22nd *Computer Vision Winter Workshop*
Nicole M. Artner, Ines Janusch, Walter G. Kropatsch (eds.)
Retz, Austria, February 6–8, 2017

Towards large-scale traffic sign detection and recognition

Peter Uršič, Domen Tabernik, Rok Mandeljc and Danijel Skočaj
Faculty of Computer and Information Science, University of Ljubljana
Večna pot 113, 1000 Ljubljana
peter.ursic@fri.uni-lj.si

Abstract. *Recognition of traffic signs is a well researched field in the computer vision community, with several commercial applications already available. However, a vast majority of existing approaches focuses on recognition of a relatively small number of traffic sign categories (about 50 or less). In this paper, we adopt a convolutional neural network (CNN) approach, i.e., the Faster R-CNN, to address the full pipeline of detection and recognition of more than 100 traffic sign categories, depicted in our novel dataset that was acquired on Slovenian roads. We report promising results on highly challenging traffic sign categories that have not yet been considered in previous works and we provide useful insights for CNN training.*

1. Introduction

The problems of traffic sign detection (TSD) and traffic sign recognition (TSR) have lately received a considerable attention from the computer-vision community, e.g., [17, 9, 41]. The purpose of TSD is to find the locations and sizes of traffic signs in natural scene images. The purpose of TSR is to classify the detected traffic signs into their specific categories. Several approaches focus solely on solving the former problem [11, 17, 21], and many focus on the latter [29, 9, 2], while some works, like our approach, attempt to solve both problems in a unified framework [5, 41, 7].

Recognition of traffic signs is the key component in driver-assistance systems [31] and autonomous vehicles [19], with many solutions already being deployed in real-world applications [1, 30]. However, a vast majority of existing approaches focuses on detection and recognition of a relatively small number of traffic sign categories (about 50 or even less), as the remaining categories are not highly important in automotive safety applications. On the other hand, verification of presence or absence of a larger number of traffic sign categories is crucial in road maintenance services [25, 4]. Our work is focused on the detection and recognition of all traffic signs, which would help eliminate the tedious manual verification in such tasks, and could also be useful in applications of autonomous vehicles; for example, to augment the navigation when GPS signal is poor or when the traffic signalization changes.

In this paper, we adopt a convolutional neural network (CNN) approach to address both stages of the recognition pipeline (TSD and TSR). We are tackling the problems of detection and recognition of more than 100 categories of traffic signs (Figure 1) on our novel challenging dataset, acquired on the roads of

Figure 1. The 123 traffic sign categories considered in our work. Top: 91 categories of the base set. Bottom: additional categories of the extended set. Note that for a certain information sign category individual instances can contain various text and numbers.

Slovenia. In particular, we employ the highly efficient Faster region-based convolutional neural network (Faster R-CNN) model [23], which demonstrated great accuracy and speed in the field of object recognition [23]. Although some previous works already employed CNN approaches in both stages of the pipeline to some extent [41], they have focused on a highly limited subset of traffic sign categories. On the other hand, approaches that considered a large set of categories [6] adhered to the traditional hand-crafted features and focused only on ideogram-based (non-text) traffic signs. In contrast, our convolutional approach is applied to a broad set of categories, where individual traffic sign instances are not only subject to changes in lighting conditions, scale, viewing angle, blur, and occlusions, but also to intra-category appearance variations. That is, several challenging traffic sign categories are considered, whose instances may be of different real-world sizes, aspect ratio, color, or may contain various text that significantly differs between individual instances (Figure 2).

Figure 2. Examples of challenging traffic sign categories in our dataset, demonstrating by-design within-category appearance variability of individual instances. First two rows: instances with relatively low appearance variability from categories of the base set. Last two rows: instances with high appearance variability from categories of the extended set.

The remainder of the paper is organized as follows. Section 2 provides an overview of the related work, the employed method for TSD and TSR is presented in Section 3, experimental results are discussed in Section 4, while Section 5 concludes the paper.

2. Related work

An enormous amount of literature exists on the topics of TSD and TSR, and several review papers are available [20, 8, 34]. In general, it is very difficult to decide which approach gives better overall results, mainly due to the lack of a standard publicly available benchmark dataset that would contain an extensive set of various traffic sign categories, as emphasized in several recent studies [34, 5]. Some authors evaluate their approaches on one of the many public datasets with a relatively limited number of traffic sign categories:

- The German Traffic Sign Detection Benchmark (GTSDB) [11] comprises 3 super-categories of traffic signs and is primarily intended for TSD.
- The German Traffic Sign Recognition Benchmark (GTSRB) [29] covers 43 categories and is intended for TSR.
- The Belgium Traffic Signs (BTS) dataset [32] is suitable for TSD as well as TSR and contains 62 different categories of traffic signs.
- The Mapping and Assessing the State of Traffic Infrastructure (MASTIF) datasets were acquired in the scope of a commercial road maintenance assessment service in Croatia [25]. The proposed set of 9 categories of traffic signs [33] was later extended to 31 categories [12].
- The Swedish Traffic Sign dataset [15] is suitable for experimentation with 10 categories.
- The Laboratory for Intelligent and Safe Automobiles (LISA) Dataset [20] contains 49 categories of traffic signs and was acquired on the roads in the United States.

To enrich the set of considered traffic signs, some approaches sample images from multiple datasets to perform the evaluation [16, 37]. On the other hand, a vast number of authors use their own private datasets [17, 24, 6, 21]. To the best of our knowledge, the largest set of categories was considered in the private dataset of [6], who distinguished between 131 categories of non-text traffic signs from the roads of United Kingdom.

Various methods have been applied for TSD and TSR. Traditionally hand-crafted features have been used, like histogram of oriented gradients (HOG) [35, 39, 6, 18, 5, 9, 12, 11], scale invariant feature transform (SIFT) [9], local binary patterns (LBP) [5], GIST [22], or integral channel features [18], whereas a wide range of machine learning methods have been employed, ranging from support vector machine (SVM) [6, 5, 38], logistic regression [22], and random forests [5, 38], to artificial neural networks in the form of an extreme learning machine (ELM) [12].

Recently, like the entire computer vision field, TSD and TSR have also been subject to CNN renaissance. A modern CNN approach that automatically extracts multi-scale features for TSD has been applied in [36]. In TSR, CNNs have been used to automatically learn feature representations and to perform the classification [3, 26, 14, 33]. In order to further improve the recognition accuracy, a combination of CNN and a Multilayer Perceptron was applied in [2], while an ensemble classifier consisting of several CNNs is proposed in [3, 14]. A method that uses CNN to learn features and then applies ELM as a classifier has been applied in [40], while [10] employed a deep network consisting of spatial transformer layers and a modified version of inception module. It has been shown in [28] that the performance of CNN on TSR outperforms the human performance on GTSRB. Both stages of the recognition pipeline were addressed using CNNs in [41]. They applied a fully convolutional network to obtain a heat map of the image, on which a region proposal algorithm was employed for TSD. Finally, a different CNN was then employed to classify the obtained regions.

3. Method

In this section we first present the data augmentation techniques that were applied to enrich the training set of images, then, a method employed for model learning and test-stage detection and recognition, Faster R-CNN [23], is briefly described.

3.1. Data augmentation

Since CNNs require a huge amount of images for learning, data augmentation is often performed to obtain additional learning samples [6, 41]. In this work we have created synthetic traffic sign instances by following one of the two considered approaches:

- Creation by distortion of *graphical template images*; a graphical template for every class was available and it was used to artificially generate various traffic sign appearances.

- Creation by modification of *segmented real-world training samples*; the traffic signs in our database (see Sec. 4.1) are annotated with tight bounding boxes, allowing to segment them from the training images.

In both cases, two classes of distortions were performed: (i) geometric/shape distortions (affine transformations along horizontal and vertical dimension, changes in scale), and (ii) appearance distortions (variations in brightness and contrast, motion blur, and variable lighting simulating partial shadowing). A number of generated traffic signs for both approaches are depicted in Figure 3.

In the approach using graphical template, about 200 samples were created from a single template for each category. Similarly, also in the second approach a similar number of synthetic instances were acquired by randomly sampling from a set of distorted training instances. Generated traffic sign samples were inserted into street-environment-like background images, acquired from the subset of the BTS dataset [32], which originally contained no other traffic signs. From at least one to at most five traffic signs were placed in non-overlapping manner on random locations onto each background image, avoiding the bottom central part where only the road is usually seen, which resulted in about 6000 new training images.

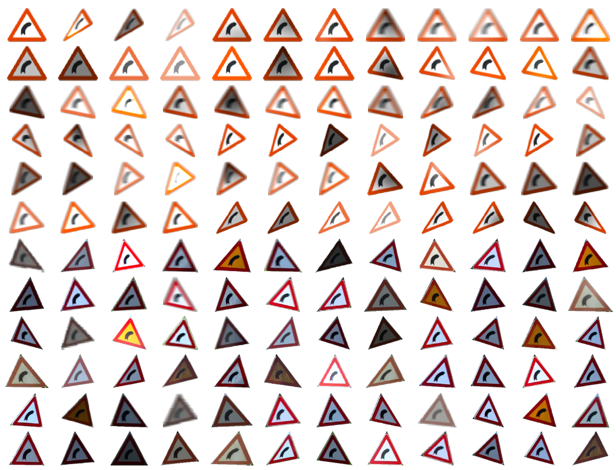

Figure 3. Examples of synthetic traffic sign instances. Top half: generated by performing distortions on a single graphical template (the first instance). Bottom half: generated by performing distortions on real-world samples.

3.2. Faster R-CNN

The Faster R-CNN network [23] is composed of two modules. The first module is a deep fully convolutional network, a so-called Region Proposal Network (RPN), that takes an input image and produces a set of rectangular object proposals, each with an objectness score. The second module is a region-based CNN, called Fast R-CNN, that classifies the proposed regions into the set of predefined categories. Fast R-CNN is highly efficient since it shares convolutions across individual proposals. It also performs bounding box regression to further refine the quality of the proposed regions. The entire system is a single unified network, in which RPN and Fast R-CNN are merged by sharing their convolutional features. Following the recently popular terminology of neural networks with attention mechanisms, the RPN module tells the Fast R-CNN module where to look (Figure 4).

Training a Faster R-CNN network is a 4-step optimization process that alternates between fine-tuning for the region proposal task and fine-tuning for the classification task, both being performed by backpropagation with stochastic gradient descent. In the first step, the RPN is initialized with an ImageNet-pre-trained model and then fine-tuned end-to-end. In the second step, a separate Fast R-CNN network, also initialized with the ImageNet-pre-trained model, is trained using the proposals generated by the step-1 RPN. At this point the two networks do not share convolutional layers. In the third step, the classification network is used to initialize RPN training, and then, keeping the convolutional layers fixed, only the layers unique to RPN are fine-tuned. Now the two networks share convolutional layers. Finally, keeping the convolutional layers fixed, the unique layers

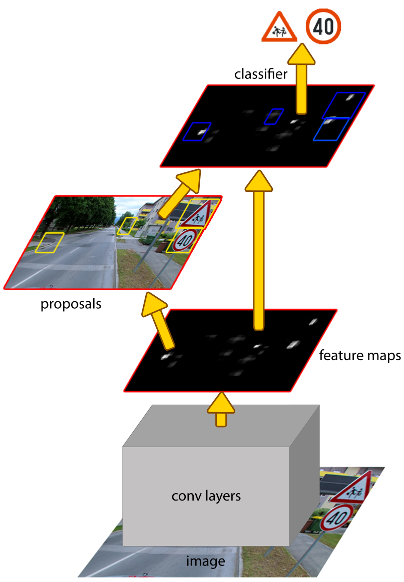

Figure 4. Faster R-CNN is a single, unified network for object detection and classification [23].

of Fast R-CNN are fine-tuned.

Faster R-CNN enables rapid detection and recognition in the test-phase. For each input image the trained model outputs a set of object bounding boxes, where each box is associated with a category label and a softmax score in the interval $[0, 1]$.

4. Experimental results

We first present our dataset with corresponding base and extended set of categories in Section 4.1, whereas implementation details are introduced in Section 4.2. Results of the experiments with the base set are reported in the context of the detection stage performance in Section 4.3, followed by the full pipeline evaluation in Section 4.4, whereas the analysis of efficiency in the practical application of semiautomatic road image database maintenance system is discussed in Section 4.5. Finally, results of the experiment on the extended set are presented in Section 4.6, and a discussion on failure cases is given in Section 4.7.

4.1. The dataset

Our dataset was acquired by the DFG Consulting d.o.o. company for the purpose of maintaining records of traffic signalization. The RGB images were acquired with a camera mounted on a vehicle that was driven through six different Slovenian municipalities. The image data was acquired in rural as well as urban areas. Only the images containing at least one traffic sign were selected from the vast corpus of collected data. Moreover, the selection was performed in such a way that there is usually a significant scene change between any pair of selected consecutive frames.

There is a total of 4989 images in the dataset, with 10801 tightly annotated traffic sign instances corresponding to 270 categories. The total number of instances is different for each category. In this work we have focused only on a subset of categories for which a sufficient number of samples are available. We chose 20 samples to be the minimum amount, which yielded 9943 annotations corresponding to 123 categories (Figure 1). Among these categories, 91 of them correspond to traffic signs with a relatively uniquely defined appearance. These are the categories of the base set. Although some by-design appearance variability is also present in the base set, significantly larger variability is present in the remaining 32 categories. These signs can be of variable sizes or color and can contain various text and numbers (Figure 2). These challenging categories together with the base set form the extended set.

To obtain a valid train-test split with sufficient number of samples for each category in both sets we have considered the following approach. For each image we have its associated world coordinates that were acquired using a GPS device. Images were first clustered into groups such that each pair of images that were acquired less than 50 meters apart were assigned into the same cluster. A cluster of images was then randomly assigned to either a train or test set, which ensured that any similar images were in the same set. Moreover, a restriction was set that about 25 % of traffic sign instances for each category have to appear in the test set. In this way 2752 images were selected for training and 928 for testing when the base set of categories was considered. When the extended set of categories was used the procedure selected 3305 training and 1272 testing images.

4.2. Implementation details

For the Faster R-CNN we use the publicly available Matlab implementation [23] that is based on Caffe framework [13]. The VGG-16 network model [27], which has 13 convolutional layers and 3 fully-connected layers, was employed as a network model, due to its great performance in the field of object recognition [23, 27]. We use the default parameters provided with the Faster R-CNN implementation. In particular, input images are re-scaled such that their shorter side equals 600 pixels, the Intersection-over-Union (IoU) threshold for non-maximum suppression (NMS) in the detection phase equals 0.7, the top-300 ranked proposals are then selected to proceed into the classification stage, where the regressed boxes are again filtered using NMS with IoU threshold of 0.35. For RPN a learning rate of 0.001 is used for the first 60k iterations, and 0.0001 for the next 20k iterations. A momentum of 0.9 and a weight decay of 0.0005 is applied. The same hyper-parameter values are used with Fast R-CNN, where the only differences are that the learning rate drop is performed at 30k iterations, and the optimization stops at 40k iterations.

4.3. Evaluation of the detection stage

We evaluated the detection stage using the base set of categories. An example input image is shown in Figure 5. It depicts five traffic signs that were successfully detected and recognized (denoted with green bounding boxes). Note that the green right arrow traffic sign corresponds to the extended set of categories, and is therefore correctly not detected in this image. The yellow bounding boxes denote the first (best) 30 region proposals (out of 300).

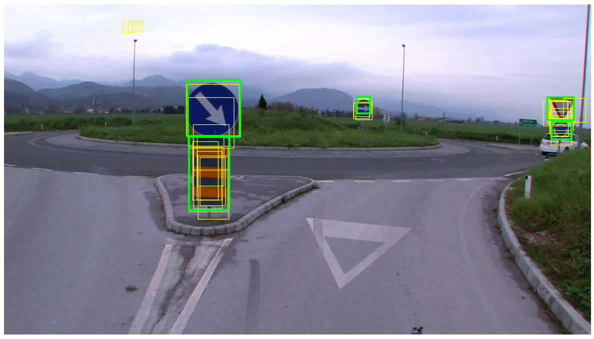

Figure 5. Example of traffic sign detection.

We can see that even 10 % of all region proposals generated with the implemented method cover pretty well all the present traffic signs. Such a performance is quite typical and can be observed in other test images as well.

To quantitatively asses the performance of the region proposal generator we measured the success of different numbers of region proposals. In this step of the detection pipeline we are interested in the recall; the proportion of the ground truth bounding boxes that are covered with the proposed regions. We counted a ground truth box to be detected if the IoU between the two boxes exceeded a predetermined threshold. We therefore measured recall of detected boxes with respect to different numbers of region proposals and different values of IoU. The results are plotted in Figure 6. We can see that the

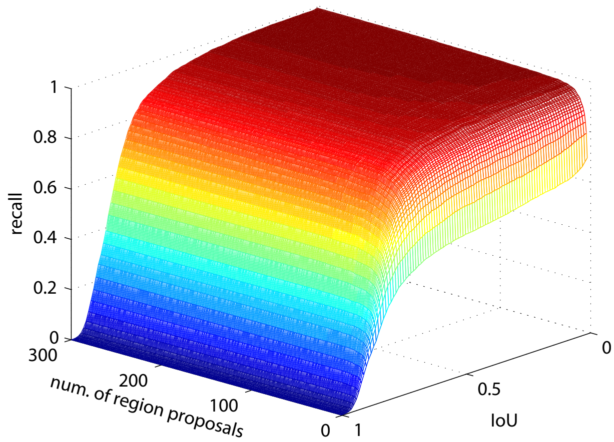

Figure 6. Recall in the detection stage for different IoU values and number of considered region proposals.

top ranked region proposals are very successful, as a small number of region proposals, for example 30, suffices to achieve a quite high recall, while after 100 region proposals, the performance practically does not improve any further. One can also observe that the recall stays relatively high until IoU of 40–60 % and then drops quite rapidly. However, as it turns out, such overlaps are already sufficient for reliable recognition of traffic signs.

4.4. Full pipeline evaluation

To evaluate the entire detection and recognition pipeline we measured the mean average precision (mAP) over all 91 classes in the base set. In the first experiment we trained our model without data augmentation, i.e., using only available 2752 training images (Section 4.1). Second, we augmented the training set using images with synthetically generated templates (6388 images). Finally, in the third experiment, the original training set was augmented with 5209 images generated by altering the real image regions that contain traffic signs (Section 3.1). The results are presented in Table 1. The results show that augmenting the training images with the generated ones proved to be useful; we were able to achieve 90 % mAP.

Table 1. mAP for different training sets.

Training set	mAP
Training images	85.59
Train. images + varied syn. templates	88.15
Train. images + varied train. samples	90.00

The performance of the method varies for different classes. Average precisions for individual classes are depicted as blue bars in Figure 7, while the blue bars in Figure 8 depict the histogram of all individual average precisions. The detector achieves the optimal performance for more than a third of all classes, and above 90 % mAP for two thirds of classes; however, there are still a few difficult classes where the average precision remains relatively low.

4.5. Semi-supervised verification

One possible practical application of our method is semiautomatic road image database maintenance system with a human verifier in the loop. The operators' task would be to manually verify the classifications with low confidence values, i.e., instances with low softmax score (Section 3.2), which would

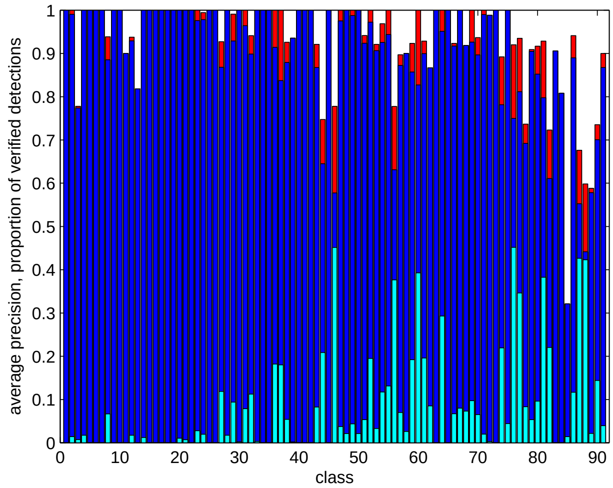

Figure 7. Precisions of all classes before (blue) and after (red) verification, proportion of verified detections (cyan) at the verification threshold of 0.95.

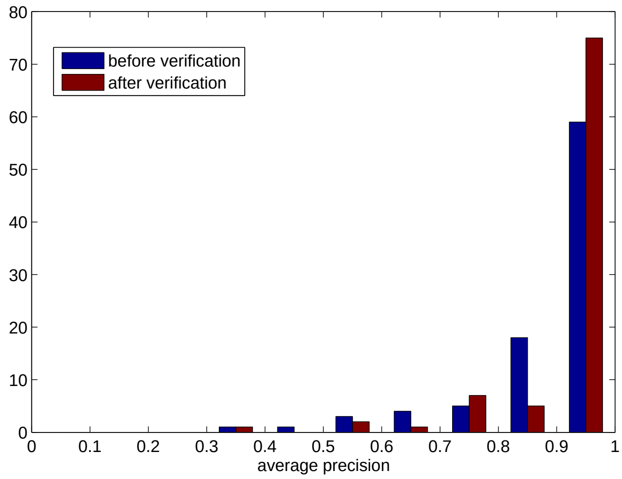

Figure 8. Histograms of class APs before and after verification.

enable the overall reduction of false positives (but not of false negatives).

We analyzed the performance of such a semi-supervised approach by eliminating the detected false positives with a confidence lower than the predefined threshold (these false positives would be spotted by the human verifier). The red bars in Figure 7 show the improvement over the original results. In fact, the overall mAP increases to 93.25 % if the threshold for verification is set to 0.95, which requires verification of 8.31 % of all detections. The proportion of detections that had to be verified is depicted for the individual classes with the cyan bars in Figure 7. The improvement is also observable in the histogram of all individual class average precisions, depicted in Figure 8, where red bars denote the results after verification.

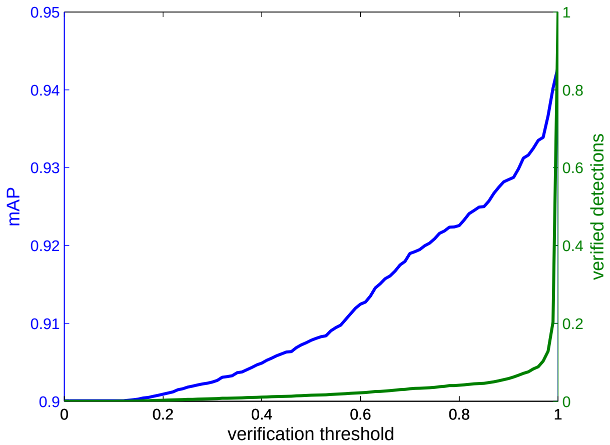

Figure 9. mAP (left, blue) and proportion of verified detections (right, green) for different verification thresholds.

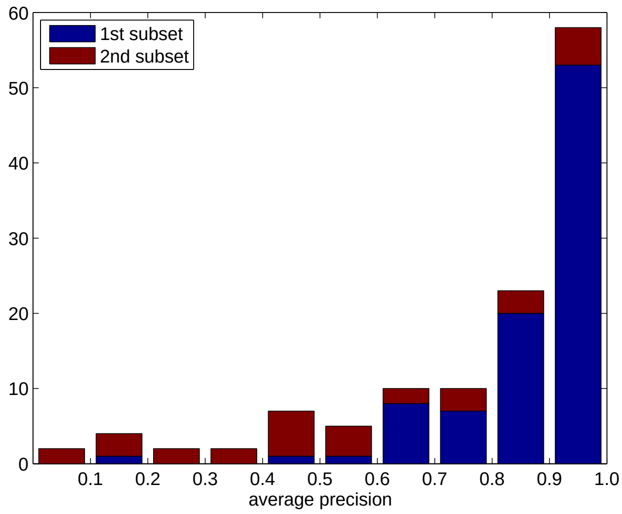

Figure 10. Histogram of APs for 123 classes.

We also varied the verification threshold and observed the improvement of the detector performance, as well as measured the additional effort that would have to be made by the operator. Figure 9 depicts the mAP, i.e., the mean of all individual proportions of the positive detections (the blue curve, left scale) as well as the mean of all proportions of the verified detections (thus the number of verifications over the number of detections, averaged over all classes). One can observe that a significant increase of mAP can be achieved by requiring a relatively small proportion (below 10 %) of detections to be verified. Having a human in the loop, we can therefore semiautomatically eliminate a large number of false positives, as in general they tend to have a lower detection score than the true positives.

4.6. Experiment with the extended set

Lastly, we also evaluated the performance of the approach on all 123 classes. The results are presented in Table 2 for all 123 classes together, as well as for the subset of 91 classes considered in the first experiment, and for the additional subset of 32 classes. One can observe that the performance dropped significantly; predominantly due to very low average precisions achieved on the additional classes. Figure 10 depicts the histogram of the class average precisions. The precisions of the base subset of 91 classes are depicted in blue, while the precisions of the additional classes are depicted in red. It is evident that the performance on the extended set is low. The reason for this is the highly challenging nature of the corresponding traffic sign classes (see Figure 2) and an insufficient number of available training instances to cover the very large intra-class appearance variability.

4.7. Discussion

The detector performs relatively well on the base 91 classes; the vast majority of detections are successful. There are, however, still a number of failures. Figure 11 illustrates a couple of failure cases. The first two rows show four examples (pairs of the detected image region on the left, and the recognized traffic sign on the right), where the detector detected the traffic signs that were not included in the training set, and recognized them as the most similar ones among the trained traffic signs. Although we consider these detections as false positives, these failures seem to be quite understandable. In the third row three more false positives are shown. In general, most of the failures are false negatives. The last two rows depict some of them. Some of traffic signs that were not detected are badly illuminated, damaged, occluded, or unusually oriented (the fourth row). There is, however, a number of traffic signs with a clear, undistorted appearance that were also not detected. Some of them are depicted in the fifth row of Figure 11. We observed that 82% of all missed objects, i.e., false negatives, were smaller than 50 pixels after input image down-scaling. This

Table 2. mAP for subsets of classes in 123 classes experiment.

Classes	mAP
All 123 classes	79.41
First subset (91 classes)	88.13
Second subset (32 classes)	54.62

Figure 11. A couple of failure cases. See text for detailed explanation.

indicates that detector is not robust enough to detect smaller traffic sign objects and future improvements should be made to address this issue.

5. Conclusion

In this paper we applied a convolutional neural network (CNN) approach to the problem of traffic sign recognition in both stages of the pipeline, i.e., the detection and classification stage. The Faster R-CNN model was applied for this purpose. Experiments were performed on our novel challenging dataset with a vast number of categories. Several of these categories have not yet been considered in previous works, perhaps due to their highly challenging nature, i.e., the appearance of the corresponding individual traffic sign instances can be different from instance to instance. Experimental results demonstrate that the considered approach delivers good performance in both detection and recognition stage, and that learning with data augmentation in a form of distorted graphical templates further improves the recognition performance. A further boost in recognition is obtained when synthetic samples are generated from real world instances. Moreover, we demonstrate that in the practical application of a database maintenance system, a relatively small number of interventions from a human operator could significantly improve the performance.

In future work, we will compare the performance of the Faster R-CNN approach with other state-of-the art methods in the context of traffic sign detection and recognition. We will also address the issue of detecting smaller objects which proved to be a majority of missed objects. We will address this issue by increasing image resolution, if permitted by hardware limitation, or by employing a multi-scale approach. We will also acquire additional learning samples that cover a larger spectrum of all possible intra-instance variations to improve the results in the experiments of the extended set. Furthermore, our dataset will be extended to contain a sufficient number of traffic sign instances for each of the remaining 147 (out of 270 total) categories. Finally, we are going to do our best to arrange the public availability of the dataset used in this work, i.e., making it the traffic sign detection and recognition benchmark.

References

[1] L.-P. Becker. Method and device for traffic sign recognition. *Patent, US 2010/0283855 A1*, 2014. 1

[2] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber. A committee of neural networks for traffic sign classification. In *IJCNN*, pages 1918–1921, 2011. 1, 3

[3] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber. Multi-column deep neural network for traffic sign classification. *Neural Networks*, 32:333–338, 2012. 3

[4] G. A. P. Coronado, M. R. Muòoz, J. M. Armingol, A. de la Escalera, J. J. Muòoz, W. van Bijsterveld, and J. A. Bolaòo. Detection and classification of road signs for automatic inventory systems using computer vision. *ICAE*, 19(3):285–298, 2012. 1

[5] A. Ellahyani, M. E. Aansari, and I. E. Jaafari. Traffic sign detection and recognition using features combination and random forests. *IJACSA*, 7(1):6861–6931, 2016. 1, 2, 3

[6] J. Greenhalgh and M. Mirmehdi. Real-time detection and recognition of road traffic signs. *Transactions on Intelligent Transportation Systems*, 13(4):1498–1506, 2012. 2, 3

[7] J. Greenhalgh and M. Mirmehdi. Recognizing text-based traffic signs. *Transactions on Intelligent Transportation Systems*, 16(3):1360–1369, 2015. 1

[8] A. Gudigar, S. Chokkadi, and R. U. A review on automatic detection and recognition of traffic sign. *Multimedia Tools and Applications*, 75(1):333–364, 2014. 2

[9] M. Haloi. A novel plsa based traffic signs classification system. *CoRR*, abs/1503.06643, 2015. 1, 2

[10] M. Haloi. Traffic sign classification using deep inception based convolutional networks. *CoRR*, abs/1511.02992, 2015. 3

[11] S. Houben, J. Stallkamp, J. Salmen, M. Schlipsing, and C. Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *IJCNN*, pages 1–8, 2013. 1, 2

[12] Z. Huang, Y. Yu, J. Gu, and H. Liu. An efficient method for traffic sign recognition based on extreme learning machine. *IEEE Transactions on Cybernetics*, (99):1–14, 2016. 2, 3

[13] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014. 5

[14] J. Jin, K. Fu, and C. Zhang. Traffic sign recognition with hinge loss trained convolutional neural networks. *Transactions on Intelligent Transportation Systems*, 15(5):1991–2000, 2014. 3

[15] F. Larsson and M. Felsberg. *SCIA*, chapter Using Fourier Descriptors and Spatial Models for Traffic Sign Recognition, pages 238–249. 2011. 2

[16] H. Li, F. Sun, L. Liu, and L. Wang. A novel traffic sign detection method via color segmentation and robust shape matching. *Neurocomputing*, 169:77–88, 2015. 2

[17] J. Lillo-Castellano, I. Mora-Jimenez, C. Figuera-Pozuelo, and J. Rojo-Alvarez. Traffic sign segmentation and classification using statistical learning methods. *Neurocomputing*, 153:286–299, 2015. 1, 2

[18] M. Mathias, R. Timofte, R. Benenson, and L. V. Gool. Traffic sign recognition - how far are we from the solution? In *IJCNN*, pages 1–8, 2013. 2

[19] A. Mogelmose. *Visual Analysis in Traffic & Re-identification*. PhD thesis, Faculty of Engineering and Science, Aalborg University, 2015. 1

[20] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. *Transactions on Intelligent Transportation Systems*, 13(4):1484–1497, 2012. 2

[21] G. Overett and L. Petersson. Large scale sign detection using hog feature variants. In *Intelligent Vehicles Symposium*, pages 326–331, 2011. 1, 2

[22] D. Pei, F. Sun, and H. Liu. Supervised low-rank matrix recovery for traffic sign recognition in image sequences. *IEEE SPL*, 20(3):241–244, 2013. 2, 3

[23] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 2, 3, 4, 5

[24] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. D. Stefano. Traffic sign detection via interest region extraction. *Pattern Recognition*, 48(4):1039–1049, 2015. 2

[25] S. Segvic, K. Brkic, Z. Kalafatic, V. Stanisavljevic, M. Sevrovic, D. Budimir, and I. Dadic. A computer vision assisted geoinformation inventory for traffic infrastructure. In *ITSC*, pages 66–73, 2010. 1, 2

[26] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. In *IJCNN*, pages 2809–2813, 2011. 3

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5

[28] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The german traffic sign recognition benchmark: A multi-class classification competition. In *IJCNN*, pages 1453–1460, 2011. 3

[29] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012. 1, 2

[30] G. Stein, O. Shachar, Y. Taieb, and U. Wolfovitz. Detecting and recognizing traffic signs. *Patent, US 8064643*, 2014. 1

[31] R. Timofte, V. A. Prisacariu, L. J. V. Gool, and I. Reid. Combining traffic sign detection with 3d tracking towards better driver assistance. In *Emerging Topics in Computer Vision and its Applications*. 2011. 1

[32] R. Timofte, K. Zimmermann, and L. V. Gool. Multi-view traffic sign detection, recognition, and 3d localisation. In *WACV*, pages 1–8, 2009. 2, 3

[33] V. Vukotić, J. Krapac, and S. Šegvić. Convolutional neural networks for croatian traffic signs recognition. In *CCVW*, pages 15–20, 2014. 2, 3

[34] S. B. Wali, M. A. Hannan, A. Hussain, and S. A. Samad. Comparative survey on traffic sign detection and recognition: a review. *Przeglad Elektrotechniczny*, 1(12):40–44, 2015. 2

[35] G. Wang, G. Ren, Z. Wu, Y. Zhao, and L. Jiang. A hierarchical method for traffic sign classification with support vector machines. In *IJCNN*, pages 1–6, 2013. 2

[36] Y. Wu, Y. Liu, J. Li, H. Liu, and X. Hu. Traffic sign detection based on convolutional neural networks. In *IJCNN*, pages 1–7, 2013. 3

[37] X. Yang, Y. Qu, and S. Fang. Color fused multiple features for traffic sign recognition. In *ICIMCS*, pages 84–87, 2012. 2

[38] F. Zaklouta and B. Stanciulescu. Real-time traffic sign recognition in three stages. *Robotics and Autonomous Systems*, 62(1):16–24, 2014. 3

[39] F. Zaklouta, B. Stanciulescu, and O. Hamdoun. Traffic sign classification using k-d trees and random forests. In *IJCNN*, pages 2151–2155, 2011. 2

[40] Y. Zeng, X. Xu, Y. Fang, and K. Zhao. Traffic sign recognition using deep convolutional networks and extreme learning machine. In *IScIDE*, volume 9242, pages 272–280, 2015. 3

[41] Y. Zhu, C. Zhang, D. Zhou, X. Wang, X. Bai, and W. Liu. Traffic sign detection and recognition using fully convolutional network guided proposals. *Neurocomputing*, 214:758–766, 2016. 1, 2, 3