Traffic sign classification with batch and on-line linear support vector machines

Rok Mandeljc, Domen Tabernik, Matej Kristan, Danijel Skočaj

Visual Cognitive Systems Laboratory, Faculty of Computer and Information Science, University of Ljubljana
E-mail: rok.mandeljc@fri.uni-lj.si

Abstract

This paper presents a comprehensive benchmark of several feature types and colorspace representations on the task of traffic sign classification. We focus on linear Support Vector Machine classifiers, and test several multi-class formulations, as well as a formulation that allows on-line training and updates. Experiments on two standard traffic sign classification datasets show that despite their relative simplicity, these classifiers offer competitive performance, and ultimately allow design of a flexible classification system in the context of application for automatic maintenance of traffic signalization inventory.

1 Introduction

The problem of traffic sign recognition (TSR) has lately received a considerable attention from the computer-vision community, as it represents the key component in applications such as traffic sign inventory maintenance, driver-assistance systems, and autonomous vehicles [10].

This paper is a part of our research in the context of an automatic traffic sign inventory maintenance project, which involves cataloguing the traffic signalization along the major roads. Such systems have somewhat different requirements than the TSR typically found in driver-assistance systems; while the latter focus on accurate detection and recognition of a relatively small subset of relevant signs (e.g., speed limit or warning signs), the former need to be able to detect and recognize a much broader range of traffic signalization. Therefore, we require a flexible classification system that scales with the number of categories, and can be (re)trained in an on-line manner.

Here, we present a comprehensive benchmark of several different features and color representations on the task of traffic sign classification, which typically represents the second part of the TSR pipeline, and assumes that the bounding boxes, corresponding to traffic signs, have already been detected and localized in the image. We focus on linear Support Vector Machine (SVM) classifiers, and test several multi-class formulations, as well as a formulation that allows on-line training and updates.

The rest of the paper is structured as follows. In Section 2, we present a brief overview of the related work, followed by description of features and classifiers in Sections 3 and 4. Results are presented and discussed in Section 5, while Section 6 wraps up the paper.

2 Related work

The interest in the problem of traffic sign classification was recently reinforced by the German Traffic Sign Recognition Benchmark (GTSRB) challenge [15] and the public availability of its dataset, which consists of 51 840 cropped images of 43 classes of German road signs. The two top-performing approaches on this challenge were using a committee of Convolutional Neural Networks [2] and a multi-scale CNN [14], trained on contrast-normalized grayscale patches, and achieved recognition rates of 99.46 % and 98.31 %, respectively. The third-ranked entry used the dataset-provided Histograms of Oriented Gradients (HOG) features and a Random Forests classifier, with 97.20 % recognition rate [18]. Wang *et al.* [17] improved these results even further to 99.52 %, mainly by introducing a class-based perspective adjustment step before the classification.

Mathias *et al.* [9] performed a comprehensive comparison of several classifiers and dimension-reduced features, obtained by various projection methods when applied on top of intensity values from 28×28 patches, the pyramids of HOGs, and the dataset-provided HOG features. Their best achieved results for a single projection/classifier combination on the GTSRB and Belgium Traffic Sign Classification (BTSC) datasets are 97.65 % and 97.83 %, respectively; these results are further improved by merging the projections of all base features (98.53 % and 98.32 %, respectively). The BTSC dataset is an earlier work of the same authors [16], and encompasses 4591 training and 2434 testing images, cropped around instances of 62 different classes of Belgian traffic signs.

Pei *et al.* [13] compared the results of logistic regression when applied on top of the original features (raw intensities, GIST, or HOG) and the ones obtained by a supervised low-rank matrix recovery model, showing that the latter offers better performance; their best recognition rate on GTSRB dataset, 97.51 %, was achieved using HOG as the base feature. Fan *et al.* [6] use a hierarchical sparse representation for traffic sign recognition. Combined with the dataset-provided HOG features, the proposed scheme achieves 97.36 % recognition rate on the GTSRB dataset, and, using the raw intensities extracted from 40×40 patches, 92.56 % on the BTSC dataset.

3 Features

We evaluate several feature types that are commonly used for object detection and classification. Before feature extraction, each patch is rescaled to 40×40 pixel size, using bilinear interpolation. We also evaluate the effect of different color representations, and thus each feature is evaluated in several color spaces: grayscale, HSV, LAB, LUV, RGB, and YCbCr. Multi-channel features are obtained by separate extraction from each channel, and subsequent concatenation into a single feature vector.

Histograms of Oriented Gradients (HOG). We use the VLFeat implementation of the UoCTTI HOG variant [7] that includes both signed and unsigned gradient information. The number of orientations is left at the default value of 9, and cell size was set to 5. The resulting feature vector has 1984 elements for single-, and 5952 elements for three-channel images.

Local Binary Patterns (LBP). We obtain the histograms of LBP [11] using the VLFeat implementation. The input 40×40 patches are divided into 5×5 cells, and a histogram of 58 quantized patterns is computed for each cell. The obtained histograms are unrolled into a single feature vector, which has 3712 and 11 136 elements for single- and three-channel images, respectively.

GIST descriptor. For GIST descriptor [12], we use the publicly-available Matlab implementation, with default parameters (8 orientations/scale, 4 blocks, and 4 cycles per image for the Gaussian pre-filter). The resulting feature vectors have 512 elements per image channel.

Integral Channel Features (ICF). Integral Channel Features [4] consist of six gradient orientation channels, a single gradient magnitude channel, and one or three color channels. When applied to input 40×40 patches with shrink factor 2 (the rest of parameters are left at default), the resulting feature vectors are 3200-dimensional (single-channel) or 4000-dimensional (three-channel).

4 Classifiers

In this paper, we focus on linear support vector machine classifiers; we use several different multi-class formulations of the SVM, as well as an online formulation.

One-vs-rest classifier. This multi-class formulation comprises an array of binary classifiers, one for each class. Each classifier is trained in one-vs-all manner, with the instances of the corresponding class being positive, and the instances of all other classes being negative samples. For each classifier, we independently optimize the regularization parameter C via three-fold cross validation on the training set. The output scores of the classifiers are calibrated into posterior probabilities using an improved version of Platt's probabilistics calibration [8]; during prediction, the sample is proclaimed as belonging to the class with the highest estimated probability. For classifiers, linear SVM from the LIBLINEAR [5] package is used, with the default solver, and without bias ($B = 0$).

Multi-class tree. This tree-like classifier structure is an extension of regular ("flat") one-vs-rest classifier; the sign classes are (manually) grouped into super-classes with their own nodes in the tree, based on similar attributes (e.g., blue round signs, triangular signs, etc.), and fine-grained classification into final classes is preceded by coarse classification into super-classes. Each branch of the tree ends with a leaf node, corresponding to a classifier that represents a single, final class. All classifiers are again trained in one-vs-all fashion; however, each classifier is trained as one-vs-all with respect to its sibling nodes, and passes on to its child nodes only its positive samples. During prediction, all nodes at the same level are used to classify the sample, and the branch with the highest estimated probability is used for further classification, until the leaf-node is reached. We use two dataset-specific tree topologies, which are illustrated in Figs. 1a and 1b.

Crammer&Singer multi-class SVM. We also evaluate the Crammer&Singer multi-class SVM formulation [3], which is also provided by the LIBLINEAR package. In contrast to previous two formulations, this one directly optimizes the multi-class problem and does not require decomposition into binary ones. Again, the default solver is used, and bias is set to zero.

LaRank online linear SVM. LaRank [1] optimizes the same problem as the Crammer&Singer formulation, but does so in an online manner, meaning that the training can be performed one sample at the time. In practice, an equivalent of batch training is achieved by iterating over training set multiple times, and permuting the order of samples at the beginning of each training epoch. We evaluate the classifier after the first and after the tenth training epoch. As this is an online method, we also evaluate it in an online test scenario, where after predicting the label of a test sample, we use its true label to update the classifier before classifying next sample.

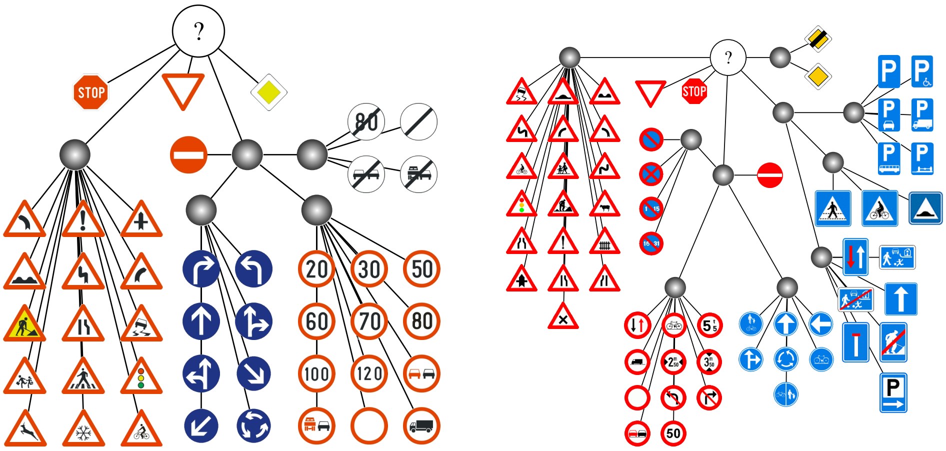
5 Results and discussion

The benchmark was conducted on two standard traffic sign recognition datasets — the German Traffic Sign Recognition Benchmark (GTSRB) dataset and the Belgium Traffic Sign Classification (BTSC) dataset. All experiments were performed on a workstation with a six-core Intel i7-5930K processor (3.50 GHz) and 32 GB of RAM, running 64-bit linux and MATLAB R2015a.

The results for all combinations of used features, colorspaces, and classifiers, are summarized in Table 1, with bold entries denoting the best-performing combination for the given classifier type.

On the GTSRB dataset, the best classification performance is obtained by combining LBP features and RGB colorspace. This trend can be observed across all tested classifier types. The best result in the off-line test setting is obtained by the classifier tree (97.81 %), which also outperforms all individual combinations of the projection methods and sparse classifiers reported in [9], and offers competitive performance even compared to the top results on the GTSRB dataset. The other classifiers offer slightly worse, but still comparable, performance.

On the BTSC dataset, the best performing combination appears to be HOG features and LAB colorspace.



(a) *Tree structure for GTSRB dataset.* **(b)** *Tree structure for BTSC dataset.*

Figure 1: *Visualization of tree structures, as well as all traffic sign classes that are present in each dataset.*

Table 1: *Benchmark results on both GTSRB (43 classes) and BTSC (62 classes) dataset. All values denote the percentage of correctly classified samples (i.e., classification accuracy).*

classifier	colorspace	GTSRB dataset				BTSC dataset			
		HOG	LBP	GIST	ICF	HOG	LBP	GIST	ICF
one-vs-rest	gray	96.83	96.71	96.62	95.93	97.63	95.22	94.87	96.96
	HSV	96.10	95.75	96.06	87.09	97.63	96.69	96.37	95.50
	LAB	96.95	96.75	97.00	96.08	**98.18**	96.76	96.45	96.72
	LUV	97.01	96.93	96.52	94.60	97.99	97.00	96.05	97.08
	RGB	97.24	**97.60**	97.28	95.76	97.91	97.28	97.51	96.88
	YCbCr	96.98	96.95	96.76	96.18	97.71	96.92	96.25	97.20
tree-structure	gray	97.02	96.67	96.77	96.16	97.28	94.67	93.01	96.21
	HSV	96.34	95.98	96.14	87.59	97.83	96.53	95.82	95.78
	LAB	97.05	97.05	97.21	96.33	**98.26**	96.61	96.25	97.08
	LUV	97.35	97.14	97.04	94.86	98.03	97.04	95.38	96.76
	RGB	97.37	**97.81**	97.59	95.97	98.22	97.16	96.57	97.12
	YCbCr	97.10	97.02	97.19	96.47	97.59	96.96	96.21	97.40
Cram. & Sing.	gray	97.02	96.64	96.67	96.07	97.75	94.75	95.54	96.96
	HSV	96.06	95.82	96.25	87.55	97.55	95.42	96.41	95.78
	LAB	96.96	96.83	97.11	96.14	**97.83**	95.42	96.25	97.12
	LUV	97.04	96.82	96.75	94.67	97.71	95.34	95.90	97.00
	RGB	97.32	**97.59**	97.40	95.79	**97.83**	96.33	97.24	97.00
	YCbCr	97.05	96.81	97.21	96.18	97.32	95.46	96.37	97.32
LaRank (1 ep.)	gray	96.77	96.52	96.71	95.80	97.71	94.24	94.48	96.76
	HSV	96.06	95.53	95.53	85.10	97.67	94.83	95.82	95.50
	LAB	96.73	96.79	96.74	95.98	97.67	94.75	96.37	96.37
	LUV	96.79	96.72	96.10	94.34	97.67	94.79	96.05	96.96
	RGB	97.14	**97.68**	97.33	95.56	**97.91**	96.25	97.32	97.08
	YCbCr	96.76	96.74	96.78	96.06	97.32	94.59	96.13	96.84
LaRank (10 ep.)	gray	96.88	96.57	96.68	95.96	97.63	94.40	94.44	96.92
	HSV	96.06	95.82	95.64	85.83	97.63	95.15	96.01	95.74
	LAB	96.67	96.76	96.82	96.00	**97.91**	95.42	95.86	97.04
	LUV	96.84	96.76	96.39	94.37	97.67	95.38	95.58	97.04
	RGB	97.13	**97.66**	97.34	95.72	97.87	96.41	97.28	97.20
	YCbCr	96.75	96.62	96.88	96.03	97.32	95.50	96.33	97.51
LaRank (online)	gray	98.69	98.48	98.70	98.34	99.05	97.20	97.91	98.70
	HSV	98.19	98.11	98.13	90.91	98.93	97.36	98.26	97.83
	LAB	98.75	98.60	98.80	98.26	**99.09**	97.95	98.54	98.62
	LUV	98.65	98.54	98.71	96.90	98.86	97.63	98.34	98.50
	RGB	99.09	99.10	**99.19**	98.02	99.05	98.26	98.74	98.66
	YCbCr	98.76	98.60	98.94	98.38	98.93	97.75	98.62	98.70

Again, the best result in off-line setting is obtained by the classifier tree structure, with classification accuracy of 98.26 %. It not only again outperforms all individual combinations from [9], but also comes very close to their merged variants.

Overall, we can see that using multi-channel image representation instead of a grayscale image tends to improve the classification performance. Perhaps interestingly, an exception to this trend is the HSV colorspace, which consistently yields worse classification accuracy.

Among the classifiers themselves, the hand-crafted classifier tree structures (Figs. 1a and 1b) tend to slightly outperform "flat" one-vs-rest classifier structures, as well as the Crammer&Singer multi-class SVM formulation. As shown by results, the LaRank SVM reaches almost-optimal classification accuracy after a single training epoch, and not much is gained by repeating the training for ten epochs. But more importantly, its classification accuracy is close to that of the other tested classifiers, which makes this online method a very attractive choice over the competing offline classifiers.

Online scenario. The results of the on-line test scenario with LaRank consistently show further improvement in classification results. This is perhaps unsurprising, given the nature of the on-line scenario, but on the other hand accentuates the advantage of an online classifier, namely the ability to adapt to the data in on-line manner. In both datasets, a large portion of difficult test samples (i.e., the ones that are misclassified by most classifiers) belongs to few traffic sign instances with poses that were not captured by the training set. In the off-line (batch) test scenario, all such samples end up being misclassified. On the other hand, updating the classifier with the sample after the prediction is made (online scenario) increases the chances for correct classification of other samples that belong to the same instance.

6 Conclusion

We presented results of a comprehensive benchmark of several feature types and colorspace representations on the task of traffic sign classification, as well as several multi-class formulations of a linear SVM. We have shown that such combinations offer competitive performance, even when compared to state of the art. The most important finding is that the on-line LaRank SVM performs as well as its batch counterparts, but also allows us to update the model as the new data arrives. The on-line scenario assumes that the true label of the sample is known, for example by asking a human operator (application with human in the loop); however, as the practical aspects and considerations of such application are beyond the scope of this paper, we plan to address that in the course of our future work.

Acknowledgments

This work was supported by the Slovenian Research Agency, via the Research Programme ARRS P2-0214 and the Research Project ARRS L2-6765.

References

[1] A. Bordes, L. Bottou, P. Gallinari, and J. Weston. Solving multiclass support vector machines with larank. In *Proc. of ICML 2007*, pages 89–96, 2007.

[2] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32(0):333–338, 2012.

[3] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *J. Mach. Learn. Res.*, 2:265–292, 3 2002.

[4] P. Dollar, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *Proc. of BMVC 2009*, pages 91.1–91.11, 2009.

[5] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, 2008.

[6] Y. Fan, H. Sun, S. Zhou, and H. Zou. Hierarchical sparse representation for traffic sign recognition. In *Proc. of the CIAC 2013*, pages 653–660, 2013.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 32(9):1627–1645, 2010.

[8] H.-T. Lin, C.-J. Lin, and R. C. Weng. A note on platt's probabilistic outputs for support vector machines. *Machine Learning*, 68(3):267–276, 2007.

[9] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool. Traffic sign recognition — how far are we from the solution? In *Proc. of IJCNN 2013*, pages 1–8, 2013.

[10] A. Møgelmose, M. M. Trivedi, and T. B. Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *IEEE Trans. Intell. Transport. Syst.*, 13(4):1484–1497, 2012.

[11] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE TPAMI*, 24(7):971–987, 2002.

[12] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vis.*, 42(3):145–175, 2001.

[13] D. Pei, F. Sun, and H. Liu. Supervised low-rank matrix recovery for traffic sign recognition in image sequences. *IEEE Signal Processing Lett.*, 20(3):241–244, 2013.

[14] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *Proc. of IJCNN 2011*, pages 2809–2813, 2011.

[15] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32(0):323–332, 2012.

[16] R. Timofte, K. Zimmermann, and L. Van Gool. Multi-view traffic sign detection, recognition, and 3d localisation. In *Proc. of WACV 2009*, pages 1–8, 2009.

[17] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang. A hierarchical method for traffic sign classification with support vector machines. In *Proc. of IJCNN 2013*, pages 1–6, 2013.

[18] F. Zaklouta, B. Stanciulescu, and O. Hamdoun. Traffic sign classification using k-d trees and random forests. In *Proc. of IJCNN 2011*, pages 2151–2155, 2011.