

Deep Learning for Large-Scale Traffic-Sign Detection and Recognition

Domen Tabernik^{ID} and Danijel Skočaj

Abstract—Automatic detection and recognition of traffic signs plays a crucial role in management of the traffic-sign inventory. It provides an accurate and timely way to manage traffic-sign inventory with a minimal human effort. In the computer vision community, the recognition and detection of traffic signs are a well-researched problem. A vast majority of existing approaches perform well on traffic signs needed for advanced driver-assistance and autonomous systems. However, this represents a relatively small number of all traffic signs (around 50 categories out of several hundred) and performance on the remaining set of traffic signs, which are required to eliminate the manual labor in traffic-sign inventory management, remains an open question. In this paper, we address the issue of detecting and recognizing a large number of traffic-sign categories suitable for automating traffic-sign inventory management. We adopt a convolutional neural network (CNN) approach, the mask R-CNN, to address the full pipeline of detection and recognition with automatic end-to-end learning. We propose several improvements that are evaluated on the detection of traffic signs and result in an improved overall performance. This approach is applied to detection of 200 traffic-sign categories represented in our novel dataset. The results are reported on highly challenging traffic-sign categories that have not yet been considered in previous works. We provide comprehensive analysis of the deep learning method for the detection of traffic signs with a large intra-category appearance variation and show below 3% error rates with the proposed approach, which is sufficient for deployment in practical applications of the traffic-sign inventory management.

Index Terms—Deep learning, traffic-sign detection and recognition, traffic-sign dataset, mask R-CNN, traffic-sign inventory management.

I. INTRODUCTION

A PROPER management of a traffic-sign inventory is an important task in ensuring safety and efficiency of the traffic flow [1], [2]. Most often this task is performed manually. Traffic signs are captured using a vehicle-mounted camera and a manual localization and recognition is performed off-line by a human operator to check for consistency with the existing database. However, such manual work can be extremely time-consuming when applied to thousands of kilometers of roads. Automating this task would significantly reduce the

Manuscript received May 15, 2018; revised January 28, 2019; accepted March 31, 2019. Date of publication May 8, 2019; date of current version March 27, 2020. This work was supported in part by the ARRS Research Projects L2-6765 (VILLarD) and J2-9433 (DIVID) and ARRS Research Programme P2-0214. The Associate Editor for this paper was N. Papanikopoulos. (*Corresponding author: Domen Tabernik.*)

The authors are with the Faculty of Computer and Information Science, University of Ljubljana, Ljubljana 1000, Slovenia (e-mail: domen.tabernik@fri.uni-lj.si; danijel.skocaj@fri.uni-lj.si).

Digital Object Identifier 10.1109/TITS.2019.2913588



Fig. 1. The DFG traffic-sign dataset consists of 200 categories including a large number of traffic signs with a high intra-category appearance variations.

amount of manual work and improve the safety through a quicker detection of damaged or missing traffic signs [3].

A crucial step towards the automation of this task is replacing a manual localization and recognition of traffic signs with an automatic detection. In the computer-vision community the problem of traffic-sign recognition has already received a considerable attention [4]–[6], and excellent detection and recognition algorithms have already been proposed. But these solutions have been designed only for a small number of categories, mostly for traffic signs associated with advanced driver-assistance systems (ADAS) [7] and autonomous vehicles [8].

The detection and recognition of a large number of traffic-sign categories remains an open question. Various previous benchmarks have addressed the traffic-sign recognition and detection task [9]–[13]. However, several of them focused only on the traffic-sign recognition (TSR) and ignored the much more complex problem of the traffic-sign detection (TSD) where finding an accurate location of a traffic sign is needed. Other benchmarks that do address TSD mostly cover only a subset of traffic-sign categories, most often ones important for ADAS and autonomous-vehicle applications. Most categories appearing in such benchmarks have a distinct

appearance with a low inter-category variance and can be detected using hand-crafted detectors and classifiers. Such examples include round mandatory signs or triangular prohibitory signs. However, many other traffic-sign classes that are not included in the existing benchmarks can be much more difficult to detect as they have a high-degree of variation in appearance. Instances of these categories may have a different real-world size, aspect ratio, color, and may contain various text and symbols (e.g., arrows) that significantly differ between instances of the same class. This often leads to a large degree of intra-category (i.e. within-category) appearance variation and at the same time leads to a low degree of inter-category (i.e. between-categories) variations due to a similar appearance of objects from different categories.

Modifying existing methods with hand-crafted features and classifiers to handle such categories would be one option; however, that would be a time-consuming task, particularly when considering that many traffic-sign appearances are not consistent between countries. A much more sensible way is to use feature learning based on real examples. This can easily adapt and capture high degree of variability in the appearance over a large number of traffic signs. Recent advances in deep learning have shown promising results on the detection and recognition of general objects. Previous works already employed deep learning approaches for traffic-sign detection and recognition to some extent [6]; however, their evaluation focused only on a highly limited subset of traffic-sign categories [13]. One of the main limitations preventing the deep learning from being applied to a large set of traffic-sign categories is a lack of an extensive dataset with several hundred different categories and a sufficient number of instances for each category. This issue is particularly important in deep learning where models have tens of millions of learnable parameters and large numbers of samples are needed to prevent overfitting.

In this paper, we address the issue of learning and detecting a large number of traffic-sign categories for the road-based traffic-sign inventory management. As our main contribution, we propose a deep-learning-based system for learning a large number of traffic-sign categories using convolutional neural networks. We base our system on the state-of-the-art detector Mask R-CNN [14], which demonstrated a great accuracy and speed in the field of object detection. The same network architecture is used not only for the TSR but also for accurate localization using a region proposal network, resulting in an efficient end-to-end learning. In contrast to traditional approaches with hand-crafted features, the convolutional approach is applied to a broad set of categories, where individual traffic-sign instances are not only subject to change in lighting conditions, scale, viewing angle, blur, and occlusions, but also to significant intra-category appearance variations as well as low inter-category variations. Furthermore, we propose improvements to Mask R-CNN that are crucial for the domain of traffic signs. We propose adaptations that increase the recall rate, particularly for small traffic signs, and introduce a novel augmentation technique suitable for traffic-sign categories.

As our secondary contribution, we present a novel challenging dataset with 200 traffic-sign categories spread over



Fig. 2. Several different categories from the DFG traffic-sign dataset with a low inter-category and a high intra-category variability.

13000 traffic-sign instances and 7000 high-resolution images. The dataset represents a novel benchmark for a complex traffic-sign detection and recognition task with a large number of classes having a low inter-category and high intra-category appearance variability (e.g., see Figure 2). Additionally, the dataset contains enough instances to ensure appropriate learning of deep features. We achieve this by providing annotations of 200 traffic-sign categories with at least 20 instances per category (see Figure 1). Furthermore, our qualitative analysis serves as an important study for appropriateness of deep learning for the detection of a large number of traffic-sign categories.

The remainder of the paper is organized as follows. Section II provides the related work overview, Section III describes the employed method, Section IV describes the proposed traffic-sign dataset, Section V presents the experimental results, and the discussion on the qualitative analysis is provided in Section VI. The paper concludes with the discussion in Section VII.

II. RELATED WORK

An enormous amount of literature exists on the topics of TSR and TSD, and several review papers are available [11], [15]. 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 [15], [16]. Most 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) [10]: 3 super-categories, primarily intended for detection.
- The German Traffic-Sign Recognition Benchmark (GTSRB) [9]: 43 categories, intended for recognition only.
- The Belgium Traffic Signs (BTS) dataset [17]: 62 categories, for detection and recognition.
- The Mapping and Assessing the State of Traffic Infrastructure (MASTIF) [18]: 9 original categories, extended to 31 categories [19], acquired for road maintenance assessment service in Croatia.

- The Swedish traffic-sign dataset (STSD) [20]: 10 categories, for detection.
- The Laboratory for Intelligent and Safe Automobiles (LISA) Dataset [11]: 49 categories of traffic signs, acquired on the roads in the USA.
- The Tsinghua-Tencent 100K dataset [13]: 45 categories, large dataset with 10000 images containing at least one traffic sign and 90000 background images.

To enrich the set of considered traffic signs, some approaches sample images from multiple datasets to perform the evaluation [21], [22]. On the other hand, a vast number of authors use their own private datasets [4], [23]–[25]. To the best of our knowledge, the largest set of categories was considered in the private dataset of [24], distinguishing between 131 categories of non-text traffic signs from the roads of United Kingdom.

Despite a large number of traffic-sign datasets, a comparison of traffic-sign detectors for large numbers of categories remains a challenging problem. In contrast to the existing benchmarks that focus mostly on small numbers of super-categories (GTSDB [10]), or on small numbers of simple traffic signs (BTS [17], MASTIF [18], STSD [20], LISA [11]), our comprehensive dataset contains 200 traffic-sign categories, including a large number of categories with a significant intra-category variability. The closest large-scale dataset is the Tsinghua-Tencent 100K dataset; however, their evaluation still focuses only on 45 simple traffic signs. On the other hand, our dataset enables a comprehensive analysis of detectors in the context of the traffic-sign inventory management.

Various methods have been employed in TSR and TSD. Traditionally hand-crafted features have been used, like histogram of oriented gradients (HOG) [5], [10], [12], [16], [19], [24], [26], scale invariant feature transform (SIFT) [5], local binary patterns (LBP) [16] or integral channel features [26]. A wide range of machine learning methods have also been employed, ranging from support vector machine (SVM) [16], [24], [27], logistic regression [28], and random forests [16], [27], to artificial neural networks in the form of an extreme learning machine (ELM) [19].

Recently, like the entire computer vision field, TSR and TSD has also been subject to CNN renaissance. A modern CNN approach that automatically extracts multi-scale features for TSD has been applied in [29]. In TSR, CNNs have been used to automatically learn feature representations as well as to perform the final classification [30]–[33]. In order to further improve the recognition accuracy, a combination of CNN and Multilayer Perceptron was applied in [34], while an ensemble classifier consisting of several CNNs was proposed in [30], [32]. A method that uses CNN to learn features and then applies ELM as a classifier has been applied in [35], while [36] employed a deep network consisting of spatial transformer layers and a modified version of the inception module. It has been shown in [37] that the performance of CNN on recognition outperforms the human performance on GTSRB. A combined problems of TSR and TSD were addressed using CNNs in recent works of [6], [13]. In the

latter, they use a heavily modified OverFeat [38] network, while in the former they applied a fully convolutional network to obtain a heat map of the image, on which a region proposal algorithm was employed for detection. Finally, a separate CNN was then employed to classify the obtained regions.

Our proposed deep-learning-based approach differs from previous related works. In contrast to the traditional approaches with hand-crafted features and machine learning [12], [24], we propose a full feature learning with end-to-end learning. Our approach also differs from other deep-learning-based traffic-sign detection methods. Our method, which is based on Mask R-CNN, uses region proposal network instead of using a separate method for generating region proposals as in [6], and in contrast to [13], we employ deeper networks based on the VGG16 [39] and ResNet-50 [40] architectures. As opposed to both [6] and [13], we also employ a network pre-trained on ImageNet, which significantly reduces the need for training samples. In addition, we have implemented several extensions leading to superior performance.

III. TRAFFIC-SIGN DETECTION WITH MASK R-CNN

In this section, we present our system for traffic-sign detection using the Mask R-CNN detector extended with several improvements. First, we present the original Mask R-CNN detector, then we present our adaptation for learning traffic-sign categories, and finally, we present our data augmentation technique.

A. Mask R-CNN

Here we briefly describe Mask R-CNN and refer the reader to [14] for a more detailed description. The Mask R-CNN network [14] is an extension of Faster R-CNN [41], both of which are composed of two modules. The first module is 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 a 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 the “attention” mechanism, the RPN module tells the Fast R-CNN module where to look. Mask R-CNN then improves this system by combining the underlying network architecture with a Feature Pyramid Network (FPN) [42]. With the FPN, the detector is able to improve the performance on small objects, since FPN extracts features from the lower layers of the network, before the down-sampling removes important details in small objects. The underlaying network architecture, which is VGG16 [39] in Faster R-CNN, is replaced with a residual network (ResNet) [40] in Mask R-CNN.

Faster and Mask R-CNN are trained for the region proposal task as well as for the classification task. This is performed with a stochastic gradient descent. Mask R-CNN learns both networks simultaneously using end-to-end learning. The original Faster R-CNN implementation performed this with a 4-step optimization process that alternated between the two tasks. However, the newer end-to-end learning scheme from Mask R-CNN is also applicable to Faster R-CNN. Commonly, both networks are initialized with the ImageNet pre-trained model before they are trained on the specific domain.

Both methods enable fast 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 of $[0, 1]$.

B. Adaptation to Traffic-Sign Detection

Mask R-CNN is a general method developed for the detection and recognition of general objects. In order to adapt it to the particular domain of TSD, we developed and implemented several domain specific improvements.

1) Online Hard-Example Mining: We first incorporate online hard-example mining (OHEM) into the classification learning module (Fast R-CNN module). Following the work of Shrivastava *et al.* [43], that introduced OHEM for Faster R-CNN, we replace the method for selecting regions of interest (ROIs) that are passed to the classification learning module. Normally, 256 ROIs per image are selected randomly, some as foreground (traffic signs) and some as background. In our approach, we replace the random selection of ROIs with the selection based on their classification loss value. Regions are sorted based on their loss value and only ones with a high enough loss are passed to the classification learning module. This ensures learning on samples on which the network was mistaken the most, i.e., on hard examples. We perform selection for the background and the foreground objects separately to ensure a sufficient number of positive and negative samples during each gradient descent step.

We implement OHEM in an end-to-end learning framework by utilizing the existing classification module to obtain the classification losses for ROIs. Note that the classification loss, which represents a criteria for selecting ROIs, is not computed for all possible ROIs generated by the RPN but only for the top ROIs based on their objectness score. We take 2000 regions and perform a non-maxima suppression (NMS) to eliminate duplicated ROIs. This is a standard approach to reduce the number of ROIs in Mask R-CNN before they are selected for learning. We experimented with using more than 2000 regions before the NMS but this significantly increased the learning time due to the slower NMS without contributing to any performance gain.

2) Distribution of Selected Training Samples: The mechanism for selecting the training samples for the region proposal network is also improved in the proposed approach. Originally, the Mask R-CNN selects ROIs randomly. This is done separately for foreground and background. However, when many small and large objects are present in the image at the

same time the random selection introduces imbalance into the learning process. The imbalance arises due to large objects having a large number of ROIs that cover it, while small objects have only a small number of ROIs. Selecting samples based on this distribution skews the learning process, since larger objects are observed more often and favored much more than the smaller ones. To alleviate this issue we change the distribution of the selected training samples to evenly cover all sizes of the training objects. We achieve this by selecting the same number of ROIs for each object present in the image.

3) Sample Weighting: We incorporate additional weighting of samples during the learning process. Our evaluation showed that Mask R-CNN cannot achieve 100% recall due to missing region proposals in certain cases. We address this issue with different weighting of the training regions. During the learning, both foreground and background regions are selected; however, there are often many more background regions, since most of the traffic signs in images are small and only a few region proposals exist for those traffic signs. Without any weighting the learning process will observe background objects more often and will focus on learning the background instead of the foreground. We address this problem with smaller weights for the background regions, which forces the network to learn foreground objects first. This is implemented for the training process of the region proposal network as well as for the classification network, weighting backgrounds with 0.01 for the RPN and 0.1 for the classification network. This improvement is particularly important for the RPN, since regions missed at this point in the pipeline cannot be recovered later by the classification module and would lead to poor overall recall if not addressed.

4) Adjusting Region Pass-Through During Detection: Lastly, we also change the number of ROIs passed from the RPN to the classification network during the detection stage. The number of regions passed through need to be adjusted due to a large number of small objects that are commonly present in the traffic-sign domain. We increase this number from 1000 to 10000 regions per one FPN level before the NMS. After merging ROIs from all FPN levels and performing the NMS 2000 regions are retained.

C. Data Augmentation

An important factor to consider when learning deep models is the size of the training set. Due to millions of learnable parameters the system becomes undetermined without a sufficient number of training samples. We partially address this issue with a pre-trained model, one learned on 1.2 million images of ImageNet, but we also propose an additional data augmentation. The nature of the traffic-sign domain allows us to construct a large number of new samples using artificial distortions of existing traffic-sign instances.

Additional synthetic traffic-sign instances are created by modifying segmented, real-world training samples. The traffic signs in the proposed dataset are annotated with tight polygons (see Figure 7), and can therefore be segmented from the training images. Two types of distortions were performed: (i) geometric/shape distortions (perspective change, changes

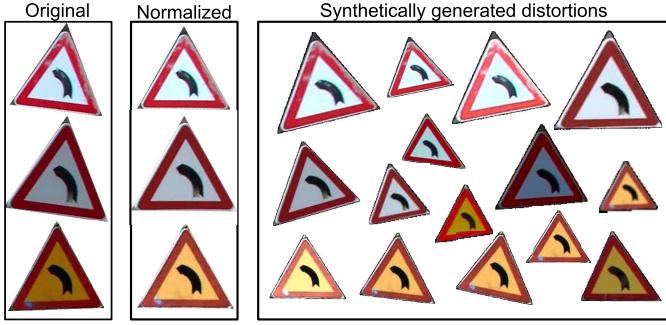


Fig. 3. Several examples of traffic-sign instances as generated during the process of data augmentation: the original image on the left, the normalized geometry and appearance in the middle, and the generated samples with synthetic distortions on the right.

in scale), and (ii) appearance distortions (variations in brightness and contrast).

Before applying geometric and appearance distortions we first normalized each traffic-sign instance. For the appearance normalization, we normalized the contrast of the intensity channel in the L*a*b domain, while for the geometric normalization, we calculated the homography between the instance annotation points and a geometric template for a specific traffic-sign class. We manually created templates for most of the classes with the exception of several classes where this was not possible (e.g. the train crossing sign, the arrow shaped direction signs, etc.). We generated new synthetic instances for those classes as well but without performing geometry normalization and without applying geometric distortions to synthetic instances.

In order to generate synthetic training samples that are as realistic as possible, we followed the distribution of the training-set's geometry and appearance variability. For the geometry change we estimated the distribution of Euler rotation angles (around X, Y and Z axis) of training examples, while for the appearance change, we estimated the distribution of averaged intensity values. We additionally estimated the distribution of scales using the size of geometry normalized (rectified) instances. We modeled all changes with a Gaussian mixture model, but used a single mixture component, $K = 1$, for the geometry and appearance, and two mixture components, $K = 2$, for the scale. Several examples of original, normalized and synthetically generated samples are shown in Figure 3, while histograms and their corresponding distributions for different distortions are depicted in Figure 4.

When generating synthetic distortions we sampled random values from the corresponding distributions. However, variance that is twice as large as the variance in the observed distribution was used to increase the likelihood of generating larger distortions. In the appearance distortion the distributions were not generic for all classes, but instead, we used a different distribution for every class. We used a class specific mean instead of a mean over all categories but we still applied a common variance calculated from all the categories. This guarded us from generating invalid contrast values for very dark/bright categories, such as gray or white direction signs.

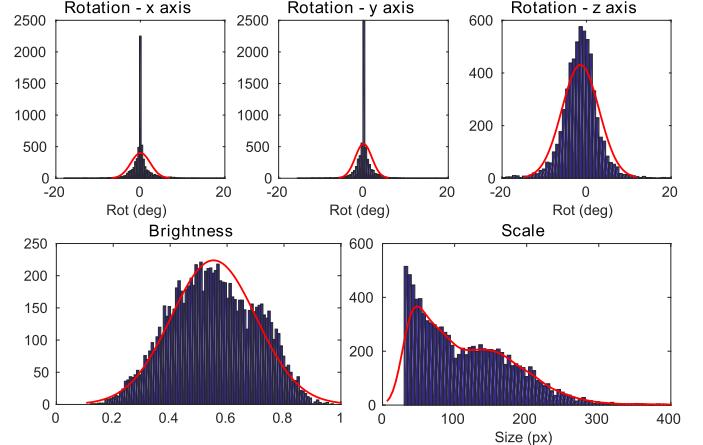


Fig. 4. Distributions of traffic-sign distortions computed for rotation in the top row, the appearance (i.e. brightness) in the bottom left side and the scale in the bottom right side. Red lines represent the Gaussian distributions, which are sampled when generating new examples.

To emulate the real-world settings, the newly generated traffic-sign instances were inserted into the street-environment-like background images. Background images were taken from the subset of the BTS dataset [17], which contains no other traffic signs. At least two, and at most five, traffic signs were placed in a non-overlapping manner in random locations of each background image, avoiding the bottom central part where only the road is usually seen. With the whole augmentation process we generated enough new instances to ensure that each category has at least 200 instances. This resulted in around 30000 new traffic-sign instances spread over 8775 new training images.

IV. THE DFG TRAFFIC-SIGN DATASET

Our dataset was acquired by the DFG Consulting d.o.o. company for the purpose of maintaining the inventory of traffic signs on Slovenian roads. The RGB images¹ were acquired with a camera mounted on a vehicle that was driven through several different Slovenian municipalities. The image data was acquired in rural as well as in urban areas. Only 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 images. Since images were acquired for the purpose of maintaining the traffic-sign inventory, this allowed the image acquisition to be performed in the day-time avoiding bad weather conditions such as rain, snow and fog. Nevertheless, the dataset does include other challenging variations in the weather and the environment that are present in the real-world environment such as: rural and city/urban landscape, different levels of natural occlusions and shadows, and various ranges of a cloudy sky and direct sunlight. A few images taken in winter conditions with a slight snow cover were also included.

¹The images have been anonymized by blurring the faces and vehicle license plates to comply with the EU GDPR legislation.



Fig. 5. Several examples of images from the DFG traffic-sign dataset.

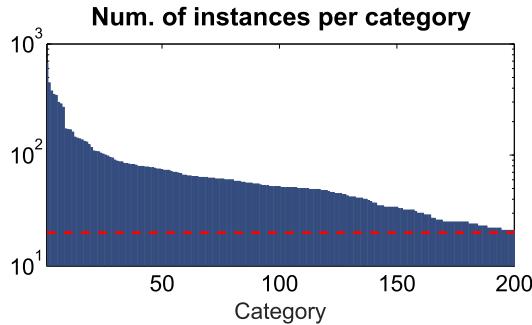


Fig. 6. Distribution of number of instances over categories in the DFG traffic-sign dataset. Horizontal red dashed line represents 20 instances per category, which we use as a cut-off point. Note, the distribution is shown in logarithmic scale.

Several examples of images in the DFG dataset are shown in Figure 5.

The dataset, termed the DFG traffic-sign dataset,² contains a total of 6957 images with 13239 tightly annotated traffic-sign instances corresponding to 200 categories. Most of images are of high-resolution (1920×1080), while less than 200 images are of a lower resolution (720×576). The total number of instances is different for each category (see Figure 6). Each image contains annotations of all traffic signs larger than 20 pixels for any of the 200 categories in form of a tightly annotated polygon (see Figure 7). Categories in the dataset represent a subset of all categories from the corpus of raw images provided by the company; however, some categories in the corpus did not meet the necessary criteria to create a quality dataset. In particular, all categories in the public dataset now meet the following three criteria: (a) each category has a sufficient number of instances (at least 20 instances with a minimal bounding box size of 30 pixels), (b) each category represents a planar object and (c) each category contains traffic signs that have at least some visual consistency. Among all categories in the DFG traffic-sign dataset roughly 70% of them correspond to traffic signs with low appearance changes, while a significantly larger appearance variability is present in the remaining 30%. Latter signs can be of a variable aspect

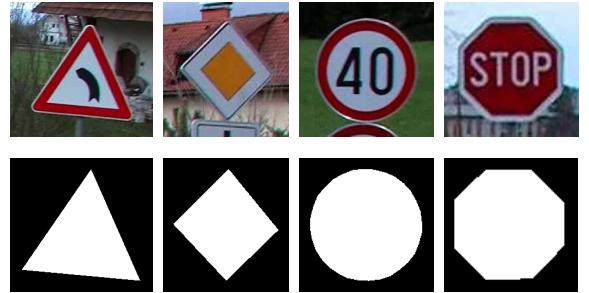


Fig. 7. Several examples of traffic signs in the DFG traffic-sign dataset with their corresponding annotation masks showing the precision of the annotations.

ratio or color and can contain various text and numbers. See 200 categories of traffic signs depicted in Figure 1.

Note that the dataset contains annotations as small as 15-20 pixels. However, annotations smaller than 30 pixels are marked with the "ignore" flag and are not considered neither for the training nor for the testing. We selected 30 pixels as a minimal size based on the down-sampling of features, which in Faster and Mask R-CNN results in 32x32 pixels being represented by 1x1 feature pixel. Many traffic signs marked with the "ignore" flag are annotated with a bounding box instead of a tight polygon. We also marked with the "ignore" flag a few traffic signs for which even humans had difficulty classifying into the specific category due to its small size, blurriness or occlusion.

A suitable train-test split was generated to provide a sufficient number of samples for both the training and the test set. A restriction was set that 25% of traffic-sign instances for each category have to appear in the test set. For the smallest categories with only 20 instances, this ensured 15 samples for the training set and 5 samples for the test set. Images were assigned randomly to either the training or the test set. However, an additional constraint mechanism was employed to ensure all images of the same physical object are always present either in the test set or in the training set but never in both of them at the same time. This was ensured by clustering the images that were acquired within 50 meter distance and assigning the whole clusters to the training or to the test set. In this way, we generated a training set with 5254 images and a test set with 1703 images.

²The dataset, termed DFG traffic-sign dataset, is publicly available at <http://www.vicos.si/Downloads/DFGTSD>

V. EXPERIMENTAL EVALUATION

In this section, we describe extensive evaluation of deep learning methods that are appropriate for the traffic-sign detection and recognition. We focus on evaluating two state-of-the-art, region-proposal-based methods: Faster R-CNN and Mask R-CNN. We first perform evaluation on the existing public traffic-sign dataset to establish a baseline comparison with the related work. Swedish traffic-sign dataset (STSD) is used for this purpose. Then, an extensive evaluation on the newly proposed DFG traffic-sign dataset is performed with a comprehensive analysis of the proposed improvements.

A. Implementation Details

A publicly available Caffe2-based, Python implementation of the Detectron [44] is used for both Faster and Mask R-CNN.³ For the Faster R-CNN, we employ the VGG16 [39] network with 13 convolutional layers and 3 fully-connected layers, while for the Mask R-CNN, we employ a residual network [40] with 50 convolutional layers (ResNet-50). The ResNet-50 architecture consists of 16 convolutional filters with kernel sizes of 3×3 or larger, while remaining convolutional layers use 1×1 kernel sizes. Mask R-CNN also implements Feature Pyramid Network (FPN) [42], which collects features from different layers of the network to capture the information from small objects, which may be removed in higher layers due to down-sampling. Both networks are initialized with a model pre-trained on ImageNet as provided by [44]. We also experimented with a larger variant of the residual network using 101 layers (ResNet-101), but performance did not improve compared to ResNet-50. We therefore focused only on the ResNet-50, which at the same time is faster with half the layers of ResNet-101.

Both methods use similar learning hyper-parameters. A learning rate of 0.001 is used for Faster R-CNN with a weight decay of 0.0005, while a learning rate of 0.0025 and a weight decay of 0.0001 is used for Mask R-CNN. Both approaches also use momentum of 0.9. The same hyper-parameters are used in all experiments. Note that the same hyper-parameters are used in [44] to pre-train the model on ImageNet dataset. Both methods are trained end-to-end with simultaneous learning of both the region proposal network and the classification network. We learn both methods for 95 epochs and reduce the learning rate by a factor of 10 at the 50th and 75th epoch. We use two images per batch per GPU and train on STSD with 2 GPUs and on DFG dataset with 4 GPUs. This resulted in effectively using 4 images per batch on the STSD and 8 images per batch on the DFG dataset.

B. Performance Metrics

Several different metrics are used in this study to evaluate the proposed approach. As a primary metric, we report the mean average precision (mAP), which is commonly used in the evaluation of visual object detectors. We use two

³Our proposed improvements have been implemented in the Detectron framework and are publicly available in the GitHub repository: <https://github.com/skokec/detectron-traffic-signs>

variants of the mAP: (i) mAP⁵⁰, based on the PASCAL visual object challenge [45], and (ii) mAP^{50:95}, based on the COCO challenge [46]. Both metrics define a minimal intersection-over-union (IoU) overlap with the ground-truth region for a detection to be considered as a true positive, and both compute the average precision (AP) as the area under the precision-recall curve to accurately capture the trade-off between the miss rate and the false-positive rate. AP is calculated for each category independently and the final metric consists of AP values averaged over all categories. A fixed IoU overlap is used in the mAP⁵⁰—using the PASCAL-based IoU overlap of 0.50. However, in mAP^{50:95}, the reported value is an average of mAP values calculated at a range of IoU overlap values. The reported values are averaged over the IoU overlap range of [0.50, 0.95] with 0.05 increments, the same range as used in the COCO detection challenge [46]. Thus, the COCO-based mAP puts more emphasis on the quality of region overlaps, while the PASCAL-based mAP ignores that aspect.

For a comparison with the state-of-the-art, we also report precision and recall values at the best F-measure and their corresponding error rates, i.e. false-positive rate as $1 - \text{precision}$ and miss rate as $1 - \text{recall}$, respectively. The false-positive rate shows how many detections are false, while the miss rate reveals how many traffic signs were not detected at all.

C. Comparison to the State-of-the-Art

Although many previously proposed approaches exist, it is quite difficult to perform a reliable comparison with those approaches, since they are mostly evaluated on non-public datasets or, only on the TSR task. To this end, we evaluated the proposed method on the Swedish traffic-sign dataset (STSD), comparing the results to the previously best performing methods published in [6], and indirectly to other methods reported therein.

The STSD benchmark contains around 20 categories with simple traffic signs in over 19236 images separated equally into the training (denoted *Set1* in STSD) and the test set (denoted *Set2*). However, only a subset of 3777 images from both sets contain annotations (denoted as *Part0* in each set). We follow the evaluation protocol of [6] and use only ten categories with images from *Set1Part0* for the training and images from *Set2Part0* for the testing. For a fair evaluation with [6], we consider only annotations with bounding box sizes of at least 50 pixels. The remaining annotations are ignored in both the train and the test stage. Due to the GPU memory limitations, we resized images to have image size of at least 918 pixels (i.e., both width and height are at least 918 pixels). For fair a comparison between different architectures, the same image size was used in all variants of Faster/Mask R-CNN. We did not use data augmentation in this experiment.

Detailed results on STSD are reported in Tables I and II, with the corresponding error rates depicted in Figure 8. When focusing on the related work and Faster/Mask R-CNN without our adaptations it is clear that pre-computed region proposals from R-CNN (as reported in [6]) perform worse

TABLE I
DETAILED RESULTS ON SWEDISH TRAFFIC-SIGN DATASET (STDS) FOR DIFFERENT CATEGORIES

Traffic Sign	FCN [6]			Faster R-CNN			Mask R-CNN (ResNet-50)					
							No adaptations			With adaptations (our)		
	Prec.	Rec.	Prec.	Rec.	AP ⁵⁰	Prec.	Rec.	AP ⁵⁰	Prec.	Rec.	AP ⁵⁰	
PED. CROS.	100.0	95.2	92.6	92.6	94.1	100.0	97.5	98.2	99.2	97.6	97.6	
PASS RIGHT SIDE	95.3	93.8	98.1	98.1	99.5	94.8	98.2	98.6	100.0	98.2	99.8	
NO STOP/STAN	100.0	75.0	92.3	92.3	86.5	81.2	100.0	95.4	86.7	100.0	83.9	
50 SIGN	100.0	100.0	81.2	92.9	90.3	87.5	100.0	97.5	90.0	96.4	96.9	
PRIORITY ROAD	100.0	98.9	98.7	95.1	92.1	97.5	97.5	96.9	98.7	92.9	89.8	
GIVE WAY	96.7	96.7	100.0	94.1	94.1	100.0	91.4	91.4	100.0	94.1	94.1	
70 SIGN	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	
80 SIGN	94.4	77.3	100.0	95.2	95.2	95.2	100.0	99.8	100.0	100.0	100.0	
100 SIGN	90.5	100.0	94.1	88.9	92.5	100.0	61.1	74.8	100.0	93.8	93.8	
NO PARKING	100.0	92.1	96.8	90.9	98.5	96.7	90.6	95.9	100.0	93.9	96.5	
Averaged	97.7	92.9	95.4	94.0	94.3	95.3	93.6	94.9	97.5	96.7	95.2	

TABLE II
EVALUATION ON SWEDISH TRAFFIC-SIGN DATASET (STSD) WITH REPORTED AVERAGED VALUES OVER TEN CATEGORIES

Average	R-CNN [6]	FCN [6]	Faster R-CNN	Mask R-CNN (ResNet-50)	
				No adapt.	Adapt. (ours)
Precision	91.2	97.7	95.4	95.3	97.5
Recall	87.2	92.9	94.0	93.6	96.7
F-measure	88.8	95.0	94.6	93.8	97.0
mAP ⁵⁰	/	/	94.3	94.9	95.2

than the newer R-CNN variants with the region proposal network. Error rates for R-CNN are twice as large as for the Faster/Mask R-CNN. On the other hand, the fully convolutional method (FCN) proposed by [6] achieves a significantly lower false-positive rate of 2.3% than both original Faster and Mask R-CNN, but has a slightly worse miss rate of 7.1%. Faster and Mask R-CNN have a lower miss rate by 1 percentage point (pp.). The standard mAP⁵⁰ metric in Table II also shows Faster R-CNN and Mask R-CNN with ResNet-50 achieving mAP⁵⁰ of 94.3% and 94.9%, respectively.

Results also show that the best performance is obtained when our adaptations are applied to the Mask R-CNN. Our proposed approach, in this case, achieves mAP⁵⁰ of 95.2%, with an average false-positive rate of 2.5% and an average miss rate of 3.3%. Compared to the related work, the FCN [6] achieves a similar false-positive rate but has at least twice as large miss rate at 7.1%. Improvements in our approach are better reflected in F-measure, which is defined as a harmonic mean between the precision and the recall. Our approach clearly outperforms the state-of-the-art with 2pp. higher F-measure. Those improvements directly stem from our proposed adaptations and not from the Faster/Mask R-CNN as the average miss and false-positive rates without our adaptations are still 6.6% and 4.7%, respectively, while they are reduced to only 3.3% and 2.5% with the proposed improvements. This is reflected in an improved F-measure and in mAP⁵⁰ as well.

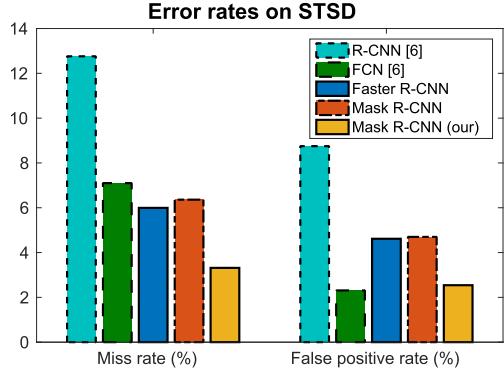


Fig. 8. Miss rates ($1 - \text{recall}$) and false positive ($1 - \text{precision}$) rates on Swedish traffic-sign dataset averaged over ten categories. Values are calculated at ideal F-measure. Note, smaller values are better.

D. Evaluation on DFG Traffic-Sign Dataset

Next, the proposed method is extensively evaluated on the DFG traffic-sign dataset. We use the train-test split as presented in Section IV with 200 categories in 5254 training and 1703 testing images, and using only annotations with at least 30 pixels in size. Annotations below 30 pixels are ignored during the training and during the evaluation we ignore detections of those objects to prevent penalizing the detector when it correctly detects small objects. We further resize images for both the training and the testing due to memory limitations. We resize images in all variants of Faster/Mask R-CNN to have image sizes of at least 840 pixels in both width and height. This was made for fair comparison under the same hardware limitations for all network models. Considering images are Full-HD with the image height of 1080 pixels, this change represents slightly less than a 25% reduction in size.

1) *Region Proposal Evaluation:* We first evaluate the region proposal network separately from the classification network. This allows us to assess the quality of region proposals as generated by the RPN before they are passed to the classification module. We take top N regions from the RPN and observe the miss rate and the recall rate of all annotated traffic signs.

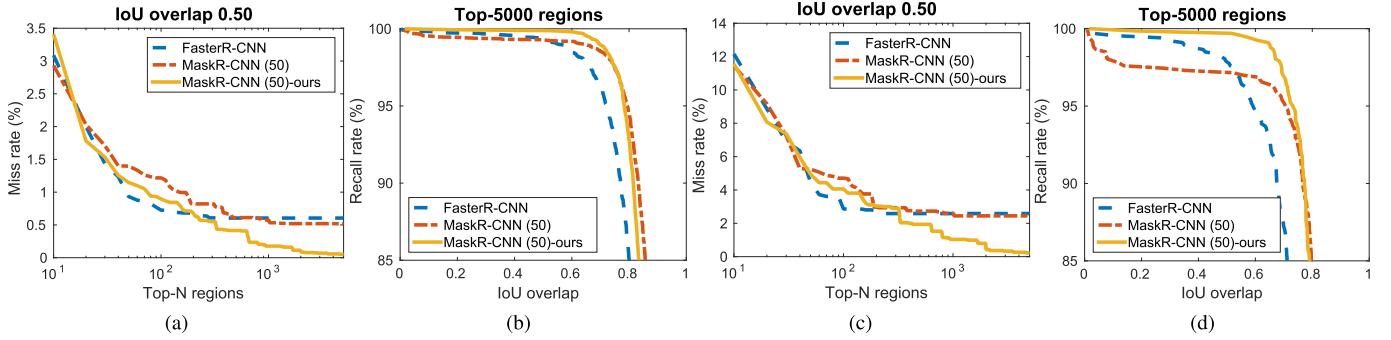


Fig. 9. Miss rate and recall for region proposals generated by the RPN. Graphs (a) and (b) show results when considering all valid annotations, while graphs (c) and (d), when considering only ground-truth traffic signs in sizes of 30-50 pixels. We show in (a) and (c) the miss rate over top-N regions using IoU overlap of 0.50, and in (b) and (d), the recall rate over different IoU overlaps using the top 5000 region proposals.

To ensure a correct balance between the categories with either a small or a large number of instances, we calculate the metric for individual categories and then report the average over all categories.

Results are reported in Figure 9, with (a) - (b) showing results when all annotations are considered and with (c) - (d), for smaller traffic signs only, i.e., when considering only ground-truth traffic signs that are 30-50 pixels in size. In both cases, we report the miss rate over the top-N regions using an IoU overlap of 0.50 in (a) and (c), and the recall over different IoU overlaps using the top 5000 region proposals in (b) and (d). Figure 9b first reveals that Faster R-CNN performs worse than the other methods. This is particularly evident at higher IoU overlaps where Faster R-CNN performs more than 5pp. worse.

The miss rates of various top-N regions, shown in Figure 9a, demonstrate that all methods perform extremely well with over 99% of all traffic signs found. However, only our proposed method achieves close to zero miss rate, and as indicated by the recall over IoU overlaps in Figure 9b, the proposed method is able to retain a high recall at high overlap values. This suggests that our adaptations decrease the miss rate of the RPN and higher quality regions can be produced, i.e., regions with a high overlap with the ground-truth. Moreover, improvements are more significant in smaller regions, as shown in Figure 9c and 9d. In this case, our adaptation achieves a significantly better miss rate than Faster/Mask R-CNN that did not use our adaptation. Even at a more liberal IoU overlap of 0.50, the standard approach achieves a 3% miss rate, while our adaptation achieves a miss rate close to zero. This difference is well observed in Figure 9d, showing our proposed method achieving higher recall rates at larger IoU.

Improvements in the miss rate at this level are important for the whole pipeline, since objects missed by the region proposals at this stage cannot be recovered later by the classification network. Results show that Mask R-CNN is unable to achieve a full detection of all objects, particularly for small objects; however, our adaptations overcome this issue and achieve a miss rate near zero.

2) *Full Pipeline Evaluation:* Next, we evaluate the whole detection pipeline with the RPN and classification networks combined. We report our results in terms of the mean average

TABLE III
RESULTS ON DFG TRAFFIC-SIGN DATASET

Faster R-CNN	Mask R-CNN (ResNet-50)		
	No adapt.	With adapt.	With adapt. and data augment.
mAP ⁵⁰	92.4	93.0	95.2
mAP ^{50:95}	80.4	82.3	82.0
Max recall	93.8	94.6	96.5

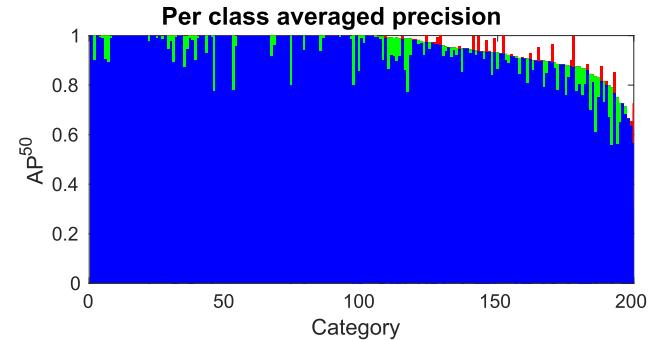


Fig. 10. Sorted per-class AP⁵⁰ distribution on the test set of the DFG traffic-sign dataset. The blue bars depict Mask R-CNN (ResNet-50) with our improvements and data augmentation, while green and red bars show change in performance (increased for green and decreased for red) compared to the base Mask R-CNN (ResNet-50) without our improvements.

precision (mAP) over all 200 categories as well as in terms of maximal possible recall that can be attained with the final detections when thresholding the score at 0.01. This value is directly related to the miss rate and the recall rate of region proposals in the previous section, and when both values are compared, we can deduce how many traffic signs were missed due to a poor performance of the classification network only.

Results are reported in Table III and clearly show that Faster R-CNN performs the worst among all methods, while the best results are achieved with our adaptations for Mask R-CNN. Nevertheless, all methods achieve mAP⁵⁰ of over 90%. Compared to the original Mask R-CNN, our proposed adaptations already improve results when measured in mAP⁵⁰ and maximal recall/miss rate metrics, even without data augmentation.

TABLE IV
RESULTS ON DFG TRAFFIC-SIGN DATASET WHEN CONSIDERING DIFFERENT SIZES OF TRAFFIC SIGNS

Traffic-sign size (% signs retained)	Faster R-CNN		Mask R-CNN				Mask R-CNN with adapt. and data augmentation (ours)			
			ResNet-50		ResNet-101		ResNet-50		ResNet-101	
	Max recall	mAP ⁵⁰	Max recall	mAP ⁵⁰	Max recall	mAP ⁵⁰	Max recall	mAP ⁵⁰	Max recall	mAP ⁵⁰
min 30 px (100%)	93.8	92.4	94.6	93.0	94.8	93.2	96.5	95.5	96.1	95.2
min 40 px (89%)	96.1	95.0	96.8	95.3	96.8	95.3	97.4	96.7	97.0	96.4
min 50 px (80%)	96.6	95.0	96.7	94.9	96.8	95.2	97.2	96.0	96.8	95.5

The performance in mAP⁵⁰ metric is improved from 93% to over 95%, and the miss rate error is almost halved from 5.4% to 3.5%. Slightly worse results are achieved in the mAP^{50:95} metric but this is improved when augmentation is enabled. With augmentation we slightly improve mAP⁵⁰, and significantly improve mAP^{50:95} from 82-82.3% with the original Mask R-CNN to 84.4% for the case when our adaptations and data augmentation are used. The data augmentation has contributed mostly to improving the precision of bounding boxes. Results also reveal that while overall miss rate has been reduced by half compared to the original Mask R-CNN, there still remain 3.5% missed objects despite, as shown in the previous section, having near zero miss rate in the region proposals. This points to traffic-sign detections being lost by the classification network.

3) *Different Traffic-Sign Sizes*: We also perform the evaluation considering different traffic-sign sizes with the results reported in Table IV. This analysis reveals a poor performance on smaller objects when using the original Faster and Mask R-CNN. The difference in both mAP⁵⁰ and the maximal recall rate between small and large objects is around 2pp.. However, with our adaptations, the detection of smaller objects is improved significantly and completely eliminates the performance gap between the detection of small and large objects. Moreover, this is achieved on top of the improved detection on large objects.

4) *Deeper Residual Network*: We also show results obtained with ResNet-101 architecture in Table IV. ResNet-101 performs similarly to the smaller ResNet-50 in most cases. When our improvements are not included, ResNet-101 performs less than 0.2pp. better; however, this reverses when our improvements are included. The difference between both of them still remains minimal at below 0.4pp.. Since ResNet-101 is larger with twice as many layers with more computational resources required, the ResNet-50 represents a significantly better choice.

VI. QUALITATIVE ANALYSIS

In this section, we demonstrate the performance of our approach on traffic-sign detection with additional qualitative analysis. We focus only on the best performing model, namely Mask R-CNN using ResNet-50 with our adaptations and data augmentation. All results in this section are reported on the test set of the DFG traffic-sign dataset.

A per-class distribution of AP⁵⁰ is depicted in Figure 10. This graph clearly shows that a large number of traffic-sign classes (108) are detected and recognized with average precision of 100%, i.e. with no errors. For the remaining categories our approach still achieved AP of above 90% on 60 of them, and above 80% on 23 of them.

Figure 11 further shows the traffic-sign classes with their corresponding AP⁵⁰ sorted by their AP⁵⁰ in descending order. The best performing categories at the top of the list are mostly the traffic signs with a low intra-category variations, i.e. with a fixed size and a fixed appearance. This includes various triangular danger signs, circular prohibitory signs, speed limit signs, rectangular information signs, etc. On the other hand, the worst performing signs at the bottom are traffic signs with a large variation of their sizes/aspect ratios as well as with a large intra-category variations, i.e., their content significantly varies from instance to instance. This includes particularly complex class of mirrors (both rectangular and round mirrors), digital speed signs, various direction signs and signs marking the start or the end of the towns.

A. Traffic Signs With High Intra-Category Variations and Good Performance

Figure 11 reveals several traffic signs with extremely good detection rate despite having large intra-category variations in their appearance. Samples for three such traffic-sign categories are depicted in Figure 12, namely: (i) *large-direction-with-separate-lanes*, (ii) *left-arrow-shaped-direction* and (iii) *right-gray-direction*. Each row in this figure depicts one category with eight instances. For clarity we display only the relevant part of the image. True detections are shown in green, false detections in red and missing detections in magenta. Examples are also sorted by their descending detection score from the left to the right. Therefore if true (green) and false (red) positive detections can be successfully separated with a threshold, then false detections can be trivially eliminated by setting an appropriate detection threshold. Note that this is important when looking at false detections as many of them are not problematic at all.

When focusing on the *large-direction-with-separate-lanes* traffic-sign category in the first row in Figure 12, an extremely good performance is clearly shown for the traffic signs that have a quite significant variation in their content as well as a large variation in their sizes and aspect ratios. The first image

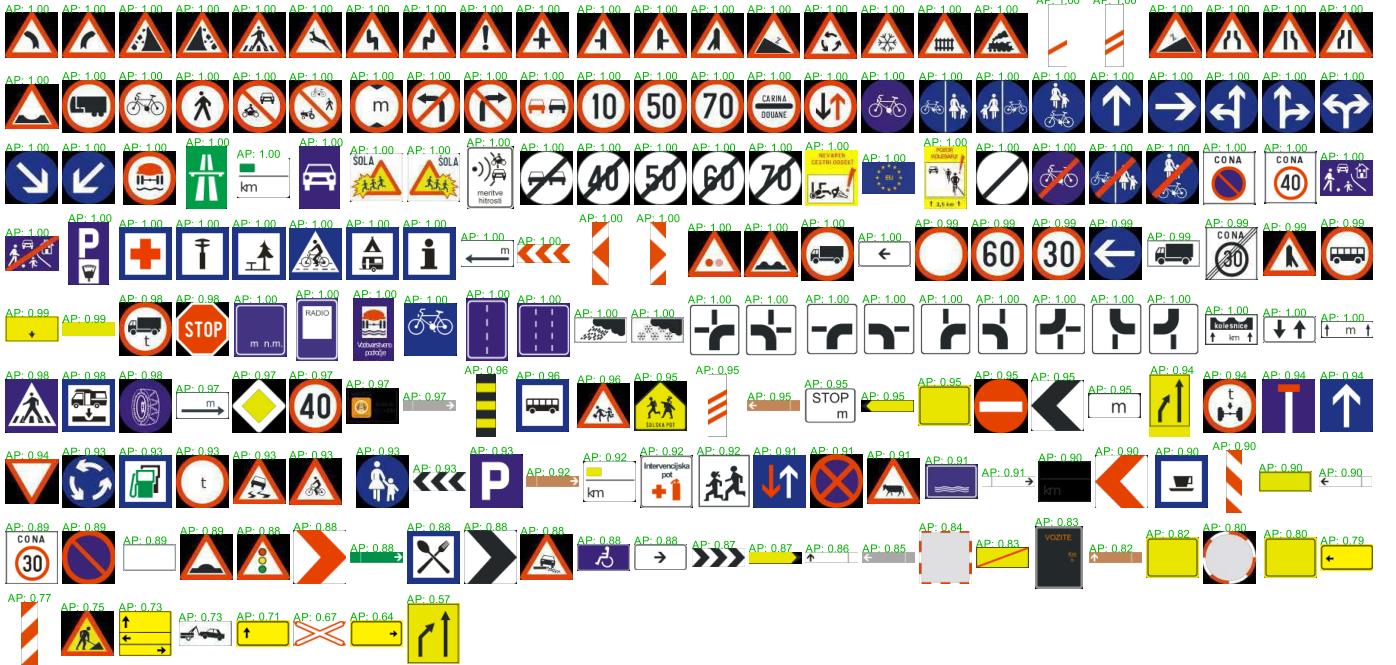


Fig. 11. DFG traffic-sign categories sorted by the average precision (AP^{50}) calculated when using Mask R-CNN ResNet-50 with our adaptations and data augmentation.



Fig. 12. Examples of complex traffic signs with a variable content and a good detection on the test set of the DFG traffic-sign dataset. True positives are depicted in green, false positives in red, and missing detections (false negatives) in magenta. Images are best viewed in colors.

in the top row depicts a good example of this as the traffic sign was detected with a high score despite having completely different color combination than other instances of the same class. Several detected instances are also quite small, yet our approach successfully detects them.

The second row in Figure 12 depicts detections of a *left-arrow-shaped-direction* traffic sign. This category is fairly difficult to detect as aspect ratios vary quite significantly from instance to instance, mostly due to wide viewing angles, yet the detector did not have significant issues finding them. The second-to-last example in the second row is also significantly cropped; however, the detector is still able to correctly find it.

Finally, detections for the *right-gray-direction* traffic sign are shown in the last row in Figure 12. Detection of this category is difficult mostly due to a significant variation of the content. Those traffic signs also often appear side-by-side in multiple rows which makes it difficult to generate the correct region proposal. Nevertheless, most instances have been correctly found.

B. Traffic Signs With Poor Performance and Low Intra-Category Variations

Next, we focus on three worst performing traffic signs despite having low appearance variation within a category,

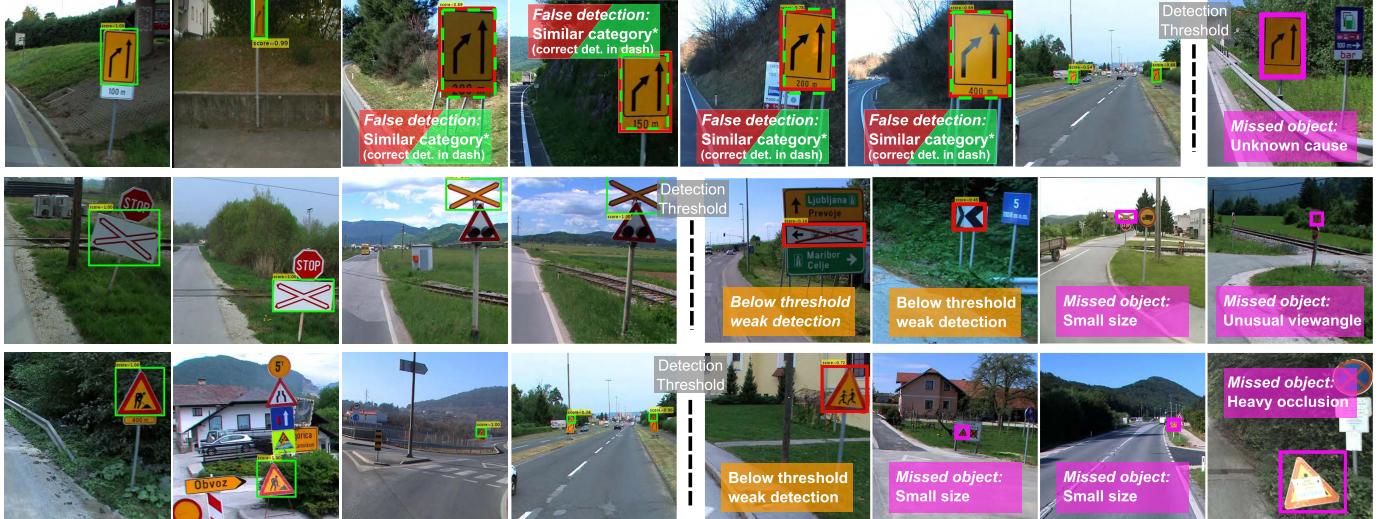


Fig. 13. Examples of traffic signs with a fixed content but poor detection on the test set of the DFG traffic-sign dataset. True-positive detections are marked in green, false positives in red and missing detections (false negatives) in magenta. (*) Note that false detections in the first row occur due to two almost identical traffic-sign categories in the dataset (one with distance label below and one without). True detections with the other category detector are shown in dashed green line. Images are best viewed in colors.

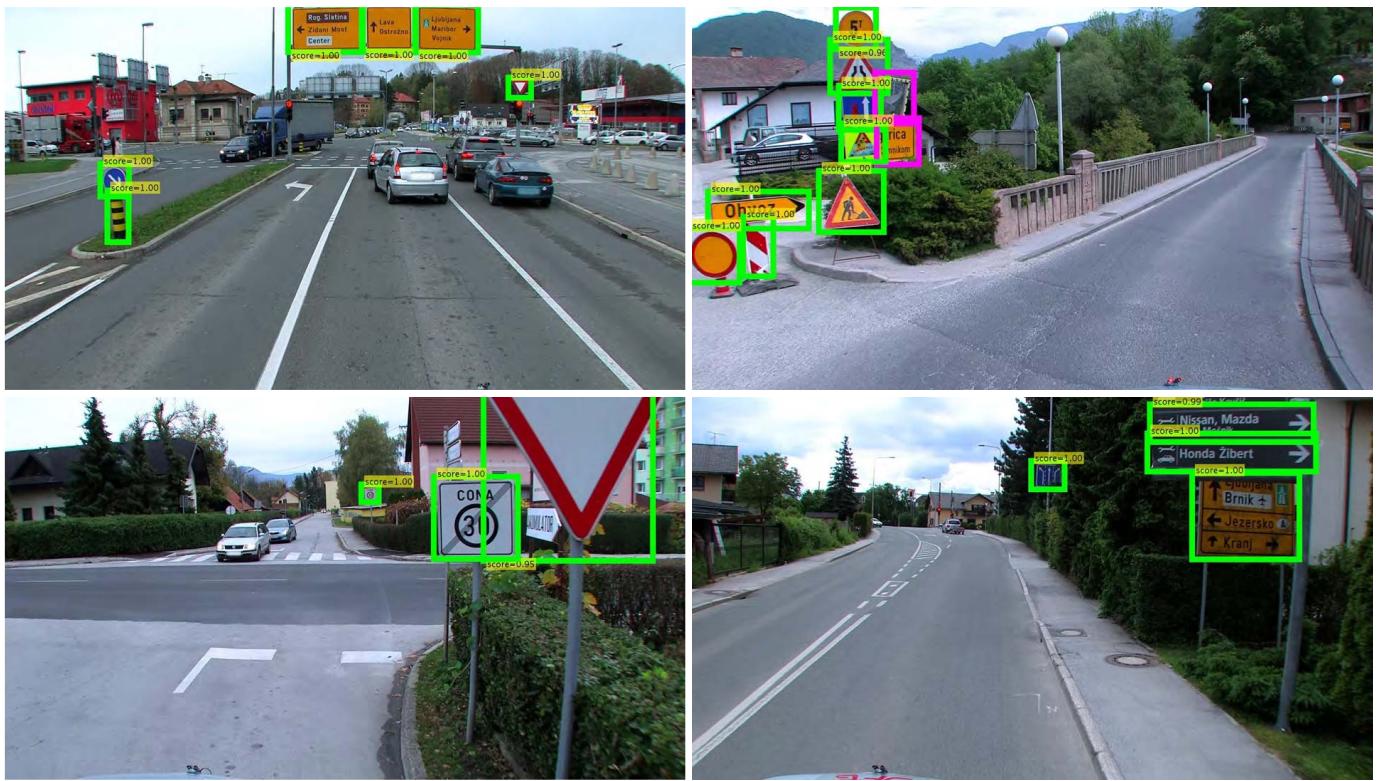


Fig. 14. Examples of detections on the test set of the DFG traffic-sign dataset. True detections shown in green and missing, in magenta. Images are best viewed in colors.

namely: (i) *left-lane-merger*, (ii) *train-crossing* and (iii) *work-in-progress*. Samples are depicted in Figure 13 and are organized in a similar manner as in Figure 12, with eight examples per category in a row, sorted by their descending detection score.

The worst results are achieved for the *left-lane-merger* traffic sign with the AP⁵⁰ of 57%. Mask R-CNN correctly detects

four out of five test instances, but appears to detect four false traffic signs as well, as can be seen in the top row. However, those false detections should not be considered problematic as the traffic sign is identical to the *left-lane-merger* sign with the only difference in the distance value printed below the sign. Since the correct category is also detected (shown with the dashed green line), those false detections would be eliminated

by the across-category non-maxima suppression, meaning that even in this case the issue is not as bad as it might seem. Still, such extremely minor differences between those two categories appear to pose a challenge for deep learning and point to existing limitations of deep learning methods.

The detector is also exhibiting inferior performance for the *train-crossing* traffic sign as seen in the second row in Figure 13. The reason in this case can be found in two missed detections out of total six traffic signs. Both missed objects are very small, with one having a fairly wide viewing angle, making the detection also extremely difficult. A few detections on false objects are also visible, most likely due to the presence of a cross-like shape. However, they do not contribute to a poor performance due to their low detection score.

The primary issue for the low performance on the *work-in-progress* sign, depicted in the third row of Figure 13, is a high miss rate. Three out of eleven traffic signs are not detected. Most objects missed are fairly small. The exception is the instance depicted in the last column where a significant occlusion would pose difficulty even for humans—its category was deduced mostly from the context.

C. Overall Detection

Despite some missed detections shown in Figure 13, the detector still performs extremely well even for several difficult cases. For instance, the second example in the first row of Figure 13 is extremely difficult to detect due to a large viewing angle, but the detector still managed to find it—even with a large score. The detector was also able to find some fairly small instances, such as ones in the first and the last row.

A good performance is also presented in Figure 14 where all traffic-sign detections are displayed for a couple of full-resolution images. This figure shows detections of several complex instances with occlusions and small traffic-sign sizes; however, the detector still performs extremely well.

VII. DISCUSSION AND CONCLUSION

In this work, we have addressed the problem of detecting and recognizing a large number of traffic-sign categories for the main purpose of automating the traffic-sign inventory management. Due to a large number of categories with a small inter-class but a high intra-class variability, we proposed detection and recognition utilizing an approach based on the Mask R-CNN [14] detector. The system provides an efficient deep network for learning a large number of categories with efficient and fast detection. We proposed several adaptations to Mask R-CNN that improve the learning capability on the domain of traffic signs. Furthermore, we proposed a data augmentation technique based on the distribution of geometric and appearance distortions. As an important contribution, we also present a novel dataset, termed the DFG traffic-sign dataset, with a large number of traffic-sign categories that have a low inter-class and a high intra-class variability. This dataset has been made publicly available together with the code for our improvements, allowing the research community

to make further progress on this problem, and enabling a reliable and a fair comparison of different methods on a large-scale traffic-sign detection problem. We also extensively evaluated our proposed improvements and compared them against the original Faster and Mask R-CNN. Our evaluation on the DFG and the Swedish traffic-sign datasets showed that the proposed adaptations improve the performance of Mask R-CNN in several metrics. This includes improvement in the miss rate of the RPN network for small objects, improvement in the overall recall of the full pipeline for both small and large objects, as well as improvement in the overall performance in terms of the mean average precision.

Our qualitative analysis further revealed how a 2–3% average error rate is reflected in actual detections. This is well demonstrated in Figure 14 where detections of several complex traffic-sign categories are depicted. Overall, we showed that the deep learning based approach is able to achieve extremely good performance for many traffic-sign categories, including several complex ones with large intra-class variability. Large error rates for problematic traffic-sign categories are mostly due to similarity to other categories, wide viewing angles and large occlusions. However, those issues do not pose a problem for the application of maintaining an accurate record of the traffic-sign inventory. They can be mitigated by the detection over several video frames or matching 3D locations from stereo cameras. In particular, this system is already being deployed for traffic-sign inventory management on Slovenian roads. However, the proposed solution is also applicable to other problems requiring the capability of traffic-sign detection such as autonomous driving and advanced driver-assistance systems.

Despite excellent performance of the proposed approach there is still room for improvement. Our analysis revealed that the ideal performance is still not achieved, mostly due to several missed detections that are being lost by the classification network. Future improvements should focus on improving this part of the system.

ACKNOWLEDGEMENTS

The authors would also like to thank the company DFG Consulting d.o.o., in particular Domen Smole, Simon Jud, and mag. Tomaž Gvozdanović for capturing and annotating images, and for their help in creating the dataset.

REFERENCES

- [1] V. Balali, A. A. Rad, and M. Golparvar-Fard, “Detection, classification, and mapping of U.S. traffic signs using Google street view images for roadway inventory management,” *Vis. Eng.*, vol. 3, no. 1, p. 15, 2015.
- [2] K. C. P. Wang, Z. Hou, and W. Gong, “Automated road sign inventory system based on stereo vision and tracking,” *Comput.-Aided Civil Infrastruct. Eng.*, vol. 25, no. 6, pp. 468–477, 2010.
- [3] V. Balali and M. Golparvar-Fard, “Evaluation of multiclass traffic sign detection and classification methods for U.S. roadway asset inventory management,” *J. Comput. Civil Eng.*, vol. 30, no. 2, 2016, Art. no. 04015022.
- [4] J. M. Lillo-Castellano, I. Mora-Jiménez, C. Figuera-Pozuelo, and J. L. Rojo-Alvarez, “Traffic sign segmentation and classification using statistical learning methods,” *Neurocomputing*, vol. 153, pp. 286–299, Apr. 2015.
- [5] M. Haloi. (2015). “A novel pLSA based traffic signs classification system.” [Online]. Available: <https://arxiv.org/abs/1503.06643>

- [6] 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*, vol. 214, pp. 758–766, Nov. 2016.
- [7] R. Timofte, V. Prisacariu, L. Van Gool, and I. Reid, "Combining traffic sign detection with 3D tracking towards better driver assistance," in *Emerging Topics in Computer Vision and Its Applications*. Singapore: World Scientific, 2011, pp. 425–446. doi: 10.1142/8103.
- [8] A. Mogelmose, "Visual analysis in traffic & re-identification," Ph.D. dissertation, Fac. Eng. Sci., Aalborg Univ., Aalborg, Denmark, 2015.
- [9] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. Computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.
- [10] 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 *Proc. IJCNN*, Aug. 2013, pp. 1–8.
- [11] A. Mogelmose, 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. Transp. Syst.*, vol. 13, no. 4, pp. 1484–1497, Dec. 2012.
- [12] F. Zaklouta and B. Stanciulescu, "Real-time traffic-sign recognition using tree classifiers," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1507–1514, Apr. 2012.
- [13] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *Proc. CVPR*, Jun. 2016, pp. 2110–2118.
- [14] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2980–2988.
- [15] S. B. Wali, M. A. Hannan, A. Hussain, and S. A. Samad, "Comparative survey on traffic sign detection and recognition: A review," *Przegląd Elektrotechniczny*, vol. 1, no. 12, pp. 40–44, 2015.
- [16] A. Ellahyani, M. El Ansari, I. El Jaafari, and S. Charfi, "Traffic sign detection and recognition using features combination and random forests," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 1, pp. 683–693, 2016.
- [17] R. Timofte, K. Zimmermann, and L. Van Gool, "Multi-view traffic sign detection, recognition, and 3D localisation," in *Proc. WACV*, Dec. 2009, pp. 1–8.
- [18] S. Šegvić *et al.*, "A computer vision assisted geoinformation inventory for traffic infrastructure," in *Proc. 13th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Sep. 2010, pp. 66–73.
- [19] Z. Huang, Y. Yu, J. Gu, and H. Liu, "An efficient method for traffic sign recognition based on extreme learning machine," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 920–933, Apr. 2017.
- [20] F. Larsson and M. Felsberg, "Using Fourier descriptors and spatial models for traffic sign recognition," *Image Anal.*, vol. 6688, pp. 238–249, May 2011.
- [21] H. Li, F. Sun, L. Wang, and L. Liu, "A novel traffic sign detection method via color segmentation and robust shape matching," *Neurocomputing*, vol. 169, no. 2, pp. 77–88, Dec. 2015.
- [22] X. Yang, Y. Qu, and S. Fang, "Color fused multiple features for traffic sign recognition," in *Proc. ICIMCS*, 2012, pp. 84–87.
- [23] S. Salti, A. Petrelli, F. Tombari, N. Fioraio, and L. D. Stefano, "Traffic sign detection via interest region extraction," *Pattern Recognit.*, vol. 48, no. 4, pp. 1039–1049, Apr. 2015.
- [24] J. Greenhalgh and M. Mirmehdi, "Real-time detection and recognition of road traffic signs," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1498–1506, Dec. 2012.
- [25] G. Overett and L. Petersson, "Large scale sign detection using HOG feature variants," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2011, pp. 326–331.
- [26] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition—How far are we from the solution?" in *Proc. IJCNN*, Aug. 2013, pp. 1–8.
- [27] F. Zaklouta and B. Stanciulescu, "Real-time traffic sign recognition in three stages," *Robot. Auto. Syst.*, vol. 62, no. 1, pp. 16–24, 2014.
- [28] D. Pei, F. Sun, and H. Liu, "Supervised low-rank matrix recovery for traffic sign recognition in image sequences," *IEEE Signal Process. Lett.*, vol. 20, no. 3, pp. 241–244, Mar. 2013.
- [29] Y. Wu, Y. Liu, J. Li, H. Liu, and X. Hu, "Traffic sign detection based on convolutional neural networks," in *Proc. IJCNN*, Aug. 2013, pp. 1–7.
- [30] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "Multi-column deep neural network for traffic sign classification," *Neural Netw.*, vol. 32, pp. 333–338, Aug. 2012.
- [31] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *Proc. IJCNN*, Jul./Aug. 2011, pp. 2809–2813.
- [32] J. Jin, K. Fu, and C. Zhang, "Traffic sign recognition with hinge loss trained convolutional neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 5, pp. 1991–2000, Oct. 2014.
- [33] V. Vukotić, J. Krapac, and S. Šegvić, "Convolutional neural networks for croatian traffic signs recognition," in *Proc. CCVW*, 2014, pp. 15–20.
- [34] D. Ciresan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *Proc. IJCNN*, Jul./Aug. 2011, pp. 1918–1921.
- [35] Y. Zeng, X. Xu, Y. Fang, and K. Zhao, "Traffic sign recognition using deep convolutional networks and extreme learning machine," in *Proc. IScIDE*, vol. 9242, 2015, pp. 272–280.
- [36] M. Haloi. (2015). "Traffic sign classification using deep inception based convolutional networks." [Online]. Available: <https://arxiv.org/pdf/1511.02992.pdf>
- [37] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The German traffic sign recognition benchmark: A multi-class classification competition," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Aug. 2011, pp. 1453–1460.
- [38] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," in *Proc. Int. Conf. Learn. Represent.*, 2014, pp. 1–16.
- [39] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
- [40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.
- [41] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, 2015, pp. 91–99.
- [42] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 936–944.
- [43] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 761–769.
- [44] R. Girshick, I. Radosavovic, G. Gkioxari, P. Dollár, and K. He. (2018). Detectron. [Online]. Available: <https://github.com/facebookresearch/detectron>
- [45] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. (2008). *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results*. [Online]. Available: <http://www.pascal-network.org/challenges/VOC/voc2008/workshop/index.html>
- [46] T. Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Computer Vision—ECCV*, vol. 8693. Springer, 2014, pp. 740–755.



Domen Tabernik received the bachelor's degree in computer and information science in 2010. Since 2010, he has been a Computer Vision Researcher with the Visual Cognitive System Laboratory, Faculty of Computer and Information Science, University of Ljubljana, where he has been enrolled in the Doctoral Program of the Faculty of Computer and Information Science, since 2015, and is working on the topics of compositional hierarchies and deep learning.



Danijel Skočaj received the Ph.D. degree in computer and information science from the University of Ljubljana in 2003. He is currently an Associate Professor at the Faculty of Computer and Information Science, University of Ljubljana, and is also the Head of the Visual Cognitive Systems Laboratory. His main research interests lie in the fields of computer vision, pattern recognition, machine learning, and cognitive robotics; therefore he is involved in basic and applied research in these research areas.