Semantic segmentation-based traffic sign detection and recognition using deep learning techniques

Călin Timbuş, Vlad Miclea, Camelia Lemnaru
Technical University of Cluj-Napoca, Romania
Computer Science Department
{calin.timbus}@student.utcluj.ro {vlad.miclea, camelia.lemnaru}@cs.utcluj.ro

***Abstract*—We present a method for detecting and classifying traffic signs based on two deep neural network architectures. A Fully Convolutional Network (FCN) – based semantic segmentation model is modified to extract traffic sign regions of interest. These regions are further passed to a Convolutional Neural Network (CNN) for traffic sign classification. We propose a novel CNN architecture for the classification step. In evaluating our approach, we contrast the efficiency and the robustness of the deep learning image segmentation approach with classical image processing filters traditionally applied for traffic sign detection. We also show the effectiveness of our CNN-based recognition method by integrating it in our system.**

I. Introduction

Convolutional neural networks have recently become ubiquitous in large-scale image recognition tasks, owing to the exponential advancement in computing power. In addition to the considerable gain in hardware performance, widely available big datasets have contributed towards state-of-the-art improvements. Having pushed the boundaries in several computer vision tasks, such as object classification and detection [1], they have likewise been proven to excel at semantic segmentation. The latter is perhaps one of the biggest challenges of the modern deep learning era.

The detection and classification of traffic signs constitutes one of the key challenges in obtaining a good visual perception of traffic scenes. This stems from the fact that not only is the traffic environment very complex and subjected to continuous changes, such as weather conditions, but also to the aspect of the traffic signs in themselves. The 'yield' traffic sign in Germany is very different from other countries, albeit it may not seem at first sight. Such discrepancies preclude both a facile pathway towards solving this task and the generalization of its solutions.

In this paper we propose a novel traffic sign detection and recognition method. Instead of using traditional image processing methods for traffic sign detection, we train a Convolutional Neural Network (CNN) that outputs a semantic segmentation of the traffic scene. For this purpose we modify a Fully Convolutional Network (FCN) [2] architecture and extract traffic sign regions from the resulting semantic map. The second novelty in our paper is a new CNN architecture for traffic sign recognition, which is trained to classify the traffic sign regions obtained from the aforementioned semantic map.

978-1-5386-8445-0/18/$31.00 ©2018 IEEE

II. Related Work

A. Semantic Segmentation

Although scene segmentation has been traditionally computed by intensity-based methods such as super-pixels [3] [4] [5] or by using geometric depth information through plane fitting [6], the emergence of deep learning techniques has lead to improved results for this task. This boost in performance came in conjunction with the apparition of large training datasets such as Cityscapes [7], Kitti [8] or Mapillary [9]. The authors of FCN [2] modified known classification networks such as VGG [10] or GoogleNet [11] to perform semantic segmentation, adapting them to upsample the output feature maps and making them fully convolutional. A different approach has been proposed by the authors of SegNet [12]. They used an encoder-decoder CNN architecture, that uses max-pooling blocks for under-sampling and then performs a finer unpooling for upsampling the resulting segmentation. Besides the improvement in accuracy, segmentation methods based on deep learning [13], [14] bring the advantage of speed (as in time performance) – semantic maps are generated in (or close-to) real-time on a regular GPU.

B. Traffic Sign Recognition

Road sign classification is an extremely important problem in visual perception of traffic scenes. Classic image processing techniques using RGB [15], HOG [16], or Haar-like [17] features can bring satisfactory results for this problem. However, state of the art benchmarks [18] [19] show that learning-based methods using boosting [20] or convolutional neural networks [21] can surpass traditional counterparts in accuracy, speed and robustness. For this particular task, the method introduced by Mao et. al. [22] obtains highly accurate results by using a hierarchical CNN. The main idea behind their approach is to divide the traffic sign recognition problem into several subtasks, each such task being solved by a particular smaller CNN.

An end-to-end approach for traffic sign detection and classification is presented in [23]. The approach utilizes a specialized CNN, which achieves both detection and classification. The authors have also created a large dataset for benchmarking traffic detection and classification, which comprises of more than 100.000 images and 127 different traffic labels (the TENCENT dataset).

III. Our Approach

Figure 1 presents the proposed pipeline: we initially feed the system with a 3-channel 1024x512 input image. The image is then passed through our modified FCN8s architecture. The semantically segmented result is sent to the ROI extraction module, which is responsible for cropping the candidate traffic signs. The delineated region at the previous step is fed into a classification network, which assigns a label to the traffic sign.

A. Image Segmentation Model

1) Description of the Segmentation Dataset: The dataset used to train the segmentation model is CityScapes [7]. The dataset comprises 19 different labels, including the traffic sign class. The photographs are taken from various cities around Germany and even Switzerland (Zurich) and France (Strasbourg). It consists of 2975 training set photos and 500 photos are reserved for the validation set. All of these are either fine-grained or coarse-grained. Although both categories can be used for the purpose of training and validation, we opted for using only the fine-grained images. Each image has a dimension of 2048 (width) x 1024 (height). Out of the 1.821.900.000, which is the total number of pixels contained in the entire dataset, considering the halving resize, 4,78% account for traffic signs. It is crucial to mention that out of these traffic signs, many do not represent a particular class, but rather fingerposts or similar unrelated sign posts. Therefore, it immediately follows that relevant traffic signs represent a considerably small pixel percentage.

2) Image Segmentation Architecture: We adopted an incremental approach in developing the segmentation model. We initially focused on the SegNet architecture [12], since it is much more memory-efficient than FCNs, as the max pooling indices are transfered to the decoder to improve the segmentation resolution. However, as expected, we obtained unsatisfactory results on the CityScapes dataset [7], which led us to consider a more complex model, leading us to the FCN8s architecture [2].

The FCN8s combines predictions from the final layer, the third and the fourth pooling layers which provide much better precision. In contrast to the original architecture, we modified standard dilation rate of 1 in the second 4096 feature maps convolutional layer to a dilation rate of 4. The rationale behind our decision is that dilated convolution increases the image resolution, which is a crucial aspect, particularly for small object instances, such as traffic signs. Due to the exponential increase in the field of view, we observed an increase of 0.7% accuracy, which represents a significant gain in case of the segmentation task.

Considering the initial size of the images, we gradually increased the dimensions of the photos. After several tests, we arrived at a size of 1024 x 512. The bigger memory of the video card also allowed us to gradually increase the batch-size. Initially, we increased the batch-size to a dimension of 2. However, since during training we observed a strong drop in accuracy tendency after a certain number of epochs ($\sim$ 50 epochs), we decided to increase the batch-size to 4, which led to an increase in the overall accuracy and eliminated the previously mentioned issue altogether.

3) Training Considerations: We initially trained for 50 epochs ($\sim$ 30 hours) with the SegNet architecture. As an optimizer, our choice was Adam [24]. The training for the FCN8s model two times took longer. Both models were trained on a GTX 1080TI@11Gb, which allowed us to train with bigger mini-batches (instead of traditional SGD), leading to better overall results.

FCN8s was able to detect all the classes within the dataset and rendered much fewer false positives than SegNet. In addition to this, FCN8s succeeded in identifying much better less represented classes (such as traffic signs or semaphores), labels which SegNet failed to detect. We have not assessed the Intersection over Union (IoU) achieved by the segmentation model. The reason for our decision is that we are not interested in attaining a very high IoU, but to have a reasonable delineated/outlined region of interest. The official benchmarks on the CityScapes dataset report a global IoU class average of 65.3, while at the instance level class they report a value of 41.6 for the FCN8s model.

During the initial epochs of the training our model was underfitting. After 6 epochs it stopped underfitting. After 60 epochs, we obtained the best accuracy which was 86,84%. We trained the neural network with the best model accuracy checkpoint. We used categorical cross-entropy as a loss function.

B. Image Classification Models

1) Description of the Classification Dataset: The dataset used for the classification task is the GTSRB dataset (German Traffic Sign Recognition Benchmark) [25]. The dataset contains 43 different classes, consisting of more than 50000 photos.

The histogram of the class distribution is shown in Figure 2. As we can observe, the data is highly imbalanced (2448 instances for the best represented class, as opposed to 216 instances for the least represented class). This observation is essential, as the training of the CNN is strongly influenced by the distribution of the classes.

2) The Image Classification Architectures: The convolutional neural network that was used for image classification is inspired from [10]. Similar to a classical VGG, we use convolutional layers with ReLU as an activation function, followed by a down-sampling (max pooling) step.

Both architectures use a kernel of 3x3 at the convolutional layer level, while max pooling is performed over a 2x2 pixel window, using a stride of two. The first configuration contains four stacked pairs of Convolutional-Max-pooling-ReLU activation. We started with 32 feature detectors at the level of the first convolutional layer and doubled the feature maps at each following pair. The final layer is the Softmax layer, which contains one unit for each class. The same final layer is applied to both architectures. In the first configuration,

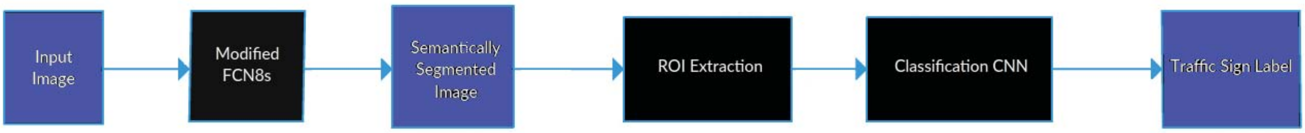

Fig. 1: Workflow of the proposed method

TABLE I: CNN Architectures

ConvNet Configurations	
ReLU	eLU
Conv3-32	Conv3-32
MaxPooling(2x2)	
Conv3-64	Conv3-64
MaxPooling(2x2)	
Conv3-128	Conv3-128
MaxPooling(2x2)	
Conv3-256	Conv3-256
MaxPooling(2x2)	
Hidden Layer(128)(ReLU)	Hidden Layer(256)(eLU) Hidden Layer(256)(eLU)
Dense(43)(Softmax)	

these pairs are followed by a single hidden layer of 128 neurons, upon which we used the ReLU activation function.

The second architecture comes with a different configuration at the hidden layer level: instead of one hidden layer with 128 neurons, we used two hidden layers with 256 units. As compared to the former architecture, the activation function for both layers is the exponential Linear Unit (eLU). Our choice was motivated by the work of [26].

3) Training Considerations: We have trained both architectures on the GTSRB dataset [25].

We experimented with two different configurations (described in Table I). The input to both architectures is a fixed-size 64x64 RGB channel. The preprocessing step consisted in applying several image augmentation techniques: scaling, shearing, zooming, rotation, width and height shift.

Moreover, to address the imbalanced class distribution, we applied random oversampling with replacement for each underrepresented class, to reach the number of instances of the best represented class (i.e. 2448 instances). This procedure has been proven to work better than undersampling or class-weighted training [27]. The next section provides comparisons of the results on balanced and imbalanced datasets.

In each training scenario, we used Adam [24] as an optimizer, with a batch-size of 32. The loss function was categorical cross-entropy, while the final metric was classification accuracy.

The training time lasted approximately 12 hours on an Intel I5-6500 @3.2GHz. By contrast, while training the neural network on GTX 1050TIm, boasting 4GB of VRAM, the training time took four times less (97 seconds per epoch vs 405 seconds per epoch).

We experimented with a dropout rate of 10% after each hidden layer, which did not improve the overall accuracy, but rather contributed towards underfitting. We have also tried other regularization techniques, such as L1 and L2 regularization techniques. Neither the previous two nor the batch normalization contributed to overall accuracy improvement. Although we trained with a batch-size of 32, as proven by [28], a small batch-size, in conjunction with batch normalization does not render good results (note that we added a layer of batch normalization after each convolutional layer).

IV. Evaluation

In this section we report the results obtained by the two classification architectures, on the GTSRB dataset and we analyze the efficacy of the deep learning segmentation model as opposed to the traditional filter-wise approach, in the context of the entire processing pipeline. Performing a quantitative analysis on the entire pipeline is not possible at the moment, due to the absence of labeled city landscape images.

A. Evaluation of the Detection System

In this section we evaluate the results of our detection system. As the test dataset for the classification already contains cropped traffic-sign images, we had to create a dataset which contained real traffic scenes to contrast the detection approaches. Therefore, we created a dataset to test the efficiency of the deep learning semantic segmentation approach against the traditional strategy, which we implemented in OpenCV. The dataset contains 55 photos taken in the city of Wuppertal (part of the CityScapes dataset). The traditional approach consists in applying several filters, such as color, contour and shape matching. As traffic signs exhibit quite specific properties, such as having a particular circular or square shape, being blue, red or yellow, we implemented these aforementioned filters to work together for the task of detecting traffic signs in a road-environment.

However, the traffic environment is too complex and prone to constant change; for example, a cast of shadow upon a traffic sign may result in this traditional approach not being able to detect a traffic sign. In addition to this probable issue, rectangular red shapes could also constitute a strong false candidate, hence the lack of generalization of the traditional approach. Such examples are innumerable and we expect them to cause such an approach to perform poorly, particularly in a traffic-scene environment.

Our intuition is confirmed by the results obtained on our dataset. While the classical approach yields a 28% accuracy, the semantic segmentation with deep learning produces a much higher accuracy value of 89%. Note that the deep learning model is also able to capture different traffic signs such as

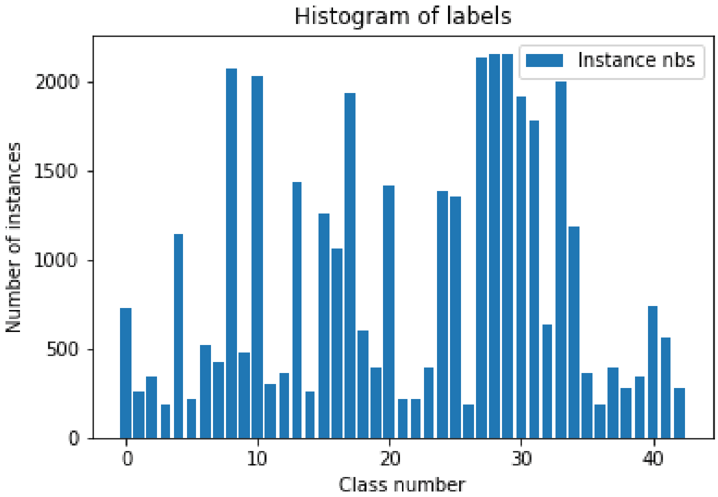

Fig. 2: Class distribution across the GTSRB dataset

signposts. Moreover, it is also able to capture traffic signs which are not in the exact proximity (several meters) of the camera. Furthermore, the segmentation model is not sensitive to the position of the traffic sign with respect to the camera. While the traditional approach requires affine transformations in many situations for the traffic sign to be detected, such a prerequisite is absent when employing the deep learning model.

Figure 3 illustrates the difference in detection quality between the traditional and the deep learning segmentation approach. On each line, the left photo represents the output of the detection algorithm using the traditional approach and the right part of each line represents the output of the ROI Extraction from the proposed workflow. Although in the first example the segmentation model detects a false positive (in front of the pharmacy), it is able to better determine the ROI for the priority road sign as well as the two ”No entry for vehicular traffic” signs on the opposite street. In addition to this, it also outlines the bus station traffic sign. The first example demonstrates the inability of the traditional approach when it is subjected to scenes where the traffic signs are not perpendicular on the camera's point of view.

The second example proves that the traditional approach is very sensitive to lighting. In a cloudy environment, it is unable to detect blue traffic signs which are in close proximity to the driver's camera. The segmentation model detects both the closer traffic sign and signposts, as well as the two traffic signs at a greater distance. However, the two traffic signs partially covered by the trees are not detected by the segmentation model either.

B. Evaluation of the Classification

The results in Table II demonstrate that ReLU performs slightly better on the imbalanced dataset. However, for both architectures, the underrepresented classes are poorly classified. For example, the 'caution wet' class is often confused with 'wild animals' sign.

TABLE II: Accuracy obtained by training on the imbalanced dataset

Imbalanced Dataset	ReLU	eLU
Training Set	97.3%	97.8%
Validation Set	96.95%	96.77%
Test Set	94.02%	94.01%

TABLE III: Accuracy obtained by training on the balanced dataset

Balanced Dataset	ReLU	eLU
Training Set	97.3%	98.3%
Validation Set	95.3%	96.2%
Test Set	94.02%	94.15%

The mean of accuracies on each class (computed on the test set) for the ReLU architecture is 89%, while the eLU model yields 88,4% in this regard. On the balanced dataset, the eLU network renders better results. The mean of accuracies on the balanced dataset reached 90% for the ReLU based architecture, and 91,5% for eLU.

We conclude the image classification evaluation with the observation that oversampling was a very good approach for dealing with the imbalanced dataset problem. Furthermore, we emphasize the fact that eLU performs better in all regards, with respect to the balanced dataset. A final observation to be made is that the dice metric could have been used instead of accuracy (on the imbalanced dataset).The F1-score similar dice is more relevant in case of the imbalanced dataset, for penalizing classification errors proportionally to the number of instances within a specific class.

C. Evaluation of the Entire System

As the results of the detection module indicate, the segmentation neural network provided significantly better detection

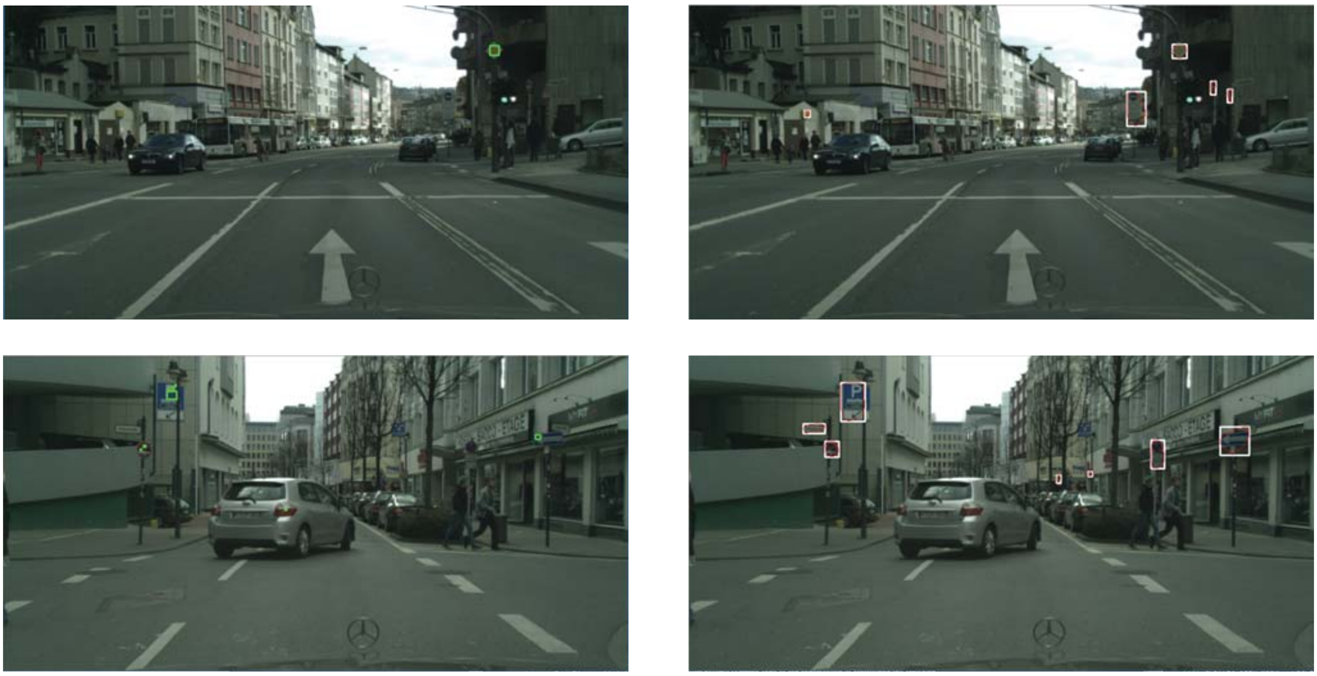

Fig. 3: Two examples of ROI detection using the traditional approach (left) and the deep learning segmentation model (right)

results. When coupling the entire system, it is essential to mention the crucial role that the ROI Extraction has. The ROI Extraction module does not merely extract the presumable traffic signs; in the first step it draws contours of the supposed traffic signs. The next step is to sort in descending order by contour area. The motivation for our choice is that small color blobs do not represent traffic signs; even if there is a slim probability of them representing one, they are at a considerable distance from the driver to be taken into consideration. Thus, we eliminate the color blobs. Then, we apply the convex hull.

The images in Figure 4 illustrate the results obtained by applying the entire flow. The first line corresponds to the input photos. The next line represents the semantically segmented colored output. Albeit we could have assigned a color only to the traffic sign ids, we wanted to demonstrate the completeness of the modified FCN8s model. Following the ROI extraction and applying the aforementioned transformations, we obtain the labels which are displayed close to their corresponding traffic signs (the last row).

In Figure 4, the contour drawn upon the ROI extraction is not exactly smooth; it immediately follows that if we were to extract the traffic sign exclusively according to the contour, we would partly extract it. This is due to the fact that our modified segmentation model does not perfectly segment instances. In spite of this fact, by applying the convex hull upon the contour (which in turn is composed by a set of 2D points), we obtain a much better region of interest. Even if we extract a bigger region of interest, provided that it contains the desired traffic sign, it aids the following classification module when labelling cropped images. The quintessence is that the convex hull compensates for the lack of accuracy of the segmentation model. Its employment over the generic contour drawing technique is of paramount importance, particularly when small object instances such as traffic signs are concerned.

TABLE IV: Detection results (% of accuracy)

Dataset	Accuracy
Traditional Approach Filters	28%
Modified FCN8s	89%

The forward pass time takes approximately 0.010 seconds under most circumstances. The ROI extraction module, in conjunction with the classification one takes 0.020 seconds at most. Thereupon a forward pass through the entire pipeline requires 0.030 - 0.035 seconds.

We also need to outline some shortcomings. The segmentation module detects several traffic signs which are not included in the GTSRB dataset. We tried the thresholded approach to reject traffic signs which do not belong to our dataset. This approach works only under some circumstances and cannot be considered as a solution to this problem. The best way to solve such an issue would be the creation of a more comprehensive dataset.

The second drawback refers to the segmentation module. Traffic signs which are very close to one another (on a pole) are very likely to be considered as a whole. Such a problem may be solved by choosing a more complex model to better segment the images. Thus, the ROI extraction module would extract two instead of one traffic sign under a similar given circumstance.

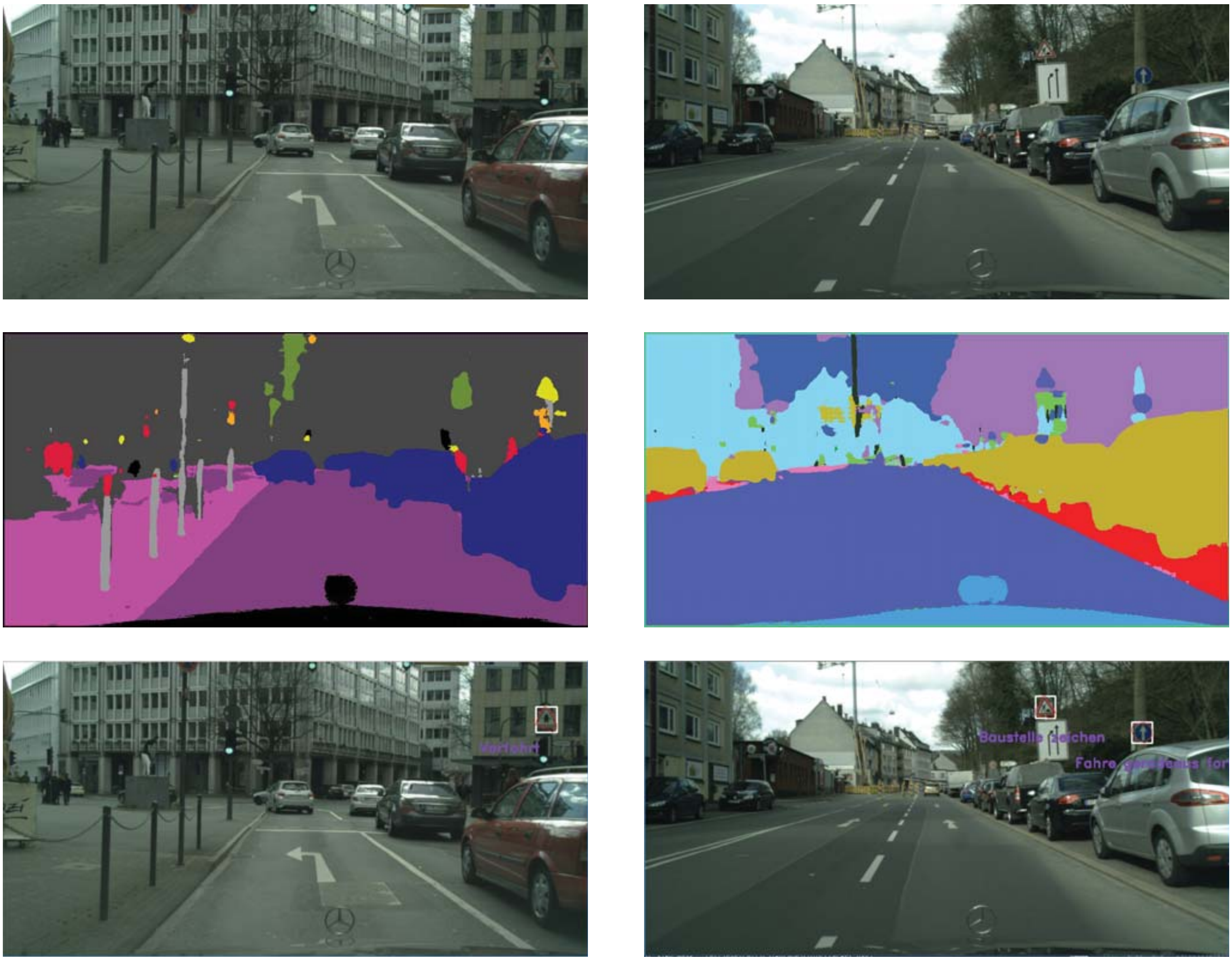

Fig. 4: Two examples of images processed with the full detection and classification flow

V. Conclusions

In this paper we presented a deep learning-based traffic sign detection and recognition using semantic segmentation. An RGB image is fed into a modified FCN8s which outputs a semantically segmented image. The FCN8s is modified by replacing the last convolutional layer with dilated convolution (of rate 4) instead of the traditional convolution (of rate 1). This modification generated an increase of 0.7% in the final accuracy. Although the accuracy may have not increased by a great margin, it improved the segmentation quality of the less well-represented object categories, such as traffic signs and semaphores. The semantically segmented image is then passed on to a Region Of Interest (ROI) module, which extracts the candidate traffic signs. These extractions are cropped and sent to a classification module which assigns a label to each cropped ROI image received. The classification module consists of two different models (eLU) and (ReLU) that we created in order to contrast the efficiency of the former activation function to the latter. To the best of our knowledge, the proposed approach is novel. The possible improvements to the actual implementation have been discussed in the previous section.

References

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg, "SSD: single shot multibox detector," in *ECCV (1)*, ser. Lecture Notes in Computer Science, vol. 9905. Springer, 2016, pp. 21–37.

[2] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April 2017.

[3] X. Ren and J. Malik, "Learning a classification model for segmentation," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Oct 2003, pp. 10–17 vol.1.

[4] J. Mehra and N. Neeru, "A brief review : Super-pixel based image segmentation methods," 2016.

[5] X. Tian, L. Jiao, L. Yi, K. Guo, and X. Zhang, "The image segmentation based on optimized spatial feature of superpixel," *J. Vis. Comun. Image Represent.*, vol. 26, no. C, pp. 146–160, Jan. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.jvcir.2014.11.005

[6] L. Guan, T. Yu, P. Tu, and S. N. Lim, "Simultaneous image segmentation and 3d plane fitting for rgb-d sensors; an iterative framework," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, June 2012, pp. 49–56.

[7] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[9] G. Neuhold, T. Ollmann, S. R. Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *2017 IEEE International Conference on Computer Vision (ICCV)*, vol. 00, Oct. 2018, pp. 5000–5009. [Online]. Available: doi.ieeecomputersociety.org/10.1109/ICCV.2017.534

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[12] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec 2017.

[13] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, "Enet: A deep neural network architecture for real-time semantic segmentation," *CoRR*, vol. abs/1606.02147, 2016. [Online]. Available: http://arxiv.org/abs/1606.02147

[14] E. Romera, J. M. Álvarez, L. M. Bergasa, and R. Arroyo, "Efficient convnet for real-time semantic segmentation," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 1789–1794.

[15] M. Shahbazi, M. Satari, S. Homayouni, and M. Saadatseresht, "Object-based traffic sign detection and recognition from integrated range and intensity images," 12 2011.

[16] C. Yao, F. Wu, H. j. Chen, X. l. Hao, and Y. Shen, "Traffic sign recognition using hog-svm and grid search," in *2014 12th International Conference on Signal Processing (ICSP)*, Oct 2014, pp. 962–965.

[17] Z. Li, C. Dong, L. Zheng, and L. Liu, "Traffic signs detection based on haar-like features and adaboost classifier," 06 2013, pp. 1128–1135.

[18] 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 *International Joint Conference on Neural Networks*, no. 1288, 2013.

[19] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," vol. 32, pp. 323–32, 02 2012.

[20] X. Baro, S. Escalera, J. Vitria, O. Pujol, and P. Radeva, "Traffic sign recognition using evolutionary adaboost detection and forest-ecoc classification," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 1, pp. 113–126, March 2009.

[21] P. Sermanet and Y. LeCun, "Traffic sign recognition with multi-scale convolutional networks," in *The 2011 International Joint Conference on Neural Networks*, July 2011, pp. 2809–2813.

[22] X. Mao, S. Hijazi, R. Casas, P. Kaul, R. Kumar, and C. Rowen, "Hierarchical cnn for traffic sign recognition," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 130–135.

[23] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu, "Traffic-sign detection and classification in the wild," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2110–2118.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[25] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Networks*, no. 0, pp. –, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893608012000457

[26] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (elus)," *CoRR*, vol. abs/1511.07289, 2015.

[27] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *CoRR*, vol. abs/1710.05381, 2017. [Online]. Available: http://arxiv.org/abs/1710.05381

[28] S. Ioffe, "Batch renormalization: Towards reducing minibatch dependence in batch-normalized models," *CoRR*, vol. abs/1702.03275, 2017. [Online]. Available: http://arxiv.org/abs/1702.03275