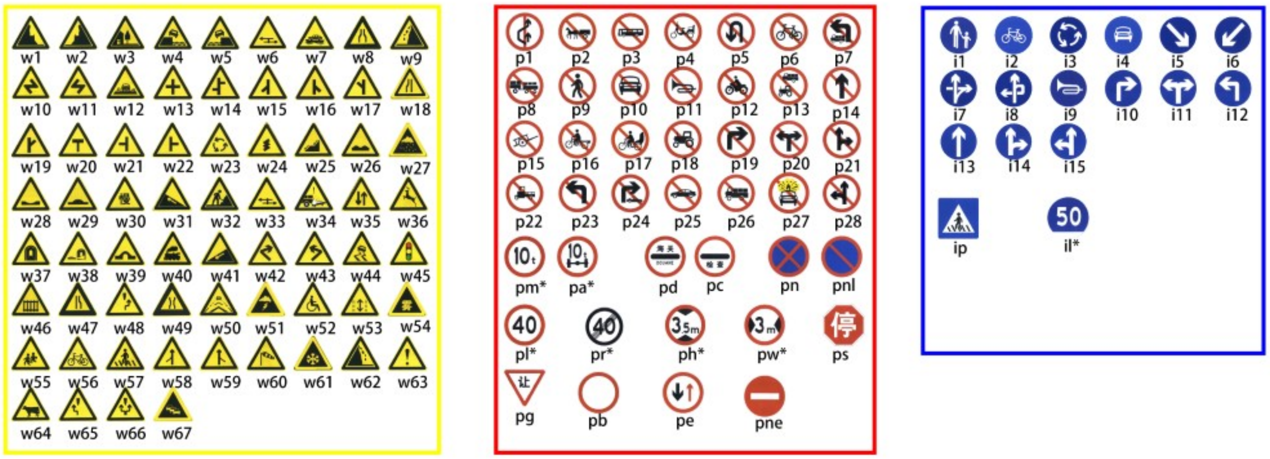


Figure 1: Traffic sign examples in TT100K.

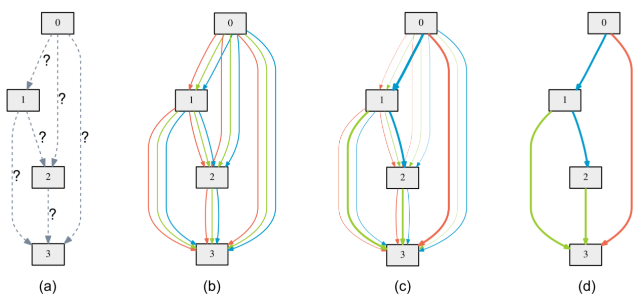
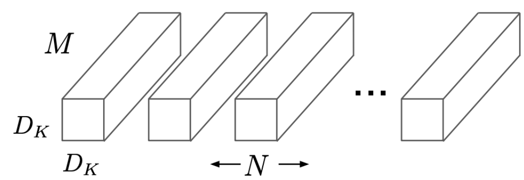
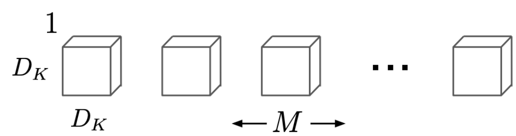


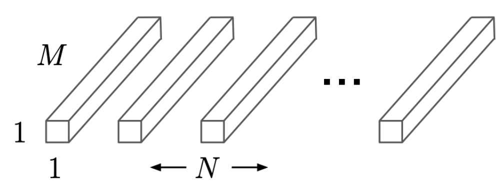
Figure 2: DARTS: (a) Operations on the edges are initially unknown. (b) Continuous relaxation of the search space by placing a mixture of candidate operations on each edge. (c) Joint optimization of the mixing probabilities and the network weights by solving a bi-level optimization problem. (d) Inducing the final architecture from the learned mixing probabilities.



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figure 3: The standard convolutional filters in (a) are replaced by two layers: depth-wise convolution in (b) and point-wise convolution in (c) to build a depth-wise separable filter.

5.1. Image Classification

5.1.1 Mobilenet baseline

The goal of this part is to get a quick baseline of the complexity of our task. The goal here is not just to fine tune to the best possible results but to see what a model such as a MobileNet can do.

You can see in Fig. 6 the evolution of the main parameters during training. It can be noted from this graph that if we could have trained further to improve the validation accuracy, then the validation accuracy would not have gone much higher than 90%.

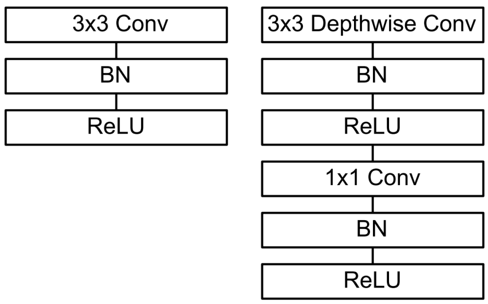


Figure 4: Left: Standard convolutional layer with batch norm and ReLU. Right: Depth-wise Separable convolutions with Depth-wise and Point-wise layers followed by batch norm and ReLU.

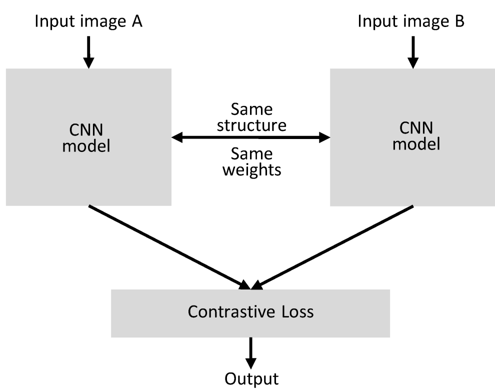
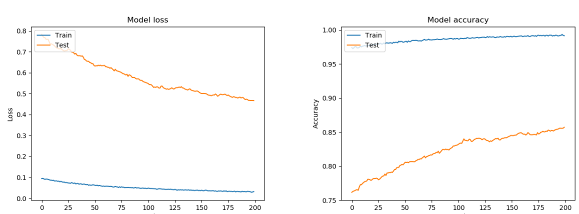


Figure 5: Architecture of Siamese Network.



(a) Loss (b) Accuracy

Figure 6: Evolution of the loss and metrics during training of MobileNet on TT100k.

5.1.2 Darts

Our first step in this study was to see if using Darts [6] could help to produce a better architecture than what we had with MobileNet [9] to do traffic sign classification while maintaining a high prediction speed.

This experiment has proven to be unsuccessful given the small number of data available. Darts quickly overfits the training data, reaching 100% top one accuracy, while having only 7% top one accuracy on the validation data, hence proving that this approach is not the way out to solve our specific problem.

More aggressive data augmentation or dropout could be used to improve the results, but given the time available we weren't able to access those points.

(a) One of the 4 FN cases (b) One of the 58 FP cases

Figure 7: Example of miss classification by our model on TT100k.

5.2. Image Class Verification

In this section we cover the results we obtained from the Siamese architecture described in Section 4.1. We will first show the results on the validation dataset and then see how these results can be extended to other data, without any additional training.

In each case, we randomly sampled $10,000$ pairs of images from our dataset each with a 0.5 probability to be of the same class.

5.2.1 TT100k results

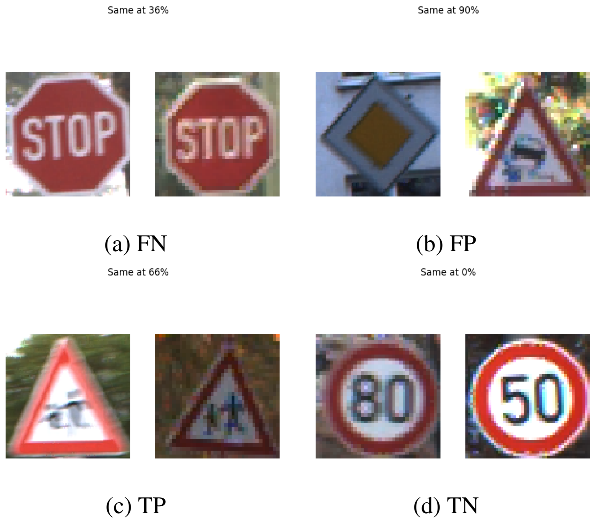
These results directly come from the validation split of our dataset and so are expected to be very good as every class were seen during training, and in similar conditions.

As you can see on Tab. 1, the results are following this expectation with a very high 98.9% precision and a 99.9% recall. You can find two example of miss classification on Fig. 7

Reviewing the results from our network actually make us discover error in the TT100k dataset, as our model was predicting the correct label, but the ground truth label given to that image was wrong. This shows that the dataset could be improved upon, by annotating the ground truth more accurately, so there are no errors taught to the models in the training phase.

5.2.2 GTSDB

GTSDB [12] is a dataset that is smaller than TT100k as it is only composed of $1,000$ images. But this size is still perfectly fine for our testing purpose. This dataset came from Germany and hence follows a traffic sign convention similar



(a) FN (b) FP (c) TP (d) TN

Figure 8: Example example of all the different cases of prediction of our model over GTSDB.

to the one used in china, with however a lot of differences. Hence, it makes this set a very interesting test bed as it provides a completely new set of classes, but with similarly shaped signs. The results on this set will illustrate how well this model is able to generalise on unseen traffic signs.

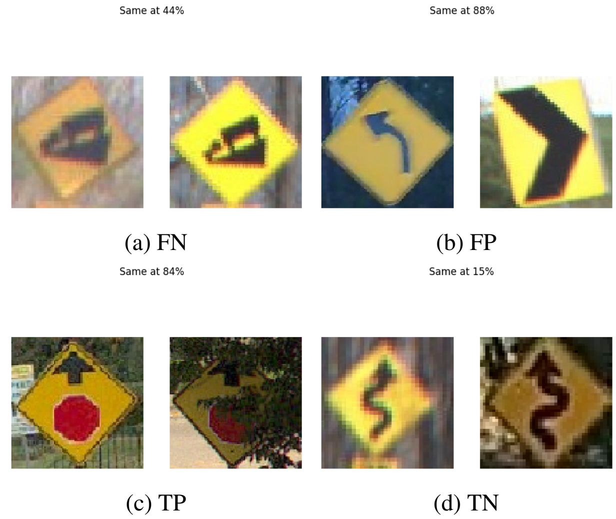
Tab. 1 give the numbers for this experiment. As you can see Fig. 8, even if our model never saw the given sign class before it's still able to generalise well and predict the correct label. You can see that the stop sign in China doesn't have an English word in it but instead has a Chinese character, showing that our model didn't learn about English character enough to generalise here. However, on the other side our model is able to tell the numbers apart as you can see on Fig. 8d.

In conclusion from this set of data, we can tell that our model is able to generalise to unseen classes in the cases of German Traffic signs.

5.2.3 US Curve

In this case we evaluated our model on a dataset of warning signs in the US. This dataset is not publicly available yet. These kind of signs being completely different from the signs used in our training we decided to use it to further test the generalisation capabilities of our model.

You can see on Tab. 1 that this dataset is getting lower precision and recall than the previous one, as expected. Fig. 9, show some interesting case we find during our study. As illustrated by Fig. 9a, our model as some trouble with pictographs that weren't seen during training previously, but is however still able to handle obstruction and minor modification such as the mirror transformation in Fig. 9c and 9d.



(a) FN (b) FP (c) TP (d) TN

Figure 9: Example example of all the different cases of prediction of our model over our US curve sign dataset.

Data	TP	TN	FP	FN	Prec.	Recall
TT100k	5042	4896	58	4	98.9%	99.9%
GTSDB	4424	4289	717	570	86.1%	88.6%
Curve	4474	2510	2476	540	64.4%	89.2%

Table 1: Results for the model trained on TT100k

So, if the precision is very low, due to a large number of False Positives, we were still impressed to get such accurate results on a dataset completely different from what we used for training, which shows the robustness and generalization capabilities of the developed architecture, as this dataset is composed of signs with completely different shapes and colors than the one used for training.

6. Conclusions

In this project, we performed traffic sign classification on TT100K dataset to do image classification in two different ways. We first explored the classical approach by directly classifying the image, using well know architecture such as MobileNet or trying to find the best performer for our task with an architecture search. We then tried another approach involving asking the network to compute the similarity between two given traffic signs using a Siamese network.

The first approach quickly showed its limits, reaching high accuracy but having problem with the classes with only few available examples. The second approach, however, fixed that issue and allowed to generalise the trained model to signs completely different from the one seen during training, by skipping the memorisation aspect of the task and so allowing to train a classification model with virtually only one example.

This last approach was proven to be more successful at solving this task than the first one, hence showing the benefits of incorporating a Siamese model in this kind of a classification task.

Acknowledgements

We would like to thank Prof. Dhruv Batra for giving us the opportunity to work on this project through the course CS7643 which we took in Fall 2019. We would also like to thank the TAs who've work on scoring our homeworks and for providing insights during their office hours. The guest lectures proved to be very insightful about diving deeper into the state-of-the-art deep learning methods used to solve various computer vision and NLP tasks. We learned a lot from the course and we believe that we can utilize some of the learned concepts through this course in our future research and careers.

References

[1] M. A. A. Sheikh, A. Kole, and T. Maity. Traffic sign detection and classification using colour feature and neural network. In *2016 International Conference on Intelligent Control Power and Instrumentation (ICICPI)*, pages 307–311, Oct 2016.

[2] Faming Shao, Xinqing Wang, Fanjie Meng, Ting Rui, Dong Wang, and Jian Tang. Real-time traffic sign detection and recognition method based on simplified gabor wavelets and cnns. *Sensors*, 18:3192, 09 2018.

[3] Faming Shao, Xinqing Wang, Fanjie Meng, Jingwei Zhu, Dong Wang, and Juying Dai. Improved faster r-cnn traffic sign detection based on a second region of interest and highly possible regions proposal network. *Sensors*, 19:2288, 05 2019.

[4] Domen Tabernik and Danijel Skocaj. Deep learning for large-scale traffic-sign detection and recognition. *CoRR*, abs/1904.00649, 2019.

[5] Zhe Zhu, Dun Liang, Songhai Zhang, Xiaolei Huang, Baoli Li, and Shimin Hu. Traffic-sign detection and classification in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Hanxiao Liu, Karen Simonyan, and Yiming Yang. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*, 2018.

[7] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578*, 2016.

[8] Hieu Pham, Melody Y Guan, Barret Zoph, Quoc V Le, and Jeff Dean. Efficient neural architecture search via parameter sharing. *arXiv preprint arXiv:1802.03268*, 2018.

[9] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[10] Sumit Chopra, Raia Hadsell, Yann LeCun, et al. Learning a similarity metric discriminatively, with application to face verification. In *CVPR (1)*, pages 539–546, 2005.

[11] Yoosoo Jeong, Seungmin Lee, Daejin Park, and Kil Park. Accurate age estimation using multi-task siamese network-based deep metric learning for frontal face images. *Symmetry*, 10(9):385, 2018.

[12] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2013.