Traffic Sign Recognition with Multi-Scale Convolutional Networks

Pierre Sermanet and Yann LeCun
Courant Institute of Mathematical Sciences, New York University
{sermanet,yann}@cs.nyu.edu

***Abstract*— We apply Convolutional Networks (ConvNets) to the task of traffic sign classification as part of the GTSRB competition. ConvNets are biologically-inspired multi-stage architectures that automatically learn hierarchies of invariant features. While many popular vision approaches use hand-crafted features such as HOG or SIFT, ConvNets learn features at every level from data that are tuned to the task at hand. The traditional ConvNet architecture was modified by feeding 1^{st} stage features in addition to 2^{nd} stage features to the classifier. The system yielded the 2nd-best accuracy of 98.97% during phase I of the competition (the best entry obtained 98.98%), above the human performance of 98.81%, using 32x32 color input images. Experiments conducted after phase 1 produced a new record of 99.17% by increasing the network capacity, and by using greyscale images instead of color. Interestingly, random features still yielded competitive results (97.33%).**

I. Introduction

TRAFFIC sign recognition has direct real-world applications such as driver assistance and safety, urban scene understanding, automated driving, or even sign monitoring for maintenance. It is a relatively constrained problem in the sense that signs are unique, rigid and intended to be clearly visible for drivers, and have little variability in appearance. Still, the dataset provided by the GTSRB competition [1] presents a number of difficult challenges due to real-world variabilities such as viewpoint variations, lighting conditions (saturations, low-contrast), motion-blur, occlusions, sun glare, physical damage, colors fading, graffiti, stickers and an input resolution as low as 15x15 (Fig. 1). Although signs are available as video sequences in the training set, temporal information is not in the test set. The present project aims to build a robust recognizer without temporal evidence accumulation.

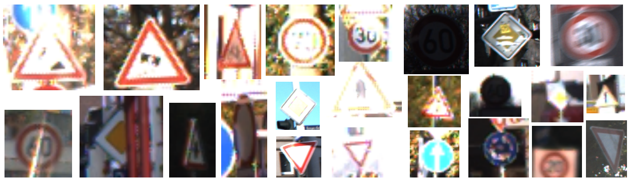

Fig. 1. Some difficult samples.

A number of existing approaches to road-sign recognition have used computationally-expensive sliding window approaches that solve the detection and classification problems simultaneously. But many recent systems in the literature separate these two steps. Detection is first handled with computationally-inexpensive, hand-crafted algorithms, such as color thresholding. Classification is subsequently performed on detected candidates with more expensive, but more accurate, algorithms. Although the task at hand is solely classification, it is important to keep in mind the ultimate goal of detection while designing a classifier, in order to optimize for both accuracy and efficiency. Classification has been approached with a number of popular classification methods such as Neural Networks [2], [3], Support Vector Machines [4], etc. In [5] global sign shapes are first detected with various heuristics and color thresholding, then the detected windows are classified using a different Multi-Layer neural net for each type of outer shape. These neural nets take 30x30 inputs and have at most 30, 15 and 10 hidden units for each of their 3 layers. While using a similar input size, the networks used in the present work have orders of magnitude more parameters. In [6] a fast detector was used for round speed sign, based on simple cross-correlation technique that assumes radial symmetry. Unfortunately, such color or shape assumptions are not true in every country (e.g. U.S. speed signs [7]).

By contrast, we will approach the task as a general vision problem, with very few assumptions pertaining to road signs. While we will not discuss the detection problem, it could be performed simultaneously by the recognizer through a sliding window approach, in the spirit of the early methods. The approach is based on Convolutional Networks (ConvNets) [8], [9], a biologically-inspired, multi-layer feed-forward architecture that can learn multiple stages of invariant features using a combination of supervised and unsupervised learning (see Figure 2). Each stage is composed of a (convolutional) filter bank layer, a non-linear transform layer, and a spatial feature pooling layer. The spatial pooling layers lower the spatial resolution of the representation, thereby making the representation robust to small shifts and geometric distortions, similarly to "complex cells" in standard models of the visual cortex. ConvNets are generally composed of one to three stages, capped by a classifier composed of one or two additional layers. A gradient-based supervised training procedure updates every single filter in every filter bank in every layer so as to minimizes a loss function.

In traditional ConvNets, the output of the last stage is fed to a classifier. In the present work the outputs of all the stages are fed to the classifier. This allows the classifier to use, not just high-level features, which tend to be global, invariant, but with little precise details, but also pooled low-level features, which tend to be more local, less invariant, and more accurately encode local motifs.

The ConvNet was implemented using the EBLearn C++ open-source package [1] [10]. One advantage of ConvNets is that they can be run at very high speed on low-cost, small-

[1] http://eblearn.sf.net

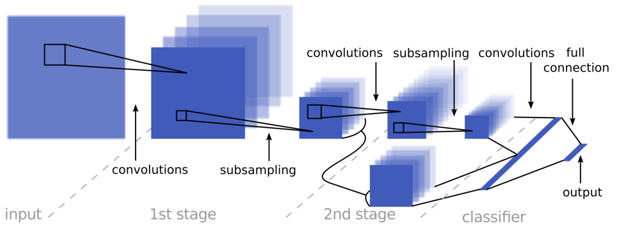

Fig. 2. A 2-stage ConvNet architecture. The input is processed in a feed-forward manner through two stage of convolutions and subsampling, and finally classified with a linear classifier. The output of the 1st stage is also fed directly to the classifier as higher-resolution features.

form-factor parallel hardware based on FPGAs or GPUs. Embedded systems based on FPGAs can run large ConvNets in real time [11], opening the possibility of performing multiple vision tasks simultaneously with a common infrastructure.

The ConvNet was trained with full supervision on the color images of the GTSRB dataset and reached 98.97% accuracy on the phase 1 test set. After the end of phase 1, additional experiments with grayscale images established a new record accuracy of 99.17%.

II. Architecture

The architecture used in the present work departs from traditional ConvNets by the type of non-linearities used, by the use of connections that skip layers, and by the use of pooling layers with different subsampling ratios for the connections that skip layers and for those that do not.

A. *Multi-Scale Features*

Usual ConvNets are organized in strict feed-forward layered architectures in which the output of one layer is fed only to the layer above. Instead, the output of the first stage is branched out and fed to the classifier, in addition to the output of the second stage (Fig. 2). Contrary to [12], we use the output of the first stage after pooling/subsampling rather than before. Additionally, applying a second subsampling stage on the branched output yielded higher accuracies than with just one. Therefore the branched 1^{st}-stage outputs are more subsampled than in traditional ConvNets but overall undergoes the same amount of subsampling (4x4 here) than the 2^{nd}-stage outputs. The motivation for combining representation from multiple stages in the classifier is to provide different scales of receptive fields to the classifier. In the case of 2 stages of features, the second stage extracts "global" and invariant shapes and structures, while the first stage extracts "local" motifs with more precise details. We demonstrate the accuracy gain of using such layer-skipping connections in section III-B.

B. *Non-Linearities*

In traditional ConvNets, the non-linear layer simply consists in a pointwise sigmoid function, such as $tanh()$. However more sophisticated non-linear modules have recently been shown to yield higher accuracy, particularly in the small training set size regime [9]. These new non-linear modules include a pointwise function of the type $|\tanh()|$ (rectified sigmoid), followed by a subtractive local normalization, and a divisive local normalization. The local normalization operations are inspired by visual neuroscience models [13], [14]. The subtractive normalization operation for a given site x_{ijk} computes: $v_{ijk} = x_{ijk} - \sum_{ipq} w_{pq}.x_{i,j+p,k+q}$, where w_{pq} is a Gaussian weighting window normalized so that $\sum_{ipq} w_{pq} = 1$. The divisive normalization computes $y_{ijk} = v_{ijk}/\max(c, \sigma_{jk})$ where $\sigma_{jk} = (\sum_{ipq} w_{pq}.v^2_{i,j+p,k+q})^{1/2}$. For each sample, the constant c is set to the $mean(\sigma_{jk})$ in the experiments. The denominator is the weighted standard deviation of all features over a spatial neighborhood.

Finding the optimal architecture of a ConvNet for a given task remains mainly empirical. In the next section, we investigate multiple architecture choices.

III. Experiments

A. *Data Preparation*

1) Validation: Traffic sign examples in the GTSRB dataset were extracted from 1-second video sequences, i.e. each real-world instance yields 30 samples with usually increasing resolution as the camera is approaching the sign. One has to be careful to separate each track to build a meaningful validation set. Mixing all images at random and subsequently separating into training and validation will result in very similar sets, and will not accurately predict performance on the unseen test set. We extract 1 track at random per class for validation, yielding 1,290 samples for validation and 25,350 for training. Some experiments will further be reported using this validation set. While reporting cross-validated results would be ideal, training time currently prohibits running many experiments. We will however report cross-validated results in the future.

2) Pre-processing: All images are down-sampled or up-sampled to 32x32 (dataset samples sizes vary from 15x15 to 250x250) and converted to YUV space. The Y channel is then preprocessed with global and local contrast normalization while U and V channels are left unchanged. Global normalization first centers each image around its mean value, whereas local normalization (see II-B) emphasizes edges.

	Size	Validation Error
Original dataset	25,350	1.31783%
Jittered dataset	126,750	1.08527%

TABLE I
Performance difference between training on regular training set and jittered training set.

Additionally, we build a jittered dataset by adding 5 transformed versions of the original training set, yielding 126,750 samples in total. Samples are randomly perturbed in position ([-2,2] pixels), in scale ([.9,1.1] ratio) and rotation ([-15,+15] degrees). ConvNets architectures have built-in invariance to small translations, scaling and rotations. When a dataset does not naturally contain those deformations, adding them synthetically will yield more robust learning to potential deformations in the test set. We demonstrate the error rate gain on the validation set in table I. Other realistic perturbations would probably also increase robustness such as other affine transformations, brightness, contrast and blur.

B. *Network Architecture*

[15] showed architecture choice is crucial in a number of state-of-the-art methods including ConvNets. They also demonstrate that randomly initialized architectures performance correlates with trained architecture performance when cross-validated. Using this idea, we can empirically search for an optimal architecture very quickly, by bypassing the time-consuming feature extractor training. We first extract features from a set of randomly initialized architectures with different capacities. We then train the top classifier using these features as input, again with a range of different capacities. In Fig. 3, we train on the original (non jittered) training set and evaluate against the validation set the following architecture parameters:

- Number of features at each stage: 108-108, 108-200, 38-64, 50-100, 72-128, 22-38 (the left and right numbers are the number of features at the first and second stages respectively). Each convolution connection table has a density of approximately 70-80%, i.e. 108-8640, 108-16000, 38-1664, 50-4000, 72-7680, 22-684 in number of convolution kernels per stage respectively.
- Single or multi-scale features. The single-scale architecture (SS) uses only 2nd stage features as input to the classifier while multi-scale architecture feed either the (subsampled) output of the first stage (MS).
- Classifier architecture: single layer (fully connected) classifier or 2-layer (fully connected) classifier with the following number of hidden units: 10, 20, 50, 100, 200, 400.
- Color: we either use YUV channels or Y only.
- Different learning rates and regularization values.

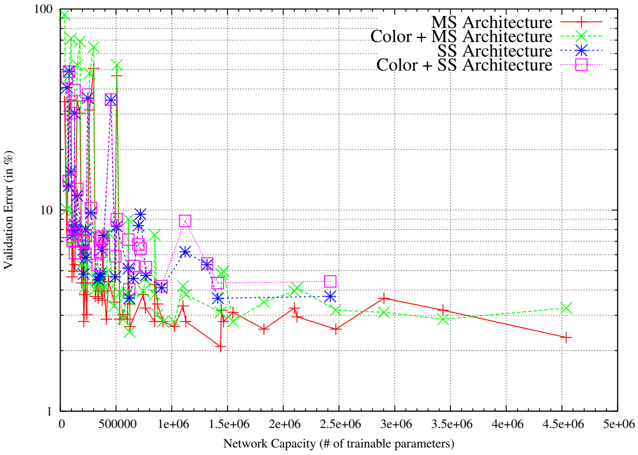

Fig. 3. Validation error rate of random-weights architectures trained on the non-jittered dataset. The horizontal axis is the number of trainable parameters in the network. For readability, we group all architectures described in III-B according to 2 variables: color and architecture (single or multi-scale).

We report in Fig. 3 the best performing networks among all learning rates and regularization factors. MS architecture outperforms SS architecture most of the time. Surprisingly, using no color is often better than using color. The most successful architecture uses MS without color and has 1,437,791 trainable parameters. It uses the 108-200 feature sizes and the 2-layer classifier with 100 hidden units. Note that this systematic experiment was not conducted for the competition's first phase. The architectures used at that time were the 38-64 and 108-108 features with MS architecture, the single layer classifier and were using color. Here, a deeper and wider (100 hidden units) classifier outperformed the single-layer fully connected classifier. Additionally, the optimal feature sizes among the tested ones are 108-200 for the 1st and 2nd stage, followed by 108-108. We evaluate this random-feature ConvNet on the official test set in section IV-B.

We then train in a supervised manner the entire network (including feature extraction stages) with the top performing architectures (108-200 and 108-108 with 2-stage classifiers with 100 and 50 hidden units, without color and with the MS feature architecture) on the jittered training set. Combining 108-108 features, 50 hidden units and no color performed best on the validation set, reaching 0.31% error. We finally evaluate this trained network against the official test set. Results are shown in section IV-B.

IV. Results

We report and analyze results both during and after competition's phase I.

A. *GTSRB Competition Phase I*

#	Team	Method	Accuracy
197	IDSIA	cnn_hog3	98.98%
196	IDSIA	cnn_cnn_hog3	98.98%
178	**sermanet**	**EBLearn 2LConvNet ms 108 feats**	**98.97%**
195	IDSIA	cnn_cnn_hog3_haar	98.97%
187	**sermanet**	**EBLearn 2LConvNet ms 108 + val**	**98.89%**
199	INI-RTCV	Human performance	98.81%
170	IDSIA	CNN(IMG)_MLP(HOG3)	98.79%
177	IDSIA	CNN(IMG)_MLP(HOG3) _MLP(HAAR)	98.72%
26	**sermanet**	**EBLearn 2-layer ConvNet ms**	**98.59%**
193	IDSIA	CNN 7HL norm	98.46%
198	**sermanet**	**EBLearn 2-layer ConvNet ms reg**	**98.41%**
185	**sermanet**	**EBLearn 2L CNN ms + validation**	**98.41%**
27	**sermanet**	**EBLearn 2-layer ConvNet ss**	**98.20%**
191	IDSIA	CNN 7HL	98.10%
183	Radu.Timofte@VISICS	IKSVM+LDA+HOGs	97.88%
166	IDSIA	CNN 6HL	97.56%
184	Radu.Timofte@VISICS	CS+I+HOGs	97.35%

TABLE II

Top 17 test set accuracy results during competition's first phase.

We reached 98.97% accuracy during competition's first phase by submitting 6 results. We reproduce in table II results as published on the GTSRB website [2]. It is interesting to note that the top 13 systems all use ConvNets with at least 98.10% accuracy and that human performance (98.81%) is outperformed by 5 of these. Again, the MS architecture (#26) outperformed the SS architecture (#27).

[2] http://benchmark.ini.rub.de/index.php?section=results

We now describe each of the 6 systems submitted during Phase I. All systems have 2 stages of feature extraction ("2-layer") followed by a single fully connected linear classifier and use color information. The poorest performing network (#27) uses the traditional single-scale ("ss") feature architecture while other networks use multi-scale ("ms") features by feeding their first and second stage features to the classifier.

- (#178) 2-layer ConvNet ms 108 feats (98.97%): This network uses 108 features at the first stage (100 connected to grayscale Y input channel and 8 to U and V color channels) and 108 features at the second stage.
- (#187) 2-layer ConvNet ms 108 + val (98.89%): Same network as #178, with additional training on validation data.
- (#26) 2-layer ConvNet ms (98.59%): This network uses 38 features at the first stage (30 for grayscale and 8 for U and V) and 64 at the second stage.
- (#198) 2-layer ConvNet ms reg (98.41%): Same network as #26 with stronger regularization.
- (#185) 2-layer ConvNet ms + validation (98.41%): Same network as #26 with additional training on validation data.
- (#27) 2-layer ConvNet ss (98.20%): Same network as #26 except it only uses second stage features as input to the classifier.

B. Post Phase I results

Similarly to phase I, we evaluated only a few newly trained network on the test set to avoid overfitting. We establish a new record of 99.17% accuracy (see Table III) by increasing the classifier's capacity and depth (2-layer classifier with 100 hidden units instead of the single-layer classifier) and by ignoring color information (see corresponding convolutional filters in Fig 4).

#	Team	Method	Accuracy
	sermanet	**EBLearn 2LConvNet ms 108-108 + 100-feats CF classifier + No color**	**99.17%**
197	IDSIA	cnn_hog3	98.98%
196	IDSIA	cnn_cnn_hog3	98.98%
178	**sermanet**	**EBLearn 2LConvNet ms 108-108**	**98.97%**
	sermanet	**EBLearn 2LConvNet ms 108-200 + 100-feats CF classifier + No color**	**98.85%**
	sermanet	**EBLearn 2LConvNet ms 108-200 + 100-feats CF classifier + No color + Ramdom features + No jitter**	**97.33%**

TABLE III
POST PHASE I NETWORKS EVALUATED AGAINST THE OFFICIAL TEST SET BREAK THE PREVIOUS 98.98% ACCURACY RECORD WITH 99.17%.

We also evaluate the best ConvNet with random features in section III-B (108-200 random features by training the 2-layer classifier with 100 hidden units only) and obtain 97.33% accuracy on the test set (see convolutional filters in Fig 4). Recall that this network was trained on the non-jittered dataset and could thus perform even better. The exact same architecture with trained features reaches 98.85% accuracy only while a network with a smaller second stage (108 instead of 200) reached 99.17%. Comparing random and trained convolutional filters (Fig 4), we observe that 2^{nd} stage trained filters mostly contain flat surfaces with sparse positive or negative responses. While these filters are quite different from random filters, the 1^{st} stage trained filters are not. The specificity of the learned 2^{nd} stage filters may explain why more of them are required with random features, thus increasing the chances of containing appropriates features. A smaller 2^{nd} stage however may be easier to train with less diluted gradients and more optimal in terms of capacity. We therefore infer that after finding an optimal architecture with random features, one should try smaller stages (beyond the 1^{st} stage) with respect to the best random architecture, during full supervision.

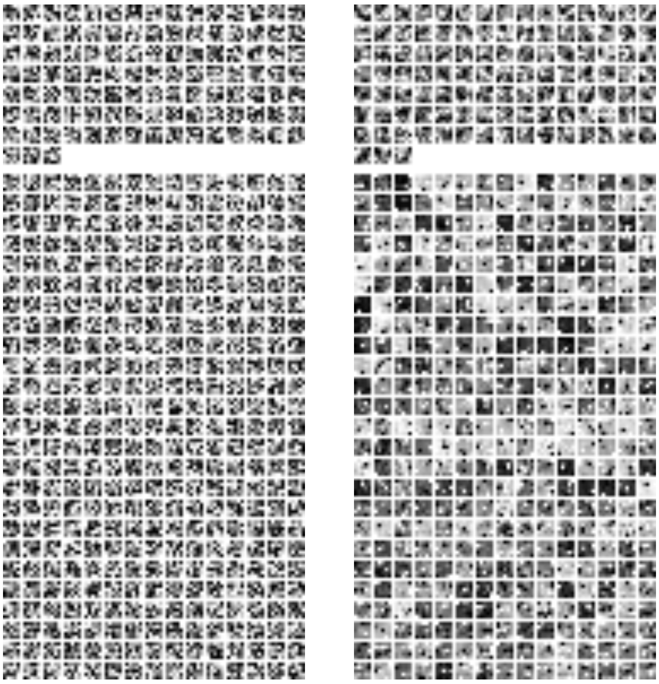

Fig. 4. 5x5 convolution filters for the first stage (top) and second stage (bottom). **Left:** Random-features ConvNet reaching 97.33% accuracy (see Table III), with 108 and 16000 filters for stages 1 and 2 respecitvely. **Right:** Fully trained ConvNet reaching 99.17% accuracy, with 108 and 8640 filters for stages 1 and 2.

Finally, we analyze the remaining test set errors of the 99.17% accuracy network by displaying each 104 sample's input channels, i.e. normalized Y intensity and non-normalized U and V colors in Fig 5. This particular network uses grayscale only (left column) and could have clearly benefited from color information for the worst errors (top), where an arrow is hardly visible in grayscale but clearly is in color channels. We however showed that non-normalized color yielded overall worse performance. Still, future experiments with normalized color channels may reveal that color edges may be more informative than raw color. Thus, we hypothesize raw UV color may not be an optimal input format, or additionally that the wide array lighting conditions (see Fig 1) makes color in general unreliable. Additionally, a few errors seem to arise from motion blur and low contrast, which may be improved by extending jitter to additional real-world deformations. Remaining errors, are likely due to physically degraded road-signs and too low-resolution inputs for which classification is impossible with a single image instance.

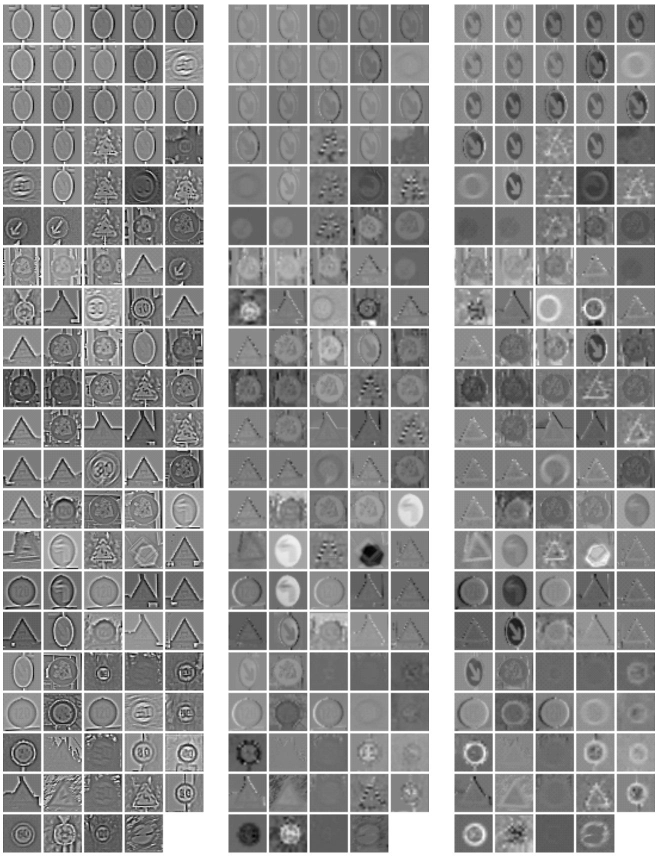

Fig. 5. Remaining 104 errors out of 12569 samples of the test set with the 99.17% accuracy ConvNet (using the left grayscale channel only). The samples are sorted from top to bottom by energy error (top samples are the furthest away from their target vector). The top worst answers clearly could have benefited the use of the color channels. Images are resized to 32x32 and preprocessed to YUV color scheme. **Left:** The normalized Y channel (intensity). **Middle:** U color channel (non-normalized). **Right:** V color channel (non-normalized).

V. Summary and Future Work

We presented a Convolutional Network architecture with state-of-the-art results on the GTSRB traffic sign dataset implemented with the EBLearn open-source library [10]. During phase I of the GTSRB competition, this architecture reached 98.97% accuracy using 32x32 colored data while the top score was obtained by the IDSIA team with 98.98%. The first 13 top scores were obtained with ConvNet architectures, 5 of which were above human performance (98.81%). Subsequently to this first phase, we establish a new record of 99.17% accuracy by increasing our network's capacity and depth and ignoring color information. This somewhat contradicts prior results with other methods suggesting that colorless recognition, while effective, was less accurate [16]. We also demonstrated the benefits of multi-scale features in multiple experiments. Additionally, we report very competitive results (97.33%) using random features while searching for an optimal architecture rapidly. We suggest that feature stages past the 1^{st} stage should be smaller than the optimal random architecture stages.

Future work should investigate the impact of unsupervised pre-training of feature extracting stages, particularly with a higher number of features at each stage, which can be more easily learned than with a purely supervised fashion. The impact of input resolution should be studied to improve both accuracy and processing speed. More diverse training set deformations can also be investigated such as brightness, contrast, shear and blur perturbations to address the numerous real-world deformations highlighted in Fig. 1. Additionally, widening the second feature extraction stage while sparsifying its connection table might allow using a lighter classifier, thus reducing computation. Finally, ensemble processing with multiple networks might further enhance accuracy. Taking votes from colored and non-colored networks can probably alleviate both situations where color may be used or not. By visualizing remaining errors, we also suspect that normalized color channels may be more informative than raw color.

References

[1] Stallkamp, J, Schlipsing, M, Salmen, J, and Igel, C. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In *International Joint Conference on Neural Networks*, 2011.

[2] Torresen, J, Bakke, J, and Sekanina, L. Efficient recognition of speed limit signs. In *Intelligent Transportation Systems, 2004. Proceedings. The 7th International IEEE Conference on*, pages 652 – 656, 2004.

[3] Nguwi, Y.-Y and Kouzani, A. Detection and classification of road signs in natural environments. *Neural Computing and Applications*, 17:265–289, 2008. 10.1007/s00521-007-0120-z.

[4] Lafuente-Arroyo, S, Gil-Jimenez, P, Maldonado-Bascon, R, Lopez-Ferreras, F, and Maldonado-Bascon, S. Traffic sign shape classification evaluation i: Svm using distance to borders. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 557 – 562, 2005.

[5] de la Escalera, A, Moreno, L, Salichs, M, and Armingol, J. Road traffic sign detection and classification. *Industrial Electronics, IEEE Transactions on*, 44(6):848 –859, December 1997.

[6] Barnes, N, Zelinsky, A, and Fletcher, L. Real-time Speed Sign Detection using the Radial Symmetry Detector. *IEEE Transactions on Intelligent Transport Systems*, 2008.

[7] Keller, C, Sprunk, C, Bahlmann, C, Giebel, J, and Baratoff, G. Real-time recognition of u.s. speed signs. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 518 –523, 2008.

[8] LeCun, Y, Bottou, L, Bengio, Y, and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998.

[9] Jarrett, K, Kavukcuoglu, K, Ranzato, M, and LeCun, Y. What is the best multi-stage architecture for object recognition? In *Proc. International Conference on Computer Vision (ICCV'09)*. IEEE, 2009.

[10] Sermanet, P, Kavukcuoglu, K, and LeCun, Y. Eblearn: Open-source energy-based learning in c++. In *Proc. International Conference on Tools with Artificial Intelligence (ICTAI'09)*. IEEE, 2009.

[11] Farabet, C, Martini, B, Akselrod, P, Talay, S, LeCun, Y, and Culurciello, E. Hardware accelerated convolutional neural networks for synthetic vision systems. In *Proc. International Symposium on Circuits and Systems (ISCAS'10)*. IEEE, 2010.

[12] Fan, J, Xu, W, Wu, Y, and Gong, Y. Human tracking using convolutional neural networks. *Neural Networks, IEEE Transactions on*, 21(10):1610 –1623, 2010.

[13] Lyu, S and Simoncelli, E. P. Nonlinear image representation using divisive normalization. In *Proc. Computer Vision and Pattern Recognition*, pages 1–8. IEEE Computer Society, Jun 23-28 2008.

[14] Pinto, N, Cox, D. D, and DiCarlo, J. J. Why is real-world visual object recognition hard? *PLoS Comput Biol*, 4(1):e27, 01 2008.

[15] Saxe, A, Koh, P. W, Chen, Z, Bhand, M, Suresh, B, and Ng, A. In *Adv. Neural Information Processing Systems (NIPS*10), Workshop on Deep Learning and Unsupervised Feature Learning.*, 2010.

[16] Paclk, P and Novovicov, J. Road sign classification without color information. In *Proceedings of the 6th Conference of Advanced School of Imaging and Computing*, 2000.